Exploiting RS/6000 SP Security: Keeping It Safe

- Learn PSSP 3.2 security features step by step
- Discover time-saving troubleshooting tips
- Explore additional security tools

Abbas Farazdel
Chris DeRobertis
Marc Genty
Maarten Kreuger
Michael Wilkop

ibm.com/redbooks
Exploiting RS/6000 SP Security: Keeping It Safe

September 2000
First Edition (September 2000)

This edition applies to IBM Parallel System Support Programs for AIX (PSSP) Version 3, Release 2 for use with AIX 4.3.3.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B  Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2000. All rights reserved.
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Contents

Preface .................................................................................. ix
The team that wrote this redbook .............................................. ix
Comments welcome ............................................................... xi

Part 1. Basic security concepts and the SP ................................. 1

Chapter 1. Introduction ....................................................... 3
1.1 General security considerations ........................................ 3
1.2 SP security design ......................................................... 4
   1.2.1 The SP Trusted Services ....................................... 4
   1.2.2 The SP central point of management ......................... 5
1.3 How to use this book .................................................... 6
   1.3.1 How much security do you need? ............................. 6
   1.3.2 Choosing the right security technology for your environment 6
   1.3.3 Proceeding from here ............................................. 8

Chapter 2. SP security concepts and terminology .................... 11
2.1 Authentication and authorization ...................................... 11
2.2 Restricted Root Access (RRA) ....................................... 13
2.3 SP security attributes .................................................. 14
2.4 SP security states ......................................................... 16
2.5 Each SP partition has its own security state ...................... 17

Part 2. Planning for DCE ......................................................... 21

Chapter 3. What you need to know if you plan to use DCE .......... 23
3.1 The DCE Secure Core ................................................. 23
3.2 Synchronized time ....................................................... 24
3.3 The concept of a DCE cell ............................................ 25
3.4 DCE administrative principals ....................................... 26
3.5 DCE groups for PSSP .................................................. 27
3.6 DCE keyfiles for PSSP ............................................... 29
3.7 Split configuration for DCE clients ................................. 47
3.8 The DCE security server bootstrap file ......................... 48
3.9 Network interface routing considerations ....................... 49
3.10 Options for setting up DCE ........................................... 49
3.11 An important note about partitioning .............................. 52
3.12 Suggested reading for DCE ........................................ 52
Part 3. SP security ..........................55

Chapter 4. Implementing a specific security state ..........................57
4.1 Overview of the examples ..............................................57
4.2 Our lab environment .................................................58
4.3 A template for the security installation process .................59
4.4 Example 1: Minimal security ..........................................60
  4.4.1 Example 1: Planning considerations ..........................61
  4.4.2 Example 1: Installation steps ..................................61
4.5 Example 2: Compatibility security .................................66
  4.5.1 Example 2: Planning considerations ..........................66
  4.5.2 Example 2: Installation steps ..................................67
4.6 Example 3: Simple DCE cell.........................................71
  4.6.1 Example 3: Planning considerations ..........................72
  4.6.2 Dealing with mksysb images in a DCE environment ...........73
  4.6.3 Setting up the CWS as DCE server ............................74
  4.6.4 Example 3: Installation steps ..................................81
4.7 Example 4: Advanced DCE cell .....................................91
  4.7.1 Example 4: Planning considerations ..........................92
  4.7.2 Adding the SP administrator DCE principal to the DCE cell ....93
  4.7.3 Example 4: Installation steps ..................................95
  4.7.4 Setting up a secondary CDS server on a node ...............106
  4.7.5 Setting up a DCE Security Replica Server on a node .......110
4.8 Example 5: Existing DCE cell ......................................112
  4.8.1 Example 5: Planning considerations ..........................113
  4.8.2 Setting up a DCE client on the CWS .........................114
  4.8.3 Example 5: Installation steps ..................................118

Chapter 5. Changing from one security state to another .............131
5.1 Security state transitions ..........................................131
5.2 Our lab environment .................................................132
5.3 Moving from Compatibility to DCE ................................133
  5.3.1 From Compatibility to DCE: Planning considerations ........134
  5.3.2 From Compatibility to DCE: High-level outline ...............134
  5.3.3 From Compatibility to DCE: Step-by-step procedure ..........135
  5.3.4 From Compatibility to DCE: Further considerations ..........160
5.4 Moving from Compatibility to Minimal ............................160
  5.4.1 From Compatibility to Minimal: Planning considerations ....161
  5.4.2 From Compatibility to Minimal: High-level outline ..........161
  5.4.3 From Compatibility to Minimal: Step-by-step procedure ....162
5.5 Moving from Minimal to Compatibility ............................169
  5.5.1 From Minimal to Compatibility: Planning considerations ....170
  5.5.2 From Minimal to Compatibility: High-level outline ..........170
5.5.3 From Minimal to Compatibility: Step-by-step procedure  .... 171

Chapter 6. Restricted Root Access (RRA) .......................... 183
6.1 How does it work? .............................................. 184
   6.1.1 AIX and PSSP remote commands ....................... 184
   6.1.2 The sysctl facility ..................................... 185
6.2 Limitations .................................................. 186
   6.2.1 Coexistence ........................................... 186
   6.2.2 VSD/GPFS ............................................. 186
   6.2.3 HACMP ............................................... 186
   6.2.4 HACWS ............................................... 187
   6.2.5 Boot/install servers .................................. 187
   6.2.6 Ecommands ........................................... 188
   6.2.7 System management commands ......................... 188
   6.2.8 Additional security implications .................... 189
6.3 RRA enablement - General steps ................................ 189
   6.3.1 Spltten and RRA ..................................... 190
   6.3.2 Security state DCE .................................. 191
   6.3.3 Security state Compatibility ......................... 192
   6.3.4 Security state Minimal .............................. 194
   6.3.5 Multiple Boot/install Servers in RRA ............... 195
   6.3.6 Node install customization ............................ 196
6.4 Troubleshooting tips ....................................... 196

Chapter 7. Example security scenarios ............................. 199
7.1 Open environment ........................................... 200
7.2 Standard commercial systems ............................... 201
   7.2.1 NIS and NFS risks ................................... 203
   7.2.2 NIS+ and Secure NFS ................................. 203
   7.2.3 Disabling services .................................. 204
   7.2.4 Restricting user access ............................... 205
   7.2.5 Do not allow automatic login authorization .......... 206
   7.2.6 Securing X11 connections ............................ 206
   7.2.7 Securing your backups ................................ 208
7.3 Mission-critical commercial systems ......................... 209
   7.3.1 Restrict access to the system by setting up IP Security 210
   7.3.2 Using encrypted login sessions ....................... 213
   7.3.3 Umask settings for root ................................ 215
7.4 Internet connected systems ................................. 216
   7.4.1 Set up DCE to properly authenticate PSSP daemons 218
   7.4.2 Enhance filtering by IPsec ............................ 218
   7.4.3 Using TCP Wrappers to restrict access to services.. 220
   7.4.4 Sendmail ............................................. 221
7.4.5 DCE user management on an SP ........................................ 223
7.5 Special considerations for high-security environments ................. 224
  7.5.1 Using DCE as the only authentication method ...................... 225
  7.5.2 Installing or restoring a node in a safe way ...................... 225
7.6 Intrusion detection and monitoring ........................................ 226
  7.6.1 Trusted Computing Base (TCB) ..................................... 227
  7.6.2 Tripwire ....................................................... 228
  7.6.3 SATAN .......................................................... 229
  7.6.4 The Deception Toolkit (DTK) .................................... 230
  7.6.5 COPS ............................................................ 231
  7.6.6 TAMU’s Tiger ................................................... 232
  7.6.7 Log files you should check ...................................... 233

Part 4. Troubleshooting .......................................................... 235

Chapter 8. Problem isolation and resolution .................................. 237
  8.1 PSSP security installation and configuration ............................ 240
    8.1.1 Scripts setupdce or config_spsec did not run successfully .... 241
    8.1.2 Authentication methods cannot be enabled on a node .............. 243
  8.2 Steady-state run time ................................................ 244
    8.2.1 Disjoint security settings (authentication methods) ............... 245
    8.2.2 User identities and access controls ................................ 268
    8.2.3 Other events .................................................... 342
  8.3 DCE cell reconfiguration impacts ........................................ 353
    8.3.1 compat/k5:k4:std ............................................. 354
    8.3.2 dce:compat/k5:k4:std ....................................... 355
    8.3.3 dce/k5:std .................................................... 357
    8.3.4 dce/k5 .......................................................... 360
  8.4 Temporarily disabling dce/k5 when DCE is not available .............. 363
    8.4.1 compat/k5:k4:std ............................................. 364
    8.4.2 dce:compat/k5:k4:std ....................................... 366
  8.5 DCE attributes and Per Node Key Management (PNKM) ................... 368
    8.5.1 DCE attributes that impact PNKM ................................ 369
    8.5.2 Other factors that impact PNKM ................................ 370
    8.5.3 Displaying DCE attributes ..................................... 371
  8.6 Using PNKM to diagnose problems ...................................... 373
    8.6.1 Examples ....................................................... 375
  8.7 Recovering from a missing or corrupt PSSP DCE keyfile ............... 384
    8.7.1 Keyfile differences .......................................... 385
    8.7.2 Missing keyfile example ..................................... 386
    8.7.3 Re-creating missing keyfile .................................. 387
  8.8 DCE client/server problem determination ................................ 390
    8.8.1 DCE clients cannot communicate with DCE servers ............... 391
8.8.2 General DCE checkpoints ........................................ 405
8.9 Other DCE problem determination tips ............................... 406
  8.9.1 DCE /etc/dce/cfg.dat file ..................................... 406
  8.9.2 DCE CDS client configuration failure .......................... 410
  8.9.3 DCE local configuration - Unable to get key version .......... 412
8.10 DCE and Kerberos V4 credentials .................................. 416
  8.10.1 Credentials cache locations .................................. 417
  8.10.2 Typical credential states .................................... 426
  8.10.3 Atypical credential states ................................... 429
8.11 AIX and PSSP remote commands .................................... 435
  8.11.1 Documentation/reference materials .......................... 436
  8.11.2 About the AIX authentication methods ........................ 438
  8.11.3 About the remote command protocols ........................ 439
  8.11.4 AIX remote command message summary ......................... 441
  8.11.5 AIX and PSSP remote commands ................................ 442
  8.11.6 Remote command client and server must share a protocol .. 443
  8.11.7 AIX and PSSP remote command debug tips/checklist .......... 452
  8.11.8 The .k5login file demystified ................................ 454
  8.11.9 Zero length authorization files ................................ 456
  8.11.10 Using s1term to log in to a node ............................ 461
  8.11.11 The AIX K5MUTE variable .................................... 465
  8.11.12 Additional DCE/k5 AIX remote errors ........................ 469
8.12 How to clean up a failed DCE node installation or customization .............................................................................. 478
  8.12.1 Unconfigure the node (local-only) ............................. 480
  8.12.2 Unconfigure the node (admin-only) ............................ 480
  8.12.3 Reconfigure the node (admin only) ............................ 481
  8.12.4 Reconfigure the node (local-only) ............................. 483

Appendix A. How Kerberos works ......................................... 485
  A.1 Kerberos keys and initial setup .................................... 486
  A.2 Authenticating to the Kerberos server ............................ 487
  A.3 Authenticating to an application server .......................... 489

Appendix B. Security command and tool reference ..................... 493

Appendix C. Using the additional material .............................. 499
  C.1 Using the CD-ROM or diskette ...................................... 499
    C.1.1 System requirements for using the CD-ROM or diskette .... 499
    C.1.2 How to use the diskette ....................................... 500
  C.2 Locating the additional material on the Internet ............... 500
Preface

This IBM Redbook explains how to exploit the enhanced security features of PSSP 3.2. It walks you through the process of moving from where you are today to making full use of the new security features. Similarly, if you are already in a trusted environment, it shows you how to turn off many of the security features you may not need.

Much of this redbook is written in a cookbook style. It is primarily aimed at Information Technology (IT) professionals responsible for managing and securing SP systems. The focus is on installing, configuring, administering, and troubleshooting security technologies ranging from standard AIX to Kerberos 4, DCE, Restricted Root Access, and beyond. The reader should be familiar with both AIX and PSSP in an SP environment before reading this redbook.

The subjects discussed in this redbook include:

- Planning for SP security
- Implementing SP security states
- Moving from one security state to another
- Planning for and implementing Restricted Root Access (RRA)
- Packet filtering by IPsec
- NIS and NFS security risks
- Intrusion Detection tools: TCB, Tripwire, and The Deception Toolkit
- Additional security tools: TCP Wrappers, Postfix, COPS, SATAN, and Tiger
- DCE user management in an SP environment
- Configuring DCE clients
- Creating DCE Security and CDS Master and Replica servers

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Abbas Farazdel is an SP System Strategist, Technical Consultant, and Senior Project Manager at the International Technical Support Organization, Poughkeepsie Center. Before joining the ITSO in 1998, Dr. Farazdel worked
in the Global Business Intelligence Solutions (GBIS) group at IBM Dallas as an Implementation Manager for Data Warehousing and Data Mining Solutions and in the Scientific and Technical Systems and Solutions (STSS) group at the IBM Thomas J. Watson Research Center as a High Performance Computing Specialist. Dr. Farazdel holds a Ph.D. in Computational Quantum Chemistry and an M.Sc. in Computational Physics from the University of Massachusetts.

Chris DeRobertis has been the IBM RS/6000 SP, PSSP Functional Verification Test (FVT), Security Team Lead since 1997. His responsibilities include developing the overall FVT security test strategy, testing specific PSSP components and their interoperability with IBM AIX DCE, and working closely with the PSSP software security developers on the architecture, implementation, and delivery of PSSP security services. Chris’ duties also include working with other PSSP test and development areas to develop security test strategies, debugging and corrective action analysis, and interdepartmental security test projects. In addition, Chris works closely with IBM AIX and DCE security teams on solutions that span PSSP, AIX, and DCE. Prior to his FVT position, Chris was a co-team lead for an IBM Global Services Web solutions and services department where he worked on projects for large-scale e-commerce accounts and several large IBM U.S. intranet accounts.

Marc Genty is a Systems Management Integration Professional working for IBM Global Services at the Western Geoplex Service Delivery Center in Boulder, Colorado. He has worked on UNIX environments for 10 years and is an RS/6000 Certified Advanced Technical Expert (CATE). He holds a BS degree in Manufacturing from Colorado State University. His areas of expertise include UNIX, AIX, RS/6000 SP, HACMP, and DCE/DFS. He is currently the SP complex lead for the Global Web Architecture (GWA) Internet complex in Boulder.

Maarten Kreuger is an IT specialist in the AIX software services department of IBM Global Services in The Netherlands. He joined IBM after studying electrical engineering at the Hogeschool in Alkmaar. His areas of expertise include SP systems, performance tuning, and HACMP.

Michael Wilkop is a IT specialist in the Systems Management and Networking Services division of IBM Global Services, Germany. Since joining IBM in 1996, he has provided support for AIX and RS/6000 SP. He holds a degree in electrical engineering from the University of Bochum. His areas of expertise include UNIX, AIX, RS/6000 SP, HACMP, and ADSM.

Thanks to the following people for their invaluable contributions to this project:
Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 541 to the fax number shown on the form.
- Use the online evaluation form found at ibm.com/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com
Part 1. Basic security concepts and the SP
Chapter 1. Introduction

The subject of computer and network security is huge, and many volumes have been written about it. It is not our intention to teach you the fundamentals of security here. We have provided references to several fine books on the subject at the end of this redbook. Instead, the goal of this book is to offer practical information on how you can implement security on an SP system. In addition to covering the built-in security features of the SP, we show you some other tools that you can use to enhance and augment your SP security environment. This redbook in no way claims to be the final word on SP security. Rather, it should be viewed as a starting point for developing and implementing your overall SP security policies and environment.

1.1 General security considerations

With respect to security, an SP system can, for all practical purposes, be viewed as a cluster of RS/6000 workstations connected to one or more LANs, with the control workstation (CWS) serving as the central point of control for the entire system. The SP system is exposed to the same security threats as any other networked cluster of workstations.

At the simplest level, computer and network security come down to a trade-off between trust and money:

- Trust
  There is no such thing as absolute security, as much as we, system administrators, would like to believe otherwise. At some point, you have to have a certain level of trust in the environment in which your system operates. It is really a question of acceptable risk.

- Money
  In general, the more trust you want, the more money you have to spend to get it. That may be money spent implementing security of your SP system, or it may be money spent on other security measures, such as firewalls and physical security measures.

  Deciding how much money to spend and where to spend it is a business decision that needs to be made in the context of your overall security environment. You have to look at the environment as a whole and develop your security strategy at that level.

Like many other business trade-offs, this one follows the 80/20 rule. In the spectrum from no security to fortress-like security, an 80 percent level of trust
can be achieved with 20 percent of the funds. Your business environment and your overall security strategy will determine where and how you fortify your system.

An SP system can be situated anywhere from an isolated network to a network directly connected to the Internet. In addition, the SP system may or may not be an integral part of the overall security strategy. Different environments and different strategies call for different security levels on the SP system. This redbook shows you how to tailor the security state of the SP system to meet these diverse needs.

1.2 SP security design

As we already mentioned, for all practical purposes, an SP system can be viewed as a cluster of RS/6000 workstations connected to one or more LANs, with the CWS serving as the central point of control for the entire system. It is a distributed system that follows the client/server model.

In an SP system, the Parallel System Support Program (PSSP) software provides the centralized command and control functions. It runs on all machines in an SP system and enables the CWS to act as the management console for all SP system activities. PSSP is implemented as a cooperating set of client and server (daemon) programs, which are collectively known as the SP Trusted Services.

1.2.1 The SP Trusted Services

Table 1 contains a partial listing of the SP Trusted Services.

<table>
<thead>
<tr>
<th>SP Trusted Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Data Repository (sdrd)</td>
<td>The central repository of SP-specific configuration information.</td>
</tr>
<tr>
<td>Hardware Monitor (hardmon)</td>
<td>The subsystem that monitors and manages the state of the frames, nodes, and switches in an SP system.</td>
</tr>
<tr>
<td>Secure System Command Execution Tool (sysctl)</td>
<td>An authenticated client/server system for running commands remotely and in parallel.</td>
</tr>
<tr>
<td>Topology Services (hats)</td>
<td>The subsystem that provides information about the state of the nodes and network adapters to other PSSP subsystems.</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

The SP Trusted Services comprise a set of distributed applications. As with all distributed applications, security is an issue. The clients and servers should have some way of verifying one another's identity. They should also have some way of controlling who has access to what for the resources they manage. In an SP system, these security mechanisms are provided by some combination of one or more of the following technologies: AIX, Kerberos V4, and DCE, which includes Kerberos V5.

1.2.2 The SP central point of management

The CWS provides the single point of control for monitoring and managing an SP system. The SP Trusted Services provide the foundation for this centralized control, but there are additional tasks that must be performed by the system administrator (root) in the normal day-to-day operation of an SP system. It is the remote command facility that provides the economy of scale to these tasks. A single command executed by the system administrator on the CWS can simultaneously perform tasks on any number of nodes in the SP system concurrently. This is a tremendously powerful facility, but it is also one that requires the same type of security mechanisms as the SP Trusted Services.

This redbook describes the security mechanisms for the SP Trusted Services and remote command execution in detail. It gives you information on how to

<table>
<thead>
<tr>
<th>SP Trusted Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Services (hags)</td>
<td>The subsystem that provides facilities for coordinating and monitoring changes to the state of a client program running on a set of nodes.</td>
</tr>
<tr>
<td>Event Management (haem)</td>
<td>The subsystem that generates events based on the state of system resources and resource monitor requests.</td>
</tr>
<tr>
<td>Problem Management (pmand)</td>
<td>The subsystem that provides the ability to execute actions based on the occurrence of specific events.</td>
</tr>
<tr>
<td>LoadLeveler</td>
<td>A workload management facility for interactive and batch processing.</td>
</tr>
<tr>
<td>General Parallel File System</td>
<td>A distributed filesystem that provides concurrent, shared access to files spanning multiple disk drives located on multiple nodes.</td>
</tr>
<tr>
<td>Parallel Environment</td>
<td>A facility for developing and executing parallel, message-passing applications.</td>
</tr>
</tbody>
</table>
make good decisions about which security mechanism(s) to use, and it shows you ways in which to implement those choices on your SP system.

1.3 How to use this book

This redbook is primarily a cookbook on how to implement various types of security on your SP system. You should have some knowledge of and experience with AIX and PSSP before reading this book. If you plan to take advantage of the PSSP integration with DCE, you should be familiar with DCE. We have provided a number of references to introductory redbooks on the SP and DCE in the bibliography at the end of this redbook.

Before you implement a specific security technology, such as Restricted Root Access, you first need to decide how much security you need and can afford. That decision will guide you in choosing the right security technology for your environment.

1.3.1 How much security do you need?

Depending on the kind of business you are in and the environment in which your system operates, you will have to decide on the level of security you are going to implement. As described earlier, it mostly depends on the amount of trust you have in your environment and the amount of money you can spend to enhance your security to reach that target level of trust.

Suppose your system is connected to your intranet, and you do not trust that network. You will have to shield your system from that network using firewalls or a comparable technology. This will raise the level of trust you have in the data transfer over that network. It does, however, mean that you will have to pay for a firewall and its maintenance.

Before adding any kind of security feature to your system, you should first plan for the amount of time it takes to implement and maintain it. Maintaining a high-security environment will require a large investment in time.

1.3.2 Choosing the right security technology for your environment

Communication between systems is always risky. You will be sending information through a network that you can, probably, never trust completely. Since most of the SP management is done through the network, your system is vulnerable to two common types of network fraud:

- Sniffing
- Spoofing
Sniffing means that somebody on the network has a network device that will read all packets on the network, regardless of whether they were meant for them. You can try to prevent this by designing your physical network using switches, bridges, and subnets so that it will be harder to gain access to your stream of packets. You can also encrypt all sensitive data on your network using software or hardware methods.

Spoofing means assuming the identity of another system on the network. To achieve this, the original system should be disabled. This can be done with a denial of service attack (for instance, flooding the system with packets). The false system can now profit from any trust you placed in the real system. You can prevent this by using an authentication scheme. This will guarantee that the system in which you are placing trust is really that system and not a deceiver.

Both sniffing and spoofing are also used for bona fide purposes. Sniffing is used to determine network problems, and spoofing is used in Network Address Translation (NAT).

In an SP environment, it is important to have the system and network security at the level consistent with the amount of trust that is placed in the environment. To achieve this, PSSP supports three security mechanisms:

- Standard AIX
- Kerberos V4
- DCE (using Kerberos V5)

Standard AIX uses the normal UNIX facilities. When a host sends you a request across the network, AIX only checks the originating IP address. Spoofing is possible by using the same IP address as the original system. Sniffing can, for example, be used to catch a password that was typed in during a telnet session.

Kerberos V4 uses a secret key mechanism based on system IDs to identify all systems within its realm. The whole authentication process is encrypted, making it a secure way to prove the identity of your systems. An explanation of how it works is included in Appendix A, “How Kerberos works” on page 485.

DCE is the most advanced method because it uses Kerberos V5. Kerberos V5 improvements over Kerberos V4 include:

- Use of an improved level of encryption
- More secure ticketing system (less vulnerable to sniffing)
• Use of the Generic Security Services API (GSSAPI), an industry-standard application programming interface (API)

Of course, DCE provides much more than just Kerberos V5. It also has a directory service, a time service, user management, and, through the separate DFS LPP, a distributed file service.

Because DCE is more like a framework, of which Kerberos V5 is an integral part, it is much more complex than a Kerberos V4 implementation. We do not recommend using DCE by PSSP unless you also wish to use additional features of DCE, such as user management or DFS.

We will have more to say about DCE in this redbook. More information on Kerberos V5 can be found at the MIT Web site:

1.3.3 Proceeding from here

So, having analyzed your environment and determined your security needs, where do you go from here? The remainder of the redbook is organized as follows:

• Read Chapter 2, “SP security concepts and terminology” on page 11. It contains concepts and terminology used throughout the remainder of the redbook and is essential to your understanding of the remaining chapters.

• If you are thinking about using DCE, read Chapter 3, “What you need to know if you plan to use DCE” on page 23. It highlights the areas of DCE with which you must be familiar when implementing it in an SP environment.

• If you are installing a new SP system, read Chapter 4, “Implementing a specific security state” on page 57. It gives you step-by-step instructions on how to implement your chosen security state.

• If you wish to change the security mechanism used on your SP, read Chapter 5, “Changing from one security state to another” on page 131. It gives you step-by-step instructions on how to change security states. You might also read this chapter as a follow-up to Chapter 4, “Implementing a specific security state” on page 57. For example, suppose you just installed your SP system with minimal security to shorten your installation time. Now that your system is up and running, you want to implement stronger security prior to connecting it to an external network. This chapter shows you how to do that.

• If you want to restrict root access on your SP nodes, read Chapter 6, “Restricted Root Access (RRA)” on page 183. It explains the ramifications...
of restricting root access and shows you how to enable this feature of PSSP 3.2.

- Now that your SP system is at a specific security state, what else can you do to fine tune your security environment? Read Chapter 7, “Example security scenarios” on page 199. It will help you further analyze your environment and show you additional tools and techniques you can use to enhance your existing SP security state.

- Implementing security is not a simple task. If something goes wrong along the way, Chapter 8, “Problem isolation and resolution” on page 237, gives you step-by-step problem determination and troubleshooting procedures for the most common situations.

- If you want to know the internal workings of Kerberos, turn to Appendix A, “How Kerberos works” on page 485. It contains a great description of Kerberos from the redbook The RS/6000 SP Inside Out, SG24-5374.

- An overview of most of the commands and tools used throughout this redbook can be found in Appendix B, “Security command and tool reference” on page 493.
Chapter 2. SP security concepts and terminology

To understand the rest of the redbook, you must become familiar with the SP security concepts and terms defined in this chapter.

The RS/6000 SP is an enormously flexible and versatile computing platform. SPs are used in many environments for many different purposes, from playing chess, to modeling weather phenomena, to housing corporate databases, to hosting Web sites - just to name a few. This, coupled with other considerations, such as company policy, business needs, cost and effort versus risk, and so on, define unique security requirements, ranging from as little as possible to fortress-like and everything in between. To accommodate these varied security requirements, IBM has provided the capability that enables you to tailor the security configuration of your SP system to your specific needs. In this redbook, the term *security state* is used to describe the configuration of your SP with respect to security.

The purpose of the present chapter is to give you conceptual information about the various SP security attributes and explain how they can be combined into a specific SP security state. We begin by covering some general SP and security concepts that are prerequisite to the discussion of the security attributes. We next move into a discussion about the SP security attributes and show you how they can be combined into specific well-defined security states. The following chapters contain step-by-step instructions for installing and configuring your SP system to a specific security state and migrating from one security state to another.

2.1 Authentication and authorization

Before going any further, we need to discuss the terms *authentication* and *authorization*. Most computer security mechanisms are based on a two-step process. The first stage is authentication, which ensures that a user is who he or she claims to be. The second stage is authorization, which allows the user access to various resources based on the user's identity.

In other words, authentication is the process of validating the identity of an entity (usually based on a user name and password), but says nothing about the access rights of the entity. Authorization is the process of granting or denying individuals access to system objects based on their identity.

Think of it this way, if I call you up on the telephone, I first need to identify myself to you in such a way that you are sure I am who I claim to be (authentication); then, depending on my relationship to you, you decide
whether or not to give me the information for which I am asking (authorization). If I happen to be a member of your immediate family, I will probably have higher authorization than a casual acquaintance for family-related matters.

You need to consider both authentication and authorization in the security planning of your SP system. When talking about authentication within an SP environment, there are two distinctions to be made:

- Authentication used by AIX remote commands (telnet, rsh, ftp, and so on)
- Authentication used by SP Trusted Services (hardmon, sysctl, sdr, and so on)

The security settings may differ for these two groups. They are managed by setting attributes in the SDR, and should only be changed by running the appropriate PSSP commands. Authorization only applies to AIX remote commands.

The following security mechanisms are supported by PSSP:

- **Standard AIX** - The login authentication method is done through the DES-encrypted password stored in the /etc/security/passwd file and the UNIX login program. Remote command authentication is done by reading the IP address and user ID in the network request packet from the originating host.
  
  The authorization method is implemented through the normal UNIX base file permissions and ownership, through the AIX extended permissions, and, optionally, through mechanisms, such as .rhosts or hosts.equiv files.
  
  Additional security features and tools for AIX are covered in detail in Chapter 7, “Example security scenarios” on page 199.

- **Kerberos V4** - The authentication method is implemented through the kerberos daemon on the master security server (typically, the control workstation (CWS)), and the kinit program on the client machines, which include the nodes and the CWS.

  The authorization method is implemented through the kerberos daemon and mechanisms, such as .klogin and /etc/krb-srvtab files.

  For a detailed description of how Kerberos works, see Appendix A, “How Kerberos works” on page 485.

- **DCE (Kerberos V5)** - This represents the authentication and authorization services provided by DCE or, more specifically, Kerberos V5. The authentication method is implemented by the secd daemon on the DCE Security server(s) and the dced daemon on the DCE clients.
The authorization method is implemented through DCE Access Control Lists (ACLs) and mechanisms, such as the .k5login file and DCE keytab files in the /spdata/sys1/keyfiles/ssp/<hostname>/directory.

Chapter 3, “What you need to know if you plan to use DCE” on page 23, contains additional information about implementing DCE for security in an SP system. Section 3.12, “Suggested reading for DCE” on page 52, gives you pointers to additional sources of information about DCE.

As you can see, there are three different security mechanisms available for use, each having its own authentication and authorization methods. To simplify the process of categorizing and specifying the different combinations of each, PSSP has defined these authentication and authorization settings in a finite set of security attributes. It is these security attributes taken together that define the security state of your SP system.

### 2.2 Restricted Root Access (RRA)

One of the key aspects of SP system management is keeping the effort to manage large numbers of nodes low. To achieve this, Kerberos was implemented and, because of the strong authentication it provided, it was possible to securely authorize root to perform actions on all nodes from any node. This simplifies SP management enormously since remote command and copy instructions can be run to any node, and scripts that can access anything anywhere within the SP can be written. This makes the SP function more like a single system.

From a security standpoint, this means that once root access on a single node has been compromised, the entire SP is compromised. It is also not possible to grant root access to an application manager on a single node because he or she will immediately have access to the entire SP. This is desirable within a scientific/technical environment where the SP functions as a parallel system because all nodes are running the same application anyway. Most commercial sites will not have a problem with this because all nodes are typically managed by the same people.

However, there are sites that will have a problem with automatic root access:

- High security commercial, mostly financial environments
- Internet sites that need to contain security breaches
- Commercial sites that have different customers within their SP
- Server consolidation sites that need to be able to, temporarily, give root access to non-administrators on specific nodes
To prevent this automatic root access, it would suffice to remove all entries in the appropriate authorization file (/.rhosts, /.klogin, or /.k5login), but this would also disable a lot of PSSP functionality. The Restricted Root Access feature, which is available as of PSSP 3.2, will set up the SP system so only the CWS has remote access to the nodes, and the nodes do not have root access to each other. This means that each node has its own root user, which has no authority over other nodes.

The Restricted Root Access feature is turned off by default and can be turned on using SMIT, SP TaskGuides, or the command line. If activated, it will change all authorization files within the SP, and configure sysctl so that it can be used to take over the tasks that normally required rsh or rcp access.

There are some considerations to keep in mind when using the SP in Restricted Root Access mode:

- Since nodes do not have remote command ability on other nodes, boot/install servers other than the CWS are not supported.
- Since the Restricted Root Access feature required changes to PSSP, all nodes within the SP need to be at least at PSSP 3.2.
- Manual action is required to update Kerberos on the CWS when HACMP/ES is used in high security mode.
- GPFS and VSD are not supported in Restricted Root Access mode.

Restricted Root Access is a new feature with PSSP 3.2 and is described in Chapter 6, “Restricted Root Access (RRA)” on page 183. This chapter describes the Restricted Root Access feature, how it works, restrictions imposed on the SP when using it, and how to activate this new feature.

### 2.3 SP security attributes

To completely and unambiguously describe the authentication and authorization settings of an SP system, PSSP has added the following four security attributes to the Syspar class in the SP System Data Repository (SDR):

- auth_install
- auth_root_rcmd
- ts_auth_methods
- auth_methods
Each attribute can be set to one or more of the authentication/authorization methods available for it. Each method is called a *base security option*, and when more than one method is used, we call it a *combination security option*. When used in combination, each method has a *failover* behavior. The method listed first is tried first; if it fails, the method listed second is tried second, and so on.

The **auth_install** attribute defines which authentication capabilities to install and/or configure on the nodes. This attribute only takes effect when the nodes are installed or customized. Since the Kerberos V4 and AIX filesets are installed by default on the nodes, this attribute only controls whether or not they get configured.

The **auth_methods** attribute defines which authentication methods will be used for AIX remote commands, such as telnet, ftp, rlogin, rcp, rsh, etc. At boot time, the /etc/rc.sp script, which is launched from the inittab, sets the authentication methods using the AIX `chauthent` command.

The **auth_root_rcmd** attribute defines which authorization methods are used for root access to the AIX remote commands. At boot time, the /etc/rc.sp script, which is launched from the inittab, runs the PSSP `upauthfiles` command to create or update the appropriate authorization file(s).

The **ts_auth_methods** attribute defines which authentication methods will be used for the SP Trusted Services, such as hardmon, sysctl, sdr, etc. The setting takes effect immediately.

Table 2 lists the four security attributes and their respective base security options. It explains the purpose of each attribute and the effect of setting each base security option on it.

**Table 2. Descriptions of SP security attributes and their settings**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_install</td>
<td>Defines which authentication capabilities to install and/or configure on the nodes.</td>
</tr>
<tr>
<td>Base options:</td>
<td></td>
</tr>
<tr>
<td>dce</td>
<td>Install and configure DCE client filesets on the nodes.</td>
</tr>
<tr>
<td>k4</td>
<td>Configure Kerberos V4 on the nodes. The filesets are installed by default.</td>
</tr>
<tr>
<td>std</td>
<td>No installation or configuration is required for std.</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>Defines which authorization methods are used for root access to the AIX remote commands.</td>
</tr>
</tbody>
</table>
In some cases, the settings of one security attribute limit the valid settings of another security attribute. Care should be taken not to use conflicting settings. For instance, if the auth_methods attribute is set to k5 and the auth_root_rcmd attribute is set to std, the AIX authentication method performs only Kerberos V5 authentication. The .rhosts authorization file is created but never used because the standard AIX authentication method is not enabled. Even worse, the .k5login file that it requires is never created.

### 2.4 SP security states

The SP security state is defined by the settings of the four security attributes. Many combinations of security attribute settings are possible, only a small number of which result in valid security states. In this redbook, we cover scenarios dealing with three security states: *Minimal*, *Compatibility*, and *DCE*.

The Minimal security state is the easiest possible security implementation to manage. It is also the least secure. The AIX remote commands and the SP Trusted Services use only standard AIX authentication and authorization.
The Compatibility security state is equivalent to how PSSP operates in versions prior to PSSP 3.2. The authentication mechanism is Kerberos V4 combined with AIX. Authorization is done through Kerberos V4 only. The SP Trusted Services use the compat option, which means that some services use AIX authentication and others use Kerberos V4 authentication.

The DCE security state is the most secure and the most complex to manage. The authentication mechanism is Kerberos V5 combined with AIX. Authorization is done through Kerberos V5 only. The SP Trusted Services use the dce option, which means that all services use DCE (Kerberos V5) authentication.

Table 3 displays the attribute settings for each of these three security states.

<table>
<thead>
<tr>
<th>Security Attribute</th>
<th>Security State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minimal</td>
</tr>
<tr>
<td>auth_install</td>
<td>std</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>std</td>
</tr>
<tr>
<td>ts_auth_methods</td>
<td>&quot;&quot; (none)</td>
</tr>
<tr>
<td>auth_methods</td>
<td>std</td>
</tr>
</tbody>
</table>

Notice that some of the security attributes are set to base security options, while others are set to combination security options. It is also important to know that, when adding additional security options, you must work from the top down, that is, from auth_install to auth_methods. When removing security options, you must work from the bottom up, that is, from auth_methods to auth_install. This is covered in detail in Chapter 5, “Changing from one security state to another” on page 131.

2.5 Each SP partition has its own security state

An RS/6000 SP system is comprised of a CWS, processor nodes, frames, and, optionally, switches. By default, all nodes in an SP system are included in a single system partition, but you can change this. Partitioning enables you to divide an SP system into separate logical SPs. For example, you may want to have one partition for production, another for test, and a third for development.

With partitioning, the nodes in one partition are no longer able to communicate across the switch to nodes in other partitions. In other words,
partitioning an SP system partitions the switch. However, the CWS remains a member of all partitions.

One of the benefits of partitioning is that you can set different policies in each partition. This now includes security states. For example, you could set the production partition to DCE, the test partition to Compatibility, and the development partition to Minimal. This is shown in Figure 1.

![Figure 1. SP system partitioned with different security states](image)

By definition, the CWS is a member of each and every partition on your SP system. Therefore, the CWS must be configured with each and every security state used. For Figure 1, that means that the CWS would be configured with the combination of the DCE, Compatibility, and Minimal security states.

**Important**

Because the CWS is a member of all partitions, the entire SP system is only as secure as the least secure partition in it.

For this redbook, we assume an SP system configured as a single system partition. However, when we give an example of a command that takes a
partition name as an option, we use the default partition name (spcws, for our SP system) for the option in the example.
Chapter 3. What you need to know if you plan to use DCE

Important
The use of DCE in the SP environment is optional. If you do not plan to implement DCE as a security state on your SP system, skip this chapter.

In the previous chapter, you were introduced to the concept of SP security states. The primary focus of this chapter is to provide you with information on the way in which DCE is implemented as a security state within the SP. The information presented here builds the foundation for what follows in subsequent chapters.

DCE, the Distributed Computing Environment from the Open Software Foundation (OSF), is comprised of a set of integrated services designed to support distributed applications, such as PSSP. You can think of DCE as middleware or enabling technology because it is not intended to exist alone, but rather to provide a platform on which to run distributed applications.

Note
The purpose of this chapter is not to teach you DCE. There are already many fine books whose purpose is to do just that, and we have listed some of them for you in Section 3.12, “Suggested reading for DCE” on page 52.

3.1 The DCE Secure Core

The primary components of DCE are:
- DCE Threads
- Remote Procedure Call (RPC)
- Distributed Time Service (DTS)
- Cell Directory Service (CDS)
- Security Service (Security)
- Global Directory Service (GDS)
- Distributed Management Environment

Although often described as a DCE primary component, the Distributed File System (DFS) is really a DCE application. It is the most widely-used DCE
application and, for that reason, is, generally, grouped with the list of DCE primary components. These components are illustrated in Figure 2.

![DCE Architecture Diagram](image)

**Figure 2. DCE architecture**

The Threads, RPC, CDS, Security, and DTS components are commonly referred to as the Secure Core. Of these, CDS and Security are the components that you will work with in configuring DCE security on the SP.

### 3.2 Synchronized time

All DCE components rely on highly-synchronized time. In particular, Kerberos is extremely time-sensitive. Therefore, it is critical to have the clocks on all DCE machines synchronized.

DTS is the time synchronization facility provided with DCE, but its use is not mandatory. All that matters is that time is synchronized. The time synchronization facility provided with the SP is the Network Time Protocol (NTP). It provides much the same service as DTS.

When using NTP with DCE, it is always a good idea to check a machine’s notion of time before starting DCE. For example, when the motherboard is replaced in a machine, it is not uncommon for that machine to boot up with a
notion of time that is very far back in the past. This will cause problems if you attempt to start DCE before adjusting the clock to the current time. A simple check is to run `date` on the local machine and compare it with the output of a `setclock` from a normal user login against your time master. For example, if `jueneman` is your normal user login and the control workstation (CWS), `spcws`, is your time master:

```
spn01/ # date ; su - jueneman -c setclock spcws
Fri Oct 15 07:27:57 MDT 1999
Fri Oct 15 07:27:57 1999
spn01/ #
```

If the two values do not match, set the clock on the local machine to match the time master before starting DCE. Internally, DCE uses Universal Coordinated Time (UTC), and it is the UTC time that must match between clients and servers. This becomes important if your cell is spread across multiple timezones.

### 3.3 The concept of a DCE cell

Another key concept in DCE is that of the **cell**. A cell is a collection of machines that are managed together as a DCE unit. The cell is the administrative domain for DCE. DCE cells can be comprised of anywhere from one to many hundreds of machines, but an SP system can only be part of one cell. At a minimum, a cell must contain a DCE Security Master server, a Primary CDS server, and a Time server (DTS, NTP, or some other). These servers can all run on a single machine, or they can be distributed across multiple machines in the cell.

In addition, you can create additional DCE **Security Replica** servers and **Secondary CDS** servers in the cell. They can be used to provide the cell with higher availability and enhanced performance.

The Security and CDS components of DCE store information in their own databases. The Security database is known as the **registry**, and the CDS database is known as a **clearinghouse**. DCE Security Master servers and Primary CDS servers hold the read-write copies of the databases, while DCE Security Replica servers and Secondary CDS servers hold read-only copies. In a typical DCE cell with replication, the DCE Security Master server and the Primary CDS server are on one machine, with two additional machines each holding a DCE Security Replica server and Secondary CDS server pair.
3.4 DCE administrative principals

There are three DCE principals that you need to be familiar with:

- The cell-administrator principal
- The self-host principal
- The optional SP-administrator principal(s)

The DCE cell-administrator principal is the principal used to do the initial cell configuration. In a rough sense, the cell-administrator principal is to DCE what root is to AIX. The typical name for the cell-administrator principal is cell_admin, but you are free to use another name. If you plan to use DCE with PSSP, you will need to have valid DCE cell-administrator credentials to perform some of the routine tasks associated with installing an SP system and/or customizing SP nodes.

The next DCE principal that you need to be familiar with is the self-host principal. Each machine in a DCE cell has an associated self-host principal (/./hosts/<hostname>/self). The self-host, or machine, principal is the principal that is used by the DCE client daemons to authenticate to DCE. The self-host principal is also the principal that a root user, who is unauthenticated to DCE, inherits by default. If you plan to use DCE with PSSP, you will need to know which tasks require self-host credentials and which tasks require cell-administrator credentials.

In a DCE-only configuration, many of the routine installation, configuration, and monitoring tasks associated with managing an SP system require valid DCE credentials. You have a lot of flexibility here. You have control over which DCE principals have access to the different components of the SP system, such as writing to the SDR or controlling hardmon. For example, you could create several new SP administrator principals and assign different rights to each, or you could assign all rights to either the cell-administrator principal or the self-host principal for the CWS. Which principals to create and what access rights to give to each really depends on your specific environment. Here are some questions to consider:

---

**Important**

All DCE servers, especially Security servers, should be configured to not allow end-user access, and they should be secured with stringent physical access controls. If the security registry is compromised, your entire DCE cell is compromised.
• Do you have the need to divide up the SP administrative tasks?
• Do you own the DCE cell-administrator principal, or is that controlled by another group?
• Do you want to use DCE with PSSP but allow the root user to perform the administrative tasks without having to explicitly authenticate to DCE?

In this redbook, we show examples with all three types of administrative principals. Giving all access rights to either the cell-administrator principal or the self-host principal on the CWS simplifies the installation and initial configuration of your SP system. Then, once your SP system is set up with your chosen security state, you can create the SP administrator principal(s) with the appropriate access rights.

You can use the klist command to figure out who you are at any given time as shown in the following screen.

```
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login: root
root's Password: ...
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # dce_login cell_admin
   Enter Password:
   DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/cell_admin
sp3en0/ # kdestroy ; exit
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/hosts/sp3en0/self
```

3.5 DCE groups for PSSP

The PSSP software creates a number of groups in DCE. These groups are used by the SP Trusted Services to grant privileges. Determining which principals to make members of the PSSP DCE groups all depends on how you wish to manage your SP system.

During installation and configuration of PSSP, you have a choice of using the following principals when configuring PSSP components:

• The cell_admin principal
• The self-host principal
• Some other principal that you have already created, such as sp_admin
The DCE administrative principal (cell_admin) has administrative privileges for the entire DCE cell. The cell_admin principal is a very powerful principal. The cell_admin principal controls membership to all DCE groups, and, if DCE user management is implemented, the rights of the cell_admin principal extend into the AIX domain as well. The cell_admin principal is to DCE what the root user is to UNIX. When the DCE cell extends beyond the SP system, it is not advisable to use the cell_admin principal for SP tasks.

There is a unique self-host principal (/./hosts/hostname/self) for each machine in a DCE cell. The root user automatically inherits self-host principal credentials on the CWS and the nodes when the SP system is configured in a DCE cell. Adding the self-host principal to the SP groups is very convenient for the root user because a separate DCE login is not required for doing SP tasks. However, it does circumvent an added layer of security.

Creating a new principal for SP management tasks is, probably, the most secure option. It splits the responsibilities into clearly-defined areas. It is also the most complex solution. To use this option, you will need a thorough understanding of DCE and the way it is used by the PSSP software.

By adding a principal to one of the groups listed in Table 4, you authorize that principal to do certain tasks and use certain commands.

<table>
<thead>
<tr>
<th>User access group</th>
<th>Authorizes a member to do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>haem-users</td>
<td>Authorizes use of Event Management.</td>
</tr>
<tr>
<td>hm-admin</td>
<td>Authorizes administrative tasks used to manage the SP Hardware Monitor.</td>
</tr>
<tr>
<td>hm-control</td>
<td>Authorizes all SP Hardware Monitor tasks except administration.</td>
</tr>
<tr>
<td>hm-monitor</td>
<td>Authorizes monitoring of SP hardware. This is read-only access to the SP Hardware Monitor.</td>
</tr>
<tr>
<td>sdr-admin</td>
<td>Authorizes SDR tasks on partitioned classes.</td>
</tr>
<tr>
<td>sdr-write</td>
<td>Authorizes SDR updates to existing partitioned classes, but not the addition or deletion of classes or other administrative tasks.</td>
</tr>
<tr>
<td>sdr-system-class-admin</td>
<td>Authorizes any SDR tasks on global system classes.</td>
</tr>
<tr>
<td>sdr-system-class-write</td>
<td>Authorizes SDR updates to existing system classes, but not the addition or deletion of system classes or other administrative tasks.</td>
</tr>
</tbody>
</table>
### 3.6 DCE keyfiles for PSSP

Servers, which can also be called applications or services, that engage in communications across the network can run under their own network identity or the network identity of the principal who started them. To run under their network identity, they must be granted "principal" status.

For more information about the DCE groups created for PSSP, see the **PSSP Administration Guide**, SA22-7348.

<table>
<thead>
<tr>
<th>User access group</th>
<th>Authorizes a member to do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>spsec-admin</td>
<td>Authorizes DCE ACL control authority for PSSP object ACL management tasks.</td>
</tr>
<tr>
<td>switchtbd-clean</td>
<td>Authorizes cleanup (the unloading) of switch tables.</td>
</tr>
<tr>
<td>switchtbd-load</td>
<td>Authorizes loading of switch tables.</td>
</tr>
<tr>
<td>switchtbd-status</td>
<td>Authorizes querying the status of loaded switch tables.</td>
</tr>
<tr>
<td>sysctl-cwsroot</td>
<td>Authorizes use of certain switch management commands by non-root users of SP Perspectives. Sysctl creates a group entry for it in the /etc/sysctl.rootcmds.acl DCE ACL file.</td>
</tr>
<tr>
<td>sysctl-default</td>
<td>Sysctl creates a group entry for it in any ACL added by customization (not supported by IBM).</td>
</tr>
<tr>
<td>sysctl-logmgt</td>
<td>Authorizes use of log management commands by non-root users. Sysctl creates a group entry for it in the /etc/logmgt.acl DCE ACL file.</td>
</tr>
<tr>
<td>sysctl-master</td>
<td>Authorizes full access to all Sysctl facilities including Sysctl administration. Sysctl creates a group entry for it in the /etc/sysctl.acl DCE ACL file.</td>
</tr>
<tr>
<td>sysctl-mmcmd</td>
<td>Authorizes access to GPFS commands. Sysctl creates a group entry for it in the /etc/sysctl.mmcmd.acl DCE ACL file.</td>
</tr>
<tr>
<td>sysctl-pman</td>
<td>Authorizes access to Problem Management commands. Sysctl creates a group entry for it in the /etc/sysctl.pman.acl DCE ACL file.</td>
</tr>
<tr>
<td>sysctl-vsd</td>
<td>Authorizes access to Problem Management commands. Sysctl creates a group entry for it in the /etc/sysctl.vsd.acl DCE ACL file.</td>
</tr>
</tbody>
</table>

PSSP also creates a number of **service access** groups. These are used by service principals, such as hardmon. The names of these groups end in `-services`.

For more information about the DCE groups created for PSSP, see the **PSSP Administration Guide**, SA22-7348.
own identity, servers must be programmed to perform a login and authenticate that identity.

During login, all principals (human, server, and machine) must pass their password to the DCE Authentication Service, which uses these passwords to generate authentication keys. The most common method for human users is to simply enter their password. A different method must be provided for server principals. This is accomplished through the use of a keyfile.

A keyfile contains principal keys, which are the basis of DCE security, and encrypted password data. A keyfile exists as an encrypted file in the AIX file system and contains principal names, types, versions, and values. Specific DCE commands and APIs are used to read from and write to a keyfile. During login, a server can access a keyfile to obtain its key (password), pass its key to the authentication service, log in, and be authenticated.

Part of a DCE client’s local configuration is the creation of a DCE keytab object for the DCE self-host, or machine, principal and an associated keyfile for the self-host principal. As part of PSSP’s configuration for DCE security, keytab objects and associated keyfiles are created for the set of PSSP services that comprise its Trusted Services, such as Sysctl, Hardmon, and SDR.

A list of all keytab objects for a particular host can be displayed by issuing the command, shown in the following screen, as the root user:

Control workstation example:

```
sp3en0/ # dcecp -c keytab cat
/.../sp_cell/hosts/sp3en0/config/keytab/self
/.../sp_cell/hosts/sp3en0/config/keytab/LoadL/sp3en0/GSmonitor
/.../sp_cell/hosts/sp3en0/config/keytab/LoadL/sp3en0/Kbd
...
/.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/switchtbld
/.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
```

Node example:

```
sp3n05/ # dcecp -c keytab cat
/.../sp_cell/hosts/sp3n05/config/keytab/self
/.../sp_cell/hosts/sp3n05/config/keytab/LoadL/sp3n05/GSmonitor
...
/.../sp_cell/hosts/sp3n05/config/keytab/ssp/sp3n05/switchtbld
/.../sp_cell/hosts/sp3n05/config/keytab/ssp/sp3n05/sysctl
```
The output shown includes PSSP keytab objects in addition to the self-host principal’s keytab object.

The location of the physical keyfile associated with a keytab object can be determined by showing the contents of the keyfile. This does not actually display the encrypted password data of a keyfile. The physical location of a keyfile is defined to DCE for read and update purposes. The location is stored in the keyfile and is part of the storage field when the contents of a keyfile are displayed.

As shown in the next screen, the physical location of the self-host principal’s keyfile is /krb5/v5srvtab, while that of the PSSP principal sysctl is /spdata/sys1/keyfiles/ssp/sp3en0/sysctl.

```
sp3en0/ # dcecp -c keytab show /.../sp_cell/hosts/sp3en0/config/keytab/self
{uuid 00096565-4301-1d77-9108-0000c09ce054}
{annotation {Host Principal Keytab}}
{storage /krb5/v5srvtab}
{.../sp_cell/hosts/sp3en0/self des 1}
{.../sp_cell/host/sp3en0 des 1}
{.../sp_cell/host/sp3en0 des 2}
{.../sp_cell/ftp/sp3en0 des 1}
{.../sp_cell/ftp/sp3en0 des 2}
{.../sp_cell/hosts/sp3en0/self des 2}
{.../sp_cell/hosts/sp3en0/cds-server des 1}
{.../sp_cell/hosts/sp3en0/cds-server des 2}
...
```

While the above use of keytab show is an excellent general way to determine the physical locations of keyfiles, the following shows the default locations of DCE self-host principal and PSSP DCE principal keyfiles.

dced (the self-host principal) keyfile is /krb5/

Under normal conditions, this directory will exist and will contain the following keyfile:

```
sp3en0/ # ls -l /krb5
 total 16
-rw-r--r-- 1 root system 23 Nov 11 09:14 krb.conf
-rw------- 1 root system 436 Nov 11 09:18 v5srvtab
```

Chapter 3. What you need to know if you plan to use DCE  31
PSSP Trusted Services keyfiles are under \texttt{/spdata/sys1/keyfiles/} (main path)

Under normal conditions, this directory will exist and will contain the following directories:

```
sp3en0/spdata/sys1/keyfiles/ # ls -l
total 40
-dw-r-xr-x 3 root system 512 Oct 26 13:04 LoadL/
dw-r-xr-x 3 root system 512 Oct 26 13:04 mmfs/
dw-r-xr-x 3 root system 512 Oct 26 13:04 ppe/
dw-r-xr-x 4 root system 512 Oct 26 13:04 rsct/
dw-r-xr-x 3 root system 512 Oct 26 13:04 ssp/
```

There will be another directory under each of these directories. The name of this directory is the DCE hostname of the control workstation or nodes:

```
sp3en0/spdata/sys1/keyfiles/ssp/ # ls -l
total 32
-dw-r-xr-x 2 root system 512 Nov 17 12:29 sp3en0
```

The DCE keyfile(s) for a given service or services will be under the DCE hostname directory:

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # ls -l
total 64
-rw------- 1 root system 298 Oct 27 23:43 css
-rw------- 1 root system 382 Oct 27 23:43 pmand
-rw------- 1 root system 407 Oct 27 23:44 sp_configd
-rw------- 1 root system 397 Oct 27 23:44 spbgroot
-rw------- 1 root system 382 Oct 27 23:43 spmgr
-rw------- 1 root system 407 Oct 27 23:43 switchtbld
-rw------- 1 root system 152 Oct 26 12:22 sysctl
```

The keyfiles on the control workstation and on the nodes differ in two ways:

1. On the control workstation, the \texttt{ssp/} directory will contain at least two subdirectories. In addition to the DCE hostname directory, a second directory will appear, and it will be named after the default partition of the system. If more than two subdirectories exist, their names will correspond to the names of the remaining PSSP partitions in the system. The \texttt{ssp/{partition name}} subdirectories contain a DCE keyfile for each sdrd. The following example relates to a system running with three partitions, named \texttt{secsys1}, \texttt{secsys2}, and \texttt{secsys3}:
2. There are keyfiles that will only appear on the control workstation. These keyfiles are for sdr, hardmon, and SNMP Manager as shown in the following screen:

Even if the SP system is not configured for LoadLeveler, POE, GPFS, or an SP switch, PSSP DCE keyfiles are created for these services, including a css keyfile for use with PSSP switch services/commands.

This does not pose or create a security exposure, as the keyfiles can only be read and used by a root id.

It is important to note that keyfiles are stored on the same machine as the services whose keys they contain; so, if these keys are compromised, security can also be compromised. Therefore, it is critical to maintain the security of both the keytab objects (with DCE ACLs) and the keyfiles themselves (with AIX file permissions).

Creating separate individual keyfiles for each service principal that runs on a local node prevents services from accessing each other's keys and, thus, impersonating each other. Furthermore, keyfiles should exist such that they are readable only by root. If you do this, the servers must be started by root in order to read their keytab files and obtain their key during login. During login,
the service can access this file to obtain its key, pass its key to the authentication service, log in, and be authenticated.

In an SP system configured for DCE security, the control workstation and each node have their own set of PSSP DCE keyfiles for the Trusted Services on that host. Each PSSP DCE keyfile contains data unique to the service on a given host; so, it is not possible to use PSSP DCE keyfiles that are copied from one host to another, except in the case of RSCT services. (RSCT services use a DCE keyfile to encrypt/decrypt messages, not for login purposes.)

---

**Note**

Hardmon's keyfile exists only on the control workstation, since this is the only host within an SP where Hardmon runs. Likewise, the SNMP Manager keyfile exists only on the control workstation.

Access to DCE keytab objects is maintained through the local dced daemon, which protects all keytab objects on the host via ACLs. By default, only the self-host principals have read and write access to the keytab objects, and only a well-defined set of administrative DCE principals have read access to the objects. It is recommended that the default values be used in production environments.

To view the ACL permissions for keytab objects on a host, use the DCE `acl show` command as root in conjunction with a keytab object:

```bash
sp3en0/ # dcecp -c keytab cat | grep self
/.../sp_cell/hosts/sp3en0/config/keytab/self
sp3en0/ # dcecp -c acl show /.../sp_cell/hosts/sp3en0/config/keytab/self
{unauthenticated ------}
{user hosts/sp3en0/self acdepr}
{group subsys/dce/dced-admin a-depr}
{any_other ------}
...
sp3en0/ # dcecp -c keytab cat | grep sysctl
/.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
sp3en0/ # dcecp -c acl show /.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
{unauthenticated ------}
{user hosts/sp3en0/self acdepr}
{group subsys/dce/dced-admin a-depr}
{any_other ------}
```

Given that keyfiles contain password data, there must be a way to manage the contents of a keyfile. Specifically, there must be a way to change a password in a DCE keyfile.
In actuality, DCE does not provide a way to change a password stored in a keyfile. Passwords or, as DCE refers to them, keys, can only be added to or deleted from a keyfile. This is how a keyfile can contain multiple keys (passwords). Each key within a keyfile contains an associated principal name, type, version, and value. In doing so, a single keyfile can contain various keys with various principal names, which a server or service may use during a login. Typically, a keyfile contains only the principal name for the server for which it was created. However, there may be many key versions for that one principal name.

This is an important concept to note because, while a keyfile can contain many keys, the DCE Registry for a principal's account stores only one key. This means that when a server uses its keyfile to log in, there must exist, in its keyfile, a key version that matches the key stored in its Registry account. If not, the server cannot log in. This is no different than a human typing an incorrect password at login. However, unlike a human that can correct a typo, the server can only use the keys that are stored in its keyfile. This makes managing a keyfile critical to the normal operation of servers.

Depending on how the server or service application is written, managing a DCE keyfile is either done by the server or a key management service or manually maintained.

For example, the DCE self-host principal's keyfile is managed by the local dced (daemon). The dced spawns a thread and manages its own keyfile based on default DCE values that cannot be changed.

On the other hand, PSSP DCE keyfiles are managed in one of two ways, depending on the PSSP service for the keyfile:

**Manual update of the keyfile and then the keyfile is pushed out the needed hosts.**

This is true only for HATS (RSCT) services that use DCE keyfiles for message encryption purposes.

To add a new key version to the HATS keyfile and then distribute the keyfile to all nodes in the partition, perform the following steps:

1. Add a new key version to the HATS keyfile as the DCE self-host principal (as root) on the control workstation.
   The new key version is determined by taking the highest key version number in the current HATS keyfile and adding 1 to it.
Keep in mind that, in the keytab show example, the `sp3en0` in the
`/.../sp_cell/hosts/sp3en0/config/keytab/rsct/sp3en0/hats` string
represents the partition name.

2. Copy the updated keyfile to all nodes in the partition from the control
workstation as shown in the following screen:

Keep in mind that, in each of the rcp examples, the `sp3en0` in the
`/spdata/sys1/keyfiles/rsct/sp3en0/hats` strings represents the partition
name, while the `sp3n05` and `sp3n06` in the
`/spdata/sys1/keyfiles/rsct/{sp3n0*}/hats` represent the DCE hostname on
the target.

3. Refresh the RSCT subsystems from the control workstation as root:

This will refresh HATS, HAGS, and HAEM.
A PSSP Per Node Key Management daemon that runs on each host and manages all other PSSP DCE keyfiles that are not HATS (RSCT)-related.

To see the list of keyfiles managed by Per Node Key Management on the current host, issue the following from the command line as root: `spnkeymand -l`.

**Note**

An expiration value equal to 0 (zero) indicates that keys do not need to be updated. A value other than 0, represented in ISO time stamp format, indicates the expiration value returned by a query to the `spsec-services` organizational attributes.

**Control workstation example:**

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
service=LoadL/Schedd expiration=0
service=LoadL/Startd expiration=0
service=LoadL/Starter expiration=0
service=LoadL/Negotiator expiration=0
service=LoadL/Master expiration=0
service=LoadL/Mtd expiration=0
service=LoadL/GSmonitor expiration=0
service=mmfs/mmfd expiration=0
service=ract/ract expiration=0
service=ssp/switchtbld expiration=0
service=ssp/css expiration=0
service=ssp/pmand expiration=0
service=ssp/spmgr expiration=0
service=ssp/sp_configd expiration=0
service=ssp/spbgroot expiration=0
```

To see the list of keyfiles managed by Per Node Key Management on a remote host, issue an rsh to the host as root.

**Node example:**

```
```
The Per Node Key Management daemon runs on the control workstation and each node in the SP. Once every 24 hours, the daemon wakes up and checks to see if the keyfiles that it manages need to be updated. The 24-hour period is a fixed value and cannot be changed. The check is done by having the Per Node Key Management daemon log in to DCE using a service’s keyfile and then perform the actual check.

If, based on the results of the spsec-services organizational query, the key in the Registry account is slated to expire within the next 24 hours, Per Node Key Management updates the account first and the keyfile second with a new key version. This process ensures that the account password and the keyfile password are not out of sync with each other. (Given that the account can store only one password, or key, at a time, and given that a keyfile can contain many keys, if the account password gets updated with a new value but not the keyfile, the PSSP service cannot log in to DCE.) The key is a randomized value that is automatically generated by Per Node Key Management.

If the key is not ready to expire, no updates are performed, and Per Node Key Management goes back to sleep.

The 24-hour cycle begins at the time the daemon is initialized. For example, if the daemon is started at 4:05 p.m., it performs its keyfile checks, returns to sleep, and, then, 24 hours from the start time, the daemon will wake up and perform its check.

Per Node Key Management does not have its own log. Instead, it writes any errors detected during processing to the AIX error log. The AIX `errpt` command should be used to review the contents of the error log. (The `errpt -N spnkeyman` command is useful for quickly identifying just those entries that

```bash
sp3en0/ # rsh sp3n05 "/usr/lpp/ssp/bin/spnkeymand -l"
```

<table>
<thead>
<tr>
<th>service</th>
<th>expiration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppe/pmdv3</td>
<td>0</td>
</tr>
<tr>
<td>ppe/dpcl</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Schedd</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Startd</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Startd</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Negotiator</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Master</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/Idmd</td>
<td>0</td>
</tr>
<tr>
<td>LoadL/ISmonitor</td>
<td>0</td>
</tr>
<tr>
<td>mfsa/mfmd</td>
<td>0</td>
</tr>
<tr>
<td>ract/ract</td>
<td>0</td>
</tr>
<tr>
<td>ssp/switchtibld</td>
<td>0</td>
</tr>
<tr>
<td>ssp/css</td>
<td>0</td>
</tr>
<tr>
<td>ssp/gmdl</td>
<td>0</td>
</tr>
<tr>
<td>ssp/sp_configd</td>
<td>0</td>
</tr>
<tr>
<td>ssp/spbgroot</td>
<td>0</td>
</tr>
</tbody>
</table>
Chapter 3. What you need to know if you plan to use DCE

Normal processing (must update or no need to update) messages are not currently written to any log. This means that, to determine if a keyfile has been updated, the date/time stamp of a keyfile must be inspected, and the contents of the keyfile must be examined via a DCE keytab show command. (If the keyfile is being updated, there will be multiple key version numbers listed in the keyfile. Provided that an administrator is not pruning, or deleting, older key versions in a keyfile.)

The Per Node Key Management daemon is automatically started on the nodes during PSSP installation and configuration. It is manually started on the control workstation as part of the PSSP control workstation installation and configuration procedure.

To see whether or not Per Node Key Management is active on a host, issue the following `lssrc -s spnkeyman` command.

```
sp3en0/ # lssrc -s spnkeyman
Subsystem  Group    PID     Status
spnkeyman   14044    active
```

To see the state of Per Node Key Management on all hosts in the system, issue the following command: `dsh -avG "lssrc -s spnkeyman" | dshbak`.

```
sp3en0/ # dsh -avG "lssrc -s spnkeyman" | dshbak
HOST: sp3n05
----------
Subsystem  Group    PID     Status
spnkeyman   11108    active
HOST: sp3n06
----------
Subsystem  Group    PID     Status
spnkeyman   12142    active
```

If Per Node Key Management is not running on a host (the `lssrc` returns inoperative), the daemon can be started by issuing the following command as root: `startsrc -s spnkeyman`. (Per Node Key Management does not take any arguments or flags at startup.)
The remainder of this section shows an example of the DCE Registry (account) and keyfile changes when Per Node Key Management updates a keyfile, as well as an example of when Per Node Key Management determines there are no updates to make. For both examples, the results of an `errpt` query for `spnkeyman` reveal no matches since there were no errors detected during processing.

**Note**

Given that Per Node Key Management manages over fifteen keyfiles, the examples focus on two services, css and GSmonitor, to make the examples readable and easy to follow. Any of the queries done against the two services in the examples can be performed against any of the services managed by Per Node Key Management.

**Example when key data does not need to be updated.**

Establish the current date and time. This will be useful for translating the ISO date stamp in the examples.

```
sp3en0/ # date
Sun Dec 19 09:31:15 EST 1999
```

The `spsec-services` organization does not contain an explicit `pwdexpdate` value. This means that, for all accounts in the `spsec-services` organization, their password expiration is equivalent to `none`, that is, they do not have an expiration date.

```
sp3en0/ # dcecp -c org show spsec-services -all
{fullname {}}
{orgid 100}
{uuid 00000064-9844-21d3-9202-02608c2d4a7f}
{nopolicy}
```
A query of the expiration data reveals 0, indicating that the keys are not ready to expire.

Examine the AIX date/time stamps of some of the keyfiles managed by Per Node Key Management. This provides a "before" view of the keyfiles' file system attributes. Should Per Node Key Management update the keyfiles' key versions, the date/time stamps and the file size will change along with the list of key versions stored within the file.

Examine the DCE account attributes that relate to one of the keyfiles that Per Node Key Management will manage, and search for the "lastchange" attribute. This provides a "before" view of when account data was last changed. When Per Node Key Management (or any other authorized principal) updates the account's key (password), the lastchange value is
updated to show when the change was made and what principal made the change.

```
sp3en0/ # dcecp -c account show spn/sp3en0/css | grep last
(lastchange /.../sp_cell/spn/sp3en0/css 1999-11-11-10:33:15.000-05:00I-----)
```

Repeat the AIX file attributes and DCE account attribute "before" checks for the second service in the example, GSmonitor:

```
sp3en0/ # ls -l /spdata/sys1/keyfiles/LoadL/sp3en0/
```

```
sp3en0/ # dcecp -c account show LoadL/sp3en0/GSmonitor | grep last
(lastchange /.../sp_cell/LoadL/sp3en0/GSmonitor 1999-11-11-10:32:54.000-05:00I-----)
```

Stop Per Node Key Management and then start it to trigger the check:

```
sp3en0/ # stopsrc -s spnkeyman
0513-044 The spnkeyman Subsystem was requested to stop.
```

```
sp3en0/ # lssrc -s spnkeyman
Subsystem Group PID Status
spnkeyman inoperative
```

```
sp3en0/ # startsrc -s spnkeyman
0513-059 The spnkeyman Subsystem has been started. Subsystem PID is 14068.
```

```
errpt shows no error conditions for Per Node Key Management processing.
```

```
sp3en0/ # errpt -s 1219000199 -N spnkeyman
sp3en0/ #
```

Reexamining the AIX file attributes reveals no changes to the files as the result of Per Node Key Management processing. Likewise, no changes are made to the DCE account attributes.
Example when key data does need to be updated

Establish the current date and time. This will be needed when comparing the ISO date stamp value in the examples.

The `spsec-services` organization `pwdexpdate` value reveals that passwords for accounts in the `spsec-services` organization will expire on 1999-12-20 at 07:59:00, less than twenty-four hours from the current time.
A query via the `spnkeymand` command reveals an expiration value of 945694740:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=945694740
service=ppe/dpcl expiration=945694740
service=LoadL/Schedd expiration=945694740
service=LoadL/Startd expiration=945694740
service=LoadL/Startd expiration=945694740
service=LoadL/Negotiator expiration=945694740
service=LoadL/Master expiration=945694740
service=LoadL/Dbdd expiration=945694740
service=LoadL/OSMonitor expiration=945694740
service=ssfs/mmfsd expiration=945694740
service=rsc/rsc expiration=945694740
service=sp3en0/switchtlb expiration=945694740
service=sp3en0/css expiration=945694740
service=sp3en0/pmand expiration=945694740
service=sp3en0/spmgr expiration=945694740
service=sp3en0/sp_confgd expiration=945694740
service=sp3en0/spbgroot expiration=945694740
```

Examine the AIX date/time stamps of some of the keyfiles managed by Per Node Key Management. This provides a "before" view of the keyfiles' file system attributes. Should Per Node Key Management update the keyfiles' key versions, the date/time stamps and the file size will change along with the list of key versions stored within the file.

```
sp3en0/ # ls -l /spdata/sys1/keyfiles/ssp/sp3en0/
total 72
-rw------- 1 root system 106 Nov 11 10:33 css
-rw------- 1 root system 114 Nov 11 10:33 hardmon
-rw------- 1 root system 110 Nov 11 10:33 pmand
-rw------- 1 root system 106 Nov 11 10:33 sdr
-rw------- 1 root system 120 Nov 11 10:33 sp_confgd
-rw------- 1 root system 116 Nov 11 10:33 spbgroot
-rw------- 1 root system 110 Nov 11 10:33 spmgr
-rw------- 1 root system 120 Nov 11 10:33 switchtlb
-rw------- 1 root system 112 Nov 11 10:33 sysctl
```

Examine the DCE account attributes that relate to one of the keyfiles Per Node Key Management will manage, and search for the lastchange attribute. This provides a "before" view of when account data was last changed. When Per Node Key Management (or any other authorized principal) updates the
account's key (password), the lastchange value is updated to show when the change was made and what principal made the change.

```
sp3en0/ # dcecp -c account show ssp/sp3en0/css | grep last
[lastchange /.../sp_cell/ssp/sp3en0/css 1999-11-11-10:33:15.000-05:00I-----]
```

Repeat the AIX file attributes and DCE account attribute "before" checks for the second service in the example, GSmonitor.

```
sp3en0/ # ls -l /spdata/sys1/keyfiles/LoadL/sp3en0/
total 56
-rw------- 1 root system 122 Nov 11 10:32 GSmonitor
-rw------- 1 root system 112 Nov 11 10:32 Kbdid
-rw------- 1 root system 116 Nov 11 10:32 Master
-rw------- 1 root system 124 Nov 11 10:32 Negotiator
-rw------- 1 root system 116 Nov 11 10:33 Schedd
-rw------- 1 root system 116 Nov 11 10:33 Startd
-rw------- 1 root system 118 Nov 11 10:33 Starter
sp3en0/ # dcecp -c keytab cat | grep GSmonitor
sp3en0/ # dcecp -c keytab show /.../sp_cell/hosts/sp3en0/config/keytab/LoadL/sp3en0/GSmonitor
{uuid 435b5182-984d-11d3-b8ea-02608c2d4a7f}
{annotation {}}
{storage /spdata/sys1/keyfiles/LoadL/sp3en0/GSmonitor}
{/.../sp_cell/LoadL/sp3en0/GSmonitor des 1}
{/.../sp_cell/LoadL/sp3en0/GSmonitor des 2}
sp3en0/ # dcecp -c account show LoadL/sp3en0/GSmonitor | grep last
[lastchange /.../sp_cell/LoadL/sp3en0/GSmonitor 1999-11-11-10:32:54.000-05:00I-----]
```

Stop Per Node Key Management, and then start it to trigger the check:

```
sp3en0/ # stopsrc -s spnkeyman
0513-044 The spnkeyman Subsystem was requested to stop.
sp3en0/ # lssrc -s spnkeyman
Subsystem  Group  PID  Status
spnkeyman  inoperative
sp3en0/ # startsrc -s spnkeyman
0513-059 The spnkeyman Subsystem has been started. Subsystem PID is 30164.
```

Get the current date/time to compare it against date/time stamps of attributes to check. Any changes made by Per Node Key Management will be relative to this data.

```
sp3en0/ # date
Sun Dec 19 11:07:03 EST 1999
```

errpt shows no error conditions for Per Node Key Management processing.
Reexamining the AIX file attributes reveals changes to the files managed by Per Node Key Management as a result of its processing. Likewise, changes are made to the DCE account attributes. Notice that, in this example, there are three unchanged files: hardmon, sdr, and sysctl. Each of these keyfiles is managed by the server that owns them, not by Per Node Key Management.

A new key version appears in the keyfiles that were updated:

The lastchange data has also been updated to reflect modifications to the account.

The same type of checks are performed for the other service in the example, GSmonitor.
3.7 Split configuration for DCE clients

When configuring machines as DCE clients, you can split the configuration steps into two parts: An admin-only part, and a local-only part. The admin-only part is done by someone with DCE cell administrator credentials and can be done from any machine that is already a member of the DCE cell. This step preconfigures the client machine into the DCE cell. The local-only part is then done by the local system administrator of the client machine and only requires root-level authority on that machine.

The DCE split configuration mechanism maps quite nicely to the way in which nodes are normally installed in an SP system. In the standard node install process, information is first entered into the SP System Data Repository (SDR), and then the nodes are installed. If PSSP is enabled for DCE, in addition to the data being entered into the SDR, the admin-only portion of the DCE node configuration is done first. During node install, the local-only portion is performed on the nodes by the /usr/lpp/ssp/bin/spauthconfig script, which is called as part of the normal node customization process.

If you plan to use DCE with PSSP, you will need to keep in mind this two-step process for DCE node configuration. Note that, if you are already a DCE cell administrator, this may represent a significant change from the way you normally configure and unconfigure DCE clients.
3.8 The DCE security server bootstrap file

When a DCE client needs to communicate with a DCE Security server, it typically does so by first looking up the server name in the CDS namespace. If these CDS lookups are frequent, the impact on performance can be significant.

The DCE Security server bootstrap file, `/opt/dcelocal/etc/security/pe_site`, is available on all DCE clients and servers in the cell. It contains the names and locations of the cell's DCE Security server(s). When the `TRY_PE_SITE` environment variable is set to 1, the DCE client first attempts to locate a DCE Security server using the information in the `pe_site` file, thus, bypassing the CDS lookup. If that fails, the DCE client will then try to find the DCE Security server with a normal CDS namespace lookup. In general, locating a DCE Security server using the `pe_site` file is faster than a CDS namespace lookup.

The `TRY_PE_SITE` variable is, typically, set in the `/etc/environment` file:

```
sp3en0/ # cat /etc/environment
...
ODMDIR=/etc/objrepos
DOCUMENT_SERVER_MACHINE_NAME=localhost
DOCUMENT_SERVER_PORT=49213
CGI_DIRECTORY=/var/docsearch/cgi-bin
DOCUMENT_DIRECTORY=/usr/docsearch/html
# IMNSearch DBCS environment variables
IMQCONFIGSRV=/etc/IMNSearch
IMQCONFIGCL=/etc/IMNSearch/dbcshelp
RPC_UNSUPPORTED_NETIFS=css0
DCE_USE_WCHAR_NAMES=1
TRY_PE_SITE=1
```

When you configure your SP system for DCE security, this entry is added to the `/etc/environment` file for you.

---

**Important**

Changes to your DCE cell are not automatically reflected in the `/opt/dcelocal/etc/security/pe_site` file. For example, if you configure your cell for high-availability by adding additional DCE Security Replica servers and Secondary CDS servers, you need to run `/usr/bin/chpesite` on all DCE clients to add the entries for the additional DCE Security replica server(s) into the `pe_site` file.
3.9 Network interface routing considerations

By default, DCE servers use and advertise all available network interfaces. This process is referred to as binding. For performance reasons, it may be desirable to exclude some interfaces from binding if there are clients that do not have a route to those interfaces. This implies that you have a good understanding of your network topology. Care must be taken not to inadvertently disable an interface through which a DCE client is solely dependent on reaching a DCE server.

DCE provides the following environment variables to exclude specific interfaces:
- RPC_UNSUPPORTED_NETIFS
- RPC_UNSUPPORTED_NETADDRES

The first environment variable allows you to exclude the interfaces by interface name, and the second allows you to exclude the interfaces by IP address. For example:

- To exclude the en1 and tr1 interfaces:
  ```
  export RPC_UNSUPPORTED_NETIFS=en1:tr1
  ```
- To exclude the interface corresponding to IP address 123.45.67.89:
  ```
  export RPC_UNSUPPORTED_NETADDRES=123.45.67.89
  ```

It is best to pick one form of environment variable and use it consistently throughout your cell. RPC_UNSUPPORTED_NETIFS is the most common one in use. The recommended way to set it is by editing the /etc/environment file so that all processes in the system inherit this value by default.

If your SP system is contained in the DCE cell and you are using file collections, you can add /etc/environment to the user.admin collection and have supper propagate it to all of your nodes.

3.10 Options for setting up DCE

You now have a number of choices when it comes to using DCE on your SP system. They are:

- Do not use it at all.
- Create a new self-contained cell within the SP system.
- Integrate the SP system into an existing cell.
Before choosing one of these options, you need to consider your current environment as well as your future growth plans. Time spent in up-front planning of a DCE cell will more than pay for itself when the time comes to implement and support the cell. The following options provide a starting point for your planning activity:

**Do not use it at all**

DCE adds complexity and an extra layer of administration to your SP system. If you have no need for the added security that DCE provides, you can choose not to install or enable it. Besides DCE, there are two other security options available to you:

- **Kerberos Version 4**
  
  This is the standard option for earlier releases of PSSP. It is also known as *compatibility mode* because it provides the migration entry point for systems migrating from earlier releases of PSSP, such as PSSP 3.1. This option is a good second choice if you are not ready to take on the additional tasks involved with deploying DCE.

- **Standard AIX**
  
  This option is best suited for a self-contained SP system that is completely isolated from external networks and has adequate physical access controls. It is the simplest option to install and administer. It also provides a good starting point for the initial installation of an SP system, provided the above-mentioned controls are in place. Once the installation is complete, you can migrate the SP system to a higher level of security and then connect it to external networks.

**Create a new, self-contained cell within the SP system**

You can think of this option as entry-level DCE. It is a good way to get some exposure to DCE without having to deal with the complexities of managing a large-scale DCE cell.

With this option, you configure the CWS as the DCE Security Master server and the Primary CDS server, and your nodes become DCE clients. You will need to set up the master servers on the CWS prior to choosing this as a security option, but PSSP will handle making the nodes DCE clients.

Once your cell is in production, you may also want to configure DCE Security Replica servers and Secondary CDS servers for higher availability or enhanced performance. For example, you could configure the first node on each ethernet segment as a DCE Security Replica and Secondary CDS server.
Integrate the SP system into an existing cell
This option assumes that you already have an existing DCE cell and that you want to integrate your SP system into it. How you enable this option depends on how your existing cell is configured:

- All DCE servers are located on machines external to the SP system.
  This is the simplest configuration to work with. There are four main steps:
  a. If necessary, upgrade the cell to the same level of DCE required by PSSP.
  b. Verify that the CWS and all DCE-enabled nodes have network routes to the servers in your cell.
  c. Configure your CWS as either a DCE client only or as a DCE Security Replica server and/or a Secondary CDS server.
  d. Implement the DCE security state.

- All DCE servers are located on machines (nodes) internal to the SP system.
  This option is a bit more challenging. There are five main steps:
  a. Relocate the DCE Security Master server and the Primary CDS server to a machine or machines external to the SP system. This is required to prevent a circular dependency between the CWS and the DCE server node. For example, if both are down, the CWS may hang waiting for the DCE server node to come up, but the DCE server node cannot be powered on using PSSP if the CWS is down.
  b. If necessary, upgrade the cell to the same level of DCE required by PSSP.
  c. Verify that the CWS and all DCE-enabled nodes have network routes to the servers in your cell.
  d. Configure your CWS as either a DCE client only or as a Security Replica server and/or a Secondary CDS server.
  e. Implement the DCE security state.

- Some DCE servers are on external machines; others are on internal ones.
  This is a combination of the other two configurations. The key consideration is where the DCE Security Master server and the Primary CDS server are located:
    - If the these servers are on a machine external to the SP system, proceed as though all DCE servers are machines external to the SP system.
- If the these servers are on a node in the SP system, proceed as though all DCE servers are on machines (nodes) internal to the SP system.

In Chapter 4, “Implementing a specific security state” on page 57, we show you how to install your SP system to your chosen security state. After that, in Chapter 5, “Changing from one security state to another” on page 131, we show you how to change from one security state to another.

### 3.11 An important note about partitioning

The SP security options can be enabled on a partition-by-partition basis. In other words, you can have DCE enabled in one partition, Kerberos 4 in a second, and Standard AIX in a third. The CWS must be configured with all security options used within the partitions of the SP system. Since the CWS is contained within all partitions and effectively connects all partitions together, the entire SP system is no more secure than its least secure partition.

### 3.12 Suggested reading for DCE

Redbooks:
- *DCE Cell Design Considerations*, SG24-4746
- *Administering IBM DCE and DFS Version 2.1 for AIX and OS/2 Clients*, SG24-4714
- *DCE and DFS Performance Tuning and Problem Determination on AIX and OS/2 Warp*, SG24-4949

Other publications:

---

**Note**

The DCE documentation is shipped as softcopy on the DCE product media.

- *Distributed Computing Environment for AIX: Quick Beginnings*
- *IBM Distributed Computing Environment for AIX: Introduction to DCE*
- *IBM Distributed Computing Environment for AIX: Administration Guide -- Introduction*
- *IBM Distributed Computing Environment for AIX: Administration Guide -- Core Components*
• RS/6000 SP Planning Volume 2, Control Workstation and Software Environment, GA22-7281

• Parallel System Support Programs for AIX Administration Guide Version 3 Release 2, SA22-7348

External publications:
Chapter 4. Implementing a specific security state

Now that you know about the different security states, the question of how you can implement a specific security state on your SP system remains. This chapter provides you with step-by-step procedures for setting up your SP system to use any desired security state. Note that, when you decide on a particular security state, you are not locked into it. Chapter 5, “Changing from one security state to another” on page 131, will give you directions on how to go from one security state to another.

4.1 Overview of the examples

In the following paragraphs, we discuss the security-related installation steps of five examples. There, we give you a detailed view into the configuration of SP security that should help you make the right choices in the numerous security options offered by PSSP 3.2. The described installation steps cover the setup of the three common security states, Minimal, Compatibility, and DCE, as defined in Section 2.4, “SP security states” on page 16. For the DCE security state, we go more into detail. We explain how you can configure and install a DCE cell on your SP system in the last three examples.

- **Example 1: Minimal Security** covers the simplest security state you can configure on your SP system.
- **Example 2: Compatibility Security** covers the security state that was used by default in PSSP 3.1 and earlier versions.
- **Example 3: Simple DCE cell** shows you how to implement the DCE security state and how to configure a DCE cell in the simplest setup you can use with PSSP 3.2.
- **Example 4: Advanced DCE cell** expands the configuration of Example 3 to use a DCE Security Replica Server, a Secondary CDS Server, and an SP administrative principal in your system.
- **Example 5: Existing DCE cell** shows you how to integrate your SP system in an already working DCE cell.

You can see a comparison of all five examples and their security states in Table 5.

<table>
<thead>
<tr>
<th>Example</th>
<th>Security state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1: Minimal Security</td>
<td>Minimal</td>
</tr>
</tbody>
</table>

You can see a comparison of all five examples and their security states in Table 5.
### 4.2 Our lab environment

Before we go into the implementation processes of the different examples, you might want to know more about the hardware and network environment we used in this project. The SP system, shown in Figure 3 on page 59, consists of one frame with twelve nodes: One high node, ten thin nodes, and one wide node. The networks used by the nodes are the switch network and the SP administrative ethernet with the control workstation (CWS) connected to it. The CWS also has a token ring interface that connects to two external systems that are used as DCE security server and DCE client for the SP external DCE cell in Section 4.8, “Example 5: Existing DCE cell” on page 112.
The CWS and the external nodes are installed with AIX 4.3.3 and the latest available PTF sets. The lppsource (directory name: aix433) also has this AIX level.

Note

There must not be any DCE filesets older than release 3.1 contained in the lppsource.

To keep it simple, we have not configured PSSP user management, supper, or automounter. Additional PSSP features, such as GPFS or LoadLeveler, are also not included in our installation.

4.3 A template for the security installation process

Before we continue with a detailed step-by-step description of the security-related installation steps in the specific examples, it can be very useful to get an overview of the main procedures of this process. All
examples go through the following seven steps, and you can use this list as a template to implement any security state on your system:

1. Install the authentication method server(s) on the CWS, SP nodes, and/or systems external to the SP, for instance, your DCE servers.
2. Configure the CWS as a client for the authentication method.
3. Set the authentication method to be automatically installed/configured on the nodes within the desired partition.
4. Set the authentication methods to be used for AIX remote command authorization for the root user.
5. Ensure that the authentication configuration is updated on the nodes within the partition (either by rebooting the nodes and letting the pssp_script command do the job for you or by using the dsh command to change the security authentication on each node in the partition).
6. Enable the authentication method for use within the AIX remote commands in the partition.
7. Enable the authentication method for use within SP Trusted Services in the partition.

### 4.4 Example 1: Minimal security

In this example, we go through the security-related installation steps of the PSSP Installation and Migration Guide, GA22-7347, for a new installation. We install the SP system with the simplest security state offered by PSSP 3.2, that is, the Minimal security state defined in Chapter 2, “SP security concepts and terminology” on page 11. Recall, from Table 2 on page 15, that this configures only AIX standard methods for authentication as well as authorization for remote commands and no authentication method for SP Trusted Services. Enabling no authentication method, SP Trusted Services, which are critical for installation, administration, and operation of the SP system, still function correctly using methods, such as “client must be root” or “client must be authorized by an ACL”. In this scenario, you do not have to deal with Kerberos V4 or DCE; so, this setup can be a good starting point for a fast SP installation. On the other hand, implementing this security state requires.rhosts files to grant root access through remote commands. It is,
therefore, not a recommended configuration for an untrusted environment. The security state for this scenario is shown in Table 6.

Table 6. SP security state of Example 1 (Minimal Security)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Security State: Minimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auth_install</td>
</tr>
<tr>
<td>Minimal Security</td>
<td>std</td>
</tr>
</tbody>
</table>

High-level outline for example 1

1. Planning considerations
2. Installation steps

4.4.1 Example 1: Planning considerations

There are no special planning considerations for this security state compared to a normal RS/6000 environment. Authentication is based on IP address or a user ID and password. Access is granted based on the .rhosts file in the user's home directory, which contains the list of authorized hosts and user names.

4.4.2 Example 1: Installation steps

Go through the installation steps provided in Chapter 2, “Installing and Configuring a New RS/6000 SP System” of the PSSP Installation and Migration Guide, GA22-7347.

1. In “Step 14: Copy the AIX LPP Images and Other Required AIX LPPs and PTFs”, you do not need the dce.* filesets. Continue with the normal installation steps.


Issue the chauthent command, and confirm the setting with the lsauthent command:

```
sp3en0: chauthent -std
sp3en0: lsauthent
Standard Aix
```
Exploiting RS/6000 SP Security: Keeping It Safe


Issue the chauthts command without any parameters:

```
sp3en0 # chauthts
sp3en0 #
```

4. Skip “Step 22: Set the Authentication Method for SP Trusted Services on the Control Workstation” and go on with your installation proceeding with the following steps as described in the PSSP Installation and Migration Guide, GA22-7347, to “Step 39: Select Security Capabilities Required on Nodes”.

To check your current security settings, you can issue the splstdata -p command:

```
sp3en0/ # splstdata -p
List System Partition Information
System Partitions:
------------------
s3en0
Syspar: s3en0
-------------------------------------------------------------------------------
syspar_name s3en0
ip_address 192.168.3.130
install_image default
syspar_dir ""
co...
In this step, we set the auth_install attribute in the SDR. As described in Section 2.3, “SP security attributes” on page 14, this attribute defines the set of authentication capabilities for the node that PSSP uses to determine if it needs to install and/or configure authentication mechanisms on the nodes. Since all standard AIX authentication methods come with a minimal AIX installation by default, setting this attribute to std does not have any effect on the nodes. Nevertheless, setting this attribute to std can help with problem determination or problem source identification.

Issue the smitty spauth_config command to get to the SMIT panel, RS/6000 SP Security; select Select Security Capabilities Required on Nodes, insert your system partition name, and enter std as the authentication method.

You can also use the following command:

spsetauth -i -p <syspar> std

After this, you can verify the change of your security settings with the splstdata -p command. The settings should now look like the following:
5. Skip **Step 40**: Create DCE Hostnames (Required for DCE) to **Step 42**: Configure Admin Portion of DCE Clients, and proceed with **“Step 43: Select Authorization Methods for AIX Remote Commands”**.

**Issue the `smitty spauth_config` command and select Select Authorization Methods for AIX Remote Commands from the “RS/6000 SP Security” SMIT panel; then, insert your system partition name and specify std as the authorization method.**

```
sp3en0/ # splstdata -p
List System Partition Information

System Partitions:
------------------
sp3en0

Syspar: sp3en0

-------------------------------------------------------------------------------
<table>
<thead>
<tr>
<th>syspar_name</th>
<th>sp3en0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>192.168.3.130</td>
</tr>
<tr>
<td>install_image</td>
<td>default</td>
</tr>
<tr>
<td>syspar_dir</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>code_version</td>
<td>PSSP-3.2</td>
</tr>
<tr>
<td>auth_install</td>
<td>std</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>std</td>
</tr>
<tr>
<td>ts_auth_methods</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>auth_methods</td>
<td>std</td>
</tr>
</tbody>
</table>
-------------------------------------------------------------------------------

sp3en0/ # smitty spauth_config
-> Select Authorization Methods for AIX Remote Commands

Select Authorization Methods for AIX Remote Commands

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

* System Partition name           sp3en0  +
* Authorization Methods           std     +
```

**Or use the following command:**

```
spsetauth -d -p <syspar> std
```

Since we did not configure any DCE hostnames in the SDR, this results in the following output:

```
updauthfiles: Attention! DCE hostname information obtained from SDR for the CMS is not valid. It will be ignored for the .k5login entry.
```
6. Skip "Step 44: Configure SP Trusted Services to Use DCE Authentication (Required for DCE)" and "Step 45: Create SP Services DCE Keyfiles (Required for DCE)" and proceed to “Step 46: Enable Authentication Methods for AIX Remote Commands”.

Select Enable Authentication Methods for AIX Remote Commands from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to yes, and set the “Force change on nodes” flag to no. Select your system partition and std as authentication methods:

```
sp3en0/ # smitty spauth_config
-> Enable Authentication Methods for AIX Remote Commands
[Entry Fields]
| Enable on Control Workstation Only | yes |
| Force change on nodes             | no  |
| * You cannot select YES for both entries above. |
| System Partition name             | sp3en0 |
| Authentication Methods            | std  |
```

or use the following command:

```
chauthpar -p <syspar> std
```

7. In “Step 47: Enable Authentication Methods for SP Trusted Services”, select Enable Authentication Methods for SP Trusted Services from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to yes, and set the “Force change on nodes” flag to no. Select your system partition and “” (none) as authentication methods:

```
sp3en0/ # smitty spauth_config
-> Enable Authentication Methods for SP Trusted Services
[Entry Fields]
| Enable on Control Workstation Only | yes |
| Force change on nodes             | no  |
| * You cannot select YES for both entries above. |
| System Partition name             | sp3en0 |
| Authentication Methods            | std  |
```
You can also use the following command:

\texttt{chauthpts -p <syspar> -v std}

to enable the authentication methods for SP Trusted Services.

8. Skip \textit{"Step 48: Start the Key Management Daemon"}, and finish the normal installation process as described in the \textit{PSSP Installation and Migration Guide}, GA22-7347.

### 4.5 Example 2: Compatibility security

In this example, we go through the security-related installation steps of the \textit{PSSP Installation and Migration Guide}, GA22-7347, for a new installation. We install the SP system with the security state that was used by default setting for PSSP 3.1 and earlier versions as described in Section 2.4, “SP security states” on page 16. We call this the \textit{Compatibility security} state. This configures Kerberos V4 for authentication and authorization for remote commands and SP Trusted Services. Recall, from Chapter 2, “SP security concepts and terminology” on page 11, that this sets the security-related SDR attributes as shown in Table 7.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|l|}
\hline
\textbf{Scenario} & \textbf{Security State: Compatibility} \\
\hline
Compatibility Security & auth_install & auth_root_rcmd & ts_auth_methods & auth_methods \\
\hline
Compatibility Security & k4 & k4 & compat & k4:std \\
\hline
\end{tabular}
\caption{SP security state of Example 2 (Compatibility Security)}
\end{table}

High-level outline for example 2

1. Planning considerations
2. Installation steps

#### 4.5.1 Example 2: Planning considerations

To set up the compatibility security state, you can have four different configurations of the Kerberos V4 environment:

- CWS as the primary authentication server with Kerberos database

\begin{itemize}
\item CWS as the primary authentication server with Kerberos database
\end{itemize}
• CWS as a secondary authentication server with Kerberos database (a primary server must be initialized prior to this)
• CWS as a AFS authentication server (and AFS Client)
• CWS as an authenticated client (at least a Kerberos V4 primary server must be initialized prior to this)

We decided for this example to configure the CWS as a primary Kerberos V4 server, which is the most common configuration. Refer to the *RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment*, GA22-7281, for more information. It is necessary to have the whole SP system, including the CWS, in a single Kerberos realm. The authentication servers can be on any workstation in this realm, but not on your SP nodes.

To use Kerberos V4 for SP security, you must, at least, define one user principal as a system administrator, which can have any name with the instance, admin. We recommend to use the user principal, root.admin, by the root user on the CWS for installation and administration tasks of the SP system.

### 4.5.2 Example 2: Installation steps

Go through the installation steps provided in Chapter 2, “Installing and Configuring a New RS/6000 SP System”, of the *PSSP Installation and Migration Guide*, GA22-7347.

1. In "**Step 14:** Copy the AIX LPP Images and Other Required AIX LPPs and PTFs", you do not need the `dce.*` filesets. Continue with the normal installation steps.

2. In "**Step 18:** Set Authentication Methods for AIX Remote Commands on the Control Workstation", select `k4` and `std` as the authentication methods for remote commands. Issue the `chauthent` command, and use the `lsauthent` command to verify the correct settings.

   ```bash
   sp3en0 # chauthent -k4 -std
   sp3en0 # lsauthent
   Kerberos 4
   Standard Aix
   ```

   **Important**

   Issuing the `chauthent` command without any parameters turns off all authentication methods for this machine, effectively disabling remote logins for everyone.

4. Skip “Step 20: Initialize RS/6000 SP Kerberos V4” and go on to “Step 21: Configure DCE for the Control Workstation (Required for DCE)”. Select Compatibility as the authentication method for SP Trusted Services. Issue the chauthts command, and verify the settings with the lsauthts command:

```
sp3en0 # chauthts compat
sp3en0 # lsauthts
Compatibility
```

5. Skip “Step 22: Set the Authentication Method for SP Trusted Services on the Control Workstation”, and proceed with the normal installation steps, described in the PSSP 3.2: Installation and Migration Guide, GA22-7347, to “Step 39: Select Security Capabilities Required on Nodes”. Issue the smitty spauth_config command to get to the “RS/6000 SP Security” SMIT panel; select Select Security Capabilities Required on Nodes; insert your system partition name, and specify k4 as the authentication method.

```
sp3en0/ # smitty spauth_config
-> Select Security Capabilities Required on Nodes
```

Instead of using SMIT, you can use the following command:

```
spsetauth -i -p <syspar> k4
```

6. Go on with the installation process described in the PSSP 3.2: Installation and Migration Guide, GA22-7347, and skip “Step 40: Create DCE Hostnames (Required for DCE)”, “Step 41: Update the SDR with DCE Master Security and CDS Server Hostnames (Required for DCE)”, and

Select **Select Authorization Methods for AIX Remote Commands** from the “RS/6000 SP Security” SMIT panel; insert your system partition name, and specify k4 as the authorization method:

You can also use the following command:

```
spsetauth -p <syspar> -d k4
```

Do not worry if you get the following output:

```
updauthfiles: Attention! DCE hostname information obtained from SDR for the CWS is not valid. It will be ignored for the .k5login entry.
```

7. Skip “Step 44: Configure SP Trusted Services to Use DCE Authentication (Required for DCE)” and “Step 45: Create SP Services DCE Keyfiles (Required for DCE)”, and go on to “Step 46: Enable Authentication Methods for AIX Remote Commands”.

Select **Enable Authentication Methods for AIX Remote Commands** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to no and the “Force change on nodes” flag to no. Select your system partition and specify k4 and std as the authentication methods.
8. **Step 47: Enable Authentication Methods for SP Trusted Services**.

Select **Enable Authentication Methods for SP Trusted Services** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **no** and the “Force change on nodes” flag to **no**. Select your system partition, and specify **compat** as the authentication method.

You can also use the following command:

```
chauthpts -p <syspar> -v compat
```

Verify the settings of the security state with the `splstdata -p` command. This should result in output similar to that shown in the following screen:
9. Skip “Step 48: Start the Key Management Daemon”, and finish the normal installation process as described in the *PSSP 3.2: Installation and Migration Guide*, GA22-7347.

### 4.6 Example 3: Simple DCE cell

In this scenario, we go through the security-related installation steps of the *PSSP Installation and Migration Guide*, GA22-7347, to install the SP system with the DCE security state. As described in Chapter 2, “SP security concepts and terminology” on page 11, this configures Kerberos V5 as the method for authentication as well as authorization for remote commands and SP Trusted Services. The security state for this scenario is shown in Table 8.

**Table 8. SP security state of Example 3 (DCE Security; PSSP only)**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Security State: DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auth_install</td>
</tr>
<tr>
<td>simple DCE cell</td>
<td>dce</td>
</tr>
</tbody>
</table>

We configure the DCE Security server and CDS server on the CWS for PSSP authentication only and do not use any other features, such as user management or DFS. This simple DCE cell only contains the SP system with the CWS used as DCE Security Server and CDS Primary Server and all nodes configured as DCE clients. No DCE Security Replica Server or Secondary CDS Server is configured in the cell.
High-level outline for example 3

1. Planning considerations
2. Dealing with mksysb images in a DCE environment
3. Setting up the CWS as DCE server
4. Installation steps

4.6.1 Example 3: Planning considerations

Like most distributed systems, DCE is highly-dependent on the integrity of the underlying network. Before configuring any DCE component, verify that the following are properly configured and functioning correctly:

- Host name settings (hostname)
- Forward and reverse name resolution (nslookup, host, /etc/resolv.conf, /etc/netsvc.conf, and /etc/hosts)
- Network routings (netstat -rn, ping, and traceroute)

Also, verify that your date, time, and timezone settings are correct. DCE is extremely sensitive to time discrepancies.

DCE processes write their data (including core files) into /var/dce. These core files can be quite large. For this reason, we recommend creating a /var/dce filesystem at least 200 MB in size, especially on machines that are DCE servers. However, if you do not have the disk space or are not concerned about preserving DCE core files, you can go with the recommended minimum size of 32 MB.

During the node installation process, PSSP installs and configures a DCE client on each node. Unfortunately, this process does not contain the creation of a separate /var/dce filesystem on the nodes. To prevent the use of the root

---

Note

This is a limited DCE environment. Going this route may limit you if you want to expand your cell at a later date. Refer to the RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment, GA22-7281, and the other DCE examples for more complex DCE configurations. This DCE configuration is also not recommended if you need a highly-available environment because the CWS is a single point of failure as long as no DCE Security Replica Servers are configured in the DCE cell (see Example 4 and Example 5).
filesystem by the DCE client, you can add the following line to your script.cust file:

```
crfs -v jfs -g 'rootvg' -a size='64000' -m /var/dce -A 'yes' -p 'rw' -t 'no'
```

In the following step-by-step-procedure, you will be asked to supply the following:

- A name for your DCE cell
- A name for your DCE cell administrator (typically cell_admin)

For this first scenario, we assume that the SP administrator with root authority will also be the DCE cell administrator. This seems like a fair assumption since the DCE cell exits only within the SP system. In the section following this example, we discuss other scenarios where this assumption may not be true. As explained in the *PSSP Administration Guide*, SA22-7348, you can decide to use other DCE principals that do not need root authorization for SP configuration and monitoring tasks if you want to establish a higher granularity of object protection on your SP system.

### 4.6.2 Dealing with mksysb images in a DCE environment

You will not be able to use your mksysb images of your nodes to clone new nodes in your system when a DCE client is configured on the node without a few reconfiguration steps in the cloning process because PSSP will not reconfigure the DCE hostname of the new node during installation. For this reason, you must deinstall the DCE client before booting the new node for the first time. You can automate these configurations in the script.cust file as shown in the following screen. This will deinstall the DCE client filesets after the image installation on the new node but before it boots for the first time. This way, the psspfb_script will be able to install and configure DCE properly.
To install a new node with a mksysb image containing a configured DCE client you must replace the number 9999 in the case statement with the new node's node number before you reboot the node for installation. Do not forget to change the number back again after the installation has succeeded. This procedure is not suitable for disaster recovery of the same node or for cloning to an existing node, that was configured in your DCE cell before, because the procedure does not remove the admin part of the DCE client configuration on the DCE server.

To restore a system image of the same node where it was created for disaster recovery reasons do not use this procedure. In this case you can run into problems with DCE when the keyfiles in the mksysb image are out of date. You can restore the keyfiles from your ADSM server to complete your disaster recovery.

To restore a system image on a node that was configured in your DCE cell before this process, you will also have to remove the admin part of the DCE client configuration. Refer to Section 8.12.2, “Unconfigure the node (admin-only)” on page 479, and Section 8.12.3, “Reconfigure the node (admin only)” on page 480, for a step-by-step procedure for the additional actions you must perform.

4.6.3 Setting up the CWS as DCE server

Perform the following steps to set up the CWS as a DCE server:

1. Set the `RPC_UNSUPPORTED_NETIFS` variable to exclude network interfaces that should not be used for DCE traffic by either the control workstation or the

```
# DCE mksysb image cloning cleanup procedure
#/usr/lpp/ssp/install/bin/node_number | read NN || NN=-1
case $NN in
  9999)  rm -rf /var/dce
         rm -rf /etc/dce
         rm -f /etc/krb5.conf
         rm -rf /var/ysact1
         rm -rf /spdata/sys1/keyfiles
         rm -rf /krb5
         # the rm commands are needed to trick DCE in thinking
         # it is not configured. It will then un-install.
         installp -u dce* ;;
esac
```

Figure 5. Node cloning procedure entries in script.cust
nodes (for example, interfaces on the nodes to which the control workstation has no route):

```
sp3en0/ # export RPC_UNSUPPORTED_NETIFS=tr1:en1:css0
sp3en0/ #
```

---

**Note**

DCE does not check for the existence of these network interfaces. Therefore, you can use a common setting for the control workstation and all nodes in your SP system.

2. You may also want to add the `RPC_UNSUPPORTED_NETIFS=tr1:en1:css0` line to the `/etc/environment` file on the control workstation and the nodes:

3. Before installing the DCE filesets, create the DCE related directories, filesystems, and mount points. We recommend that you use separate file systems for the DCE configuration files. A good starting point for the server installation on the CWS can be using the following parameters:

   a. Create filesystem: `crfs -v jfs -g rootvg -a size=’400000’ -m /var/dce -A ’yes’ -p rw -t no`

   b. Mount filesystem: `mount -v jfs /var/dce`

4. Remove any DCE version 2.x filesets from the lppsource directory:

```
sp3en0/ # rm -i /spdata/sys1/install/aix433/lppsource/dce*
...`

---

**Note**

Some DCE Version 2.x filesets are included on the AIX 4.3.3 media. If you copied the entire AIX 4.3.3 distribution into your lppsource directory, you will need to remove the DCE Version 2.x filesets.

5. Copy the DCE version 3.1 filesets into the lppsource directory:
These filesets are not part of the AIX 4.3.3 distribution. Instead, they can be found on the DCE Version 3.1 distribution media, which is a separately-purchasable product from IBM. If you did not use `bffcreate` or SMIT to copy the files to the `lppsource`, do not forget to rebuild the `.toc` file in the `lppsource` directory using the `inutoc` command.

6. Install the DCE filesets on the CWS using the `installp` command or SMIT. Follow normal software installation steps to install the DCE filesets. Table 9 contains the filesets that need to be installed.

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dce.cds</strong></td>
<td>DCE Cell Directory Services</td>
</tr>
<tr>
<td>dce.cds.rte</td>
<td>DCE Cell Directory Services</td>
</tr>
<tr>
<td>dce.cds.smit</td>
<td>DCE SMIT Cell Directory Services</td>
</tr>
<tr>
<td><strong>dce.client</strong></td>
<td></td>
</tr>
<tr>
<td>dce.client.rte.admin</td>
<td>DCE Client Administrative Tools</td>
</tr>
<tr>
<td>dce.client.rte.cds</td>
<td>DCE Client CDS Tools</td>
</tr>
<tr>
<td>dce.client.rte.config</td>
<td>DCE Client Configuration Tools</td>
</tr>
<tr>
<td>dce.client.rte.rpc</td>
<td>DCE Client RPC Tools</td>
</tr>
<tr>
<td>dce.client.rte.security</td>
<td>DCE Client Security Tools</td>
</tr>
</tbody>
</table>
To configure the DCE Security Server, issue, as root, the smitty dce command to get to the DCE main SMIT panel, and perform the following sequence of SMIT menu options:

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce.client.rte</td>
<td>DCE Client Services</td>
</tr>
<tr>
<td>dce.client.core.rte</td>
<td>DCE Client Services - FOR UPGRADES</td>
</tr>
<tr>
<td>dce.client.rte.time</td>
<td>DCE Client Time Tools</td>
</tr>
<tr>
<td>dce.client.rte.zones</td>
<td>DCE Client Time Zones</td>
</tr>
<tr>
<td>dce.client.smit</td>
<td>DCE SMIT Client Tools</td>
</tr>
<tr>
<td>dce.client.rte.pthreads</td>
<td>DCE Threads Compatibility Library</td>
</tr>
<tr>
<td>dce.client.rte.web</td>
<td>DCE Web Secure</td>
</tr>
<tr>
<td><strong>dce.security</strong></td>
<td></td>
</tr>
<tr>
<td>dce.security.smit</td>
<td>DCE SMIT Security Services</td>
</tr>
<tr>
<td>dce.security.rte</td>
<td>DCE Security Services</td>
</tr>
<tr>
<td><strong>dce.xdsxom</strong></td>
<td></td>
</tr>
<tr>
<td>dce.xdsxom.rte</td>
<td>X.500 API Library</td>
</tr>
</tbody>
</table>

Configure DCE Security and CDS servers after the DCE filesets have been installed. Note that this automatically configures a DCE client.
Figure 6. DCE Security Server configuration for example 3

At the "MASTER SECURITY Server" screen, fill in the fields as appropriate. Some conventions are:

- Choose a name for your dce cell in the “CELL name” entry field.
- Do not change the “Cell ADMINISTRATOR's account” data.
- Insert the symbolic TCP/IP hostname of the first TCP/IP address that appears after the loopback address when a netstat -i is issued, which, in most cases, should be an en* address, in the “Machine's DCE HOSTNAME” entry field.
- Select Yes to “Start components at System restart”
- Select Yes to “Clean up DCE at System restart”
- Do not change the “protocol” field.
- Use the value specified in the "Machine's DCE HOSTNAME" field as “Security Server Name”. Note the value used, as it will be required during the CDS server configuration process.
- Ignore the remaining fields.
Choose a password for the principal cell_admin, and enter it when prompted. It will take several minutes to configure.

8. After the Security Server is configured, you have to configure the CDS Server. Issue, as root, the smitty dce command to get to the DCE main SMIT panel, and perform the following sequence of SMIT menu options:

```
sp3en0/ # smitty dce
-> Configure DCE/DFS
   -> Configure DCE/DFS Servers
      -> CDS (Cell Directory Service) Server
         -> 1 initial

CDS (Cell Directory Service) Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
* CELL name [sp_cell]
* Cell ADMINISTRATOR's account [cell_admin]
Machine's DCE HOSTNAME [sp3en0]
* Start components at System restart
  * Clean up DCE at System restart
* Protocol
tcp udp +
MASTER SECURITY Server [sp3en0]
Minutes to wait between pe_site file updates [1440] #
Synchronize Clocks
  Time Server to Synchronize Clocks with []
  * LAN PROFILE [lan-profile]

Figure 7. CDS Server configuration for example 3

All the fields should be filled in automatically. However, verify that the "MASTER Security Server" field contains the same address that was used in the "Security Server Name" field during the Security Server configuration steps. You will be prompted for your cell_admin password again to start the configuration. This will take several minutes to configure.

9. DCE Security and CDS servers should now be configured and running. You can verify this with a few simple commands:

- Issuing ps -ef | grep dce should show at least five daemons:
  - dced - The security service client daemon
  - cdsadv - CDS part of the client, one per machine
  - cdsclerk - Used by cdsadv, one or more per machine
  - cddsd - The cds process manages the clearinghouse on the CDS server.
  - secd - The security server daemon.
- Issuing `klist` prints out all information concerning your DCE identity. Especially the second line (Global Principal:) of the output shows your current global DCE principal.

sp3en0/ # klist
DCE Identity Information:
  Global Principal: /.../sp_cell/hosts/sp3en0/self
  Cell:  c2c9bc24-888a-11d3-8bce-02608c2d4a7f /.../sp_cell
  Principal: 00000066-888a-21d3-8b00-02608c2d4a7f hosts/sp3en0/self
  Group: 0000000c-888a-21d3-b801-02608c2d4a7f none
  Local Groups: 0000000c-888a-21d3-b801-02608c2d4a7f none
             0000006b-888a-21d3-b501-02608c2d4a7f subsys/dce/sec-servers
             0000006d-888a-21d3-b501-02608c2d4a7f subsys/dce/dts-servers
...

- Issuing the command `cdsli -cworld` prints out the whole CDS namespace of the cell. This information is gathered from the `cdsd` process.

sp3en0/ # cdsli -cworld
  o p  /.:/cell-profile
  o   /.:/fs
  o   /.:/lan-profile
  o g  /.:/sec
  o g  /.:/sec-v1
  c   /.:/sp3en0_ch
  d   /.:/hosts
  d   /.:/hosts/sp3en0
  o b  /.:/hosts/sp3en0/cds-clerk
  o b  /.:/hosts/sp3en0/cds-server
  o b  /.:/hosts/sp3en0/config
  o p  /.:/hosts/sp3en0/profile
  o b  /.:/hosts/sp3en0/self
  ...
  o b  /.:/subsys/dce/sec/sp3en0
  d   /.:/users
4.6.4 Example 3: Installation steps

Go through the installation steps provided in Chapter 2, “Installing and Configuring a New RS/6000 SP System”, of the *PSSP 3.2: Installation and Migration Guide*, GA22-7347.

1. Proceed with the installation steps as described in the *PSSP Installation and Migration Guide*, GA22-7347, to **Step 18: Set Authentication Methods for AIX Remote Commands on the Control Workstation**. Make sure that you have set up your DCE server as described in Section 4.6.3, “Setting up the CWS as DCE server” on page 74.

Select k5 and std as the authentication methods for remote commands, and use the chauthent command. You can verify your settings with the lsauthent command.

```
sp3en0/ # chauthent -k5 -std
sp3en0/ # lsauthent
Kerberos 5
Standard Aix
```

--- Caution ---

Issuing the command chauthent without any parameters turns off all authentication methods for this machine, effectively disabling remote logins for everyone.

2. Skip **Step 19: Set Authentication Methods for AIX Remote Commands on the Control Workstation**. To continue with **Step 20: Initialize RS/6000 SP Kerberos V4**, you must first have a DCE server installed and configured on your system. Refer to the verification section, Section 4.6.3, “Setting up the CWS as DCE server” on page 74, if you are not sure about this.

--- Important ---

The following installation steps require DCE cell administrator credentials.

If the DCE server is up and running, log in as the cell_admin principal. It is normal to be prompted to change your password the first time you do this. You can reenter the initial password you have chosen when you set up the server.
Going through “Step 20.2: Create DCE Groups, Organizations, Principals and Accounts”, you have to use the `config_spsec` command and enter `cell_admin` for the cell administrator ID. When prompted, enter your password. This must be done using the `cell_admin` principal.

```bash
sp3en0/ # dce_login cell_admin
Enter Password:
Password must be changed!
DCE LOGIN SUCCESSFUL
Warning: This account has been marked by an administrator, recommending that the password be changed.
Do you wish to change now [y/n]? (y)
Enter New Password:
Re-enter New Password:
PASSWORD SUCCESSFULLY CHANGED
```

3. In “Step 20.3: Create SP Administrative Principals”, you must add an administrative principal to several DCE access groups. We recommend that you use the `cell_admin` principal as the administrative principal for the rest of the installation in this example. Make sure you are still logged in as `cell_admin`.

```bash
sp3en0/ # config_spsec -c -v
This command requires cell administrator authority. Continue? (y/n) y
Running "check_prereqs" subroutine ...  
Checking state of DCE ... 
Running "open_io" subroutine ... 
Running "parse_defaults" subroutine ... 
Parsing spsec_defaults file ... 
Running "parse_overrides" subroutine ... 
Running "create_default" subroutine ... 

Please enter cell administrator id to be added to ACL admin group: 
  cell_admin

... 
```

```bash
sp3en0/ # klist | grep Global
Global Principal: /.../sp_cell/cell_admin
sp3en0/ # dcecp -c group add sdr-admin -member cell_admin
sp3en0/ # dcecp -c group add hm-admin -member cell_admin
sp3en0/ # dcecp -c group add sdr-system-class-admin -member cell_admin
sp3en0/ # dcecp -c group add spsec-admin -member cell_admin
sp3en0/ # dcecp -c group add hm-control -member cell_admin
```
Later in the installation process, it can be necessary to add additional membership to more access groups, for example, to be authorized to issue switch commands.

---

**Important**

The following installation steps require **DCE self-host** credentials.

4. Because the `create_keyfiles` command must be run with the self-host principal in "Step 20.4: Create Control Workstation Specific Keyfiles", log out of your `cell_admin` principal, and issue the `create_keyfiles` command. You can verify your current DCE identity using the `klist` command.

```
sp3en0/ # exit
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # create_keyfiles -c -v
...
```

5. Go on to “Step 21: Configure DCE for the Control Workstation (Required for DCE)”, and select **Kerberos Version 4** as the authentication method for SP Trusted Services. Use the `chauthts` command, and verify the settings with the `lsauthts` command.

```
sp3en0/ # chauthts dce
sp3en0/ # lsauthts DCE
```

---

**Important**

The following installation steps require **SP administrator (cell_admin)** credentials.

6. Because you have changed the DCE credentials of the SP administrative principal `cell_admin` prior to “Step 22: Set the Authentication Method for SP Trusted Services on the Control Workstation” you must log in as `cell_admin` again to get the new credentials. You can verify that all access groups are added to your principal with the `klist` command.
7. Continue with the normal installation steps to “Step 39: Select Security Capabilities Required on Nodes”.

   Issue the smitty spauth_config command to get to the “RS/6000 SP Security” SMIT panel, and select Select Security Capabilities Required on Nodes. Insert your system partition name and specify dce as the authentication method.

```
sp3en0/ # dce login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep -p Groups
DCE Identity Information:
Warning: Identity information is not certified
Global Principal: /.../sp_cell/cell_admin
Cell: 4b6608a0-9842-11d3-a5a0-02608c2d4a7f /.../sp_cell
Principal: 00000064-9842-21d3-a500-02608c2d4a7f cell_admin
Group: 0000006c-9842-21d3-a501-02608c2d4a7f none
Local Groups: 0000006c-92c2-21d3-a501-0004ac5e69d6 none
0000065d-92c2-21d3-9d01-0004ac5e69d6 subsys/dce/dts-servers
00000086-9397-21d3-9d01-0004ac5e69d6 sdr-admin
00000080-9397-21d3-9d01-0004ac5e69d6 hm-admin
0000007c-9397-21d3-9d01-0004ac5e69d6 spec-s-admin
00000088-9397-21d3-9d01-0004ac5e69d6 sdr-system-class-admin
00000081-9397-21d3-9d01-0004ac5e69d6 hm-control
```

You can also use the following command:

```
spsetauth -i -p <syspar> dce
```

8. In “Step 40: Create DCE Hostnames (Required for DCE)”, the reliable hostname of the nodes are entered as DCE hostname in the SDR. Select Create DCE hostnames from the “RS/6000 SP Security” SMIT panel, or use the create_dcehostname command.
You can verify the creation of the DCE hostnames in the SDR by issuing the `splstdata -n` command.

```
sp3en0/ # splstdata -n
List Node Configuration Information

node# frame# slot# slots initial_hostname reliable_hostname dce_hostname
description default_route processor_type processors_installed description
----------------- --------------- -------------- -------------------- ---------------
1 1 1 4 sp3n01 sp3n01 sp3n01 192.168.3.130 MP
... 5 1 5 1 sp3n05 sp3n05 sp3n05 192.168.3.130 UP

```

9. To insert the DCE and CDS server hostnames in the SDR, go to **“Step 41: Update the SDR with DCE Master Security and CDS Server Hostnames (Required for DCE)”**, and select **Update SDR with DCE Master Security and CDS Server Hostnames** from the “RS/6000 SP Security” SMIT panel. Insert the hostname of the DCE Master Security server and the Primary CDS server in the appropriate entry fields.

```
sp3en0/ # smitty spauth_config
-> Update SDR with DCE Master Security and CDS Server Hostnames

Update SDR with DCE Master Security and CDS Server Hostnames

Type or select values in entry fields. Press Enter AFTER making all desired changes.

[Entry Fields]
Master Security Server hostname [sp3en0]
CDS Server hostname [sp3en0]
```

Or use the following command:
You can verify the settings in the SDR with the `splstdata -e` command.

```
sp3en0/ # splstdata -e
...
sec_master  sp3en0
cds_server  sp3en0
cell_name   /.../sp_cell
...
```

10. In "Step 42: Configure Admin Portion of DCE Clients", you create the needed DCE security and CDS entries for the SP nodes. Select Configure DCE Clients (Admin portion) from the "RS/6000 SP Security" SMIT panel, and enter `cell_admin` as the administrator ID and "/./lan-profile" as the profile ID.

```
sp3en0/ # smitty spauth_config
  -> Configure DCE Clients (Admin portion)
   Configure DCE Clients (Admin portion)
   Type or select values in entry fields.
   Press Enter AFTER making all desired changes.

   Cell Administrator id [cell_admin]
   Lan Profile id [/./lan-profile]
```

Instead of using SMIT, you can use the following command:

```
setupdce -c cell_admin -l /./lan-profile
```

**Important**

The next installation step requires the DCE cell administrator password.

When you are prompted, enter the password for the `cell_admin` principal. After the configuration of all your DCE hosts has ended successfully, you can verify the creation of the entries with the `dcecp -c cell show` command.
Chapter 4. Implementing a specific security state

This commands provides a view of all defined hosts in a DCE cell. You should now see all your SP nodes including the CWS in the output.

11. Go on with “Step 43: Select Authorization Methods for AIX Remote Commands”, and issue the smitty spauth_config command to get to the SMIT security menu. Select Select Authorization Methods for AIX Remote Commands. In the next panel, insert your system partition name, and specify dce and std as the authorization methods.

Or use the following command:

```
spsetauth -d -p <syspar> dce std
```

Note

You need to use the two methods, dce and std, at this point because you must have at least one common method defined for the system partition and the remote commands. Since we have not yet changed the authentication method for the partition, it is still set to std. At a later time, we will change these methods to only use dce.

12. In “Step 44: Configure SP Trusted Services to Use DCE Authentication (Required for DCE)”, select Configure SP Trusted Services to use DCE Authentication from the “RS/6000 SP Security” SMIT panel, and enter y
when prompted. Then, enter `cell_admin` as the administrator ID and the password when required.

```
sp3en0/# smitty spauth_config
 -> Configure SP Trusted Services to use DCE Authentication

This command requires cell administrator authority. Continue? (y/n) y

Running "check_preregs" subroutine ...
Checking state of DCE ...
Running "open_io" subroutine ...
Running "parse_defaults" subroutine ...
Parsing spsec defaults file ...
Running "parse_overrides" subroutine ...
Running "create_default" subroutine ...

Please enter cell administrator id to be added to ACL admin group: cell_admin

Adding cell administrator id to group spsec-admin ...
Creating org - spsec-services ...
Creating group - spsec-services ...
Running "create_accts_on_cws" subroutine ...

Your cell administrator password is required to create accounts.
Please enter your cell administrator password: ...
```

Instead of using SMIT, you can use the following command:

```
config_spsec -v
```

**Important**

The following installation step requires DCE self-host credentials.

13. To continue with "Step 45: Create SP Services DCE Keyfiles (Required for DCE)", you must exit your `cell_admin` principal. This step has to be done with the self-host principal.

```
sp3en0/# klist | grep Global
Global Principal: /.../sp_cell/cell_admin
sp3en0/# exit
sp3en0/# klist | grep Global
Global Principal: /.../sp_cell/hosts/sp3en0/self
```

When you run “Create SP Services Keyfiles” from the “RS/6000 SP Security” SMIT panel, you should get an output, such as the one shown in the next screen:
You can also use the following command:

create_keyfiles -v

---

**Important**

The remaining installation steps require SP administrator (cell_admin) credentials.

14. In “Step 46: Enable Authentication Methods for AIX Remote Commands”, log in as cell_admin again to have the right credentials to change attributes of the SDR. Then, issue the `smitty spauth_config` command, and select **Enable Authentication Methods for AIX Remote Commands** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition, and set **k5** and **std** as the authentication methods.

```
sp3en0/ # smitty spauth_config
-> Create SP Service Keyfiles
  Running "check_prereqs" subroutine ...
  Checking state of DCE ...
  Running "parse_defaults" subroutine ...
  Parsing spsec_defaults file ...
  Running "parse_overrides" subroutine ...
  Running "create_keys" subroutine ...
  Keyfile /spdata/sys1/keyfiles/LoadL/sp3en0/Master already exists.
  Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/sspctl already exists.
  ...
  Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/sysctl already exists.

The remaining installation steps require SP administrator (cell_admin) credentials.
```

```
sp3en0/ # dce_login cell_admin
Enter Password: DCE LOGIN SUCCESSFUL
sp3en0/ # smitty spauth_config
-> Enable Authentication Methods for AIX Remote Commands
  Enable Authentication Methods for AIX Remote Commands
  Type or select values in entry fields.
  Press Enter AFTER making all desired changes.

  [Entry Fields]
  Enable on Control Workstation Only yes +
  Force change on nodes no +
  * You cannot select YES for both entries above.

  * System Partition name
  * Authentication Methods k5 std +
```

Chapter 4. Implementing a specific security state  89
Instead of using SMIT, you can use the following command:

```bash
chauthpar -c -p <syspar> k5 std
```

15. In “Step 47: Enable Authentication Methods for SP Trusted Services”, you do this by selecting **Enable Authentication Methods for SP Trusted Services** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition and **dce** as the authentication methods.

You can also use the following command:

```bash
chauthpts -c -p <syspar> dce
```

This can take several minutes.

16. You can now change the authorization methods for the AIX remote commands to only use dce (remember Step 11 of this procedure). Issue the `smitty spauth_config` command to get to the SMIT security menu, and select **Select Authorization Methods for AIX Remote Commands**. In the next panel, insert your system partition name, and specify **dce** as the authorization method.
You can also use the following command:

```
spsetauth -d -p <syspar> dce
```

Verify your SRD settings with the `splstdata -p` command.

```
sp3en0/ # splstdata -p
List System Partition Information

System Partitions:
------------------
sp3en0
Syspar: sp3en0

<table>
<thead>
<tr>
<th>syspar_name</th>
<th>sp3en0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>192.168.3.130</td>
</tr>
<tr>
<td>install_image</td>
<td>default</td>
</tr>
<tr>
<td>syspar_dir</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>code_version</td>
<td>PSSP-3.2</td>
</tr>
<tr>
<td>haem_cdb_version</td>
<td>20111964,28101999,0</td>
</tr>
<tr>
<td>auth_install</td>
<td>dce</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>dce</td>
</tr>
<tr>
<td>ts_auth_methods</td>
<td>dce</td>
</tr>
<tr>
<td>auth_methods</td>
<td>k5:std</td>
</tr>
</tbody>
</table>
```

17. In "Step 48: Start the Key Management Daemon", use the `spnkeyman_start` command to start the key management daemon on the CWS. You can verify that the daemon is running with the `lssrc` or `ps` command.

```
sp3en0/ # spnkeyman_start
0513-059 The spnkeyman Subsystem has been started. Subsystem PID is 2068.
sp3en0/ # lssrc -a | grep key
spnkeyman 2068 active
sp3en0/ # ps -ef | grep key | grep -v grep
   root 2068 4170 0 14:30:29 - 0:02 /usr/lpp/ssp/bin/
      spnkeymand
sp3en0/ #
```

18. Finish the configuration of the CWS and the installation of the nodes following the PSSP 3.2: Installation and Migration Guide, GA22-7347.

After the successful NIM installation of the nodes, the `psspfb_script` runs the `spauthconfig` command on each node. This script installs and configures all needed DCE filesets automatically. Then, you can see the LED u95 during installation on the 3-digit display.

4.7 Example 4: Advanced DCE cell

In this scenario, we go through the security-related installation steps of the PSSP Installation and Migration Guide, GA22-7347, for a new installation. We
install the SP system with the DCE security state. For more information on this security state, see Chapter 2, “SP security concepts and terminology” on page 11. As described there, this security state uses Kerberos V5 as the method for authentication and authorization for remote commands and SP Trusted Services. The security state for this scenario is shown in Table 10.

Table 10. SP security state of Example 4 (DCE Security - full DCE cell)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Security State: DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auth_install</td>
</tr>
<tr>
<td>advanced DCE cell</td>
<td>dce</td>
</tr>
</tbody>
</table>

We will configure the CWS as DCE Security Master Server and Primary CDS Server. To improve high availability, we will add a DCE Security Replica Server and a Secondary CDS Server on an SP node. For installation and configuration tasks, we will add an SP administrator principal to the DCE cell.

High-level outline for example 4

1. Planning considerations
2. Adding the SP administrator DCE principal to the DCE cell
3. Installation steps
4. Setting up a Secondary CDS server on a node
5. Setting up a DCE Security Replica server on a node

4.7.1 Example 4: Planning considerations

Refer to “Example 3: Planning considerations” on page 72 for the basic planning issues that need to be considered when installing an SP system to use the DCE security state.

In this example, we add a special SP administrator DCE principal, called sp_admin, to the DCE cell for installation and administration tasks on the SP system.

Recall from Section 3.3, “The concept of a DCE cell” on page 25, that DCE servers, especially Security servers, should be configured to not allow end-user access, and they should be secured with stringent physical access controls. If the security registry is compromised, your entire DCE cell is compromised. For this reason, it is not recommended to install any DCE server on a node or the CWS. Refer to Section 4.8, “Example 5: Existing DCE cell” on page 112, for an appropriate configuration.
Nevertheless, we will install a DCE Security Replica server and a Secondary CDS server on an SP node for high availability and performance reasons. You are free to choose a system for the DCE Security Master server and the Primary CDS server. If you decide to use the CWS, go to Section 4.6.3, “Setting up the CWS as DCE server” on page 74, for information about how to set up the CWS as a DCE server. If you like to use a system that is external to the SP, see Section 4.8.2, “Setting up a DCE client on the CWS” on page 114.

Also see Section 4.6.2, “Dealing with mksysb images in a DCE environment” on page 73, to learn how to work with the mksysb images of your SP nodes.

4.7.2 Adding the SP administrator DCE principal to the DCE cell

The following installation steps require DCE cell administrator credentials.

1. As cell administrator on any DCE client node, issue `smitty dce` to get to the DCE main SMIT panel, and perform the sequence of SMIT menu options shown in Figure 8 on page 93.

```
sp3en0/ # smitty dce
-> DCE Security & Users Administration
   -> Principals and Aliases
      -> Add a Principal

Add a Principal

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
* Principal NAME [sp_admin]
  Principal ID []
  FULL name [SP administrator]
  Registry object QUOTA []

Figure 8. SP administrator principal configuration for example 4
```

On the “Local DCE/DFS Client Configuration” SMIT panel, use the following conventions:

a. Insert `sp_admin` as “Principal Name”.

b. Insert a description for this principal in the “FULL name” entry field, for example, `SP administrator`.

2. To add an account to this principal, issue `smitty dce` to get to the DCE main SMIT panel, and perform the sequence of SMIT menu options shown in Figure 9 on page 94.
Figure 9. SP administrator account configuration for example 4

On the “Add an Account” SMIT panel, use the following conventions:

- Insert `sp_admin` as the “Principal Name”.
- Select Yes in the “LOGIN user?” entry field.
- Select none in the “GROUP to associate with this account” entry field.
- Select none in the “ORGANIZATION to associate with this account” entry field.

3. You can verify the process using the `dce_login` and `klist` commands:

```
sp3en0/ # exit
sp3en0/ # dce_login sp_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep Global
global Principal: /.../sp_cell/sp_admin
```
4.7.3 Example 4: Installation steps

Go through the installation steps provided in Chapter 2, “Installing and Configuring a New RS/6000 SP System”, of the *PSSP Installation and Migration Guide*, GA22-7347.

1. Proceed with the installation steps as described in the *PSSP Installation and Migration Guide*, GA22-7347, up to “Step 18: Set Authentication Methods for AIX Remote Commands on the Control Workstation”. Make sure that you have set up your DCE server as described in Section 4.6.3, “Setting up the CWS as DCE server” on page 74.

   Select k5 and std as the authentication methods for remote commands, and use the `chauthent` command. You can verify your settings with the `lsauthent` command:

   ```
   sp3en0/ # chauthent -k5 -std
   sp3en0/ # lsauthent
   Kerberos 5
   Standard Aix
   ```

   **Important**

   Issuing the `chauthent` command without any parameters turns off all authentication methods for this machine, effectively disabling remote logins for everyone.

2. Skip “Step 19: Set Authentication Methods for AIX Remote Commands on the Control Workstation”, and go to “Step 20: Initialize RS/6000 SP Kerberos V4”. You need to have a DCE server installed and configured on your system before you can go on. If you are not sure about this, refer to the verification section, Section 4.6.3, “Setting up the CWS as DCE server” on page 74.

   **Important**

   The following installation steps require DCE cell administrator credentials.
If the DCE server is up and running, log in as the `cell_admin` principal. Since it is normally the first time you do this, you are prompted to change your password. You can reenter the initial password you have chosen when you set up the server.

```
sp3en0/ # dce_login cell_admin
Enter Password:
Password must be changed!
DCE LOGIN SUCCESSFUL
Warning: This account has been marked by an administrator, recommending that the password be changed.
Do you wish to change now [y/n]? (y)
Enter New Password:
Re-enter New Password:
PASSWORD SUCCESSFULLY CHANGED
```

Going through “Step 20.2: Create DCE Groups, Organizations, Principals and Accounts”, you have to use the `config_spsec` command and enter `cell_admin` as the cell administrator ID. When prompted, enter your password. This must be done using the `cell_admin` principal.

```
sp3en0/ # config_spsec -c -v
This command requires cell administrator authority. Continue? (y/n) y
Running "check_prereqs" subroutine ... 
Checking state of DCE ... 
Running "open_io" subroutine ...
Running "parse_defaults" subroutine ...
Parsing spsec_defaults file ...
Running "parse_overrides" subroutine ...
Running "create_default" subroutine ...
Please enter cell administrator id to be added to ACL admin group: \cell_admin ...
```

3. In “Step 20.3: Create SP Administrative Principals”, you must add an administrative principal to several DCE access groups. We recommend that you use the `sp_admin` principal, which you added in Section 4.7.2, “Adding the SP administrator DCE principal to the DCE cell” on page 93, on the CWS as the administrative principal for the rest of the installation.
Later, in the installation process and for administrative tasks, it might be necessary to add membership to more access groups, for example, to be authorized to issue switch commands.

4. Because the `create_keyfiles` command must be run with the self-host principal in "Step 20.4: Create Control Workstation Specific Keyfiles", log out of your `cell_admin` principal, and issue the `create_keyfiles` command. You can verify your current DCE identity using the `klist` command.

5. Go on to "Step 21: Configure DCE for the Control Workstation (Required for DCE)", and select *Kerberos Version 4* as the authentication method for SP Trusted Services. Use the `chauthts` command, and verify the settings with the `lsauthts` command.

---

Important

The following installation steps require DCE self-host credentials.

---

Important

The following installation steps require SP administrator (`sp_admin`) credentials.
6. Because you have changed the DCE credentials of the SP administrative principal, sp_admin, prior to “Step 22: Set the Authentication Method for SP Trusted Services on the Control Workstation”, you must log in as sp_admin to get the new credentials. You can verify that all access groups are added to your principal with the klist command.

```
sp3en0/ # dce login sp_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep -p Groups
DCE Identity Information:  
Warning: Identity information is not certified  
Global Principal: /.../sp_cell/sp_admin  
Cell: 4b6608a0-9842-11d3-a5a0-02608c2d4a7f /.../sp_cell  
Principal: 0000007f-984e-21d3-9200-02608c2d4a7f sp_admin  
Group: 0000000c-9842-21d3-a501-02608c2d4a7f none  
Local Groups:  
0000000c-9842-21d3-a501-02608c2d4a7f none  
00000086-9844-21d3-9201-02608c2d4a7f sdr-admin  
00000080-9844-21d3-9201-02608c2d4a7f hm-admin  
00000088-9844-21d3-9201-02608c2d4a7f sdr-system-class-admin  
0000007c-9844-21d3-9201-02608c2d4a7f spsec-admin  
00000081-9844-21d3-9201-02608c2d4a7f hm-control  
```

7. Continue with the normal installation steps to “Step 39: Select Security Capabilities Required on Nodes”.

Issue the smitty spauth_config command to get to the “RS/6000 SP Security” SMIT panel, and select Select Security Capabilities Required on Nodes. Insert your system partition name, and specify dce as the authentication method.

```
sp3en0/ # smitty spauth_config
-> Select Security Capabilities Required on Nodes
Select Security Capabilities Required on Nodes

Type or select values in entry fields.  
Press Enter AFTER making all desired changes.  

[Entry Fields]  
* System Partition Name  
* Authentication Methods

You can also use the following command:

spsetauth -i -p <syspar> dce
```

8. In “Step 40: Create DCE Hostnames (Required for DCE)”, the reliable hostname of the nodes are entered as DCE hostname in the SDR. Select Create DCE hostnames from the “RS/6000 SP Security” SMIT panel, or use the create_dcehostname command:
You can verify the creation of the DCE hostnames in the SDR by issuing the `splstdata -n` command.

```bash
sp3en0/ # splstdata -n
```

List Node Configuration Information
```
node# frame# slot# slots initial_hostname reliable_hostname dce_hostname
default_route processor_type processors_installed description
------- ------- ------- ------- ----------------- ----------------- -------- ----------
          ------- ------- ------- ------- ------- ------- ------- ------- -------
         ------- ------- ------- ------- ------- ------- ------- ------- -------
1         1     1         4      sp3n01    sp3n01    sp3n01
          192.168.3.130  MP 1
```

To insert the DCE and CDS server hostnames in the SDR, go to “Step 41: Update the SDR with DCE Master Security and CDS Server Hostnames (Required for DCE)”, and select Update SDR with DCE Master Security and CDS Server Hostnames from the “RS/6000 SP Security” SMIT panel. Insert the hostname of the DCE Master Security server and the Primary CDS server in the appropriate entry fields.

```bash
sp3en0/ # smitty spauth_config
```

- > Update SDR with DCE Master Security and CDS Server Hostnames

Update SDR with DCE Master Security and CDS Server Hostnames

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

<table>
<thead>
<tr>
<th>[Entry Fields]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Security Server hostname</td>
</tr>
<tr>
<td>CDS Server hostname</td>
</tr>
</tbody>
</table>

Or use the following command:
You can verify the settings in the SDR with the `splstdata -e` command:

```
sp3en0/ # splstdata -e
...
sec_master           sp3en0
creds_server         sp3en0
cell_name             /.../sp_cell
...
```

10. In "Step 42: Configure Admin Portion of DCE Clients", you create the necessary DCE security and CDS entries for the SP nodes. Select **Configure DCE Clients (Admin portion)** from the “RS/6000 SP Security” SMIT panel, and enter `cell_admin` as the cell administrator ID and `././lan-profile` as the profile ID.

```
sp3en0/ # smitty spauth_config
-> Configure DCE Clients (Admin portion)

Configure DCE Clients (Admin portion)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Cell Administrator id          [Entry Fields]
[cell_admin]
Lan Profile id                 [././lan-profile]
```

Instead of using SMIT, you can use the following command:

```
setupdce -c cell_admin -l ././lan-profile
```

**Important**

The next installation step requires the DCE cell administrator password.

When you are prompted, enter the password for the `cell_admin` principal. After the configuration of all your DCE hosts has ended successfully, you can verify the creation of the entries with the `dcecp -c cell show` command:
This command provides a view of all defined hosts in a DCE cell. You should now see all your SP nodes including the CWS in the output.

11. Go to “Step 43: Select Authorization Methods for AIX Remote Commands”, and issue the `smitty spauth_config` command to get to the SMIT security menu. Select Select Authorization Methods for AIX Remote Commands. In the next panel, insert your system partition name and specify `dce` and `std` as the authorization methods.

Or use the following command:

```
spsetauth -d -p <syspar> dce std
```

**Note**

You need to use the two methods, `dce` and `std`, at this point because you must have at least one common method defined for the system partition and the remote commands. Since we have not yet changed the authentication method for the partition, it is still set to `std`. Later, we will change these methods to only use `dce`.
12. In “Step 44: Configure SP Trusted Services to Use DCE Authentication (Required for DCE)”, select **Configure SP Trusted Services to use DCE Authentication** from the “RS/6000 SP Security” SMIT panel and enter y when prompted. Then, type in your administrator ID and the password when required.

```
sp3en0/ # dce_login cell_admin
Enter Password: 
DCE LOGIN SUCCESSFUL
sp3en0/ # smitty spauth_config
-> Configure SP Trusted Services to use DCE Authentication

This command requires cell administrator authority. Continue? (y/n) y

Running "check_prereqs" subroutine ...
Checking state of DCE ...
Running "open_io" subroutine ...
Running "parse_defaults" subroutine ...
Parsing spsec_defaults file ...
Running "parse_overrides" subroutine ...
Running "create_default" subroutine ...

Please enter cell administrator id to be added to ACL admin group: \cell_admin

Adding cell administrator id to group spsec-admin ...
Creating org - spsec-services ...
Creating group - spsec-services ...
Running "create_accts_on_cws" subroutine ...

Your cell administrator password is required to create accounts. Please enter your cell administrator password: ...
```

Instead of using SMIT, you can use the following command:

```
config_spsec -v
```

13. To continue with “Step 45: Create SP Services DCE Keyfiles (Required for DCE)”, you must exit your cell administrator principal. This step has to be done as the self-host principal.

---

**Important**

The following installation step requires DCE self-host credentials.

---

102 Exploiting RS/6000 SP Security: Keeping It Safe
When you run “Create SP Services Keyfiles” from the “RS/6000 SP Security” SMIT panel, you should get an output, such as the one shown in the next screen:

```
sp3en0/ # exit
sp3en0/ # klist | grep Global
    Global Principal: /.../sp_cell/hosts/sp3en0/self
```

You can also use the following command:

```
create_keyfiles -v
```

**Important**

The remaining installation steps require SP administrator (sp_admin) credentials.

14. In “Step 46: Enable Authentication Methods for AIX Remote Commands”, log in as sp_admin again to have the right credentials to change attributes of the SDR. Then, issue `smitty spauth_config`, and select **Enable Authentication Methods for AIX Remote Commands** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition, and specify **k5** and **std** as the authentication methods.
Instead of using SMIT, you can use the following command:

```
chauthpar -c -p <syspar> k5 std
```

15. In “**Step 47: Enable Authentication Methods for SP Trusted Services**”, you do this by selecting **Enable Authentication Methods for SP Trusted Services** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition and specify **dce** as the authentication method.

You can also use the following command:

```
chauthpts -c -p <syspar> dce
```

This can take several minutes.

16. You can now change the authorization methods for the AIX remote commands to only use **dce** (remember step 11 of this procedure). Issue
the `smitty spauth_config` command to get to the SMIT security menu, and select **Select Authorization Methods for AIX Remote Commands**. In the next panel, insert your system partition name, and specify `dce` as the authorization method.

```
sp3en0/ # smitty spauth_config
-> Select Authorization Methods for AIX Remote Commands

Select Authorization Methods for AIX Remote Commands

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

* System Partition name
  sp3en0
+  
* Authorization Methods
  dce
+  
```

You can also use the following command:

```
spsetauth -d -p <syspar> dce
```

Verify your SDR settings with the `splstdata -p` command.

```
sp3en0/ # splstdata -p

List System Partition Information

System Partitions:
-----------------
sp3en0

Syspar: sp3en0
-----------------------------------------------
syspar_name sp3en0
ip_address 192.168.3.130
install_image default
syspar_dir 
code_version PSSP-3.2
haem_cdb_version 940617228,332373504,0
auth_install dce
auth_root_rcmd dce
ts_auth_methods dce
auth_methods k5:std

```

17. In **“Step 48: Start the Key Management Daemon”**, use the `spnkeyman_start` command to start the key management daemon on the CWS. You can verify that the daemon is running with the `lssrc` or `ps` command.
18. Finish the configuration of the CWS and the installation of the nodes following the *PSSP Installation and Migration Guide*, GA22-7347.

After the successful NIM installation of the nodes, the `psspfb_script` runs the `spauthconfig` command on each node. This script installs and configures all needed DCE filesets automatically. Then, you can see the LED u95 during installation on the 3-digit display.

### 4.7.4 Setting up a secondary CDS server on a node

Perform the following steps to set up a secondary CDS server on a node:

1. As root, on the Secondary CDS Server node, issue `smitty dce` to get to the DCE main SMIT panel, and perform the following sequence of SMIT menu options:

```
sp3en0/ # smitty dce
  -> Configure DCE/DFS
    -> Configure DCE/DFS Servers
      -> CDS (Cell Directory Service) Server
        -> 2 additional

CDS (Cell Directory Service) Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* CELL name [sp_cell]
* Cell ADMINISTRATOR's account [cell_admin]
 Machine's DCE HOSTNAME [sp3n05]
* Start components at System restart Yes +
* Clean up DCE at System restart Yes +
* Protocol tcp udp +
 MASTER SECURITY Server [sp3en0]
 Minutes to wait between pe_site file updates. [1440] #
 Other CDS Server (If in a separate network) [sp3en0]
 List of additional CDS Servers []
 Synchronize Clocks No +
 Time Server to Synchronize Clocks with []
* LAN PROFILE [lan-profile]
```

[Figure 10. Secondary CDS Server configuration for example 4]
Make sure that all entry fields match your configuration (especially the “Machine’s DCE HOSTNAME” and the “MASTER SECURITY Server” fields) before you start the configuration.

When prompted, enter the password for your cell administrator, cell_admin. You should get the following output:

```
2. Verify the configuration with the dcecp -c cell show command:

sp3n05/ # dcecp -c cell show
{secservers
  /.../sp_cell/subsys/dce/sec/sp3en0}
{cdsservers
  /.../sp_cell/hosts/sp3en0
  /.../sp_cell/hosts/sp3n05}
{dtsservers}
{hosts
  /.../sp_cell/hosts/sp3en0
  /.../sp_cell/hosts/sp3n01
  /.../sp_cell/hosts/sp3n05
...}
```

To make sure that all necessary daemons are running, you can use the `ps` command. Here, you should see the `cdsd` daemon running on the node:

```
sp3n05/ # ps -ef | grep dce | grep -v grep
  root 5617  1  0 Oct 29 - 1:09 /opt/dcelocal/bin/dced -b \
-t 1440
  root 19196 21152  0 14:02:55 - 0:08 /opt/dcelocal/bin/cdsclerk \
-w FATAL:STDOUT:-;FILE:/opt/dcelocal/var/svc/fatal.log -w ERROR:STDOUT\n  \-FILE:/opt/dcelocal/var/svc/error.log -w WARNING:STDOUT:-;FILE:\n  /opt/dcelocal/var/svc/warning.log -w NOTICE:DISCARD: -w NOTICE_VERBOSE:DISCARD:
  root 20544  1  0 14:31:16 - 0:01 /opt/dcelocal/bin/cdsd
  root 21152  1  0 Oct 29 - 0:03 /opt/dcelocal/bin/cdsadv
```

3. Unfortunately, this process only replicates the `./subsys/dce/sec` directory of the DCE namespace and not the whole namespace into the new Secondary CDS Server. Nevertheless, since this directory contains the
binding information to locate the master security server, this replication increases accessibility to the security server even when the initial CDS Server is unavailable.

To replicate the whole DCE namespace into the Secondary CDS Server you can use the following script:
Chapter 4. Implementing a specific security state

Figure 11. Script to replicate the CDS namespace (cds-replicate.ksh)
4.7.5 Setting up a DCE Security Replica Server on a node

Perform the following steps to set up a DCE Security Replica Server on a node:

1. As root, on the DCE Security Replica Server node, issue `smitty dce` to get to the DCE main SMIT panel, and perform the sequence of SMIT menu options shown in the following screen:

```
sp3en0/ # smitty dce
-> Configure DCE/DFS
   -> Configure DCE/DFS Servers
      -> Security Server
         -> 2 secondary

SECURITY Server
Type or select values in entry fields. Press Enter AFTER making all desired changes.

[Entry Fields]
[TOP]
* CELL name [sp_cell]
* Cell ADMINISTRATOR's account [cell_admin]
Machine's DCE HOSTNAME [sp3en05]
* Start components at System restart Yes +
* Clean up DCE at System restart Yes +
* Protocol tcp udp +
MASTER SECURITY Server [sp3en0]
Minutes to wait between pe_site file updates. [1440] #
CDS Server (If in a separate network) [sp3en0]
List of additional CDS Servers []
Synchronize Clocks No +
Time Server to Synchronize Clocks with []
* LAN PROFILE [lan-profile]
Rebroadcast CDS Server Location No +
Security Server Name []
* Use CERTIFICATe based login? No +
ENTRUST PROFILE for the Security server []
ENTRUST INITIALIZATION file []
ENTRUST PROFILE Password []
```

Figure 12. Secondary DCE Security Server configuration for example 4

Make sure that all entry fields match your configuration, especially the "Machine’s DCE HOSTNAME" and the “MASTER SECURITY Server” fields before you start the configuration.

When prompted, enter the password for your cell administrator, cell_admin. You should get the output shown in the following screen:
Chapter 4. Implementing a specific security state

2. Verify the configuration with the `dcecp -c cell show` command:

```
sp3n05/ # dcecp -c cell show
{secservers
 /.../sp_cell/subsys/dce/sec/sp3en0
 /.../sp_cell/subsys/dce/sec/sp3n05}
{cdsservers
 /.../sp_cell/hosts/sp3en0
 /.../sp_cell/hosts/sp3n05}
{dtsservers}
{hosts
 /.../sp_cell/hosts/sp3en0
 /.../sp_cell/hosts/sp3n01
 /.../sp_cell/hosts/sp3n05
...
```

To make sure that all necessary daemons are running, you can use the `ps` command. Here, you should see the `secd` daemon running on the node.

```
sp3n05/ # ps -ef | grep dce | grep -v grep
    root 4914  1  0 Oct 29 - 1:09 /opt/dcelocal/bin/dced -b -t 1440
    root 19196 21152  0 14:02:55 - 0:08 /opt/dcelocal/bin/cdsclerk \
    root 20544  1  0 14:31:16 - 0:01 /opt/dcelocal/bin/cdsd
    root 21152  1  0  Oct 29 - 0:03 /opt/dcelocal/bin/cdsadv
    root 21788  1  0 14:45:58 - 0:02 /opt/dcelocal/bin/secd
```

3. To populate the information about the new DCE Security Replica Server to your nodes, you can use the `chpesite` command. This command must run on all DCE clients in your cell to manipulate the `/opt/dcelocal/etc/security/pe_site` file. Do not forget to also run this command on the CWS.
4. To verify the population process, you should see an entry for your new DCE Security Replica Server in each /opt/dcelocal/etc/security/pe_site file.

```bash
sp3en0/ # /usr/bin/chpesite
sp3en0/ # dsh -av /usr/bin/chpesite
...
```

The pe_site file is parsed from top to bottom; so, you may want to rearrange the order of the entries in the file for performance or load balancing reasons.

4.8 Example 5: Existing DCE cell

In this scenario, we go through the security-related installation steps of the PSSP Installation and Migration Guide, GA22-7347, to install the SP system with the DCE security state. This security state uses Kerberos V5 as the method for authentication and authorization for remote commands and SP Trusted Services. The security state for this scenario is shown in Table 11.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Security State: DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_install</td>
<td>dce</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>dce</td>
</tr>
<tr>
<td>ts_auth_methods</td>
<td>dce</td>
</tr>
<tr>
<td>auth_methods</td>
<td>k5:std</td>
</tr>
</tbody>
</table>

We will use the DCE Security Server and CDS Server of a node external to the SP. The DCE Security Master server and the Primary CDS server for this cell, called itso_cell, are located on a system with the hostname, arthur. The CWS and all SP nodes will be added as DCE clients to this existing DCE cell.
In our lab environment, we need to configure the CWS as gateway because the nodes do not have a direct connection to the external network.

**High-level outline for example 5**
1. Planning considerations
2. Setting up a DCE Client on the CWS
3. Installation steps

4.8.1 Example 5: Planning considerations

Like most distributed systems, DCE is highly-dependent on the integrity of the underlying network. Before configuring any DCE component, verify that the following are properly configured and functioning correctly:

- Host name settings (hostname)
- Forward and reverse name resolution (nslookup, host, /etc/resolv.conf, /etc/netsvc.conf, and /etc/hosts)
- Network routings (netstat -rn, ping, and traceroute)

Also, verify that your date, time, and timezone settings are correct. DCE is extremely sensitive to time discrepancies.

DCE processes write their data (including core files) into /var/dce. These core files can be quite large. For this reason, we recommend creating a /var/dce filesystem at least 200 MB in size, especially on machines that are DCE servers. However, if you do not have the disk space or are not concerned about preserving DCE core files, you can go with the recommended minimum size of 32 MB.

During the node installation process, PSSP installs and configures a DCE client on each node. Unfortunately, this process does not contain the creation of a separate /var/dce filesystem on the nodes. To prevent the use of the root filesystem by the DCE client, you can add the following line in your script.cust file:

```bash
crfs -v jfs -g'rootvg' -a size='64000' -m /var/dce -A'yes' -p'rw' -t'no'
```

In the following step-by-step-procedure, you will be asked to supply the following:

- The IP address and hostname of your DCE and CDS server
- The name of your DCE cell administrator (typically, cell_admin)
For this example, we assume that the SP administrator with root authority will not be the DCE cell administrator. You will need to have somebody with cell administrator authority available at several steps during the install process.

We will use the self-host principal as an SP administrator with authorization for SP Trusted Services and access to the SDR for ease-of-use. As explained in the PSSP Administration Guide, SA22-7348, you can decide to use other DCE principals that do not need root authorization for SP configuration or monitoring tasks if you want to establish a higher granularity of object protection on your SP system. Refer to Section 4.7.2, “Adding the SP administrator DCE principal to the DCE cell” on page 93, and the PSSP Administration Guide, SA22-7348, for information about how to set up the authorizations.

For performance reasons, it can be very useful to install a DCE Security Replica Server and a Secondary CDS Server on one of the SP nodes. Refer to Section 4.7.4, “Setting up a secondary CDS server on a node” on page 106, and Section 4.7.5, “Setting up a DCE Security Replica Server on a node” on page 110, for a detailed step-by-step procedure for this. On the other hand, all DCE servers, especially DCE Security servers, should be configured to not allow end-user access, and they should be secured with stringent physical access controls. If the security registry is compromised, your entire DCE cell is compromised. For this reason, it is not recommended that you install any DCE server on a node or the CWS.

See also Section 4.6.2, “Dealing with mksysb images in a DCE environment” on page 73, for information about how to work with the mksysb images of your SP nodes.

### 4.8.2 Setting up a DCE client on the CWS

As described in Section 3.7, “Split configuration for DCE clients” on page 47, you can split a DCE client configuration into two parts: The admin-only part, which is done by the DCE cell administrator on the DCE Security Server, and the local-only part, which must be done by root on the client node. In the following paragraphs, we will give you a step-by-step procedure for both parts. Nevertheless, it is easier to do a full client configuration on the CWS if you have root access to the CWS and cell administrator access in the DCE cell.

**Preparing the CWS for DCE client configuration**

1. Before installing the DCE filesets, create the DCE-related directories, filesystems, and mount points. We recommend that you use separate file
systems for the DCE configuration files. A good starting point for the server installation on the CWS can be using the following parameters:

c. Create filesystem: crfs -v jfs -g 'rootvg' -a size='64000' -m /var/dce -A 'yes' -p 'rw' -t 'no'
d. Mount filesystem: mount -v jfs /var/dce

2. Install the DCE filesets on the CWS using the `installp` command or SMIT. Follow normal software installation steps to install DCE filesets. Refer to Table 12 for more information about which filesets must be installed.

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce.client</td>
<td></td>
</tr>
<tr>
<td>dce.client.rte.admin</td>
<td>DCE Client Administrative Tools</td>
</tr>
<tr>
<td>dce.client.rte.cds</td>
<td>DCE Client CDS Tools</td>
</tr>
<tr>
<td>dce.client.rte.config</td>
<td>DCE Client Configuration Tools</td>
</tr>
<tr>
<td>dce.client.rte.rpc</td>
<td>DCE Client RPC Tools</td>
</tr>
<tr>
<td>dce.client.rte.security</td>
<td>DCE Client Security Tools</td>
</tr>
<tr>
<td>dce.client.rte</td>
<td>DCE Client Services</td>
</tr>
<tr>
<td>dce.client.core.rte</td>
<td>DCE Client Services - FOR UPDATES</td>
</tr>
<tr>
<td>dce.client.rte.time</td>
<td>DCE Client Time Tools</td>
</tr>
<tr>
<td>dce.client.rte.zones</td>
<td>DCE Client Time Zones</td>
</tr>
<tr>
<td>dce.client.smit</td>
<td>DCE SMIT Client Tools</td>
</tr>
<tr>
<td>dce.client.rte.pthreads</td>
<td>DCE Threads Compatibility Library</td>
</tr>
<tr>
<td>dce.client.rte.web</td>
<td>DCE Web Secure</td>
</tr>
<tr>
<td>dce.xdssom</td>
<td></td>
</tr>
<tr>
<td>dce.xdssom.rte</td>
<td>X.500 API Library</td>
</tr>
</tbody>
</table>

3. Before you begin with the client configuration, make sure that your name resolution and routing information is set up correctly on the client and server (see Section 4.8.1, “Example 5: Planning considerations” on page 113).

**Configuring the admin-only part of the DCE client**
1. On your DCE Security Server, log in as cell administrator and issue `smitty dce` to get to the DCE main SMIT panel. Perform the following sequence of SMIT menu options:

```
splen0/ # smitty dce
-> Configure DCE/DFS
  -> Configure DCE/DFS Clients
    -> 3 admin only configuration for another machine

Administrator DCE Client Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
* CLIENTS to configure
  sec_cl cds_cl +
* Cell ADMINISTRATOR's account
  [cell_admin]
* Machine's DCE HOSTNAME
  [sp3en0]
* Machine's name or TCP/IP address
  [sp3cw0]
* LAN PROFILE
  [lan-profile]
```

Example 5

**Important**
The admin-only part of the DCE client configuration requires DCE cell administrator credentials.

Perform the following steps from the “Administrator DCE Client Configuration” SMIT panel:

a. Select `sec_cl` and `cds_cl` as from the list of DCE clients in the entry field for “CLIENTS to configure”

b. If you do not use the default `cell_admin` as cell administrator insert the name of your cell administrator’s account.

c. Enter the DCE hostname of the CWS. This should normally be the name of the en0 IP-label or the hostname of the CWS.

d. Enter the hostname or the equivalent IP-address of the CWS.

e. If you have use an other lan profile than the default `lan-profile` for the clients in your cell, enter the profile the CWS should use in the appropriate entry field.

This will update the namespace entries and security registry database of the DCE cell.

2. You can verify the settings with the `dcecp` and `cdsli` commands.
Chapter 4. Implementing a specific security state

1. As root, on the client node (CWS), issue `smitty dce` to get to the DCE main SMIT panel, and perform the following sequence of SMIT menu options:

   `cdsli -cworld
   ...
   d  /.:/hosts/sp3en0
   o b  /.:/hosts/sp3en0/cds-clerk
   o p  /.:/hosts/sp3en0/profile
   o b  /.:/hosts/sp3en0/self
   ...

   [Entry Fields]
   * CLIENTS to configure
   * CELL name
   * Machine’s DCE HOSTNAME
   * Start components at System restart
   * Clean up DCE at System restart
   * Protocol
   * MASTER SECURITY Server
   * Minutes to wait between pe_site file updates
   * CDS Server (If in a separate network)
   * List of additional CDS Servers
   * Synchronize Clocks
   * Time Server to Synchronize Clocks with
   * Rebroadcast CDS Server Location

   Figure 14. DCE client configuration (local only) for example 5

   Perform the following steps in the “Local DCE/DFS Client Configuration” SMIT panel:
a. Select `rpc`, `sec_cl`, and `cds_cl` as from the list of DCE clients in the entry field for “CLIENTS to configure”

b. Insert the (fully-qualified) cell name of your DCE cell.

c. Insert the DCE hostname of the CWS that was chosen in the admin-only part of the DCE client configuration.

d. Select `yes` for “Start components at System restart”.

e. Select `yes` for “Clean up DCE at System restart”.

f. Insert the hostname of the Master Security Server and the hostname of the CDS Server.

2. To verify if your configuration succeeded, you can use the `klist` command.

```
sp3en0/ # klist | grep Global
Global Principal: /.../itso_cell.itso.ibm.com/hosts/sp3en0/self
```

### 4.8.3 Example 5: Installation steps

Go through the installation steps provided in Chapter 2, “Installing and Configuring a New RS/6000 SP System”, of the *PSSP 3.2: Installation and Migration Guide*, GA22-7347.

1. Proceed with the installation steps as described in the Installation and Migration Guide to **Step 18: Set Authentication Methods for AIX Remote Commands on the Control Workstation**. Make sure that you have setup your DCE client as described in the previous section “Setting up a DCE client on the CWS” on page 114.

Select `k5` and `std` as authentication method for remote commands and use the `chauthent` command. You can verify your settings with the `lsauthent` command:

```
sp3en0/ # chauthent -k5 -std
sp3en0/ # lsauthent
Kerberos 5
Standard Aix
```

**Important**

Issuing the `chauthent` command without any parameters turns off all authentication methods for this machine, effectively disabling remote logins for everyone.
2. Skip “Step 19: Set Authentication Methods for AIX Remote Commands on the Control Workstation”. To go on with “Step 20: Initialize RS/6000 SP Kerberos V4”, you need to have a DCE client installed and configured on the CWS before you can go on. Refer to the verification section, Section 4.8.2, “Setting up a DCE client on the CWS” on page 114.

   Important

   The following installation step requires **DCE cell administrator credentials**.

   Going through “Step 20.2: Create DCE Groups, Organizations, Principals and Accounts”, you have to use the `config_spsec` command and enter `cell_admin` as the cell administrator ID. When prompted, enter your password. This must be done using the `cell_admin` principal.

```
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # config_spsec -c -v

This command requires cell administrator authority. Continue? (y/n) y

Running "check_prereqs" subroutine ...
Checking state of DCE ...
Running "open_io" subroutine ...
Running "parse_defaults" subroutine ...
Parsing spsec_defaults file ...
Running "parseOverrides" subroutine ...
Running "create_default" subroutine ...

Please enter cell administrator id to be added to ACL admin group: \cell_admin

...```

3. In “Step 20.3: Create SP Administrative Principals”, you must add an administrative principal to several DCE access groups. We recommend that you use the self-host principal as the administrative principal for the rest of the installation in this example.

   ```
   sp3en0/ # dce_login cell_admin
   Enter Password:
   DCE LOGIN SUCCESSFUL
   sp3en0/ # config_spsec -c -v

   This command requires cell administrator authority. Continue? (y/n) y

   Running "check_prereqs" subroutine ...
   Checking state of DCE ...
   Running "open_io" subroutine ...
   Running "parse_defaults" subroutine ...
   Parsing spsec_defaults file ...
   Running "parseOverrides" subroutine ...
   Running "create_default" subroutine ...

   Please enter cell administrator id to be added to ACL admin group: \cell_admin
   ...
```
Later in the installation process, it can be necessary to add additional membership to more access groups, for example, to be authorized to issue switch commands.

**Important**
The following installation step requires DCE self-host credentials.

4. In "Step 20.4: Create Control Workstation Specific Keyfiles", you must run the command with the self-host principal. Log out of your cell_admin principal, and issue the `create_keyfiles -c -v` command. You can verify your current DCE identity using the `klist` command.

```
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # dcecp -c group add sdr-admin -member ./hosts/sp3en0/self
sp3en0/ # dcecp -c group add hm-admin -member ./hosts/sp3en0/self
sp3en0/ # dcecp -c group add spsec-admin -member ./hosts/sp3en0/self
sp3en0/ # dcecp -c group add sdr-system-class-admin -member ./hosts/sp3en0/self
sp3en0/ # dcecp -c group add hm-control -member ./hosts/sp3en0/self
```

The following installation steps require SP administrator (self-host) credentials.

5. Go on to “Step 21: Configure DCE for the Control Workstation (Required for DCE)”, and select Kerberos Version 4 as the authentication method for SP Trusted Services. Use the `chauthts` command, and verify the settings with the `lsauthts` command.

```
sp3en0/ # exit
sp3en0/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # create_keyfiles -c -v
...
sp3en0/ # chauthts dce
sp3en0/ # lsauthts
DCE
```

---

120 Exploiting RS/6000 SP Security: Keeping It Safe
6. Because you have changed the DCE credentials of the SP administrative principal self-host prior to "Step 22: Set the Authentication Method for SP Trusted Services on the Control Workstation", you have to stop your DCE client and start it again to get the new credentials. You can verify that all access groups are added to your principal with the `klist` command.

7. Continue with the normal installation steps to "Step 39: Select Security Capabilities Required on Nodes".

Issue the `smitty spauth_config` command to get to the “RS/6000 SP Security” SMIT panel, and select Select Security Capabilities Required on Nodes. Insert your system partition name, and specify `dce` as the authentication method.

8. In “Step 40: Create DCE Hostnames (Required for DCE)”, the reliable hostname of the nodes are entered as DCE hostname in the SDR. Select
Create DCE hostnames from the “RS/6000 SP Security” SMIT panel, or use the `create_dcehostname` command.

```
sp3en0/ # smitty spauth_config
  --> Create DCE hostnames

  create_dcehostname: Checking state of dce ...
  create_dcehostname: Reading SDR data...

  Obtaining DCE host entries...

  Setting DCE hostnames in the SDR...
```

You can verify the creation of the DCE hostnames in the SDR by issuing the `splstdata -n` command.

```
sp3en0/ # splstdata -n
   List Node Configuration Information

   node#  frame#  slot#  slots  initial_hostname  reliable_hostname  dce_hostname
       ---   ------   -----   -----                  -----------------   -----------------
   description
   -----   ------   -----   -----------------   -----------------   -----------------

   1   1   1   4   sp3n01   sp3n01   sp3n01  192.168.3.130   MP
   ***
   5   1   5   1   sp3n05   sp3n05   sp3n05  192.168.3.130   UP
   ***

   ...
```

9. To insert the DCE and CDS server hostnames in the SDR, continue with “Step 41: Update the SDR with DCE Master Security and CDS Server Hostnames (Required for DCE)”, and select **Update SDR with DCE Master Security and CDS Server Hostnames** from the “RS/6000 SP Security” SMIT panel. Insert the hostname of the DCE Master Security server and the Primary CDS server in the appropriate entry fields.

```
sp3en0/ # smitty spauth_config
  --> Update SDR with DCE Master Security and CDS Server Hostnames

  Update SDR with DCE Master Security and CDS Server Hostnames

  Type or select values in entry fields.
  Press Enter AFTER making all desired changes.

  [Entry Fields]

  Master Security Server hostname  [arthur]
  CDS Server hostname             [arthur]
```
Or use the following command:

```
setupdce -v -u -s <master_security_server> -d <initial_cds_server>
```

You can verify the settings in the SDR with the `splstdata -e` command.

```
sp3en0/ # splstdata -e
...
sec_master arthur
cds_server arthur
cell_name /.../itso_cell.itso.ibm.com
...
```

10. In “**Step 42: Configure Admin Portion of DCE Clients**”, you create the necessary DCE security and CDS entries for the SP nodes. Select **Configure DCE Clients (Admin portion)** from the “RS/6000 SP Security” SMIT panel, and enter `cell_admin` as the cell administrator ID and “/.:/lan-profile” as the profile ID.

```
sp3en0/ # smitty spauth_config
  --> Configure DCE Clients (Admin portion)

  Configure DCE Clients (Admin portion)
  
  Type or select values in entry fields.
  Press Enter AFTER making all desired changes.

  [Entry Fields]
  Cell Administrator id [cell_admin]
  Lan Profile id [/.:/lan-profile]
```

Instead of using SMIT, you can use the following command:

```
setupdce -c cell_admin -l /.:/lan/profile
```

**Important**

The next installation step requires the **DCE cell administrator** password.

When you are prompted, enter the password for the `cell_admin` principal. After the configuration of all your DCE hosts has ended successfully, you can verify the creation of the entries with the `dcecp -c cell show` command:
This commands provides a view of all defined hosts in a DCE cell. You should now see all your SP nodes, including the CWS, in the output.

11. Go on to “Step 43: Select Authorization Methods for AIX Remote Commands”, and issue the `smitty spauth_config` command to get to the SMIT security menu. Select **Select Authorization Methods for AIX Remote Commands**. In the next panel, insert your system partition name and specify **dce** and **std** as authorization methods.

Or use the following command:

```
spsetauth -d -p <syspar> dce std
```

**Note**

You need to use the two methods, **dce** and **std**, at this point because you must have at least one common method defined for the system partition and the remote commands. Since we have not yet changed the authentication method for the partition, it is still set to **std**. Later, we will change these methods to only use **dce**.
12. In “Step 44: Configure SP Trusted Services to Use DCE Authentication (Required for DCE)”, select **Configure SP Trusted Services to use DCE Authentication** from the “RS/6000 SP Security” SMIT panel, and enter y when prompted. Then, type in your administrator ID and the password when required.

```bash
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # smitty spauth_config
-> Configure SP Trusted Services to use DCE Authentication
This command requires cell administrator authority. Continue? (y/n) y
```

Running "check_prereqs" subroutine ...
Checking state of DCE ...
Running "open_io" subroutine ...
Running "parse_defaults" subroutine ...
Parsing spsec_defaults file ...
Running "parse_overrides" subroutine ...
Running "create_default" subroutine ...

Please enter cell administrator id to be added to ACL admin group: \cell_admin
Adding cell administrator id to group spsec-admin ...
Creating org - spsec-services ...
Creating group - spsec-services ...
Running "create_accts_on_cws" subroutine ...

Your cell administrator password is required to create accounts. Please enter your cell administrator password: ...

Instead of using SMIT, you can use the following command:

```
cfgsp_sspec -v
```

**Important**
The following installation step requires DCE cell administrator credentials.

13. To continue with “Step 45: Create SP Services DCE Keyfiles (Required for DCE)”, you must exit your cell administrator principal. This step has to be done as the self-host principal.
When you run “Create SP Services Keyfiles” from the “RS/6000 SP Security” SMIT panel, you should get an output, such as the one shown in the next screen:

```
s3en0/ # exit
s3en0/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/sp3en0/self
```

You can also use the following command:

```
create_keyfiles -v
```

---

**Important**

The remaining installation steps require SP administrator (self-host) credentials.

14. In “Step 46: Enable Authentication Methods for AIX Remote Commands”, issue `smitty spauth_config`, and select **Enable Authentication Methods for AIX Remote Commands** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition, and specify **k5** and **std** as the authentication methods.
Instead of using SMIT, you can use the following command:

```bash
chauthpar -c -p <syspar> k5 std
```

15. In “Step 47: Enable Authentication Methods for SP Trusted Services”, you do this by selecting **Enable Authentication Methods for SP Trusted Services** from the “RS/6000 SP Security” SMIT panel. Set the “Enable on Control Workstation Only” flag to **yes** and the “Force change on nodes” flag to **no**. Select your system partition, and specify **dce** as the authentication method.

```bash
sp3en0/ # smitty spauth_config
-> Enable Authentication Methods for AIX Remote Commands
Enable Authentication Methods for AIX Remote Commands
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
Enable on Control Workstation Only  yes  +
Force change on nodes               no  +
* You cannot select YES for both entries above.

* System Partition name  sp3en0  +
* Authentication Methods  k5 std  +
```

You can also use the following command:

```bash
chauthpts -c -p <syspar> dce
```

This can take several minutes.

16. You can now change the authorization methods for the AIX remote commands to only use **dce** (remember step 11 of this procedure). Issue the **smitty spauth_config** command to get to the SMIT security menu, and select **Select Authorization Methods for AIX Remote Commands**.

```bash
sp3en0/ # smitty spauth_config
-> Enable Authentication Methods for SP Trusted Services
Enable Authentication Methods for SP Trusted Services
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
Enable on Control Workstation Only  yes  +
Force change on nodes               no  +
* You cannot select YES for both entries above.

* System Partition name  sp3en0  +
* Authentication Methods  dce
```

You can also use the following command:

```bash
chauthpts -c -p <syspar> dce
```

This can take several minutes.
the next panel, insert your system partition name, and specify dce as the authorization method.

```
sp3en0/ # smitty spauth_config
-> Select Authorization Methods for AIX Remote Commands

Select Authorization Methods for AIX Remote Commands

Type or select values in entry fields. Press Enter AFTER making all desired changes.

[Entry Fields]
* System Partition name
  sp3en0
* Authorization Methods
  dce
```

You can also use the following command:

```
spsetauth -d -p <syspar> dce
```

Verify your SRD settings with the `splstdata -p` command.

```
sp3en0/ # splstdata -p
List System Partition Information

System Partitions:
------------------
sp3en0

Syspar: sp3en0

-----------------------------------------------------------------------------
syspar_name sp3en0
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
hsm_cdb_version 940617228,332373504,0
auth_install dce
auth_root_rcmd dce
ts_auth_methods dce
auth_methods k5:std
-----------------------------------------------------------------------------
```

17. In “Step 48: Start the Key Management Daemon”, use the `spnkeyman_start` command to start the key management daemon on the CWS. You can verify that the daemon is running with the `lssrc` or the `ps` command.

```
sp3en0/ # spnkeyman_start
0513-059 The spnkeyman Subsystem has been started. Subsystem PID is 2068.
sp3en0/ # lssrc -a | grep key
spnkeyman 2068 active
sp3en0/ # ps -ef | grep key | grep -v grep
  root 2068 4170 0 14:30:29 0:02 /usr/lpp/ssp/bin/
spnkeyman
sp3en0/ #
```
18. Finish the configuration of the CWS and the installation of the nodes following the *PSSP 3.2: Installation and Migration Guide, GA22-7347.*

After the NIM installation of the nodes has succeeded, the `psspf6_script` runs the `spauthconfig` command on each node. This script installs and configures all needed DCE filesets automatically. Then, you can see the LED u95 during installation on the 3-digit display.
Chapter 5. Changing from one security state to another

In the previous chapter, we showed you how to install and configure your SP system with a specific security state. This chapter shows you how to migrate from one security state to another. Not all migration scenarios are covered here. Instead, we have picked the three most typical ones to illustrate the step-by-step procedures and to give you a feel for the kinds of things you need to consider when changing security states. The scenarios are based on a single-partition SP system with all nodes at PSSP 3.2 or higher.

5.1 Security state transitions

Our first two scenarios begin with the SP system in the Compatibility security state. This is the typical state for an SP system that has been migrated from a release of PSSP prior to version 3.2. The first scenario moves the SP system from Compatibility up to DCE. The second one moves it from Compatibility down to Minimal. The third scenario moves the SP system from Minimal back up to Compatibility. This last scenario might be used on a newly-installed, SP system that was initially configured with the Minimal security state prior to connecting it to outside networks and allowing normal user access.

Each of the scenarios takes you from the initial security state to a transitional security state in which both the old and the new states are active, and then to the final security state. The transitional state is the point at which you can verify that the new state is functioning as expected while still having access to the old state in case of problems. The transitional state also enables you to do the migration with the least amount of disruption to the production applications running on the SP system. The migration scenarios are summarized in Table 13.

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Transitional state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatibility</td>
<td>Compatibility:DCE</td>
<td>DCE</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatibility:Minimal</td>
<td>Minimal</td>
</tr>
<tr>
<td>Minimal</td>
<td>Minimal:Compatibility</td>
<td>Compatibility</td>
</tr>
</tbody>
</table>

The procedures in the following sections all follow the same general outline:

- Scenario introduction
- Planning considerations
- Procedure overview
5.2 Our lab environment

Our lab environment for the following scenarios was comprised of a single-frame SP system with a switch and twelve nodes: One high node, ten thin nodes, and one wide node, and, of course, the control workstation (CWS). The nodes and CWS were connected by the SP administrative Ethernet, and the nodes all shared the SP switch network. The CWS had a token ring interface connecting it to an outside network on which were two additional standalone machines configured as an external DCE cell. One of the standalones was configured as the DCE Security Master and Primary CDS server, and the other was configured as a DCE client. The system diagram is shown in Figure 15 on page 133.
5.3 Moving from Compatibility to DCE

If your SP system existed prior to the release of PSSP 3.2, you will have to begin with it in the Compatibility state. Once you have migrated your SP system (or at least one partition in it) to PSSP 3.2, it will, by definition, have a security state of Compatibility. This procedure shows you how to change the security state from Compatibility to DCE. If you have not already done so, read Chapter 3, “What you need to know if you plan to use DCE” on page 23.

In this procedure, we will be configuring a self-contained, DCE cell on the SP system. The only purpose of the DCE cell is to support configuration of the

---

**Figure 15. SP security redbook lab environment**

- **DCE CDS & Security Master**
  - 9.12.0.18 (arthur)

- **DCE Client**
  - 9.12.0.17 (lancelot)

- **spcws Control Workstation**
  - 9.12.0.3 (spcws)
  - 192.168.3.130 (spn00)

- **Frame 1**
  - Ethernet: 192.168.3.0/24

- **SP Switch**
  - 9.12.0.19 (arthur)

**Note**

The external DCE cell does not exist for the initial DCE scenario. In that scenario we create a self-contained DCE cell in the SP system. The CWS serves as the DCE Security Master and Primary CDS server, and the nodes are DCE clients.
SPI system at the DCE security state. To that end, we will be configuring the
CWS as the DCE Security Master and Primary CDS server. Our DCE cell
name is sp_cell, and DCE administrative principal is cell_admin. Following
this procedure, we describe other possible DCE scenarios and configuration
variations.

5.3.1 From Compatibility to DCE: Planning considerations

Before you begin this procedure, read through it, and make sure you
understand its general flow. Again, if you have not already done so, read
Chapter 3, “What you need to know if you plan to use DCE” on page 23. As
with any change to your SP system, be sure that you have a current set of
verified system backups. If possible, you should perform this procedure
during a normal maintenance window for your SP system.

Like most distributed systems, DCE is highly-dependent on the integrity of the
underlying network. Before configuring any DCE component, verify that the
following are properly configured and are functioning correctly:

- Host name settings (hostname)
- Forward and reverse name resolution (nslookup, host, /etc/resolv.conf,
  /etc/netsvc.conf, and /etc/hosts)
- Network routings (netstat -rn, ping, and traceroute)
- Packet forwarding settings (ipforwarding)

Also, verify that your date, time, and timezone settings are correct. DCE is
extremely sensitive to time discrepancies.

In this procedure, you will be asked to supply the following:

- A name for your DCE cell
- A name for your DCE cell administrator (typically, cell_admin)

For this first scenario, we assume that the SP administrator with root authority
will also be the DCE cell administrator. This seems like a fair assumption
since the DCE cell exits only within the SP system. In the section following
the procedure, we discuss other scenarios where this assumption may not be
ture.

5.3.2 From Compatibility to DCE: High-level outline

The step-by-step procedure to move from Compatibility to DCE can be broken
down into these main sections:

1. Create the DCE cell
2. Initial verification
3. Move from Compatibility to Compatibility:DCE
4. Midpoint review
5. Move from Compatibility:DCE to DCE
6. Final verification

After creating a DCE cell, you will get a snapshot of your current environment. Transition to a point where you have both the old and new security states enabled and verified; transition to just the new security state, and finish with a final snapshot of the environment.

---

### Important

Do not try to go directly from the old security state to the new one, bypassing the transitional state in which both old and new are active. The purpose of the transitional state is to ensure that your system remains operative while you enable the new state. Shutting off the old state too early in the process will result in communication disruptions between the CWS and the nodes.

---

### 5.3.3 From Compatibility to DCE: Step-by-step procedure

This procedure is based on the following assumptions:

- The SP system is configured with a single system partition.
- All nodes are at PSSP 3.2 or higher.
- The CWS and nodes are not configured with Kerberos V5 as an AIX authentication method.
- The CWS and nodes are not already members of a DCE cell.

To move from Compatibility to DCE, do the following:

#### Create the DCE cell

1. From the command line and, optionally, in the /etc/environment file, set the `RPC_UNSUPPORTED_NETIFS` variable to exclude network interfaces that should not be used for DCE traffic by either the CWS or the nodes (for example, interfaces on the nodes to which the CWS has no route):
Exploiting RS/6000 SP Security: Keeping It Safe

2. Create `/var/dce` as a separate filesystem on the CWS:

```
spcws/ # export RPC_UNSUPPORTED_NETIFS=tr1:en1:css0
spcws/ # cat /etc/environment

RPC_UNSUPPORTED_NETIFS=tr1:en1:css0
```

DCE does not check for the existence of these network interfaces. Therefore, you can use a common setting for the CWS and all nodes in your SP system.

3. Remove any DCE filesets earlier than Version 3.1 from the lppsource directory:

```
spcws/ # crfs -v jfs -g rootvg -a size=400000 -m /var/dce \
-A yes -p rw -t no
```

Based on the parameters chosen, the new `/var/dce` JFS file system is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 401408

```
spcws/ # mount /var/dce
spcws/ # df -k /var/dce

Filesystem 1024-blocks Free %Used Iused %Iused Mounted on
/dev/lv00 200704 192676 4% 17 1% /var/dce
```

DCE processes write their core files into `/var/dce`. These core files can be quite large. For this reason, we recommend creating a `/var/dce` filesystem at least 200 MB in size, especially on machines that are DCE servers. However, if you do not have the disk space or are not concerned about preserving DCE core files, you can go with the recommended minimum size of 32 MB.

```
spcws/ # rm -i /spdata/sys1/install/aix433/lppsource/dce*
```

Note

DCE does not check for the existence of these network interfaces. Therefore, you can use a common setting for the CWS and all nodes in your SP system.
4. Copy the DCE Version 3.1 filesets into the lppsource directory and rebuild the .toc file:

```
spcws/ # ls /spdata/sys1/install/aix433/lppsource | grep dce
dce.cds.3.1.0.0.I
dce.client.3.1.0.0.I
dce.compat.3.1.0.0.I
dce.doc.en_US.3.1.0.0.I
dce.doc.rte.3.1.0.0.I
dce.msg.es_ES.3.1.0.0.I
dce.msg.Ja_JP.3.1.0.0.I
dce.msg.Zh_TW.3.1.0.0.I
dce.msg.en_US.3.1.0.0.I
dce.msg.es_ES.3.1.0.0.I
dce.msg.ja_JP.3.1.0.0.I
dce.msg.ko_KR.3.1.0.0.I
dce.msg.zh_TW.3.1.0.0.I
dce.priv.3.1.0.0.I
dce.security.3.1.0.0.I
dce.sysmgmt.3.1.0.0.I
dce.tools.3.1.0.0.I
dce.web.3.1.0.0.I
dce.xdsxom.3.1.0.0.I
```

These filesets are not part of the AIX 4.3.3 distribution. Instead, they can be found on the DCE Version 3.1 distribution media, which is a separately purchasable product from IBM.

5. Using the information in Table 14, install the DCE filesets on the CWS using the `installp` command or the smitty `install_latest` fastpath:

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce.security</td>
<td></td>
</tr>
<tr>
<td>dce.security.smit</td>
<td>DCE SMIT Security Services</td>
</tr>
<tr>
<td>dce.security.rte</td>
<td>DCE Security Services</td>
</tr>
<tr>
<td>dce.cds</td>
<td></td>
</tr>
<tr>
<td>dce.cds.rte</td>
<td>DCE Cell Directory Services</td>
</tr>
</tbody>
</table>

Table 14. DCE Security, CDS, and client filesets
6. Use SMIT to configure the CWS as the DCE Security Master server for the new sp.Cell DCE cell:

<table>
<thead>
<tr>
<th>Fileset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce.cds.smit</td>
<td>DCE SMIT Cell Directory Services</td>
</tr>
<tr>
<td><strong>dce.client</strong></td>
<td></td>
</tr>
<tr>
<td>dce.client.rte.admin</td>
<td>DCE Client Administrative Tools</td>
</tr>
<tr>
<td>dce.client.rte.cds</td>
<td>DCE Client CDS Tools</td>
</tr>
<tr>
<td>dce.client.rte.config</td>
<td>DCE Client Configuration Tools</td>
</tr>
<tr>
<td>dce.client.rte.rpc</td>
<td>DCE Client RPC Tools</td>
</tr>
<tr>
<td>dce.client.rte.security</td>
<td>DCE Client Security Tools</td>
</tr>
<tr>
<td>dce.client.rte</td>
<td>DCE Client Services</td>
</tr>
<tr>
<td>dce.client.core.rte</td>
<td>DCE Client Services - FOR UPGRADES</td>
</tr>
<tr>
<td>dce.client.rte.time</td>
<td>DCE Client Time Tools</td>
</tr>
<tr>
<td>dce.client.rte.zones</td>
<td>DCE Client Time Zones</td>
</tr>
<tr>
<td>dce.client.smit</td>
<td>DCE SMIT Client Tools</td>
</tr>
<tr>
<td>dce.client.rte.pthreads</td>
<td>DCE Threads Compatibility Library</td>
</tr>
<tr>
<td>dce.client.rte.web</td>
<td>DCE Web Secure</td>
</tr>
<tr>
<td><strong>dce.xdsxom</strong></td>
<td></td>
</tr>
<tr>
<td>dce.xdsxom.rte</td>
<td>X.500 API Library</td>
</tr>
</tbody>
</table>
You will need to enter information for your DCE cell as follows:

- For **Cell name**, enter the name of your DCE cell, such as `sp_cell`.

- For **Cell ADMINISTRATOR's account**, if you are not using `cell_admin` as the name of the administrative principal for your DCE cell, enter your chosen name instead.

- For **Machine's DCE HOSTNAME**, enter a symbolic TCP/IP hostname for the CWS, such as `spcws`. The name you pick should be the first name that appears after the loopback address in a `netstat -i` listing. In most cases, this will be an `en*` address.

- For **Start components at System restart**, select Yes to have an entry added to the `/etc/inittab` file.

- For **Clean up DCE at System restart**, select Yes if you also selected Yes for **Start components at System restart**.

- For **Security Server Name**, enter the same name you entered for **Machine's DCE HOSTNAME**, for example, `spcws`. (You will also use this same name for the Initial CDS Server in the next step.)

- For all other fields, take the defaults.
It normally takes SMIT several minutes to configure the DCE Security Master server.

7. Use SMIT to configure the CWS as the Primary CDS server for the new sp_cell DCE cell:

```
spcws/ # smitty dce
  -> Configure DCE/DFS
    -> Configure DCE/DFS Servers
      -> CDS (Cell Directory Service) Server
        -> 1 initial

CDS (Cell Directory Service) Server

Type or select values in entry fields. Press Enter AFTER making all desired changes.

[Entry Fields]

* CELL name [sp_cell]
* Cell ADMINISTRATOR's account [cell_admin]
Machine's DCE HOSTNAME [spcws]
* Start components at System restart Yes +
* Clean up DCE at System restart Yes +
* Protocol tcp udp +
MASTER SECURITY Server [spcws]
Minutes to wait between pe_site file updates [1440] #
Synchronize Clocks No +
Time Server to Synchronize Clocks with []
* LAN PROFILE [lan-profile]
```

SMIT sets up the screen with the values you used to configure the DCE Security Master server. You should not have to change anything here. As was the case with the DCE Security Master server configuration, it normally takes SMIT several minutes to configure the Primary CDS server.

8. Verify that the DCE Security and CDS servers have been created on the CWS:
9. Verify that the sp_cell DCE cell has been created:

```
spcws/ # show.cfg
Gathering component state information...

Component Summary for Host: spcws

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Master server</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Initial Directory server</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>
```

```
spcws/ # dcecp -c cell show
{secservers
 /.../sp_cell/subsys/dce/sec/spcws}
{cdsservers
 /.../sp_cell/hosts/spcws}
{dcsservers}
{hosts
 /.../sp_cell/hosts/spcws}
```

Initial verification

10. Get a baseline listing of the current security state:

```
spcws/ # splstdata -p 2>\&1 | tee /tmp/splstdata.$(date +%Y%m%d
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
---------------------------------------------------------------
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4
auth_root_rcmd k4
ts auth_methods compat
auth_methods k4:std
```

Move from Compatibility to Compatibility:DCE

11. Archive the SDR:
12. Login as the DCE administrative principal, and reset the password:

```
spcws/ # dce_login cell_admin
Enter Password:
Password must be changed!
DCE LOGIN SUCCESSFUL
Warning: This account has been marked by an administrator, recommending that the password be changed.
Do you wish to change now [y/n]? (y) y
Enter New Password:
Re-enter New Password:
PASSWORD SUCCESSFULLY CHANGED
```

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/cell_admin
```

13. Create the DCE groups, organizations, principals, and accounts for PSSP, and verify that they were created:

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/cell_admin
```

```
spcws/ # config_spsec -v -c
This command requires cell administrator authority. Continue? (y/n) y
...
spcws/ # dcecp -c group cat | pg
...
spcws/ # dcecp -c organization cat
...
spcws/ # dcecp -c principal cat | pg
...
spcws/ # dcecp -c account cat | pg
...
```

14. Create the DCE keyfiles for PSSP, and verify that they were created:
15. Set the authentication method for SP Trusted Services on the CWS to both DCE and Compatibility, and verify the settings:

```
spcws/ # exit
spcws/ # klist | grep Global
   Global Principal: /.../sp_cell/hosts/spcws/self
spcws/ # create_keyfiles -c -v
...
spcws/ # dcecp -c keytab cat | pg
...
spcws/ # ll -l /spdata/sys1/keyfiles/spcws
```

```
total 72
-rw------- 1 root system 102 Oct 27 15:31 css
-rw------- 1 root system 110 Oct 27 15:31 hardmon
-rw------- 1 root system 106 Oct 27 15:31 pmand
-rw------- 1 root system 102 Oct 27 15:31 sdr
-rw------- 1 root system 116 Oct 27 15:31 sp_configd
-rw------- 1 root system 112 Oct 27 15:31 spbgroot
-rw------- 1 root system 106 Oct 27 15:31 spmgr
-rw------- 1 root system 116 Oct 27 15:33 switchtbd
-rw------- 1 root system 108 Oct 27 15:31 sysctl
```

```
spcws/ # chauthts dce compat
spcws/ # lsauthts
DCE
Compatibility
```

**Important**

This next step makes `cell_admin` the DCE SP administrative principal. You might want to use the self-host principal or a special principal that you have already created for this purpose instead.

16. Add the DCE administrative principal to the SP Trusted Services administrative and control groups, and obtain fresh DCE credentials to verify:
spcws/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
spcws/ # klist | grep Global
    Global Principal: /.../sp_cell/cell_admin
spcws/ # dcecp -c group add sdr-admin -member cell_admin
spcws/ # dcecp -c group add lm-admin -member cell_admin
spcws/ # dcecp -c group add spsec-admin -member cell_admin
spcws/ # dcecp -c group add sdr-system-class-admin -member cell_admin
spcws/ # dcecp -c group add lm-control -member cell_admin
spcws/ # exit
spcws/ # klist | grep Global
    Global Principal: /.../sp_cell/hosts/spcws/self
spcws/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
spcws/ # klist | grep -p Groups
DCE Identity Information:
Warning: Identity information is not certified
    Global Principal: /.../sp_cell/cell_admin
    Cell: 602218d2-8c9a-11d3-99de-02608c2d4a7f /.../sp_cell
    Principal: 00000064-8c9a-21d3-9900-02608c2d4a7f cell_admin
    Group: 0000000c-8c9a-21d3-9901-02608c2d4a7f none
    Local Groups:
0000000c-8c9a-21d3-9901-02608c2d4a7f none
00000064-8c9a-21d3-8201-02608c2d4a7f acct-admin
00000065-8c9a-21d3-8201-02608c2d4a7f subsys/dce/sec-admin
00000066-8c9a-21d3-8201-02608c2d4a7f subsys/dce/cds-admin
00000068-8c9a-21d3-8201-02608c2d4a7f subsys/dce/dfs-admin
00000069-8c9a-21d3-8201-02608c2d4a7f subsys/dce/audit-admin
0000006a-8c9a-21d3-8201-02608c2d4a7f subsys/dce/dced-admin
00000067-8c9a-21d3-8201-02608c2d4a7f subsys/dce/dats-admin
0000007c-8ca4-21d3-8201-02608c2d4a7f spsec-admin
00000086-8ca4-21d3-8201-02608c2d4a7f sdr-admin
00000080-8ca4-21d3-8201-02608c2d4a7f lm-admin
00000088-8ca4-21d3-8201-02608c2d4a7f sdr-system-class-admin
00000081-8ca4-21d3-8201-02608c2d4a7f lm-control

--- Note ---

Obtaining fresh DCE credentials is required so that you pick up the changes made to the SP Trusted Services administrative and control groups.

17. Complete system support installation on the CWS:
18. Add dce as one of the security capabilities required on nodes, and verify that the auth_install attribute is set to dce:k4:

```bash
spcws/ # spsetauth -i -p spcws dce k4
spcws/ #
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
------------------
spcws
Syspar: spcws
---------------------------------------------
syspar_name  spcws
ip_address   192.168.3.130
install_image default
syspar_dir   
code_version  PSSP-3.2
haem_cdb_version 941039977,692774656,0
auth_install dce:k4
auth_root_rcmd k4
ts_auth_methods compat
auth_methods  k4:std
```

19. Create DCE hostnames for the nodes, and verify that they were created:
20. Update the SDR with the DCE Security Master and Primary CDS server hostnames, and verify that it was updated:

```
spcws/ # setupdce -v -u -s spcws -d spcws
setupdce: Checking state of dce ... setupdce: Updated cell_name information into SDR. setupdce: Updated DCE security master information into SDR. setupdce: Updated DCE CDS server information into SDR.
```

```
spcws/ # splstdata -e | pg
List Site Environment Database Information

<table>
<thead>
<tr>
<th>attribute</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>control_workstation</td>
<td>spcws</td>
</tr>
<tr>
<td>sec_master</td>
<td>spcws</td>
</tr>
<tr>
<td>cds_server</td>
<td>spcws</td>
</tr>
<tr>
<td>cell_name</td>
<td>/.../sp_cell</td>
</tr>
</tbody>
</table>
```

21. Configure the nodes as DCE clients (admin-only portion):

```
spcws/ # create_dcehostname
spcws/ #
spcws/ # splstdata -n | pg
List Node Configuration Information

node# frame# slot# slots initial_hostname reliable_hostname  
dce_hostname default_route processor_type processors_installed description
----- ------ ----- ----- ----------------- ----------------- -------- -------- --------------- ------ -------- 
spn01 1 1 1 4 spn01 spn01 192.168.3.130 MP 1 ""  
spn05 5 1 5 1 spn05 spn05 192.168.3.130 UP 1 ""  
```

...
Chapter 5. Changing from one security state to another

22. Verify that the nodes have been added to the DCE cell:

```
spcws/ # setupdce -v -c cell_admin -l /./:lan-profile
...
Configuration of DCE Host, spn01, will now begin.
Configuring the Security client...
Security client configuration is complete.
Configuring the Directory client...
Directory client configuration is complete.
Configuration of DCE Host, spn01, was successful.
...
Configuration completed successfully.
```

---

**Note**

In the sp_cell DCE cell, we used `cell_admin` for our DCE administrative principal and `/./:lan-profile` for our DCE lan profile. These are the default names. If you used different names in your DCE cell, be sure to substitute the correct values in the `setupdce` command.

23. Add `dce` as one of the authorization methods for AIX remote commands, and verify that the `auth_root_rcmd` attribute is set to `dce:k4`:

```
spcws/ # dcecp -c cell show
{secservers
 /.../sp_cell/subsys/dce/sec/spcws}
{cdservers
 /.../sp_cell/hosts/spcws}
{dtservers}
{hosts
 /.../sp_cell/hosts/spcws
 /.../sp_cell/hosts/spn01
 /.../sp_cell/hosts/spn05
 /.../sp_cell/hosts/spn06
...
 /.../sp_cell/hosts/spn15}
```
24. Configure SP Trusted Services to use DCE authentication:

```
spcws/ # spsetauth -d -p spcws dce:k4
spcws/ #
spcws/ # splstdata -p 2> | tee /tmp/splstdata.$(date +%Y%m%d)
```

List System Partition Information

```
System Partitions:
------------------
spcws
```

Syspar: spcws

<table>
<thead>
<tr>
<th>syspar_name</th>
<th>spcws</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>192.168.3.130</td>
</tr>
<tr>
<td>install_image</td>
<td>default</td>
</tr>
<tr>
<td>syspar_dir</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>code_version</td>
<td>RSP-3.2</td>
</tr>
<tr>
<td>haem_cdb_version</td>
<td>941039977,692774656,0</td>
</tr>
<tr>
<td>auth_install</td>
<td>dce:k4</td>
</tr>
<tr>
<td>auth_root_rcmd</td>
<td>dce:k4</td>
</tr>
<tr>
<td>ts_auth_methods</td>
<td>compat</td>
</tr>
<tr>
<td>auth_methods</td>
<td>k4:std</td>
</tr>
</tbody>
</table>

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/cell_admin
spcws/ # config_spsec -v
...`

25. Create and/or update the SP Trusted Services keyfiles:

```
spcws/ # exit
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self
```

```
spcws/ # create_keyfiles -v
```

26. Update the .k5login and .klogin files on the CWS:

```
spcws/ # updauthfiles
spcws/ #
```

**Note**

It is not uncommon for `updauthfiles` to take several minutes to complete.
27. Enable `dce` one of the authentication methods for SP Trusted Services, and verify that the `ts_auth_methods` attribute is set to `dce:compat`:

```
spcws/ # chauthpts -c -p spcws -v dce compat
0513-095 The request for subsystem refresh was completed successfully.
spcws/ #
spcws/ # splsdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
--------------------
spcws

Syspar: spcws

------------------------------------------------------------------
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
haem_cdb_version 941039977,692774656,0
auth_install dce:k4
auth_root_rcmd dce:k4
ts_auth_methods dce:compat
auth_methods k4:std
```

**Note**

The `-c` flag on the `chauthpts` command sets `ts_auth_methods` on the CWS only. The change will be propagated to the nodes in a later step.

28. Add `k5` as one of the authentication methods for AIX remote commands, and verify that the `auth_methods` attribute is set to `k5:k4:std`:
29. Start the key management daemon on the CWS:

```
spcws/ # spnkeyman_start
0513-071 The spnkeyman Subsystem has been added.
0513-059 The spnkeyman Subsystem has been started. 
  Subsystem PID is 20260.
spcws/ #
```

30. Create `/var/dce` as a separate file system on the nodes (you could also do this with the `script.cust` script):

```
spcws/ # dsh -av 'crfs -v jfs -g rootvg -a size=400000 -m /var/dce 
    -A yes -p rw -t no'
spcws/ # dsh -av 'mount /var/dce'
spcws/ # dsh -av 'df -k /var/dce'
```

---

Note

The `-c` flag on the `chauthpar` command sets `auth_methods` on the CWS only. The change will be propagated to the nodes in a later step.
Chapter 5. Changing from one security state to another

31. Configure the nodes as DCE clients (local-only portion):

```bash
spcws/ # dsh -av '/usr/lpp/ssp/bin/spauthconfig'
...
```

**Note**

DCE processes write their core files into /var/dce. These core files can be quite large. For this reason, we recommend creating a /var/dce filesystem at least 200 MB in size, especially on machines that are DCE servers. However, if you do not have the disk space or are not concerned about preserving DCE core files, you can go with the recommended minimum size of 32 MB.

32. Verify that the nodes are configured correctly:

```bash
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self
```

```bash
spcws/ # dsh -av 'lsauthts;lsauthent;wc -c /.k5login' | dshbak -c
HOSTS ---------------------------------------------------------------
spn01 spn05 spn06 spn07
spn08 spn09 spn10 spn11
spn12 spn13 spn14 spn15
---------------------------------------------------------------------
DCE Compatibility
Kerberos 5
Kerberos 4
Standard Aix
650 /.k5login
```

```bash
spcws/ # dsh -av 'show.cfg' | pg
spn01: Gathering component state information...
spn01:
spn01: Component Summary for Host: spn01
spn01: Component Configuration State Running State
spn01: Security client Configured Running
spn01: RPC Configured Running
spn01: Directory client Configured Running
spn01:
spn01: The component summary is complete.
...
```

**Note**

You can monitor the progress of spauthconfig by doing a `tail -f /var/adm/SPlogs/auth_install/log` on the nodes.
Midpoint review

33. At this point, your `splstdata -p` listing should look like the following:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
-------------------------------------------------------------------
syspar_name  spcws
ip_address  192.168.3.130
install_image default
syspar_dir    **
code_version  PSSP-3.2
haem_cdb_version 941039977,692774656,0
auth_install  dce:k4
auth_root_rcmd dce:k4
ts_auth_methods dce:compat
auth_methods   k5:k4:std
```

*Figure 16. SDR settings for the Compatibility:DCE transitional security state*

- You are at the transitional state of Compatibility:DCE.
- You have verified that DCE is configured correctly.
- You are now ready to turn off Compatibility.

**Move from Compatibility:DCE to DCE**

34. Archive the SDR:

```
spcws/ # SDRArchive Compatibility-DCE.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is
/spdata/sys1/sdr/archives/backup.99304.1625.Compatibility-DCE.19991031
```

35. Remove `k4` as one of the authentication methods for AIX remote commands, and verify that the `auth_methods` attribute is set to `k5:std`:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
-------------------------------------------------------------------
syspar_name  spcws
ip_address  192.168.3.130
install_image default
syspar_dir    **
code_version  PSSP-3.2
haem_cdb_version 941039977,692774656,0
auth_install  dce:k4
auth_root_rcmd dce:k4
ts_auth_methods dce:compat
auth_methods   k5:k4:std
```
36. Remove `compat` as one of the authentication methods for SP Trusted Services, and verify that the `ts_auth_methods` attribute is set to `dce:`
37. Remove k4 as one of the authorization methods for AIX remote commands, and verify that the auth_root_rcmd attribute is set to dce:

```bash
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self

spcws/ # chauthpts -p spcws -v dce
...

spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
------------------
spcws
Syspar: spcws
--------------------------------------------------------------------------
syspar_name  spcws
ip_address   192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
haem_cdb_version 941039977,692774656,0
auth_install   dce:k4
auth_root_rcmd dce:k4
ts_auth_methods dce
auth_methods   k5:std

spcws/ # klist | grep Global
Global Principal: /.../sp_cell/cell_admin

spcws/ # spsetauth -d -p spcws dce
spcws/ 

spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
------------------
spcws
Syspar: spcws
--------------------------------------------------------------------------
syspar_name  spcws
ip_address   192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
haem_cdb_version 941039977,692774656,0
auth_install   dce:k4
auth_root_rcmd dce:k4
ts_auth_methods dce
auth_methods   k5:std
```
38. Remove $k_4$ as one of the security capabilities required on nodes, and verify that the auth_install attribute is set to dce.

```
spcws/ # klist | grep Global
    Global Principal: /.../sp_cell/cell_admin
spcws/ # spsetauth -i -p spcws dce
spcws/ #
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
   ------------------
spcws
Syspar: spcws
   ------------------
   syspar_name  spcws
   ip_address   192.168.3.130
   install_image default
   syspar_dir   ""
   code_version  PSSP-3.2
   haem_cdb_version 941039977,692774656,0
   auth_install  dce
   auth_root_rcmd dce
   ts_auth_methods dce
   auth_methods  k5:std
spcws/ # klist | grep Global
    Global Principal: /.../sp_cell/hosts/spcws/self
spcws/ # dsh -av 'lsauthent | dshbak -c
HOSTS ---------------------------------------------------------------
  spn01  spn05  spn06  spn07
  spn08  spn09  spn10  spn11
  spn12  spn13  spn14  spn15
---------------------------------------------------------------------
Kerberos 5
Standard Aix
```

39. Verify that Kerberos 5 and Standard AIX are now set as the authentication methods for AIX remote commands on the nodes:

```
spcws/ # klist | grep Global
    Global Principal: /.../sp_cell/hosts/spcws/self
spcws/ # dsh -av 'lsauthent | dshbak -c
HOSTS ---------------------------------------------------------------
  spn01  spn05  spn06  spn07
  spn08  spn09  spn10  spn11
  spn12  spn13  spn14  spn15
---------------------------------------------------------------------
Kerberos 5
Standard Aix
```

40. Verify that DCE is now set as the authentication method for SP Trusted Services on the nodes:
41. Verify that the `.k5login` file exists on the nodes and is not zero-length:

**DCE**

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self

spcws/ # dsh -av 'lsauthts' | dshbak -c
HOSTS ---------------------------------------------------------------
spn01    spn05    spn06    spn07
spn08    spn09    spn10    spn11
spn12    spn13    spn14    spn15
---------------------------------------------------------------------
```

```
dsh -av 'wc -c /.k5login' | dshbak -c
HOSTS ---------------------------------------------------------------
spn01    spn05    spn06    spn07
spn08    spn09    spn10    spn11
spn12    spn13    spn14    spn15
---------------------------------------------------------------------
650 /.k5login

```

42. Verify that DCE is configured and running on the nodes:

**DCE**

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self

spcws/ # dsh -av 'show.cfg' | pg
spn01: Gathering component state information...
spn01:
spn01: Component Summary for Host: spn01
spn01: Component          Configuration State  Running State
spn01: Security client    Configured          Running
spn01: RPC                Configured          Running
spn01: Directory client   Configured          Running
spn01: The component summary is complete.
...```

43. Verify that the DCE keyfiles for PSSP exist on the nodes and are not zero-length:
44. Verify that the SP Per Node Key Management is functioning properly on the nodes:

```
spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self
```

```
spcws/ # dsh -av 'ls -l /spdata/sys1/keyfiles/ssp/$(hostname)/' | pg
spn01: total 48
spn01: -rw------ 1 root system 102 Oct 29 16:13 css
spn01: -rw------ 1 root system 106 Oct 29 16:13 pmand
spn01: -rw------ 1 root system 116 Oct 29 16:13 sp_configd
spn01: -rw------ 1 root system 112 Oct 29 16:13 spbgroot
spn01: -rw------ 1 root system 116 Oct 29 16:13 switchtbld
spn01: -rw------ 1 root system 108 Oct 29 16:13 sysctl

spcws/ # klist | grep Global
Global Principal: /.../sp_cell/hosts/spcws/self
```

```
spcws/ # dsh -av 'spnkeymand -l' | dshbak -c
HOSTS ----------------------------------------------------------------
spn01 spn05 spn06 spn07
spn08 spn09 spn10 spn11
spn12 spn13 spn14 spn15
HOSTS ----------------------------------------------------------------
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
service=LoadL/Schedd expiration=0
service=LoadL/Startd expiration=0
service=LoadL/Startd expiration=0
service=LoadL/Startd expiration=0
service=LoadL/Negotiator expiration=0
service=LoadL/Master expiration=0
service=LoadL/Kbdd expiration=0
service=LoadL/GSmonitor expiration=0
service=mmfs/mmfsd expiration=0
service=rsct/rsct expiration=0
service=ssp/switchtbld expiration=0
service=ssp/css expiration=0
service=ssp/pmand expiration=0
service=ssp/sp_configd expiration=0
service=ssp/spbgroot expiration=0
```

45. Remove k4 as one of the authentication methods for AIX remote commands on the CWS, and verify that Kerberos 5 and Standard AIX are still enabled:

```
spcws/ # chauthent -k5 -std
spcws/ # lsauthent
Kerberos 5
Standard Aix
```
46. Update the boot logical volume on the CWS with the authentication method change:

```
spcws/ # /usr/sbin/savebase
```

47. Stop the kerberos daemons on the CWS:

```
spcws/ # stopsrc -s kerberos
0513-044 The kerberos Subsystem was requested to stop.
spcws/ # stopsrc -s kadmind
0513-044 The kadmind Subsystem was requested to stop.
```

48. Comment out the `kerberos` and `kadmind` entries in the `/etc/inittab` file on the CWS:

```
spcws/ # cat /etc/inittab
... :kerb:2:once:/usr/bin/startsrc -s kerberos
:kadmind:2:once:/usr/bin/startsrc -s kadmind
...
```

49. Remove old Kerberos V4 ticket cache files on the CWS and the nodes:

```
spcws/ # /usr/bin/k4destroy
...
```

```
spcws/ # dsh -av '/usr/bin/k4destroy'
```

50. You may also want to clean up other miscellaneous remnants from Kerberos V4, such as:

- The `/etc/krb.conf`, `/etc/krb.realms`, and `/etc/krb-srvtab` files
- The `.klogin` files
- The `.k` file on the CWS
- The Kerberos V4 database in the `/var/kerberos/database` directory
- The Kerberos V4 entries in the `/etc/services` file

**Final verification**

51. Your `splstdata -p` listing should now look like the following:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws

<table>
<thead>
<tr>
<th>syspar_name</th>
<th>ip_address</th>
<th>install_image</th>
<th>syspar_dir</th>
<th>code_version</th>
<th>haem_cdb_version</th>
<th>auth_install</th>
<th>auth_root_rcmd</th>
<th>ts_auth_methods</th>
<th>auth_methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>spcws</td>
<td>192.168.3.130</td>
<td>default</td>
<td>&quot;&quot;</td>
<td>PSSP-3.2</td>
<td>941039977,692774656,0</td>
<td>dce</td>
<td>dce</td>
<td>dce</td>
<td>k5:std</td>
</tr>
</tbody>
</table>
```

*Figure 17. SDR settings for the DCE security state*

52. Archive the SDR:

```
spcws/ # SDRArchive DCE.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is /spdata/sys1/sdr/archives/backup.99304.1654.DCE.19991031
```

53. (Optional) At this point, you can configure some of your nodes as DCE Security Replica and Secondary CDS servers for the cell. For example, to increase cell performance and availability, you might choose to configure the first node on each ethernet segment as a DCE Security Replica and Secondary CDS server. The process to configure these servers is virtually the same one you used to configure the DCE Security Master and Primary CDS servers. For the DCE Security Replica server, select `secondary` instead of primary. For the Secondary CDS server, select `additional` instead of `initial`. For more details on how to set up these additional servers, see Section 4.6.3, “Setting up the CWS as DCE server” on page 74, and Section 4.7.5, “Setting up a DCE Security Replica Server on a node” on page 110.
5.3.4 From Compatibility to DCE: Further considerations

If your SP system is already part of or will be joining an existing DCE cell, there are further considerations that you need to be aware of prior to migrating your SP system to the DCE security state:

- You will need to coordinate the migration with your DCE cell administrator. Certain steps in the procedure require cell_admin credentials.
- The DCE cell will need to be upgraded to DCE 3.1 or higher if it is not already there. DCE 3.1 is the minimum version of DCE supported by PSSP 3.2.
- If the DCE Security Master server and/or Primary CDS server is on a node, it will need to be moved to either the CWS or, preferably, a machine outside of the SP system. This is required to avoid a circular dependency that can occur between the node and the CWS. From the DCE standpoint, the CWS (as DCE client or DCE Security Replica and Secondary CDS server) cannot come up unless the node (as DCE Security Master and Primary CDS server) is up. On the other hand, the node cannot be powered up using PSSP if the CWS is down. The result is a deadlock between the two.
- You will need to configure the CWS as either a DCE client only or a DCE Security Replica and Secondary CDS server, rather than a DCE Security Master and Primary CDS server.
- You will need to verify that all machines in your SP system have network routes to the other machines in the DCE cell. Name resolution must also be thoroughly validated throughout the cell.
- Joining an existing DCE cell creates a software dependency between your SP system and the DCE cell. Software fixes and upgrades to the DCE cell will also need to be applied to the machines in your SP system.

If there is even a remote possibility that you will be joining an existing DCE cell, do not configure your SP system as a self-contained DCE cell. Reconfiguring your SP system to be part of the existing cell at a later date will be painful at best. You are much better off taking the time now to architect the correct solution. DCE is not trivial, merging two cells is even less so.

5.4 Moving from Compatibility to Minimal

If your SP system existed prior to the release of PSSP 3.2, you will have to begin with it in Compatibility state. Once you have migrated your SP system (or at least one partition in it) to PSSP 3.2, it will, by definition, have a security state of Compatibility. This procedure shows you how to change the security...
state from Compatibility to Minimal. If you are interested in removing Kerberos V4, this is the procedure for you.

5.4.1 From Compatibility to Minimal: Planning considerations

Before you begin this procedure, read through it, and make sure you understand its general flow. As with any change to your SP system, be sure that you have a current set of verified system backups. If possible, you should perform this procedure during a normal maintenance window for your SP system.

---

**Important**

This procedure shows you how to reduce the security state of your SP system. We do not recommend using this procedure if your SP system is connected to external networks, and under no circumstance should you use it if your SP system is connected to the Internet.

---

5.4.2 From Compatibility to Minimal: High-level outline

The step-by-step procedure to move from Compatibility to Minimal can be broken down into these main sections:

1. Initial verification
2. Move from Compatibility to Compatibility:Minimal
3. Midpoint review
4. Move from Compatibility:Minimal to Minimal
5. Final verification

You will get a snapshot of your current environment. Transition to a point where you have both the old and new security states enabled and verified; transition to just the new security state, and finish with a final snapshot of the environment.

---

**Important**

Do not try to go directly from the old security state to the new one, bypassing the transitional state in which both old and new are active. The purpose of the transitional state is to ensure that your system remains operative while you enable the new state. Shutting off the old state too early in the process will result in communications disruptions between the control workstation and the nodes.
5.4.3 From Compatibility to Minimal: Step-by-step procedure

This procedure is based on the following assumptions:

- The SP system is configured as a single-partition system.
- All nodes are at PSSP 3.2 or higher.
- The nodes and CWS are not configured with Kerberos V5 as an AIX authentication method.

To move from Compatibility to Minimal, do the following:

Initial verification

1. Get a baseline listing of the current security state:

   ```
   spcws/ # splstdata -p 2>1 | tee /tmp/splstdata.$(date +%Y%m%d)
   List System Partition Information
   System Partitions:
   ------------------
   spcws
   Syspar: spcws
   -----------------------------------------------
   syspar_name   spcws
   ip_address    192.168.3.130
   install_image default
   syspar_dir    
   code_version  PSSP-3.2
   haem_cdb_version 940342361,764607232,0
   auth_install  k4
   auth_root_rcmd k4
   ts_auth_methods compat
   auth_methods   k4:std
   
   Move from Compatibility to Compatibility:Minimal
   
   2. Archive the SDR:

   ```
   spcws/ # SDRArchive Compatibility.$(date +%Y%m%d)
   0025-322 SDRArchive: SDR archive file name is /spdata/sys1/sdr/archives/backup.99303.1058.Compatibility.19991030
   ```

   3. Add `std` as one of the security capabilities required on nodes, and verify that the `auth_install` attribute is set to `k4:std`:
4. Add `std` as one of the authorization methods for AIX remote commands, and verify that the `auth_root_rcmd` attribute is set to `k4:std`

```
spcws/ # spsetauth -i -p spcws k4 std
spcws/ #
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
-----------------------------------------------

syspar_name  spcws
ip_address    192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install  k4:std
auth_root_rcmd k4
ts_auth_methods compat
auth_methods   k4:std

spcws/ # spsetauth -d -p spcws k4 std
spcws/ #
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
-----------------------------------------------

syspar_name  spcws
ip_address    192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install  k4:std
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods   k4:std
```

5. Distribute or update the `.rhosts` and `.klogin` files on the nodes with the entries in the corresponding files on the CWS:
6. Verify that the `.rhosts` files exist on the nodes, that they are not zero-length, and that there is at least one entry for the CWS in them:

```
spcow/ # dsh -av 'ls -l /.rhosts'
spn01: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn05: -rw------- 1 root system 218 Oct 19 18:17 /rhosts
spn06: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn07: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn08: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn09: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn10: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn11: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn12: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn13: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn14: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spn15: -rw------- 1 root system 218 Oct 19 18:16 /rhosts
spcow/ # dsh -av 'grep spcow /.rhosts'
spn01: spcow
spn05: spcow
spn06: spcow
spn07: spcow
spn08: spcow
spn09: spcow
spn10: spcow
spn11: spcow
spn12: spcow
spn13: spcow
spn14: spcow
spn15: spcow
```

**Midpoint review**

7. At this point, your `splstda -p` listing should look like the following:
Chapter 5. Changing from one security state to another

Figure 18. SDR settings for the Compatibility: Minimal transitional security state

- You are at the transitional state of Compatibility: Minimal.
- You have verified that Minimal is configured correctly.
- You are now ready to turn off Compatibility.

Move from Compatibility: Minimal to Minimal

8. Archive the SDR:

```
spcws/ # SDRArchive Compatibility-Minimal.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is 
/spdata/sys1/sdr/archives/ 
backup.99303.1101.Compatibility-Minimal.19991030
```

9. Remove k4 as one of the authentication methods for AIX remote commands, and verify that the `auth_methods` attribute is set to `std`:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------

spcws

Sympar: spcws
-----------------------------------------------
sympar_name spcws
ip_address 192.168.3.130
install_image default
sympar_dir ""
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods k4:std
```
10. Remove `compat` as one of the authentication methods for SP Trusted Services, and verify that the `ts_auth_methods` attribute is set to `""` (none):

```bash
spcws/ # chauthpar -p spcws -v std
...
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
------------------
spcws
Syspar: spcws
-----------------------------------------------
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir "
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods std
```

11. Remove `k4` as one of the authorization methods for AIX remote commands, and verify that the `auth_root_rcmd` attribute is set to `std`:

```bash
spcws/ # chauthpts -p spcws -v
...
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information
System Partitions:
------------------
spcws
Syspar: spcws
-----------------------------------------------
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir "
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods ""
auth_methods std
```
12. Remove k4 as one of the security capabilities required on nodes, and verify that the auth_install attribute is set to std:

```
spcws/ # spsetauth -d -p spcws std
spcws/ #
spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:  
------------------
spcws
Syspar: spcws

syspar_name     spcws
ip_address      192.168.3.130
install_image   default
syspar_dir       
code_version    PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install    k4:std
auth_root_rcmd  std
auth_methods    ""
auth_methods    std
```

```
spcws/ # spsetauth -i -p spcws std
spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:  
------------------
spcws
Syspar: spcws

syspar_name     spcws
ip_address      192.168.3.130
install_image   default
syspar_dir       
code_version    PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install    std
auth_root_rcmd  std
auth_methods    ""
auth_methods    std
```

13. Remove k4 as one of the authentication methods for AIX remote commands on the CWS, and verify that Standard Aix is still enabled:

```
spcws/ # spsetauth -d -p spcws std
spcws/ #
spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:  
------------------
spcws
Syspar: spcws

syspar_name     spcws
ip_address      192.168.3.130
install_image   default
syspar_dir       
code_version    PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install    std
auth_root_rcmd  std
auth_methods    ""
auth_methods    std
```

```
spcws/ # spsetauth -i -p spcws std
spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:  
------------------
spcws
Syspar: spcws

syspar_name     spcws
ip_address      192.168.3.130
install_image   default
syspar_dir       
code_version    PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install    std
auth_root_rcmd  std
auth_methods    ""
auth_methods    std
```

Chapter 5. Changing from one security state to another
Update the boot logical volume on the CWS with the authentication method change:

```
spcws/ # /usr/sbin/savebase
spcws/ #
```

Stop the kerberos daemons on the CWS:

```
spcws/ # stopsrc -s kerberos
0513-044 The kerberos Subsystem was requested to stop.
spcws/ # stopsrc -s kadmind
0513-044 The kadmind Subsystem was requested to stop.
```

Comment out the kerberos and kadmind entries in the /etc/inittab file on the CWS:

```
spcws/ # cat /etc/inittab
...:
:kerb:2:once:/usr/bin/startsrg -s kerberos
:kadm:2:once:/usr/bin/startsrg - s kadmind
...
```

Remove old Kerberos V4 ticket cache files on the CWS and the nodes:

```
spcws/ # /usr/bin/k4destroy
...
spcws/ # dsh -av '/usr/bin/k4destroy'
...
```

You may also want to clean up other miscellaneous remnants from Kerberos V4, such as:

- The /etc/krb.conf, /etc/krb.realms, /etc/krb-srvtab files
Chapter 5. Changing from one security state to another

- The ~/.klogin files
- The ~/.k file on the CWS
- The Kerberos V4 database in the /var/kerberos/database directory
- The Kerberos V4 entries in the /etc/services file

**Final verification**

19. Your `splstdata -p` listing should now look like the following:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws

Syspar: spcws
--------------------------------------------------------------------------------------------------------------------------------
syspar_name  spcws
ip_address    192.168.3.130
install_image default
syspar_dir    ""
code_version  PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install std
auth_root_rcmd std
ts_auth_methods ""
auth_methods std
```

*Figure 19. SDR settings for the Minimal security state*

20. Archive the SDR:

```
spcws/ # SDRArchive Minimal.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is
/spdata/sys1/sdr/archives/backup.99303.1105.Minimal.19991030
```

### 5.5 Moving from Minimal to Compatibility

This procedure will typically be used by someone who has just finished installing a new SP system with PSSP 3.2 configured to the Minimal security state. The SP system should not be connected to an external network until this procedure has been successfully completed and the new security state fully tested.
5.5.1 From Minimal to Compatibility: Planning considerations

Before you begin this procedure, read through it, and make sure you understand its general flow. You should also read through the steps for setting up Kerberos V4 in the PSSP 3.2: Installation and Migration Guide, GA22-7347. As with any change to your SP system, be sure that you have a current set of verified system backups. If possible, you should perform this procedure during a normal maintenance window for your SP system.

This procedure assumes that you will be configuring your CWS as the Kerberos V4 master. When configuring Kerberos V4, you will be asked to supply the following:

- The master password for the Kerberos V4 database.
- The Kerberos V4 administrative principal and instance name (typically root.admin)

Moreover, if you do not want to use the default Kerberos V4 realm name, which is your DNS domain name in uppercase, you will need to create an /etc/krb.conf file before running setup_authent. Details can be found in the PSSP Installation and Migration Guide, GA22-7347.

5.5.2 From Minimal to Compatibility: High-level outline

The step-by-step procedure to move from Minimal to Compatibility can be broken down into these main sections:

1. Initial verification
2. Move from Minimal to Minimal:Compatibility
3. Midpoint review
4. Move from Minimal:Compatibility to Minimal
5. Final verification

You will get a snapshot of your current environment. Transition to a point where you have both the old and new security states enabled and verified; transition to just the new security state, and finish with a final snapshot of the environment.
5.5.3 From Minimal to Compatibility: Step-by-step procedure

This procedure is based on the following assumptions:

- The SP system is configured with a single system partition.
- All nodes are at PSSP 3.2 or higher.
- The nodes and CWS are not configured with Kerberos V5 like an AIX authentication method.
- The CWS will house the master copy of the Kerberos V4 database.

To move from Minimal to Compatibility, do the following:

Initial verification

1. Get a baseline listing of the current security state:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
-------------------------------------------------------------------------------------------------------------------------
syspar_name  spcws
ip_address   192.168.3.130
install_image default
syspar_dir   
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install  std
auth_root_rcmd std
ts_auth_methods  
auth_methods   std
```

Move from Minimal to Minimal:Compatibility

2. Archive the SDR:
3. Update root’s PATH with /usr/lpp/ssp/kerberos/bin

```
spcws/ # echo $PATH
/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/local/bin:/usr/lpp/ssp/bin:/usr/lib/instl:/usr/lpp/ssp/kerberos/bin
```

4. Add k4 as one of the authentication methods for AIX remote commands on the CWS, and verify that Kerberos 4 and Standard AIX are enabled:

```
spcws/ # chauthent -k4 -std

spcws/ # lsauthent
Kerberos 4
Standard Aix
```

**Important**

Issuing the `chauthent` command without any parameters turns off all authentication methods for this machine, effectively disabling remote logins for everyone.

5. Initialize RS/6000 SP authentication services:

```
spcws/ # setup_authent
...
```

6. Verify that you now have Kerberos V4 set up on the CWS:
Chapter 5. Changing from one security state to another

7. Set `compat` as the authentication method for SP Trusted Services on the CWS, and verify the setting:

```
spcws/ # chauthts compat
```

8. Complete system support installation on the CWS:

```
spcws/ # install_cw
```

9. Add `k4` as one of the security capabilities required on nodes, and verify that the `auth_install` attribute is set to `k4:std`:

```
spcws/ # ps -ef | grep kerb | grep -v grep
root 14326 4170 0 15:57:45 - 0:00 /usr/lpp/ssp/kerberos/etc/kerberos
root 23972 4170 0 15:57:48 - 0:00 /usr/lpp/ssp/kerberos/etc/kadmind -n
```

```
spcws/ # grep kerb /etc/inittab
kerb:2:once:/usr/bin/startsrc -s kerberos
spcws/ # grep kadm /etc/inittab
kadm:2:once:/usr/bin/startsrc -s kadmind
```

```
spcws/ # klist
Ticket file: /tmp/tkt0
Principal: root.admin@SPCWS
```

```
Issued     Expires     Principal
Oct 19 15:58:19 Nov 18 14:58:19 krbtgt.SPCWS@SPCWS
```

```
```
10. Add k4 as one of the authorization methods for AIX remote commands, and verify that the auth_root_rcmd attribute is set to k4:std:

```bash
spcws/ # spsetauth -i -p spcws k4 std
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
```

```
List System Partition Information
System Partitions: 
------------------
spcws
Syspar: spcws
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
auth_install k4:std
auth_root_rcmd std
ts_auth_methods ""
auth_methods std
```

```bash
spcws/ # spsetauth -d -p spcws k4 std
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
```

```
List System Partition Information
System Partitions: 
------------------
spcws
Syspar: spcws
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods ""
auth_methods std
```

11. Enable compat as the authentication method for SP Trusted Services, and verify that the ts_auth_methods attribute is set to compat:

```bash
spcws/ # spsetauth -d -p spcws k4 std
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
```

```
List System Partition Information
System Partitions: 
------------------
spcws
Syspar: spcws
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods ""
auth_methods std
```

```bash
spcws/ # spsetauth -i -p spcws k4 std
```

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
```

```
List System Partition Information
System Partitions: 
------------------
spcws
Syspar: spcws
syspar_name spcws
ip_address 192.168.3.130
install_image default
syspar_dir ""
code_version PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install k4:std
auth_root_rcmd k4:std
ts_auth_methods ""
auth_methods std
```
12. Add k4 as one of the authentication methods for AIX remote commands, and verify that the auth_methods attribute is set to k4:std.

```
spcws/ # chauthpts -c -p spcws -v compat
...
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
------------------------------------------------
syspar_name  spcws
ip_address   192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
base_cdb_version 940342361,764607232,0
auth_install  k4:std
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods   std
```

**Note**

The -c flag on the chauthpts command sets ts_auth_methods on the CWS only. The change will be propagated to the nodes in a later step.
13. Run `setup_server` on the CWS to populate the Kerberos V4 database with `rcmd` principals for the nodes, and verify that the `rcmd` principals were added:

```bash
spcws/ # setup_server
...
setup_server: Processing complete (rc= 0).
```

```bash
spcws/ # /usr/lpp/ssp/kerberos/etc/kdb_util dump /tmp/kerberos.out
spcws/ # pg /tmp/kerberos.out
...
rcmd spn01 255 1 1 0 ...
rcmd spn05 255 1 1 0 ...
rcmd spn06 255 1 1 0 ...
...
spcws/ # rm /tmp/kerberos.out
```

**Note**

The `-c` flag on the `chauthpar` command sets `auth_methods` on the CWS only. The change will be propagated to the nodes in a later step.

Be sure to remove the `/tmp/kerberos.out` file as soon as you are finished with it.
14. Enable k4 as one of the authentication methods for AIX remote commands on the nodes:

```
spcws/ # chauthpar -p spcws -v k4 std
...
```

15. Set the nodes to customize, run setup_server, and verify the settings:

```
spcws/ # spbootins -r customize -l 1,5,6,7,8,9,10,11,12,13,14,15
> 2>&1 | tee /tmp/setup_server.out
spcws/ #
spcws/ # splstda -b
...
```

16. Customize the nodes:

```
spcws/ # dsh -av '/etc/rc.sp'
...
```

17. Verify that Kerberos 4 and Standard Aix are now set as the authentication methods for AIX remote commands on the nodes:

```
spcws/ # dsh -av 'lsauthent' | dshbak -c
HOSTS ----------------------------------------------------------------
spn01   spn05   spn06   spn07
spn08   spn09   spn10  spn11
spn12   spn13   spn14  spn15
------------------------------------------------------------------------
Kerberos 4
Standard Aix
```

18. Verify that Compatibility is now set as the authentication method for SP Trusted Services on the nodes:

```
spcws/ # dsh -av 'lsauthts' | dshbak -c
HOSTS ----------------------------------------------------------------
spn01   spn05   spn06   spn07
spn08   spn09   spn10  spn11
spn12   spn13   spn14  spn15
------------------------------------------------------------------------
Compatibility
```

19. Verify that there are rcmd service tickets for each of the nodes:
Midpoint review

20. At this point, your `splstdata -p` listing should look like the following:

```
spcws/ # klist
Ticket file: /tmp/tkt0
Principal: root.admin@SPCWS

<table>
<thead>
<tr>
<th>Issued</th>
<th>Expires</th>
<th>Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 19 15:58:19</td>
<td>Nov 18 14:58:19</td>
<td>krbtgt.SPCWS@SPCWS</td>
</tr>
<tr>
<td>Oct 19 16:08:40</td>
<td>Nov 18 15:08:40</td>
<td>hardmcn.spcws@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:32</td>
<td>Nov 18 16:04:32</td>
<td>rcmd.spn01@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:02:49</td>
<td>Nov 18 16:02:49</td>
<td>rcmd.spn05@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:26</td>
<td>Nov 18 16:04:26</td>
<td>rcmd.spn06@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:30</td>
<td>Nov 18 16:04:30</td>
<td>rcmd.spn07@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:28</td>
<td>Nov 18 16:04:28</td>
<td>rcmd.spn08@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:22</td>
<td>Nov 18 16:04:22</td>
<td>rcmd.spn09@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:42</td>
<td>Nov 18 16:04:42</td>
<td>rcmd.spn10@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:27</td>
<td>Nov 18 16:04:27</td>
<td>rcmd.spn11@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:38</td>
<td>Nov 18 16:04:38</td>
<td>rcmd.spn12@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:52</td>
<td>Nov 18 16:04:52</td>
<td>rcmd.spn13@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:23</td>
<td>Nov 18 16:04:23</td>
<td>rcmd.spn14@SPCWS</td>
</tr>
<tr>
<td>Oct 19 17:04:54</td>
<td>Nov 18 16:04:54</td>
<td>rcmd.spn15@SPCWS</td>
</tr>
</tbody>
</table>
```

Figure 20. SDR settings for the Minimal:Compatibility transitional security state

- You are at the transitional state of Minimal:Compatibility.
- You have verified that Compatibility is configured correctly.
- You are now ready to turn off Minimal.

Move from Minimal:Compatibility to Compatibility
Since the `auth_methods` and `ts_auth_methods` attributes are already set to the correct values for the Compatibility security state, we only need to change the `auth_root_rcmd` and `auth_install` attributes.

21. Archive the SDR:

```bash
spcws/ # SDRArchive Minimal-Compatibility.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is
/spdata/sys1/sdr/archives/ \
backup.99303.1148.Minimal-Compatibility.19991030
```

22. Remove `std` as one of the authorization methods for AIX remote commands, and verify that the `auth_root_rcmd` attribute is set to `k4`:

```bash
spcws/ # spsetauth -d -p spcws k4
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
------------------
spcws
Syspar: spcws
---------------------------------------------------------------
syspar_name   spcws
ip_address    192.168.3.130
install_image default
syspar_dir    
code_version  PSSP-3.2
haem_cdb_version 940342361,764607232,0
auth_install   k4:std
auth_root_rcmd k4
ts_auth_methods compat
auth_methods   k4:std
```

23. Remove `std` as one of the security capabilities required on nodes, and verify that the `auth_install` attribute is set to `k4`:
24. Set the nodes to customize, run `setup_server`, and verify the settings:

```
spcws/ # spsetauth -i -p spcws k4
spcws/ #
spcws/ # splstdata -p 2>/dev/null | tee /tmp/splstdata.$(date +%Y%m%d)
```

List System Partition Information

System Partitions:  
------------------

```
spcws/ Syspar: spcws
---------------------------------------------------------------
syspar_name     spcws
ip_address      192.168.3.130
install_image   default
syspar_dir       
code_version     PSSP-3.2
haem_col_version 940342361,764607232,0
auth_install    k4
auth_root_rcmd  k4
ts_auth_methods compat
auth_methods    k4:std
```

25. Customize the nodes:

```
spcws/ # spbootins -r customize -l 1,5,6,7,8,9,10,11,12,13,14,15 \
> 2>/dev/null | tee /tmp/setup_server.out
spcws/ #
spcws/ # splstdata -b
...
```

26. Verify the status of the `.rhosts` files on the nodes:

```
spcws/ # dsh -av `'/etc/rc.sp'
...
```

**Note**  
This step removes the PSSP entries from the `.rhosts` files on the nodes. If those entries were the only ones in the file, it also removes the file.
Final verification

27. Your `splstdata -p` listing should now look like the following:

```
spcws/ # splstdata -p 2>&1 | tee /tmp/splstdata.$(date +%Y%m%d)
List System Partition Information

System Partitions:
-------------------
spcws

Syspar: spcws
-----------------------------
  syspar_name  spcws
  ip_address   192.168.3.130
  install_image default
  syspar_dir   
  code_version PSSP-3.2
  haem_cdb_version 940342361,764607232,0
  auth_install  k4
  auth_root_rcmd k4
  ts_auth_methods compat
  auth_methods  k4:std
```

Figure 21. SDR settings for the Compatibility security state

28. Archive the SDR:
spcws/ # SDRArchive Compatibility.$(date +%Y%m%d)
0025-322 SDRArchive: SDR archive file name is
/spdata/sys1/sdr/archives/backup.99303.1149.Compatibility.19991030
Chapter 6. Restricted Root Access (RRA)

Several SP system components use `rsh` to remotely issue commands on the CWS (from a node) or on various nodes (from the CWS). Likewise, several components use `rcp` to copy files between the CWS and SP nodes (and vice versa). In order to provide this capability, PSSP has effectively defined one root user across the entire SP system. As such, gaining access as root on any node in the SP system implies that the user can gain access as root on the CWS and all other nodes in the SP.

When, for instance, the SP is used as a server consolidation system, this single root user may not be desirable. As of PSSP 3.2, a feature called Restricted Root Access (RRA) is included to limit the uses of `rsh` and `rcp` within PSSP software. When RRA is enabled, it restricts root `rsh` and `rcp` authorizations from the nodes to the CWS, but permits CWS-to-node `rsh` and `rcp` access. When RRA is activated, it will reconfigure the SP and the root owned remote command authorizations such that:

- When run as a root process on the CWS, SP software can continue to issue `rsh` and `rcp` commands to any node within the SP system. Authorization file entries will be created by PSSP on the nodes for which a root process on the CWS can obtain the necessary credentials to issue these commands.

- SP software run as a root process on a node no longer requires the capability to issue `rsh` and `rcp` commands to the CWS or to any other node within the SP system. PSSP will no longer create authorization file entries on the CWS or nodes that grant a root process on a node remote command access.

Furthermore, when PSSP software on the nodes needs to copy or access files or execute commands on other nodes or the CWS, `sysctl` is used to perform the task from the CWS. The restrictions imposed by using RRA have implications for the functionality of some PSSP and non-PSSP components as well as system management.

RRA can be activated under all supported security configurations in PSSP 3.2. However, RRA is of most interest in PSSP `dce`, `dce:compat`, and `compat` security modes. RRA use under PSSP’s Minimal security state (none/std) is of little value. RRA within all four major security configurations is described later in this chapter.
For the remainder of the RRA sections, all references to access control lists (ACLs) apply to root owned `sysctl` and AIX remote command ACLs because they apply to the different PSSP security configurations.

### 6.1 How does it work?

RRA can be turned on by changing an indicator in the SDR. Because the SDR (in non-DCE environments) can be changed by root automatically, it was necessary to create a new SDR class that can only be changed by root on the CWS. This class is the `SP_Restricted` class, which contains the `restrict_root_rcmd` attribute. The default value is false.

The indicator can be changed by using the “Site environment” SMIT panel or by using the `spsitenv` command. If changed, the authorization files on the CWS and the nodes are immediately updated.

All commands that require `rsh/rcp` access check the indicator each time access is required. They will automatically use the `sysctl` method if RRA is enabled.

#### 6.1.1 AIX and PSSP remote commands

AIX remote command authorization files for root on the nodes and the CWS are generated and updated by the `updauthfiles` script. This script is run at various times including when the RRA state is changed, when the security settings on the system are modified, and when a node is installed or booted.

When RRA is enabled, `updauthfiles` removes all known SP-generated entries in the remote command authorization files and adds the following entries, which are dependent on the authentication method set for `auth_root_rcmd`:

**Standard AIX:** `.rhosts` is modified to contain only:
- `<cwsname>`
- `<additional cws interface names>`

**Kerberos V4:** `/.klogin` is modified to contain only:
- `rcmd.<cwsname>[@<realm>]`

**Kerberos V5:** `/.k5login` is modified to contain only:
- `ssp/<cwsname>/spbroot[@<realm>]`
- `hosts/<cwsname>/self[@<realm>]`
Any manual changes previously made to these files are not removed automatically. Also, prior to PSSP 3.2, the /.klogin entries generated by the SP were not saved in a separate file. If you have migrated your system from a previous release of PSSP, there may be SP-generated entries in your /.klogin file that are no longer recognized. It is, therefore, a good idea to check the content of all root remote command authorization files after activating RRA.

6.1.2 The sysctl facility

The sysctl facility is an authenticated client/server system for running commands and TCL scripts remotely and in parallel. The sysctl server daemon, sysctld, processes all sysctl client requests for the node on which it runs. There is a sysctld daemon running on each node of the SP system as well as on the control workstation.

By default, the /etc/sysctl.conf configuration file is read and interpreted each time the sysctl server is started or restarted. This file contains the definitions of the commands that can be executed by sysctl. Additional sysctl commands are supplied with PSSP 3.2 to enable RRA. Since the contents of the /etc/sysctl.conf file on the CWS are different from the one on the nodes, the CWS version should not be included in any file distribution scheme.

Access to the new sysctl commands is provided through a set of sysctl access control list (ACL) files shipped with PSSP. For Compat mode, these ACLs are installed in the /etc directory and contain a Kerberos V4 entry for the global SP principal root.SPbgAdm. For DCE mode, the sysctld daemon initializes new DCE ACL objects (relating to the new sysctl commands) with entries for the PSSP DCE principal, ssp/.../spbgroot, which allow the principal to access the new sysctl commands. (For Minimal mode, none/std, with RRA enabled, sysctld's ACL entries must be manually updated to allow the new sysctl commands to be executed. However, this is not a recommended security configuration under which to enable RRA. This is discussed further in Section 6.3.4, “Security state Minimal” on page 194.

For sysctl to function correctly in an RRA-enabled environment, it is important that the sysctl daemon use the same default TCP/IP port number on all nodes and the CWS.

Overall, sysctl becomes a critical part of the SP environment when operating in an RRA-enabled mode.
Sysctl usage, ACLs, and other considerations are covered extensively in Chapter 7, “Example security scenarios” on page 199. The reader is encouraged to review this chapter for additional information about sysctl.

6.2 Limitations

RRA is a feature that creates new functionality within PSSP and fundamentally changes authorization methods. Obviously, this has consequences for some PSSP components and some AIX software products that are used in an SP environment.

6.2.1 Coexistence

Because of the new functionality, when RRA is enabled, it can only be used in an SP in which all nodes are, at least, at PSSP 3.2. At activation time, the spsitenv command will check whether all nodes are at a correct level. If there are nodes at a lower PSSP level, RRA cannot be activated.

6.2.2 VSD/GPFS

During configuration and runtime, VSD and GPFS rely on existing combinations of rsh and sysctl commands and nested sysctl calls to access information on the CWS as well as other nodes (node-to-node). The existing architecture relies on the existence of a common PSSP root identity that can be authorized successfully under rsh and sysctl ACLs. As such, when RRA is enabled, the common PSSP root access required by VSD and GPFS is disabled. When attempting to activate RRA, if VSD adapters are defined or if GPFS is installed on the control workstation, the activation will fail.

6.2.3 HACMP

HACMP/ES 4.3.0 or HACMP 4.3.1 can use Kerberos V4 as an authentication and authorization method that requires additional principals and interfaces defined in the Kerberos V4 database.

When RRA is not enabled, adding additional principals to the Kerberos V4 database can be done manually or by running the cl_setup_kerberos script. This script is run from an HACMP node and does the following:

- It reads the HACMP topology.
- It uses rsh and rcp to update the Kerberos database on the CWS.
- It creates new client keys for the HACMP nodes.
- It copies the keys from the CWS to the HACMP nodes.
When RRA is enabled, the actions performed by `cl_setup_kerberos` are no longer possible due to the HACMP `rsh` and `rcp` requirements from a node to the CWS. Also, the HACMP nodes can no longer use `rsh` and `rcp` to synchronize the HACMP ODM, given that the remote command authorization files on the CWS (and other nodes) will no longer have common root authorization from node-to-node or node-to-CWS. These authorizations will have to be added manually to the appropriate root-level remote command authorization file(s).

The synchronization capability can be restored by editing the `.klogin` files on the HACMP nodes, for a `compat/k4` enabled security mode, explicitly allowing both systems to have root access to each other. The RRA function will not interfere with custom settings.

Updating the Kerberos V4 database using the `cl_setup_kerberos` script requires root access to the CWS, which is exactly what RRA was designed to eliminate. Manual updating of the Kerberos V4 database is the most secure way to ensure that the database is updated, but this is a complex and error-prone method. A workaround is to temporarily add the HACMP nodes to the `.klogin` file on the CWS, for a `compat/k4` enabled security mode, run the `cl_setup_kerberos` script, and then remove the authorizations immediately afterwards.

Refer to the official HACMP documentation for details on running with HACMP under various PSSP security configurations.

### 6.2.4 HACWS

When using a Highly Available Control Workstation (HACWS), there are no problems in the authentication of both systems on the nodes. It is, however, a manual action to copy the authorization files, `.rhosts` or `.klogin`, to the backup CWS. It is also important that the activation of RRA be done from the (active) primary CWS, since this is also the Kerberos V4 Master.

There are several restrictions and limitations for HACWS when running under a DCE-enabled security mode and in other situations that apply when running under an RRA-enabled mode as well. Refer to the “Limitations and Restrictions” section in Chapter 4 of the book, *RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment*, GA22-7281, for complete details.

### 6.2.5 Boot/install servers

Currently, only one Boot/Install Server (the CWS) is supported when RRA is activated. Since Boot/Install Servers are NIM Masters, they need to have
remote command access to the CWS when changing SP nodes. Boot/Install Servers also need \texttt{rsh} capability on their client nodes during installation time. At runtime, remote command authorization on the client nodes is only needed when a software (or patch) update is required.

Refer to Section 6.3.5, “Multiple Boot/install Servers in RRA” on page 195, for details on using multiple Boot/Install Servers in an RRA-enabled environment.

### 6.2.6 Ecommands

When RRA is activated, the following switch management commands can only be run from the control workstation:

- \texttt{CSS\_test}
- \texttt{Eclock}
- \texttt{Efence}
- \texttt{Eprimary}
- \texttt{Equiesce}
- \texttt{Estart}
- \texttt{Runfence}
- \texttt{Runpartition}
- \texttt{mult\_senders\_test}
- \texttt{switch\_stress}
- \texttt{wrap\_test}

### 6.2.7 System management commands

When RRA is activated, the following list of system management commands should only be run from the control workstation. If run from a node, authorization failures may prevent successful completion.

- \texttt{dsh}
- \texttt{lppdiff}
- \texttt{pcat}
- \texttt{pcp}
- \texttt{pexec}
- \texttt{pexecscr}
- \texttt{pfind}
• pkill
• pls
• pmv
• ppred
• pps
• pm
• ptest
• setup_authent
• spacctnd
• spgetdesc
• splstdata (-d, -h, -i options only)
• spmkuser

6.2.8 Additional security implications
Using RRA will not, in itself, make your SP system more secure. The fact that PSSP does not automatically authorize root to `rsh` and `rcp` from a node and gain default access on other nodes within the SP does not prevent root from logging in to another node using `telnet`. Even if you disable the ability for root to remotely log in to nodes and the CWS, root can still `su` to any user and then exploit that user’s `rsh` or other AIX and PSSP remote command capability to access other nodes or the CWS.

You will have to adapt your user management for root to fit into the new access policy. You might want to implement the policy described in Section 7.2.4, “Restricting user access” on page 205.

6.3 RRA enablement - General steps
In order to enable (activate) RRA, all nodes within the SP system require a minimum level of PSSP 3.2. If there are any nodes that are below this level of PSSP, RRA cannot be enabled. Also, if any VSD adapters are defined, or if GPFS is installed, attempts to enable RRA will fail.

For new PSSP installs:

1. Complete all steps in Chapter 2, Task C (including CWS installation and security install/configuration steps) of the *PSSP Installation and Migration Guide*, GA22-7347.
2. Enable RRA via `spsitenv` or SMIT

3. Complete the steps for customizing the nodes in Chapter 2, Task D, of the *PSSP 3.2: Installation and Migration Guide*, GA22-7347; if performing Step 56, follow the additional instructions outlined in Section 6.3.6, “Node install customization” on page 196 of this book.

4. Complete the steps for powering on and installing the nodes in Chapter 2, Task E of the *PSSP 3.2: Installation and Migration Guide*, GA22-7347.

For migrations to PSSP 3.2:

- Complete all tasks for migrating the CWS to PSSP 3.2 in Chapter 4 of the *PSSP Installation and Migration Guide*, GA22-7347.
- Complete all tasks for migrating nodes to PSSP 3.2 in Chapter 4 of the *PSSP Installation and Migration Guide*, GA22-7347.
- Enable RRA via `spsitenv` or SMIT.
- If multiple Boot/Install Server nodes exist, follow the instructions outlined in Section 6.3.5, “Multiple Boot/install Servers in RRA” on page 195.
- For future installations of new nodes that require customization, follow the instructions in Section 6.3.6, “Node install customization” on page 196.

### 6.3.1 Spsitenv and RRA

You can set the `restrict_root_rcmd` attribute using the SMIT menu that configures the “Site environment Information”:

```
smit site_env_dialog
  -> Change the “Root remote command access restricted” field
```

or by using `spsitenv`:

```
spsitenv restrict_root_rcmd=true
```

or by using the “Set Site Environment Information” TaskGuide.

You can see the current setting by using the following command:

```
splstdata -e
```

and scanning for the line containing `restrict_root_rcmd`, or by simply issuing the following command:

```
splstdata -e | grep restrict_root_rcmd
```

A value of `false` means that RRA is not enabled, and `true` means that it is enabled.
When the state of RRA is changed to true, action is taken immediately to update all authorization files within the SP. All PSSP commands that required rsh/rcp access will switch to using sysctl procedures.

All old entries in the root remote command authorization files will be preserved. The SP-generated entries are identified by looking at the files. All entries that are present in those files will be removed from the authorization files when RRA is activated, leaving only entries that allow the CWS to access the host, as well as any previously existing custom entries.

Issuing spsitenv will generate a failure message and not enable RRA if any of the following conditions are satisfied:

• VSDs exist
• GPFS exists
• A node at a level of PSSP below 3.2
• In dce mode, when the DCE identity (principal) under which the spsitenv command is being issued is not a member of the PSSP DCE group, sdr-restricted.

6.3.2 Security state DCE

To change the RRA indicator in the SDR via spsitenv, the DCE identity (principal) under which the command is being issued must be a member of the PSSP DCE group, sdr-restricted. Once RRA is enabled, root’s /.k5login file on the nodes will be updated to contain only one PSSP DCE principal, the CWS’s spbgroot principal, and an entry for the CWS’s DCE self-host principal. All customer entries are not touched by RRA enablement.

If you were using the DCE self-host principal as the principal under which you work normally, nothing will change when operating from the CWS in an enabled RRA mode. From other nodes, however, the automatic authorization of root-level self-host principals on those nodes will no longer be allowed. This is exactly what is intended.

If you use a different DCE principal to work under, and that principal was added to /.k5login prior to enabling RRA, this entry is preserved by PSSP as noted earlier.

However, it must be noted that, if SP administrators create DCE credentials on nodes under the same DCE principal that appears in the CWS’s /.k5login file, unless the credentials are destroyed on the nodes, another root user can use these credentials to gain entry to the CWS via rsh, rlogin, and telnet. It is strongly recommended that administrators not issue dce_login commands.
on nodes under principals whose identities appear in the CWS /.k5login file or in the /.k5login files on other nodes. Doing so introduces the risk of a security compromise that RRA seeks to help reduce.

Root-owned DCE credentials that do not belong to PSSP services or the DCE self-host principal should be destroyed on the nodes as appropriate. Refer to the IBM DCE for AIX guides for commands on destroying DCE credentials and other recommendations for cleaning up DCE credentials.

6.3.3 Security state Compatibility

Before you activate RRA, first, destroy all Kerberos V4 root administrator (root.admin) tickets on the nodes: dsh -av k4destroy. This prevents root users on the nodes from gaining access on the CWS. Note, however, that the k4destroy command remotely executed on all the nodes (via dsh) will only destroy the Kerberos V4 credentials stored in root's default Kerberos V4 ticket cache location, /tmp/tkt0. All other root-owned credential files stored under /tmp or other directories must be removed manually.

As an overview, when RRA is enabled, root’s /.klogin files are changed automatically, and the PSSP software will start to use sysctl commands in place of rsh and rcp commands. The appropriate PSSP Kerberos V4 principal entries are already present in the needed sysctl ACL files.

From a more detailed point of view, when RRA is activated, the /.klogin files on the nodes are changed to contain only the PSSP Kerberos V4 rcmd.* principal for the CWS. (Non-PSSP Kerberos V4 principals are not removed.)

However, on the CWS, the root administrator entry is preserved. This means that you cannot use the root administrator principal to gain access via remote commands to the nodes, but you can use it on the nodes to the CWS. This may seem contradictory, but the idea is that the root administrator principal is no longer used for remote commands, especially not on the nodes. If a ticket file (/tmp/tkt0) is left behind on a node, the root user could get to the CWS automatically.

There are three scenarios that you could follow to do remote commands when RRA is enabled:

1. Standard: Use the root.admin principal only for administrative purposes, such as adding nodes to the SP or adding Kerberos V4 principals. Use the /usr/lpp/ssp/rcmd/bin/rcmdtgt command to get a ticket automatically for the rcmd.<CWS> principal to use for remote commands.
- You will need to use the root administrator principal for access to `hardmon`; so, you will need to switch to this principal when using the `spmon` command.

- The `rcmd` principal has a never-ending ticket. This forms a security risk.

- You could add the `rcmd` principal to the `hmacl` file (see scenario #3), but that would increase the security risk.

2. Simple: Keep using the root administrator principal for remote commands. You will need to add an entry for the root administrator to the `/klogin` files on the nodes.

- Using the root administrator principal actively may expose the CWS and all nodes any time a ticket file for this principal exists on a node.

3. Secure: Create a new Kerberos principal for daily use. All the necessary actions are explained in the following:

- Activities that involve changing the Kerberos database still require the root administrator principal, but all other activities can be performed using a new root principal that does not have administrative access to the Kerberos database.

- This is the most secure method because it separates Kerberos administration privileges from remote commands privileges.

- The following steps need to be taken on the CWS to configure the new principal:

```bash
sp3en0/ # kadmin
Welcome to the Kerberos V4 Administration Program, version 2
Type "help" if you need it.
admin: ank Usage: add_new_key <user_name>
admin: ank sp_admin
Admin password:
Password for sp_admin:
Verifying, please re-enter Password for sp_admin:
sp_admin added to database.
admin: quit
Cleaning up and exiting.
sp3en0/ # echo "1 sp_admin vsm" >> /spdata/sys1/spmon/hmacls
sp3en0/ # cat /spdata/sys1/spmon/hmacls
sp3en0 root.admin a
1 root.admin vsm
1 root.SPbgAdm vsm
1 hardmon.sp3en0 vsm
1 sp_admin vsm
sp3en0/ # stopsrc -s hardmon
sp3en0/ # startsrc -s hardmon
```
Now, change the /.klogin file on the nodes to contain sp_admin@<REALM>, and authenticate yourself as sp_admin on the CWS. You can now issue rsh and rcp commands to the nodes and use spmon on the CWS.

### 6.3.4 Security state Minimal

Enabling RRA when running the SP with the authentication method for Trusted Services set to "" (none) does make each node more independent than in other PSSP security configurations. However, there is no additional security to be gained by running with RRA enabled in a none/std mode (where std is the AIX authentication method).

In none/std, PSSP services either perform authentication and authorization based on the user’s apparent uid and gid, or, in the case of sysctl, based on the apparent uid@hostname entries in its compat ACL files. (Of course, in none/std, the sysctl daemon can only authenticate the user based on the apparent uid of the user’s request.)

There are three instances when sysctl access on the CWS is required:

1. During node installation: /etc/sysctl.install.acl
2. When using switch commands: /etc/sysctl.rootcmds.acl
3. When using event management: /etc/sysctl.haem.acl

When RRA is enabled in a none/std environment, the sysctl ACLs on the CWS, as noted above, are not automatically updated with root@node_host_name entries. Without these entries, the sysctl commands will fail. Such sysctl ACL entries must be manually entered into the needed ACL files.

However, granting the nodes such authority on the CWS will partly undo the result the RRA-enabled mode attempts to curtail, thus, the need to manually update the ACLs. But, caution must be exercised when updating the ACLs with such data. Once sysctl ACLs are updated with root@node_host_name for each node in the SP, any user on a node that appears to be root@node_host_name can now run sysctl commands protected by the ACLs.

Furthermore, as nodes are added or deleted from the system, while under a none/std mode, the CWS’s sysctl ACL entries must be updated accordingly.

You can run the SP in a Minimal security state with RRA enabled, but you will encounter problems when using Switch or Event Management and when installing nodes. To avoid these problems, you could add the _other_unauth entry in various sysctl ACL files, but this would open up the CWS. In fact, the
use of _other_unauth should be avoided at all costs because this grants authorization to any user, regardless of their uid or hostname.

--- Important ---
In short, running with RRA enabled in a none/std security environment is not recommended.

6.3.5 Multiple Boot/Install Servers in RRA

Using multiple Boot/Install Servers in RRA is not recommended and is not automatically supported by PSSP. However, depending on the size of your system and network loads, it may not be possible to install your system with a single Boot/Install Server.

Boot/Install Servers are NIM Masters and, therefore, require rsh and rcp access to both the CWS and to the nodes they serve. PSSP will not automatically create the correct entries in the authorization files to allow these commands to work.

To use additional Boot/Install Servers, follow this procedure to manually establish the correct authorizations on your system:

On the CWS, the authorization files must have the following changes, depending on the setting of auth_root_rcmd:

- standard: An entry for the Boot/Install Server node host name in /.rhosts
- k4: An entry for the Boot/Install Server node rcmd principal in /.klogin
- doe: An entry for the self-host and the spbgroot principal for the Boot/Install Server node

On the Boot/Install Server node, you need to edit /etc/sysctl.conf to include the following entries:

- Include /usr/lpp/ssp/sysctl/bin/install.cmds
- Include /usr/lpp/ssp/sysctl/bin/switch.cmds
- Include /usr/lpp/ssp/samples/sysctl/firstboot.cmds, if initiating a node install customization

You need to recycle syctld (stopsrc -s syctld; startsrc -s syctld) on the CWS (and all Boot/Install Servers) to pick up the sysctl changes.
Because multiple Boot/Install Server nodes require additional manual configuration, warning messages to this effect will be generated by the following commands:

- `spsitenv`
- `SDR_config`
- `spmkvgobj`
- `spchvgobj`

Additional considerations for node install customization are addressed in the following section.

### 6.3.6 Node install customization

In order to do install customization using the firstboot-cust procedure when RRA is enabled, you need to do the following:

- Add the sysctl call for `copy_env_files` to `firstboot.cust` on the CWS (and all Boot/Install Servers).
- Edit the `copy_env_file` TCL procedure in the `/usr/lpp/ssp/samples/sysctl/firstboot.cmds` file on the CWS (and all Boot/Install Servers) to include the `rsh` and `rcp` commands required for your customization.

Once the node is installed, or as part of `firstboot.cust`, the remote command authorization files on the node serviced by the non-CWS Boot/Install Server need to be updated with the following changes:

- `standard` - An entry for the Boot/Install Server node host name in `./.rhosts`
- `k4` - An entry for the Boot/Install Server node `rcmd` principal in `./.klogin`
- `dce` - An entry for the `self-host` and the `spbgroot` principal for the Boot/Install Server node

### 6.4 Troubleshooting tips

If you experience unexpected remote command authorization failures, the following should be checked:

- The SDR Site Environment settings. Run `splstdata -e` on the CWS to make sure the RRA setting is correct. If not, run `spsitenv` for the `restrict_root_rcmd` attribute.
• The remote command authorization files on the CWS. If they are not correct, run `spsitenv` for the `restrict_root_rcmd` attribute to force all authorization files to be regenerated.

• The remote command authorization files on the nodes. If they are not correct, run `spsitenv` for the `restrict_root_rcmd` attribute to force all authorization files to be regenerated.
Chapter 7. Example security scenarios

The security features that PSSP software provides were explained in the previous chapters, and scenarios described how to achieve a desired security state. There were three security states defined in Section Chapter 2., “SP security concepts and terminology” on page 11: Minimal, Compatibility, and DCE. This chapter outlines four common site environments that use these security states and recommends additional security measures for them. After the scenarios, a few considerations for high security environments are discussed, followed by intrusion detection techniques.

For each site environment, a set of security measures are discussed. The aim is not to provide a detailed theoretical explanation of the subject but to provide enough information to get you started. When appropriate, an example is given. You need to consult the system manuals or the documentation provided with the tool for more detailed information.


Detailed information on basic AIX security is available in the redbook, *Elements of security: AIX 4.1*, GG24-4433. It provides you with descriptions of many standard AIX security features.

The following section discusses four typical scenarios, ranging from low security to high security. You may pick one of these scenarios or simply select features from them at will. Of course, the security measures discussed in lower security scenarios also apply to the higher security ones. Keep in mind that your overall security is just as strong as its weakest part.

Scenarios discussed:

1. Open environment
   - Low security, the environment is highly-trusted
   - Uses the Minimal security state

2. Standard commercial systems
   - Used internally within a company
   - Use common sense security measures
   - Gain security with a minimum of money and effort
   - Use the Compatibility security state
3. Mission-critical commercial systems
   - High value internal systems, downtime is not acceptable
   - Cannot trust the intranet
   - Needs extra effort to prevent disruption of service
   - Uses the Compatibility security state

4. Internet connected systems
   - Potential threat is high
   - Cannot trust networks
   - Requires strong authentication and encryption
   - Requires intrusion detection
   - Uses the DCE security state

After the scenarios, some considerations for high security environments are discussed. Concluding this chapter is a section on intrusion detection. Tools to help you discover whether your system has been compromised are described in Section 7.6, “Intrusion detection and monitoring” on page 226.

7.1 Open environment

Having hardly any security at all is something that will appeal especially to technical and scientific users. This scenario should not be considered for commercial systems or systems that have confidential data stored on them.

An open environment does not imply that there is no authorization or authentication mechanism at all. Normal UNIX methods, such as password and host authentication still apply, and these give the system a certain base level of protection. The environment in which the SP operates has to be trusted to use this security level. If the network or the other systems connected through it are not trusted, better ways of authentication are required.

With PSSP 3.2, the possibility exists to refrain from using security enhancements, such as Kerberos, and rely on the basic AIX authentication and authorization schemes. This means that all authorizations required for the PSSP daemons and remote commands should be done through \texttt{.rhosts} files in the root user’s home directory. The authentication is only performed by checking the IP address referred to in the incoming packet and doing a reverse name lookup, which is then automatically accepted when listed.
This setup will reduce the complexity of installation and maintenance of PSSP significantly, but, because the network is highly-trusted, the system is very vulnerable to attack there. This should be taken into account.

**SP security state: Minimal**

The authentication methods to be installed (auth_install) are set to std, which means that no extra software needs to be installed. The authentication setting for the SP Trusted Services (ts_auth_methods) is set to "" (none); so, they only use standard AIX authentication and verification methods. The authentication methods (auth_methods) used by AIX are set to std; so, normal AIX authentication is used. The authorization for remote commands (auth_root_rcmd) is set to std, which means .rhosts files for root.

<table>
<thead>
<tr>
<th>Security State</th>
<th>Security Attribute Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auth_install</td>
</tr>
<tr>
<td>Minimal</td>
<td>std</td>
</tr>
</tbody>
</table>

**Table 15. SP security state advised for an open environment**

7.2 **Standard commercial systems**

Most commercial systems are not connected to the Internet but operate on an internal network. These networks tend to be trusted more than the Internet making the demands on security less drastic. The security measures discussed in this setup mostly apply to system internal security and avoiding network services that are not secure.

If you enhance security, it will require more time and effort. The 80/20 rule states that 80 percent of the result is reached with 20 percent of the effort. This scenario is probably closest to this rule. Once you are required to enhance security beyond this, be prepared to invest more time and money in doing so.

Security standards should be raised by using AIX and PSSP features or by disabling some of them. Standard system security measures should be taken, such as password rules, ftp restrictions, and so on.

You need to define a security policy. To help you, there are several security-related books available. For example, Chapter 2 of the O’Reilly book, *Practical Unix and Internet Security*, ISBN 1565921488, by Simson Garfinkel and Gene Spafford, explains the relevant issues when defining a policy. When defining a policy for an SP system, the choices are sometimes different than
for a standalone system. For example, the SP switch is a private network, which can be trusted more than a normal network.

To help you understand the consequences of standard AIX security measures, we discuss these topics in the following sections:

- NIS and NFS risks
- NIS+ and Secure NFS
- Disable services that are not needed
- Restrict user access
- Disable automatic authorization using the .rhosts or hosts.equiv file
- Do not use X11, or secure the connection
- Take backup tapes offline to prevent tampering

**SP security state**

PSSP is set to the Compatibility security state. This state is the same as the standard on PSSP 3.1 and earlier. It provides proper authentication for all rsh and rcp commands needed for managing the system. Authorization will be automatic for kerberized commands, and password entry is required for any other command.

The authentication methods to be installed (auth_install) are set to k4, which means the ssp.clients fileset will be installed. The authentication setting for the trusted services (ts_auth_methods) is set to compat, which will cause most of the PSSP Trusted Services to do a simple (non-K4) authentication check. The authentication methods (auth_methods) used by AIX are set to k4 and std, which will cause AIX to try to get Kerberos V4 authentication first, and, if that fails, it will try the standard AIX method. The authorization for remote commands (auth_root_rcmd) is done by using the Kerberos V4 method, which means a .klogin file is generated for root when customizing or installing a node. Table 16 shows the SP security state advised for commercial systems.

<table>
<thead>
<tr>
<th>Security State</th>
<th>auth_install</th>
<th>auth_root_rcmd</th>
<th>ts_auth_methods</th>
<th>auth_methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatibility</td>
<td>k4</td>
<td>k4</td>
<td>compat</td>
<td>k4:std</td>
</tr>
</tbody>
</table>

Table 16. SP security state advised for commercial systems
7.2.1 NIS and NFS risks

NIS and NFS both use Sun's Remote Procedure Call (RPC) protocol to access data on remote systems. The RPC protocol can use several authentication methods; the one used by NIS and NFS is called AUTH_UNIX or, sometimes, AUTH_SYS.

With AUTH_UNIX, the client will include the user and group ID of the user in the call. The server assumes that the client provides the correct information, and, based on its access control lists (for NFS, the /etc/exports file), it will either grant or deny permission.

In an SP environment, the use of NIS for providing password data should be avoided since there are other, more secure ways available. These include supper, DCE user management, and NIS+. Using NIS to access non-sensitive data is, of course, not a security risk.

NFS is required in an SP because PSSP uses it to install nodes, customize nodes, and boot nodes in maintenance mode. You cannot disable NFS on the CWS, but you should always use NFS on the internal SP networks (ethernet, switch) only.

This example shows how you can unintentionally expose your system:

You export the mksysb images filesystem on the CWS with root-access for the external interface of a node. On the external network, another system uses the same IP address as your node (spoofing) and mounts the NFS file system. Because it has root access, it can read the backup image and extract the /etc/security/passwd file. It could even alter the image.

To make NFS a bit more secure, you should always specify which hosts have access to the exported filesystems and the filesystems should be exported read-only if possible. You should also add the following line to /etc/rc.net:

no -o ipignoreredirects=1

This will prevent AIX from accepting redirect messages for IP addresses, thus, preventing a system on another network from succeeding in impersonating a node on the internal SP ethernet.

7.2.2 NIS+ and Secure NFS

NIS+ and Secure NFS use Sun's RPC protocol, just as NIS and NFS do, but they differ in the authentication protocol used. Standard NIS and NFS use the AUTH_UNIX method, while the more secure variants use AUTH_DES to authenticate all calls.
The AUTH_DES protocol is used by Secure RPC. This protocol uses DES-based encryption employing a public and private key system. This ensures proper authentication of all requests and, thus, provides a secure way of sharing data through Secure NFS and user management.

In order to use Secure RPC, you will need to set up the public and private keys and create principals for all users. The easiest way to do this is to set up NIS+, which is still no trivial matter.

NIS+ is included in AIX version 4.3.3. More information on setting up and administering NIS+ can be found in the system manual, Network Information Services (NIS and NIS+) Guide, SC23-4310.

7.2.3 Disabling services

Not giving the opportunity to exploit weaknesses in AIX is a good idea. By disabling services that you are not using anyway, you can tighten security without much effort. Whether or not you require these services depends on the use of your system. You should use caution when disabling services.

The following services are candidates for disabling:

- Run by /etc/inetd.conf:
  - rexec, this requires .netrc files, which should never be allowed.
  - ntalk, responds to the talk command
  - rwalld, responds to the rwall command
  - rstatd, gives performance information about the system
  - ruserd, responds to the rusers command
  - sprayd, responds to packets sent by the spray command
  - pcnfsd, used for NFS exports to PCs
  - echo, returns data packets to a host
  - discard, discards incoming data packets
  - chargen, discards incoming packets, and sends random packets back
  - daytime, returns the date and time in user readable form
  - time, returns the date and time in machine readable form
  - cmqd, used for the Common Desktop Environment
  - dtspcd, used for the Common Desktop Environment

- Run from the /etc/inittab:
  - rc.dt, starts up the common desktop daemons
  - rc.nfs, starts NFS server daemons, not needed for client access.
  - httpdlite, used for man page lookups
  - infod, only AIX 4.2 and earlier, used by the Info Explorer.
  - pdnsd, used by xIC++ for documentation services
- Run from /etc/rc.tcpip:
  - sendmail, if mail is not actively used, it is a security risk
  - snmpd, if the PSSP level is less than 3.1, and not used
  - dpid2, legacy snmp thing, no longer used

7.2.4 Restricting user access

Since the network in which the system operates is more or less trusted, the main threat will be users logging in to systems where they should have no rights to log in. The PSSP software provides the `spacs_cntrl` tool for controlling interactive user access to nodes. It can be used to allow or deny access on a node for a specific user.

The `spacs_cntrl` tool will change the following attributes for a denied user:

In the `/etc/security/user` file:

```
ttys = ALL, !RSH, !REXEC
login = false
rlogin = false
```

You can deny access to each user separately for each node in the system, or you can deny access to the system by default and only allow certain users to log in. If you want to deny all, you will have to set this manually:

```
chsec -f /etc/security/user -s root -a tty="ALL"
chsec -f /etc/security/user -s default -a rlogin=false
chsec -f /etc/security/user -s default -a tty="ALL, !RSH, !REXEC"
spacs_cntrl allow user1
```

Root access should be restricted to certain key administrators, but, in practice, the root password tends to go around among more people than was intended. One way of restricting access to root is to disable remote login possibilities for root.

Of course, this means that users will now have to use the `su` command to become root. This can be restricted to just a few people by creating a `suroot` group. Only members of this group can then `su` to root. This is how it is done:

```
mkgroup -a id=300 suroot
chgroup users=root,admin1,admin2 suroot
chuser sugroups=suroot rlogin=false
```

Chapter 7. Example security scenarios  205
Now, only the admin1 and admin2 user can su to root. For safety reasons, you should always leave the login attribute as true on all systems. The only way to get to the console of a node is by using the slterm program on the CWS, which requires root authority there.

7.2.5 Do not allow automatic login authorization

The standard UNIX remote authentication mechanism is not very sophisticated. It will believe whatever the client tells it; so, if an IP address has been spoofed, the incoming rlogin session will be granted access because the name belonging to that address is listed in the .rhosts file of that user.

We advise that you not have any automatic login authorization enabled within the SP. The /etc/hosts.equiv file and the /.rhosts file should not contain any entries. Also, .netrc files should not be created. Within AIX, it is not possible to prevent users from creating such files, but, if stated in the security policy, it could be enforced by regularly scanning the home directories of all users.

7.2.6 Securing X11 connections

This section has been taken from the redbook, The RS/6000 SP Inside Out, SG24-5374.

If configured improperly, the X Windows system can be a major security hole. It is used to connect X servers (machines with a graphical display like the SP administrator’s workstation) with X clients (machines that want to display graphical output on an X server, like the SP control workstation). If the X server is not secured, everybody can monitor or, even, control the X server’s resources. This includes the keyboard and screen; so, everything that is typed or displayed on an unprotected X server can be monitored.

There are two access control commands to restrict X connections to an X server: xhost, which controls access based on IP names/addresses of the clients, and xauth, which provides more fine-grained access control through the use of cookies. Both are contained in the X11.apps.config fileset.

The xhost command is the simplest way to restrict access; it has to be invoked on the X server (the machine whose display/keyboard is to be secured). Entering the command without parameters lists the current setup; xhost + or xhost - globally enables or disables access to the X server, and individual client hosts can be added or removed by specifying the host’s name after the plus or minus parameter. The following sequence of commands illustrates this. At the end of the sequence, only X clients running on the control workstation, sp5cw0, can access the local X server.
However, this is insecure if the client machine is a multi-user machine. Anybody who has a login on sp5cw0 is allowed to connect to the local machine’s X server and can, for example, record passwords typed in at this workstation.

To limit access to specific users, such as the root user on sp5cw0, the Xauthority mechanism is used. When the X server starts up, it generates a secret key called a cookie (normally in a format called MIT-MAGIC-COOKIE-1, although other formats exist) and stores this key in the $HOME/.Xauthority file in the home directory of the user who started the X server. If an X client wants to access the X server, it needs to know this key and has to present it to the X server.

The local user who started the X server immediately has access to it, but all other users on the X server machine and all users on other X client machines first need to get this key. Of course, this transfer has to be secured. Securely transferring the key can be challenging. Using an NFS-mounted file system, for example, cannot be considered secure since it is relatively easy to bypass file access permissions of NFS-exported file systems. If the shared file system is in AFS or DFS, this is much more secure. If there is no shared file system, an actual copy has to be performed, which might also expose the key.

The xauth command can be used on the client machine to add the cookie to the .Xauthority file of the user whose processes want to access the X server as shown in the following example:

```
sp3en0/ # which xhost
xhost is a tracked alias for /usr/bin/X11/xhost

sp3en0/ # xhost +
access control disabled, clients can connect from any host

sp3en0/ # xhost -
access control enabled, only authorized clients can connect

sp3en0/ # xhost -
access control enabled, only authorized clients can connect

sp3en0/ # xhost + sp5cw0
sp5cw0 being added to access control list

sp3en0/ # xhost
access control enabled, only authorized clients can connect

sp3cw0.itso.ibm.com
```
The `xauth list` command displays the contents of the `.Xauthority` file, showing one line per `<hostname>:<display>` pair. The cookies are displayed as a string of 32 hex digits. Individual keys for a `<hostname>:<display>` pair can be added and removed by the `add` and `remove` options, using the same format as the `list` output. Each time the X server starts, it creates a new cookie that has to be transferred to its clients.

The secure shell, described in Section 7.3.2, “Using encrypted login sessions” on page 213, transparently encrypts all X traffic in a separate channel. On the remote node, a new `.Xauthority` file is created, and the cookie is used to secure the communication between the sshe daemon and the X client. On the local node, the cookie is necessary to set up a secure connection between the X server and the sshe client. This is very convenient because both the text-based login and any X Windows traffic are automatically protected.

### 7.2.7 Securing your backups

Many sites use a centralized backup system with a large tape library for backups. Because of the large number of tapes in the library, it is no longer necessary to load tapes for each backup. The main reason to export tapes from the library is when they need to be transported elsewhere for disaster recovery services.

Imagine this: Hackers have gained access to your system. They start using your system for their own purposes. When they no longer need your system, they have two options: Simply leave or erase their tracks. To erase all signs of
their presence they gain access to your backup server and delete all the backups that are in the library. They then delete all files in your system. No one will now be able to find out their identity.

You could classify a successful attack on your system as a disaster and, accordingly, have a disaster recovery plan ready. This will require offline storage of a complete set of backup tapes; so, you can rebuild your system from scratch.

7.3 Mission-critical commercial systems

These systems have to be protected from misuse because their value is such that corruption could threaten the existence of the company. Since they are not connected to the Internet, the main threat is an attack from the inside. It should, therefore, not trust the intranet.

Since it is not connected to the Internet, many of the standard UNIX services are not needed outside of the SP. We can, therefore, block them more easily. The threat with internal systems is probably not from professional hackers but from your own employees. Since these employees are operating in a more controlled environment, they cannot use the same tactics as Internet-based hackers because they can be identified more easily.

To ensure that the system is well-shielded from internal attacks, It should use authentication for root commands. It can also use the packet filtering option that is built into AIX to block services that are not required on the intranet.

Encrypted login sessions protect against password sniffers. Grabbing a password simply by installing a program on a PC that listens in on all passing traffic is a way of gaining illegal access that is too simple to ignore.

We recommend the following additional security measures:

- Restrict access to the system by setting up IP security and blocking all ports lower than 1024 on the external network interface.
- Allow only encrypted logins over insecure networks via SSH.
- Optionally set `umask` on nodes and CWS for root to 027.

**SP security state**

The SP security state is the same as on a standard commercial system, Compatibility. The authentication for remote commands is done through Kerberos V4. The authorization is Kerberos V4 for `rcp` and `rsh`, and a password is required for all other access. When SSH is used instead of
standard telnet, this prevents sending passwords unencrypted over the network, thus, increasing security. Table 17 lists the SP security state advised for mission-critical commercial systems.

Table 17. SP security state advised for mission-critical commercial systems

<table>
<thead>
<tr>
<th>Security State</th>
<th>Security Attribute Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>auth_install</td>
</tr>
<tr>
<td>Compatibility</td>
<td>k4</td>
</tr>
</tbody>
</table>

### 7.3.1 Restrict access to the system by setting up IP Security

IP Security (IPSec) is a separately-installable fileset provided with AIX 4.3. The software is actually derived from the IBM Secure Network Gateway (SNG) product and provides AIX with the ability to support IP tunnels and filters.

IP tunnels are a standard way of encapsulating data transferred over IP in a secure manner. To achieve this, the tunnel is authenticated, and all data passing through it is encrypted. Tunnels are, by nature, point-to-point connections and, therefore, not of much use inside an SP. They can be very useful, however, to provide secure access to a remote system, even if the connection is made over the Internet. This could very well apply to remote system management.

IP filtering is a basic function in which incoming and outgoing packets can be accepted or denied based on a variety of characteristics. This allows a system administrator to configure the host to control the traffic between this host and other hosts. Filtering is done on a variety of packet properties, such as source and destination addresses, IP Version (4 or 6), subnet masks, protocol, port, routing characteristics, fragmentation, interface, and tunnel definition. This filtering is done only at the IP layer; nothing is changed at the application layer; so, no changes to the applications are required.

IP filtering can be set up on a per-tunnel basis. If no tunnels are defined, the filtering settings apply to regular network traffic. Filter definitions are known as rules. The rules are processed from lower to higher numbers. If a rule is found that applies to the packet, it will be used, and all remaining rules will be disregarded. There are two basic ways of setting up filtering. One way is to start out by denying all traffic on a given interface (or all interfaces) and then permitting certain ports to have inbound, outbound, or both-way traffic. The other way is to permit all traffic on an interface and then deny access to certain ports.
For a mission-critical system, it is most likely that the application (which is really the critical part) will want to use IP ports located in the higher range (larger than 1023). The ports ranging from 1-1023 are also known as *trusted ports*. These ports can only be listened to by a program running with root authority. The services these ports provide are well-defined; so, the results of blocking them are known. The application’s port should not be restricted, but the trusted ports are not needed on the external network of an SP and can, therefore, be blocked. If a specific service is needed, a special rule can be added to exclude that port from the blockade.

If you are not sure which ports are used exactly, you can look it up in the `netstat -a` output, or (and this is much nicer) by using the `lsof` (list open files) program, a public domain tool that can tell you what files (and this includes sockets) a program has open. The `lsof` tool can be downloaded in installp format from: [http://www.bull.com](http://www.bull.com).

The following steps show how to set up the filter rules and activate the filter mechanism. For more information, see the IP Security configuration section of the system manual, *AIX Version 4.3 System Management Guide: Communications and Networks*, SC23-4127.

The IP Security software can be found in the `bos.net.ipsec.rte` fileset. Before filtering can be activated, rules must be set up properly. This can be done using the following fastpath:

```
#smitty ips4_conf_filter
```

Choose **Add an IP Security Filter Rule**, and add a standard rule to prevent access to any of the trusted ports (port 1-1023) on the *tr0* interface.
If you want to be able to log in to the server, you might want to grant access on the telnet port to a certain subset of systems. Just make a rule that permits that. Once you have made the rule, you want to be sure the rules are in the correct order so that, when a telnet packet is evaluated, it is allowed through.

NFS services can be disabled by blocking port 2049. Only do this for the external interface, you should not disable NFS on the internal SP ethernet.

You will need to disable the SP Trusted Services on the external interface to prevent their outside use. The ports are different on each installation and can be found in the `/etc/services` file and by running the following command:

```
SDRGetObjects SP_ports
```

Choose **Move IP Security Filter Rules** from the Configure IP security Filter Rules SMIT menu. The filter rules are always parsed form the top down. The first rule that applies to the packet is used, and the remaining rules are not evaluated. You may need to move several entries around to fit your design.

Once you are satisfied with the filtering settings, you can start the IPSec device and then start the filtering.

```
1. mkdev -l ipsec -t 4
2. mkfilt -v 4 -u
```
To check your work:

```
sp3n15/ # lsdev -Cc ipsec
ipsec_v4 Available  IP Version 4 Security Extension
sp3n15/ # lsfilt
Beginning of IPv4 filter rules.
Rule 1:
  Rule action : permit
  Source Address : 0.0.0.0
  Source Mask : 0.0.0.0
  Destination Address : 0.0.0.0
  Destination Mask : 0.0.0.0
  Source Routing : no
  Protocol : udp
  Source Port : eq 4001
  Destination Port : eq 4001
...
```

When you are satisfied with the settings, you can use the `expfilt` command to export your filter definitions. You can then distribute them to other nodes and use the `impfilt` command to import them there.

### 7.3.2 Using encrypted login sessions

You may have very strict password rules and require your users to regularly refresh their passwords, but, if the network over which those passwords are transmitted is not secure, anybody with access to it may listen in. This is also known as **sniffing**, and, with the right equipment, it is not difficult to do.

To prevent password sniffing, you need to secure the communication between the client and the server system. You could do this by using hardware methods, such as building a separate ethernet segment that is connected to a few administrators workstations and the server, or by using a hardware encryption card in your system. You could also use a software mechanism that encrypts the traffic between two hosts. Of course, using encryption will have a performance impact if a large amount of data is sent. You will need to take this into account before implementing an encryption system.

When Kerberos V4 or DCE is used as an authentication method in PSSP, the kerberized versions of `rsh` and `rcp` only authenticate the connection. They do not encrypt the data. This means that, although the remote system is guaranteed to be the right one, passwords that are entered in a kerberized `rsh` are still sent in clear text over the network.

If you want to have encrypted sessions, you can use one of the following options:

- IP tunnels provided by IPSec
Secure Shell (SSH)

IPSec provides an encrypted, authenticated tunnel between two hosts on an IP level. This guarantees that all communication between those hosts is encrypted. This also means that, for each server that needs to be communicated with, a tunnel should already have been created.

SSH is a commercial product by Data Fellows Inc. but is freely-available for some non-commercial use. See their Web-site for more details:

http://www.ssh.org

Setting up SSH is not a difficult task. In fact, one of the design criteria was to make it as easy as possible. SSH uses a private/public key scheme, whereby the server process (sshd) and the client process (ssh, scp) each have their own set of keys. Users need their own set of client keys, and must make sure the public key is available on the remote system.

Each system that is an SSH server needs its own set of private and public keys. The server must keep the private key secret, but it will give its public key to any ssh process that makes a connection to it.

After installing SSH, create the /etc/ssh directory, and copy the config file into it. Now, you can call the ssh-keygen program that will generate a pair of private and public keys:

```
sp3n14/etc/ssh # ls
sshd_config
sp3n14/etc/ssh # ssh-keygen
Initializing random number generator...
Generating p: .....................................++ (distance 576)
Generating q: ..........................++ (distance 407)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key (//.ssh/identity): /etc/ssh/ssh_host_key
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key is:
1024 33 1082908...1766272998073 root@sp3n14
Your public key has been saved in /etc/ssh/ssh_host_key.pub
sp3n14/etc/ssh # sshd
sp3n14/etc/ssh #
```

Figure 25. Generating keys for an SSH server

Now that you have your server set up, you can set up your user account. You can use the same ssh-keygen program to generate your keys. If you just press
Enter at the Enter passphrase line, you will not be asked for a password when using SSH. We recommend that you do use a password.

```
sp3n14/home/joe # ssh-keygen
Initializing random number generator...
Generating p: .........................................+++ (distance 402)
Generating q: ......++ (distance 66)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key (/home/joe/.ssh/identity):
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /home/joe/.ssh/identity.
Your public key is:
1024 37 115969...9479637 joe@sp3n14
Your public key has been saved in /home/joe/.ssh/identity.pub
```

**Figure 26. Generating keys for an SSH user**

The `identity.pub` file contains your public key. Each user needs to have their own set of keys. The public key has to be copied to the remote site and renamed to a file, called `.ssh/authorized_keys`. You can copy the `authorized_keys` file to all nodes on which you want to use SSH.

When using DFS or automounter for home directories, there is no need to remote copy files. Just change the name of the `identity.pub` file to `authorized_keys`, and your public key is known everywhere.

More information can be found at the following Web site:

http://www.ssh.org

### 7.3.3 Umask settings for root

The default umask setting for root on the CWS or the nodes is 022. This is often not acceptable since many companies have standard rules for setting the umask to 027 or even 077. You can set the umask to these values on an SP, but you should be aware that the PSSP software expects a umask of 022. We, therefore, do not recommend changing the umask for root on an SP system.

If you decide to change the umask anyway, you should set the umask back to 022 before performing any of the following tasks (this is probably not a complete list):

- setup_server
- syspar_ctrl
- services_config
On the node, before customizing it
On the node, before migrating it

The files in the following directories should have read permission set for the world:

- /spdata/sys1/install/pssp
- /spdata/sys1/install/<lppsource name>/lppsource
- /spdata/sys1/install/pssplpp/PSSP-X.Y/
- /tftpboot (except *-srvtab files)

When restoring a system, the backup will have the umask set too restrictive. To change that, you will need to change the umask before it becomes active. You can do this by adding this line to the script.cust file:

chsec -f /etc/security/user -s root -a umask=022

You can change it back after the migration completes.

### 7.4 Internet connected systems

In the default PSSP configuration, by exposing one node in an SP system to the Internet, the whole SP system automatically becomes exposed. This is because, by default, each node has root access to all other nodes and the CWS and is connected through an Ethernet with each other node and the CWS. You can implement the Restricted Root Access feature, or you can divide the SP in partitions to separate it into multiple root-domains, but the CWS still has access to all of them. In order to protect the whole SP, each node needs to be made secure individually, but all should have the same security level.

As standard, all communication should be authenticated, and, if sensitive data, such as passwords, is transmitted, it should be encrypted. Furthermore, all services that are vulnerable to attacks should be disabled. Monitoring, integrity checking, and security tests should be performed regularly.

New security holes are being found and exploited. You will need to keep your system up-to-date with the latest patches. Information and warnings can be found at:

http://www.cert.org
http://www.ers.ibm.com
You need to know what is out there. Hackers are not the only threat, but they are the most technically advanced one. You can find out more about them at the following Web sites:

http://www.ccc.de (partly german)
http://packetstorm.securify.com (Large download area)
http://www.attrition.org/ (Not a "Hacker site" of course)
http://www.cyberarmy.com/t-50/index.shtml (Top 50 "security" sites)

Downloading anything from the above sites is of course risky; be very careful when unpacking and compiling any software you have found there.

We will now discuss a number of tools for protecting the whole SP complex (not all of them in this section - some are discussed in Section 7.6, “Intrusion detection and monitoring” on page 226).

Reducing threats from the Internet:

- Use IPSec's packet filtering ability to shield the SP system from unwanted outside access.
- Set up DCE so that the SP Trusted Services are properly authenticated

Encrypting important data:

- The *super* tool should not be used for distributing password files because it does not encrypt the data. User management should be made secure by:
  - Having no user access to the systems
  - NIS+, although the encryption is not very strong
  - Using DCE user management

Intrusion detection:

- Use TCB or Tripwire
- Check log files

Disabling or restricting unsecure services:

- The *sendmail* daemon should be disabled, unless it is tcp-wrapped or absolutely required.
- NFS should only be used over a trusted network.
- Use TCP Wrappers to restrict access to services provided by *inetd*.

**SP security state**
The potential threat caused by connecting a system to the Internet requires the highest possible security setting in PSSP. The DCE security state provides better authentication schemes than the Compatibility security state and a framework that can be exploited to increase user and system security.

The authentication methods to be installed (auth_install) are set to dce, which means that the DCE client software will be installed. The authentication setting for the SP Trusted Services (ts_auth_methods) is also set to dce, which causes them to authenticate themselves as DCE principals. The authentication methods (auth_methods) used by AIX are set to dce and std, which will cause AIX to try to get Kerberos V5 authentication first, and, if that fails, it will try the standard AIX method. The authorization for remote commands (auth_root_rcmd) is done by using the Kerberos V5 method, which means a .k5login file is generated for root when customizing or installing a node.

Table 18. SP security state settings advised for Internet connected systems

<table>
<thead>
<tr>
<th>Security State</th>
<th>Security Attribute Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCE</td>
<td>auth_install: dce</td>
</tr>
<tr>
<td></td>
<td>auth_root_rcmd: dce</td>
</tr>
<tr>
<td></td>
<td>ts_auth_methods: dce</td>
</tr>
<tr>
<td></td>
<td>auth_methods: dce:std</td>
</tr>
</tbody>
</table>

7.4.1 Set up DCE to properly authenticate PSSP daemons

When the ts_auth_methods security attribute is set to compat, the SP Trusted Services use a locally-based authentication scheme. Some daemons, such as sysctld, also use Kerberos V4. Most of the SP Trusted Services daemons will accept a request if it comes from a node inside the SP and is issued by root. This authentication method is not very strong and, certainly, is not adequate for an Internet-connected system.

As of PSSP 3.2, when DCE is set as the authentication method, Kerberos V5 is used as authentication for all of the SP Trusted Services daemons. This will prevent unauthenticated programs to disrupt or abuse PSSP services.

7.4.2 Enhance filtering by IPsec

The procedure described in Section 7.3.1, “Restrict access to the system by setting up IP Security” on page 210, illustrates how to set up IP security to block traffic on ports lower than 1024. This was done to simplify the administration, since most commercial applications are not clear about which ports they use, although they do use port numbers higher than 1023.

When operating in an Internet-exposed environment, the security should be cranked up to the highest possible level. The packet filtering done by IP
Security can be enhanced by starting out denying all ports on the external interface and only opening up specific ports. Since the way the applications running in this environment operate should be better documented, the consequences of blocking all ports and allowing only specific ports to pass traffic can be predicted better.

The filter rule that is evaluated last can be set with the `-z` option of the `mkfilt` command. When `-z D` is specified, all packets will be denied by default. `-z P` permits all packets to pass by default. This option can only be used for all interfaces at the same time. We do not recommend denying all packets on all interfaces because this also blocks the SP Ethernet.

Instead, you can deny all access to a certain adapter by filling out the following SMIT screen:

```
# smitty ips4_conf_filter
```

Choose **Add an IP Security Filter Rule**.

Now, you can add a rule that prevents any packet to pass on the `tr0` interface:

<table>
<thead>
<tr>
<th>TCP</th>
<th>[Entry Fields]</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Rule Action</td>
<td><code>deny</code></td>
</tr>
<tr>
<td>* IP Source Address</td>
<td><code>0.0.0.0</code></td>
</tr>
<tr>
<td>* IP Source Mask</td>
<td><code>0.0.0.0</code></td>
</tr>
<tr>
<td>* IP Destination Address</td>
<td><code>[]</code></td>
</tr>
<tr>
<td>* IP Destination Mask</td>
<td><code>[]</code></td>
</tr>
<tr>
<td>* Apply to Source Routing? (PERMIT/inbound only)</td>
<td><code>yes</code></td>
</tr>
<tr>
<td>* Protocol</td>
<td><code>all</code></td>
</tr>
<tr>
<td>* Source Port / ICMP Type Operation</td>
<td><code>any</code></td>
</tr>
<tr>
<td>* Source Port Number / ICMP Type</td>
<td><code>0</code></td>
</tr>
<tr>
<td>* Destination Port / ICMP Code Operation</td>
<td><code>any</code></td>
</tr>
<tr>
<td>* Destination Port Number / ICMP Type</td>
<td><code>0</code></td>
</tr>
<tr>
<td>* Routing</td>
<td><code>both</code></td>
</tr>
<tr>
<td>* Direction</td>
<td><code>both</code></td>
</tr>
<tr>
<td>* Log Control</td>
<td><code>yes</code></td>
</tr>
<tr>
<td>* Fragmentation Control</td>
<td><code>all packets</code></td>
</tr>
<tr>
<td>* Tunnel ID</td>
<td><code>0</code></td>
</tr>
<tr>
<td>* Interface</td>
<td><code>tr0</code></td>
</tr>
</tbody>
</table>

*Figure 27. Adding a filter rule to IP Security to deny all access to a specific interface*

You will now have to make rules for all services you want to allow to pass the filtering system. You can limit access to those ports even more by allowing traffic to go in only one direction, inbound or outbound.
You might need to sort the rules you entered because they are evaluated from top to bottom. You need to make sure the *deny all traffic on the adapter* rule is evaluated last.

### 7.4.3 Using TCP Wrappers to restrict access to services

Building your own local packet filtering using IPSec can be very complex. If your setup requires you to allow connections on certain ports, you might want to use an external program instead of having a very long and complex set of filtering rules. Filter rules are examined for each packet until a match has been found; so, a long, very specific list will most likely cost you performance.

The TCP Wrapper tool, written by Wietse Venema, allows for a high degree of control over incoming TCP connections. The program uses access control lists to determine if a connection is allowed or not. All TCP services supported by `inetd` can be wrapped by editing the `/etc/inetd.conf` file and inserting the wrapper program before a daemon.

Access can be granted or denied by creating the `/etc/hosts.allow` and `/etc/hosts.deny` files. The TCP Wrapper installp package (http://www.bull.com) has not been compiled to support language extensions. If you want to use those functions, you will need to compile the TCP Wrapper yourself. An explanation on the format of the permission files can be seen using:

```bash
man 5 hosts_access  # you may need to set MANPATH
```

We next list a simple setup that will allow systems on the internal Ethernet to connect to node 15 using `telnet`, but connections coming from any other network will be rejected.

```
sp3n15/ # cat /etc/hosts.allow
ALL: 192.168.3.0/255.255.255.0
sp3n15/ # cat /etc/hosts.deny
ALL: ALL
sp3n15/ # grep tcpd /etc/inetd.conf
    telnet stream tcp4 nowait root /usr/local/bin/tcpd telnetd -a
sp3n15/ # tail -2 /etc/syslog.conf
    local4.debug /var/adm/ipsec.log
    mail.debug /var/tmp/syslog
sp3n15/ # tail -4 /var/tmp/syslog
    Dec 17 14:44:37 sp3n15 telnetd[5328]: connect from sp3n14
    Dec 17 14:47:25 sp3n15 telnetd[5330]: refused connect from 9.132.122.70
    Dec 17 14:48:19 sp3n15 telnetd[5334]: refused connect from sp5cw0
    Dec 17 14:56:21 sp3n15 telnetd[6152]: connect from sp3en0
```

*Figure 28. How to use TCP wrappers*
7.4.4 Sendmail

The sendmail program is very complex and has a long history of security-related problems. Caution should be used when activating sendmail. If possible, do not use sendmail at all.

If you need to use sendmail, the following setups might help reduce the risk to your SP system:

1. Use IPSec to deny inbound mail.
2. Use TCP Wrappers to start sendmail (if required) and shield inbound traffic from untrusted networks.
3. Use TCP Wrappers to deny access to sendmail from the outside network, and set up a mail relay system outside the SP. This system will forward all mail to the external network.
4. Use postfix as a replacement for sendmail.

Option one is the simplest one to achieve. You will need to add a rule that denies all inbound traffic to port 25 from the external interface.

Option two requires setting up TCP Wrappers. You can wrap any TCP-based service as long as it exits after each connection. If you tell inetd to watch for port 25, and then start the tcpd program, you can filter requests. You only need to have sendmail running when you receive mail. When you send it yourself, the mail command will start sendmail for you.

This is how it is done:

- Comment out sendmail in /etc/rc.tcpip, and stop the daemon:
  -stopsrc -s sendmail

- Enter the following line in /etc/hosts.allow to permit incoming mail only from the internal SP Ethernet:
  -smtp: 192.168.129.0/255.255.255.0

- Enter the following line in /etc/inetd.conf, and activate it:
  -smtp stream tcp nowait root /usr/local/bin/tcpd sendmail -bs
  -refresh -s inetd

Option three applies when not all your systems have an external network interface, but you still want to be able to send mail through that network. It also ensures that there is only one copy of sendmail running, and it does not run on an SP node.
You can set up the external system as a relay host for the other systems, which, then, need to be set up as clients. You will need to modify the sendmail configuration files on all systems. Do not use the CWS as a relay host.

External system: (sendmail relay host)

• Edit the /etc/sendmail.cf file. Change the following line to contain all interface names of the system, in short and long notation:

```
Cw localhost $w $?m$w.$m$. internal-host internal-host.itso.ibm.com. \
```

On all other systems, make sendmail use the relay host:

• Edit the /etc/sendmail.cf file and fill in the Ds field:

```
DSinternal-host
```

Option four is to replace the sendmail daemon with a more secure variant. Wietse Venema wrote a more secure solution for using the SMTP protocol on UNIX systems while on working at the IBM Watson Research Center. It is available under two names: IBM Secure Mailer and postfix.

Postfix is designed to be sendmail-compatible to make migration easy. Postfix supports /var/spool/mail, /etc/aliases, NIS, and .forward files. However, Postfix also attempts to be easy to administer, and, therefore, it does not use sendmail.cf.

Postfix uses multiple layers of defense to protect the local system against intruders. Almost every Postfix daemon can run in a chroot jail with fixed low privileges. There is no direct path from the network to the security-sensitive local delivery programs; an intruder has to break through several other programs first. Postfix does not even trust the contents of its own queue files, or the contents of its own IPC messages. Postfix filters sender-provided information before exporting it through environment variables. Last but not least, no Postfix program is set-uid.

More information on Postfix can be found at: http://www.postfix.org

It can be downloaded in installp format at: http://www.bull.com
7.4.5 DCE user management on an SP

The SP environment is different from the one for which DCE was originally designed in that it is not a distributed system but, rather, a centrally-managed system. In a DCE cell, all systems share security data and have a common clock source but are managed locally. The local system administrator can add users, create backups, and so on. On an SP system, the CWS is responsible for a lot of tasks that would normally be done locally. This can also be true for user management. The root user on the CWS has root access to all nodes within the SP. In a DCE environment, this is not always the case.

Using DCE user management instead of `supper` has many advantages. The central database containing all information eases the administration effort. Passwords can be changed on any DCE client in the cell, users are instantly available when created and password data does not need to be transmitted over the network.

How to set up user management in DCE is documented in chapters 7, 30, and 31 of the DCE manual, *Administrative Core Services*, SC23-2730. After creating a DCE user, you will need to add DCE as an authentication method for logins, activate the method for all users (except root), and then make sure the `dceunixd` daemon starts after a reboot:

```bash
chsec -f /etc/security/login.cfg -s DCE \
   -a "program=/usr/lib/security/DCE"
chsec -f /etc/security/user -s root -a "SYSTEM=files"
chsec -f /etc/security/user -s default -a "SYSTEM=DCE OR COMPAT"
mkitab "dceunix:2:respawn:/usr/bin/dceunixd >/dev/console 2>&1"
telinit q
```

Instead of using `SYSTEM=DCE OR COMPAT`, you could also opt for `SYSTEM=DCE OR DCE[UNAVAIL] AND COMPAT`. The first method will allow normal AIX users to log in if their DCE login failed. The second method will only allow AIX logins if DCE is not available.

At this point, any DCE user in the cell can log in to the node. This is probably not intended. The SP access control system (`spacs_cntrl`) does not work for DCE users because they have no corresponding AIX user IDs. You can, however, circumvent this by adding those stanzas yourself. You can set the `rlogin attribute` to be `false` by default, and then grant access for specific users by adding a stanza for that user containing the `rlogin attribute` set to `true`.

To disable remote access to the node by default (excluding root `rsh` commands):
You could keep an access program list for user access to all SP nodes central on the CWS and run it on all nodes. It might look like the following example:

```bash
chsec -f /etc/security/user -s root -a rlogin=false
chsec -f /etc/security/user -s default -a rlogin=false
chsec -f /etc/security/user -s root -a tty="ALL!,RSH!,REXEC"
chsec -f /etc/security/user -s root -a tty="ALL"
```

You could keep an access program list for user access to all SP nodes central on the CWS and run it on all nodes. It might look like the following example:

```bash
#!/bin/ksh
# User access control program. Users can be allowed on a per node basis
function allow_user {
    for user in $*; do
        chsec -f /etc/security/user -s $user -a rlogin=true
        chsec -f /etc/security/user -s $user -a tty="ALL"
    done
}

/usr/lpp/ssp/install/bin/node_number | read NN || NN=-1
case $NN in
    1|5|9|11|13) allow_user user01 user02 user03 ;;
    *) allow_user user04 user05 ;;
esac
```

Figure 29. Example script to manage remote login permissions (con_user.ksh)

DCE does not provide a central mechanism that limits access to a node for specific users. It does provide a local mechanism, the `/opt/dcelocal/etc/passwd_overwrite` file, which can be used to alter user attributes locally. By providing an illegal length password, it will prevent access to the system for the user with that password.

### 7.5 Special considerations for high-security environments

The SP system was originally not designed to be used in a high-security environment but, rather, in a low-security technical environment. Because of its distributed setup, with a central point of management, it is vulnerable because of its prolific use of network-based services.

Most of the vulnerabilities can be solved by using secure authentication methods, such as Kerberos, and by encrypting sensitive data. However, there are still a number of vulnerabilities left if the environment in which the SP operates cannot be trusted. Sometimes, the SP design requires the use of tools and services that cannot be made totally secure.

If the internal SP Ethernet cannot be trusted, there are some major problems concerning the installation of new nodes and the restoration of `mksysb` images.
Of course, users must not be allowed to log in without proper authentication and encryption, but, in a high-security environment, this option should not even be possible.

### 7.5.1 Using DCE as the only authentication method

As discussed in Section 7.4.5, “DCE user management on an SP” on page 223, you can use DCE user management in addition to regular AIX user management. If you do not include standard AIX authentication as a method, there are some caveats with which you should be concerned.

The authentication method set by `chauthent` only applies to remote commands, such as `rsh`, `rlogin`, `telnet`, and so on. The `SYSTEM` attribute in the `/etc/security/user` file determines the authentication methods used when logging into a node. This also applies to logging in to the console.

If you want to restrict remote access to your system to only DCE-authenticated users, you will have to set the AIX authentication method to DCE:

```
chauthpar -v k5
```

You can only change the authentication methods for the whole partition. If you do it manually on the node, the `rc.sp` script that is run from the `inittab` will change the value back to the partition default.

The authentication for logging into the node should also be set to DCE, except for root, who is only allowed console access:

```
chsh -f /etc/security/user -s root -a "SYSTEM=files"
chsh -f /etc/security/user -s default -a "SYSTEM=DCE"
```

If no DCE Security servers are available, all access to the system is denied, and password lookups for DCE users will fail. DCE users that are already logged on will not experience problems because file access is done by UID and GID instead of names.

When there are such problems, the root user can use the `s1term` program and access the node through the serial cable from the CWS.

### 7.5.2 Installing or restoring a node in a safe way

When a node is being installed, there are a number of security weaknesses:

- It will broadcast for a `bootp` server accepting the first response.
- It will create a temporary `.rhosts` file for use during installation.
• NFS is used during the installation.

Most of the vulnerabilities can be solved by disconnecting the node for any external network during installation and only connecting it when the installation has completed.

If the SP Ethernet is not trusted, the following procedure may be a solution:
1. Create a separate Ethernet
   - When using bnc cabling, create an Ethernet containing only the node and a unused Ethernet card in the CWS.
   - When using utp cabling, attach the node to an unused hub, and connect an unused Ethernet card in the CWS.
2. Set up the SDR to use a new subnet for this node
3. Install and configure the node
4. Move the IP address from the extra adapter in the CWS to the SP Ethernet adapter as an IP alias.
5. Join the node into the normal SP Ethernet

This approach will mean that any communication between this node and the other nodes over the internal Ethernet will go through the control workstation. The CWS will then need to have ipforwarding turned on.

A node is as vulnerable to attack during restoration as it is during installation. You will need to use the same procedure as for installing a new node, but you will need to change the IP address of the en0 adapter so that it is in a different subnet. This also means that you will need to reconfigure DCE to reflect this change.

7.6 Intrusion detection and monitoring

Even if you are up-to-date with all the latest security fixes and have implemented a very strict security policy, a hacker might still be able to break through your defenses. New exploits are continuously created by hackers, and software fixes are generated to stop them. Since there is no such thing as bug-free software, there is no such thing as an absolutely secure system.

Therefore, if a break-in has occurred, it is vital to know this as soon as possible so that countermeasures can be taken to avoid further damage. But, how can you tell if your system has been compromised? Hackers will do their utmost to avoid detection by exchanging system commands for their own, by editing log files, and by making it seem as if they are not present.
You will need to be vigilant if you want to keep your defences up and observant to detect if they have been breached. Vigilance can be exercised by testing the defences of your SP using tools, such as SATAN, SAINT, COPS, or Tiger. Observing the system can be done by checking its integrity with tools, such as TCB or Tripwire, and by monitoring your log files.

7.6.1 Trusted Computing Base (TCB)

The Trusted Computing Base (TCB) is a part of AIX that is designed to make sure the operating system can be trusted and to provide a means to ensure it remains that way. This is done by regular checks with the `tcbck` tool, and the use of the trusted shell. A detailed description can be found in the book, *AIX Version 4.3 System Management Guide: Operating System and Devices*, SC23-4126.

The TCB needs to be enabled at install time so it is assured that the binaries on the system can be trusted. It will then create a database with the attributes of a list of files. These files are then trusted, which can be seen by using the `-e` flag on the `ls` command.

In an SP environment, the minimal image provided by IBM to install the nodes does not have TCB enabled. You will, therefore, need to create an image from a system with TCB enabled from outside the SP, and use that as a minimal image. When setting up an SP, the CWS could be used for such a purpose.

The TCB database is located in the `/etc/security/sysck.cfg` file and contains stanzas, such as the following:

```bash
/usr/bin/lssrc:
  owner = root
group = system
mode = TCB,SGID,555
type = FILE
class = apply,inventory,bos.rte.SRC
size = 3994
checksum = "59775 4 "
```

File authenticity is defined by the size and checksum. Since there are programs available that can create new files with identical size and checksum as the original one, we do not recommend using the Trusted Computing Base for file integrity checking. We suggest using `tripwire` for that purpose.
7.6.2 Tripwire

Tripwire is a public domain tool that can perform about the same function as the tcbck tool from the TCB. It generates a database containing information about each file listed in its configuration file and can then compare the current situation against a known (non-compromised) one.

The main purpose of this procedure is to spot whether a system has been compromised. If a hacker has gained root access to the system, he will try to hide his presence from the system administrator by altering commands, such as netstat, who, ls, and so on. If the system administrator wants to spot these changes, he or she will need a reference to which to compare the current files. Tripwire will generate such a reference consisting of file size, date, permissions, and two types of checksum. The double checksum and size data makes it virtually impossible to create a replacement file with the same characteristics.

The tripwire database needs to be stored in a safe location, or a hacker might change it to reflect the alterations he or she has made. This safe location is preferably a removable read-only device, such as a floppy drive. Since most SP nodes do not have floppy drives, a secure location elsewhere in the network should be used. The CWS would apply because it has the highest security requirements. Also, the tripwire database of the CWS can be written to a floppy drive.

Once you have installed the tripwire files, you have two important files:

- The program file: /usr/security/bin/tripwire
- The configuration file: /usr/adm/tcheck/tw.config

You may need to adapt the configuration file to suit your installation. It is prepared for AIX 4.1, but you will have to add the SP binaries to it yourself.

When you are ready, initialize the database:

```
/usr/security/bin/tripwire -initialize
```

This will create a database in the /usr/adm/tcheck/databases directory called tw.db_<hostname>. This file may be several megabytes in size. You will need to secure this database. The problem with securing the tripwire database binary and config file is that you need a location that is secure but can be reached easily from the node. The node should not, normally, have access to this location because, if it was compromised, a hacker could alter the tripwire files. Since all nodes have root access to the CWS, it is not a safe place to store the tripwire files. You will need an external system if you want the tripwire files to be secure.
To access the trusted versions of the tripwire files, you need to access them
in a secure manner. This could be done through DFS or by using an
authenticated remote copy. NFS cannot be used because it is not secure. The
checking should be initiated from the external system, which would need
remote command authority on all nodes and the CWS. The checking of a
node could be done using the following script that uses \texttt{rcp} and \texttt{rsh} (when
DFS is used, you could simplify the script):

\begin{verbatim}
#!/bin/ksh
# Tripwire control program, options are a list of nodes.
OUT=/home/root/tripwire.out.'date +%y%m%d'
TR=/tripwire
for node in $* ; do
   /bin/echo "Starting tripwire on $node at `date`
   /bin/rcp $TR/tw.db_$node $node:/usr/adm/tcheck/databases/tw.db_$node ||
      /bin/echo "copy of database failed on $node"
   /bin/rcp $TR/tw.config $node:/usr/adm/tcheck/tw.config ||
      /bin/echo "copy of config file failed on $node"
   /bin/rcp $TR/tripwire $node:/tmp/tripwire.$$ ||
      /bin/echo "copy of tripwire binary failed on $node"
   /bin/rsh $node /tmp/tripwire.$$ -q
   /bin/rsh $node /bin/rm /tmp/tripwire.$$ -q
   /bin/rsh $node /bin/rm /usr/adm/tcheck/tw.config
   /bin/rsh $node /bin/rm /usr/adm/tcheck/databases/tw.db_$node
done 2>&1 >> $OUT
\end{verbatim}

(Figure 30. Example script to use with tripwire (runtrip.ksh))

The script copies the tripwire files to the nodes and then executes the
program. This is just a simple example. For instance, you could copy the files
back to the server and compare them to see if they have been altered. It is
always good to be a little paranoid when it comes to integrity checking.

7.6.3 SATAN

SATAN, by Dan Farmer and Wietse Venema, is the Security Administrator Tool for Analyzing Network. This tool can help you find security weaknesses
in the network configuration of your system. It probes the system from the
outside and tries to find weaknesses.

A complete description of all of SATAN’s features can be found here:
http://www.cerias.purdue.edu/coast/satan.html

A packaged version of SATAN can be found at http://www.bull.com. This
version requires \texttt{mosaic} and \texttt{fping} as prerequisites. You can delete the \texttt{PERL}
package from the .toc file because an SP node already has \texttt{perl} installed by
default.
SATAN can be very useful to find out about weaknesses in your network security. If used regularly, new weaknesses can be spotted early on.

### 7.6.4 The Deception Toolkit (DTK)

The Deception Toolkit (DTK), written by Fred Cohen, is a novel approach to intrusion detection. By faking programs on certain ports and logging any connects to them, intruders are detected early and, quite possibly, confused about the responses they are getting.

You can turn all entries in the `/etc/inetd.conf` file that are deactivated into deceptive ports, and ports that are active can be guided through the TCP Wrappers tool into a deception program while approved source addresses are handled normally.

You can download the toolkit and find more information about it on: http://all.net
When installing the COPS distribution available from [http://www.bull.com](http://www.bull.com) you will need to delete the following lines from the .toc file:

```
x0 {
  *prereq freeware.gnu.perl.rte 5.3.0.0
  *prereq freeware.perl.rte 5.4.0.0
}
```

Since `perl` is already installed, these prerequisites are no longer required. You will need to add a link in `/usr/local/bin` to point to `perl` in `/usr/bin`:

```bash
ln -s /usr/bin/perl /usr/local/bin/perl
```

The perl-version of COPS can now be run by typing `pcops` on the command line. You can find the output in the `/usr/local/lib/p-cops/<hostname>` directory.

### 7.6.6 TAMU's Tiger

Tiger is a set of scripts that scan a UNIX system looking for security problems, in the same fashion as Dan Farmer's COPS. Tiger was originally developed to provide a check of UNIX systems on the Texas A&M University after a major security breach.

Tiger was written in 1993 by Doug Schales, and last updated in 1994. It supports AIX 3.x but can be used with AIX 4.x when default scripts are used. Not all scripts will function when doing this, and some should be disabled. Some tests require compiled C code; so, a compiler should be available for those.

Tiger can be downloaded from:


The following steps should get you going:

1. Unpack the archive
2. Change to the Tiger directory and enter the following:
   ```bash
   make install
   ```
3. Edit the `/usr/local/tiger/tigerrc` file, and change following:
   ```
   -Tiger_Check_RHOSTS=N
   -Tiger_Check_PERMS=N
   -Tiger_Check_SIGNATURES=N
   -Tiger_Check_FILESYSTEM=N
   -Tiger_Check_PATH=N
   -Tiger_Check_EMBEDDED=N
   ```
4. Create the necessary subdirectories:
-mkdir -p /usr/spool/tiger/logs
-mkdir -p /usr/spool/tiger/work
-mkdir -p /usr/spool/tiger/bin

5. Run Tiger by typing: /usr/local/tiger/tiger

The output file will contain messages such as these:

```
# Performing NFS exports check...
--WARN-- [nfs011w] Unprotected directory /store is exported with root \n    access to host(s) sp3en0.
```

You can check what each message means by calling the `tigexp` command with the message ID:

```
# /usr/local/tiger/tigexp nfs011w
Exporting a file-system R/W to everyone means that anyone can modify
the data on your system, possibly making changes that allow them to
login to the system and access or destroy other files.
See 'nfs013i'
```

If you want to check for weak passwords, you will need to install the `crack` program. Tiger does check for some useful security vulnerabilities, but it also reports settings that are normal on AIX but non-standard on other versions of UNIX. We recommend running this program weekly, or, at least, monthly.

7.6.7 Log files you should check

The following logs should be checked on a daily basis:

`/var/adm/sulog`

This file contains all uses of the `su` command in the system.

For example, to see today’s switch user commands:

```
# grep "^\date +%m/%d" /var/adm/sulog
```

`/etc/security/failedlogin`

This file contains a list of all failed login attempts in the `utmp` format.

For example, to get today’s failed login attempts:
syslog daemon log files

The system log daemon (syslogd) can be configured to record messages at different levels. The messages can also be forwarded to another system. For analyzing attempts at compromising a system or analyzing a compromise, it can give invaluable information. When setting up this daemon, care should be taken that there is enough free space in the filesystem, and the log files should, preferably, be written into their own filesystem.

On the nodes, the configuration file /etc/syslog.conf forwards the messages. In this example, we forward them to a system called sec-host:

```
# tail -3 /etc/syslog.conf
daemon.notice /var/adm/SPlogs/SPdaemon.log
*.crit @sec-host
*.debug;mail.none; @sec-host
```

One system sec-host the incoming messages are redirected to a log file:

```
# tail -2 /etc/syslog.conf
*.crit /var/syslog/syslog.crit
*.debug;mail.none; /var/syslog/syslog.debug
```

The /var/syslog/syslog.debug file on sec-host will probably grow quite fast. You will need to prune that file regularly.
Part 4. Troubleshooting
Chapter 8. Problem isolation and resolution

The security services of PSSP have been designed and implemented to minimize the complexities that are often associated with security features, especially during their installation and configuration phases.

However, due to the nature of security - both its pervasiveness and global scope - conditions may arise that may, at first, appear to be "security" problems but are, in many cases, actually, problems with environmental conditions that affect security services or problems with the interactions of software components and their related or dependent security services.

This chapter provides examples of some common problems and their corrective actions, as well some general troubleshooting tips and problem determination techniques as they relate to PSSP security services, especially in conjunction with DCE.

While the majority of the items in this chapter focus on PSSP and DCE, several of the techniques can be generalized and applied to PSSP’s compatibility mode, PSSP Kerberos Version 4, and other security software that may be part of your SP system.

Note, however, that this chapter is not intended to be an exhaustive examination of root cause analysis and corrective actions relating to general security problems or customized security considerations. The wealth of existing information technology literature is abundant with white papers, articles, and texts covering this type of material. Plus, there is a growing number of Web sites dedicated to the broad spectrum of computer security. Also, this chapter is not meant to replace the problem determination procedures established at a site or to circumvent its established problem reporting process(es).

The reader is encouraged to use the aforementioned items, along with product-specific problem determination guides, such as DCE’s Problem Determination Guide, software diagnostic guides, and other problem determination and isolation techniques/materials when analyzing a problem event.
There are several possible periods during which problems/issues might arise for PSSP security services:

- **During installation and configuration of security services**
  
  During installation and configuration of security services, dependencies between various installation and configuration steps (and their processes) and any requirements for additional software installation and configuration, such as those associated with DCE, problems/issues are often due to a lack of planning for the security services required, bypassed steps during installation and configuration, and/or run time process failure.

  The PSSP Planning Guide covers considerations and requirements needed prior to installing and configuring PSSP security services, while the Migration and Installation Guide provides a step-by-step approach to establishing the planned security settings.

  Installation and configuration of security services is the most likely place for problems/issues, if there are any, to arise.

  Section 8.1, “PSSP security installation and configuration” on page 240, provides a list of problem determination and isolation items for PSSP security installation and configuration procedures.

- **Events that may occur during normal run time, that is, steady-state events after security services installation and configuration**

  Once PSSP is running with a security configuration, the problems/issues that may occur will, most likely, be due to DCE servers (client or master) not being available, file systems that are full, a daemon that hangs, or a user that does not have the proper authentication and/or authorization to access a requested resource.
These things may occur as out-of-the-ordinary anomalous events, or as the result of a process failure.

DCE and PSSP diagnostic guides, administrative guides, and command references should be consulted under these scenarios.

Several of these problems are examined in expanded detail in Section 8.2, “Steady-state run time” on page 244, and from Section 8.5 to the end of the chapter.

- **A system is repartitioned, a node is reinstalled, and a node is added or deleted from an existing system**

  Partitioning, re-installing a node, or adding/deleting a node from an existing system requires several additional steps that take into account the security services of PSSP.

  These are scenarios that take place on an irregular basis, and problems/issues are often the result of a process failure, misconfiguration, or lack of planning for changes in the security environment.

  DCE and PSSP diagnostic guides, administrative guides, and command references should be consulted under these scenarios.

  The items presented in Section 8.1, “PSSP security installation and configuration” on page 240, apply to these scenarios.

- **The DCE cell is reconfigured**

  If the DCE cell into which PSSP services is reconfigured (for whatever reasons), then the following will be required:

  - Disabling and/or enabling PSSP and AIX authentication methods
  - Cleaning up existing DCE-related files used by PSSP (and originally created by PSSP installation and configuration)
  - Rerunning PSSP installation and configuration scripts relating to DCE security services

  This is the least likely scenario. Reconfiguring a DCE cell would not only impact PSSP security services, but any other software component dependent upon DCE.

  Section 8.3, “DCE cell reconfiguration impacts” on page 353, provides a procedure to recover PSSP DCE security services when a DCE cell is reconfigured.

- **DCE Security and CDS servers cannot be reached or contacted (includes master and replica servers)**

  When PSSP security services is running in a dce/k5 only mode and all DCE Security and CDS servers, including replica servers, cannot be
reached or contacted, PSSP will hang and/or generate authentication and authorization failures. In this type of environment, PSSP processing effectively comes to a halt because the security services protecting PSSP resources cannot be satisfied.

DCE network connectivity must be restored. If DCE network connectivity cannot be restored, or the problem turns out to be a catastrophic failure with all hosts running DCE servers, it must be determined if the site should run its SP system without dce/k5 PSSP security.

When PSSP security services is running in a dce:compat/k5:k4:std mode, or in a compat/k5:k4:std mode, and all DCE Security and CDS servers, including replica servers, cannot be reached or contacted, PSSP will hang and/or generate authentication and authorization failures for the dce mode and then fail-over to compat and k4:std processing. In this type of environment, PSSP performance is impacted because dce-related security calls will time-out due to DCE server unavailability. However, PSSP processing will continue under compat mode (once the dce calls time-out).

DCE network connectivity must be restored. If DCE network connectivity cannot be restored, or the problem turns out to be a catastrophic failure with all hosts running DCE servers, it must be determined if the site should run its SP system with only compat/k4:std security.

These are the least likely scenarios. The unavailability of DCE servers would not only impact PSSP security services, but any other software component dependent upon DCE.

Section 8.4, “Temporarily disabling dce/k5 when DCE is not available” on page 363, provides a general procedure to disable the dce and/or k5 methods from PSSP use when DCE Security and CDS servers cannot be reached.

8.1 PSSP security installation and configuration

It is imperative that the IBM RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment, GA22-7281, be used prior to any installation of PSSP Version 3.2 (this includes both migration and new installs) to determine which security configuration is right for a site and identify any prerequisites required by the selected security configuration.

Given that there are several possible PSSP security configurations available in an SP environment, and that, within a security configuration, there are PSSP and AIX security settings, choosing the appropriate combinations of security values for a configuration, having the selected security mechanisms
installed and configured, populating the SDR and, possibly, DCE with needed values, and then enabling the SP system to use the security settings presents new considerations and requirements for PSSP administrators.

Strict adherence to the PSSP Installation and Migration Guide for security installation and configuration will take PSSP administrators through the necessary steps to ensure that needed security settings and scripts are executed in the required order.

The list to follow represents common problem areas related to installation and configuration.

### 8.1.1 Scripts setupdce or config_spsec did not run successfully

Probable cause(s) for the setupdce or config_spsec scripts not running successfully include the following:

- DCE daemons are not running on the control workstation. Use the DCE command, `show.cfg`, to display the status of DCE.

- DCE is configured on the control workstation but is not at the level required for PSSP DCE security services. Use the following AIX command to see information for all the installed DCE filesets on the host:
  
  `lslpp -l "*dce*"

- The DCE master servers (Security and CDS) are not running or are not reachable. There are several DCE and AIX commands that can be used to determine the state of DCE services and network connectivity. DCE commands, such as `show.cfg`, `dcecp -c cell ping`, `dcecp -c cdsli -c -C high`, and `dce_login`, are useful for determining DCE server status. AIX commands, such as `netstat -m`, `ping (with -R)`, `traceroute`, and `no -a | grep ipf (check for ipforwarding)`, are useful for determining network connectivity between points. Run the AIX command, `ps -ef | more`, and look for the existence of DCE daemons.

- The Security and/or CDS server data specified on the PSSP script command lines contained a typo.

- The DCE cell administrator password is not valid.

- The SDR does not contain hostnames for the nodes.

- Write operations to the SDR fail.

- The filesystems where the DCE Security and CDS databases are stored are full.

- The DCE Security servers are in read-only mode and cannot be updated.
The following is/are the corrective action(s):

1. Start the DCE daemons on the control workstation.
2. Upgrade DCE to the required level for PSSP DCE security services.

3. Start the DCE master servers and/or restore network connectivity between the control workstation and the hosts running the DCE master servers.
4. Obtain the correct names of the DCE servers, and enter this information via PSSP scripts, where appropriate.
5. Specify the correct DCE cell administrator password.
6. Rerun PSSP installation procedures that populate the SDR with hostnames for the nodes.
7. Root authority on the control workstation is required to write to the SDR.
8. On the host(s) running the DCE Security and CDS servers, alleviate the file system full state so that the DCE databases can be updated.
9. On a host already configured for DCE and logged in as a DCE cell administrator, enable the Security server for write operations. The following example represents a DCE cell where only one Security server is running. In cells with more than one Security server, the server must be explicitly named on the DCE registry enable command.

Note

Once corrective actions are complete, the scripts that failed must be rerun.

Note

Before upgrading DCE to a new level, refer to DCE's documentation on recommended upgrade procedures. This will ensure that existing DCE services not required by PSSP services, such as DFS and/or DTS, are properly upgraded.
8.1.2 Authentication methods cannot be enabled on a node

If the PSSP trusted services authentication method or AIX authentication methods cannot be enabled on a node, the probable cause(s) are:

- The SDR security attributes are not correct.
- Required PSSP and/or DCE filesets are not installed.
- Kerberos Version 4 and/or DCE is not configured properly.
- PSSP DCE keyfiles were not created on a host.

The following is/are the corrective action(s):

---

**Note**

Once corrective actions are complete, the scripts that failed must be rerun.

1. Verify that SDR security attributes for the partition in which the node resides contain the correct values. If not, use the PSSP `chauthpts` and `chauthpar` commands, with their `-c` flags, to alter the SDR values. Likewise, use `spsetauth` to set other security values in the SDR.

2. If DCE was selected to be installed, configured, and started on a node, but was not successfully installed, configured, and started on a node, verify that the control workstation's `/spdata/sys1/install/{default}/lppsource` directory contains the correct level of IBM AIX DCE filesets.

---

**Note**

{default} is replaced with a value chosen by the customer at installation and configuration time. Refer to the *IBM RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment*, GA22-7281, and *PSSP: Installation and Migration Guide*, GA22-7347 for DCE level requirements.
3. Examine PSSP and DCE logs on the node for additional information. For PSSP errors, refer to the *PSSP Diagnosis Guide*, GA22-7350, for corrective measures. For DCE errors, refer to the *DCE Problem Determination Guide*.

4. Ensure that DCE is configured and running, and then rerun node customization, or, manually, run `spauthconfig` on the node.

The DCE installp process might generate a failed return code due to a pre-req failure or a back-leveled DCE fileset existing in the same directory as up-to-date DCE filesets, but the DCE client itself may still install and configure successfully. (A failure in the installp process does not necessarily mean that the DCE client configuration process will not complete.) However, since the installp generated a failed return code, this will be captured by PSSP installation and configuration scripts, and PSSP security configuration will exit before PSSP DCE keyfiles are created and before the dce trusted service method is enabled on the node. Provided that the DCE client configured successfully and that DCE is running on the node, simply rerunning spauthconfig on the node (as root) will complete the security configuration process for the node.

The following is/are the additional corrective action(s):

1. Ensure that `setupdce` and `config_spsec` ran successfully on the control workstation. If these scripts did not run successfully, a node set to run with dce as a PSSP trusted service authentication method cannot be configured properly.

2. Ensure that there is enough filespace in the system for DCE to be installed and configured.

3. Ensure that the SDR is reachable and can be read from the node. PSSP security installation and configuration scripts attempt to set values on the node based on values read from the SDR.

4. Ensure that the node has network connectivity to the host or hosts running the DCE Security and CDS servers. DCE client configuration on a node must update entries in these servers.

### 8.2 Steady-state run time

The examples in this section are not meant to be exhaustive. Rather, they provide specific PSSP and AIX examples of security concepts that can be applied to various areas within an SP environment. They are, for all intents and purposes, representative samples.
Problems during steady state operation normally fall into three general areas; the first two are the most likely impact points:

- Disjoint security settings (authentication methods) between nodes or between the control workstation and nodes
- User (SP administrator and non-administrator) identities and access controls
- "Other" events, which include anomalous events, such as a hung daemon, all services appear to be hung, low paging space, a file system that reaches its storage capacity, or a severe time skew between hosts

8.2.1 Disjoint security settings (authentication methods)

This section covers disjoint security settings (authentication methods) between nodes, or between the control workstation and nodes.

Beginning with PSSP V3.2, security controls need to be viewed in two flavors:

- AIX Authentication Methods (for Remote Commands)
  - AIX level (operating system)
    - Methods: k5 (Kerberos 5), k4 (Kerberos 4), std (Standard AIX)

  An SP can be configured and enabled for one or more combinations of AIX authentication methods, but at least one AIX authentication method must be enabled in the SP (including the control workstation)

  - Used directly by AIX remote commands (rsh, rcp, telnet, ftp, rlogin) for client/server authentication and authorization.
  - AIX methods are known only to AIX remote command services
  - AIX methods are used indirectly by PSSP remote parallel commands (dsh, pcp, pexec, pexecr, and others). PSSP remote commands invoke rsh and/or rcp
    - The AIX commands, chauthent and lsauthent, are used to manage the methods on a per-host basis. chauthent requires root authority. lsauthent is available to all users.
    - The PSSP commands, chauthpar and lsauthpar, are used to manage the methods on a per-partition boundary. chauthpar requires root authority and authorization to write to the SDR. lsauthpar is available to all users.

- PSSP Authentication Methods (for Trusted Services)
  - PSSP level (PSSP is an LPP on AIX).
  - PSSP Authentication Methods: dce, compat.
An SP can be configured and enabled for one, both, or none of the methods.

- Used directly by PSSP Trusted Services (SDR, Hardmon, Sysctl, and others) for client/server authentication and authorization.

- PSSP methods are known only to PSSP Trusted Services.

- The PSSP commands, `chauthts` and `lsauthts`, are used to manage the methods on a per-host basis. `chauthts` requires root authority. `lsauthts` is available to all users.

- The PSSP commands, `chauthpts` and `lsauthpts`, are used to manage the methods on a per-partition boundary. `chauthpts` requires root authority and authorization to write to the SDR. `lsauthpts` is available to all users.

The AIX and PSSP authentication method sets exist and are managed independently. Likewise, the way in which they influence the behavior of their respective client/server applications is independent of one another. The only "relationship" between the two is established by the overall PSSP security design, as follows:

When a PSSP authentication method is enabled for use by Trusted Services, the corresponding authentication method must also be enabled for the AIX remote commands.

Examples:

If the PSSP authentication methods include dce, the AIX authentication methods must include at least k5. (A situation where PSSP's dce method is enabled, but AIX's k5 method is not enabled, is precluded.)

If the PSSP authentication methods include compat, the AIX authentication methods must include at least k4. (A situation where PSSP's compat method is enabled, but AIX's k4 method is not enabled, is precluded.)

If the PSSP authentication methods include dce and compat, the AIX authentication methods must include at least k5 and k4. The following variations would be valid: dce:compat/k5:k4 and dce:compat/k5:k4:std. The following variations would not be valid: dce:compat/k5, dce:compat/k4, dce:compat/k5:std, and dce:compat/k4:std.

8.2.1.1 PSSP authentication methods

Used by the PSSP code only. These methods are not used, or known, to the operating system. PSSP authentication methods are not used for remote command processing of any kind.
Partition level settings
1. methods = dce, compat, "none" (where "none" means that no PSSP authentication method is set).
2. SDR attribute value = ts_auth_methods.
3. SDR class value = Syspar.
4. To query the PSSP authentication methods stored in the SDR, for all partitions in the system, issue: splstdata-p.

Local host settings
1. methods = dce, compat, "none," dce:compat (where "none" means that no PSSP authentication method is set).
2. Methods are stored in a file on the local host in /spdata/sys1/spsec/auth_methods.

Failover flow
The precedence ordering of the PSSP authentication methods is dce:compat, or {} in the case of no PSSP authentication methods. When dce processing fails and the compat method is enabled, compat processing is tried. When compat fails, a PSSP service will not process requests.

In the case of a single authentication method, such as dce, when dce processing fails, the service will reject the request due to authentication failure. There is no fail-over.

Querying the PSSP methods
- Partition Level - 1sauthpts (PSSP command)
- Local Host - 1sauthts (PSSP command)

About the PSSP methods
The dce method uses DCE authentication and authorization, implemented through the PSSP library, libspsec.a.

compat method means that a service is to use, or behave, as it did in releases prior to PSSP v3.2, with respect to security processing.

The following are service examples of security processing policies for dce and compat:
- SDR
  - dce - DCE credentials and DCE group membership(s)
  - compat - requires the user to be root
- Sysctl
dce - DCE credentials and DCE ACLs
compat - Kerberos V4 credentials and Kerberos V4 ACLs

• Hardmon
dce - DCE credentials, DCE ACLs, and DCE group membership(s)
compat - Kerberos V4 credentials and Kerberos V4 ACLs

8.2.1.2 AIX authentication methods
Used directly by AIX remote commands, such as rcp, rsh, and so on, and indirectly by PSSP remote parallel commands that exploit AIX remote commands, such as dsh and pcp, to achieve their functionality.

Partition level settings
1. methods = k5, k4, std, "none" (where "none" means that no authentication method is set; note that the SP requires and enforces that at least one AIX authentication method be enabled)
2. SDR attribute value = auth_methods
3. SDR class value = Syspar
4. To query the AIX authentication methods stored in the SDR, for all partitions in the system, issue: splstdata -p

Local host settings
1. methods = k5, k4, std, "none" (where "none" means that no authentication method is set; note that the SP forces at least one AIX authentication method to be set).
2. Methods are stored on the local host in the ODM, in the file CuAt, under /etc/objrepos. (CuAt is also the name of the object class.)

Failover flow
The precedence ordering of the AIX authentication methods is the SP enforced order of k5:k4:std. When k5 processing fails, k4 processing is attempted. When k4 fails, std processing is attempted. When std fails, no further attempts are made at completing the remote command request.

In the case where two authentication methods are set, such as k5:k4, when k5 fails, k4 is attempted. When k4 fails, no further attempts are made to complete the remote command request.

The same holds true for the case where a single authentication method is set, such as k4. When k4 fails, no further attempts are made at completing the remote command request.
Querying the AIX methods

- Partition Level - lsauthpar (PSSP command)
- Local Host - lsauthent (AIX command)

About the AIX methods

The k5 method uses the Kerberos Version 5 protocol, implemented through Kerberos Version 5 libraries supplied by DCE and AIX, and DCE credentials, which are upgraded to Kerberos Version 5 format for use with remote commands.

The $HOME/.k5login file is used for authorization when k5 is enabled.

The k4 method uses the Kerberos V4 protocol, implemented through a Kerberos V4 compatibility library (supplied by and shipped only with PSSP), and Kerberos V4 credentials. The Kerberos V4 compatibility library is /usr/lib/libspk4rcmd.a.

The $HOME/.klogin file is used for authorization when k4 is enabled.

Note

Only the AIX remote commands, rsh and rcp, use Kerberos V4 processing. All other AIX remote commands DO NOT use the k4 method. (They simply ignore it.) PSSP remote commands, such as dsh and pcp, use the k4 method when enabled because these commands exploit rsh and rcp.

std method uses IP-based "authentication," implemented through the operating system. There are no credentials or tickets, such as those used with k5 and k4.

The $HOME/.rhosts file is used for authorization when std is enabled.

"none" means that all AIX remote commands are disabled and cannot be used. This, in turn, means that those PSSP remote commands that implement their functionality through the exploitation of AIX remote commands are, effectively, "disabled."

AIX and PSSP remote commands

- AIX remote commands:
  - rsh
  - rcp
  - telnet
- ftp
- rlogin

- PSSP remote commands:
  Items in parentheses denote other command dependencies.
  - dsh (rsh)
  - pcps (dsh; rsh; rcp)
  - pexec (dsh; rsh)
  - pexec (dsh; rsh)
  - p-cat (dsh; rsh)
  - pfind (dsh; rsh)
  - pls (dsh; rsh)
  - pmv (dsh; rsh)
  - ppred (dsh; rsh)
  - pps (dsh; rsh)
  - pmp (dsh; rsh)

8.2.1.3 Disjoint PSSP authentication methods
PSSP Trusted Service client requests will fail when the client and the server do not share at least one PSSP authentication method in common. For example, if the client is running in a dce only mode, then the server must be running in a mode that includes dce, as in dce (only) or dce:compat. An example of a disjoint set of authentication methods between a client and a server is when the client is running in a compat only mode and the server is running in a dce only mode.

A disjoint set of PSSP authentication methods occurs under the following conditions:

1. In a multi-partition environment, where each partition is running a different security mode

2. In a multi- or single partition environment where the PSSP authentication methods on a host (node or the control workstation) are changed to something other than the partition or system-wide security settings

There are two ways to query the authentication methods in the partition to determine if methods are disjoint.
The first way is to use the `lsauthpts -v` command to compare the current partition's SDR values against each of the nodes in the partition. (In a one partition system, this is equivalent to testing the entire system.) `lsauthpts` automatically reads, from the SDR, the authentication methods for the partition and compares a node’s values to the SDR values. (When the partition name is not specified by the issuer, the current partition is assumed.) For each host with a disjoint set of methods, a "discrepancy" message is displayed.

```
sp3en0/ # lsauthpts -v
  Trusted services authentication methods for the partition: dce:compat
  No discrepancies were found.
```

```
sp3en0/ # lsauthpts -v
  Trusted services authentication methods for the partition: dce:compat
  lsauthpts: 0016-321 On sp3en06 the trusted services authentication methods are
  incorrectly set to "dce".
```

The second way is to use the `dsh -a` command to query the local authentication methods on each host in the current partition and pipe the results to `dshbak -c`. (dsh -avG will work over all nodes in the system, regardless of their partition association.) The `dshbak` program collapses identical output from more than one host so that identical results are displayed only once. Unlike the `lsauthpts -v` command, the administrator must know what the authentication methods are for the current partition and visually compare the node results to this value.

Further, if environmental variables that control the set of nodes over which dsh will operate (WCOLL, SP_NAME) include nodes that span partitions, the results will indicate a discrepancy where one does not actually exist. Conversely, if the environment variables include a subset of nodes within a partition, the results will not indicate a true view of the entire partition.

The following example uses `dsh` to display the authentication methods on a set of nodes. However, the `dsh` WCOLL variable points to a file containing only a subset of nodes within the system. In contrast, the `splstnodes` command reveals all the node names in the system.
Knowing exactly which nodes within the system (or within a partition) are disjoint (if any) with respect to PSSP authentication methods is critical for normal security operations. Using the `lsauthpts -v` command is the easiest and most reliable way of gathering that information. Corrective actions can then be taken depending on the type of disjoint environment.

1) Multi-partition environment, each running a different security mode

There is no corrective action for this case, in the sense of correcting a problem, since it's a PSSP requirement that, for two of its Trusted Services to communicate, they must share at least one PSSP authentication method in common. (This is equivalent to humans speaking a common language in order to communicate verbally.) However, multiple partitions don't have to be enabled for the same set of authentication methods, if that is what is desired in the system.

The example set for a multi-partition environment assumes three partitions, sp3en0, sp3sp1, and sp3sp2, where each partition is enabled for a different set of PSSP authentication methods.

In the first example, the commands are issued from the control workstation out to the nodes, and all attempts are successful. The control workstation contains the union of all the PSSP authentication methods in the SP, given

```
sp3en0/ # env WCOLL
/th-node-group
sp3en0/ cat /3-node-group
sp3n01
sp3n06
sp3n09
sp3en0/ dsh -a "lsauthpts" | dshbak -c
HOSTS
sp3n06
------------------------------
DCE
HOSTS
sp3n01 sp3n09
------------------------------
DCE
Compatibility
sp3en0/ splstnodes -s reliable_hostname reliable_hostname
reliable_hostname
sp3n01
sp3n05
...
sp3n14
sp3n15
```
that the control workstation must be able to communicate with each node in
the system.

In the second set of examples, the commands are issued from a node in one
partition to nodes in other partitions. Some of the requests are successful
because a common PSSP authentication method is shared, while other
requests are not successful because a method is not shared.

For the scope of these examples, the following partition security methods
(including PSSP authentication methods), along with a node name in each
partition, were in place. The data was extracted from the splstdata -p
command.

```
Syspar: sp3en0
------------------------
  ts_auth_methods dce:compat
  auth_methods  k5:k4:std
  node          sp3n01

Syspar: sp3sp1
------------------------
  ts_auth_methods compat
  auth_methods  k4:std
  node          sp3n04

Syspar: sp3sp2
------------------------
  ts_auth_methods dce
  auth_methods  k5:std
  node          sp3n03
```

Control workstation to nodes example. dce:compat to compat, dce:compat,
and dce nodes.

The Sysctl client request from the dce:compat control workstation to nodes at
various authentication method settings is successful because the client
shares at least one method in common with each host.

Security configuration and user's credential identities.

```
sp3en0/ # lsauths
DCE Compatibility
  Global Principal: /.../sp_cell/hosts/sp3en0/self
  Principal:     root.admin@ITSO.IBM.COM
```

Client request to multiple nodes.
Node to node example #1. compat to dce node and compat node

The Sysctl client request from a compat node to a compat node is successful, while the compat to dce node request is denied.

```
sp3n04/ # sysctl -h sp3n01 -h sp3n04 -h sp3n03 whoami -v
>> sp3n04
DCE:
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
>> sp3n01
DCE: /.../sp_cell/hosts/sp3en0/self
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
>> sp3n03
DCE: /.../sp_cell/hosts/sp3en0/self
K4: root
AIX: root
<<
```

Node to node example #2. dce to dce node and compat node

The Sysctl client request from a dce node to a dce node is successful, while the dce node to compat node request is denied.

```
sp3n04/ # lsauthts
Compatibility
sp3n04/ # k4list | grep pal:
Principal: root.admin@ITSO.IBM.COM
sp3n04/ # sysctl -h sp3n01 -h sp3n03 whoami -v
>> sp3n03
<<
>> sp3n01
DCE:
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
```
To establish communication between the client and server, the security settings of one (or both) partitions must be changed to include a common method. This is accomplished through PSSP administrative commands that either enable or disable PSSP authentication methods.

Remove an authentication method from a partition. (The change is reflected in the SDR and on the nodes.)

```
sp3n03/ # lsauthpts ; klist | grep lob ; k4list | grep pal:
  DCE
    Global Principal: /.../sp_cell/hosts/sp3n03/self
sp3n03/ # sysctl -h sp3n04 -h sp3n01 whoami -v
  >> sp3n01
    DCE: /.../sp_cell/hosts/sp3n03/self
    K4: 
    AIX: root
  <<
  >> sp3n04
  <<
```

Add an authentication method to a partition. (The change is reflected in the SDR and on the nodes.)

```
sp3en0/ # lsauthpts -p sp3en0
  DCE Compatibility
sp3en0/ # chauthpts -p sp3en0 dce
  0513-095 The request for subsystem refresh was completed successfully.
sp3en0/ # lsauthpts -p sp3en0
  DCE Compatibility
```

Attempt to add an authentication method to a partition, but the required software to support the method is not installed.

```
sp3en0/ # lsauthpts -p sp3en0
  Compatibility
sp3en0/ # chauthpts -p sp3en0 dce compat
  0513-095 The request for subsystem refresh was completed successfully.
sp3en0/ # lsauthpts -p sp3en0
  DCE Compatibility
```

Chapter 8. Problem isolation and resolution  255
The example that enables authentication methods assumes that the required security software is already installed in the needed partition. Note, however, that there are no restrictions when disabling already set authentication methods. In an environment where the software is not already installed for the partition, the attempt to set PSSP authentication methods for the partition will fail, as shown. In the event of such a failure, refer to the PSSP Administration guide for the steps needed to install the required security code on the nodes, then re-run the command. Also note that chauthpts automatically refreshes certain PSSP subsystems as part of the PSSP authentication methods enablement.

Likewise, in a coexistence environment where SP nodes include Version 3.2 and pre-3.2 PSSP levels, attempting to disable a PSSP authentication method that is required to be compatible with previous levels of PSSP is not permitted, as shown here:

```
sp3en0/ # lsauthpts -p sp3sp2
DCE
sp3en0/ # chauthpts -p sp3sp2 dce compat
chauthpts: 0016-349 You cannot enable compat, because the partition has not been configured for Kerberos V4 use.
sp3en0/ # lsauthpts -p sp3sp2
DCE
```

2. Methods do not match system or partition level values

In the case of disjoint PSSP authentication methods being on one host only, the corrective action is for the SP administrator to bring that host's PSSP authentication methods in line with the partition settings. This is accomplished through PSSP administrative commands in one of three ways:

1. Issue chauthpts -f {methods list} from the control workstation:
2. `rsh` the command `chauthts {methods list}` to the host:

```
sp3en0/ # lsauthpts
DCE
sp3en0/ # rsh sp3n05 lsauthpts
compat
sp3en0/ # chauthts -f doe compat
0513-095 The request for subsystem refresh was completed successfully.
sp3en0/ # rsh sp3n05 lsauthpts
DCE
compat
```

3. Log in to the node and issue the `chauthts {methods list}` locally

   Except for logging in to the node, the PSSP commands needed to set the
   PSSP authentication methods locally is shown in the `rsh` portion of the
   previous example.

   The first approach is the preferred synchronization method, as it
   automatically refreshes certain PSSP subsystems as part of the PSSP
   authentication methods enablement. The other approaches do not provide
   this automation and should be used only under conditions where `chauthpts`
   fails.

   However, should the `chauthpts` fail, the chances that the `rsh` approach will
   work are low, given that `chauthpts` exploits the PSSP `dsh` command, which, in
   turn, exploits `rsh`. If `chauthpts` fails due to an `rsh` failure, then the `rsh`
   alone will most likely fail. (Issuing an individual `rsh` command to each host that
   `chauthpts` will operate over is an excellent way to help identify any problems
   with the state of AIX authentication method settings within the system.)
   `chauthpts` failing because it cannot write/contact an SDR is due to a stopped
   SDR daemon, hung SDR daemon, invalid partition name on the `chauthpts`
   command, or an invalid SDR hostname in the PSSP environment variable
   `SP_NAME`. These errors are clearly identified by `chauthpts`.

   SDR daemon is not running.
Attempt to enable authentication methods for a non-existent partition.

When neither chauthpts nor rsh/chauthts is successful, logging in to the node and changing the methods locally is the only course of synchronization action, short of rebooting the node, and is less disruptive than rebooting a node. (Node log in may take the form of telnet, rlogin, s1term (in write mode), ssh, or other processes that open a terminal session to a node.)

Given that chauthpts ultimately issues a chauthts on hosts for which the underlying rsh was successful, chauthpts and chauthts will experience a failure (will not set methods as requested) under the following conditions:

- /sptdata is full (the file /spdata/sys1/spsec/auth_methods cannot be updated).
- The required software for a method is not installed on the host.
- The order of the methods is incorrect, or the methods are invalid.
8.2.1.4 Disjoint AIX authentication methods

Like the PSSP authentication methods, the AIX authentication methods for remote commands (both AIX and PSSP remote commands) must also share at least one method in common if the client and server are to communicate. In fact, the PSSP authentication model, as well as its authentication management commands, were modeled after the AIX authentication methods and management commands.

Remote command client requests will fail when the client and the server do not share at least one AIX authentication method in common. For example, if the client is running in a k5 only mode, the server must be running in a mode that includes k5, as in k5 (only), k5:k4, or k5:k4:std. An example of a disjoint set of authentication methods between a client and a server is when the client is running in a k4 only mode and the server is running in a k5 only mode.

A disjoint set of AIX authentication methods occurs under the following conditions:

1. In a multi-partition environment, where each partition is running a different security mode
2. In a single partition environment, where the AIX authentication methods on a host (node or the control workstation) are changed to something other than the system-wide security settings

There are two ways to query the authentication methods in the partition to determine if methods are disjoint.

The first is to use the `lsauthpar -v` command to compare the current partition’s SDR values against each of the nodes in the partition. (In a one partition system, this is equivalent to testing the entire system.) `lsauthpar` automatically reads from the SDR the authentication methods for the partition and compares a node’s values to the SDR values. (When the partition name is not specified by the issuer, the current partition is assumed.) For each host with a disjoint set of methods, a "discrepancy" message is displayed.
The second way is to use the `dsh -a` command to query the local authentication methods on each host in the current partition and pipe the results to `dshbak -c`. (`dsh -avG` will work over all nodes in the system, regardless of their partition association.) The `dshbak` program collapses identical output from more than one host so that identical results are displayed only once. Unlike the `lsauthpar -v` command, the administrator must know what the authentication methods are for the current partition and visually compare the node results to this value. Furthermore, if environmental variables that control the set of nodes over which `dsh` will operate (WCOLL, SP_NAME) include nodes that span partitions, the results will indicate a discrepancy where one does not actually exist. Conversely, if the environment variables include a subset of nodes within a partition, the results will not indicate a true view of the entire partition.

Using `dsh` to display the authentication methods on a set of node. The `dsh` WCOLL variable is set and points to a file containing only a subset of nodes within the system. `sp1stnodes` reveals all the node names in the system.
Knowing exactly which nodes within the system, or within a partition, are disjoint (if any) with respect to AIX authentication methods is critical for normal security operations. Using the `lsauthpar -v` command is the easiest and most reliable way of gathering that information. Corrective actions can then be taken depending on the type of disjoint environment:

1. Multi-partition environment, each running a different security mode

There is no corrective action for this case, in the sense of correcting a problem, since it's an AIX requirement that, for its client/server processes to communicate, they must share at least one AIX authentication method in common. (This is equivalent to humans speaking a common language in order to communicate verbally.) However, multiple partitions don't have to be enabled for the same set of authentication methods, if that is what is desired in the system.

The example set for a multi-partition environment assumes three partitions, `sp3en0`, `sp3sp1`, and `sp3sp2`, where each partition is enabled for a different set of AIX authentication methods.
In the first example, the commands are issued from the control workstation out to the nodes, and all attempts are successful. The control workstation contains the union of all the AIX authentication methods in the SP, given that the control workstation must be able to communicate with each node in the system. Note that the control workstation can contain more AIX authentication methods beyond the union of the AIX authentication methods of the SP. For example, if the union of all AIX authentication methods in the SP is k5:k4, the control workstation can have k5:k4:std, with k5:k4 being the minimum set of methods. This allows the SP administrator to tailor which remote command requests are satisfied by the control workstation by host requests external-to-the-SP, in environments where such remote access to the control workstation is required.

In the second set of examples, the commands are issued from a node in one partition to nodes in other partitions. Some of the requests are successful because a common AIX authentication method is shared while other requests are not successful because a method is not shared.

For the scope of these examples, the following partition security methods (including AIX authentication methods), along with a node name in each partition, were in place. The data was extracted from the `splstdata -p` command.

```
Syspar: sp3en0
----------------------------------
ts_auth_methods dce:compat
auth_methods k5:k4
node sp3n01

Syspar: sp3sp1
----------------------------------
ts_auth_methods compat
auth_methods k4
node sp3n04

Syspar: sp3sp2
----------------------------------
ts_auth_methods dce
auth_methods k5
node sp3n03
```

**Control workstation to nodes example. k5:k4 to k4, k5:k4, and k5 nodes**

The rsh client request from the k5:k4 control workstation to nodes at various authentication method settings is successful because the client shares at least one method in common with each host. Error messages returned by the
remote daemon are for failed remote command protocol attempts and are not failures of the commands actually issued on the remote hosts.

Security configuration and user's credential identities.

```
sp3en0/ # lsauthent ; klist | grep lob ; k4list | grep pal:
Kerberos 5
Kerberos 4
  Global Principal: /.../sp_cell/hosts/sp3en0/self
  Principal: root.admin@ITSO.IBM.COM
```

rsh to a k5:k4 enabled node.

```
sp3en0/ # rsh sp3n01 "hostname -s ; lsauthent"
sp3n01
Kerberos 5
Kerberos 4
```

rsh to a k4 enabled node.

```
sp3en0/ # rsh sp3n04 "hostname -s ; lsauthent"
krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3n04
Kerberos 4
```

rsh to a k5 enabled node.

```
sp3en0/ # rsh sp3n03 "hostname -s ; lsauthent"
sp3n03
Kerberos 5
```

The protocol error message going from k5:k4 to k4 is expected, given that the client tried to communicate with the remote server via k5 first. Once that attempt failed, the client and server communicated via k4.

**Node to node example #1. k5:k4 to k5 node and k4 node**

The rsh client request from a k5:k4 node to a k4 node is successful, as is the k5:k4 to k5 node request. Error messages returned by the remote daemon are for failed remote command protocol attempts only. The commands to be run on the remote host were ultimately executed because a common authentication method between client and server was found.
The protocol error message going from k5:k4 to k4 is expected since the client tried to communicate with the remote server via k5 first. Once the k5 attempt failed, the client and server communicated via k4.

Node-to-node example #2. k4 to k5 node and k5:k4 node

The rsh client request from a k4 node to a k4 node is successful, while the k4 to k5 node request is denied. For failed rsh attempts, error messages returned by the remote daemon are for failed remote command protocol attempts. The commands to be run on the remote host were never executed, due to the disjoint authentication methods between client and server.

The protocol error message going from k4 to k5 is expected since the client and server do not share a common method. The commands to be run on the remote host, `hostname -s` and `lsauthent`, are not attempted.

The protocol error message going from k4 to k5:k4 is expected since the client only tried to communicate with the remote server via k4.

Node to node example #3. k5 to k5:k4 node and k4 node.

The rsh client request from a k5 node to a k5:k4 node is successful, while the k5 to k4 node request is denied. For failed rsh attempts, error messages
returned by the remote daemon are for failed remote command protocol attempts. The commands to be run on the remote host were never executed due to the disjoint authentication methods between client and server.

The protocol error message going from k5 to k4 is expected since the client and server do not share a common method. The commands to be run on the remote host, `hostname -s` and `lsauthent`, are not attempted.

To establish communication between the client and server, the security settings of one (or both) partitions must be changed to include a common method. This is accomplished through PSSP administrative commands that either enable or disable AIX authentication methods.

Removing an authentication method from a partition (The change is reflected in the SDR and on the nodes.):

```
sp3n03/ # lsauthent ; klist | grep lob
Kerberos 5
    Global Principal: /.../sp_cell/hosts/sp3n01/self
sp3n03/ # rsh sp3n01 "hostname -s ; lsauthent"
sp3n01
Kerberos 5
Kerberos 4
sp3n03/ # rsh sp3n04 "hostname -s ; lsauthent"
krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
```

Adding an authentication method to a partition. (The change is reflected in the SDR and on the nodes.):
Attempt is made to add an authentication method to a partition, but the required software to support the method is not installed:

```
sp3en0/ # lsauthpar -p sp3sp1
Kerberos 4
sp3en0/ # chauthpar -p sp3sp1 k5 k4
sp3n04: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3n06: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3en0/ # lsauthpar -p sp3sp1
Kerberos 5
Kerberos 4
sp3en0/ # rsh sp3n04 "hostname ; lsauthent"
Kerberos 5
Kerberos 4
sp3en0/ # rsh sp3n06 "hostname ; lsauthent"
Kerberos 5
Kerberos 4
```

The example that enables authentication methods assumes that the required security software is already installed in the needed partition. Note, however, that there are no restrictions when disabling already-set authentication methods. In an environment where the software is not already installed for the partition, the attempt to set PSSP authentication methods for the partition will fail, as shown. In the event of such a failure, refer to the PSSP Administration Guide, SA22-7348, for the steps needed to install the required security code on the nodes, and then rerun the command.

Likewise, in a coexistence environment where SP nodes include Version 3.2 and pre-3.2 PSSP levels, attempting to disable an AIX authentication method that is required to be compatible with previous levels of PSSP is not permitted, as shown in the following screen:
2. Methods do not match system or partition level values

In the case of disjoint AIX authentication methods on one host only, the corrective action is for the SP administrator to bring that host's AIX authentication methods in line with the partition settings. This is accomplished through PSSP or AIX administrative commands in one of three ways:

1. Issue chauthpar -f {methods list} from the control workstation:

```
sp3en0/ # chauthpar -f sp3coexist k5 std
chauthpar: 2545-047 The set "k5:std" is not valid; k4 must be included since there is at least one node in the partition running a PSSP code level prior to 3.2.0.0.
```

2. rsh the command, chauthent {methods list}, to the host:

```
sp3en0/ # lsauthpar
Kerberos 5
Kerberos 4
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 5
sp3en0/ # chauthpar -f k5 k4
sp3n05: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 5
Kerberos 4
```

3. Log in to the node and issue the chauthent {methods list} locally

Except for logging in to the node, the PSSP commands needed to set AIX authentication methods locally is shown in the rsh portion of the previous example.

The first approach is the PSSP preferred synchronization method. The other approaches should be used under conditions where chauthpar fails.
However, should the chauthpar fail, the chances that the rsh approach will work are low, given that chauthpar exploits the PSSP dsh command, which, in turn, exploits rsh. If chauthpar fails due to an rsh failure, the rsh alone will, most likely, fail. (Issuing an individual rsh command to each host over which chauthpar will operate is an excellent way to help identify any problems with the state of AIX authentication method settings within the system.)

Given that chauthpar ultimately issues a chauth on hosts for which the underlying rsh was successful, chauthpar and chauthent will experience a failure (will not set methods as requested) under the following conditions:

- /tmp is full (chauthent comments/uncomments rshd and rlogind entries in /etc/inetd.conf, based on the inclusion or absence of -std in chauthent, and uses /tmp to hold a copy of the file it has modified). The error message received is:

```
sp3en0/ # chauthent -k4 -std
chsubserver: error in updating /etc/inetd.conf
chsubserver: error in updating /etc/inetd.conf
```

- The required software for a method is not installed on the host:

```
sp3en0/ # chauthent -k5 -k4 -std
Kerberos 4 permitted on SP node only.
Kerberos 5 requires DCE version 2.2 or greater.
```

- The order of the methods is incorrect, or the methods are invalid:

```
sp3en0/ # chauthent -std -k5 -k4
Invalid authentication method or ordering of methods
```

When neither chauthpar nor rsh/chauthent is successful, logging in to the node and changing the methods locally is the only course of synchronization action, short of rebooting the node, and is less disruptive than rebooting a node. (Node login may take the form of telnet, rlogin, s1term (in write mode), ssh, or other processes that open a terminal session to a node.)

### 8.2.2 User identities and access controls

This section covers user (SP administrator and non-administrator) identities and access controls.

This area covers items that a user needs to do, or things that a user needs to be a member of, before access is granted by PSSP services. These are areas
that the user can directly manipulate or that an SP administrator must manipulate or enable.

The user's identity, the state of his/her credentials, and the access controls that grant/deny a user access to resources are the key impact items during normal state operation and administration.

Many problems a user may face can be quickly isolated and solved by asking the following questions:

- What is my identity?
- How do I establish an identity that I need?
- Am I authorized to do what I have requested? (Am I allowed to access the resource I am trying to use?)

8.2.2.1 What is your identity?
This applies to AIX (user/group identity), DCE (principal identity), and Kerberos V4 (principal identity).

An SP user can determine what his/her identities are on any SP node, plus the control workstation, with the following commands:

AIX: whoami
DCE: klist (requires DCE installed)
KV4: k4list (requires Kerberos V4 installed)

Depending on the PSSP security configuration of the system (or partition), one or more of these identities may need to be established to use or administer PSSP resources. PSSP services grant access to resources based on the authenticated identities of a user and the access controls (specific authorizations) associated with the identities.

8.2.2.2 How do you establish an identity that you need?
An identity is “who you say you are.” An identity is established by authenticating to some service and that service providing you with a way to prove who you say you are to other services or applications. (This would be akin to obtaining a driver's license and presenting it to someone to prove your identity.)

It's possible that a single SP user can have up to three identities at any given time: AIX, DCE, and Kerberos V4. (This is true for both SP administrators and non-administrators.)

In the case of AIX, this is the AIX user ID/group ID pair assigned at the creation of the ID and established with each AIX login. When a user
successfully logs in to a system with a valid user ID and password, the AIX user ID/group ID pair is established for the user by AIX. Once logged in to AIX, a user can change to another AIX identity by issuing the AIX `su` command and supplying the appropriate password.

AIX user IDs are managed across an SP system through PSSP-supplied software or through customer-supplied controls.

A DCE identity is the unique DCE principal name within a specific DCE cell. A DCE identity is established after an AIX user successfully issues a dce_login. (For AIX/DCE integrated login environments, a separate dce_login is not required to establish an initial DCE principal identity. A DCE identity is established as part of the AIX login.) A DCE cell administrator creates DCE user principals and accounts based on the site's user policies. Once logged in as a DCE principal, a user can change to another DCE identity by issuing a dce_login for another principal name and supplying the appropriate password.

PSSP installation and configuration provides scripts that will create PSSP-specific DCE principals/accounts, groups, and organizations for use by PSSP Trusted Services when the services run under a dce security configuration (dce or dce:compat).

All DCE principals and accounts are managed by the DCE cell administrator. PSSP does not supply management routines for DCE principals, accounts, groups, or organizations. Management of these DCE items is covered extensively by DCE administrative facilities.

As with DCE, to establish a Kerberos V4 principal identity, an AIX user must issue a successful `k4init` command. (Refer to the "Integrating Login For Kerberos V4 with AIX" subsection of Chapter 3, "Managing and Using SP Security Services", of the PSSP Administration Guide, SA22-7348, for details on how to set-up the AIX/PSSP Kerberos V4 integrated login.) PSSP installation and configuration scripts create specific Kerberos V4 principals used by PSSP services in a compat security configuration (compat or dce:compat). An SP administrator creates additional Kerberos V4 principals based on the site's user policies and requirements. Once logged in as a Kerberos V4 principal, a user can change to another Kerberos V4 principal by issuing a k4init for another principal name and supplying the appropriate password.

All Kerberos V4 principals are managed by the SP administrator. PSSP provides a Kerberos V4 administrative facility to manage its database of principals.
Chapter 8. Problem isolation and resolution

In the case of DCE and Kerberos V4 credentials, the state of the held credentials at the time of command execution will impact whether or not PSSP services, or AIX secure remote command services, will respond to the request. (Examples of DCE and Kerberos Version 4 credential states are covered elsewhere in this chapter.)

8.2.2.3 Are you authorized?

You may ask yourself: Am I authorized to do what I have requested? (Am I allowed to access the resource I'm trying to use?)

Simply because a user has an AIX identity, a DCE identity, a Kerberos Version 4 identity, or some combination thereof, does not automatically grant the user access to a particular resource. The ability to selectively grant/deny access to a resource is made possible through access controls, commonly referred to as access control lists (ACLs).

Note, however, that an ACL can take the form of text files, AIX user/group memberships, DCE objects, or other formats recognized by a given server or program. Also, an ACL need not be a simple "list." It may be a complex series of cascading controls that join multiple data elements to determine grant/deny access. The actual implementation of an ACL is up to the security mechanisms associated with a server or resource.

For example, the AIX command shutdown requires the following for the command to be executed by the operating system:

- If the user is root, run the command.
- If the user is not root, but the user is in the AIX group shutdown, then setuid to 0 (zero), and run the command.

Therefore, to use the shutdown command, the issuer must either be root or a member of the AIX shutdown group. Access is controlled by the AIX user ID of 0 (zero), or membership in the AIX group shutdown. The AIX operating system controls access to the shutdown command through programs that determine the AIX identity and the AIX group membership of the user issuing shutdown.

In this case, the ACL comes in two forms: The first is a comparison of the issuer's user ID to that of an internal list, where the list contains one element, 0 (zero). If that check fails, the issuer's user identity is compared to those in the AIX shutdown group. If the issuer's user identity appears as a member of the shutdown group, the command is issued; otherwise, the command attempt is rejected.
As would be expected, the ability to query, or display, elements of an ACL are dependent on the security mechanisms that support a particular ACL type. For instance, DCE has commands that show the contents of a DCE ACL object and to manipulate the elements within an ACL object.

In the case of the `shutdown` command, the AIX command, `lsgroup`, displays users that are a member of a particular group, while the AIX command `lsuser` displays group and user ID information about a particular user.

In the samples shown below, the AIX users, `sp_admin` and `dero`, are both members of the `shutdown` group, although, they are not root IDs. These IDs can issue the `shutdown` command. The user, `maarten`, is not a member of the `shutdown` group, is not a root ID, and, therefore, cannot issue the `shutdown` command.

```
sp3en0/ # lsgroup shutdown
shutdown id=21 admin=true users=sp_admin,dero dce_export=false
sp3en0/ # lsuser -a id groups dero
 dero id=202 groups=staff,security,shutdown
sp3en0/ # lsuser -a id groups sp_admin
 sp_admin id=300 groups=system,shutdown
sp3en0/ # lsuser -a id groups maarten
 maarten id=201 groups=staff
sp3en0/ # whoami
 maarten
sp3en0/ # shutdown -Fr
ksh: shutdown: 0403-006 Execute permission denied.
...
sp3en0/ # whoami
 maarten
sp3en0/ # lsuser -a id groups dero
 dero id=201 groups=staff,security,shutdown
sp3en0/ # shutdown -Fr
SHUTDOWN PROGRAM
 Wed Dec 29 09:38:34 EST 1999
Wait for 'Rebooting...' before stopping.
```

In terms of PSSP services, access control is maintained through Kerberos V4 ACLs, DCE ACLs, DCE group memberships, and AIX user/group identity. Additionally, the AIX secure remote commands, such as `rsh` and `rcp`, have their own set of ACLs for each AIX user. The AIX secure remote command ACLs come in the form of authorization files that exist in the user's home directory. (Examples of AIX remote command authentication and authorization messages are covered in a separate section of this chapter.)
8.2.2.4 PSSP examples

The remainder of this subsection contains examples of PSSP Trusted Services under various user identities and ACL states. The examples focus on three areas of PSSP: SDR, Sysctl, and Hardmon.

SDR

The SDR protects its resources (the SDR databases and files) through DCE group membership (in a dce only mode), or when the user is root on the control workstation (in dce:compat or compat security modes). There are eight SDR DCE groups configured by PSSP services that an SDR daemon uses to grant/deny access under a dce only security mode. (The groups can be displayed by issuing `dcecp -c group cat | grep sdr`.)

SDR read operations are permitted by any AIX user, under any security mode, and do not require any specific DCE or AIX group memberships or DCE credentials.

In the first sequence, a root user running in a dce only security mode attempts to write to existing entries in the SDR but is denied due to lack of DCE group membership in a required PSSP DCE group, sdr-write.

After the cell administrator adds the DCE principal to the needed group, the user must destroy and create new DCE credentials in order to reflect the new membership. In the case of a root user holding the DCE self-host principal credentials, that is, the DCE machine principal context/identity, DCE (or just its security client, sec_cl) must be stopped and then started on the host (node or control workstation) in order for the self-host principal’s credentials to reflect the new group membership. (User membership in DCE groups is reflected through the user’s DCE credentials and is considered part of the user’s ticket privilege attributes. The credentials, also called tickets, can only reflect group memberships that were in place at the time the user’s credentials were created. Any group associations established after user credentials were generated will not be reflected until new credentials are created. For more information about DCE credentials (tickets) and their privilege attributes, refer to the “Displaying Privilege Attributes and Tickets” subsection of Chapter 31, “Creating and Maintaining Accounts”, of the book, *IBM DCE Version 3.1 for AIX and Solaris: Administration Guide -- Core Components.*)

Once this takes place, the user can update existing items in the SDR:
As a cell administrator, add the DCE principal, dero2, to the sdr-write class:

```
sp3en0/ # dcecp -c group add sdr-write -member dero2
sp3en0/ # dcecp -c group list sdr-write
  /.../sp_cell/cell_admin
  /.../sp_cell/dero2
```

Note that even as the DCE principal dero2, which is now a member of the sdr-write group, the command still fails. The user must destroy the current credentials and create new ones:

```
sp3en0/ # kdestroy ; exit
sp3en0/ # dce_login dero2
```

The reason why the user had to destroy and then create new credentials after the cell administrator added the principal to the needed group is due to the way in which DCE credentials are created.

At the time dero2 logged into DCE, dero2 was not a member of the sdr-write group; so, the DCE credentials obtained did not reflect such an association. When the cell administrator added dero2 to the sdr-write group, the dero2 credentials still reflected the memberships at the time the credentials were originally created by the user. For dero2’s credentials to reflect the sdr-write membership, the current credentials must be destroyed and the security...
context re-created by way of a kdestroy/dce_login sequence. Note that simply issuing kinit will not create credentials that reflect the new group association.

This is what the dero2 credentials looked like before and after the kdestroy/dce_login. Notice that the new credentials reveal membership in the sdr-write group.

Before:

```
sp3en0/ # klist
DCE Identity Information:
   Warning: Identity information is not certified
   Global Principal: /.../sp_cell/dero2
   Cell: 4b6608a0-9842-11d3-a5a0-02608c2d4a7f /.../sp_cell
   Principal: 0000019e-be0f-21d3-9a00-02608c2d4a7f dero2
   Group: 0000000c-9842-21d3-a501-02608c2d4a7f none
   Local Groups:
      0000000c-9842-21d3-a501-02608c2d4a7f none
... Ticket cache: /opt/dcelocal/var/security/creds/dcecred_1cb46000
Default principal: dero2@sp_cell
...
```

kdestroy and dce_login take place.

After:

```
sp3en0/ # klist
DCE Identity Information:
   Warning: Identity information is not certified
   Global Principal: /.../sp_cell/dero2
   Cell: 4b6608a0-9842-11d3-a5a0-02608c2d4a7f /.../sp_cell
   Principal: 0000019e-be0f-21d3-9a00-02608c2d4a7f dero2
   Group: 0000000c-9842-21d3-a501-02608c2d4a7f none
   Local Groups:
      0000000c-9842-21d3-a501-02608c2d4a7f none
... Ticket cache: /opt/dcelocal/var/security/creds/dcecred_40146600
Default principal: dero2@sp_cell
...
```

In this next sequence, a root user is running in dce:compat, and then just compat. The user then attempts to create items in the SDR and is successful even though the user’s DCE identity is not a member of the sdr-write group. The create operation is permitted since the user is root and is running the command from the control workstation, which is node number 0 (zero) in SP terms.
Sequence 3’s security configuration information, SP node number, user credential identities, and DCE group membership:

```bash
sp3en0/ # /usr/lpp/ssp/install/bin/node_number
0
sp3en0/ # whoami ; klist | grep lob
root
   Global Principal: /.../sp_cell/dero2
sp3en0/ # lsauths
dCE
Compatibility
sp3en0/ # dcecp -c group list sdr-write
/.../sp_cell/atest
/.../sp_cell/sp_admin
```

Update values in the SDR:

```bash
sp3en0/ # SDRGetObjects -x -q Syspar auth_methods
k5
sp3en0/ # SDRChangeAttrValues Syspar auth_methods=k5:std
sp3en0/ # SDRGetObjects -x -q Syspar auth_methods
k5:std
```

In the third sequence, a non-root user running in a dce only security mode attempts to create items in the SDR, but is denied due to lack of DCE group membership in the PSSP DCE group, sdr-admin. After the cell administrator adds the DCE principal to the needed group, the user must destroy and create new DCE credentials in order to reflect the new membership. Once this takes place, the user can create items in the SDR:

```bash
sp3en0/ # whoami ; klist | grep lob
dero
   Global Principal: /.../sp_cell/dero
sp3en0/ # SDRCreateClass testclass attr=string
SDRCreateClass: 0025-001 A read-only SDR session was obtained. Operations that create or change data are not allowed.
sp3en0/ # dcecp -c prin show dero | grep groups
{groups spsec-admin hm-control}
```

As a cell administrator, add the DCE principal dero to the sdr-admin class:

```bash
sp3en0/ # dcecp -c group add sdr-admin -member dero
sp3en0/ # dcecp -c group list sdr-admin
/.../sp_cell/sp_admin
/.../sp_cell/dero
```
As the DCE principal, dero, the SDRCreateClass command still fails, even though dero was added to the sdr-admin group. The user must destroy the current credentials and create new ones:

```
sp3en0/ # klist | grep lob
   Global Principal: /.../sp_cell/dero
sp3en0/ # SDRCreateClass testclass attr=string
SDRCreateClass: 0025-001 A read-only SDR session was obtained. Operations that create or change data are not allowed.
sp3en0/ # ls -l /spdata/sys1/sdr/defs/testclass
ls: 0653-341 The file /spdata/sys1/sdr/defs/testclass does not exist.
sp3en0/ # kdestroy ; exit
sp3en0/ # dce_login dero
... DCE LOGIN SUCCESSFUL ...
sp3en0/ # SDRCreateClass testclass attr=string
sp3en0/ # ls -l /spdata/sys1/sdr/defs/testclass
-rw------- 1 root system 10 Dec 28 14:08 /spdata/sys1/sdr/defs/
testclass
sp3en0/ # cat /spdata/sys1/sdr/defs/testclass
PS1=attr
```

The reason why the user had to destroy and then create new credentials after the cell administrator added the principal to the needed group is due to the way in which DCE credentials are created.

At the time dero logged into DCE, dero was not a member of the sdr-admin group; so, the DCE credentials obtained did not reflect such an association. When the cell administrator added dero to the sdr-admin group, the dero credentials still reflected the user's memberships at the time the credentials were created. For dero's credentials to reflect an sdr-admin membership, the current credentials must be destroyed and the security context re-created through a kdestroy/dce_login sequence. Note that simply issuing kinit will not create credentials that reflect the new group association.

This is what the dero credentials looked like before and after the kdestroy/dce_login. Notice that the new credentials reveal membership in the sdr-admin group.

Before:
kdestroy and dce_login take place.

After:

In the fourth sequence, a root user running in dce:compat and compat security modes attempts to create items in the SDR and is successful even though the user's DCE identity is not a member of the sdr-admin group. The create operation is permitted since the user is root and is running the command from the control workstation, which is node number 0 (zero) in SP terms.

Sequence four's security configuration information, SP node number, user credential identities, and DCE group membership follow:
Create an SDR class:

```bash
sp3en0/ # /usr/lpp/ssp/install/bin/node_number
0
sp3en0/ # whoami ; klist | grep lob
root
Global Principal: /.../sp_cell/derol
sp3en0/ # lsauths
DCE Compatibility
sp3en0/ # dcecp -c group list sdr-admin
/.../sp_cell/sp_admin
/.../sp_cell/cell_admin
```

**Sysctl**

Sysctl is a powerful PSSP client/server package. In addition to the default PSSP Sysctl Systems Management procedures and wealth of built-in commands and scripts, Sysctl can be customized and extended through the use of customer-supplied Tool Control Language (TCL) programs. Plus, Sysctl commands can be defined so that sets of users--or just one user--can execute Sysctl commands. Further, non-root users can be granted access to root level commands by wrapping the root level commands in Sysctl procedures, which, in turn, become the actual "commands" issued by users.

Since a Sysctl server runs as root and all Sysctl client command requests are actually executed by the server, it is possible for a non-root user to run a command, a program, or a script that requires root authority. Since a non-root user can be granted authority to specific sets of Sysctl commands, the user can have access to various root functions without actually having any true root authority. It is an excellent way to define AIX and PSSP administrative responsibilities to an administrative team without granting them total root access. (An excellent use would be to allow certain users the ability to issue startsrc commands for a well-defined set of servers, such as Web servers, in the event that a server is down.) The same approach can be applied to non-root users requiring access to specific sets of root commands, ranging from technical computing, to data processing, to custom applications, without actually granting them complete root authority.
Obviously, the need for Sysctl to securely and reliably grant/deny access to the resources it protects is critical. This is accomplished through a two-tier process.

First, when a Sysctl client connects to a Sysctl server (daemon), the Sysctl svcconnect built-in procedure is evaluated before evaluating any commands sent by the user (or before entering an interactive Sysctl session). The results of the svcconnect procedure determines whether the server will allow client requests to be processed. In short, if the client can be authenticated, Sysctl command processing will begin; otherwise, the client's request is denied due to an authentication failure.

Sysctl servers are shipped with a default svcconnect of “AUTH”, meaning the client must be authenticated by the server before allowing the client's command requests to be processed. This setting can be changed by adding an svcconnect statement to the server's configuration file, /etc/sysctl.conf, and then restarting the server. For example, to allow any client to connect to a Sysctl server on a given node without any authentication requirements, add the line “rename svcconnect {}” to the configuration file (less the quotes). Note, however, that this is not a recommended way to run a Sysctl server because it takes away the first tier of Sysctl protection. (For complete details on Sysctl configuration and the svcconnect command, refer to the PSSP Administration guide and Sysctl on-line help.)

The second tier comes into play once the client is authenticated by the server. After the server authenticates the client, it determines whether or not the client is authorized to execute the requested Sysctl command. The server determines the authorization level associated with the command, and, if the authorization criteria for a command are satisfied, the server will run the command for the client; otherwise, the request is denied due to an authorization failure. (A Sysctl command has one of four possible authorizations associated with it: NONE - any user can run the command; AUTH - any authenticated user can run the command; ACL - only principals that appear in the command's ACL can run the command; SYSTEM - only the server can run the command. Clients are not permitted to run the command.)

By and large, once a client is authenticated and connected to a server, the most common Sysctl problems are those related to command authorization failures. Most authorization failures are due to a lack of membership in one or more Sysctl ACLs. Unlike the earlier SDR examples, where DCE group membership or AIX user ID are the deciding grant/deny factors, or the forthcoming Hardmon examples, where DCE group membership and/or ACLs (DCE or Kerberos V4) are the deciding grant/deny factors, Sysctl bases final grant/deny access on its ACLs.
Sysctl ACLs come in two formats: Kerberos V4 ACLs, used in a compat mode, and DCE ACL objects, used in a dce mode. When a Sysctl server runs under a combined dce:compat mode, both ACL types may be checked depending on the authenticated identities of the client). Under conditions where both ACL types are checked, the Sysctl server checks DCE ACLs first, then Kerberos V4 ACLs.

---

**Note**

When a Sysctl server under a dce:compat mode determines that the client is authorized by way of a DCE ACL, it does not check Kerberos V4 ACLs.

The Sysctl examples to follow cover authentication failures, authorization failures, various Sysctl ACL query commands, failures when the Sysctl client is running in one PSSP security mode and the remote Sysctl server is running in a different PSSP security mode, and suggestions on managing Sysctl ACLs.

The “Controlling Remote Execution by using Sysctl” chapter of the *PSSP Administration Guide*, SA22-7348, contains additional information that will prove useful to the SP administrator.

**Authentication Failure Examples**

The Sysctl requests fail because the client could not be authenticated to the server. To correct the situation, the client must obtain valid credentials.

**Expired DCE and Kerberos V4 credentials:**

```bash
sp3en0/ # sysctl whoami -v
2502-605 Your DCE credentials have expired.
Authentication ticket expired
2502-605 Your Kerberos V4 credentials have expired.
```

**DCE credentials do not exist and Kerberos V4 credentials have expired:**

```bash
sp3en0/ # sysctl whoami -v
2502-603 You do not have DCE credentials.
No currently established network identity for this context exists
2502-605 Your Kerberos V4 credentials have expired.
```

**Neither DCE nor Kerberos V4 credentials exist:**

```bash
sp3en0/ # sysctl whoami -v
2502-603 You do not have DCE credentials.
No currently established network identity for this context exists
2502-605 Your Kerberos V4 credentials have expired.
```
DCE and Kerberos V4 credentials are corrupt.

A non-root user attempts to "steal" root's DCE and Kerberos V4 credentials:

The preceding example is particularly interesting because it shows how klist (a DCE program), k4list (a Kerberos V4 program), and Sysctl (a PSSP program) react to a user that attempts to use DCE and Kerberos V4 credentials created and owned by another user - access is denied. In this example, the non-root user, dero, set his KRB5CCNAME and KRBTKFILE...
credential cache environment variables to point to credential cache files owned by root:system.

The programs that generate (create) DCE and Kerberos V4 credential files ensure that only the AIX users for whom the credentials are created have read/write permissions to these files. (Of course, root can read any file.) The files created are set to the user's AIX user and group names, and the permission bits are set to 600 (-rw-------). Users should not alter the file permissions or user/group ownership of their credential files. Doing so may allow another user to read, use, and, possibly, modify the credentials.

In the following examples, the Sysct1 client and server are in a dce:compat environment. The client requests are successful because the client can be authenticated to the server, even when the client only has valid credentials for one of the security methods. Also, note that there are no authentication errors returned when the client only has one type of valid credentials (either DCE or Kerberos V4). When a Sysct1 server can authenticate a client under at least one of the enabled PSSP security methods, it does not return error messages for failures in authenticating the client against the other method. This is shown by the results of the Sysct1 command, whoami -v. (whoami is not ACL protected and can be issued by any authenticated request.)

User's DCE credentials have been destroyed (via kdestroy), but the Kerberos V4 credentials are valid.

<table>
<thead>
<tr>
<th>sp3en0/</th>
<th># lsauthts</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCE</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sp3en0/</th>
<th># klist</th>
</tr>
</thead>
<tbody>
<tr>
<td>No DCE identity available: No currently established network identity for this context exists (dce/sec)</td>
<td></td>
</tr>
</tbody>
</table>

Kerberos Ticket Information:

| klist: No credentials cache file found (dce / krb) (ticket cache /opt/dcslocal/var/security/creds/cred_cred_bd587100) |
|---------|-------------|
| sp3en0/ | # klist | grep pal: |
| Principal: root.admin@ITSO.IBM.COM |
| sp3en0/ | # sysctl whoami -v |
| DCE: |
| K4: root.admin@ITSO.IBM.COM |
| AIX: root |

User's DCE credentials are valid, but the Kerberos V4 credentials are destroyed (via kdestroy).
Sysct1 client requests will fail when the client and the server do not share at least one PSSP authentication method in common. (For example, if the client is running in a dce only mode, the server must be running in a mode that includes dce, as in dce (only) or dce:compat.)

In a multi-partition environment, where each partition is running a different security mode, it is expected that the client and server will not communicate unless they share a PSSP authentication method in common. (This is equivalent to humans speaking a common language in order to communicate verbally.)

In a single partition environment, where all hosts should be running the same security mode, should the PSSP authentication methods between a client and server (nodes or the control workstation) become disjoint such that they do not share a method in common, communication will not be successful.

In the next example, the control workstation is at dce:compat, while the nodes on which the Sysct1 client request is to run are at compat, dce:compat, and dce. As such, the Sysct1 client request from the dce:compat control workstation to the nodes is successful in all cases because the client shares at least one method in common with each node.
In this next example, a compat node sends a Sysctl request to a node in dce and a node in compat. As such, the Sysctl client request from a compat node to a compat node is successful, while the request from a compat to a dce node is denied.

```
sp3en0/ # lsauths ; klist | grep lob ; k4list | grep pal:
DCE
Compatibility
  Global Principal: /.../sp_cell/hosts/sp3en0/self
  Principal: root.admin@ITSO.IBM.COM
sp3en0/ # sysct1 -h sp3n01 -h sp3n04 -h sp3n03 whoami -v
>> sp3n04
DCE:
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
>> sp3n01
DCE: /.../sp_cell/hosts/sp3en0/self
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
>> sp3n03
DCE: /.../sp_cell/hosts/sp3en0/self
K4: 
AIX: root
<<
```

In the third example, a dce node sends a sysct1 request to a node in dce and a node in compat. As such, the Sysctl client request from the dce node to a dce node is successful, while the request from the dce node to the compat node is denied.

```
sp3n04/ # lsauths ; k4list | grep pal:
Compatibility
  Principal: root.admin@ITSO.IBM.COM
sp3n04/ # sysct1 -h sp3n01 -h sp3n03 whoami -v
>> sp3n03
<<
>> sp3n01
DCE:
K4: root.admin@ITSO.IBM.COM
AIX: root
<<
```

In this next example, a compat node sends a Sysct1 request to a node in dce and a node in compat. As such, the Sysct1 client request from a compat node to a compat node is successful, while the request from a compat to a dce node is denied.
Authorization failure examples

In all authorization examples, the user has been authenticated by the server; otherwise, the authorization process would not have taken place, and all Sysctl command examples imply that the command is ACL protected, that is, it has a callback of ACL.

When Sysctl requests fail due to authorization, the user's DCE principal or Kerberos V4 principal does not appear in a required ACL. The first step in Sysctl authorization failure problem determination is to determine whether or not the user has access to a Sysctl object, where a Sysctl object is a Sysctl command, variable, or class.

There are two Sysctl commands that are important to SP administrators and other users:

1. checkauth - This command executes the authorization callback of an object and determines whether or not a user is authorized to access or run a Sysctl command, variable, or class.

2. getauth - This command displays the authorization callback of an object, that is, what type of authorization is required to access a Sysctl command, variable, or class.

When checkauth is issued against a Sysctl object that is ACL protected, the user's authorization is checked against both DCE and Kerberos V4 ACLs, provided that both the dce and compat methods are enabled; otherwise, only DCE or Kerberos V4 ACLs are checked.

If the user is authorized by way of, at least, one of the enabled ACL types, an Authorization OK message is returned; otherwise, an Authorization denied message is returned. In the event that an ACL object exists in one mode but not in the other, in a dce:compat environment, the results of checkauth are based on the ACL object that was found.
Checking access requirements to the Sysctl command, pdf. Authorization granted:

```
sp3en0/ # getauth -cmd pdf
Object Name: pdf
Object Type: command
Object Callback = {AUTH}
```

```
sp3en0/ # sysctl pdf | head -n 3
Filesystem  Size-KB  Used-KB  Free-KB  %Free  iUsed  iFree  %iFree
--------------------------------------------------------------------------------
/          32768    8604  24164   74%    1263  15121   93%
```

Checking access requirements to the Sysctl environment variable, SCDEPRIN. Authorization granted:

```
sp3en0/ # sysctl checkauth -var SCDEPRIN
SCDEPRIN: Authorization OK.
sp3en0/ # sysctl getauth -var SCDEPRIN
Object Name: SCDEPRIN
Object Type: variable
Object Callback = {NONE}
```

```
sp3en0/ # sysctl echo '$SCDEPRIN'
/.../sp_cell/hosts/sp3en0/self
```

Checking access requirements for the Sysctl command, whoami. Authorization granted:

```
sp3en0/ # sysctl checkauth -cmd whoami
whoami: Authorization OK.
sp3en0/ # sysctl getauth -cmd whoami
Object Name: whoami
Object Type: command
Object Callback = {NONE}
```

```
sp3en0/ # sysctl whoami
/.../sp_cell/hosts/sp3en0/self
```

Checking access requirements for the Sysctl command include (used in Sysctl server configurations). Authorization denied:
Even the checkauth and getauth commands can be checked using themselves because every Sysctl resource has an associated authorization callback.

A benefit of using the `getauth` command is that, when an object's callback is ACL, the name of the ACL is also returned. For `getauth` results containing only the string "{ACL}," the default Sysctl ACL, `/etc/sysctl.acl`, is the name of the ACL object associated with the Sysctl function.

Checking access requirements for the TCL commands `puts` and `exec`. Authorization denied.
Chapter 8. Problem isolation and resolution

Checking access requirements for puts and exec after the administrator added the user's DCE principal to the needed ACL. Authorization granted.

Checking access requirements for the PSSP Problem Management command pmanSDRadd. It doesn't matter that the command is used incorrectly because authorization to the command itself is denied.

Checking access requirements for the custom Sysctl procedure test-proc1:

Checking access requirements for svclog. Authorization denied.
Checking access requirements for svclog after the administrator added the user’s DCE principal to the needed ACL. Authorization granted.

For a single Sysctl client request directed to Sysctl servers on different hosts, Sysctl command authorization is controlled by the ACLs associated with the command on each host. Thus, the ACLs for Sysctl commands can vary from host to host, depending on the requirements of the site.

In this simple example, a Sysctl whoami command is directed to two hosts, where the authorization callback of whoami on the first host is NONE and ACL on the second host. (The command’s authorization callback was changed by the SP administrator.) The client’s principal identity is not in the whoami ACL on the second host.

Taking this idea one step further, in a dce.compat mode, where the user’s DCE identity is not authorized by a Sysctl DCE ACL, but the user’s Kerberos V4 identity is authorized by the associated Kerberos V4 version ACL, the Sysctl command runs without a DCE authorization failure. Like its authentication process, Sysctl’s authorization process will not report an authorization failure if the client can be successfully authorized by at least one of the enabled security methods.
In the next example, the user has DCE and Kerberos V4 identities but is only permitted to execute the `svclog` command according to the Kerberos V4 ACL. When the Kerberos V4 ACL is updated and no longer includes the user’s Kerberos V4 identity, the user is no longer authorized to run the `svclog` command.

```
sp3en0/ # sysctl whoami -v
DCE: /.../sp_cell/sp_admin
K4: dero@ITSO.IBM.COM
AIX: root
sp3en0/ # sysctl getauth -cmd svclog
...
Object Callback = {ACL}
sp3en0/ # dcecp -c acl show /.:/subs/sys/sp3en0/sysctl/etc/sysctl.acl
{user sp_admin --t}
...
sp3en0/ # sysctl acllist -f /etc/sysctl.acl
_principal root.admin@ITSO.IBM.COM
_principal dero@ITSO.IBM.COM
sp3en0/ # sysctl svclog {...MESSAGE LOGGED...}
sp3en0/ # tail -n 1 /var/adm/SPlogs/sysctl/sysctld.log
Jan 16 11:58:18 ...MESSAGE LOGGED...

sp3en0/ # sysctl svclog {...NEXT MESSAGE LOGGED...}
sp3en0/ # sysctl checkauth -cmd svclog
sysctl: 2501-131 checkauth: Authorization denied for object svclog.
```

dero@ITSO.IBM.COM is removed from `/etc/sysctl.acl`:

```
sp3en0/ # sysctl acllist -f /etc/sysctl.acl
_principal root.admin@ITSO.IBM.COM
sp3en0/ # tail -n 1 /var/adm/SPlogs/sysctl/sysctld.log
Jan 16 11:58:18 ...MESSAGE LOGGED...
```

### Querying and managing Sysctl ACLs

Determining which users require access to Sysctl ACL protected resources is critical to ensuring the security and integrity of the SP nodes and the control workstation. A well-defined policy that outlines which Sysctl resources are available to the user population is the first level of control. The second is a documented procedure that users follow to request access to Sysctl resources. Once it is determined that a user’s DCE and/or Kerberos V4 principal will be added to, removed from, or changed within a Sysctl ACL, an SP administrator must then take the necessary steps to manage Kerberos V4 and/or DCE ACLs. Managing Sysctl ACLs is achieved through the use of PSSP, DCE, and basic UNIX commands.

### Sysctl Kerberos V4 ACL

The names of the PSSP shipped (default) Kerberos V4 ACLs are:
There are six Sysctl commands to manage its Kerberos V4 ACLs, one Sysctl command to check a user’s authorization based on entries in a Kerberos V4 ACL (denoted with "***"), and one Sysctl command to display the names of all ACLs used by the server (denoted with "****").

- acladd - Inserts news entries into an existing ACL file
- *aclcheck - Determines whether a user is authorized by the entries in an ACL
- aclcreate - Creates a new ACL file and, optionally, allows principals to be added to it
- acldelete - Removes entries from an ACL file. It does not delete the ACL file itself
- acldestroy - Deletes an ACL file from the local host
- acllist - Verifies that the ACL file exists and, if so, displays all its entries
- aclderecreate - Deletes an ACL file from the local host, then re-creates it, and optionally allows principals to be added to it
- **whatacls - Displays the names of all ACLs used by a Sysctl server on a host. It does not display the entries in the ACLs

The aclcheck and whatacls commands are the only ones in the list that work with both Kerberos V4 and DCE ACLs. All of the other commands are only

---

**Note**

The last two ACLs will only appear when the Restricted Root Access (RRA) function of PSSP is enabled.

Refer to the RRA section of this book for information on enabling and using the RRA function.
valid for use with Kerberos V4 ACLs. The aclcheck, acllist, and whatacl
commands are the only ones in the list that have an authorization callback of
NONE. All other commands have an authorization callback of ACL and
require a principal entry in /etc/sysctl.acl to be executed by a user.

The documentation for these commands exists in softcopy only. To access
command syntax and associated examples, use the Sysctl help command.
Example: # sysctl help acladd. The results of the command can be piped to
"more" or redirected to a file.

Except for acllist, caution and care must be exercised when using the acl*
commands. The acl* commands will work over a specific ACL file when they
are invoked with -f {filename}. In the absence of an explicit -f file name,
/etc/sysctl.acl is assumed as the default. /etc/sysctl.acl is the ACL used for all
Sysctl commands where no explicit ACL file name has been specified. This
includes the commands, svcrestart, svclog, exec, and system. It also gives
users the authority to delete or modify ACLs and ACL entries for any other
Kerberos V4 ACL. Only those users that require this level of authority should
be placed in /etc/sysctl.acl. (Chapter 2 of the PSSP: Command and Technical
Reference, SA22-7351, contains a section, called sysctl.acl, which identifies
the required file format and valid entry types for all Sysctl Kerberos V4 ACLs.)

Since Kerberos V4 ACL files are stored in plain text (ASCII) format, the files
can also be edited using a text editor, such as vi. However, managing the
entries using a text editor requires root authority because the files are owned
by root:system and have permissions of 644 (-rw-r--r--), except for
/etc/sysctl.pman.acl, which has permissions of 600 (-rw-------). Users other
than root should not have AIX write authority to these files. Whether or not
general users should have read authority to these files should be evaluated
carefully for security reasons.

From one perspective, general read access to the ACL files does not
compromise system integrity. They do not contain password data, user profile
information, or access protocols. From another perspective, the files do
contain "sensitive" data in that they list the exact principals that do/do not
have authorization to Sysctl resources. A hacker can use this information to
scan for credential files that match specific principals, steal those credentials,
and then continue other attacks through Sysctl (depending on the stolen
principal's membership in the ACLs).

While this might sound like an unrealistic scenario, keep in mind that some
security compromises are clandestine stealth attacks. "Closing" otherwise
"open" data, such as the Kerberos V4 ACL information, only helps to reduce
risk points within the system. This is not unlike the reason for displaying the
message, *You entered an invalid login name or password*, at an AIX login
when an incorrect password has been entered. If the message stated that the
password was incorrect, it implies that the user ID is correct. This is valuable
information! Now, a hacker has only to find the right password to go with the
account in order to access the system. (And there are plenty of password
generating/dictionary attack programs available on the Internet to do just
that.) The bottom line is this: If there is no legitimate reason to make data
read accessible by general users, restrict it. In the end, any data that can be
gathered about a system can, eventually, be used to build attack profiles, for
social engineering, and to map out possible weaknesses.

A list of Kerberos V4 ACLs known to a Sysctl daemon can be displayed with
the `whatacls` command. The list displayed is built when Sysctl's configuration
files are processed during server initialization. However, the server does not
actually scan the file system at start time to determine if the files actually
exist. The only time the server attempts to access an ACL file is when a
command with an authorization callback of ACL is processed. If the ACL file
does not exist, authorization is denied; otherwise, authorization is
granted/denied based on the client's identity in the ACL file.

The ACL file `/etc/no-acl-file.acl` doesn't exist in the file system, but `whatacls`
lists it as a known ACL. The client's request to the no-acl-file command is
denied because the authorization file doesn't exist:

```
sp3en0/ # tail -n 3 /etc/sysctl.conf
create proc no-acl-file {} {ACL /etc/no-acl-file.acl} {
    puts "This message is only displayed when the file /etc/no-acl-file.acl exists
    and a valid principal is in it."
}
sp3en0/ # ls -l /etc/no-acl-file.acl
ls: 0653-341 The file /etc/no-acl-file.acl does not exist.
sp3en0/ # sysctl whatacls
etc/sysctl.acl
etc/no-acl-file.acl
...
sp3en0/ # sysctl checkauth -cmd no-acl-file
sp3en0/ # sysctl no-acl-file
```

The `/etc/no-acl-file.acl` was created, and the client's Kerberos V4 principal
was added to it. The Sysctl server did not need to be restarted in order to
“know” about the ACL since it already had a reference to the ACL file.
As far as synchronizing, or managing, the contents of Kerberos V4 ACL files is concerned, if the contents of the various ACL files are identical throughout the system or identical with respect to some collection(s) of nodes, the PSSP remote parallel copy command, `pcp`, can be used to move the files from a source host to the target hosts. The `pcp` command can copy on partition, node group, and system-wide boundaries. It also has the flexibility to copy an entire directory in one invocation. (`pcp` could even be automated through a shell script launched by cron.)

A similar approach to pushing the ACL files from a source location to various destinations is to use PSSP’s File Collections facility. The Kerberos V4 ACLs can be pushed from the control workstation to nodes in different collections at specific intervals. In essence, a “master set” of Kerberos V4 ACLs are used to periodically refresh the files on the nodes.

Still another approach is for the SP administrator to issue Sysctl Kerberos V4 ACL management commands to multiple hosts. (You cannot connect to multiple servers in Sysctl’s interactive mode.) Note, however, that, except for the `aclcheck` and `acllist` commands, `acl*` commands that add, delete, or modify ACLs require that the SP administrator have authority to run the commands on the remote hosts. The administrator's Kerberos V4 principal identity must appear in `/etc/sysctl.acl` on the remote hosts; otherwise, the server will deny the request.

In the following example, the `/etc/sysctl.acl` file is checked on two hosts for a principal entry of dero. A 0 (zero) return code means that no match was found.
The SP administrator adds the principal dero to the two ACL files:

```
sp3en0/ # sysctl -h sp3n04 -h sp3n05 acladd -f /etc/sysctl.test.acl -p dero
>> sp3n04
_principal dero@ITSO.IBM.COM
<<
>> sp3n05
_principal dero@ITSO.IBM.COM
```

Then, the SP administrator checks the ACLs again for the principal entry, dero, and receives a return code of 1, which means a principal match was found:

```
sp3en0/ # sysctl -h sp3n04 -h sp3n05 aclcheck -f /etc/sysctl.test.acl dero
>> sp3n04
1
<<
>> sp3n05
1
```

The SP administrator pushes the control workstation's /etc/sysctl.custom-cmds.acl file to all the nodes in the system via pcp. The date/time stamp of the file is preserved in the copy.

```
sp3en0/ # pcp ^-avG* -p /etc/sysctl.custom-cmds.acl /etc/sysctl.custom-cmds.acl
```

Sysctl DCE ACLs

The names of the PSSP shipped (default) DCE ACLs are:

- etc/sysctl.acl
- etc/sysctl.mmcmd.acl
- etc/sysctl.vsd.acl
- etc/logmgt.acl
- etc/sysctl.rootcmds.acl
- etc/sysctl.pman.acl
- etc/sysctl.haem.acl
- etc/sysctl.install.acl
There are two commands to manage Sysctl DCE ACLs (PSSP supplied and DCE supplied), one Sysctl command to check a user's authorization, based on entries in a DCE ACL (denoted with "*"), and one Sysctl command to display the names of all ACLs used by the server (denoted with "**").

- **spacl (SP ACL Management)** - This is the PSSP command to display and alter PSSP DCE ACL object entries (for Hardmon and Sysctl) by authorized users. The command only works with PSSP DCE ACLs and does not create/delete ACL objects.

- **acl** - This is the DCE dcecp command that, in conjunction with its qualifiers (check, show, modify, and others), allows authorized DCE principals to display and alter Sysctl ACL object entries.

- **aclcheck** - This is the Sysctl command that determines whether a user is authorized by the entries in an ACL.

- **whatacls** - This command displays the names of all ACLs used by a Sysctl server on a host. It does not list the entries within the ACLs.

Sysctl DCE ACLs are not stored in individual files, such as the Sysctl Kerberos V4 ACL files. The DCE ACL objects are part of a DCE ACL database that the Sysctl server manages. The database exists on a per-host basis and is actually made up of three files, db_acl, db_name, and db_object, and stored under /var/sysctl/. (These files are stored in a binary format and cannot be edited directly.) All Sysctl and dcecp client requests that view or modify the contents of an ACL object are actually performed through a Sysctl daemon's DCE ACL manager. The Sysctl daemon's DCE ACL manager is transparent to all client requests, including dcecp acl commands, but is ultimately responsible for reading/writing the Sysctl DCE database files.

Unlike Kerberos V4 ACL files, which can be copied from host-to-host for synchronization reasons, copying the DCE database files from one host to another will cause a Sysctl server to hang (on the host to which the database files were copied). This, in turn, will hang Sysctl client requests that connect to the hung server. Data that is specific to an instance of the Sysctl daemon running on a node (or control workstation) is stored in the db_* files.
Moreover, unlike Kerberos V4 ACL files, the SP administrator does not directly create a Sysctl DCE ACL object. Instead, at initialization time, the Sysctl daemon creates DCE ACL objects in its database files, based on the Sysctl include and proc statements found in its configuration file, /etc/sysctl.conf. Of the procedures detected by the Sysctl daemon at initialization, those that have ACL authorization callbacks will have corresponding DCE ACL objects created in its local DCE ACL database. To define a new ACL object for a custom procedure, the SP administrator need only ensure that a unique ACL name appears in the proc definition of a custom procedure. To associate a new command with an existing ACL object, the administrator need only include the name of an existing ACL object in the command’s proc definition.

For example, when the following proc definition is placed in the /etc/sysctl.conf file and the daemon is restarted, a new Sysctl command, TclTime, and a new DCE ACL object, tcltime.acl, will exist.

```tcl
create proc TclTime {} {ACL /tcltime.acl} {
    puts [fmtclock [getclock] %X]
}
```

Issuing the Sysctl whatacls commands reveals the new DCE ACL object:

```
sp3en0/ # sysctl whatacls
/etc/sysctl.acl
/etc/sysctl.mmcmd.acl
/etc/sysctl.vsd.acl
/etc/sysctl.logmgt.acl
/etc/sysctl.rootcmds.acl
/etc/sysctl.pman.acl
```

When the following proc definition is placed in the /etc/sysctl.conf file and the daemon is restarted, a new Sysctl command, syscd-DCE-identity, would exist, and its ACL would be associated with default ACL, etc/sysctl.acl. (This small procedure actually lists the DCE credentials of the Sysctl server at the time the command is executed. Such a command might prove to be useful when debugging custom Sysctl procedures, or debugging Sysctl DCE problems in general.)
Up to this point, Sysctl DCE ACL objects have been referenced in an
ambiguous way, as just some "objects" that live in a special DCE database
that is owned and accessed by a Sysctl daemon. In order to effectively use
either the PSSP `spacl` command or the DCE `dcecp acl` command, some
explanation of what a Sysctl DCE ACL object "looks like" is in order.

Whether `spacl` or `dcecp` commands are used to access Sysctl DCE ACL
objects, the objects are, ultimately, referenced as fully-qualified DCE Cell
Directory Service (CDS) names consisting of the cell name (or the string "/.: to
denote the local cell), the CDS directory path for the DCE RPC entry for the
Sysctl server, and the object name itself. For Sysctl, the CDS path to the
Sysctl server on a DCE host, named DCEhostname, is
"/./subsys/ssp/DCEhostname/sysctl".

The object name, appended to the CDS path, is taken from the ACL name
found in a command's proc definition statement. A good way to find the object
name is to use the Sysctl `getauth` command because it returns, as part of its
output, the ACL name associated with a command (when the command is
ACL protected). The Sysctl `whatacls` command displays only the object names
of all Sysctl DCE objects known to the local daemon.

For example, a fully-qualified CDS name for the default Sysctl ACL object
might be: "/./subsys/ssp/DCEhostname/sysctl/etc/sysctl.acl".

The default Sysctl DCE object names, listed earlier, are associated with
Sysctl procedures and commands as part of the daemon's initialization
process. In a previous example, a new ACL object, tcltime.acl, was created at
the time the daemon initialized and appeared in the ACL name in TclTime's
proc definition.

Consider, for a moment, what is required and how to use the `dcecp acl` and
`spacl` commands. The common mistake, when managing Sysctl ACLs through
dcecp, is to use only the DCE object name instead of the fully-qualified CDS
name; use etc/sysctl.acl instead of
./:subsys/ssp/DCEhostname/sysctl/etc/sysctl.acl. When managing Sysctl
DCE ACLs via dcecp, using anything other than a fully-qualified name will
result in an error.

Actually, this is only "mostly" true. A user with DCE cell administrator
authority can actually manipulate the ACL of
./:subsys/ssp/DCEhostname/sysctl through the dcecp acl command,
provided the modifier, -entry, is specified. (spacl does not permit this type of
access, however.) A Sysctl server stores its network location information in a
server entry in the CDS. (Network location information is also referred to as
RPC end-points in DCE terms.) The server entry in the CDS has the same
name as the server itself and an associated CDS ACL. The acl command's
-entry option is used to operate on the server entry ACL in the CDS, rather
than the Sysctl server's ACL objects. In general, the SP administrator should
never need to manipulate Sysctl's ACL entry in the CDS.

To be able to manage each of the Sysctl DCE ACL objects, the
administrator's DCE principal must be a member of the PSSP DCE group,
spsec-admin. To facilitate this effort, it is recommended that a unique DCE
principal be created for PSSP DCE administrative purposes. Make this
principal a member of the spsec-admin group. This group is created by the
PSSP DCE security installation and configuration routines, and spsec-admin
is automatically added as a member to the default Sysctl DCE ACLs.

The Sysctl DCE ACL manipulation examples in this section were done by a
DCE principal, sp_admin. sp_admin, which was added to the spsec-admin
group.

The sp_admin attempts to display DCE ACL entries using dcecp before the
principal is a member of the spsec-admin group.

```
sp3en0/ # klist | grep lob
Global Principal: /.../sp_cell/sp_admin
sp3en0/ # dcecp -c acl show /.:/subsys/ssp/sp3en0/sysctl/etc/sysctl.acl
Error: msgID=0x17122025 permission not valid for this acl
sp3en0/ # dcecp -c account show sp_admin | grep groups
[group none]
```

The sp_admin attempts to display DCE ACL entries using spacl before the
principal is a member of the spsec-admin group. (-n instructs spacl to work
over the local instance of the specified ACL object. In this case, it's for the
control workstation.)
As the DCE cell administrator, add the DCE principal to the spsec-admin group.

After sp_admin issues a new dce_login to reflect membership in the spsec_admin group, access to Sysctl DCE ACLs is authorized.

sp_admin, along with any other DCE principal, has authorization to view Sysctl's CDS ACL entry but does not have authority to alter the ACL. (If this type of authority is required by the SP administrator, the DCE cell administrator needs to add the appropriate entry in the CDS ACL.)
One way to manage the entries of Sysctl DCE ACL objects throughout an entire system is through the use of the `spacl` command. (As noted previously, to use `spacl`, the issuer must have valid DCE credentials and a DCE principal that is authorized to access Sysctl DCE ACLs.)

Displaying the entries of `/etc/sysctl.acl` on all nodes in the system where PSSP services are configured for DCE security, including the control workstation.

```
sp3en0/ # spacl -a show -s ssp/sysctl -> etc/sysctl.acl -G
sp3n02
{user hosts/sp3n0/self act}
{user hosts/sp3n05/self act}
{group spsec-admin -c-}
{group sysctl-master a-t}
sp3n05
{group spsec-admin -c-}
{group sysctl-master a-t}
sp3n06
{group spsec-admin -c-}
{group sysctl-master a-t}
{group spsec-admin -c-}
sp3n09
{group sysctl-master a-t}
sp3en0
{group spsec-admin -c-}
{group sysctl-master a-t}=  
```

Adding the DCE principal to all `/etc/sysctl.acl` objects in the system, with "at" permissions:
Attempting to add a non-existent principal to an ACL. Unless spac1 is issued with its `-v` (verbose mode) flag, the dcecp version of the command provides more information on the nature of the failure. (Note that `-v` does increase the number of messages received for each host over which spac1 operates. This is in contrast to the single message response of dcecp.)

Displaying what the permissions mean for etc/sysctl.acl.
The preceding examples touch on the notion of ACL permissions. Each DCE ACL object contains zero or more authorization entries. In DCE terms, each entry is comprised of two or three primary parts: privilege attribute (or entry type); the optional specific principal or group to whom the entry applies; the set of permissions associated with a privilege attribute (commonly referred to as permission bits). In different terms, an entry contains who/what has access to a resource and precisely how much authority the who/what has over a resource, not to mention over the ACL object itself.

The list of valid entry types (privilege attributes) and their permission bits that can appear in an ACL are defined by the designers and developers of an application. Sysctl supports fifteen (15) entry types and three (3) permission bits.

The permission bits are represented by a single character when a permission is granted, and by a "-" (dash) when it is not. Only entries that relate to SP administrator principals should have control ("c") authority. This level of authority grants a principal the ability to modify or delete existing entries, and to add new entries, to an ACL. c authority should be viewed as administrative control over the ACL object itself. Access authority ("a") grants access to the resources protected by the ACL. Test authority ("t") allows a principal to run DCE's acl check command against an ACL. (acl check returns to the principal entering the command the permissions granted by the ACL.)

In the example below, the principal dero has access to Sysctl resources protected by etc/sysctl.acl, but cannot alter the entries of etc/sysctl.acl. The principal, hosts/sp3en0/self, can alter the entries of the ACL but doesn't have access to Sysctl resources protected by etc/sysctl.acl. (Of course, with "c" authority the principal can change its own permission bits to include "a.") Principals that are members of the sysctl-users group (a group created by the book's team) only have authority to check their access rights, while members of the spsec-admin group have administrative authority over the ACL object itself. Members of the sysctl-master group have the same access authority as the principal dero.

```
sp3en0/ # dcecp -c acl show ./albeys/ssp/sp3en0/sysctl/etc/sysctl.acl
{user dero a-t}
{user hosts/sp3en0/self -ct}
{group sysctl-users --t}
{group spsec-admin -c}
{group sysctl-master a-t}
```

As the principal dero.
As the principal, hosts/sp3en0/self

Modifying an entry’s permission bits with all dashes ("---") is the same as modifying the entry without explicitly specifying any permission bits. In the absence of an explicit dash or permission bit character, an implied "disallow" (deny) state is assumed, and a dash is added to the missing permission value, as shown in the following screen:

An attempt to add a permission other than the Sysctl defined set will result in an error, and the ACL entry will not be modified.
Sysct1 DCE ACL objects support the following entry types (privilege attributes) and, if they appear in ACL, will be checked in the precedence order shown.

1. mask_obj
2. unauthenticated
3. user
4. group
5. other_obj
6. any_other
7. user_delegate
8. group_delegate
9. other_obj_delegate
10. any_other_delegate

The following entry types are also supported in the ACLs, but can only be added when at least two DCE cells have been configured for DCE intercell communication. The environment for this book did not include intercell configurations, so the entries are not shown in any preference order, or in relationship to the previous ordered list. They are included here for reference purposes only.

- foreign_user
- foreign_group
- foreign_other
- foreign_user_delegate
- foreign_group_delegate

The following ACL entry types are not valid for Sysct1 DCE ACLs, and cannot be added under any conditions.

- user_obj


- group_obj
- user_obj_delegate
- group_obj_delegate

When an attempt is made to add entries for any unsupported entry types, or for foreign_* types when intercell communication has not be configured, error messages are returned, and the ACL is not altered. Likewise, when an attempt is made to add a DCE principal or group that does not exist in, or is not registered with, the current DCE cell, an error message will be displayed, and the ACL entry will not be modified.

```
sp3en0/ # dcecp -c acl modify /./subsys/ssp/sp3en0/sysctl/etc/sysctl.acl -add {user_obj t}
Error: msgID=0x1460101B user_obj ACL entries not allowed in this ACL

sp3en0/ # dcecp -c acl modify /./subsys/ssp/sp3en0/sysctl/etc/sysctl.acl -add {foreign_user /.../itso_cell2/alien_dero t}
Error: msgID=0x1131F383 ACL entry key is not valid.
```

An error results when an invalid entry type is specified (the first field is interpreted as the entry type) or when the entry type appears outside of the curly braces:

```
sp3en0/ # dcecp -c acl modify /./subsys/ssp/sp3en0/sysctl/etc/sysctl.acl -change {hosts/sp3en0/self -ct}
Error: msgID=0x1131F382 ACL entry type is not valid.

sp3en0/ # dcecp -c acl modify /./subsys/ssp/sp3en0/sysctl/etc/sysctl.acl -add user {hosts/sp3n05/self cat}
Error: msgID=0x1131F008 Argument 'hosts/sp3n05/self cat' is not recognized
```

For most sites, the ACL entry types of user and group are probably sufficient. They provide the most flexibility while maintaining a high degree of control. Plus, it is intuitive to manage ACL contents along the lines of DCE users and groups, given their similarities to AIX user and group management concepts.

Entry types of mask_obj and *_delegate are somewhat esoteric, and are of limited value with respect to protecting Sysctl resources. Likewise, an unauthenticated entry is of little use when the Sysctl server is configured to accept only authenticated client requests. (The default configuration.)

The robustness of the ACL entry types (and their associated permission sets) provides a wide range of possibilities for access controls. But, as the possibilities increase, so does the complexity of effectively managing the access controls.
For more information about DCE ACLs, see the section about the acl command in the IBM DCE for AIX Administrators Commands Reference, and also consult the IBM DCE for AIX Applications Development Guide -- Core Components, Part 5. DCE Security Service, Chapter 26, "Authorization", for overviews and definitions of the DCE authorization protocol, ACL types, ACL entries, ACL permissions, and the DCE algorithms used in precedence checking. (It is recommended that SP administrators read all of the chapters under Part 5 because they examine general security concepts, such as identity, authentication, authorization, credentials, and auditing, as implemented in DCE security services.)

Displaying the out-of-box ACL entries of the default Sysctl ACLs.

```bash
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/sysctl.acl
   {group spsec-admin -c-}
   {group sysctl-master a-t}
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/sysctl.mmcmd.acl
   {group sysctl-mmcmd-services a-t}
   {group sysctl-mmcmd a-t}
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/sysctl.vsd.acl
   {group spsec-admin -c-}
   {group sysctl-vsd-services a-t}
   {group sysctl-vsd a-t}
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/logmgt.acl
   {group spsec-admin -c-}
   {group sysctl-logmgt-services a-t}
   {group sysctl-logmgt a-t}
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/rootcmds.acl
   {group spsec-admin -c-}
   {group sysctl-root a-t}
sp3en0/ # dcecp -c acl show /./subsys/ssp/sp3en0/sysctl/etc/pman.acl
   {group spsec-admin -c-}
   {group sysctl-pman a-t}
```

In contrast to dcecp acl commands, the PSSP spacl command is an excellent tool for SP administrators who are not familiar with DCE ACL objects and for those who wish to exploit spacl’s “send ACL request to PSSP DCE security hosts only”, thereby, reducing the need for the administrator to issue individual dcecp commands to all required hosts. (The PSSP manuals, PSSP: Command and Technical Reference, SA22-7351, and PSSP: Administration Guide, SA22-7348, covers spacl syntax, examples, and security information related to the use of the command.)

spacl also permits the administrator to work over SP node groups, lists, or ranges, and/or just the control workstation. (Node groups, lists, and ranges are implemented in the same way that they are supported in splstdata and dsh commands.)
Plus, spacl operations are only focused on PSSP DCE ACL objects. In PSSP V3.2, there are only two PSSP services that implement DCE ACLs: Sysctl (ssp/sysctl) and Hardmon (ssp/hardmon).

Moreover, by default, spacl only displays the results from hosts that responded successfully to the ACL request, while any failed response, from one or more hosts, is displayed as a single "not all actions completed successfully" message. Adding the verbose flag, -v, to any spacl command displays all responses from the target hosts including the hosts for which an action was not successful.

```
sp3en0/ # spacl -a check -s ssp/sysctl -o etc/sysctl.acl
sp3n05: a-t
sp3n06: a-t
sp3en0: --t
spacl: Not all check actions were performed successfully.

sp3en0/ # spacl -a check -s ssp/sysctl -o etc/sysctl.acl -v
spacl -a check -s ssp/sysctl -o etc/sysctl.acl -v
sp3n05: a-t
sp3n06: a-t
sp3en0: --t
spacl: 0016-557 DCE ACL object instance on DCE host sp3n07 not found.
sp3en0: --t
spacl: Not all check actions were performed successfully.
```

As an alternative to the spacl command, an SP administrator can manage Sysctl DCE ACLs using PSSP’s distributed remote shell command, dsh, in conjunction with DCE’s dcecp acl commands. This is particularly useful for administrators who are familiar or comfortable with DCE dcecp commands and/or those that are comfortable with dsh. Unlike spacl, dsh can be used in conjunction with the dshbak program to collapse identical output from more than one host so that identical output is displayed only once. ACL query results that are not identical can be quickly identified along with their hosts.

Also, an administrator with root authority can take advantage of dsh’s node, partition, or system-wide distribution points, root’s access to the special DCE self-host (machine) principal credentials, and DCE’s ability to substitute the results of TCL commands within a dcecp invocation, in order to use dcecp acl commands in the same manner as issuing PSSP or AIX administrative commands via dsh.

Specifically, the root user automatically has access to the DCE self-host (machine) principal credentials on a host. This is true for every SP node running DCE. So, by virtue of being a root user, root can issue a dsh out to all the nodes in the system and issue sysctl whatacls for each local host. When the Sysctl command is executed as root on a node, the Sysctl client uses the
self-host principal credentials on that host to authenticate the request to the server. The whatacls results can then be collected into identical groups by piping the results of dsh to dshbak -c. In one pass, an administrator can see if Sysctl DCE ACLs differ on any set of hosts.

ACL objects are different on one host in the system.

```
sp3en0/ # klist | grep lob
        Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # dsh -avG "sysctl whatacls" | dshbak -c
HOSTS ----------------------------------------------------------------------
spn02
-----------------------------------------------------------------------------
etc/sysctl.acl
tcltime.acl
px-acl.acl
etc/sysctl.mmcmd.acl
etc/sysctl.vsd.acl
etc/logmgt.acl
etc/sysctl.rootcmds.acl
etc/sysctl.pman.acl
HOSTS ----------------------------------------------------------------------
spn05 spn06 spn09
-----------------------------------------------------------------------------
etc/sysctl.acl
etc/sysctl.mmcmd.acl
etc/sysctl.vsd.acl
etc/logmgt.acl
etc/sysctl.rootcmds.acl
etc/sysctl.pman.acl
```

---

**Note**

When Restricted Root Access (RRA) is enabled, this technique is only possible from the control workstation.

Another powerful aspect of dsh is its ability to forward the user's DCE credentials to another service. This is of particular interest to the SP administrator when the entries of a Sysctl DCE ACL must be manipulated on multiple hosts. To accomplish this through a single dsh command invocation, the administrator uses a DCE dcecp command and DCE dcecp command substitution.

An example of such an operation is presented and then examined in detail.
As shown in the example above, the administrator logs into DCE as sp_admin and requests that the credentials be marked forwardable. (This is for use with dsh.) dsh is then issued to the nodes that are the target of the DCE administrative command. dsh, in turn, issues an rsh to a specific host, which, in turn, runs the dcecp command on the target host.

The dcecp acl shown requires a DCE hostname, which is dynamically obtained and substituted through the use of DCE TCL commands. _h is called a DCE convenience variable and contains the DCE hostname on which a dcecp command is executed. The TCL code shown parses the variable from its hosts/{dce-host-name} format into a {dce-host-name}, which is the portion that is required in the acl show command.

When Restricted Root Access (RRA) is enabled, this technique is only possible from the control workstation.

As shown in the example above, the administrator logs into DCE as sp_admin and requests that the credentials be marked forwardable. (This is for use with dsh.) dsh is then issued to the nodes that are the target of the DCE administrative command. dsh, in turn, issues an rsh to a specific host, which, in turn, runs the dcecp command on the target host.

The dcecp acl shown requires a DCE hostname, which is dynamically obtained and substituted through the use of DCE TCL commands. _h is called a DCE convenience variable and contains the DCE hostname on which a dcecp command is executed. The TCL code shown parses the variable from its hosts/{dce-host-name} format into a {dce-host-name}, which is the portion that is required in the acl show command.

Even though the acl show command is used in this example, the same TCL code can be used in an acl modify command, as well as for other DCE acl commands.

The DCE services on the remote host will only permit an acl command to be executed by an authorized DCE principal. For DCE to authorize sp_admin, sp_admin's credentials must be forwarded from sp3en0 (the origination of the request) to each node in the system (the target hosts of the request). This is
accomplished by specifying -D on the dsh invocation. dsh instructs the underlying rsh command to forward the user's credentials to the next command, dcecp, which then uses the credentials for authorization to the local DCE daemons. The results of dsh are piped to the program, dshbak -c, and are displayed in groups of identical results. As shown, the ACL entries of etc/sysctl.acl on sp3n02 are different than on the other nodes.

It is important to note that, for this command construction to work, the entire target command must be wrapped in single quotes (not double quotes), and the "$" (dollar sign) character must be preceded (escaped) with a backslash in order for the TCL code to be successful. These things are not problems but have to do with how UNIX shell environments interpret and handle certain characters.

Finally, this example assumes that root's /.k5login contains an entry for sp_admin; otherwise, the dsh attempt will fail due to authorization. Likewise, if sp_admin's DCE credentials were not obtained with a dce_login -f, the dsh -D command would fail because forwardable credentials could not be generated for the request. If sp_admin cannot obtain forwardable credentials at DCE login due to the forwardabletkt attribute of its account set to "no", then a DCE cell administrator needs to modify the sp_admin account and set the value to yes. By default, DCE principal accounts are created with the forwardabletkt attribute set to "yes", unless directed otherwise.

Here is an example of a DCE cell administrator changing the forwardabletkt attribute of an account. Account attributes cannot be modified via dcecp -c. They must be modified via an interactive dcecp> session or through a dcecp -s {script name} invocation.

```
sp3en0/ # dcecp -c account show dero5 | grep forward
{forwardabletkt: no}
sp3en0/ # dcecp
dcecp> account modify dero -forwardabletkt yes
dcecp> quit
sp3en0/ # dcecp -c account show dero5 | grep forward
{forwardabletkt: yes} 
```

While the Tcl "[string trimleft \$_h hosts/]" part of the previous example looks esoteric, it actually ensures that the DCE hostname of the host on which dcecp is being executed is, in fact, the valid DCE hostname. If your site's DCE hostname and node hostname are identical, a simpler substitution can be used: [exec hostname] in place of [string trimleft \$_h hosts/].
The SP administrator can also use the same substitution approach on single host invocations. The equivalent rsh invocation of the dsh command is also provided.

```bash
sp3en0/ # dsh -avG \'hostname ; echo "------------" ; dcecp -c \"puts [string trimleft\$_h hosts/]\"\' | dshbak
HOST: sp3n02
------------
sp3n02
------------
sp3n02
...
sp3en0/ # dsh -D -avG \'dcecp -c acl show \.:/subsys/ssp/\[exec hostname\]/sysctl/acl\' | dshbak -c
HOSTS -----------------------------------------------------------------------------
spn05 sp3n06 sp3n09
-----------------------------------------------------------------------------
{group spsec-admin -c-}
{group sysctl-master a-t}
...
```

Another way to issue a dcecp acl show for each ACL object known to a Sysctl server is to wrap a few commands in a script, push the script out to the desired hosts, and then run the script remotely (as needed). The prototype script below is a simple implementation of this idea. (A more robust version might include checking the Trusted Services authentication method setting for the string "dce" and checking the states of the Sysctl and DCE daemons.)

```bash
#!/bin/ksh
# Name: sysctl-acl-show
# Description: Simple prototype to issue dcecp acl show for all known Sysctl ACL objects.
# set -A acllst `sysctl whatacls`
for wacl_obj in ${acllst[@]}
  do
    echo "--> ACL entries for $wacl_obj ..."
    dcecp -c acl show \\./subsys/ssp/\$wacl_obj
  done
```

Chapter 8. Problem isolation and resolution  313
Combining elements from the previous examples (dsh, credentials forwardability, and authorization requirements) in conjunction with the prototype script yields the following.

```bash
sp3en0/ # dce_login -f sp_admin
...
sp3en0/ # pcp "-w sp3n05,sp3n06" /tmp/sysctl-acl-show /tmp/sysctl-acl-show
sp3en0/ # dsh -w sp3n05,sp3n06 "ls -l /tmp/sysctl-acl-show"
sp3n05: -rwxr--r-- 1 root system 304 Jan 15 07:50 /tmp/sysctl-acl-show
sp3n06: -rwxr--r-- 1 root system 304 Jan 15 07:50 /tmp/sysctl-acl-show
sp3en0/ # dsh -D -w sp3n05,sp3n06 "/tmp/sysctl-acl-show" | dshbak -c
HOSTS ---------------------------------------------------------------
sp3n05   sp3n06
---------------------------------------------------------------------
--> ACL entries for etc/sysctl.acl ...
{group spsec-admin -c-}  
{group sysctl-master a-t}
--> ACL entries for etc/sysctl:mmcmd.acl ...
{group spsec-admin -c-}  
{group sysctl:mmcmd a-t}
--> ACL entries for etc/sysctl:vsd.acl ...
{group spsec-admin -c-}  
{group sysctl:vsd-services a-t}  
{group sysctl:vsd a-t}
--> ACL entries for etc/logmgt.acl ...
{group spsec-admin -c-}  
{group sysctl:logmgt-services a-t}  
{group sysctl:logmgt a-t}
--> ACL entries for etc/sysctl:rootcmds.acl ...
{group spsec-admin -c-}  
{group sysctl:rootcmd a-t}
--> ACL entries for etc/sysctl:pman.acl ...
{group spsec-admin -c-}  
{group sysctl:pman a-t}
```

**Sysctl DCE ACL manager**

To access Sysctl DCE ACL objects requires an active Sysctl daemon whose DCE ACL manager is also running. By default, the Sysctl server's DCE ACL manager is automatically started when the server is initialized in a dce-enabled security mode (dce or dce:compat). The SP administrator cannot directly start or stop Sysctl's ACL manager.

When the ACL manager does not start properly, the following should be expected:

- An error is written to the Sysctl log (provided the server is started with its -ds flags)
- The Sysctl server cannot access the DCE ACL objects in its DCE ACL database.
• The Sysctl server may hang.

Likewise, the following commands may hang when directed to the Sysctl server whose ACL manager is not running:
• DCE dcecp ACL commands
• PSSP spacl (directed at Sysctl DCE objects)
• Sysctl aclcheck and whatacl

Some ACL manager problems are easily corrected. For example, if the local CDS client is not running at the time Sysctl starts, stop Sysctl, start the needed DCE clients, and then start Sysctl. Likewise, when the CDS master server is not running, when Sysctl starts (Sysctl will not be able to export its RPC bindings to the CDS), stop Sysctl, start the needed DCE servers, and then start Sysctl. Or, should the DCE client cache become corrupt on a host, stop Sysctl, stop the DCE clients, run the DCE clean_up.dce to clean-up the corruption, start the DCE clients, then start Sysctl.

Sysctl ACL Manager not started due to the local DCE clients not running. Sysctl client requests results in a connection refused message.

```
sp3n06/ # show.cfg
Gathering component state information...

Component Summary for Host: sp3n06

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
</tbody>
</table>

The component summary is complete.

sp3n06/ # lssrc -s sysctld
Subsystem Group PID Status
sysctld inoperative

sp3n06/ # startsrc -s sysctld -a '-d'
0513-059 The sysctld Subsystem has been started. Subsystem PID is 4508.

sp3n06/ # lssrc -s sysctld
Subsystem Group PID Status
sysctld 4508 active

sp3n06/ # sysctl whoami -v
sysctl: 2501-018 Connection refused

sp3n06/ # cat /var/adm/SPlogs/sysctl/sysctld.log
...
Nov 21 13:20:09 Server starting (pid=4508)
Nov 21 13:20:10 svc_set_up_DCE: spsec_start_acl_mgr failed: 2502-606 DCE error in rpc_ep_register: Connection request rejected (dce / rpc)
...
```
The corrective action is to stop Sysctl, start the DCE clients, and then start the Sysctl server.

Sysctl ACL Manager not started due to the master CDS not running. Sysctl client requests result in a Connection refused message.

The corrective action is to Stop Sysctl, start the DCE master CDS server, and then start the Sysctl server.

However, should the DCE ACL database files become corrupt, or should some files be deleted, the only corrective action is to stop Sysctl, delete the database files, and start Sysctl again so that it will re-create the database files with the default Sysctl ACL entries. (Refer to an earlier subsection for the default values.)

Unfortunately, this recovery destroys any custom modifications made to any of the ACL entries on that host. For example, if the SP administrator added
the principal dero5 to etc/sysctl.acl, once Sysctl re-creates the database files, the dero5 entry no longer exists in etc/sysctl.acl.

Therefore, it is imperative to maintain accurate records that document the ACL entries for each Sysctl DCE ACL object on each host in the unlikely event that a local Sysctl database must be deleted and re-created due to corruption.

---

**Note**

Sysctl *does* re-create all ACL objects (even custom ACL objects) that are referenced by its /etc/sysctl.conf file at initialization time. However, as previously stated, it cannot populate the ACLs with any custom entries. Instead, the ACLs will be populated with the same default ACL entries as etc/sysctl.acl.

The following is an example of the steps needed for a Sysctl ACL database that had to be deleted and then re-created.

1. The state of things before the database corruption is this: One custom ACL, show-syscd-dce-info.acl, and two ACLs containing customer-changed entries.
2. At some point after the corruption takes place, Sysctl client requests fail. Examining the log reveals that Sysctl's ACL manager didn't start during the server's last initialization.

3. The corrective action is to Stop Sysctl, delete the ACL database files, and start Sysctl.
Chapter 8. Problem isolation and resolution

The custom ACL entry definitions for show-syscd-dce-info.acl and etc/sysctl.acl are gone.

Synchronizing Sysctl Kerberos V4 and DCE ACLs

There is no PSSP tool that synchronizes entries between Kerberos V4 and DCE ACLs. There are several reasons for this. First, the tool would not know if an entry in a Kerberos V4 ACL should automatically be a member of a DCE ACL. Second, there is no way to guarantee that there is a corresponding DCE principal for every Kerberos V4 principal. Third, given that Sysctl ACLs exist on a per-host basis, it is possible that the ACL entries between the control workstation and nodes, or among and between nodes, are, intentionally, not the same. Finally, given that Sysctl runs on all nodes in an SP system, and the nodes can be divided into multiple partitions, each running with a different security setting, one host may only have DCE ACLs while another host only has Kerberos Version 4 ACLs.
It is up to the SP administrator to ensure that the entry definitions between DCE and Kerberos V4 ACLs of the same name are synchronized. Likewise, the SP administrator must also synchronize DCE and Kerberos V4 ACLs among and between different hosts in the SP. However, as stated earlier, the PSSP spacl command is an excellent choice to synchronize all Sysctl DCE ACLs in a system, while the use of File Collections or pcp is an excellent way to refresh all Sysctl Kerberos V4 ACL files on the nodes.

In general, the requirement to change ACLs should be an infrequent one. (Notable exceptions are prototyping and test environments.) Given that ACLs are control points in the system, it is imperative that they remain as stable as possible. Still, the need to alter ACL contents at some point is a reality. One way to ensure that ACL management happens in a timely and controlled manner is to add triggers to AIX user account, Kerberos V4 principal, and DCE principal creation/deletion procedures. When a user account/principal is created, Sysctl ACL membership consideration should be part of that process. When a user account/principal is deleted, Sysctl ACL membership removal should be part of that process. This will keep ACL entries from becoming stale or invalid.

**Hardmon**

The system hardware monitor (Hardmon) allows authorized users to control and monitor the status of the SP system's frames, nodes, and switch via the SP supervisor subsystem. In addition to retrieving status information about the processors, the monitor allows authorized users to perform operations, such as controlling the power and the logical key switch position for a node processor.

Hardmon protects its resources, SP hardware (nodes and frames), through DCE group and DCE ACL memberships (in a dce only mode), Kerberos V4 ACL membership (in a compat only mode), or a combination of DCE and Kerberos V4 controls (in a dce:compat security mode).

There is one out-of-box Hardmon Kerberos V4 ACL file provided by PSSP: /spdata/sys1/spmon/hmacls. This is a plain text file.

There are five Hardmon DCE groups configured by PSSP services used to grant/deny access under a dce security mode. The groups can be displayed by issuing the following:

dcecp -c group cat | grep hm-

There are two out-of-box default Hardmon DCE ACLs provided by PSSP: hardmon and system. (Additional Hardmon DCE ACL objects can be created. Refer to the **PSSP Administration Guide**, SA22-7348, for complete details.)
Displaying Default Hardmon DCE ACLs

In the first sequence, a root user running in a dce-only security mode attempts to use s1term in write mode but is denied due to lack of DCE group membership in a required PSSP DCE group, hm-control. After the cell administrator adds the DCE principal to the needed group, the user must destroy and create new DCE credentials in order to reflect the new membership. In the case of a root user holding the DCE self-host principal credentials, that is, the DCE machine principal context/identity, DCE (or just its security client, sec_cl) must be stopped and then started on the host (node or control workstation) in order for the self-host principal's credentials to reflect the new group membership. (User membership in DCE groups is reflected through the user's DCE credentials and is considered part of the user's ticket privilege attributes. The credentials, also called tickets, can only reflect group memberships that were in place at the time the user's credentials were created. Any group associations established after user credentials were generated will not be reflected until new credentials are generated.)

In order to successfully run the DCE acl show command against the default Hardmon DCE ACLs, the user's DCE credentials must have, at least, read authority to those ACLs; otherwise, the permission is denied. (The message returned from DCE says permission not valid for this acl, which means the DCE principal is not authorized to view the ACL.) To facilitate this effort, it is recommended that a unique DCE principal be created for PSSP DCE administrative purposes and that it then be made into a member of the PSSP group, spsec-admin, created during PSSP DCE security installation and configuration. spsec-admin is automatically added as a member to the default Hardmon DCE ACLs at configuration time. In the previous example, the DCE principal, sp_admin, was created and added to various PSSP DCE groups, one of which is spsec-admin.
created. For more information on DCE credentials (tickets) and their privilege attributes, refer to IBM DCE Version 3.1 for AIX and Solaris: Administration Guide -- Core Components, Chapter 31. “Creating and Maintaining Accounts”, subsection “Displaying Privilege Attributes and Tickets”.

Once this takes place, the user can s1term out to a node in write mode. Note that in this example, Hardmon DCE ACLs do not play a role in the authorization of the user. Authorization is based solely on membership in a Hardmon DCE group.

```
sp3en0/ # whoami ; klist | grep lob
root
Global Principal: /.../sp_cell/dero
sp3en0/ # s1term -w 1 5
s1term: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # dcecp -c prin show dero | grep groups
{groups spsec-admin sdr-admin}
```

As a cell administrator, add the DCE principal dero to the hm-control group.

```
sp3en0/ # dcecp -c group list hm-control
/.../sp_cell/sp_admin
sp3en0/ # dcecp -c group add hm-control -member dero
sp3en0/ # dcecp -c group list hm-control
/.../sp_cell/sp_admin
/.../sp_cell/dero
```

As the DCE principal dero, the command still fails, even though dero was added to the hm-control group. The user must destroy the current credentials and create new ones.

```
sp3en0/ # s1term -w 1 5
s1term: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # kdestroy ; exit
sp3en0/ # dce_login dero
...
DCE LOGIN SUCCESSFUL
...
sp3en0/ # s1term -w 1 5
...
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login:
...
```
In the second sequence, a non-root user running in a dce:compat security mode attempts to use s1term (in write mode), but is denied because the user's DCE principal is not a member of the PSSP DCE P group, hm-control, nor is the user's Kerberos V4 principal in the hmacls file. After the SP administrator adds the user's Kerberos V4 principal to hmacls and issues the hmadm setacls command, the user can then issue s1term successfully. Note that the non-root user's DCE principal membership has not changed.

root adds the dero-sec principal to the hmacls file (writable by root only), and then issues hmadm setacls to update the Hardmon daemon's internal ACL table.

In the above example, the principal is added to hmacls with a permission of s, the ability to read and write to a serial port (s1term). (There are other hmacls permissions. Refer to the PSSP Administration Guide, SA22-7348, for complete details.) Since hmacls is a file that only root can update, an SP administrator (or other root user) must make the needed changes.

Once hmacls is updated, the hmadm setacls command must be invoked so that the Hardmon daemon's internal ACL tables are updated to reflect the new hmacls entry. If this step is omitted, the daemon's tables will not be updated, and the user will not be able to issue s1term. However, to run hmadm with the...
setacls operand, the issuer of hmadm must have Kerberos V4 administrator credentials. (hmadm can be invoked by any user, but to run it with setacls requires a specific level of authorization - a Kerberos V4 administrative identity. The Kerberos V4 administrator ID is the ID defined as the primary authentication services administrator by the setup_authent command during compat installation and configuration. In this case, it was root.admin. Refer to the PSSP installation and administration guides for complete details.)

In this scenario, the administrative duties required both AIX root level authority and Kerberos V4 administrative authority. SP administrators may be one in the same. The same concept applies to SP administrators and DCE controls. Manipulating DCE organization, group, and ACL memberships requires DCE cell administration authority to varying degrees, while doing other SP administrative tasks requires root authority, such as issuing the AIX command startsrc. The SP administrators can be DCE administrators, or they can have their DCE principal grant the needed DCE administrative authority needed for PSSP DCE controls. Or, perhaps, the DCE cell administrator is an individual other than an SP administrator.

Another important aspect of this example is the way in which authorization is granted. Unlike the Hardmon dce-only example, membership in a Kerberos V4 ACL is required instead of membership in a particular PSSP DCE group. This is an important distinction in understanding the general differences between the dce and compat PSSP security modes for Hardmon.

Under a dce mode, DCE group membership and/or DCE ACL membership are the hurdles to overcome before authorization is granted. Under a compat mode, Kerberos V4 principal membership in hmacls, coupled with the permissions associated with the Kerberos V4 principal, are the control factors. And, in a dce:compat mode, should the DCE controls deny access, the compat (Kerberos V4) controls can grant access, provided the user meets the compat requirements, as the preceding example reveals.

Another way to view this contrast is the following: Kerberos V4 has no “group” structure. A Kerberos V4 principal exists within a Kerberos V4 realm. The ability to grant/deny access to these principals is based on whether or not they appear in hmacls, as shown in this last case. DCE, on the other hand, not only has principals, but a principal is a member of an organization and one or more groups. Within a DCE cell, multiple organizations and multiple groups may exist. As such, the ability to grant/deny access is based on one or more combinations of organization, group, and ACL memberships. The first Hardmon example demonstrated DCE group membership requirements. The next example demonstrates authorization based on Hardmon DCE ACL membership.
Principal dero attempts to run an `hmcmds` command but is denied due to authorization failures. dero is not a member of the DCE ACL, “system”, nor is the principal a member of the PSSP DCE group, hm-monitor. The principal also doesn't have permission to list the entries in the system ACL (via `dcecp acl`), or to list what hardware access permissions are granted to dero through `hmgetacls`. (In this example, the default Hardmon hardware DCE ACL, system, is the only ACL in use for all the resources in the system.)

```
sp3en0/ # klist | grep lob
Global Principal: /.../sp_cell/dero
sp3en0/ # hmcmds -G flash 1:1
hmcmds: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # dcecp -c prin show dero | grep group
   {groups none sdr-admin}
sp3en0/ # hmgetacls 1:1-1
   {frame/slot1 - - - -}
Error: msgID=0x17122025 permission not valid for this acl
```

The SP administrator adds dero to the system ACL, with `v` and `t` permissions. `v` allows dero to issue `hmcmds` commands, while `t` allows the principal to display the contents of the system ACL via `dcecp` and to list his/her ACL permissions with `hmgetacls`.

```
sp3en0/ # hmdceobj -q
   system
   hardmon
sp3en0/ # dcecp -c acl show ./:subsys/sp3en0/hardmon/system
   {group spsec-admin -c-----}
   {group hm-control --vsmu-}
   {group hm-control-services --vsmu-}
   {group hm-monitor ----m--}
   {group hm-monitor-services ----m--}
sp3en0/ # dcecp -c acl modify ./:subsys/sp3en0/hardmon/system -add {user dero tv}
sp3en0/ # dcecp -c acl show ./:subsys/sp3en0/hardmon/system
   {user dero --v---t}
   {group spsec-admin -c-----}
   {group hm-control --vsmu-}
   {group hm-control-services --vsmu-}
   {group hm-monitor ----m--}
   {group hm-monitor-services ----m--}
```

The dero principal can now run the `hmcmds`, `hmgetacls`, and `dcecp acl` commands successfully. (The principal does not have to obtain new DCE credentials to be recognized by Hardmon's system ACL. This would only be required if the SP administrator added dero to the PSSP DCE group, hm-monitor.)
When the user has neither DCE credentials nor Kerberos V4 credentials, Hardmon cannot authenticate the user and, therefore, cannot authorize the user; so, the request is denied.

In a dce:compat security mode, the user needs only one authenticated identity in order to be authorized by Hardmon ACL controls. As long as one identity is authenticated and authorized, no error messages are generated in response to the unauthenticated identity as shown next.

The user does not have a valid DCE identity but has a valid Kerberos V4 identity. The Kerberos V4 identity is authorized with "m" (monitor) permission in hmacls.
Querying and managing Hardmon ACLs

Determining which users require access to Hardmon ACL protected resources is critical to ensuring the security and integrity of SP nodes and the control workstation. A well-defined policy that outlines which Hardmon resources are available to the user population is the first level of control. The second is a documented procedure that users follow to request access to Hardmon resources. Once it is determined that a user's DCE and/or Kerberos V4 principal will be added to, removed from, or changed within Hardmon ACLs, an SP administrator must then take the needed steps to manage Kerberos V4 and/or DCE ACLs. Managing Hardmon ACLs is achieved through the use of PSSP, DCE, and basic UNIX commands.

All of the concepts presented in the “Querying and Managing Sysctl ACLs” subsection can be applied to Hardmon ACL objects. Limiting read/write access to Hardmon’s Kerberos V4 ACL file and its DCE ACL objects is critical. Only SP administrator accounts (AIX, DCE, Kerberos V4) should have access to Hardmon’s ACLs. Since Hardmon and Sysctl share the same ACL concepts, the Sysctl material is not repeated here for Hardmon. Refer to the Sysctl subsection for details and examples on why limiting access to ACLs is critical.

The remainder of this subsection focuses on the ACL tools and authorization items particular to Hardmon.

Hardmon ACL permissions
Hardmon’s DCE and Kerberos V4 ACLs contain permission bits, which are nearly identical in meaning and presentation between its DCE and Kerberos V4 ACL types. This is in sharp contrast to Sysctl ACLs where permission bits exist only for its DCE ACLs.

Both Hardmon ACL types support the following permissions:

- **a** - permission to perform administrative functions
- **v** - permission to control hardware through the VFOP (virtual front panel operator) interface
- **s** - permission to open a serial connection (s1term)
- **m** - permission to monitor hardware state data

DCE ACLs support the following additional permissions:

- **c** - permission to modify ACL entries
- **u** - permission to update SP supervisor microcode
- **t** - permission to check access rights of an ACL

A microcode update permission bit is not supported in Kerberos V4 ACLs. However, **v** permission is needed in a Kerberos V4 ACL for microcode updates.

**Hardmon Kerberos V4 ACL**

The name of the PSSP shipped (default) Kerberos V4 ACL is /spdata/sys1/spmon/hmacls:

The ACL entries in this file reflect access to hardware on a per-frame basis.

There is one PSSP command to check if authorization permission exists in an ACL, one PSSP command to display the permissions associated with an ACL entry (denoted with "+"), and one PSSP command to instruct the Hardmon daemon to refresh its cached authorization tables (denoted with "**").

- hmckacls
- * hmgetacls
- ** hmadm - Reads the Hardware Monitor access control list configuration file to update the daemon’s internal ACL tables

hmckacls and hmgetacls are the only commands in the list that work with both Kerberos V4 and DCE ACLs. hmadm is only valid for use with Kerberos V4 ACLs.
hmadm has AIX permissions of 555 (-r-xr-x), meaning any user can execute the command, but in order to send the setacls directive to Hardmon, the invoker must be authorized in hmacls with administrative permission. The hmadm setacls command instructs the Hardmon daemon to update its internal ACL tables by rereading the ACL configuration file, hmacls. When an SP administrator makes changes to hmacls, but doesn't issue hmadm setacls afterward, Hardmon will not be aware of the ACL changes.

The following screen shows a non-root user with valid Kerberos V4 credentials, but lacking proper Hardmon authorization:

```
/u/dero/ # k4list | grep pal:
sp_admin
/u/dero/ # grep sp_admin /spdata/sys1/spmon/hmacls ; echo $? 1
/u/dero/ # hmadm setacls
hmadm: 0026-614 You do not have authorization to access the Hardware Monitor.
```

After the SP administrator adds sp_admin to hmacls (with the administrative permission "a") and issues hmadm setacls, the non-root user can issue hmadm setacls successfully.

```
/u/dero/ # k4list | grep pal:
sp_admin
/u/dero/ # grep sp_admin /spdata/sys1/spmon/hmacls ; echo $? 0
sp3en0 sp_admin a 0
/u/dero/ # hmadm setacls
hmadm: 0026-641I The ADMIN command "setacls" was sent.
```

**Note**

Users other than SP administrators should not have "a" authority to Hardmon because this level of authority grants the principal the ability to administer the Hardmon daemon. The daemon is used to monitor and control the SP hardware. The example merely illustrates that Hardmon Kerberos V4 ACLs control the grant/deny authority for hmadm setacls.

Like hmadm, hmckacls and hmgetacls have AIX permissions of 555 (-r-xr-x), which means that any user can execute the command, but in order to use the commands, the invoker must have an authenticated Kerberos V4 identity and be authorized in hmacls.

Kerberos V4 principal is not in hmacls. Output contains all dashes in output.
The SP administrator adds the principal to hmacls and then issues hmadm setacls. Hardmon then recognizes the Kerberos V4 principal.

In a dce:compat mode, where the invoker's DCE identity is not authorized to Hardmon or the DCE identity cannot be authenticated, hmgetacls and hmckacls will be authorized when the user's Kerberos V4 identity is in hmacls.

PSSP does not have a tool to add, delete, or modify the contents of hmacls. The SP administrator edits hmacls and makes the appropriate changes. The structure of the hmacls file, including definitions on how entries need to appear and additional information on permission bits, are documented in chapter 2 of the *PSSP: Command and Technical Reference*, SA22-7351.
**Hardmon DCE ACLs**

The names of the PSSP-shipped (default) DCE ACLs are:

- hardmon
- system

There are four PSSP commands and one DCE command used to manage, display, and check Hardmon DCE ACL objects.

- `acl` - DCE dcecp command that, in conjunction with its qualifiers (check, show, modify, et al.), allows authorized DCE principals to display and alter Hardmon ACL object entries.
- `hmckacls` - PSSP command that checks if an authorization permission exists in an ACL.
- `hmdceobj` - PSSP command that adds, deletes, and lists all hardware monitor DCE objects known to Hardmon.
- `hmgetacls` - PSSP command to display the permissions associated with an ACL entry.
- `spacl` - (SP ACL Management) PSSP command to display and alter PSSP DCE ACL object entries (for Hardmon and Sysctl) by authorized users. The command works only with PSSP DCE ACLs and does not create/delete ACL objects.

Hardmon DCE ACLs are not stored in an individual file, as is the case with its Kerberos V4 ACL file. The DCE ACL objects are part of a DCE ACL database that the Hardmon server manages. The database exists only on the control workstation and is, actually, made up of three files, `db_acl`, `db_name`, and `db_object`, which are stored under `/spdata/sys1/spmon/hmdceacls`. (These files are stored in a binary format and cannot be edited directly.) All Hardmon and dcecp client requests that view or modify the contents of an ACL object are actually performed through the Hardmon daemon's DCE ACL manager. The Hardmon daemon's DCE ACL manager is transparent to all client requests, including `dcecp acl` commands, but it is, ultimately, responsible for reading/writing the Hardmon DCE database files.

In order to effectively use the PSSP or DCE ACL management commands, some explanation of what a Hardmon DCE ACL object "looks like" is in order.

Whether `hmdceobj`, `hmgetacls`, `hmckacls`, `spacl`, or `dcecp` commands are used to access Hardmon DCE ACL objects, the objects are ultimately referenced as fully-qualified DCE Cell Directory Service (CDS) names consisting of the cell name (or the string `./:` to denote the local cell), the CDS directory path for the
DCE RPC entry for the Hardmon server, and the object name itself. For Hardmon, the CDS path to the Hardmon server is always on the control workstation and is represented here as 
/./subsys/ssp/cwsDCEhostname/hardmon.

The object name, appended to the CDS path, is one of the two default Hardmon DCE object names that were listed earlier or taken from custom frame/slot definitions.

For example, a fully-qualified CDS name for a default Hardmon ACL object would be similar to /./subsys/ssp/DCEhostname/hardmon/system.

The common mistake when managing Hardmon ACLs through dcecp is to use only the DCE object name instead of the fully-qualified CDS name; system instead of /./subsys/ssp/cwsDCEhostname/hardmon/system. When managing Hardmon DCE ACLs via dcecp, using anything other than a fully-qualified name will result in an error.

Actually, this is only “mostly” true. A user with DCE cell administrator authority can actually manipulate the ACL of /./subsys/ssp/cwsDCEhostname/system through dcecp acl, provided the modifier -entry is specified. (spacl does not permit this type of access, however.) A Hardmon server stores its network location information in a server entry in the CDS. (Network location information is also referred to as RPC end-points in DCE terms.) The server entry in the CDS has the same name as the server itself, and an associated CDS ACL. The acl command's -entry option is used to operate on the server entry ACL in CDS rather than the server's ACL objects. In general, the SP administrator should never need to manipulate Hardmon's ACL entry in the CDS.

To be able to manage each of the Hardmon DCE ACL objects, the administrator's DCE principal must be a member of the PSSP DCE group, spsec-admin. To facilitate this effort it is recommended that a unique DCE principal be created for PSSP DCE administrative purposes, and then make this principal a member of the spsec-admin group. This group is created by the PSSP DCE security installation and configuration routines; spsec-admin is automatically added as a member to the default Hardmon DCE ACLs.

The Hardmon DCE ACL manipulation examples in this section were done by a DCE principal called sp_admin. sp_admin was added to the group spsec-admin. It was also added to the Hardmon group hm-control. Unless such a principal exists, the administrator will not be permitted to view or modify Hardmon DCE ACL objects.
Hierarchical DCE objects and ACLs

Hardmon allows authorized users to control and monitor the status of the SP system's frames, nodes, and switch via the SP supervisor subsystem. In addition to retrieving status information about the processors, the monitor allows authorized users to perform operations, such as controlling the power and the logical key switch position for a node processor.

Hardmon protects its resources - SP hardware (nodes and frames) - through DCE group and DCE ACL memberships (in a dce-only mode), Kerberos V4 ACL membership (in a compat only mode), or a combination of DCE and Kerberos V4 controls (in a dce:compat security mode).

The objects defined and controlled by the Hardmon daemon are the:

- system
- frame
- slot (containing hardware, such as a node or switchboard)
- Hardmon (system monitor daemon)

The system object is the initial object and is a container for the frame and slot objects. The monitor allows authorized users to perform monitor and control operations on frame and slot objects. The Hardmon daemon object represents the administration functions of Hardmon.

An ACL may be associated with each monitor object. A Hardmon administrator can grant access to monitor and control the system through the ACLs or through a combination of ACLs and group memberships.

The objects controlled by Hardmon are hierarchical; the system contains frames that contain slots (with nodes and switch processors). Administrators may not wish to manage the (potentially) hundreds or thousands of ACLs that may be required to specify user access for each frame, node, and switch processor in a system. Since the object structure is inherently hierarchical, the ACL associated with an object is the default for lower-level (child) objects if they don't have an overriding ACL. For example, if an ACL is associated with a frame, that same ACL would apply, by default, to all slots contained within that frame. However, if an ACL is associated directly with a slot, that ACL would be used to determine access for the slot.

With this hierarchical structure of objects and ACLs, each installation can determine the level of granularity to be used for specifying access to system monitor objects.
**Hardmon DCE ACL management/query command examples**

Listing all DCE objects known to hardmon as an authorized and unauthorized user.

```
sp3en0/ # hdmdceobj -q
system
hardmon
frame1/slot1
frame2/slot1
frame4
frame4/slot1
...
sp3en0/ # hdmdceobj -q
hdmdceobj: 0026-614 You do not have authorization to access the Hardware Monitor.
```

As sp_admin, display the permission bits of the system object via the hmgetacls, dcecp, and spacl commands in a system where there are no frame/slot DCE ACL objects defined.

```
sp3en0/ # hdmdceobj -q
system
hardmon
sp3en0/ # hmgetacls: 1:1-1
frame1/slot1  v s m u
sp3en0/ # dcecp -c acl show /./:subs/ssp/sp3en0/hardmon/system
{group spec-admin -c-----}
{group hm-control --vsmu-}
{group hm-control-services --vsmu-}
{group hm-monitor ----m--}
{group hm-monitor-services ----m--}
sp3en0/ # spacl -a show -s ssp/hardmon -o system -v
spacl -a show -s ssp/hardmon -o system -v
sp3en0
{group spec-admin -c-----}
{group hm-control --vsmu-}
{group hm-control-services --vsmu-}
{group hm-monitor ----m--}
{group hm-monitor-services ----m--}
spacl: All show actions were performed successfully.
```

As sp_admin, display the permission bits of the frameX/slotY object via hmgetacls, dcecp, and spacl commands in a system where frame/slot DCE ACL objects are defined.

```
sp3en0/ # hdmdceobj -q
system
hardmon
sp3en0/ # hmgetacls 1:1-1
frame1/slot1  v s m u
sp3en0/ # dcecp -c acl show /./:subs/ssp/sp3en0/hardmon/system
{group spec-admin -c-----}
{group hm-control --vsmu-}
{group hm-control-services --vsmu-}
{group hm-monitor ----m--}
{group hm-monitor-services ----m--}
sp3en0/ # spacl -a show -s ssp/hardmon -o system -v
spacl -a show -s ssp/hardmon -o system -v
sp3en0
{group spec-admin -c-----}
{group hm-control --vsmu-}
{group hm-control-services --vsmu-}
{group hm-monitor ----m--}
{group hm-monitor-services ----m--}
spacl: All show actions were performed successfully.
```
sp_admin adds user principal dero1 to only the frame2/slot13 ACL.

DCE principal dero1 has no administrative authority over the Hardmon daemon, or the frame2/slot13 ACL object, but has s1term, VFOP, and monitor authority to the hardware in frame2/slot13, as well as test permission to the ACL object itself.
The PSSP command, hmdceobj, is an administrative command intended for SP administrators and not as a tool for general users. hmdceobj -q requires that the invoker's DCE principal be a member of the PSSP DCE group, hm-admin. This requirement is an hmdceobj command restriction. Only SP administrative DCE principals should have membership in the hm-admin group, considering that principals whose membership includes hm-admin have the authorization to add/delete/modify Hardmon DCE ACL objects.

Non-SP administrative DCE principals that require the ability to list the contents of a Hardmon ACL object should be added to the ACL object as an explicitly-named user entry with a permission of “t” only. Such principals are then authorized to issue a dcecp acl show command on the ACL contents, but have no authority to manipulate (or manage) the ACL’s contents. Whether or not general users should have read authority to these objects should be evaluated carefully.

From one perspective general read access to the ACL objects does not compromise system integrity. They do not contain password data, user profile information, or access protocols. From another perspective, the objects do contain sensitive data in that they list the exact principals that do/do not have authorization to Hardmon resources. A hacker can use this information to scan for credential files that match specific principals, steal those credentials,
and then attack resources that are accessible through Hardmon (depending on the stolen principal’s membership in the ACLs).

While this might sound like an unrealistic scenario, keep in mind that some security compromises are clandestine, stealth attacks. “Closing” otherwise “open” data, like Hardmon ACL information, only helps to reduce risk points. Bottom line: If there’s no legitimate reason to make data read accessible by general users, restrict it. In the end any data that can be gathered about a system can eventually be used to build attack profiles, for social engineering, and to map out possible system weaknesses.

Over all, the hmdceobj command is an important command for SP administrators when working in a dce-enabled security mode. (For complete details on the command, refer to the PSSP: Command and Technical Reference, SA22-7351.) In fact, it’s the only interface available to the SP administrator that is capable of adding or deleting ACL frame, or frame/slot, objects to and from the Hardmon DCE ACL database. This is in sharp contrast to the Sysctl DCE ACL object creation/deletion functions, which are performed indirectly by the administrator.

It must be noted, however, that hmdceobj’s ability to add an ACL entry is limited to the DCE group type only. That is, hmdceobj only allows the administrator to add DCE group entries in an ACL. All other valid ACL entry types must be added to an ACL object either through the DCE dcecp acl command, or the PSSP spacl command. Hardmon DCE ACL objects share the same set of valid ACL entry types as those of Sysctl. (Refer to an earlier subsection on Sysctl for complete details.)

Precisely how DCE Hardmon ACLs should be set-up in an SP system comes down to control versus manageability.

Hardmon DCE ACLs can apply to either a high-level system view (all nodes in all frames), or a granular frame/slot view (a particular node in a particular frame). The level of granularity needed depends on several things, not the least of which is the complexity of managing the ACLs. SP systems with many frames, or several administrators, may find it more logical to define Hardmon ACL objects in terms of frame/slot definitions for maximum access control, or to limit certain administrator principals to various sets of frame/slot definitions. If the system only contains a few frames (or one to two frames), or if there is only a small number of administrators (perhaps even one or two), building and managing Hardmon ACLs from a system-wide, non-granular view may prove to be the best approach.
Either way, the ability to move from system-wide ACLs to granular ACLs, or to collapse granular ACLs into a system-wide view is always possible. The bottom line is to make the ACLs fit the scope of the resources they protect while balancing them against manageability.

**Hardmon DCE ACL Manager**

To access Hardmon DCE ACL objects requires an active Hardmon daemon whose DCE ACL manager is also running. By default, the Hardmon server’s DCE ACL manager is automatically started when the server is initialized in a dce-enabled security mode (dce or dce:compat). The SP administrator cannot directly start or stop Hardmon’s ACL manager.

When the ACL manager does not start properly, an error is written to the currently active Hardmon log; the Hardmon server cannot access the DCE ACL objects in its DCE ACL database, and the following commands will generate DCE errors or SP Security Services errors: DCE `dcep acl` commands, PSSP `spacl` (directed at Hardmon DCE objects), `hmdceobj`, `hmckacls`, and `hmgetacls`.

```plaintext
sp3en0/ # hmdceobj -q
0026-412 User cannot be authenticated on sp3en0.
  SP Security Services error code: 0.
  SP Security Services error message: hmdceobj: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # hmgetacls 1:1-3
0026-412 User cannot be authenticated on sp3en0.
  SP Security Services error code: 0.
  SP Security Services error message: hmgetacls: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # dcep -c acl show /.:/subsys/ssp/sp3en0/hardmon/system
Error: msgID=0x1712201A acl object not found
```

Some ACL manager problems are easily corrected. For example, if the local CDS client is not running at the time Hardmon starts, stop Hardmon, start the needed DCE clients, and then start Hardmon. Likewise, when the CDS master server is not running when Hardmon starts (Hardmon will not be able to export its RPC bindings to the CDS), stop Hardmon, start the needed DCE servers, and then start Hardmon. Or, should the DCE client cache become corrupt on the control workstation, stop Hardmon, stop the DCE clients, run the DCE command, `clean_up.dce`, to clean-up the corruption, start the DCE clients, and then start Hardmon. (Refer to the **Sysctl ACL Manager** section for corrective action examples.)

However, should the DCE ACL database files become corrupt, or if some files have been deleted, the only corrective action is to stop Hardmon, delete the 

---

338 Exploiting RS/6000 SP Security: Keeping It Safe
database files, and start Hardmon so that it will re-create the database files with the default Hardmon ACL entries. (Refer to an earlier subsection for the default values.)

Unfortunately, this recovery destroys any custom modifications made to any of the ACL entries on that host. For example, if the SP administrator added the principal, dero5, to the hardmon ACL, once Hardmon re-creates the database files, the dero5 entry no longer exists in the hardmon ACL. Further, any frame or frame/slot ACL objects created by an SP administrator will not be re-created. Such objects, if they exist, are created exclusively by SP administrators.

Therefore, it is imperative that accurate records be maintained documenting the ACL entries for each Hardmon DCE ACL object (both default Hardmon DCE ACLs (hardmon and system) and any custom frame/slot ACLs) in the unlikely event that the Hardmon DCE ACL database must be deleted and re-created due to corruption.

The following is an example of a Hardmon ACL database that had to be deleted and then re-created.

Part 1: The state of things before the database corruption: One custom ACL, frame1/slot, and one default ACL, hardmon, containing customer-changed entries.

```bash
sp3en0/ # hmdceobj -q
system
hardmon
frame1/slot1
sp3en0/ # dcecp -c acl show /.:/subsys/ssp/sp3en0/hardmon/frame1/slot1
{user dero ----m--}
{group spsec-admin -c-----}
{group hm-control --vsmu-}
{group hm-control-services --vsmu-}
{group hm-monitor ----m--}
{group hm-monitor-services ----m--}
{group none ---sm--}
sp3en0/ # dcecp -c acl show /.:/subsys/ssp/sp3en0/hardmon/hardmon
{user dero4 ------t}
{group sp_admin ac----t}
{group spsec-admin -c-----}
{group hm-admin a------}
```

Part 2: At some point after the corruption takes place, Hardmon client requests fail. Examining the log reveals that Hardmon's ACL manager experienced a problem during the server's last initialization.
Part 3: The corrective action is to Stop Hardmon, delete the ACL database files, and start Hardmon.
The custom ACL, frame1/slot, no longer exists, and the default hardmon ACL no longer contains customer-changed entries.

Synchronizing Hardmon Kerberos V4 and DCE ACLs

There is no PSSP tool that synchronizes entries between Kerberos V4 and DCE ACLs. There are several reasons for this. First, the tool would not know if an entry in a Kerberos V4 ACL should automatically be a member of a DCE ACL. Second, there is no way to guarantee that for every Kerberos V4 principal, there is a corresponding DCE principal. Third, it is possible to create individual frame ACLs for either Kerberos V4 or DCE ACLs, but the existence of frame ACL controls in a dce security mode does not automatically imply their existence in a compat mode. Further, individual slot ACLs exist only in DCE ACL objects.

It is up to the SP administrator to ensure that the entry definitions between DCE and Kerberos V4 ACLs of the same name are synchronized, but given that Hardmon ACLs only reside on the control workstation, managing and maintaining synchronization between the two types of ACLs can be part of a well-defined process or procedural task.

In general, the requirement to change ACLs should be an infrequent one. (Two notable exceptions are prototyping and test environments.) Given that ACLs are control points in the system, it is imperative that they remain as stable as possible. Still, the need to alter ACL contents at some point is a reality. One way to ensure that ACL management happens in a timely and controlled manner is to add triggers to AIX user account, Kerberos V4 principal, and DCE principal creation/deletion procedures. When a user account/principal is created, Hardmon ACL membership consideration should be part of that process. When a user account/principal is deleted, Hardmon ACL membership removal should be part of that process. This will keep ACL entries from becoming “stale” or invalid.
8.2.3 Other events

“Other” events include anomalous events, such as a hung daemon, low paging space, a file system that reaches its storage capacity, or a severe time skew between hosts.

8.2.3.1 All PSSP services appear to be hung

When all PSSP services appear to be hung in a dce-enabled security mode, the DCE Security servers and/or DCE CDS servers may be down or not reachable.

A quick test to help determine if at least one Security server is reachable is to issue a dce_login on the host where all PSSP services appear hung. DCE Security servers that are down or unreachable will cause the dce_login to fail or may take up to eight minutes to actually log in. The longer it takes to receive a failed or successful login response, the more likely that some type of problem exists with respect to the DCE Security servers or to the local DCE configuration.

A quick test to help determine if at least one CDS server is reachable is to issue the dcecp -c cdal1 -world -C low command on the host where all PSSP services appear hung. This command recursively lists all the directories in the CDS master server - provided the server can be reached.

Refer to Section 8.8, “DCE client/server problem determination” on page 390 for more information.

8.2.3.2 One PSSP daemon appears to be hung

In instances when a PSSP server hangs and there is no clear indication from server or system logs what caused the hang, the event should be logged, along with any relevant environmental data, and the logs should be saved. If the problem becomes persistent or begins to appear with greater frequency, a cause/effect profile may be established from the incident reports. Such events are rare, and a simple stop/start of the server usually corrects the anomalous situation.

Prior to “recycling” a server/service uses the AIX commands, lsarc -s or lsarc -ls, to obtain status on the server. Using -l requests that a subsystem send its current status in long form. Long status requires that a status request be sent to the subsystem; it is the responsibility of the subsystem to return the status. Some PSSP services, such as Hardmon, SDR, and Sysctl, do not support -l but do respond to -s.

*Problem management lsarc -ls*
Host responds lssrc -ls

Note that the PSSP command, hr query, will return the same results.
Several PSSP daemons that don’t support `-l` but respond to `lssrc -s`

```
0513-005 The Subsystem, hardmon, only supports signal communication.
sp3en0/ # lssrc -s hardmon
Subsystem   Group   PID    Status
hardmon     hardmon 17142 active
```

```
0513-005 The Subsystem, sdr.sp3en0, only supports signal communication.
sp3en0/ # lssrc -s sdr.sp3en0
Subsystem   Group   PID    Status
sdr.sp3en0  sdr      32648 active
```

```
0513-005 The Subsystem, sysctld, only supports signal communication.
sp3en0/ # lssrc -s sysctld
Subsystem   Group   PID    Status
sysctld     sysctld 25882 active
```

**Status information on all daemons under inetd control**

This can be obtained with one `lssrc -ls` command. Several resources under inetd control are critical to overall network communications.
8.2.3.3 Low paging space

The amount of paging space required depends on the type of activities performed on the system. If paging space runs low, processes may be lost, and if paging space runs out, the system may panic. When a paging-space low condition is detected, follow local corrective action procedures.

For more information on paging space, refer to AIX V4.3 System Management Concepts: Operating System and Devices, SC23-4311, Chapter 8, “Paging Space and Virtual Memory”. For performance implications related to paging space, see the section on performance considerations of paging spaces in the AIX Performance Tuning Guide.
8.2.3.4 Cannot obtain DCE and/or Kerberos V4 credentials

When the /var or /var/dce directories become full, that is, when they run out of free file space to write new files or update existing files, attempts to log in to DCE will fail.

The error messages associated with this type of problem vary between the types of DCE login functions (PSSP-supplied or DCE commands), and whether the file system is “close to full” or full.

The AIX command, df, reveals 100 percent of /var is used, but 160 i-nodes are still free. Having some i-nodes remaining allows the creation of partial context credentials, though they are of no value.
In the previous examples, notice that dce_login_noexec didn't generate any errors. However, since repeated dce_login_noexec commands consume free space in /var, it's just a matter of time before dce_login_noexec generates an error. Notice, too, that as additional dce_login_noexec commands are executed, the number of free i-nodes decreases.

Eventually, all free space under /var that can be allocated will be consumed, and the error messages for all types of DCE login attempts will take the same form as shown here:
/var (or /var/dce) becoming full has implications beyond the inability of users and PSSP services to log in to DCE. One implication is that the CDS client on the host crashes. Also, stopping DCE servers when /var is full prevents DCE from writing any critical data to disk as part of its shutdown process.

To correct the situation, either add more space to the file system, stop the DCE daemons on the host, and restart them. Or, stop the DCE daemons on the host, delete all DCE credentials files, run clean_up.dce, start DCE daemons, and stop/start all PSSP services on the host because their credentials have been destroyed, yet they still have environmental variable references to those files.

Note that using the DCE command, rmxcred, to delete credentials as an alternative to deleting the files is not likely to relieve the space problem. rmxcred only deletes expired credentials.

**Unable to obtain Kerberos V4 credentials**

By default, the Kerberos V4 credentials cache location is located under /tmp. (Users can redefine this location by specifying a different file path in their KRBTKFILE environment variable prior to logging in to Kerberos V4.) When /tmp becomes full (runs out of free file space to write new files or update existing files) attempts to log in to Kerberos V4 will fail.
The error messages relating to this type of problem do not vary between the types of Kerberos V4 login functions, nor do they vary depending on whether the file system is “close to full” or full.

### 8.2.3.5 Time skew too great

Time is critical to normal security operations, not to mention other PSSP operations. Both DCE and Kerberos V4 have a low tolerance for time skews—five minutes to be exact. DCE and Kerberos V4 rely on credentials (tickets) and a certain time frame in which they are valid. If a DCE-enabled or Kerberos V4-enabled host has a time stamp difference of greater than five minutes between itself and its time server(s), credential creation on the host will not be possible, nor will other DCE and Kerberos V4 functions. Likewise, external requests attempting to remotely log in via AIX remote commands will be rejected. RSCT’s Topology Services component may begin to see packets arriving out of chronological order and may cause RSCT to falsely detect that one of its peer nodes has failed. Other partition-related PSSP services will also be impacted.

The common configuration of an SP system provides all the nodes with the same time zone setting. But what about a more complex configuration with nodes running in different time zones; how do you synchronize them? It's not important how many time zones you run on your nodes because the time stamp is the same for all of them. What is important is the time setting. If the time set on the node differs from the time of its time server by more than five minutes, DCE and Kerberos V4 get into trouble decoding the authorization data because the time stamp is part of it.
**DCE/k5 responses**

The following screens show the various commands:

```
sp3n06/ # dce_login sp_admin
Enter Password:
Error - Clock skew too great (dce / krb)
sp3n06/ # dsrvrgt. sp/sysect1
dsrvrgt: space_login_as_service: 2502-606 DCE error in sec_login_valid_from_keytable: Clock skew too great (dce / krb)
sp3n06/ # sysct1 whoami -v
2000-01-22-20:53:54.869-05:00----- PID=5080 ERROR gass authentication gassapi.c 4267 0x00000001 msgID=0xd2862113
Can't get server ticket in gss_init_sec_ctx (14129025)
Clock skew too great
sp3n06/ # dovecp -c acl show /..subsys/sp/sp3n06/sysct1/etc/sysct1.acl
Warning: Not able to acquire initial login context.Error: msgID=0x14129025 Clock skew too great
```

```
sp3n06/ # dovecp -c cell show
Warning: Not able to acquire initial login context.Error: msgID=0x14129025 Clock skew too great
sp3n06/ # dovecp -c cdsl -world -C low
(get_objs): Error enumerating directories under /.. Clock skew too great (dce / krb)
(get links): Error enumerating objects under /.. Clock skew too great (dce / krb)
(get_dirs): Error enumerating directories under /.. Clock skew too great (dce / krb)
Warning: Not able to acquire initial login context.Error: child process exited abnormally
```

In the `/var/dce/svc/error.log`, various types of messages relating to the time skew problem exist.

```
2000-01-22-21:00:18.041-05:00----- cdsclerk(14152) ERROR cds general clerk_bind.c 638 0x00000451 msgID=0x10D0AB70 Routine rpc_binding_set_auth_info(3rpc) failed : status = 336760869.
2000-01-22-21:00:18.085-05:00----- cdsclerk(14152) ERROR cds general clerk_bind.c 638 0x00000451 msgID=0x10D0AB70 Routine rpc_binding_set_auth_info(3rpc) failed : status = 336760869.
2000-01-22-21:03:15.186-05:00----- dced ERROR dhd secval sv_clientd.c 1500 0x00001c1e msgID=0x14129025 Call to a sec_key_mgmt_poo function failed, status=0x14129025
```

Attempting to rsh via k5 to a host with a time skew problem and from a host with a time skew problem.
Starting an already-configured DCE client or attempting to run the local configuration portion of a DCE client when the time skew is too great results in the following:

```
sp3en0/ # rsh sp3n06 date
kerberos: Couldn't authenticate to the server: Server rejected authentication (during sendauth exchange).
kerberos: Server returned error code 37 (Clock skew too great).
kerberos: Error text sent from the server Clock skew too great.
rshd: 0826-813 Permission is denied.

sp3n06/ # rsh sp3en0 date
kerberos: Couldn't get credentials for the server: Clock skew too great.
rshd: 0826-813 Permission is denied.
```

The corrective action for an already configured DCE client is to stop any DCE services that were able to start, run clean_up.dce, correct the time skew, and then start DCE.

The corrective action for a failed local client configuration is to stop DCE services that were configured (and started), unconfigure the local portion of the DCE client, unconfigure the administrative portion of the DCE client portion, correct the time skew, and then reconfigure the administrative and local portions of the client.

The CDS master server will crash when the time on a host is set to a value in the past. This results in a catastrophic CDS failure and is noted in the /var/dce/svc/fatal.log.

```
1999-11-15-17:04:38.714-05:00I----- cdsd(3178) FATAL cds/server uid_lib.c 505 0x0000000e0f
msgID=0xc16D0A384 ts_new(): system time has gone backwards more than one minutes.
Exiting cdsd.
```

Chapter 8. Problem isolation and resolution 351
**Kerberos V4/k4 responses**

Various Commands

```bash
sp3n06/ # /usr/lpp/ssp/rcmd/bin/rcmdtgt
rcmdtgt: 2502-052 Error getting service ticket for
rcmd.sp3n06@ITSO.IBM.COM
2504-037 Kerberos V4 error: client and server clocks must be synchronized.
sp3n06/ # k4init root.admin
Kerberos V4 Initialization for "root.admin"
Password:
k4init: 2504-037 Kerberos V4 error: client and server clocks must be synchronized
sp3n06/ # sysctl whoami -v
2000-01-22-21:21:56.590-05:00I----- PID#19102 ERROR gss authentication gssapi.c 4001
0x0000001 msgID=0x128620FF
Could not retrieve default login context (171220ec)
2502-603 You do not have DCE credentials.
No currently established network identity for this context exists
2502-603 You do not have Kerberos V4 credentials.
```

Attempting to rsh via k4 to a host with a time skew problem and from a host
with a time skew problem:

```bash
sp3en0/ # rsh sp3n09 date
krshd: Kerberos Authentication Failed.
```

```bash
sp3en0/ # rsh sp3en0 date
spk4rsh: 0041-004 Kerberos V4 rcmd failed: 2504-037 Kerberos V4 error: client and server
clocks must be synchronized.
rshd: 0826-813 Permission is denied.
```

**errpt entries**

The AIX errpt log will also contain errors relating to a time skew from other
PSSP services.
8.3 DCE cell reconfiguration impacts

We will now cover the procedure for recovering PSSP DCE security services when a DCE cell is reconfigured.

In an SP environment where PSSP is configured and running in a DCE security mode (either dce or dce:compat) and/or when the AIX remote command authentication methods for the SP include k5 (Kerberos V5), if the DCE cell into which PSSP is configured is unconfigured/reconfigured, all PSSP DCE-related services must also be unconfigured/reconfigured. This is due to DCE configuration dependencies established at the time PSSP was configured for DCE security.

Depending on the PSSP DCE security configuration in use, the level of disruption and interruption to PSSP services ranges from low to high when the DCE cell must be reconfigured. The greater the dependency on DCE, the greater the level of disruption and interruption to PSSP services.

The scenarios below cover four SP security configurations: compat/k5:k4:std, dce:compat/k5:k4:std, dce/k5:std, and dce/k5. The scenarios are based on single partition systems. SP systems running with multiple partitions can apply the individual scenarios to partition level security configurations within the system.

The inclusion of the AIX authentication method, std, is a choice in each scenario. All recovery steps are the same with or without std, except for the dce/k5:std scenario. dce/k5:std without std is the configuration dce/k5, which is handled as a separate scenario. (The steps for dce/k5:std and dce/k5 are slightly different.) Also, for the scope of the recovery procedures, the presence of std is taken to mean that a /.rhosts file was created by PSSP and
exists in root’s home directory. This point is critical for the recovery in a
dce/k5:std environment and marks the difference between dce/k5:std and
dce/k5 recovery.

All of the unconfiguration steps can be executed before the DCE cell is
reconfigured. However, steps relating to PSSP DCE configuration routines
must take place after the DCE cell has been reconfigured.

All actions are performed as a root user. Any dependencies on DCE cell
administrator authority takes places during PSSP security installation and
configuration.

8.3.1 compat/k5:k4:std

The level of disruption and interruption in a system running a
compat/k5:k4:std security configuration is low, if not entirely transparent.
PSSP reliance on DCE is only for remote command support (k5). And given
that both k4 and std are enabled--and that in this scenario compat requires
Kerberos V4 services to be configured for remote commands--PSSP remote
command operations are successful via k4.

Unless stated otherwise, all work must be performed from the control
workstation.

1. Change the control workstation AIX authentication methods so that they
do not include k5.

   ```
   sp3en0/ # splstdata -p
   ...
   auth_install k4
   auth_root_rcmd dce:k4:std
   ts_auth_methods compat
   auth_methods k5:k4:std
   ```

   sp3en0/ # chauthent -k4 -std

2. Change the partition’s SDR security attributes so that they do not include
dce in auth_install (otherwise, PSSP will try to install/configure/start DCE
on a node); do not include dce in auth_root_rcmd (otherwise, a /.k5login
file will be generated on each host); do not include k5 in auth_methods
(otherwise, the k5 AIX remote command authentication method will be
enabled on each host).

   ```
   sp3en0/ # spsetauth -l k4
   sp3en0/ # spsetauth -d k4 std
   sp3en0/ # chauthpar -c k4 std
   ```
3. Change the AIX remote command authentication methods on the nodes.

```
sp3en0/ # dsh -avG "chauthent -k4 -std"
```

4. Stop DCE services on each host and then unconfigure DCE services on each host. (The example reflects these steps being issued remotely through a `dsh` command. An alternative to this approach is to log in to each node where DCE is installed and run the commands locally.)

```
sp3en0/ # dsh -avG "stop.dce ; unconfigure.dce -config_type local -dependents -force all"
```

all unconfigures all configured components on the local machine. -dependents unconfigures dependent components. It specifies that any components that depend on those listed on the command line, in this case, all, should also be unconfigured. -force forces unconfiguration of components named on the command line, in this case, all, even if other components depend on their presence. This option should only be used in clean-up situations or recovery situations as this procedure covers. For more information about the DCE `unconfig.dce` command, refer to the *IBM DCE for AIX Administration Command Reference*.

5. Stop DCE services on the control workstation and then unconfigure DCE services on the control workstation.

```
sp3en0/ # stop.dce ; unconfigure.dce -config_type local -dependents -force all
```

Once the DCE cell is reconfigured and running, configure DCE services on the control workstation, and then refer to the *PSSP Planning Guide* and the *PSSP Installation and Migration Guide*, GA22-7347, to configure the rest of the SP for DCE/k5 use, in order to reestablish a compat/k5:k4:std security environment.

### 8.3.2 dce:compat/k5:k4:std

The level of disruption and interruption in a system running a dce:compat/k5:k4:std security configuration is medium to medium-high. PSSP will try to use DCE for its Trusted Services (SDR, Hardmon, Sysctl, and others) security calls and for k5 remote command support. It will take several minutes to fail-over from dce to compat when DCE Security and/or CDS master servers are not running, or when the DCE cell has been reconfigured.
from under the SP. As such, performance is impacted, and some PSSP functions will appear to “hang” during the failover period.

Given that both k4 and std are enabled and that, in this scenario, compat requires Kerberos V4 services to be configured for remote commands, PSSP remote command operations are eventually successful via k4.

Unless stated otherwise, all work must be performed from the control workstation. Any dependencies on DCE cell administrator authority takes places during PSSP security installation and configuration.

1. Change the control workstation AIX authentication methods so that they don’t include k5.

   ```
   sp3en0/ # splstdata -p
   ...
   auth_install  dce:k4
   auth_root_rcmd  dce:k4:std
   ts_auth_methods  dce:compat
   auth_methods  k5:k4:std
   ```

2. Change the partition’s SDR security attributes so that they do not include dce in auth_install (otherwise, PSSP will try to install/configure/start DCE on a node). Do not include dce in auth_root_rcmd (otherwise, a /.k5login file will be generated on each host). Do not include dce in ts_auth_methods (otherwise, Trusted Services will attempt to use DCE security services); do not include k5 in auth_methods (otherwise, the k5 AIX remote command authentication method will be enabled on each host).

   ```
   sp3en0/ # chauthpts -c compat
   sp3en0/ # spsetauth -d k4 std
   sp3en0/ # chauthpar -c k4 std
   sp3en0/ # spsetauth -i k4 std
   ```

3. Change the AIX remote command authentication methods on the nodes.

   ```
   sp3en0/ # dsh -avG "chauthent -k4 -std"
   ```

   Change the Trusted Services authentication methods on the nodes.

   ```
   sp3en0/ # dsh -avG "chauthts compat"
   ```
4. Stop DCE services on each host and then unconfigure DCE services on each host. (The example reflects these steps being issued remotely through a `dsh` command. An alternative to this approach is to log in to each node where DCE is installed and run the commands locally.)

```
sp3en0/ # dsh -avG "stop.dce ; unconfigure.dce -config_type local -dependents -force all"
```

all unconfigures all configured components on the local machine. -dependents unconfigures dependent components. It specifies that any components that depend on those listed on the command line, in this case, all, should also be unconfigured. -force forces unconfiguration of components named on the command line, in this case, all, even if other components depend on their presence. This option should only be used in clean-up situations or recovery situations as this procedure covers. For more information about the DCE unconfig.dce command, refer to the *IBM DCE for AIX Administration Command Reference*.

5. Stop DCE services on the control workstation and then unconfigure DCE services on the control workstation.

```
sp3en0/ # stop.dce ; unconfigure.dce -config_type local -dependents -force all
```

Once the DCE cell is reconfigured and running, configure DCE services on the control workstation and refer to the *PSSP Planning Guide* and *PSSP Installation and Migration Guide*, GA22-7347, to configure the rest of the SP for DCE/k5 use, in order to reestablish a dce:compat/k5:k4:std security environment.

### 8.3.3 dce/k5:std

The level of disruption and interruption in a system running a dce/k5:std security configuration is high. PSSP relies on DCE for its Trusted Services (SDR, Hardmon, Sysctl, and others) security calls and k5 remote command support. It will take several minutes for PSSP DCE calls to time-out when DCE Security and/or CDS master servers are not running, or when the DCE cell has been reconfigured from under the SP. In this scenario, there is no failover for Trusted Services. As such, PSSP functions will appear to “hang” before ultimately failing.

Given that std is enabled, PSSP remote command operations will be successful via std. However, since PSSP services will fail when its Trusted
Services security calls are made to DCE, the ability to issue remote commands is useful only in recovery situations.

Unless stated otherwise, all work must be performed on/from the control workstation. Any dependencies on DCE cell administrator authority take place during PSSP security installation and configuration.

**Note**

This recovery procedure leaves the SP at a minimum level of PSSP security. SP systems connected to the Internet, or to intranets that allow access from various points in your organization, should carefully evaluate whether or not to allow Internet/intranet access while the SP is running with the minimum level of security. The steps below are meant only to allow SP administrators to place the SP in a security state that permits it to be reconfigured with DCE and PSSP DCE services. These steps should be performed only after the DCE cell has been reconfigured and is available for PSSP DCE configuration.

1. Change the control workstation AIX authentication methods so that they don’t include k5.

```bash
sp3en0/ # splstdata -p
...
auth_install  doe
auth_root_rcmd  doe:std
ts_auth_methods  doe
auth_methods  k5:std
sp3en0/ # chauthent -std
```

2. Change the partition’s SDR security attributes so that they do not include doe in auth_install (otherwise, PSSP will try to install/configure/start DCE on a node); do not include doe in auth_root_rcmd (otherwise, a /.k5login file will be generated on each host); do not include doe in ts_auth_methods (otherwise, Trusted Services will attempt to use DCE security services); do not include k5 in auth_methods (otherwise, the k5 AIX remote command authentication method will be enabled on each host).

```bash
sp3en0/ # spsetauth -i
sp3en0/ # spsetauth -d std
sp3en0/ # chauthpts -c
sp3en0/ # chauthpar -c std
```

3. Change the AIX remote command authentication methods on the nodes.
Change the Trusted Services authentication methods on the nodes.

4. Stop DCE services on each host and then unconfigure DCE services on each host. (The example reflects these steps being issued remotely through a dsh command. An alternative to this approach is to log in to each node where DCE is installed and run the commands locally.)

all unconfigures all configured components on the local machine. -dependents unconfigures dependent components. It specifies that any components that depend on those listed on the command line, in this case all, should also be unconfigured. -force forces unconfiguration of components named on the command line, in this case, all, even if other components depend on their presence. This option should only be used in clean-up situations, or recovery situations as this procedure covers. For more information on the DCE unconfig.dce command, refer to the IBM DCE for AIX Administration Command Reference.

5. Delete PSSP DCE key files and PSSP Sysctl DCE ACL databases. Normally, this would be done through the PSSP rm_spsec command, but this command requires that DCE services be functioning properly. The purpose of this section is to deal with the conditions where the DCE cell has either been unconfigured or is in the process of being unconfigured.

6. Note that the PSSP DCE configuration file, /spdata/sys1/spsec/spsec_overrides, is deleted as part of this process. This file will be re-created when the host is reconfigured for PSSP DCE services.

7. Stop DCE services on the control workstation and then unconfigure DCE services on the control workstation.
8. Delete PSSP DCE key files and PSSP Sysctl and Hardmon DCE ACL databases. Normally, this would be done through the PSSP `rm_spsec` command, but this command requires that DCE services be functioning properly. Do not delete the `/spdata/sys1/spsec/spsec_overrides` file on the control workstation.

```
sp3en0/ # stop.dce ; unconfigure.dce -config_type local -dependents -force all
```

Once the DCE cell is reconfigured and running, configure DCE services on the control workstation and then refer to the *PSSP Planning Guide* and *PSSP Installation and Migration Guide*, GA22-7347, to configure the rest of the SP for DCE/k5 use, in order to reestablish a dce/k5:std security environment.

### 8.3.4 dce/k5

This procedure is similar to the dce/k5:std scenario, with one notable exception: std is not enabled in this scenario. This causes an extra step to be taken in order to access nodes remotely. Plus, it requires node cleanup be to executed serially (one after another), rather than in a distributed parallel manner (one command executed on all hosts from the control workstation). This will significantly increase the amount of time required to perform recovery.

The level of disruption and interruption in a system running a dce/k5:std security configuration is high. PSSP relies on DCE for its Trusted Services (SDR, Hardmon, Sysctl, and so on), security calls, and k5 remote command support. It will take several minutes for PSSP DCE calls to time out when DCE Security and/or CDS master servers are not running or when the DCE cell has been reconfigured from under the SP. In this scenario, there is no failover for Trusted Services. As such, PSSP functions will appear to “hang” before ultimately failing.

Given that std is not enabled, the ability to issue the remote command, `rsh`, or the PSSP remote parallel command, `dsh`, will be refused. This means that all recovery commands that must be run on a node require the SP administrator to use the PSSP command, `s1term`, to get into a node. (A telnet to the node will not be successful since AIX on the node will only permit remote command processing through the k5 protocol, which is not available due to
the state of the DCE cell. If std had also been enabled on the nodes, a telnet or rlogin would have been possible, even if the nodes lacked a /.rhosts file.)

**Note**

This recovery procedure leaves the SP at a minimum level of PSSP security. SP systems connected to the Internet (or to intranets that allow access from various points in your organization) should carefully evaluate whether or not to allow Internet/intranet access while the SP is running with the minimum level of security. The following steps are only meant to allow SP administrators to place the SP in a security state that permits it to be reconfigured with DCE and PSSP DCE services. These steps should be performed only after the DCE cell has been reconfigured and is available for PSSP DCE configuration.

1. Change the control workstation AIX authentication methods so that they do not include k5, but *do* include std.

   ```
   sp3en0/ # splstdata -p
   ...
   auth_install dce
   auth_root_rcmd dce
   ts_auth_methods dce
   auth_methods k5
   sp3en0/ # chauthent -std
   ```

2. Change the partition's SDR security attributes so that they do not include dce in auth_install (otherwise, PSSP will try to install/configure/start DCE on a node). Do not include dce in auth_root_rcmd (otherwise, a /.k5login file will be generated on each host). Do not include dce in ts_auth_methods (otherwise, Trusted Services will attempt to use DCE security services). Do not include k5 in auth_methods (otherwise, the k5 AIX remote command authentication method will be enabled on each host).

   ```
   sp3en0/ # spsetauth -i
   sp3en0/ # spsetauth -d std
   sp3en0/ # chauthpts -c
   sp3en0/ # chauthpar -c std
   ```

3. For each node, s1term to the node and do the following: Change the AIX remote command authentication methods. Change the Trusted Services authentication methods. Create a /.rhosts file with control workstation only entries (this will permit rsh, rcp, dsh, telnet, and so on from the control workstation to the node). Stop DCE services, and then unconfigure DCE.
Delete PSSP DCE key files and PSSP Sysctl DCE ACL databases (normally, this would be done through the PSSP `rm_spsec` command, but this command requires that DCE services be functioning properly).

The `s1term` in the example below uses the first node in the first frame. Repeat the process for all nodes in the system by substituting the correct `s1term` frame/slot data for each node. A `.rhosts` file is created for root using the entries, sp3en0.itso.ibm.com root and sp3tr0.itso.ibm.com root. (These are the control workstation addresses on the SP system used during the creation of this book.)

```
sp3en0/ # s1term -w 1 1
...
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login: ...
sp3n01/ # chauthts ; chauthent -std
sp3n01/ # vi /.rhosts
...
sp3en0.itso.ibm.com root
sp3tr0.itso.ibm.com root
...
sp3n01/ # stop.dce ; unconfigure.dce -config_type local -dependents -force all
sp3n01/ # rm -R /spdata/sys1/keyfiles/* ; rm /var/sysctl/db_* ; rm /spdata/sys1/spsec/spsec_overrides
```

`all` unconfigures all configured components on the local machine. `-dependents` unconfigures dependent components. It specifies that any components that depend on those listed on the command line, in this case all, should also be unconfigured. `-force` forces unconfiguration of components named on the command line, in this case, all, even if other components depend on their presence. This option should only be used in cleanup situations or recovery situations as this procedure covers. For more information about the DCE `unconfig.dce` command, refer to the IBM DCE for AIX Administration Command Reference.

The PSSP DCE configuration file, `/spdata/sys1/spsec/spsec_overrides`, is deleted as part of this process. This file will be re-created when the host is reconfigured for PSSP DCE services.

4. Stop DCE services on the control workstation and then unconfigure DCE services on the control workstation.

```
sp3en0/ # stop.dce ; unconfigure.dce -config_type local -dependents -force all
```
5. Delete PSSP DCE key files and PSSP Sysctl and Hardmon DCE ACL databases. Normally, this would be done through the PSSP rm_spsec command, but this command requires that DCE services be functioning properly. Do not delete the /spdata/sys1/spsec/spsecoverrides file on the control workstation.

Once the DCE cell is reconfigured and running, configure DCE services on the control workstation, and then refer to the PSSP Planning Guide and PSSP Installation and Migration Guide, GA22-7347, to configure the rest of the SP for DCE/k5 use, in order to reestablish a dce/k5 security environment.

8.4 Temporarily disabling dce/k5 when DCE is not available

This section describes a general procedure to disable the dce and/or k5 methods from PSSP use when DCE Security and CDS servers cannot be reached.

For availability reasons, a DCE cell should contain at least one Security server replica (in addition to the master) and at least one CDS server replica (in addition to the master). (Refer to the DCE Administration Guide -- Core Components for information about creating and maintaining DCE server replicas.)

But even in highly-available environments, the possibility that a DCE host cannot contact needed DCE servers is a possibility. In an SP environment where PSSP is configured and running in a DCE security mode (either dce or dce:compat), and/or when the AIX remote command authentication methods for the SP include k5 (Kerberos V5), if the DCE cell into which PSSP is configured is experiencing networking problems, or the DCE master and replica servers are down or not available, all PSSP DCE-related services will be impacted by the outage.

Depending on the security configuration in use, the level of disruption and interruption to PSSP services ranges from low to high. The greater the dependency on DCE, the greater the level of disruption and interruption to PSSP services. PSSP dce and k5 authentication methods can be temporarily disabled until DCE cell services are restored to normal availability.

The scenarios below cover two SP security configurations: compat/k5:k4:std and dce:compat/k5:k4:std. Temporarily disabling dce and k5 in these...
configurations does not impact PSSP services, given that compat and k4 are enabled.

The security configurations, dce/k5:std and dce/k5, are not included because the temporary removal of dce and k5 from their configurations leaves the SP running with a minimum set of PSSP security controls. Systems configured with the highest levels of PSSP-supplied security are done so with the intent to run in a highly-secure environment. Anything less places the entire SP system at risk. In dce/k5:std and dce/k5 configurations, the normal operation of DCE cell services and the networking paths among and between DCE hosts are critical in ensuring proper SP operation. Dropping SP security to a minimum set of security controls when DCE cell services are impaired is not the safest approach. Instead, normal DCE cell operation should be reestablished as soon as possible.

In the scenarios covered, the inclusion of the AIX authentication method, std, was a configuration choice. All steps are the same with or without std present. Also, for the scope of the procedures, the presence of std is taken to mean that a /.rhosts file was created by PSSP and exists in root's home directory on each node.

Each scenario also includes steps on reinstating dce and k5 when DCE cell operations return to normal.

All actions are performed as a root user.

Node and control workstation DCE configurations, PSSP DCE ACL key files/ACL databases, and root's /.k5login file are not altered in any way by the following steps. The same holds true for the master DCE databases.

8.4.1 compat/k5:k4:std

The level of disruption and interruption in a system running a compat/k5:k4:std security configuration is low, if not entirely transparent. PSSP's reliance on DCE is only for k5 remote command support, and given that both k4 and std are enabled and that, in this scenario, compat requires Kerberos V4 services to be configured for remote commands, PSSP remote command operations are successful via k4.

Unless stated otherwise, all work must be performed from the control workstation.

1. Change the control workstation AIX authentication methods so that they do not include k5.
2. Change the partition’s SDR security attributes so that they do not include dce in auth_install (otherwise, PSSP will try to install/configure/start DCE on a node); do not include dce in auth_root_rcmd (otherwise, a /.k5login file will be generated on each host); do not include k5 in auth_methods (otherwise, the k5 AIX remote command authentication method will be enabled on each host).

```
sp3en0/ # splstda -p
...
auth_install k4
auth_root_rcmd dce:k4:std
ts_auth_methods compat
auth_methods k5:k4:std
```

sp3en0/ # chauthent -k4 -std

3. Change the AIX remote command authentication methods on the nodes.

```
sp3en0/ # spsetauth -i k4
sp3en0/ # spsetauth -d k4 std
sp3en0/ # chauthpar -c k4 std
```

Once the DCE cell is operating normally, enable DCE/k5.

1. Change the partition’s SDR security attributes so that they do include dce in auth_install and dce in auth_root_rcmd.

```
sp3en0/ # splstda -p
...
auth_install k4
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods k4:std
```

sp3en0/ # spsetauth -i dce k4
sp3en0/ # spsetauth -d dce k4 std

2. Change the AIX remote command authentication methods on the nodes to include k5.

```
sp3en0/ # dsh -avG "chauthent -k5 -k4 -std"
```
3. Change the control workstation AIX authentication methods and the SDR auth_methods to include k5.

```
sp3en0/ # chauthpar -c k5 k4 std
```

### 8.4.2 dce:compat/k5:k4:std

The level of disruption and interruption in a system running a dce:compat/k5:k4:std security configuration is medium to medium-high. PSSP will try to use DCE for its Trusted Services (SDR, Hardmon, Sysctl, and so on), security calls, and for k5 remote command support. It will take several minutes to fail-over from dce to compat when DCE Security and/or CDS master or replica servers are not running, are not reachable, or when the DCE cell has been reconfigured from under the SP. As such, performance is impacted, and some PSSP functions will appear to “hang” during the failover period.

Given that both k4 and std are enabled, and that, in this scenario, compat requires Kerberos V4 services to be configured for remote commands, PSSP remote command operations are successful via k4.

Unless stated otherwise, all work must be performed from the control workstation.

1. Change the control workstation AIX authentication methods so that they do not include k5.

```
sp3en0/ # splstdata -p
...
auth_install  dce:k4
auth_root_rcmd dce:k4:std
ts_auth_methods dce:compat
auth_methods  k5:k4:std
```

```
sp3en0/ # chauthent -k4 -std
```

2. Change the partition’s SDR security attributes so that they do not include dce in auth_install (otherwise, PSSP will try to install/configure/start DCE on a node). Do not include dce in auth_root_rcmd (otherwise, a /.k5login file will be generated on each host). Do not include k5 in auth_methods (otherwise, the k5 AIX remote command authentication method will be enabled on each host).
3. Change the AIX remote command authentication methods on the nodes.

```
sp3en0/ # spsetauth -i k4
sp3en0/ # spsetauth -d k4 std
sp3en0/ # chauthpar -c k4 std
```

4. Disable dce from the Trusted Services authentication methods in the SDR, on the control workstation, and on the nodes.

```
sp3en0/ # chauthpts compat
```

Once the DCE cell is operating normally, enable DCE/k5.

1. Change the partition’s SDR security attributes so that they include dce in auth_install and dce in auth_root_rcmd.

```
sp3en0/ # splstdata -p
...
auth_install k4
auth_root_rcmd k4:std
ts_auth_methods compat
auth_methods k4:std
sp3en0/ # spsetauth -i dce k4
sp3en0/ # spsetauth -d dce k4 std
```

2. Change the AIX remote command authentication methods on the nodes.

```
sp3en0/ # dsh -avG "chauthent -k5 -k4 -std"
```

3. Change the control workstation AIX authentication methods and the SDR auth_methods so that they include k5. Change the Trusted Services authentication methods in the SDR, on the control workstation, and on the nodes to include dce.

```
sp3en0/ # chauthpar -c k5 k4 std
sp3en0/ # chauthpts dce compat
```
8.5 DCE attributes and Per Node Key Management (PNKM)

PSSP Trusted Services enabled for DCE security are configured into a PSSP-defined DCE organization called spsec-services. PSSP configuration routines do not define explicit spsec-services organization policy data in DCE. As such, the organization is created with the default Security Registry policy data. In most cases, the default Registry policy consists of the least restrictive attributes permitted by DCE.

In particular, DCE’s out-of-the-box password expiration date (pwdexpdate) attribute for the Registry is none, while its password lifetime attribute (pwdlife) is unlimited. (These attributes are considered part of the Registry’s organization policy.) Therefore, when spsec-services is created in a DCE cell running with the DCE default values, spsec-services automatically inherits these attributes for its organization. In this case, the pwdexpdate and pwdlife values for spsec-services will be none and unlimited, respectively.

When PSSP DCE security is configured using the DCE out-of-box policy attributes, PSSP’s DCE keyfile data will never expire, given the pwdexpdate value of none. To this end the Per Node Key Management daemon (on each PSSP DCE host) will wake-up, query the organizational pwdexpdate values of the keys it manages, determine that they are not ready to expire, and return to sleep.

When the organizational data of spsec-services is modified, such that keyfile data will expire at some point, the Per Node Key Management daemon will determine if keys must be updated, based on the pwdexpdate data retrieved, and update the key data if necessary.

PNKM will update key data in both the local keyfile and in the Registry, if a key is slated to expire within the next 24 hours. Only when the Registry update is successful will the local keyfile data be updated; otherwise, the local keyfile data is not updated. This ensures that the Registry data is not out-of-sync with the keyfile data. (The Registry stores only a single password, or key, unlike the keyfile, which can hold many versions of a key.)

Note

Per Node Key Management does not prune, or delete, keys from a keyfile. It only adds keys to a file.

368  Exploiting RS/6000 SP Security: Keeping It Safe
8.5.1 DCE attributes that impact PNKM

There is a limited set of DCE Registry, DCE organizational, and DCE account attributes that impact whether or not a key can be managed.

In hierarchical order, they are:

- **Registry level:**
  - acctlife (DTS-relative time format | unlimited)
  - pwdexpdate (ISO-timestamp | none)
  - pwdlife (DTS-relative time format | unlimited)

- **Organizational level:**
  - acctlife (DTS-relative time format | unlimited)
  - pwdexpdate (ISO-timestamp | none)
  - pwdlife (DTS-relative time format | unlimited)

- **Account level:**
  - acctvalid (yes | no)
  - expdate (ISO-timestamp | none)

Also, there are two attributes for account data that are considered "Policy Attributes"; see the *DCE Commands Reference* for the account command. However, these two attributes, maxtktlife and maxtktrenew, do not impact Per Node Key Management's ability to manage a key. They should not be confused with organizational policy attributes. The account policy attributes only exist at an account level.

---

**Note**

Per Node Key Management only queries the organizational expiration data for the spsec-services organization. The attributes listed above are not queried or managed by Per Node Key Management. They are, however, DCE attributes that can impact Per Node Key Management's ability to manage keys.

At the organizational level the “policy” attributes are not displayed when the defaults are assumed. In this case, the attributes are replaced with the word "nopolicy". nopolicy means that the default DCE values are used. The policy attributes will only be listed when one of the attributes is specifically modified.

Once the modification takes place, all the attributes are revealed as shown in the following screen:
8.5.2 Other factors that impact PNKM

Other factors that impact PNKM’s ability to manage keys are:

- The keyfile does not exist.
- The keyfile is zero (0) length.
- The keyfile contains malformed data.
- The Registry designated for writes (updates) is in read-only mode.
- The Registry that permits updates has been stopped (refer to the DCE
dcecp registry command for more information).
- The Registry that permits updates is down or not reachable.
- secval has been deactivated on the DCE client host.
- DCE client services are not running.
- DCE client services have been unconfigured.
- DCE client services have been reconfigured improperly.
- The SP Security Services library is not available (there’s a symbolic link
from /usr/lib/libspsec.a to the actual library location; either may be
missing).
- The PSSP supplied spsec_defaults and/or spsec_overrides is missing or
corrupt.
- The keyfile does not contain an entry that matches the account’s Registry
entry.
• The keyfile exists, but all entries have been deleted.
• The keyfile contains the maximum number of key versions for a member. No additional key versions can be added until others are deleted.

8.5.3 Displaying DCE attributes
The following examples show how to display Registry, organization, and account attributes.

8.5.3.1 Registry
1. To display the master Security server, which should be the writeable Registry:

```
sp3en0/ # dcecp -c registry catalog -master
/{.../sp_cell/subsys/dce/sec/sp3en0
```

2. To display an attribute list of the registry-wide attributes:

```
sp3en0/ # dcecp -c registry show /{.../sp_cell/subsys/dce/sec/sp3en0 -attributes
{deftktlife 10:00:00.000I-----}
{hidepwd yes}
{maxuid 2147483647}
{mingid 100}
{minorgid 100}
{mintktlife 00:05:00.000I-----}
{minuid 100}
{version secd.dce.1.2.2}
```

3. To display only the registry-wide polices:

```
sp3en0/ # dcecp -c registry show /{.../sp_cell/subsys/dce/sec/sp3en0 -policies
{acctlife unlimited}
{maxtktlife 00:00:00.000I-----}
{maxtktrenew 00:00:00.000I-----}
{pwdalpha yes}
{pwdexpdate none}
{pwdlife unlimited}
{pwdminlen 0}
{pwdspaces yes}
```

8.5.3.2 Organization
1. To display all the organizations in the current cell:
2. To display all attributes and policy values for an organization:

```
# dcecp -c org cat
sp3en0/ # dcecp -c org cat
/.../sp_cell/none
/.../sp_cell/spsec-services
/.../sp_cell/fvt-org
/.../sp_cell/nml-org
/.../sp_cell/dero
```

This example shows the policy attributes of the organization because the organization was either configured with explicit attribute values or the attribute values were changed after the organization was created.

When no organizational attribute values are assigned at organization creation and no changes have been made to these values after the organization is created, the attributes are replaced with the word `nopolicy`. `nopolicy` means that the default DCE values are used as shown in the following screen:

```
# dcecp -c org modify none -pwdexpdate none
# dcecp -c org show none -all
{fullname {}}
{orgid 12}
{uuid 0000000c-19fe-21d3-a402-0004ac493bac}
nopolicy
#
# # An attribute is changed. All attributes are revealed.
# dcecp -c org modify none -pwdexpdate none
# dcecp -c org show none -all
{fullname {}}
{orgid 12}
{uuid 0000000c-19fe-21d3-a402-0004ac493bac}
{acctlife unlimited}
{pwdalpha yes}
{pwdexpdate none}
{pwdlife unlimited}
{pwdminlen 0}
{pwdspaces yes}
```

372 Exploiting RS/6000 SP Security: Keeping It Safe
8.5.3.3 Account

1. To display account attributes followed by account policies:

```bash
# dcecp -c account show ssp/sp3en0/css -all
{acctvalid yes}
{client yes}
{created /.../sp_cell/cell_admin 1999-09-16-15:17:07.000-04:00I-----}
{description {}}
{dupkey no}
{expire none}
{forwardabletkt yes}
{goodsince 1999-09-16-15:17:07.000-04:00I-----}
{group spec-sec-services}
{home /}
{lastchange /.../sp_cell/cell_admin 1999-09-18-10:14:41.000-04:00I-----}
{organization spec-sec-services}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
{usertouser no}
{nopolicy}
```

The nopolicy value relates only to the policy attributes that exist at the account level, namely, maxktl life and maxktlrenew. If these attributes are set for an account, the nopolicy value is replaced with the associated maxktl life and/or maxktlrenew values.

8.6 Using PNKM to diagnose problems

The PNKM daemon, spnkeymand, can be executed from the command line by a root user to determine what keys the Per Node Key Management (PNKM) daemon is managing and their expiration values (where such data can be determined) and to report any detectable errors with key file states. Of particular interest to the SP administrator is using spnkeymand to assist in the diagnosis of problems on a node (or control workstation) running in a PSSP dce security mode (dce or dce:compat).

Should PSSP daemons/services fail or generate DCE error messages during processing in a dce security mode, PNKM’s command line query option provides a way to determine the “health” of PSSP DCE keyfiles for which it manages. (Note that PNKM does not manage HATS DCE keyfiles.)

PNKM’s query function uses each keyfile that it manages and attempts to log in to DCE with each keyfile. (It does not attempt to update the key in the
The message(s) reported by the results of the login attempt can provide valuable clues to the state of the keyfile itself or the ability to contact a DCE Registry in the cell.

--- Important ---

Per Node Key Management should not be used as the only diagnostic aid during problem determination. Since the scope of the diagnosis is a DCE-only query, this approach does not indicate run time problems with PSSP daemons, or run time problems with DCE.

Further, Per Node Key Management only queries the organizational expiration data for the spsec-services organization. Registry, organization, and account attributes discussed in the previous section are not queried or managed by Per Node Key Management. They are, however, DCE attributes that can impact Per Node Key Management's ability to manage keys.

For example, the account attribute, expdate, for the PSSP DCE service, spbgroot, can be set by a DCE cell administrator to expire in the near future, while its spsec-services organizational pwdexpdate attribute is set such that the account's password will never expire. The result of spnkeymand -l will return an expiration value of 0 (zero) for the account, meaning the password never expires, but the account will be locked out at the date specified by the account's expdate value, thereby, preventing Per Node Key Management from being able to log in as the service and manage the service's keys.

```
sp3en0/ # dcecp -c account show ssp/c166cw.ppd.pok.ibm.com/spbgroot | grep exp
(expdate 1999-12-29-07:00:00.000-05:00I-----)
sp3en0/ # spnkeymand -l | grep spbgroot
service=ssp/c166cw.ppd.pok.ibm.com/spbgroot expiration=0
```

So, it is important to keep in mind that using Per Node Key Management as a diagnostic aid is just that, an aid, and not the only course of action to be taken during problem determination.

Additionally, querying is done on a per-host basis, wherever spnkeymand is installed. The daemon does not have to be active or running for the query to take place. That is, lssrc -s spnkeyman does not have to return a state of operative. In fact, spnkeyman does not even have to be registered with AIX's SRC to use its query capability. However, the spnkeymand binary must be installed on the target host.

spnkeymand location after installation:
Finally, the query can take place with or without the dce PSSP authentication method enabled.

### 8.6.1 Examples

Below are examples of data reported by spnkeymand queries.

Note that the expiration data retrieved from the Registry is displayed in raw integer format. An expiration equal to 0 (zero) indicates that key data will not expire. A value other than 0 indicates a specific date/time the key data in the DCE Registry will expire. The easiest way to determine a “human readable” format of the raw time is to query the spsec-services organizational attributes (with the -all flag), as shown previously.

However, should the results of the organizational query return 0 while a service is still experiencing problems, check the account attributes for the service or services, as well as PSSP and DCE run time logs.

The format of the query command is: `spnkeymand -l`.

1. On a node where DCE filesets do not exist, or DCE filesets exist, but DCE is not configured:

   ```
   spn05/ # whence spnkeymand
   /usr/lpp/ssp/bin/spnkeymand
   spn05/ # ls -l /usr/lpp/ssp/bin/spnkeymand
   -r-xr-x--- 1 root system 50966 Oct 20 12:29 /usr/lpp/ssp/bin/spnkeymand
   
   spn05/ # spnkeymand -l
   spsec_start error for starting DCE session. err=0
   No token returned spsec, DCE not installed or configured.
   ```

2. On a node where DCE filesets exist, DCE is configured, but PSSP has not been configured for use with DCE:

   ```
   spn07/ # spnkeymand -l
   spsec_keyman_getExpiration error for ppe/spn07/pmdv3. err=44
   2502-618 The required keyfile was not found: /spdata/sys1/keyfiles/ppe/spn07/pmdv3
   ...
   spsec_keyman_getExpiration error for ppe/spn07/dpc1. err=44
   2502-618 The required keyfile was not found: /spdata/sys1/keyfiles/ppe/spn07/dpc1
   ...
   spsec_keyman_getExpiration error for ssp/spn07/spbgroot. err=44
   2502-618 The required keyfile was not found: /spdata/sys1/keyfiles/ssp/spn07/spbgroot
   ```
3. On a node where DCE filesets exist, DCE is configured, and PSSP has been configured for use with DCE:

```
sp3no5/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
service=ssp/spbgroot expiration=0
```

4. On a control workstation where DCE filesets exist and PSSP has been configured for use with DCE:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
service=ssp/spbgroot expiration=0
```

**Note**
For the remainder of the examples, DCE filesets exist, DCE is configured on the host, and the host has been configured for use with DCE.

5. When `spsec-services` has `pwdexpdate` values of none:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
service=ssp/sp_configd expiration=0
service=ssp/spbgroot expiration=0
```

6. When `spsec-services` has an organization `pwdexpdate` of 1999-09-28-19:59:59:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=938563199
service=ppe/dpcl expiration=938563199
... 
service=ssp/spbgroot expiration=938563199
```

7. When `spsec-services` has a `pwdexpdate` of none, but one of the PSSP services account's has expired, consider the following:

The service's keyfile expired because its organization was changed from `spsec-services` to another organization. In this other organization the
pwdexpdate value caused the keyfile to expire before Per Node Key Management could refresh the key data.

It is not recommended that the members of the PSSP spsec-services organization be moved to another DCE organization within the cell. Not only can this impact PNKM’s ability to manage the key, but it can also impact the way PSSP services operate.

8. When a keyfile does not exist, but PNKM contains it in its list of managed keys:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
... 
service=mmfs/mmfsd expiration=0
Function spsec_keyman_get_expiration failed for service rsct/rsct. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Password has expired (dce / krb)
service=ssp/switchtbld expiration=0
service=ssp/css expiration=0
service=ssp/pmand expiration=0
service=ssp/ap_configd expiration=0
service=ssp/spbgroot expiration=0
```

9. When a keyfile exists, but is 0 (zero) length:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
... 
service=ssp/switchtbld expiration=0
Function spsec_keyman_get_expiration failed for service ssp/css. err=44
2502-618 The required keyfile was not found: /spdata/sys1/keyfiles/ssp/sp3en0/css
service=ssp/pmand expiration=0
... 
... 
service=ssp/spbgroot expiration=0
```

10. When a keyfile exists but contains malformed data:

```
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
... 
... 
service=ssp/switchtbld expiration=0
Function spsec_keyman_get_expiration failed for service ssp/css. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Key table file not found (dce / krb)
service=ssp/pmand expiration=0
... 
... 
service=ssp/spbgroot expiration=0
```
11. When a keyfile does not contain an entry that matches the corresponding account entry:

```
# spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
...
```

Function spsec_keyman_get_expiration failed for service ssp/css. err=1
2502-606 DCE error in sec_login_valid_from_keytable: unsupported key table format version number (dce / krb)
```
```

```
```

12. When a PSSP account does not permit logins, due to its account acctvalid attribute being set to "no", consider the following:

The acctvalid is an account attribute, not an organizational one. However, it does impact PNKM’s ability to manage the key:

```
# spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
...
```

Function spsec_keyman_get_expiration failed for service ssp/css. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Requested key is unavailable (dce / sec)
```
```

```
```

```
```

13. When a PSSP service account’s expdate has expired (that is, the expdate has passed), consider the following: The acctvalid is an account attribute, not an organizational one. However, it does impact PNKM’s ability to manage the key.
14. When a PSSP service account’s pwdvalid is set to “no”:

```bash
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
...
service=sp/switchbld expiration=0
Function spsec_keyman_get_expiration failed for service sp/css. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Client's entry in database has expired
(service / kdb)
service=sp/pnmand expiration=0
...
sp3en0/ # spnkeymand -l
service=ppe/pmdv3 expiration=0
service=ppe/dpcl expiration=0
...
...
...
service=sp/switchbld expiration=0
service=sp/css expiration=0
service=sp/pnmand expiration=0
service=sp/spmgr expiration=0
service=sp/sp_configd expiration=0
service=sp/sp_bgroot expiration=0
```

The pwdvalid is an account attribute, not an organizational one. It does not prohibit PNKM from logging in to the account.

However, the value of this attribute will cause an informational message to be displayed during dce_login time:

```bash
sp3en0/ # dce_login sp/sp3en0/css -k /spdata/sys1/keyfiles/sp/sp3en0/css
Password must be changed!
DCE LOGIN SUCCESSFUL
```

15. When a keyfile contains valid and malformed data, provided that the malformed data appears after the valid data, consider the following.

That is, provided the keyfile was created normally, the header format is valid, and the keyfile can be processed for reading and login purposes only.

When PNKM attempts to log in to DCE using this keyfile, the login will be successful. However, if the keyfile needs to be updated, the update will fail, due to the corruption of the keyfile format.
16. When a DCE Registry cannot be contacted due to problems with the network interfaces, note the following: In this example, DCE client services (and PSSP services) had been running normally on a node. At some point during normal operations, the ethernet adapter went down.
Chapter 8. Problem isolation and resolution

An attempt to start DCE under these conditions will fail. The adapter must be brought back on line to restore proper function.

17. When a DCE Registry cannot be contacted, due to problems with the pe_site file, consider the following: The DCE pe_site file plays an important role in the DCE client's ability to locate Security servers within the DCE cell. If the file is deleted or becomes corrupt after the DCE client
was initialized successfully, then spnkeymand -l will not return any errors because DCE has already established and cached the Security server location information. (As shown in the first example.)

However, if the pe_site file is missing or becomes corrupt during DCE client initialization, the DCE client will not start. When spnkeymand -l is issued under these conditions, it returns an error message stating that the KDC cannot be found. This is a prime indication that there is a problem relating to the pe_site file and that this problem has impaired DCE client services on the host. Since PSSP security services under a PSSP dce security mode relies on DCE client services, the pe_site problem impairs PSSP security services.

Either restore a valid pe_site file from a backup, or follow the corrective actions in the DCE Problem Determination Guide. If the file is to be restored from a back-up as root on the host stop DCE (via stop.dce), clean-up all DCE cache files (via clean_up.dce), and then start DCE (via start.dce).

Example 1: The pe_site does not exist, or is malformed, after the DCE clients have initialized successfully:

```
sp3n05/ # ls -l /etc/dce/security/pe_site
ls: 0653-341 The file /etc/dce/security/pe_site does not exist.
sp3n05/ /usr/lpp/ssp/bin/spnkeymand -l
service=ppe/pmdv3 expiration=0
... service=ssp/spbgroot expiration=0
sp3n05/ # ls -l /etc/dce/security/pe_site
-rw-r--r-- 1 root system 0 Dec 19 13:46 /etc/dce/security/pe_site
sp3n05/ /usr/lpp/ssp/bin/spnkeymand -l
service=ppe/pmdv3 expiration=0
... service=ssp/spbgroot expiration=0
```

Example 2: The pe_site does not exist, or is malformed, during DCE client initialization:
18. When the DCE cell does not contain a running Security Server, consider the following: The results of this query are identical to the situation where DCE client services were started without a valid `pe_site` file. However, there is one major—and significant—difference between the two results. In this case the `spnkeymand -l` may take over 15 minutes to complete, because each PNKM DCE login attempt has to time-out. The login time-out value is an internal DCE value and cannot be tailored by PNKM. If `spnkeymand -l` is taking several minutes to process (and fail) a login attempt, CTRL+C and investigate the matter further.

```
sp3n05/ # ls -l /etc/dce/security/pe_site
ls: 0653-341 The file /etc/dce/security/pe_site does not exist.

sp3n05/ # start.dce
Gathering current configuration information...
Start of DCE Host, sp3n05, will now begin.
Starting RPC...
RPC was started successfully.
0x11315a0c: Unable to obtain a binding for the Security client.
0x11315a03: The components on DCE host, sp3n05 did not start successfully.
0x113159fb: Start did not complete successfully.
Gathering component state information...

Component Summary for Host: sp3n05

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
</tbody>
</table>

The component summary is complete.
More information can be found in the configuration log:
/opt/dcelocal/etc/cfgdce.log.

sp3n05/ # /usr/lpp/ssp/bin/spnkeymand -l
spnkeyman_get_expiration error for ppe/pmdv3. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
spnkeyman_get_expiration error for ppe/dpcl. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
...
spnkeyman_get_expiration error for ssp/spbgroot. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
```

If all DCE Security Servers are down in the current cell, or not reachable due to network problems, services other than PNKM will also experience problems.

---

Note

If all DCE Security Servers are down in the current cell, or not reachable due to network problems, services other than PNKM will also experience problems.
19. When the DCE clients on a host are down, note the following: This is another case where `spnkeymand -l` will not reveal a problem, since a DCE login does not require DCE client services to be running on a host. DCE only requires that a DCE Security Server be reachable in read mode to perform a login.

However, it is worth showing the results of `spnkeymand -l` under conditions where DCE client services are not running, if only to present a case for the need to use techniques other than PNKM in order to diagnose problems.

```
sp3n05/ # /usr/lpp/ssp/bin/spnkeymand -l
spsec_keyman_get_expiration error for ppe/pmdv3. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
spsec_keyman_get_expiration error for ppe/dpcl. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
...
spsec_keyman_get_expiration error for ssp/spbgroot. err=1
2502-606 DCE error in sec_login_valid_from_keytable: Cannot find KDC for requested realm (dce / krb)
```

```
sp3n05/ # /usr/lpp/ssp/bin/spnkeymand -l
service=ppe/pmdv3 expiration=0
...
...
...
service=ssp/spbgroot expiration=0
sp3n05/ # show.cfg
Gathering component state information...

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
</tbody>
</table>

The component summary is complete.

sp3n05/ #
```

8.7 Recovering from a missing or corrupt PSSP DCE keyfile

PSSP Trusted Services keyfiles are stored under the main path, `/spdata/sys1/keyfiles/`.

The keyfiles are created during PSSP security installation and configuration in a dce security mode (dce or dce:compat), and, under normal conditions, this directory will exist and will contain the following directories:
Under each of these directories will be another directory. The name of this directory is that of the DCE hostname of a node or the control workstation:

```
sp3en5/spdata/sys1/keyfiles/ # ls -l
```

```
total 40
-dw-r-xr-x 3 root system 512 Oct 26 13:04 LoadL/
dw-r-xr-x 3 root system 512 Oct 26 13:04 mmfs/
dw-r-xr-x 3 root system 512 Oct 26 13:04 ppe/
dw-r-xr-x 4 root system 512 Oct 26 13:06 rsct/
dw-r-xr-x 3 root system 512 Oct 26 13:04 ssp/
```

Under the DCE hostname directory will be the PSSP DCE keyfile(s) for a given service or services:

```
sp3en5/spdata/sys1/keyfiles/ssp/ # ls -l
```

```
total 32
-dw-r-xr-x 2 root system 512 Nov 17 12:29 sp3en5
```

```
sp3en5/spdata/sys1/keyfiles/ssp/sp3en5/ # ls -l
```

```
total 64
-rw------- 1 root system 298 Oct 27 23:43 css
-rw------- 1 root system 382 Oct 27 23:43 pmand
-rw------- 1 root system 407 Oct 27 23:44 sp_configd
-rw------- 1 root system 397 Oct 27 23:44 spbgroot
-rw------- 1 root system 382 Oct 27 23:43 spmgr
-rw------- 1 root system 407 Oct 27 23:43 switchtbld
-rw------- 1 root system 152 Oct 26 12:22 sysctl
```

### 8.7.1 Keyfile differences

The keyfiles on the control workstation and on the nodes differ in two ways:

1. On the control workstation, the ssp/ directory will contain at least two subdirectories. In addition to the DCE hostname directory, a second directory will appear, and it will be named the same as the SP default partition. If more than two subdirectories exist, their names will correspond to the names of the remaining PSSP partitions.

The ssp/(partition name) subdirectories contain a DCE keyfile for each sdrd in the system.

The example below relates to a system running with three partitions, secsys1, secsys2, and secsys3.
2. There are keyfiles that will only appear on the control workstation. These keyfiles are for sdr and hardmon, as shown here:

```
8.7.2 Missing keyfile example

In the event that a PSSP DCE keyfile does not exist in its expected location, it can be determined whether the keyfile was removed (deleted) from the host or not created during the installation and configuration process.

On the host where the DCE PSSP keyfile is missing, run `create_keyfiles` in verbose mode. If the command is being run for keyfiles specific to the control workstation, ensure that the `-c` flag is specified on the `create_keyfiles` command.

---

Note

Even if the SP system is not configured for LoadLeveler, POE, GPFS, or an SP switch, PSSP DCE keyfiles are created for these services, including a css keyfile for use with PSSP switch services/commands. This does not pose a security exposure because the keyfiles can only be read and used by a root ID, and are valid only within the scope of their associated applications.
If the keyfile did exist but was deleted, an attempt to create a new keyfile will fail because the local DCE service will already contain a DCE keytab object for the missing keyfile.

In the example below, sysctl is the missing keyfile.

```
sp3en0/.../keyfiles/ssp/sp3en0/ # The keyfile catalog contains an entry for sysctl
sp3en0/.../keyfiles/ssp/sp3en0/ # dcecp -c keytab cat
.../sp_cell/hosts/sp3en0/config/keytab/self
.../sp_cell/hosts/sp3en0/config/keytab/LoadL/sp3en0/GSmonitor

.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # The physical sysct1 keyfile is missing
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ ls -l
```

```
total 48
-rw------- 1 root system 223 Oct 27 23:38 css
-rw------- 1 root system 229 Oct 27 23:38 pmend
-rw------- 1 root system 244 Oct 27 23:38 sp_configd
-rw------- 1 root system 238 Oct 27 23:38 spbgroot
-rw------- 1 root system 244 Oct 27 23:38 switchtbld
```

Attempts to re-create the missing keyfile:

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # create_keyfiles -v
Running "check_prereqs" subroutine ...
Checking state of DCE ...
Running "parse_defaults" subroutine ...
Parsing spsec_defaults file ...
Running "parse_overrides" subroutine ...
Running "create_keys" subroutine ...
Keyfile /spdata/sys1/keyfiles/LoadL/sp3en0/GSmonitor already exists.
Keyfile /spdata/sys1/keyfiles/LoadL/sp3en0/Kbdd already exists.
Keyfile /spdata/sys1/keyfiles/LoadL/sp3en0/Master already exists.
...
Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/spbgroot already exists.
Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/switchtbld already exists.
Running "do_keytab_work" subroutine ...
Creating keytab object, "ssp/sp3en0/sysctl" and randomizing keys.

*********************************************************************/usr/lpp/ssp/bin/create_keyfiles: 0016-511 The dcecp command below returned non-zero return code.
/opt/dcelocal/bin/dcecp -c keytab create ssp/sp3en0/sysctl -storage
/spdata/sys1/keyfiles/ssp/sp3en0/sysctl -data
    { ssp/sp3en0/sysctl plain 1 "svc_pwd_was_here" }
Command output is:
Error: msgID=0x113DB0CE Cannot create object; already exists
```

8.7.3 Re-creating missing keyfile

To re-create the missing keyfile with a random password value, perform the following steps:
1. As root (which has access to the DCE self-host principal credentials) on the host with the missing key, delete the missing keyfile’s DCE keytab object:

```bash
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # Remove the sysctl keytab object from the local catalog
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ dcecp -c keytab delete .:/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ dcecp -c keytab cat | grep sysctl
```

2. After the keytab object is deleted, the principal and account must be deleted and re-created, which requires cell administrator authority.

From another DCE host in the same cell, such as the control workstation, log in as the cell administrator, and delete the principal and account.

```bash
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
# dcecp -c principal delete ssp/sp3en0/sysctl

# # Verify that the entries are gone:
# dcecp -c principal show ssp/sp3en0/sysctl
Error: msgID=0x1712207A Registry object not found
# dcecp -c account show ssp/sp3en0/sysctl
Error: msgID=0x1712207A Registry object not found
```

Re-create the principal and account. If it is being run on the control workstation, ensure that the -c flag is specified.

```bash
config_spsec
This command requires cell administrator authority. Continue? (y/n) y

Please enter cell administrator id to be added to ACL admin group: cell_admin

Your cell administrator password is required to create accounts. Please enter your cell administrator password:
```

Verify that the principal and account were re-created.
3. On the host where the keyfile is missing, create a new keyfile.

Run the PSSP script, create_keyfiles, in verbose mode as root holding the self-host principal credentials.

Messages regarding already existing keyfiles are expected. If the create_keyfiles command is being run for keyfiles specific to the control workstation, ensure that the -c flag is specified:

```
# dcecp -c prin show ssp/sp3en0/sysctl
{fullname {}}
{uid 1883}
{uuid 000075b-9dcd-21d3-9c00-0004ac493bac}
{quota unlimited}
{groups spsec-services}
# dcecp -c account show ssp/sp3en0/sysctl -all
{acctvalid yes}
{client yes}
{created /.../sp_cell/cell_admin 1999-11-18-10:30:41.000-05:00-----}
....
{lastchange /.../sp_cell/cell_admin 1999-11-18-10:30:41.000-05:00-----}
{organization spsec-services}
```

Verify that the keyfile physically exists and is in the keytab catalog:

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # create_keyfiles -v
Running "check_prereqs" subroutine ...
Checking state of DCE ...
Running "parse_defaults" subroutine ...
Parsing spec_defaults file ...
Running "parse_overrides" subroutine ...
Running "create_keys" subroutine ...
Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/loadL/sp3en0/GSmonitor already exists.
... Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/spbgroot already exists.
Keyfile /spdata/sys1/keyfiles/ssp/sp3en0/switchtbld already exists.
Running "do_keytab_work" subroutine ...
Creating keytab object, "ssp/sp3en0/sysctl" and randomizing keys.
```

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ ls -ltr
total 48
-rw------- 1 root system 148 Oct 30 13:02 css
-rw------- 1 root system 152 Oct 30 13:02 pmand
-rw------- 1 root system 162 Oct 30 13:02 sp_configd
-rw------- 1 root system 158 Oct 30 13:02 spbgroot
-rw------- 1 root system 162 Oct 30 13:02 switchtbld
-rw------- 1 root system 154 Nov 18 11:26 sysctl
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # dcecp -c keytab cat | grep sysctl
/.../sp_cell/hosts/sp3en0/config/keytab/ssp/sp3en0/sysctl
```

Chapter 8. Problem isolation and resolution 389
4. Verify that keyfile can be used to log in to DCE. As root, use the PSSP dsrv tgt command to verify this.

```
dsrmv
```

\textit{dsrv tgt} can be used with other PSSP service/principal name pairs. (Refer to the \textit{dsrv tgt} command reference for details.)

5. In this example, the sysctld keyfile was re-created. To ensure that the sysctld (daemon) can use the new keyfile, stop sysctld (if it is not already stopped), and then start sysctld. (sysctld shouldn’t have any problems using the new keyfile, given that the \textit{dsrv tgt} validation in the previous step was successful.)

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # startsrc -s sysctld -a '-d'
0513-059 The sysctld Subsystem has been started. Subsystem PID is 21636.
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # lssrc -s sysctld
Subsystem Group PID Status
sysctld 21636 active
```

Now, verify that sysctld is working with the dce method. (In this example, the command was issued in a dce-only environment.)

```
sp3en0/spdata/sys1/keyfiles/ssp/sp3en0/ # sysctl whoami -v
DCE: /.../sp_cell/hosts/sp3en0/self
K4: AIX: root
```

For other PSSP services whose keyfiles were re-created, run the appropriate client commands against those services. Review the output of the commands and their corresponding server logs for normal processing indications (responses and entries) in a dce mode.

### 8.8 DCE client/server problem determination

This purpose of this section is to show the types of DCE client/server problems that can arise in an SP system.
8.8.1 DCE clients cannot communicate with DCE servers

There is a variety of circumstances under which clients cannot communicate with DCE servers.

8.8.1.1 Point-to-point connection checks

- Verify that the host or hosts running the DCE master servers respond to AIX ping -R requests from the client host, and vice versa.
- Verify that the AIX traceroute commands to the master servers complete successfully, and vice versa.
- In addition to the AIX ping command, the DCE cell ping command pings the machines running the DCE master and replica Security, CDS and DTS servers. (If a server is unavailable, its name is listed in the output of the command.)

All Security and CDS servers are reachable and running:

```
sp3n05/ # dcecp -c cell ping
DCE services available.
```

CDS servers cannot be contacted:

```
sp3en0/ # dcecp -c cell ping -replicas
Error: msgID=0x10D0A3EC Unable to communicate with any CDS server
```

Security server cannot be contacted:

```
sp3en0/ # dcecp -c cell ping
Error: msgID=0x1131F2FD Registry server '.../sp_cell/subsys/dce/sec/sp3en0' is not available.
```

In the following, the CDS server was not available and generated a slightly different error message than the previous check done on a different host.

```
sp3n05/ # dcecp -c cell ping -replicas
Error: msgID=0x16C9A016 Communications failure
```

The following response is a good indication that Security and CDS servers are not running anywhere in the cell:

```
sp3n05/ # dcecp -c cell ping -replicas
Warning: Not able to acquire initial login context.Error: msgID=0x1 Unknown message number 0x00000001
```
8.8.1.2 Things to check

Check the following on a DCE client machine:

- Verify that all configured DCE clients are running.

  DCE clients that are configured and should be running but are not running, should be started. In particular, the core DCE client services, RPC, Security, and CDS, should all be running:

```
sp3n05/ # whoami ; klist | grep lob
root
  Global Principal: /.../sp_cell/hosts/sp3n05/self

sp3n05/ # show.cfg
Gathering component state information...

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>

The component summary is complete.

sp3n05/ # start.dce sec_cl
Gathering current configuration information...  
Start of DCE Host, sp3n05, will now begin.  
Starting the Security client...  
The Security client was started successfully.  
Gathering component state information...  

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>

The component summary is complete.  
Start of DCE Host, sp3n05, was successful.  
Start completed successfully.

- Verify that file system(s) that DCE requires is/are not full, that is, at 100 percent capacity (/var or /var/dce; /var/dce/directory):
If one or more of the required file systems is full, DCE will stop responding. To correct this situation, alleviate the space constraint problem, stop DCE, run clean_up.dce, and then start DCE:

- Verify that secval is running. secval refers to the security validation service.

The security validation service (secval) has the following major functions:

- It maintains a login context for the host's self-identity, which includes periodic changes to the DCE self-host's key (password)
- It validates and certifies to applications, usually login programs, that the DCE security daemon, secd, is legitimate

Clients (including remote clients, local servers, host logins, and administrators) all need the security validation service to make sure that the secd process being used by the host is legitimate. The security validation service establishes the link in a trust chain between applications and secd so that applications can trust the DCE security mechanism.

The secval status and ping commands are used to determine:

- The ping operation validates the credentials returned by a security service. This operation is rarely invoked but can be used to verify that secd is trusted. The operation returns 1 if the credentials are valid and 0 if they are not.
- The status operation checks for an active secval. The status operation returns 1 if the security validation service is enabled and 0 if it is not.

The secval activate command activates a security validation service. If the service is already enabled, an error is returned but does not impact secval processing. This operation returns an empty string on success.
One of the impacts of secval not being active is that the self-host (machine) principal's DCE context will eventually expire. When this occurs, dced processing on the client machine is impacted. In fact, the security client, dced, terminates after the self-host context expires.

The example below shows how an expired self-host identity looks via klist, how a secval activate fails under this state, and, finally, how starting the security client reestablishes the self-host identity.

Normal self-host credentials.

Expired self-host credentials.
Reestablishing self-host context.

In the event that the self-host principal's credentials become corrupt or are unusable, they can be re-created by stopping and then starting the Security client. Note that attempting to refresh the credentials via a secval deactivate and secval activate sequence will fail given that the credentials are corrupt.

• In the event that the self-host principal's credentials become corrupt or are unusable, they can be re-created by stopping and then starting the Security client. Note that attempting to refresh the credentials via a secval deactivate and secval activate sequence will fail given that the credentials are corrupt.
Verify that the pe_site file exists and is correct.

The pe_site file is located in /etc/dce/security/pe_site and is stored in plain text. The contents of the file are in a required DCE format but can be edited with a text editor.

The easiest method for updating the pe_site file is through the dcecp command, secval update, which is issued by root as the self-host principal. (The command is issued on the host where the pe_site file is to be updated.)

The secval update command will create a valid pe_site when the file does not exist; when the existing file is: zero (0) length, corrupt, or does not contain a valid IP address to a Security server in the cell (but is otherwise properly formatted).

For the last case, secval update will create a valid pe_site file, but it may take several minutes for the process to complete. (It needs to dynamically
locate a security server in the cell.) Under normal conditions, the secval update process only takes seconds.

If, for some reason, secval update does not complete successfully, a pe_site from another host within the cell can be copied to the problem host. This file copy alternative is quicker than restoring the pe_site file from a back-up, which may be a requirement under conditions where the situation is critical. However, simply copying the file from another source might not solve the problem.

If the DCE cell is running with master and replica Security servers, and the contents of the pe_site differ radically from host-to-host due to security server "weighting" (see the "Preferred Security Server Replica" section of the book, DCE Administration Guide--Core Components, for details), then the copied pe_site file might not contain an address that is reachable from the problem host. However, this would only be an issue where pe_site files differ radically between DCE hosts within the same cell. For a cell running with a pe_site that is identical among machines (regardless of the number of Security servers they contain), copying the pe_site from another host to a problem host is not an issue.

In most circumstances, though, the secval update command is the quickest and most reliable approach.

```
sp3n05/ # whoami ; klist | grep lob
root
     Global Principal: .../sp_cell/hosts/sp3n05/self
sp3n05/ # cd /etc/dce/security
```

In the next example, the pe_site does not contain a valid Security server IP address. The secval update command does re-create the file properly, but it takes over two minutes for the local DCE client to locate a Security server. Normally, it only takes seconds to update the file.
In this example, pe_site is already valid but is being refreshed to pick up any recent changes.

In this example, the pe_site file does not exist.

In this example, pe_site is zero (0) length.
In this example, `pe_site` does not contain any valid data.

When DCE is installed and configured on a system without first creating a separate `/var/dce` file system, which means DCE was configured into `/var`, creating and mounting a `/var/dce` file system after the fact will prevent DCE from initializing properly. To correct the problem, unmount `/var/dce`, remove the file system (to avoid this problem in the future), run `clean_up.dce`, and then start DCE.

The file, `/etc/filesystems`, will contain a stanza for `/var/dce` in this scenario. The AIX command, `lsfs` (without any flags), will display and format the entries in `/etc/filesystems`.

- When DCE is installed and configured on a system without first creating a separate `/var/dce` file system, which means DCE was configured into `/var`, creating and mounting a `/var/dce` file system after the fact will prevent DCE from initializing properly. To correct the problem, unmount `/var/dce`, remove the file system (to avoid this problem in the future), run `clean_up.dce`, and then start DCE.

The file, `/etc/filesystems`, will contain a stanza for `/var/dce` in this scenario. The AIX command, `lsfs` (without any flags), will display and format the entries in `/etc/filesystems`.
lsfs, used in conjunction with the egrep command, can narrow the results down specifically to DCE-related filesystems.

---

**Note**

Installing DCE filesets and then creating and mounting a /var/dce file system results in the same error as creating and mounting the file system after DCE had been installed and configured. This is due to symbolic links in the file system created during fileset installation.

```sh
sp3n05/ # mount -v 'jfs' /var/dce
sp3n05/ # df | grep dce
/dev/lv01 98304 95144 4% 17 1% /var/dce
sp3n05/ # start.dce
0x11315069: An error occurred creating the file /opt/dcelocal/tmp/cfgdce.sem.
Gathering current configuration information...
Start of DCE Host, sp3n05, will now begin.
Starting RPC...
0x11315a06: Unable to start RPC.
0x11315a03: The components on DCE host, sp3n05 did not start successfully.
0x113159fb: Start did not complete successfully.
Gathering component state information...

Component Summary for Host: sp3n05

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
</tbody>
</table>

The component summary is complete.
More information can be found in the configuration log:
/opt/dcelocal/etc/cfgdce.log.
sp3n05/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)

Kerberos Ticket Information:

klist: No credentials cache file found (dce / krb) (ticket cache /tmp/krb5cc_0)
sp3n05/ # unmount /var/dce
sp3n05/ # df | grep /var/dce
sp3n05/ # clean_up.dce
Cleaning up old and possibly corrupted DCE files.
Clean up completed successfully.
sp3n05/ # start.dce
Gathering current configuration information...
Start of DCE Host, sp3n05, will now begin.

Start of DCE Host, sp3n05, was successful.
Start completed successfully.
sp3n05/ # klist | grep lob
Global Principal: /.../sp_cell/hosts/sp3n05/self
sp3n05/ # rmfs /var/dce
rmfs: Logical volume lv01 is removed.
```
• Creating a /var/dce file system and then mounting it on a system already running with DCE configured into the /var file system will immediately disable DCE processing, and DCE will seem to “disappear” from the file system entirely.

To correct the situation, unmount the /var/dce file system, and then remove the logical volume from the system. As soon as /var/dce is unmounted, DCE will be available.

Before mounting /var/dce:

```bash
sp3n05/ # df
Filesystem  512-blocks Used Available Use% Mounted on
/dev/hd4  65535 10220 51780 20% /
/dev/hd2  917504 91690 822188 99% /usr
/dev/hd9var 131072 94848 36192 28% /var
/dev/hd9  262144 252224  9918  4% /tmp
/dev/hd1 18192 17840  352  1% /home

sp3n05/ # ls -l /var/dce
total 12
-rw-r--r-- 1 root system 512 Nov 11 18:06 adm
-rw-r--r-- 1 root system 512 Nov 11 18:06 audit
... 
-rw-r--r-- 1 root system 512 Dec 19 17:21 security
...
-rw-r--r-- 1 root system 512 Nov 11 18:06 web

sp3n05/ # show.cfg
Gathering component state information...

Component Summary for Host: sp3n05
Component                 Configuration State Running State
Security client    Configured Running
RPC                Configured Running
Directory client   Configured Running

The component summary is complete.
```

After mounting /var/dce:
After unmounting /var/dce.

• When DCE is installed and configured into a /var/dce file system, should /var/dce not get mounted after an AIX reboot, DCE will not start. Also,
DCE files and directories will seem to have “disappeared” from the file system until the file system is mounted.

To correct this situation, mount /var/dce, and then start DCE. To avoid this problem in the future, have the /var/dce file system automatically mounted at boot time.

The file, /etc/filesystems, will contain a stanza for /var/dce in this scenario. The AIX command, `lsfs`, (without any flags) will display and format the entries in /etc/filesystems.

`lsfs`, used in conjunction with the `egrep` command, can narrow the results down specifically to DCE-related filesystems.

Check the following on the DCE master server machine(s):

```
s3n05/ # show.cfg
G011315069: An error occurred creating the file /opt/dce/local/tmp/cfgdce.sem.
Gathering component state information...

Component Summary for Host: s3n05

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
</tbody>
</table>

The component summary is complete.
s3n05/ # df|grep dce
s3n05/ # ls -l /var/dce
total 0
s3n05/ # lsfs | egrep "VFS|dce"

Name Nodename Mount Pt VFS Size Options Auto Accounting
/dev/lv02 -- /var/dce jfs 98304 rw no no
s3n05/ # mount /var/dce
s3n05/ # df
Filesystem 512-blocks Free %Used Used %Mounted on
/dev/hd4 65536 10584 84% 2041 13% /
... 
/dev/hd4 98304 92776 6% 127 2% /var/dce
s3n05/ # ls -l /var/dce
total 104
dmce-vr-x 4 root system 512 Nov 15 14:36 adm
... 
d3n05/ # start.dce
Gathering current configuration information...
... 
Start of DCE Host, c94n01.ppd.pok.ibm.com, was successful.
Start completed successfully.
s3n05/ # lsfs | egrep "VFS|dce"

Name Nodename Mount Pt VFS Size Options Auto Accounting
/dev/hd4 -- /var/dce jfs 98304 rw no no
s3n05/ # chfs -A'yes' /var/dce
s3n05/ # lsfs | egrep "VFS|dce"

Name Nodename Mount Pt VFS Size Options Auto Accounting
/dev/hd4 -- /var/dce jfs 98304 rw yes no
```

Chapter 8. Problem isolation and resolution 403
• Verify that all configured DCE master servers are running.
DCE servers that should be running, but are not, should be started. In particular, the core DCE servers, Security, and CDS, should be running.

```bash
sp3en0/ # klist | grep lob
Global Principal: /.../sp_cell/hosts/sp3en0/self

sp3en0/ # show.cfg
Gathering component state information...

Component Summary for Host: sp3en0

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Master server</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Initial Directory server</td>
<td>Configured</td>
<td>Not Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>

The component summary is complete.

sp3en0/ # start.dce cds_svr
Gathering current configuration information...
Start of DCE Host, sp3en0, will now begin.
RPC is already running.
The Security Master server is already running.
The Security client is already running.
The Directory client is already running.
Starting the Initial Directory server...
The Initial Directory server was started successfully.
Gathering component state information...

Component Summary for Host: sp3en0

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Master server</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Initial Directory server</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>

The component summary is complete.
Start of DCE Host, sp3en0, was successful.
Start completed successfully.
```

• Verify that the file systems DCE requires are not full, that is, 100 percent capacity (/var or /var/dce, or /var/dce and /var/dce/directory).
• IP Forwarding is enabled properly (if in use).
• Network interfaces are up and functioning properly.
• Verify that all needed file systems are mounted.
The file, /etc/filesystems, will contain a stanza for /var/dce and /var/dce/directory, one for the Security server (/var/dce), and one for the
CDS server (/var/dce/directory). The AIX command, \texttt{lsfs} (without any flags), will display and format the entries in /etc/filesystems.

\texttt{lsfs}, used in conjunction with the egrep, can narrow the results down specifically to DCE-related file systems as shown here. A value of \texttt{yes} under the \texttt{Auto} column indicates that the file system is automatically mounted at boot time. (Refer to the earlier "DCE client machine" subsection for examples of how to set a file system to be automatically mounted at boot time.)

![lsfs output]

8.8.2 General DCE checkpoints

Some of the following items are condensed adaptations of the material found in Chapter 3. "Problem Determination" of the \textit{IBM DCE for AIX Problem Determination Guide}. The reader is encouraged to refer to that guide for complete details, additional DCE items to be checked, and associated examples.

- **DCE Health Check**
  - Verify that all the necessary processes and files are running on servers and clients.
  - Check the DCE error and fatal logs to see what errors have occurred.
  - Check the functionality of the DCE core servers. This includes CDS servers, security servers, and application server entries in CDS and Security databases.

- **Check Your DCE Identity**
  - Be aware of the fact that the root user on AIX does not automatically have more permissions in DCE than another DCE principal. Only if you are root and do not log in to DCE do you inherit machine principal and credentials, which have limited permissions in the DCE environment.
  - Check the expiration of your credentials (via klist). Expired credentials have limited authority in a DCE environment.
  - When in doubt, refresh the credentials by logging back in to DCE, or issuing a kinit for an already established context. Refer to an earlier subsection for steps on "refreshing" the self-host principal credentials.

- **Check the Security Services**
- A slow or congested network adds some additional response time.
- Are all of the Security Server replicas working; are they accessible by the client system?

- Check CDS Services
  - The quickest, most thorough check for CDS is to list all directories and objects in the CDS namespace, via cdsli -world -C low.
  - Use the cdscp show cell command to display information about the CDS Servers. It displays condensed output, including the IP addresses of the master and replica CDS Servers.
  - The dcecp cdscache show operation displays address information about a clearinghouse and servers stored in the local cache of the machine. By getting an answer from this command, the assumption is that the cache on the local machine is valid.

- Check Access Permissions
  - Resources in DCE, such as directories and objects, have Access Control Lists (ACLs) associated with them. Error messages pointing to ACL problems usually include a clue in the text saying, for example, that a user does not have enough access permissions to do a specific operation.
  - The PSSP Sysctl and Hardmon services protect resources in a dce security mode via PSSP DCE group memberships and DCE ACLs.

- Time Skew
  - If the time skew between the DCE client and the DCE master servers is greater than five minutes, the servers will reject the client requests until time is brought within a five minute tolerance. The DCE Problem Determination guide provides a set of suggested corrective steps for this situation.

8.9 Other DCE problem determination tips

The following sections contain additional DCE problem determination tips.

8.9.1 DCE /etc/dce/cfg.dat file

During the local portion of the DCE configuration process (as opposed to the admin portion of the configuration), DCE creates the file, /etc/dce/cfg.dat. This file contains the results of the local configuration attempt.

The /etc/dce/cfg.dat file contains two fields that relate to:
Chapter 8. Problem isolation and resolution

1. The DCE component, as specified by the DCE short name (this is also the name used with the DCE commands config.dce, unconfig.dce, start.dce, and stop.dce). Refer to the DCE Administration Commands Reference for a complete list of DCE component short names.

2. A numeric value indicating the state of the DCE component:
   - 0 = not configured
   - 1 = partially configured
   - 2 = configured

By inspecting the contents of this file, the last known configuration state of a DCE host can be determined. Note, however, that the data in this file doesn't provide any indication of the current operational state of DCE services on host, for example, running, not running, unknown, and so on. Use the show.cfg command to determine both the configured and operational states of DCE services on a host.

The contents of this file should not be manually altered because DCE uses the data for various operations.

If you're wondering, "I shouldn't alter the contents of this file. I should use show.cfg to find out the configuration and operational states of a DCE host. Why should I even care about this file?", here's the answer: It is one more item that can be checked during DCE problem determination. Plus, an SP administrator can extract the configuration state of DCE clients on all nodes in the system in a quick, clean way.

The examples below show what /etc/dce/cfg.dat looks like for client configurations in fully-configured, partially-configured, and unconfigured states, and for fully-configured Security and CDS servers.

8.9.1.1 Fully-configured client

Fully-configured DCE client: rpc, cds_cl, and sec_cl. DFS and DTS (Distributed Time Service) components are not installed. This is representative of the "core" DCE client components. DTS and DFS clients are not required for a minimum DCE client configuration.
8.9.1.2 Partially-configured DCE client
The following display shows a partially-configured DCE client:

```
sp3n06/ # show.cfg
Gathering component state information...

Component Summary for Host: sp3n06
Component          Configuration State Running State
Security client    Configured    Running
RPC                Configured    Running
Directory client   Configured    Running

The component summary is complete.
```

```
sp3n06/ # cat /etc/dce/cfg.dat
dced:rpc 2
dced:sec_cl 2
cdsadv 2
```

On a host where DCE has been unconfigured or for a host where DCE file sets are installed but DCE is not configured, /etc/dce/cfg.dat does not exist:

```
sp3n10/ # show.cfg
Gathering component state information...

Component Summary
Component          Configuration State Running State
No DCE or DFS components are configured.

sp3n10/ # cat /etc/dce/cfg.dat
cat: 0652-050 Cannot open /etc/dce/cfg.dat.
```

8.9.1.3 DCE Security and CDS servers
A host where fully-configured DCE Security and CDS servers are running:
8.9.1.4 Using dsh to display /etc/dce/cfg.dat

When using dsh -avG in conjunction with dshbak -c to display the configuration state of DCE clients on all nodes in the system, consider the following:

dsh -avG will work over all nodes in the system, regardless of their partition association. The dshbak program collapses identical output from more than one host so that identical results are displayed only once. Results that are not identical can be quickly identified, along with their hosts.

In this example, node sp3n10’s DCE configuration is in a partial state. Its RPC client is fully-configured (as indicated by the 2), but its Security client is only partially configured (as indicated by the 1). Due to the partial state of the Security client, the CDS client was never configured, as indicated by the absence of an entry for the CDS client. (If some level of CDS client configuration had taken place, it would have been represented by a cdsadv entry.)
8.9.2 DCE CDS client configuration failure

When a CDS client configuration (local portion) takes 60 minutes to fail, it is, usually, an indication that the CDS server TCP/IP hostname is not specified or is incorrect. (The local portion of the CDS client configuration takes place after a successful DCE admin configuration for the same client.)

The /opt/dcelocal/etc/cfgdce.log will contain many entries for the message, *Requested entry does not exist*.

```
sp3en0/ # dsh -avG "cat /etc/dce/cfg.dat" | dshbak -c
HOSTS ____________________________________________________________
sp3n02 sp3n04 sp3n06

 dced:rpc 2
dced:sec_cl 2
cdsadv 2

HOSTS ____________________________________________________________
sp3n08

 dced:rpc 2
dced:sec_cl 1

Waiting up to 60 minutes for the directory server.
dir list: ./:
msgID=0x10D0A3F4 Requested entry does not exist
dir list: ./:
msgID=0x10D0A3F4 Requested entry does not exist
dir list: ./:
msgID=0x10D0A3F4 Requested entry does not exist
....
```

The DCE configuration looks like the following:
The DCE Problem Determination guide for message 0x10D0A3F4 states that the requested CDS entry does not exist or the DCE configuration does not have any permissions to the entry.

But, the entry (directory) does exist and the permissions to the directory would allow the sp3n12 self-host principal to update items in this directory. This can be validated by doing the following:

If the CDS server was not running, or if it did not contain the entry that is supposedly missing, the dcecp commands we just used would have failed. Had the ACL permissions been incorrect, the entry for "user
hosts/sp3n12/self” would have been absent or lacked modify/write/insert permissions.

The problem is that since the client configuration is taking place on one subnet and the CDS server is on another subnet, and the CDS client configuration process wasn't explicitly told where the CDS server was located, the Requested entry does not exist message is actually correct. After all, if the CDS server can't be located, the entry it is looking for does not exist. (A more appropriate message would be to indicate that the CDS server could not be contacted.)

So, the actual problem is that the DCE CDS client configuration program cannot locate the CDS, not that the entry does not exist or that its permissions are invalid.

To correct the problem, rerun the local DCE configuration for only the CDS client and explicitly specify the TCP/IP hostname where the CDS server is running.

8.9.3 DCE local configuration - Unable to get key version

During the local portion of client configuration, DCE builds the machine's self-host principal key file based on the detected number of network interfaces.

If a host contains several network interfaces, such as ethernet, token ring, and ATM, the DCE local configuration will attempt to create in its key file a host/ and ftp/ pair for each interface. At a minimum, the self-host key file will contain at least one host/ and ftp/ pair, and they relate to the DCE hostname that the machine was configured under.

For example, host sp3n06 was configured with DCE hostname sp3n06, and the TCP/IP hostname of sp3n06.itso.ibm.com:

```
sp3n06/ # show.cfg
Gathering component state information...

Component Summary for Host: sp3n06

<table>
<thead>
<tr>
<th>Component</th>
<th>Configuration State</th>
<th>Running State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security client</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>RPC</td>
<td>Configured</td>
<td>Running</td>
</tr>
<tr>
<td>Directory client</td>
<td>Configured</td>
<td>Running</td>
</tr>
</tbody>
</table>

The component summary is complete.
```

The self-host key file contains host/ and ftp/ entries for sp3n06.
The multiple key versions (1 and 2) exist because part of DCE’s client configuration is to automatically generate a new key version (password) immediately, that is, it changes its password immediately.

In general, there are only two cases when DCE cannot place in its key file host/ and ftp/ entries for interfaces other than the one specified as the TCP/IP hostname.

- The interface was down and could not be detected at the time the DCE configuration ran.
  
The solution is to initialize all needed interfaces on the host so that they’re detected as "up" by netstat -i, then run kerberos.dce -type local. Both steps must be done as root. Even if multiple interfaces must be initialized, only one invocation of kerberos.dce is required.

- The admin portion of the configuration was not performed for an interface.
  
The solution is to run the kerberos.dce -type admin {interface name} command for the missing interface(s), then run kerberos.dce -type local on the host. Running kerberos.dce with the admin flag will prompt the invoker for a DCE cell administrator password. The local flag requires the invoker to be root. Even if multiple kerberos.dce -type admin {interface name} commands must be run, only one invocation of kerberos.dce is required.

In the following example, sp3n06 is configured as a DCE client. It contains multiple network interfaces, ethernet, and CSS, but only the ethernet interfaces appear in the DCE self-host key file.
The DCE configuration log, /opt/dcelocal/etc/cfgdce.log, contains the following errors relating to the CSS interface.

There are no Registry accounts for the host/sp3sw06.msc.itso.ibm.com and ftp/sp3sw06.msc.itso.ibm.com, because kereberos.dce -type admin was not run for the CSS interface.

There are no Registry accounts for the host/sp3sw06.msc.itso.ibm.com and ftp/sp3sw06.msc.itso.ibm.com, because kereberos.dce -type admin was not run for the CSS interface.

Run kereberos.admin for the CSS interface, and then check that host/ftp accounts were created:
Run kerberos.dce -type local, and then check that the self-host key file contains new host/ftp entries for the CSS interface:

The *already exists* messages are expected since the key file already contains entries for these interfaces. Each time kerberos.dce -type local is invoked, it dynamically builds a list of active interfaces on a host and then attempts to create key file entries for all items in the list. DCE doesn't modify entries that already exist; so there's no harm in running kerberos.dce -type local. (Refer to the *DCE Administration Commands Reference* for complete details on kerberos.dce.).
Fortunately, PSSP V3.2 installation and configuration routines perform `kerberos.dce -type admin` for all needed interfaces stored in the SDR (at the time of PSSP installation and configuration).

When network interfaces are added to SP nodes after PSSP DCE installation and configuration has taken place, the second solution on the preceding page can be used to add the needed entries to the DCE Registry and the self-host key files.

### 8.10 DCE and Kerberos V4 credentials

DCE and Kerberos V4 credentials, also referred to as "tickets", are an important part of PSSP security. PSSP Trusted Services, AIX remote commands, and DCE commands all support DCE credentials, while only PSSP Trusted Services and AIX remote commands support Kerberos V4 credentials. Credentials, or just creds, establish your identity. It is information that defines who you are.

An identity is established by authenticating to some service that provides you with a way to prove who you are to other services or applications. (This would be akin to obtaining a driver's license and presenting it to someone to prove your identity.)

It is possible that a single SP user can have up to three identities at any given time: AIX, DCE, and Kerberos V4. (This is true for both SP administrators and non-administrators.)

In the case of AIX, this is the AIX user ID/group ID pair assigned at the creation of the ID and established with each AIX login. When a user successfully logs in to a system with a valid user ID and password, the AIX user ID/group ID pair is established for the user by AIX. Once logged in to AIX, a user can change to another AIX identity by issuing the AIX `su` command and supplying the appropriate password.

AIX user IDs are managed across an SP system through PSSP-supplied software or through customer supplied controls.

A DCE identity is the unique DCE principal name within a specific DCE cell. A DCE identity is established after an AIX user successfully issues a `dce_login`. (For AIX/DCE integrated login environments, a separate `dce_login` is not required to establish an initial DCE principal identity. A DCE identity is established as part of the AIX login.) A DCE cell administrator creates DCE user principals and accounts based on the site's user policies. Once logged in as a DCE principal, a user can change to another DCE identity by issuing a
dce_login for another principal name and supplying the appropriate password.

PSSP installation and configuration provides scripts that will create PSSP-specific DCE principals/accounts, groups, and organizations for use by PSSP Trusted Services when the services run under a dce security configuration (dce or dce:compat).

All DCE principals and accounts are managed by the DCE cell administrator. PSSP does not supply management routines for DCE principals, accounts, groups, or organizations. Management of these DCE items is covered extensively by DCE administrative facilities.

As with DCE, to establish a Kerberos V4 principal identity, an AIX user must issue a successful \texttt{k4init} command. (Refer to Chapter 3, “Managing and Using SP Security Services”, subsection "Integrating Login For Kerberos V4 with AIX," in the \textit{PSSP: Administration Guide}, SA22-7348, for details on how to set-up the AIX/PSSP Kerberos V4 integrated login.) PSSP installation and configuration scripts create specific Kerberos V4 principals used by PSSP services in a compat security configuration (compat or dce:compat). An SP administrator creates additional Kerberos V4 principals based on the site's user policies and requirements. Once logged in as a Kerberos V4 principal, a user can change to another Kerberos V4 principal by issuing a \texttt{k4init} for another principal name, and supplying the appropriate password.

All Kerberos V4 principals are managed by the SP administrator. PSSP provides a Kerberos V4 administrative facility to manage its database of principals.

In the case of DCE and Kerberos V4 credentials, the state of the held credentials at the time of command execution will impact whether or not PSSP services or AIX secure remote command services will respond to the request. (Examples of DCE and Kerberos Version 4 credential states are covered elsewhere in this chapter.)

Valid credentials are considered to be credentials that are not expired, not corrupt or malformed, and are in the credentials structure recognized by a service that has been coded to accept a particular credential type.

\textbf{8.10.1 Credentials cache locations}

The following sections contain examples of both DCE and Kerberos V4 credential cache location problems, tips, and corrective actions.
8.10.1.1 DCE

All DCE credential caches (also called ticket caches) reside in the directory, /opt/dcelocal/var/security/creds/. This location cannot be changed by the user at the time of DCE login. Each /opt/.../dcecred_XXXXXXXX file stores the tickets issued to a particular DCE principal (XXXXXXXX is an eight-digit hexadecimal number). A dcecred_XXXXXXXX file is created by invoking dce_login (or dce_login_noexec), the PSSP command dsrvtgt, or through an AIX/DCE integrated login to AIX. Only the user who invoked dce_login has read/write permission to the file. (For complete information on the dcecred_XXXXXXXX files refer to the DCE Administration Commands Reference, dcecred_* Files.)

dce_login and AIX/DCE integrated login automatically sets the environment variable, KRB5CCNAME (which stands for "Kerberos Version 5 credentials cache name"), to a location of the format:

FILE:/opt/dcelocal/var/security/creds/dcecred_XXXXXXXX

The dce_login command runs the shell specified in the AIX SHELL environment variable. Each time a dce_login is issued, the user is placed in a new shell environment (a subshell hereafter). The dce_login_noexec command does not run a new shell.

```
sp3n05/u/dero/ # env KRB5CCNAME
KRB5CCNAME: A file or directory in the path name does not exist.
sp3n05/u/dero/ # dce_login dero1
Enter Password:
DCE LOGIN SUCCESSFUL
$ env KRB5CCNAME
FILE:/opt/dcelocal/var/security/creds/dcecred_bf462800
```

dce_login_noexec and dsrvtgt return to the invoker of the string FILE:/opt/dcelocal/var/security/creds/dcecred_XXXXXXXX, and the user must set KRB5CCNAME manually. To use the dsrvtgt command as shown in the example below, the user must be root. (Refer to the PSSP Command and Technical Reference for complete details on dsrvtgt.)
Special DCE cache locations

The /opt/dcelocal/var/security/creds/dcecred_ffffffff ticket cache is special. It stores the tickets for the local machine's principal, known in the DCE Registry as hosts/{dce_hostname}/self. This is commonly referred to as the self-host or machine principal identity.

The root user on a machine automatically has access to these credentials and, therefore, appears as the DCE entity self-host until reauthenticating as a different DCE principal. A root user who has not reauthenticated will share the dcecred_ffffffff ticket cache with other root processes using the machine's self-host identity, such as dced and other root processes that do not explicitly issue a DCE login. Root users must be careful not to inadvertently destroy the self-host ticket cache (by issuing kdestroy -f or by deleting the actual credential files). When the self-host context is destroyed, local DCE services will be impacted significantly. The self-host credential context can be re-created by stopping and then starting the DCE services on the host.

Also, note that a root user does not have to set KRB5CCNAME to have access to the dcecred_ffffffff credentials. By virtue of being root, client commands that support DCE credentials know that in the absence of a set KRB5CCNAME variable, check for root credentials at /opt/dcelocal/var/security/creds/dcecred_ffffffff.
Exploiting RS/6000 SP Security: Keeping It Safe

/tmp/krb5cc_UID is the form of the cache file created when the DCE kinit command is invoked in the absence of a DCE login context. UID is the UNIX ID of the local user who invoked kinit.

```
sp3en0/ # whoami
root
sp3en0/ # env KRB5CCNAME
KRB5CCNAME: A file or directory in the path name does not exist.
sp3en0/ # klist
DCE Identity Information:
    Global Principal: /.../sp_cell/hosts/sp3en0/self
    Cell: 4b6608a0-9842-11d3-a5a0-02608c2d4a7f /.../sp_cell
    Principal: 00000066-9842-21d3-a500-02608c2d4a7f hosts/sp3en0/self
...
Identity Info Expires: 2000/02/04:13:40:17
Account Expires: never
Passwd Expires: never

Kerberos Ticket Information:
    Ticket cache: /opt/dcelocal/var/security/creds/dcecredffffffff
    Default principal: hosts/sp3en0/self@sp_cell
    Server: krbtgt/sp_cell@sp_cell
        valid 2000/02/04:03:40:17 to 2000/02/04:13:40:17
...
```

```
sp3en0/ # sysctl whoami
/home/sp_cell/hosts/sp3en0/self
```

```
sp3n06/u/dero/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)
Kerberos Ticket Information:
    klist: No credentials cache file found (dce / krb) (ticket cache /tmp/krb5cc_36586)
sp3n06/u/dero/ # kinit dero1
Enter Password:
sp3n06/u/dero/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)
Kerberos Ticket Information:
    Ticket cache: /tmp/krb5cc_36586
    Default principal: dero1@sp_cell
    Server: krbtgt/sp_cell@sp_cell
    valid 2000/02/04:11:59:42 to 2000/02/04:21:59:42
```

kinit is most commonly used to refresh an already established DCE context--one that was created by a DCE login. In fact, DCE credentials created by issuing only a kinit command are not considered valid by PSSP services, since there's no DCE context associated with the credentials. However, the credentials may be used with AIX remote commands under k5.
Deleting/removing DCE credentials

By default DCE credentials are not automatically deleted from a system when a user exits a DCE spawned shell, or exits AIX. Unless a host’s AIX log out functions have been customized to delete user DCE credentials upon exit, the credentials remain in the /opt/dcelocal/var/security/creds/ directory. Even when user credentials expire, the physical files remain.

The DCE command, kdestroy, deletes user credential files from the system. kdestroy doesn’t unset KRBSCCNAME if it’s set. If KRBSCCNAME is not set, the location /tmp/krb5cc_UID is referenced.
DCE provides a command, `rmxcred`, that purges expired credentials. This command will not remove individual tickets from caches that happen to contain both expired and unexpired tickets. However, it does delete credentials that represent failed DCE login attempts.

Note that `rmxcred` only deletes credentials located in the DCE credentials directory, `/opt/dcelocal/var/security/creds`. (The `kinit` command permits a user to specify a cache location other than the DCE default.)

Credentials removal is subject to other criteria controlled by the command-line options of `rmxcred`. (Refer to the DCE Administration Commands Reference for complete command details.)

Further, only a root user can invoke this command.

As a general rule, `rmxcred` should be run periodically to help control DCE credentials buildup. It can be added to cron and invoked automatically.

Depending on the number of credentials files in `/opt/.../creds`, `rmxcred` may take several minutes to complete. DCE has to open and inspect all credentials files to determine if they are valid candidates for removal.
When the file system that /opt/dcelocal/var/dce/security/creds resides in becomes full, issuing rmxcred can generate some error messages during its processing, though it will delete credentials that match removal criteria.

8.10.1.2 Kerberos V4
Kerberos V4 credential caches (also called ticket caches) are stored in the ticket file specified by the user's KRBTKFILE environment variable. If the KRBTKFILE variable is not set, the user's ticket is stored in the /tmp/tktUID file, where UID is the user identification number. Kerberos V4 credentials can be stored in any location for which the user has read/write access. A Kerberos V4 credential file is created by invoking the PSSP commands, k4init, rcmdtgt, or ksrvtgt, or through an AIX/PSSP Kerberos V4 integrated login. Only the user that invoked k4init, rcmdtgt, or ksrvtgt, or logged in via integrated Kerberos V4, has read/write permission to the file. (root notwithstanding, of course.) For complete information on PSSP Kerberos V4 credentials and cache locations, refer to the PSSP: Administration Guide, SA22-7348.

Note that both rcmdtgt and ksrvtgt require that KRBTKFILE be set prior to command invocation. Only a root user can successfully execute rcmdtgt and ksrvtgt. These commands are not intended for end-users. They were specifically designed for use by PSSP processes running as background tasks that require Kerberos V4 credentials during run time. As such, examples of the use of these commands are not included here. For more information on rcmdtgt and dsrvtgt, refer to the PSSP Command and Technical Reference, SA22-7351.
In the absence of an explicitly set KRBTKFILE variable, a root user on a machine will share the /tmp/tkt0 ticket cache with all other root users and

`root and Kerberos V4 cache locations`  
In the absence of an explicitly set KRBTKFILE variable, a root user on a machine will share the /tmp/tkt0 ticket cache with all other root users and
processes that have not explicitly set their own instances of KRBTKFILE. root users that destroy the /tmp/tkt0 ticket cache (by issuing k4destroy or deleting the file) impact all other root processes accessing /tmp/tkt0. A safe practice is to always set KRBTKFILE when obtaining Kerberos V4 credentials. This protects against another root user inadvertently deleting your root level Kerberos V4 credentials.

Deleting/removing Kerberos V4 credentials
By default, Kerberos V4 credentials are not automatically removed (deleted) from a system. Unless a host’s AIX log-out functions have been customized to delete user Kerberos V4 credentials upon exit, the credentials remain in either /tmp or the location specified in KRBTKFILE. Even when user credentials expire, the physical files remain.

The PSSP command, k4destroy, deletes user credential files from the system. k4destroy does not unset KRBTKFILE if it is set. If KRBTKFILE is not set, the location, /tmp/tktUID, is referenced.

```
sp3n08/u/dero/ # k4list
Ticket file: ./kv4creds/itso/kv401
Principal: dero@ITSO.IBM.COM

   Issued   Expires    Principal
Feb 4 13:39:37  Mar 5 13:39:37  krbtgt.ITSO.IBM.COM@ITSO.IBM.COM
sp3n08/u/dero/ # env KRBTKFILE
./kv4creds/itso/kv401
sp3n08/u/dero/ # k4list
Ticket file: ./kv4creds/itso/kv401
Principal: dero@ITSO.IBM.COM

   Issued   Expires    Principal
Feb 4 13:39:37  Mar 5 13:39:37  krbtgt.ITSO.IBM.COM@ITSO.IBM.COM
sp3n08/u/dero/ # k4destroy
Tickets destroyed.
sp3n08/u/dero/ # env KRBTKFILE
./kv4creds/itso/kv401
sp3n08/u/dero/ # k4list
Ticket file: ./kv4creds/itso/kv401
k4list: 2504-076 Kerberos V4 ticket file was not found
```
There is no Kerberos V4 equivalent of the DCE command, \texttt{rmxcred}. SP administrators must determine a site policy that defines how expired Kerberos V4 credentials files are handled. One approach is to run a cron job that periodically deletes all tkt* files from /tmp that match AIX non-root user IDs on the system. Simply deleting all credentials under /tmp that match tkt* is not a good solution. It would delete root's /tmp/tkt0, the cache files of PSSP services actively using credentials, and any other Kerberos V4 credentials in use.

### 8.10.2 Typical credential states

The following examples show what typical (normal) DCE and Kerberos V4 credentials look like and how the AIX remote command, \texttt{rsh}, and the PSSP command, \texttt{sysctl}, react to the state of the credentials.

All examples take place in a security environment of dce:compat/k5:k4:std.

DCE credentials were created by issuing a \texttt{dce\_login}. Corrective actions that state \textit{exit out of the DCE subshell} refer to exiting the shell environment created when \texttt{dce\_login} was issued.

#### 8.10.2.1 DCE and Kerberos V4 credentials are valid

Consider the following example in which DCE and Kerberos V4 credentials are valid:
Note that the Warning: Identity information is not certified message at the top of the credentials display is normal and does not impact how PSSP Trusted Services and AIX remote commands view the credentials. Only root users can obtain DCE credentials that are “certified”. For more information about certified DCE credentials, refer to the DCE Administration Commands Reference.

8.10.2.2 Neither DCE nor Kerberos V4 credentials exist
Consider the following example in which neither DCE nor Kerberos V4 credentials exist:
The corrective action to obtain new credentials is to issue dce_login and k4init.

After obtaining valid DCE and Kerberos V4 credentials, they are destroyed (deleted) through their respective credentials destroy commands:

The corrective action to obtain new credentials is to exit out of the DCE subshell, and issue dce_login and k4init.
8.10.2.3 Expired DCE and Kerberos V4 credentials

Consider the following example in which DCE and Kerberos V4 credentials are expired:

```
sp3en0/u/dero/ # klist
DCE Identity Information:
  Warning: Identity information is not certified
  Global Principal: /.../sp_cell/dero1
...
Kerberos Ticket Information:
  Ticket cache: /opt/dcelocal/var/security/creds/dcecred_caee2000
  Default principal: dero1@sp_cell

sp3en0/u/dero/ # k4list
  Ticket file: /tmp/tkt36586
  Principal: dero@ITSO.IBM.COM

  Issued       Expires       Principal
  Feb 3 17:33:36  Feb 3 17:38:36  krbtgt.ITSO.IBM.COM@ITSO.IBM.COM

sp3en0/u/dero/ # rsh sp3n06 date
kerberos: Couldn't get credentials for the server: Ticket expired.
rshd: 0826-813 Permission is denied.

sp3en0/u/dero/ # sysctl whoami -v
2502-605 Your DCE credentials have expired.
Authentication ticket expired
2502-605 Your Kerberos V4 credentials have expired.
```

The corrective action is to obtain new credentials, and issue `kdestroy` and `k4destroy`. Exit out of the DCE subshell, and issue `dce_login` and `k4init`.

8.10.3 Atypical credential states

The following examples show what atypical (abnormal) DCE and Kerberos V4 credentials look like and how the AIX remote command, `rsh`, and the PSSP command, `sysctl`, react to the state of the credentials.

This includes corrupt and malformed credentials, invalid cache locations, and attempting to access the credentials cache owned by another user.

Credentials rarely, if ever, get into a malformed or corrupt state on their own. They are usually tampered with by the credentials owner. Keep in mind that credentials are represented as AIX files owned by a user. The user has full read/write authority to those files. Users should not intentionally corrupt their credentials or, otherwise, directly manipulate them. PSSP services and AIX remote command services that require valid user credentials will reject a client's request when accompanied by corrupt or malformed credentials.
(PSSP and AIX will not view the user as authenticated, that is, the user will be seen as unauthenticated. DCE, however, accepts unauthenticated client requests. Only DCE permits very basic dcecp command support for unauthenticated users.)

Likewise, using invalid credential cache locations or attempting to access (use or steal) credentials owned by another user result in authentication failures and are promptly rejected by PSSP and AIX services.

All examples take place in a security environment of dce:compat/k5:k4:std.

DCE credentials were created by issuing a dce_login. Corrective actions that state exit out of the DCE subshell refer to exiting the shell environment created when dce_login was issued.

8.10.3.1 Credential variables reference non-existent locations
Consider the following example in which credentials' variables reference non-existent locations:

```
sp3en0/u/dero/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)

Kerberos Ticket Information:
klist: Credential cache name malformed (dce / krb) while getting default cache

sp3en0/u/dero/ # klist
Ticket file: junk
k4list: 2502-007 Can't find realm of ticket file: 2504-079 Bad Kerberos V4 ticket file format

sp3en0/u/dero/ # rsh sp3n06 date
kerberos: Couldn't get credentials for the server: Unsupported credentials cache format version number.

sp3en0/u/dero/ # rsh sp3n06 date
rshd: 0826-813 Permission is denied.

sp3en0/u/dero/ # sysctl whoami -v
2502-603 You do not have DCE credentials.
No currently established network identity for this context exists
2502-608 Kerberos V4 error in krb_mk_req: 2504-079 Bad Kerberos V4 ticket file format
```

The corrective action is to exit out of the DCE subshell. Unset KRB5CCNAME and KRBTKFILE. Set KRBTKFILE to a valid location or use the /tmp/tktUID default location. Issue dce_login and k4init.

8.10.3.2 Credentials variables reference zero-length file locations
Consider the following example in which credentials variables reference zero-length file locations:

```
```
The corrective action is to exit out of the DCE subsell. Unset KRB5CCNAME and KRBTKFILE. Set KRBTKFILE to a valid location or use the /tmp/tktUID default location, and issue dce_login and k4init.

8.10.3.3 Credentials variables reference files not in a valid format

Consider the following example in which credentials variables reference files not in a valid format:

```
sp3en0/u/dero/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)
Kerberos Ticket Information:
klist: Credential cache name malformed (dce / krb) while getting default ccache
sp3en0/u/dero/ # klist
Ticket file: junk
klist: 2502-007 Can't find realm of ticket file: 2504-079 Bad Kerberos V4 ticket file format
sp3en0/u/dero/ # rsh sp3en06 date
kerberos: Couldn't get credentials for the server: Unsupported credentials cache format version number.
spk4rsh: 0041-004 Kerberos V4 rcmd failed: 2504-079 Bad Kerberos V4 ticket file format.
rshd: 0826-813 Permission is denied.
sp3en0/u/dero/ # sysctl whoami -v
2502-603 You do not have DCE credentials.
No currently established network identity for this context exists
2502-608 Kerberos V4 error in krb_mk_req: 2504-079 Bad Kerberos V4 ticket file format
```
The corrective action is to issue `kdestroy` and `k4destroy`. Exit out of the DCE subsell. Unset KRB5CCNAME and KRBTKFILE. Set KRBTKFILE to a valid location or use the `/tmp/tktUID default location, and issue `dce_login` and `k4init`.

8.10.3.4 Credentials variables reference zero-length locations

The following example is the second variation:

```
sp3en0/u/dero/ # klist
No DCE identity available: No currently established network identity for this context exists (dce / sec)
Kerberos Ticket Information:
klist: Unsupported credentials cache format version number (dce / krb) while setting cache flags (ticket cache /opt/dcelocal/var/security/creds/dcecred_d1224300)

sp3en0/u/dero/ # k4list
Ticket file: /tmp/tkt36586
k4list: 2502-007 Can't find realm of ticket file: 2504-079 Bad Kerberos V4 ticket file format

sp3en0/u/dero/ # rsh c166n01 date
kerberos: Couldn't get credentials for the server: Bad format in credentials cache.
spk4rsh: 0041-004 Kerberos V4 rcmd failed: 2504-079 Bad Kerberos V4 ticket file format.
rshd: 0826-813 Permission is denied.

sp3en0/u/dero/ # sysctl whoami -v
2502-603 You do not have DCE credentials.
2502-608 Kerberos V4 error in krb_mk_req: 2504-079 Bad Kerberos V4 ticket file format
```

Corrective action is to issue `kdestroy` and `k4destroy`. Exit out of the DCE subsell. Unset KRB5CCNAME and KRBTKFILE. Set KRBTKFILE to a valid location or use the `/tmp/tktUID default location, and issue `dce_login` and `k4init`.

8.10.3.5 Credentials variables reference files not in a valid format

The following example is the second variation:
The corrective action is to issue kdestroy and k4destroy. Exit out of the DCE subsell. Unset KRB5CCNAME and KRBTKFILE. Set KRBTKFILE to a valid location or use the /tmp/tktUID default location, and issue dce_login and k4init.

8.10.3.6 Referencing files owned by others

Non-root user's credentials variables reference files that are owned by the root user.
The corrective action is to unset KRBTKFILE and KRB5CCNAME. Issue 
dce_login and k4init.

8.11 AIX and PSSP remote commands

An important part of SP administration is the ability to remotely manage 
nodes from a central host, commonly, the control workstation. Much of 
PSSP’s remote administration relies on the use of PSSP distributed remote 
parallel commands, such as dsh, pcp, and pexscr, and the AIX secure remote 
commands, rsh and rcp. In fact, most PSSP remote parallel commands 
ultimately issue an AIX rsh or rcp to perform the actual host-to-host 
communication. (There are other AIX secure remote commands in addition to 
rsh and rcp, namely, telnet, ftp, and rlogin. Only rsh and rcp are used in 
PSSP scripts. As such, the scope of AIX secure remote command discussion 
is limited to these two commands.)
The notion of remote command processing is not new to PSSP or AIX. Remote command processing has been a part of UNIX since its earliest implementations and continues to play a critical role in systems management and administration. Further, remote command processing is not limited solely to administrative duties, nor is it limited to root users. It is available to all users in the system, regardless of group memberships or intended roles. Both AIX and PSSP make remote command processing available to all users in a system. A non-root user has as much access to the PSSP `dsh` command and the AIX `rsh` command as a root user.

But, like many things in the computer arena, remote command processing is not without its security implications. By their very definition, remote commands provide the ability for a user on host X to remotely access host Y. Host X and host Y can be a continent apart, but as long as host Y allows the user from host X to access the system, the user can run programs, stop and start servers, and do anything that is permitted in a standard UNIX shell environment (provided the user has the proper authority, of course).

However, the real issue is not one of access to the commands themselves, but the requirements and controls that are a necessary part of using the commands. Just because remote command processing is supported in AIX and PSSP does not mean that a non-root user can suddenly run root-only programs. It simply means that the authentication and authorization concepts discussed earlier in the chapter also apply to the remote commands, but in terms of AIX authentication and authorization, which should not be confused with PSSP Trusted Services authentication and authorization.

Of particular concern to the SP administrator is how root level remote command processing is handled throughout the SP. From authentication considerations to access control lists (authorization files), knowing how AIX secure remote commands and PSSP remote parallel commands work (at a high level) and how to recover from error conditions relating to remote command processing is essential to effective SP administration.

### 8.11.1 Documentation/reference materials

Understanding and exploiting AIX and PSSP remote commands is all at once straightforward and slightly complicated. It is strongly recommended that SP administrators review the following documentation in order to see the entire remote command picture in the SP.

#### 8.11.1.1 AIX documentation

Consult the following AIX documentation:
• System User’s Guide: Communications and Networks: Communicating with Other Systems and Users

• System User’s Guide: Communications and Networks: TCP/IP Facilities, Terminology, and Application

• User’s Guide: Communications and Networks, Understanding the Secure Rcmds

• User’s Guide: Communications and Networks, Customizing TCP/IP Features, subsection "Using a .k5login File"


• The Commands Reference for the following: inetd, krshd, rcp, rsh, and rshd.

• The Communications Technical Reference, Volume 2, for the following subroutines: kvalid_user and rcmd.

The Files Reference for the following: .rhosts, /etc/hosts, /etc/hosts.equiv, /etc/inetd.conf, /etc/resolv.conf, and /etc/services.

8.11.1.2 DCE documentation
In the IBM DCE for AIX, Administration Commands Reference, look up the dce_login, kinit, kdestroy, and dcecred_* files.

8.11.1.3 PSSP documentation

In the PSSP: Command and Technical Reference, SA22-7351, look up the commands, chauthpar, dsh, dshbak, dsrvtgt, k4destroy, k4init, k4list, pcp (and other p* commands), and rcmdtgt.


8.11.1.4 Kerberos V5 documentation
The Kerberos FAQ, available at the following Web site:
http://www.faqs.org/faqs/kerberos-faq/general/
8.11.2 About the AIX authentication methods

Used directly by AIX remote commands (rcp, rsh, and so on) and indirectly by PSSP remote parallel commands that exploit AIX remote commands to achieve their functionality, such as dsh and pcp.

8.11.2.1 Partition level settings
1. Methods = k5, k4, std, and none (where none means that no authentication method is set; note that the SP requires and enforces that at least one AIX authentication method be enabled.)
2. SDR attribute value = auth_methods
3. SDR class value = Syspar
4. To query the AIX authentication methods stored in the SDR, for all partitions in the system, issue: splstdata -p

8.11.2.2 Local host settings
1. methods = k5, k4, std, and none (where "none" means that no authentication method is set; note that the SP enforces at least one AIX authentication method be set.)
2. Methods are stored on the local host in the ODM, in the file CuAt, under /etc/objrepos. (CuAt is also the name of the object class.)

8.11.2.3 Failover flow
The precedence ordering of the AIX authentication methods is the SP enforced order of k5:k4:std. When k5 processing fails, k4 processing is attempted. When k4 fails, std processing is attempted. When std fails, no further attempts are made at completing the remote command request.

In the case where two authentication methods are set, such as k5:k4, when k5 fails, k4 is attempted. When k4 fails, no further attempts are made at completing the remote command request.

The same holds true for the case where a single authentication method is set, such as k4. When k4 fails, no further attempts are made at completing the remote command request.

8.11.2.4 Querying the AIX methods
Use the following commands to query the AIX methods:

- Partition Level: lsauthpar (PSSP command)
- Local Host: lsauthent (AIX command)
8.11.3 About the remote command protocols

This section covers the remote command protocols, K5, K4, and std.

8.11.3.1 k5

The k5 method uses the Kerberos V5 protocol, implemented through Kerberos V5 libraries supplied by the Distributed Computing Environment (DCE) LPP, the AIX operating system, and DCE credentials, which are upgraded to Kerberos V5 format for use with remote commands.

k5 enablement and use requires that IBM DCE for AIX v2.2.0.1 or later be installed and configured on AIX v4.3.1 or later. (Refer to the PSSP V3.2 release documentation for the required levels of AIX and DCE with respect to PSSP V3.2 security.) Note that at the time this book was created, k5 support on AIX is only valid with IBM DCE for AIX. MIT’s Kerberos V5 package cannot be used in place of IBM DCE for AIX.

The $HOME/.k5login file is used for authorization when k5 is enabled.

All DCE credential caches (also called ticket caches) reside in the directory /opt/dcelocal/var/security/creds/. This location cannot be changed by the user at the time of DCE login.

Each /opt/dcelocal/var/security/creds/dcecred_XXXXXXXX file stores the tickets issued to a particular DCE principal (XXXXXXXX is an eight-digit hexadecimal number). A dcecred_XXXXXXXX file is created by invoking dce_login (or dce_login_noexec), the PSSP command, dsrvtgt, or through an AIX/DCE integrated login to AIX. Only the user who invoked dce_login has read/write permission to the file. (For complete information on the dcecred_XXXXXXXX files, refer to the dcecred_* Files section of the DCE Administration Commands Reference.)

dce_login and AIX/DCE integrated login automatically set the environment variable, KRB5CCNAME, which stands for Kerberos Version 5 credentials cache name, to a location of the format:

FILE:/opt/dcelocal/var/security/creds/dcecred_XXXXXXXX.

The /opt/dcelocal/var/security/creds/dcecred_ffffff ticket cache is special. It stores the tickets for the local machine’s principal, known in the DCE Registry as hosts/{dce_hostname}/self. This is commonly referred to as the self-host or machine principal identity.

The root user on a machine automatically has access to these credentials and, therefore, appears as the DCE entity self-host until reauthenticating as a different DCE principal. A root user who has not reauthenticated will share
the dcecred_ffffffff ticket cache with other root processes using the machine's self-host identity, such as dced and other root processes that do not explicitly issue a DCE login. Root users must be careful not to inadvertently destroy the self-host ticket cache (by issuing kdestroy -f or by deleting the actual credential files). When the self-host context is destroyed, local DCE services will be impacted significantly. The self-host credential context can be re-created by stopping and then starting the DCE services on the host.

Also, note that a root user does not have to set KRB5CCNAME to have access to the dcecred_ffffffff credentials. By virtue of being root, client commands that support DCE credentials know that, in the absence of a set KRB5CCNAME variable, you should check for root credentials at /opt/dcelocal/var/security/creds/dcecred_ffffffff.

8.11.3.2 k4
The k4 method uses the Kerberos V4 protocol, implemented through a Kerberos V4 compatibility library (supplied by and shipped only with PSSP), and Kerberos V4 credentials. The Kerberos V4 compatibility library is /usr/lib/libspk4rcmd.a.

The $HOME/.klogin file is used for authorization when k4 is enabled.

Kerberos V4 credential caches (also called ticket caches) are stored in the ticket file specified by the user's KRBTKFILE environment variable. If the KRBTKFILE variable is not set, the user's ticket is stored in the /tmp/tktUID file, where UID is the user identification number. Kerberos V4 credentials can be stored in any location for which the user has read/write access. A Kerberos V4 credential file is created by invoking the PSSP commands, k4init, rcmdtgt, or ksrvtgt, or through an AIX/PSSP Kerberos V4 integrated login. Only the user that invoked k4init, rcmdtgt, or ksrvtgt, or logged in via integrated Kerberos V4, has read/write permission to the file (root notwithstanding, of course). For complete information on PSSP Kerberos V4 credentials and cache locations, refer to the PSSP: Administration Guide, SA22-7348.

Note that both rcmdtgt and ksrvtgt require that KRBTKFILE be set prior to command invocation. Only a root user can successfully execute rcmdtgt and ksrvtgt. These commands aren't intended for end-users. They were specifically designed for use by PSSP processes running as background tasks that require Kerberos V4 credentials to be obtained during run time. As such, example usage of these commands is not included here. For more information on rcmdtgt and dsrvtgt, refer to the PSSP Command and Technical Reference, SA22-7351.
8.11.3.3 std
The std method uses IP-based authentication, which is implemented through the operating system. There are no credentials or tickets, such as those used with k5 and k4.

The $HOME/.rhosts file is used for authorization when std is enabled.

8.11.3.4 none
none means that all AIX remote commands are disabled and cannot be used. This, in turn, means that those PSSP remote commands that implement their functionality through the exploitation of AIX remote commands are, effectively, disabled.

8.11.4 AIX remote command message summary
The type and quantity of AIX authentication methods that are enabled impact the style and quantity of error and informational messages returned to stderr (standard error). The greater the number of authentication methods enabled, the larger the number of possible error messages returned to stderr.

Likewise, the greater the number of authentication methods enabled, the greater the possibility that errors will be returned for some protocols but that the remote command will ultimately be successful. This is due to the different types of protocol authentication and/or authorization possibilities.

In general:
- Messages that start off with “kerberos:” relate to Kerberos V5 in some way (which may or may not relate to a DCE problem).
- Messages that start off with "spk4" relate to the Kerberos V4 compatibility library (installed by PSSP).
- Messages that start off with just remote command descriptors, as in krshd:, telnetd:, rshd:, rcp:, or start off with a hostname followed by some type of text, such as \{hostname_here\}: A remote host refused an
attempted connect operation, relate to a base AIX remote command service or some facet supplied by a base AIX remote command service.

- The rcp return code will be 0 (zero) if the command worked and a non-zero value if it did not. This can be used to tell if the copy worked or not.
- The rsh return code will be 0 (zero) if the rsh itself did not have a problem and a non-zero value if it did experience a problem. Whether the command called by the rsh is or is not successful cannot be determined by the rsh return code. The only way to determine if the called command failed is to scan the output of stderr.

Caveats:
- Not all AIX remote command messages have associated 7 digit message codes.
- All spk4* messages have associated seven digit message codes.
- kerberos: (k5 related) messages can appear after leading krshd:, telnetd:, and other messages.
- Messages can relate to authentication or authorization conditions, where the former relates to k5, k4, and std, while the latter relates to the authorization files, .k5login, .klogin, and .rhosts.
- Given that remote commands are designed to failover from one authentication method to the next, if more than one method is enabled, it is possible to receive messages from k5, k4, and std processing for a single remote command attempt. The order in which the messages are returned relate to the order in which the authentication methods are set (on the client’s host).

8.11.5 AIX and PSSP remote commands

The following are AIX remote commands:
- rsh
- rcp
- telnet
- ftp
- rlogin

The following are PSSP remote commands (Items in parentheses denote other command dependencies.):
- dsh (rsh)
8.11.6 Remote command client and server must share a protocol

The AIX authentication methods for remote commands (both AIX and PSSP remote commands) must share at least one method in common if the client and server are to communicate.

Remote command client requests will fail when the client and the server do not share at least one AIX authentication method in common. For example, if the client is running in a k5 only mode, then the server must be running in a mode that includes k5 (as in k5 (only), k5:k4, or k5:k4:std). An example of a disjoint set of authentication methods between a client and a server is when the client is running in a k4-only mode and the server is running in a k5-only mode.

A disjoint set of AIX authentication methods occurs under the following conditions:

- In a multi-partition environment where each partition is running a different security mode
- In a single partition environment where the AIX authentication methods on a host (node or the control workstation) are changed to something other than the system-wide security settings

There are two ways to query the authentication methods in the partition to determine if methods are disjoint:

The first is to use the `lsauthpar -v` command to compare the current partition's SDR values against each of the nodes in the partition. (In a one partition system, this is equivalent to testing the entire system.) `lsauthpar` automatically reads from the SDR the authentication methods for the partition.
and compares a node’s values to the SDR values. (When the partition name is not specified by the issuer, the current partition is assumed.) For each host with a disjoint set of methods a discrepancy message is displayed.

<table>
<thead>
<tr>
<th>sp3en0/ # lsauthpar -v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote command authentication methods for the partition: k4:std</td>
</tr>
<tr>
<td>No discrepancies were found.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sp3en0/ # lsauthpar -v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote command authentication methods for the partition: k5:k4:std</td>
</tr>
<tr>
<td>krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.</td>
</tr>
<tr>
<td>lsauthpar: 0016-314 On sp3en15 the remote command authentication methods are incorrectly set to &quot;k4:std&quot;.</td>
</tr>
</tbody>
</table>

The second way is to use the dsh -a command to query the local authentication methods on each host (in the current partition) and pipe the results to dshbak -c (dsh -avG will work over all nodes in the system, regardless of their partition association.) The dshbak program collapses identical output from more than one host so that identical results are displayed only once. Unlike the lsauthpar -v command, the administrator must know the authentication methods for the current partition and visually compare the node results to this value. Further, if environmental variables that control the set of nodes over which dsh will operate (WCOLL, SP_NAME) include nodes that span partitions, the results will indicate a discrepancy where one might not actually exist. Conversely, if the environment variables include a subset of nodes within a partition, the results will not indicate a true view of the entire partition.

**Using dsh to display the authentication methods on a set of nodes**

The dsh WCOLL variable is set and points to a file containing only a subset of nodes within the system. splstnodes reveals all the node names in the system.
Knowing exactly which nodes within the system, or within a partition, are disjoint (if any) with respect to AIX authentication methods is critical for normal security operations. Using the `lsauthpar -v` command is the easiest and most reliable way of gathering that information. Corrective actions can then be taken depending on the type of disjoint environment.

### 8.11.6.1 Multiple partitions with different security modes

There is no corrective action for this case, in the sense of correcting a problem, since it is an AIX requirement that for its client/server processes to communicate, they must share at least one AIX authentication method in common. (This is equivalent to humans speaking a common language in order to communicate verbally.) Multiple partitions do not have to be enabled for the same set of authentication methods.

The example set for a multi-partition environment assumes three partitions, sp3en0, sp3sp1, and sp3sp2, where each partition is enabled for a different set of AIX authentication methods.

In the first example, the commands are issued from the control workstation out to the nodes, and all attempts are successful. The control workstation contains the union of all the AIX authentication methods in the SP, given that the control workstation must be able to communicate with each node in the system. Note that the control workstation can contain more AIX authentication methods.
authentication methods than the union of the AIX authentication methods of
the SP. For example, if the union of all AIX authentication methods in the SP
is k5:k4, the control workstation can have k5:k4:std, with k5:k4 being the
minimum set of methods. This allows the SP administrator to tailor which
remote command requests are satisfied by the control workstation for
external-to-the-SP host requests in environments where such remote access
to the control workstation is required.

In the second set of examples, the commands are issued from a node in one
partition to nodes in other partitions. Some of the requests are successful
because a common AIX authentication method is shared, while other
requests are not successful because a method is not shared.

For the scope of these examples, the following partition security methods
(including AIX authentication methods), along with a node name in each
partition, were in place. The data was extracted from the splstdata -p
command.

```
Syspar: sp3en0
-------------------------
ts_auth_methods dce:compat
auth_methods k5:k4
node sp3n01

Syspar: sp3sp1
-------------------------
ts_auth_methods compat
auth_methods k4
node sp3n04

Syspar: sp3sp2
-------------------------
ts_auth_methods dce
auth_methods k5
node sp3n03
```

Control workstation to nodes example - k5:k4 to a k4 node, a k5:k4
node, and a k5 node

The rsh client request from the k5:k4 control workstation to nodes at various
authentication method settings is successful because the client shares at
least one method in common with each host. Error messages returned by the
remote daemon are for failed remote command protocol attempts and are not
failures of the commands actually issued on the remote hosts.

Security configuration and user's credential identities
rsh to a k5:k4-enabled node:

```
sp3en0/ # lsauthent ; klist | grep lob ; k4list | grep pal:
Kerberos 5
Kerberos 4
Global Principal: /.../sp_cell/hosts/sp3en0/self
Principal: root.admin@ITSO.IBM.COM
```

The protocol error message going from k5:k4 to k4 is expected given that the client tried to communicate with the remote server via k5 first. Once that attempt failed, the client and server communicated via k4.

**Node-to-node example #1: k5:k4 to a k5 node and to a k4 node**

The rsh client request from a k5:k4 node to a k4 node is successful, as is the k5:k4 to k5 node request. Error messages returned by the remote daemon are for failed remote command protocol attempts only. The commands to be run on the remote host were, ultimately, executed because a common authentication method between client and server was found.
Chapter 8. Problem isolation and resolution

The protocol error message going from k5:k4 to k4 is expected since the client tried to communicate with the remote server via k5 first. Once the k5 attempt failed, the client and server communicated via k4.

**Node-to-node example #2: k4 to a k5 node and to a k5:k4 node**

The rsh client request from a k4 node to a k4 node is successful while the k4 to k5 node request is denied. For failed rsh attempts, error messages returned by the remote daemon are for failed remote command protocol attempts. The commands to be run on the remote host were never executed, due to the disjoint authentication methods between client and server.

```
sp3n01/ # lsauthent ; k4list | grep pal:
Kerberos 5
Kerberos 4
Global Principal: /.../sp_cell/hosts/sp3n01/self
Principal: root.admin@ITSO.IBM.COM
sp3n01/ # rsh sp3n04 "hostname -s ; lsauthent"
krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3n04
Kerberos 4
sp3n01/ # rsh sp3n03 "hostname -s ; lsauthent"
sp3n03
Kerberos 5
```

The protocol error message going from k4 to k5 is expected since the client and server do not share a common method. The commands to be run on the remote host are not attempted.

The protocol error message going from k4 to k5:k4 is expected since the client only tried to communicate with the remote server via k4.

**Node-to-node example #3. k5 to a k5:k4 node and to a k4 node.**

The rsh client request from a k5 node to a k5:k4 node is successful, while the k5 to k4 node request is denied. For failed rsh attempts, error messages

```
sp3n04/ # lsauthent ; k4list | grep pal:
Kerberos 4
Principal: root.admin@ITSO.IBM.COM
sp3n04/ # rsh sp3n03 "hostname -s ; lsauthent"
krshd: Kerberos Authentication Failed.
sp3n03
Kerberos 5
```

The protocol error message going from k4 to k5 is expected since the client and server do not share a common method. The commands to be run on the remote host are not attempted.

The protocol error message going from k4 to k5:k4 is expected since the client only tried to communicate with the remote server via k4.
returned by the remote daemon are for failed remote command protocol attempts. The commands to be run on the remote host were never executed due to the disjoint authentication methods between client and server.

```
sp3n03/ # lsauthent ; klist | grep lob
Kerberos 5
    Global Principal: /.../sp_cell/hosts/sp3n01/self
sp3n03/ # rsh sp3n01 "hostname -s ; lsauthent"
sp3n01
Kerberos 5
Kerberos 4
sp3n03/ # rsh sp3n04 "hostname -s ; lsauthent"
krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
```

The protocol error message going from k5 to k4 is expected since the client and server do not share a common method. The commands to be run on the remote host were never executed due to the disjoint authentication methods between client and server.

To establish communication between the client and server, the security settings of one (or both) partitions must be changed to include a common method. This is accomplished through PSSP administrative commands that either enable or disable AIX authentication methods.

Removing an authentication method from a partition. (SDR entry and nodes):

```
sp3en0/ # lsauthpar -p sp3en0
Kerberos 5
Kerberos 4
sp3en0/ # chauthpar -p sp3en0 k5
sp3en0/ # lsauthpar -p sp3en0
Kerberos 5
```

Adding an authentication method to a partition. (SDR entry and nodes):

```
sp3en0/ # lsauthpar -p sp3en0
Kerberos 5
Kerberos 4
sp3en0/ # chauthpar -p sp3en0 k5
```
Attempt to add an authentication method to a partition, but the required software to support the method is not installed:

```
sp3en0/ # lsauthpar -p sp3sp1
Kerberos 4
sp3en0/ # chauthpar -p sp3sp1 k5 k4
sp3n04: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3n06: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3en0/ # lsauthpar -p sp3sp1
Kerberos 4
sp3en0/ # chauthpar -p sp3sp1 k5 k4
```

Enabling authentication methods requires that the required security software is already installed in the target partition. Note, however, that there are no such restrictions when disabling authentication methods that are already set.

In an environment where the software is not already installed for the partition, the attempt to set PSSP authentication methods for the partition will fail as shown. In the event of such a failure, refer to the *PSSP: Administration Guide*, SA22-7348, for the steps needed to install the required security code on the nodes, then rerun the command.

Likewise, in a coexistence environment where SP nodes include Version 3.2 and pre-3.2 PSSP levels, attempting to disable an AIX authentication method that is required to be compatible with previous levels of PSSP is not permitted as shown in the following example:
8.11.6.2 Methods do not match system or partition level values

In the case of disjoint AIX authentication methods on one host only, the corrective action is for the SP administrator to bring that host's AIX authentication methods in line with the partition settings. This is accomplished through PSSP or AIX administrative commands in one of three ways:

**Issue** `chauthpar -f {methods list}` from the control workstation:

```
sp3en0/ # lsauthpar
Kerberos 5
Kerberos 4
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 4
sp3en0/ # chauthpar -f k5 k4
sp3n05: krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 5
Kerberos 4
```

or `rsh` the `chauthent {methods list}` command to the host:

```
sp3en0/ # lsauthpar
Kerberos 5
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 5
Kerberos 4
sp3en0/ # rsh sp3n05 "chauthent -k5"
sp3en0/ # rsh sp3n05 lsauthent
Kerberos 5
```

or log in to the node, and issue the `chauthent {methods list}` command locally.

Except for logging in to the node, the PSSP commands needed to set AIX authentication methods locally are shown in the `rsh` portion of the previous example.

The first approach is the PSSP preferred synchronization method. The other approaches should be used under conditions where `chauthpar` fails.
However, should the chauthpar fail, the chances that the rsh approach will work are low, given that chauthpar exploits the PSSP dsh command, which, in turn, exploits rsh. If chauthpar fails due to an rsh failure, the rsh alone will most likely fail. (Issuing an individual rsh command to each host over which chauthpar will operate is an excellent way to help identify any problems with the state of AIX authentication method settings within the system.)

Given that chauthpar ultimately issues a chauthent on hosts for which the underlying rsh was successful, chauthpar and chauthent will experience a failure (will not set methods as requested) under the following conditions:

- /tmp is full (chauthent comments/uncomments rshd and rlogind entries in /etc/inetd.conf, based on the inclusion or absence of -std in chauthent, and uses /tmp to hold a copy of the file it has modified). The error message received is:

```
sp3en0/ # chauthent -k4 -std
chsubserver: error in updating /etc/inetd.conf
chsubserver: error in updating /etc/inetd.conf
```

- The required software for a method is not installed on the host:

```
sp3n09/ # chauthent -k5 -k4 -std
Kerberos 4 permitted on SP node only.
Kerberos 5 requires DCE version 2.2 or greater.
```

- The order of the methods is incorrect, or the methods are invalid:

```
sp3en0/ # chauthent -std -k5 -k4
Invalid authentication method or ordering of methods
```

When neither chauthpar nor rsh/chauthent are successful, logging in to the node and changing the methods locally is the only course of synchronization action, short of rebooting the node, and is less disruptive than rebooting a node. (Node login may take the form of telnet, rlogin, s1term (in write mode), ssh, or other processes that open a terminal session to a node.)

### 8.11.7 AIX and PSSP remote command debug tips/checklist

There are several general items that can be checked when debugging a failed remote command event. The following list includes items that may relate only to one protocol. For example, checking to see if DCE is running on the client and target hosts makes sense only for k5 processing.
While the majority of items focus on rsh and rcp specifically and, by extension, to dsh and pcp, the same techniques can be applied to other AIX remote commands and PSSP remote parallel commands:

- Ensure that the AIX authentication methods are set properly between the client and target hosts.
- Ensure that the authorization files are populated properly on the target host.
- Ensure that the user has valid credentials. Expired, destroyed, or non-existent credentials result in errors. The commands, `klist` (for DCE) and `k4list` (for Kerberos V4), should be used to determine that state of currently-held credentials.
- If AIX remote command daemons are started through inetd, check that inetd is running on the target host. Also, check the inetd configuration for statements relating to the remote command daemons to ensure that inetd has been configured to start the daemons. Further, use `lssrc -t {subserver name}` to determine if the remote command daemon is currently active on the remote host.
- If AIX remote command daemons are not usually started through inetd, check that they are running/active on the target host.
- Ensure that the DCE daemons on the target host are configured properly.
- Ensure that the DCE master servers are running and reachable from the client and target hosts.
- Ensure that the Kerberos V4 daemons are running and reachable from the client and target hosts.
- For k5 processing, ensure that `/etc/krb5.conf` contains one or more valid security server entries for the current cell.
- For k5 processing, ensure that the remote host's self-host principal key file contains entries for all adapters on the host. The key file is `/krb5/v5srvtab`.
- For k4 processing, ensure that `/etc/krb.conf` contains one or more valid Kerberos V4 server entries for the current realm.
- Ensure that the time skew between the client and target hosts is not greater than five minutes.
- Check the return code of the `rcp` or `rsh` command (via `echo $?`).
- Check environment variables that impact the behavior of PSSP remote commands and AIX remote commands. (When set incorrectly, the remote commands will fail.) In the absence of explicitly-set variables, ensure that default values are being retrieved properly from the environment. This
includes the variables, KRBTKFILE, KRB5CCNAME, SP_NAME, WCOLL, and K5MUTE.

- Ensure that entries in the authorization files do not end with blank spaces. That is, for any given entry in an authorization file (where one line equals an entry), that entry cannot have spaces after the final alphanumeric character. Functions that parse the authorization files attempt to match the contents of the file to a user's AIX or credentials ID. Blanks at the end of a line will not match the user data and will result in an authorization failure. The same is true when the line begins with leading blanks or spaces.

- A pound sign ("#") in the first column of an authorization file is treated as a comment.

- Check that the following files contain correct data: /etc/hosts, /etc/hosts.equiv, /etc/resolv.conf.

8.11.8 The .k5login file demystified

Consult the "Using a .k5login File" subsection of the “Customizing TCP/IP Features” chapter of the AIX User's Guide: Communications and Networks for a general explanation of the format of the entries that appear in a .k5login.

.k5login entries contain DCE principal/cell entries in Kerberos V5 format. The entries are not kept in DCE principal name format. Specifically, the entries are of the form, principal_name@cell_name, where principal_name is a DCE principal name and may include valid DCE principal characters, such as the slash (/), and where cell_name is the name of the cell to which the principal belongs.

For example, here are two DCE principal names in the ITSO DCE cell:

```
sp3en0/ # klist | grep lob
Global Principal: /.../sp_cell/hosts/sp3en0/self
...
sp3en0/ # klist | grep lob
Global Principal: /.../sp_cell/dero
```

klist displays a DCE principal name in its global form, which means fully-qualified DCE names. Short-names and cell-relative representations are not shown.

Both principals in the above example are in the cell sp_cell. The name of the first principal is hosts/sp3en0/self, and the second principal is dero.
Their Kerberos V5 equivalent forms of the DCE principals are hosts/sp3en0/self@sp_cell and dero@sp_cell, respectively.

To generate a Kerberos V5 principal/cell form from a DCE Global Principal name, the following approach is valid. Everything after /.../{cell_name} is the principal_name. Everything between /.../ and the next / (slash) is the cell_name.

Here are some translation examples. The Global Principal names were taken from the results of the dcep -c principal catalog command:

<table>
<thead>
<tr>
<th>Global Principal</th>
<th>Kerberos V5 Principal/Cell Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>/.../cell/ssp/sp3n16.msc.itso.ibm.com</td>
<td>hosts/sp3n16.msc.itso.ibm.com/self@sp_cell</td>
</tr>
<tr>
<td>/.../cell/ftp/sp3sw06.msc.itso.ibm.com</td>
<td>ftp/sp3sw06.msc.itso.ibm.com@sp_cell</td>
</tr>
<tr>
<td>/.../cell/itso1</td>
<td>itso1@sp_cell</td>
</tr>
<tr>
<td>/.../cell/host/sp3n13.msc.itso.ibm.com</td>
<td>host/sp3n13.msc.itso.ibm.com@sp_cell</td>
</tr>
</tbody>
</table>

The DCE klist command also displays the Kerberos V5 form of the current DCE identity. It is the Default principal: line in the output.

```
sp3en0/ # klist
DCE Identity Information:
...  Global Principal: /.../cell/dero
...  Ticket cache: /opt/dcelocal/var/security/creds/dcecred_64d85300
    Default principal: dero@sp_cell
    Server: krbtgt/sp_cell
...  
```

Keep in mind that the .k5login file should only contain DCE principal/cell entries that have the ability to issue remote commands and for which remote command authorization should be granted. .k5login is, after all, an access control list (ACL). Placing an entry in this file grants that particular identity access to your account.

The PSSP installation and configuration and subsequent PSSP Systems Management routines give the SP administrator the ability to generate the .k5login file for root. When the administrator elects to have PSSP services generate root's .k5login file, it will populate the file with all the self-host principal entries found in the DCE Registry that also correspond to nodes in the SP where DCE is installed and configured, plus an entry for each node instance of the PSSP DCE principal spbgroot. PSSP doesn't just add all self-host entries found in the DCE Registry, since there are likely to be
self-host entries for non-SP hosts in the same DCE cell. Furthermore, if a .k5login file exists for root at the time PSSP routines are requested to generate a root level .k5login file, PSSP preserves existing entries in the file and only adds SP-related entries that do not exist in the file.

Example of a PSSP-generated .k5login for root on the control workstation:

```
sp3en0/ # cat .k5login
hosts/sp3en0/self@sp_cell
ssp/sp3en0/spbgroot@sp_cell
hosts/sp3en0/self@sp_cell
ssp/sp3en0/spbgroot@sp_cell
hosts/sp3en0/self@sp_cell
ssp/sp3en0/spbgroot@sp_cell
hosts/sp3n01/self@sp_cell
ssp/sp3n01/spbgroot@sp_cell
hosts/sp3n01/self@sp_cell
ssp/sp3n01/spbgroot@sp_cell
hosts/sp3n05/self@sp_cell
ssp/sp3n05/spbgroot@sp_cell
hosts/sp3n06/self@sp_cell
ssp/sp3n06/spbgroot@sp_cell
...
```

For sites that support intercell communication (two or more DCE cells that are registered with one another) and want to exploit k5 remote command support between their DCE cells, administrators must manually add intercell principals to root's .k5login where desired. Given that all nodes in an SP that are configured for PSSP DCE Trusted Services must be configured into the same DCE cell, including the control workstation's DCE configuration, PSSP code has no way of determining which principals from foreign DCE cells are valid for inclusion in root's .k5login file. (Doing anything else could introduce security exposures.)

8.11.9 Zero length authorization files

A 0 (zero) length AIX remote command authorization file has the net effect of denying access to any user, even if the user's AIX ID is the same on both the source and target hosts, and/or the user's Kerberos V4 or DCE principal credential identities are the same as the AIX user ID. Note that telnet and rlogin requests under this condition prompt for the user ID and password. Keep in mind that AIX remote command authorization files are access control lists (ACLs); so, it makes perfect sense to refuse remote command requests when the authorization files are 0 length or to prompt for a user ID and password even when the AIX ID is the same on both the source and target hosts. (For details on how AIX grants remote command access for the k5 and std protocols, refer to the following AIX documentation: AIX Commands Reference V4.3, SBOF-1877, for the rsh command and the krshd and rshd daemons; the AIX V4.3 Technical Reference, SBOF-1878, for the kvalid_user Subroutine.)
The following examples demonstrate how 0 length authorization files impact rsh processing. In all cases, both the source (client) host and target host have the following AIX authentication methods enabled: k5, k4, and std. (The same type of results are produced when only one or two methods are enabled.) The non-root AIX user, dero, exists on both the source and target hosts, and even though a non-root user was chosen for the examples, the same rules apply to a root user. However, since root has access to the special DCE self-host credentials, a non-existent or 0 length .k5login file will generate an authorization failure instead of a kerberos:...No credentials cache file found error message.

The first four examples reveal the behavior of rsh when the authorization files do not exist. This provides a baseline for the examples where the authorization files are 0 length:

```
sp3n06/u/dero/ # ls .k5login ; ls .klogin ; ls .rhosts
ls: 0653-341 The file .k5login does not exist.
ls: 0653-341 The file .klogin does not exist.
ls: 0653-341 The file .rhosts does not exist.
```

dero doesn't have DCE or Kerberos V4 credentials:

```
sp3n06/u/dero/ # klist ; k4list
Kerberos Ticket Information:
klist: No credentials cache file found (dce / krb) (ticket cache /tmp/krb5cc_36586)
Ticket file: /tmp/tkt36586
k4list: 2504-076 Kerberos V4 ticket file was not found
```

```
sp3n06/u/dero/ # rsh sp3n08 date
kerberos: Couldn't get credentials for the server: No credentials cache file found.
spkrsh: 0041-003 No tickets file found. You need to run "k4init".
rshd: 0826-813 Permission is denied.
```

The first two error messages are expected, given that the user lacks DCE and Kerberos V4 credentials. These errors have nothing to do with the state of the .k5login and .klogin files.

The rshd failure message is due to the absence of a .rhosts file.

dero has valid DCE credentials, but no Kerberos V4 credentials:

```
sp3n06/u/dero/ # klist ; k4list
Kerberos Ticket Information:
klist: No credentials cache file found (dce / sec) (ticket cache /tmp/krb5cc_36586)
Ticket file: /tmp/tkt36586
k4list: 2504-076 Kerberos V4 ticket file was not found
```

```
sp3n06/u/dero/ # rsh sp3n08 date
kerberos: Couldn't get credentials for the server: No credentials cache file found.
spkrsh: 0041-003 No tickets file found. You need to run "k4init".
rshd: 0826-813 Permission is denied.
```
The dero1@sp_cell authorization failure is expected. The user has valid DCE credentials, but when krshd attempts to authorize the user, the authorization fails due to the absence of the authorization file, and the fact that the DCE principal name (ID) doesn't match the AIX user name (ID).

The second failure is due to the user's lack of Kerberos V4 credentials. This error has nothing to do with the state of the .klogin file.

The rshd failure message is due to the absence of a .rhosts file.

dero has valid DCE and Kerberos V4 credentials:

The dero1@sp_cell authorization failure is expected. The user has valid DCE credentials, but when krshd attempts to authorize the user, the authorization fails due to the absence of the authorization file, and the fact that the DCE principal name (ID) doesn't match the AIX user name (ID). The situation is the same for the k4 attempt.

The rshd failure message is due to the absence of a .rhosts file.

dero has valid DCE and Kerberos V4 credentials, and the name of the principal in the credentials matches that of the AIX user ID (dero):
The first rsh is successful because the DCE principal name matches the name of the AIX ID. After the DCE credentials are destroyed (kdestroy), the second rsh attempt is successful because the Kerberos V4 principal name matches the name of the AIX ID.

The next four examples reveal the behavior of rsh when the authorization files are 0 length.

1. dero doesn't have DCE or Kerberos V4 credentials:

```
sp3n06/u/dero/ # klist | grep lob ; k4list | grep pal:
    Global Principal: /.../sp_cell/dero
Principle: dero@ITSO.IBM.COM
sp3n06/u/dero/ # rsh sp3n08 date
Thu Feb 22 15:36:41 EST 2000
sp3n06/u/dero/ # kdestroy
sp3n06/u/dero/ # klist | grep lob
    klist: No credentials cache file found (dce / krb) (ticket cache /opt/dcelocal/var/security/creds/dcecred_b12cbb00)
    sp3n06/u/dero/ # rsh sp3n08 date
    kerberos: Couldn't get credentials for the server: No credentials cache file found.
    Thu Feb 22 15:34:31 EST 2000
```

```
sp3n06/u/dero/ # touch .k5login ; touch .klogin ; touch .rhosts
sp3n06/u/dero/ # ls -l .k5login ; ls -l .klogin ; ls -l .rhosts
    -rw-r--r-- 1 dero usr 0 Feb 15 2000 .k5login
    -rw-r--r-- 1 dero usr 0 Feb 15 2000 .klogin
    -rw-r--r-- 1 dero usr 0 Feb 15 2000 .rhosts
```

The first two error messages are expected, given that the user lacks DCE and Kerberos V4 credentials. These errors have nothing to do with the state of the .k5login and .klogin files.

The rshd error is an authorization failure from the std protocol, because the .rhosts file is 0 (zero) length.

2. dero has valid DCE credentials, but no Kerberos V4 credentials:

```
sp3n06/u/dero/ # klist | grep lob ; k4list | grep pal:
    klist: No credentials cache file found (dce / krb) (ticket cache /tmp/krb5cc_36586)
    k4list: 2504-076 Kerberos V4 ticket file was not found
sp3n06/u/dero/ # rsh sp3n08 date
    kerberos: Couldn't get credentials for the server: No credentials cache file found.
    spk4rsh: 0041-003 No tickets file found. You need to run "k4init".
    rshd: 0826-813 Permission is denied.
```

The first two error messages are expected, given that the user lacks DCE and Kerberos V4 credentials. These errors have nothing to do with the state of the .k5login and .klogin files.

The rshd error is an authorization failure from the std protocol, because the .rhosts file is 0 (zero) length.
Chapter 8. Problem isolation and resolution

The dero1@sp_cell authorization failure is expected. The user has valid DCE credentials, but the .k5login file is 0 (zero) length; so, when krshd attempts to authorize the user, the authorization fails due to the no match found condition.

The spk4rsh error is expected because the user lacks Kerberos V4 credentials. This error has nothing to do with the state of the .klogin file.

The rshd error is an authorization failure from the std protocol because the .rhosts file is 0 (zero) length.

3. dero has valid DCE and Kerberos V4 credentials:

```
sp3n06/u/dero/ # klist | grep lob ; k4list | grep pal:
    Global Principal: /.../sp_cell/dero
k4list: 2504-076 Kerberos V4 ticket file was not found
krshd: Kerberos Authentication Failed: User derol@sp_cell is not authorized to login to account dero.
spk4rsh: 0041-003 No tickets file found. You need to run "k4init".
rshd: 0826-813 Permission is denied.
```

The dero1@sp_cell authorization failure is expected. The user has valid DCE credentials, but the .k5login file is 0 (zero) length; so, when krshd attempts to authorize the user, the authorization fails due to the no match found condition. Likewise, for the k4 attempt.

The rshd error is an authorization failure from the std protocol because the .rhosts file is 0 (zero) length.

4. dero has valid DCE and Kerberos V4 credentials, and the name of the principal in the credentials matches that of the AIX user ID (dero):

```
sp3n06/u/dero/ # klist | grep lob ; k4list | grep pal:
    Global Principal: /.../sp_cell/dero
Principal: dero@ITSO.IBM.COM
sp3n06/u/dero/ # rsh sp3n08 date
krshd: Kerberos Authentication Failed: User derol@sp_cell is not authorized to login to account dero.
krshd: Kerberos Authentication Failed: User dero.@ITSO.IBM.COM is not authorized to login to account dero.
rshd: 0826-813 Permission is denied.
```

The dero1@sp_cell authorization failure is expected. The user has valid DCE credentials, but the .k5login file is 0 (zero) length; so, when krshd attempts to authorize the user, the authorization fails due to the no match found condition. Likewise, for the k4 attempt.
Even though the principal name in the credentials matches that of the AIX user ID, authorization is denied due to the 0 (zero) length .k5login and .klogin files.

The rshd error is an authorization failure from the std protocol because the .rhosts file is 0 (zero) length.

PSSP scripts that allow the SP administrator to create AIX remote command authorization files for root do not create 0 length files under normal conditions. Remote commands are an important part of PSSP, and having 0 length authorization files for root would seriously impact the administrator’s ability to manage and maintain an SP.

Once a host’s authorization files are set to 0 length the SP administrator can still telnet or rlogin to the host to reset the authorization files, provided the correct root user ID and password are supplied.

### 8.11.10 Using s1term to log in to a node

There are two states related to AIX remote command processing that will prevent the root user (and other users) from accessing a node through remote commands. One: There are no AIX authentication methods set on the node. Two: The user’s authorization files are all 0 (zero) length.

Depending on the type of remote command used (rsh, telnet, ftp, and so on), the error message relating to the aforementioned states is different.

No AIX authentication methods are set on the node.
The corrective action is as follows: As root, on the control workstation, open up an s1term session in write mode to the node. Log in to the node and issue chauthent with the needed AIX authentication methods.

However, in order to issue s1term successfully, the SP administrator must be conscious of the Trusted Services authentication methods settings of the control workstation.

In the first sequence, a root user running in a dce-only security mode attempts to use s1term in write mode but is denied due to a lack of DCE group membership in the PSSP DCE group, hm-control.

After the cell administrator adds the DCE principal to the needed group, the user must destroy and create new DCE credentials in order to reflect the new membership. In the case of a root user holding the DCE self-host principal credentials, that is, the DCE machine principal context/identity, DCE (or just its security client, sec_cl) must be stopped and then started on the host (node or control workstation) for the self-host principal’s credentials to reflect the new group membership. (User membership in DCE groups is reflected through the user’s DCE credentials and is considered part of the user’s ticket privilege attributes. The credentials, also called tickets, can only reflect group memberships that were in place at the time the user’s credentials were created. Any group associations established after user credentials were generated will not be reflected until new credentials are created. For more information on DCE credentials (tickets) and their privilege attributes, refer to Chapter 31. Creating and Maintaining Accounts, subsection “Displaying Privilege Attributes and Tickets” of the IBM DCE Version 3.1 for AIX and Solaris: Administration Guide -- Core Components.

Once this takes place, the user can then s1term out to a node in write mode and change the AIX authentication methods accordingly:
As a cell administrator, add the DCE principal dero to the hm-control group:

```bash
sp3en0/ # dcecp -c group list hm-control
/.../sp_cell/sp_admin
sp3en0/ # dcecp -c group add hm-control -member dero
```

As the DCE principal dero, the command still fails, even though dero was added to the hm-control group. The user must destroy the current credentials and create new ones:

```bash
sp3en0/ # whoami ; klist | grep lob
root
Global Principal: /.../sp_cell/dero
sp3en0/ # s1term -w 1 5
s1term: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # dcecp -c prin show dero | grep groups
(groups spsec-admin sdr-admin)
```

In the second sequence, a non-root user running in a dce:compat security mode attempts to use s1term (in write mode), but is denied because the user's DCE principal is not a member of the PSSP DCE P group, hm-control, nor is the user's Kerberos V4 principal in the hmacls file. After the SP administrator adds the user's Kerberos V4 principal to hmacls and issues the hmadm setacls command, the user can issue s1term successfully. Note that the non-root user's DCE principal membership has not changed.

```
sp3en0/ # s1term -w 1 5
s1term: 0026-614 You do not have authorization to access the Hardware Monitor.
sp3en0/ # kdestroy ; exit
sp3en0/ # dce_login dero
...
DCE LOGIN SUCCESSFUL
...
sp3en0/ # s1term -w 1 5
...
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login:
...
sp3n05/ # chauthent -k5 -k4 -std
sp3n05/ #
```

462 Exploiting RS/6000 SP Security: Keeping It Safe
root adds the dero-sec principal to the hmacls file (writable by root only) and then issues `hmadm setacls` to update the Hardmon daemon's internal ACL table:

```
sp3en0/ # grep dero-sec /spdata/sys1/spmon/hmacls
1 dero-sec s
sp3en0/ # ls -l /usr/lpp/ssp/bin/hmadm
-r-xr-xr-x 1 bin bin 31176 Dec 14 10:54 /usr/lpp/ssp/bin/hmadm
sp3en0/ # hmadm setacls
hmadm: 0026-641I The ADMIN command "setacls" was sent.
sp3en0/ # s1term -w 1 5
... AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login: ...
```

In the above example, the principal is added to hmacls with a permission of `s`, the ability to read and write to a serial port (s1term). (There are other hmacls permissions. Refer to the *PSSP: Administration Guide*, SA22-7348, for complete details.) Since hmacls is a file that only root can update, an SP administrator (or other root user) must make the needed changes.

Once hmacls is updated, the `hmadm setacls` command must be invoked so that the Hardmon daemon's internal ACL tables are updated to reflect the new hmacls entry. If this step is omitted, the daemon's tables will not be updated, and the user will not be able to issue s1term. However, to run hmadm with the setacls operand, the issuer of hmadm must have Kerberos V4 administrator credentials. (hmadm can be invoked by any user, but to run it with setacls requires a specific level of authorization - a Kerberos V4 administrative identity. The Kerberos V4 administrator ID is the ID defined as the primary authentication services administrator by the `setup_authent` command during compat installation and configuration. In this case, it was root.admin. Refer to the PSSP installation and administration guides for complete details.)
In this scenario, the administrative duties required both AIX root level authority and Kerberos V 4 administrative authority. SP administrators may be one in the same. The same concept applies to SP administrators and DCE controls. Manipulating DCE organization, group, and ACL memberships requires varying degrees of DCE cell administration authority, while doing other SP administrative tasks, such as issuing the AIX command, \texttt{startsrc}, requires root authority. The SP administrator can be a DCE administrator or have his/her DCE principal granted the necessary DCE administrative authority for PSSP DCE controls. Or, perhaps, the DCE cell administrator is an individual other than an SP administrator.

0 (zero) Length Remote Command Authorization Files:

```
sp3en0/ # sp3n05 date
krshd: Kerberos Authentication Failed: User hosts/sp3en0/self@sp_cell is not authorized to login to account root.
krshd: Kerberos Authentication Failed: User root.admin@ITSO.IBM.COM is not authorized to login to account root.
rshd: 0826-813 Permission is denied.

sp3en0/ # rlogin sp3n05
krlogind: Kerberos Authentication Failed: User hosts/sp3en0/self@sp_cell is not authorized to login to account root.
root's Password:
Connection closed.

sp3en0/ # telnet sp3n05
Trying...
Connected to sp3n05
Escape character is '^]'.
[ Kerberos V5 refuses authentication because hosts/sp3en0/self@sp_cell ]
...
telnet (sp3n05)
...
AIX Version 4
(C) Copyrights by IBM and by others 1982, 1996.
login:
```

Perform the following corrective action: Telnet into the node and restore the files to their required states. The PSSP command, \texttt{updauthfiles}, can be used to populate the authorization files with PSSP-required entries.

### 8.11.11 The AIX K5MUTE variable

AIX provides the K5MUTE environment variable to suppress remote command protocol error messages. (Messages written by rsh, rcp, telnet, and so on to stderr.) When K5MUTE is set to a value other than 1 (one) or when K5MUTE is not set, all protocol error messages encountered are displayed.
When K5MUTE is set to 1, protocol error messages are suppressed provided that at least one protocol attempt is successful. If all protocol attempts fail when K5MUTE is set to 1, the user receives all error messages. (This information is useful for troubleshooting.)

---

**Note**

By default, AIX does not set K5MUTE in a user's shell environment.

K5MUTE behaves the same way for both root and non-root users.

At least one protocol is successful with K5MUTE not set - K5MUTE set to a value other than 1:

```
sp3n06/u/dero/ # env K5MUTE
KSMUTE: A file or directory in the path name does not exist.
sp3n06/u/dero/ # rsh sp3n08 date

kerberos: Couldn't get credentials for the server: No credentials cache file found.
krshd: Kerberos Authentication Failed: User dero.@ITSO.IBM.COM is not authorized to login to account dero.
Tue Feb 22 15:55:18 EST 2000
sp3n06/u/dero/ # export K5MUTE=0 ; env K5MUTE
0
sp3n06/u/dero/ # rsh sp3n08 date

kerberos: Couldn't get credentials for the server: No credentials cache file found.
krshd: Kerberos Authentication Failed: User dero.@ITSO.IBM.COM is not authorized to login to account dero.
Tue Feb 22 15:55:44 EST 2000
```

K5MUTE set to a value of 1, and at least one protocol is successful:

```
sp3n06/u/dero/ # env K5MUTE
KSMUTE: A file or directory in the path name does not exist.
sp3n06/u/dero/ # rsh sp3n08 date

kerberos: Couldn't get credentials for the server: No credentials cache file found.
krshd: Kerberos Authentication Failed: User dero.@ITSO.IBM.COM is not authorized to login to account dero.
Tue Feb 22 15:55:18 EST 2000
sp3n06/u/dero/ # export K5MUTE=1 ; env K5MUTE
1
sp3n06/u/dero/ # rsh sp3n08 date
Tue Feb 22 15:57:14 EST 2000
```

Protocol attempts are not successful, and K5MUTE is set to a value of 0 or a value of 1.
8.11.11.1 Separating protocol messages using dshbak

The following is an example of the results of a single dsh execution in an SP environment running with multiple partitions at different AIX authentication method levels and various coexistence hosts running with previous levels of AIX and PSSP.

The results shown are normal. There are no problems with dsh, rsh, DCE, AIX, or PSSP. In fact, the reason all these messages exist is the fact that all the different aspects of AIX authentication methods are occurring at once.

The example actually could have appeared worse had dshbak -c not been used in conjunction with dsh. Due to the nature of piping, all the errors from dsh are directed to stderr (standard error) and appear ahead of and out of the stream of the dshbak-filtered results. In short, all the errors come first, then all the non-errors appear in their collected form at the end. Had the filter not been used, the non-error results would have been nestled in with all the error results.

Of course, the K5MUTE variable could have been set to suppress error messages when at least one protocol attempt to a host was successful. But keep in mind that when all protocol attempts to a host fail, all error messages are returned to the user. Using the dshbak -c filter is a good choice even when K5MUTE is set to suppress error messages.

The dsh was issued from the control workstation by an SP administrator to all the nodes in all partitions.
sp3en0/  # dsh -D -avG "chauthent -k5 -k4 -std ; lsauthent" | dshbak -c
sp3n11: Kerberos: Couldn't get credentials for the server: Server not found in Kerberos database.
sp3n11: Kerberos 4 permitted on SP node only.
sp3n11: Kerberos 5 requires DCE version 2.2 or greater.
sp3n12: Kerberos: Couldn't get credentials for the server: Server not found in Kerberos database.
sp3n12: rshd: 0826-813 Permission is denied.
sp3n15: kerberos: Couldn't get credentials for the server: Server not found in Kerberos database.
sp3n15: rshd: 0826-813 Permission is denied.
...
sp3n05: kerberos: Couldn’t get credentials for the server: Server not found in Kerberos database.
sp3n05: spk4rsh: 0041-004 Kerberos rcmd failed: 2504-032 Kerberos ticket expired.
sp3n05: ksh: chauthent: not found.
sp3n05: ksh: lsauthent: not found.
sp3n06: kerberos: Couldn’t get credentials for the server: Server not found in Kerberos database.
...
...
dsh: 5025-509 sp3n13 rsh had exit code 1
...
dsh: 5025-509 sp3n15 rsh had exit code 1
dsh: 5025-509 sp3n12 rsh had exit code 1
HOSTS -----------------------------------------------------------------------
sp3n11 sp3n12 sp3n15 sp3n09
---------------------------------------------------------------------------
Kerberos 5
Kerberos 4
Standard Aix
HOSTS -----------------------------------------------------------------------
sp3n01 sp3n06 sp3n11
---------------------------------------------------------------------------
Kerberos 4
Standard Aix
8.11.12 Additional DCE/k5 AIX remote errors

The remainder of this section contains various AIX rsh protocol error messages and recommended corrective actions related to DCE/k5. (Similar errors would be generated for the remaining remote commands.) Keep in mind that the corrective actions are general recovery recommendations. Your site's policy and procedures should be considered before applying any of the recommendations.
Remote command examples that appeared earlier in this section are not repeated here. Likewise, impacts to rsh commands related to credentials states, time skew between hosts, and file-system-full problems are presented in other sections of this chapter and not repeated here. (These sections also contain Kerberos V4 remote command errors.)

8.11.12.1 Server not found
The following screen shows the Server not found error message:

```bash
sp3en0/ # rsh sp3n08 date
kerberos: Couldn't get credentials for the server: Server not found in Kerberos database.
...
```

Table 19 lists the possible causes of the Server not found error message and the corrective actions to be taken.

Table 19. Causes and corrective actions for the “Server not found” message

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The target host does not appear in the DCE database. The client could not obtain a service ticket for the host.</td>
<td>Run the DCE administrative configuration for the host, followed by the DCE local configuration on the target host. Enable the AIX k5 method on the target host. See the DCE Administration Commands Reference for the config.dce command, the AIX Commands Reference for the chauthent command, and the PSSP: Command and Technical Reference, SA22-7351 for the chauthpar command.</td>
</tr>
</tbody>
</table>

8.11.12.2 Connection is ended by software
The following screen shows the Connection is ended by software error message:

```bash
sp3en0/ # rsh sp3n08 date
kerberos: Couldn't authenticate to the server: A connection is ended by software.
...
sp3en0/ # klist | grep sp3n08
sp3en0/ #
```

Server: host/sp3n08@sp_cell
Table 20 lists the possible causes of the *Connection is ended by software* message and the corrective actions to be taken.

<table>
<thead>
<tr>
<th>Possible cause</th>
<th>Corrective action</th>
</tr>
</thead>
<tbody>
<tr>
<td>The target host appears in the DCE database but does not have DCE configured locally or k5 enabled. The client obtained a service ticket for the host.</td>
<td>Run DCE local configuration on the target host. Enable the AIX k5 method on the target host. See the DCE Administration Commands Reference for the config.dce command, the AIX Commands Reference for the chauthent command, and the <em>PSSP: Command and Technical Reference</em>, SA22-7351, for the chauthpar command.</td>
</tr>
<tr>
<td>The target host’s DCE configuration is incomplete; specifically, the DCE Security client is only partially configured.</td>
<td>Correct the problem that prevented the DCE Security client from configuring and then complete the local portion of the DCE client configuration. See the <em>DCE Problem Determination Guide</em>.</td>
</tr>
<tr>
<td>DCE is installed, configured, and running on the target host, but its self-host key file, /krb5/v5srvtab, is missing.</td>
<td>Restore the DCE host’s /krb5/v5srvtab from the most recent backup. If the key file cannot be restored from a recent backup, the DCE client must be unconfigured (local and administrative), then reconfigured (local and administrative).</td>
</tr>
</tbody>
</table>

**Note**

If PSSP DCE services were installed and configured on the host then PSSP DCE services must be unconfigured and reconfigured for the host via rm_spsec. Refer to the *PSSP: Administration Guide*, SA22-7348, and the *PSSP: Command and Technical Reference*, SA22-7351, for details.

### 8.11.12.3 Bad sendauth

The following screen shows the *Bad sendauth* error message:

```bash
sp3en0/ # rsh sp3n08 date
kerberos: Couldn't authenticate to the server: Bad sendauth version was sent.
...```

Exploiting RS/6000 SP Security: Keeping It Safe
Table 21 lists the possible causes of the *Bad sendauth* error message and the corrective actions to be taken.

**Table 21. Causes and corrective actions for the “Bad sendauth” message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The target host appears in the DCE database, has DCE installed, but not configured, and the k5 method is enabled. (AIX allows the k5 method to be enabled, provided that the needed DCE file sets are installed on the host. AIX does not determine whether or not DCE is actually configured or running before allowing k5 to be enabled.)</td>
<td>Disable the k5 authentication method on the target host; run the DCE local configuration on the target host, and then enable the AIX k5 method. Or, disable the k5 method if DCE is not to be configured on the host. See the <em>DCE Administration Commands Reference</em> for the <em>config.dce</em> command, the <em>AIX Commands Reference</em> for the <em>chauthent</em> command, and the <em>PSSP: Command and Technical Reference</em>, SA22-7351, for the <em>chauthpar</em> command.</td>
</tr>
<tr>
<td>The target host’s /etc/krb5.conf file does not exist or is invalid.</td>
<td>As root on the target host, issue the DCE commands, <code>kerberos.dce -type local</code> and <code>dcecp -c secval krb5update -cellname {cellname}</code>. Refer to the <em>DCE Administration Commands Reference</em> for the <code>kerberos.dce</code> and <code>dcecp</code> <code>secval</code> commands.</td>
</tr>
</tbody>
</table>

**8.11.12.4 Server not configured for Kerberos 5**

The following screen shows a *Server not configured for Kerberos 5* error message:

```
sp3en0/ # rsh sp3n08 date
krshd: Kerberos 5 Authentication Failed: This server is not configured to support Kerberos 5.
...
```
Table 22 lists the possible causes of the Server not configured for Kerberos 5 error message and the corrective actions to be taken.

**Table 22. Causes and corrective actions for the “Server not configured for Kerberos 5” message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The target host appears in the DCE database and has DCE installed, configured, and running, but the k5 method is not enabled.</td>
<td>Enable the k5 authentication method on the target host. See the AIX Commands Reference for the chauthent command and the PSSP: Command and Technical Reference, SA22-7351, for the chauthpar command.</td>
</tr>
<tr>
<td>The target host does not have k5 enabled, and the DCE client services are not running.</td>
<td>Start DCE and then enable the k5 authentication method on the target host. See the DCE Administration Commands Reference, the AIX Commands Reference for the chauthent command, and the PSSP: Command and Technical Reference, SA22-7351, for the chauthpar command.</td>
</tr>
</tbody>
</table>

**8.11.12.5 Error code 60 - Key version**

The following screen shows the Error code 60 error message:

```bash
sp3en0/ # rsh sp3n08 date
kerberos: Couldn't authenticate to the server: Server rejected authentication (during sendauth exchange).
kerberos: Server returned error code 60 (Generic error (see e-text)).
kerberos: Error text sent from the server: Key version number for principal in key table is incorrect.
...
```

Table 23 lists possible causes of the Error code 60 error message and the corrective actions to be taken.

**Table 23. Causes and corrective actions for the “Error code 60” message**

<table>
<thead>
<tr>
<th>Possible cause</th>
<th>Corrective action</th>
</tr>
</thead>
<tbody>
<tr>
<td>The administrative DCE client configuration was reexecuted for an already configured DCE host.</td>
<td>The DCE client must be unconfigured locally only, then reconfigured locally. See the DCE Administration Commands Reference for the unconfig.dce command.</td>
</tr>
</tbody>
</table>
Chapter 8. Problem isolation and resolution

### 8.11.12.6 Error code 31 - integrity check failed

The following screen shows the *Error code 31 - integrity check failed* error message:

```
sp3en0/ # rsh sp3n08-tr2 date
kerberos: Couldn't authenticate to the server: Server rejected authentication (during sendauth exchange).
kerberos: Server returned error code 31 (Decrypt integrity check failed).
kerberos: Error text sent from the server Decrypt integrity check failed.
... 
```

Table 24 lists the possible causes of the *Error code 31 - integrity check failed* error message and the corrective actions to be taken.

**Table 24. Causes and corrective actions for the “Error code 31 - integrity check failed” message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A network interface adapter was added to a host after the host had been configured with DCE and enabled with k5, and neither the DCE administrative nor local configuration routines were run for the interface.</td>
<td>The DCE cell administrator must run the DCE <code>kerberos.dce -type admin</code> command for the interface; then, root on the target host must issue the DCE command, <code>kerberos.dce -type local</code>. This will create the needed entries in the DCE Registry and the target host's /krb5/v5srvtab, respectively. Refer to the DCE Administration Commands Reference for the <code>kerberos.dce</code> command.</td>
</tr>
</tbody>
</table>

### 8.11.12.7 Error code 60 - Key table

The following screen shows the *Server returned error code 60* error message:

```
sp3en0/ # rsh sp3n08-tr2 date
kerberos: Couldn't authenticate to the server: Server rejected authentication (during sendauth exchange).
kerberos: Server returned error code 60 (Generic error (see e-text)).
kerberos: Error text sent from the server Key table entry not found.
... 
```
Table 25 lists the possible causes of the *error code 60* error message and the corrective actions to be taken.

**Table 25. Causes and corrective actions for the "error code 60" message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A network interface adapter was added to a host after the host had been configured with DCE and enabled with k5, but only the administrative portion of the DCE configuration for the interface was completed. (The sp3n08-tr2 hostname appears in the DCE database, but doesn't exist in the target host's self-host (/krb5/v5srvtab) key file.)</td>
<td>As root on the target host issue the DCE command &quot;kerberos.dce -type local&quot;. This will create the needed entries in /krb5/v5srvtab. Refer to the DCE Administration Commands Reference for the kerberos.dce command.</td>
</tr>
</tbody>
</table>

**8.11.12.8 Cannot contact KDC**

The following screen shows a *Cannot contact any KDC* error message:

```
sp3en0/ # rsh sp3n08 date
kerberos: Couldn't get credentials for the server: Cannot contact any KDC for requested realm.
...
```

Table 26 lists the possible causes of the *Cannot contact any KDC* message and the corrective actions to be taken.

**Table 26. Causes and corrective actions for the "Cannot contact any KDC" message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The DCE /etc/krb5.conf file on the client's DCE host doesn't exist or doesn't contain a valid DCE Security server entry.</td>
<td>As root on the client host issue the DCE commands, kerberos.dce -type local and dcecp -c secval krb5update -cellname {cellname}. Refer to the DCE Administration Commands Reference for the kerberos.dce and dcecp secval commands.</td>
</tr>
</tbody>
</table>

The DCE Security server referenced in the client host's /etc/krb5.conf is not running.  
Start the needed DCE Security server.  
Refer to the DCE Administration Commands Reference for the start.dce command.

**8.11.12.9 Connect operation refused**

The following screen shows a *connect operation refused* error message:
Table 27 lists the possible causes of a connect operation refused message and the corrective actions to be taken.

Table 27. Causes and corrective actions for a “connect operation refused” message

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The target host’s AIX krshd daemon is not running.</td>
<td>Start the krshd daemon on the target host. The krshd daemon is usually registered with AIX's System Resource Controller (SRC) with a service name of kshell and controlled by the super daemon, inetd. Verify that the inet daemon is functioning properly and has a definition for kshell in its /etc/inetd.conf file. When krshd is controlled by inetd, krshd is considered a subserver. This distinction is important because a subserver requires the use of the -t flag in startsrc, stopsrc, and lssrc commands, versus the commonly used -s flag, which stands for subsystems. (As in stopsrc -s systld; startsrc -s systld; lssrc -s systld.) Refer to the AIX Commands Reference for details on the SRC commands, inetd, krshd, and /etc/inetd.conf.</td>
</tr>
<tr>
<td>The target host's AIX operating system or TCP/IP subsystems are experiencing problems.</td>
<td>Run your site’s operating system and TCP/IP subsystem diagnostic and recovery procedures on the target host.</td>
</tr>
<tr>
<td>There are no AIX authentication methods enabled on the remote host.</td>
<td>As root on the target host enable the correct set of AIX authentication methods. Refer to the AIX Commands Reference for the chauthent command and the PSSP Command and Technical Reference for the chauthpar command.</td>
</tr>
</tbody>
</table>

8.11.12.10 Matching credential not found

The following screen shows a Matching credential not found error message:
Possible cause(s) - The user issued a DCE `kinit` command instead of a `dce_login`, and provided the incorrect password for the DCE user ID, as shown in the following screen:

```
sp3en0/u/dero/ # rsh sp3n08 date
kerberos: Couldn't get credentials for the server: Matching credential not found.
```

Corrective action(s) - Issue the DCE `kdestroy` command, and then issue `dce_login` with the needed principal. Refer to the DCE Administration Commands Reference for the `kdestroy` and `dce_login` commands.

`kinit` results in an error when issued by a root user holding DCE self-host principal credentials:

```
sp3en0/ # whoami
root
sp3en0/ # env KRB5CCNAME
KRB5CCNAME: A file or directory in the path name does not exist.
sp3en0/ # klist | grep lab
Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # kinit fvt_admin
Can't change existing principal information with kinit
Current information is for /.../sp_cell/hosts/sp3en0/self
```

8.11.12.11 Error getting forwardable credentials

The following screen shows the Error getting forwardable credentials error message:
Table 28 lists the possible causes of the *Error getting forwardable credentials* message and the corrective actions to be taken.

**Table 28. Causes and corrective actions for an “Error getting forwardable credentials” message**

<table>
<thead>
<tr>
<th>Possible causes</th>
<th>Corrective actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user's DCE credentials are not marked forwardable. The -f or -F flag on AIX remote commands requests that the credentials be marked forwardable.</td>
<td>Issue the DCE \texttt{kinit -f} command to obtain DCE credentials that can be forwarded. Note that the DCE self-host principal's credentials are not permitted to be forwardable by design. Also, for a user to obtain credentials marked forwardable, the user's DCE account must have the forwardable attribute set to yes (the \texttt{forwardabletkt} attribute). If this attribute is set to no, the \texttt{kinit -f} (or a \texttt{dce_login -f}) will fail. The DCE cell administrator can change an account's attributes provided site policy permits the forwarding of DCE credentials. Refer to the \textit{DCE Administration Commands Reference} for the \texttt{kinit}, \texttt{dce_login}, and \texttt{account} commands and for details on the forwardable attribute of DCE accounts.</td>
</tr>
</tbody>
</table>

**Note**

A target host that has k5 enabled but does not have its DCE client services running will not cause a k5 protocol failure when it receives a k5 remote command request. DCE/k5 authentication services do not require that the DCE client services be running on the target host to authenticate the incoming client request. However, a DCE Security server must be reachable and running for the authentication routines to be successful.

### 8.12 How to clean up a failed DCE node installation or customization

This procedure shows you how to clean up a failed installation or customization of a node when the PSSP security state is set to dce.
You should first determine exactly where and why the node installation failed by examining the logs in /var/adm/SPlogs/sysman and the /var/adm/SPlogs/auth_install/log on the node. In the latter log, search for the string spauthconfig. After locating the string, the following questions need to be answered:

- Were the DCE file sets installed? If not, perhaps they were not found in the expected directory, /spdata/sys1/install/aix433/lppsource.

- Were the DCE clients successfully configured and started? The DCE command, show.cfg, shows the current state of the clients.

- If the clients have not been configured at all, the node can simply be reinstalled without doing any of the steps in Section 8.12.1 through Section 8.12.4. If a reinstall is not required, the security configuration can be completed by performing Step 11 in Section 8.12.4.

- If the DCE clients are configured and running, but not all PSSP services are running, check /spdata/sys1/keyfiles on the node. If this directory has no subdirectories or the subdirectories are empty, create_keyfiles has not been run successfully. Start the DCE removal process at Section 8.12.1, step 2. However, if the keyfiles exist, start at Step 1 of Section 8.12.1.

- If spauthconfig completed successfully without errors, you should consider fixing any subsequent install failures without reinstalling the node. For example, perhaps a customize would fix the problems.

When configuring nodes as DCE clients, PSSP takes advantage of the DCE split configuration option. With this option, the DCE client node is first configured on the DCE master server by the cell administrator (the admin-only part). Later, the configuration is completed on the node by root (the local-only part). You will follow this same model for unconfiguring and reconfiguring the node: unconfigure the node (local-only), unconfigure the node (admin-only), reconfigure the node (admin-only), reconfigure the node (local-only).

In this procedure, our DCE cell name is sp_cell, and the DCE administrative principal is cell_admin.

---

Note

This entire procedure must be followed whenever a node configured for DCE is reinstalled. In contrast, a node configured for DCE may be customized at any time, without following this procedure.
8.12.1 Unconfigure the node (local-only)

Perform the following steps to unconfigure the node (local-only):

As root on the node:

1. Unconfigure the PSSP additions to DCE for the node, such as the keytab objects and the keyfiles:

```
sp3n05/ # /usr/lpp/ssp/bin/rm_spsec -v -t local
...
```

2. Unconfigure the node in DCE:

```
sp3n05/ # unconfig.dce -config_type local all
...  
Unconfiguration completed successfully.
```

3. Get a listing of any additional network interfaces on the node, such as css0 and tr0. You will need this information later to remove the host and ftp principals for those interfaces from DCE.

```
sp3n05/ # netstat -l 2>&1 | tee /tmp/host_and_ftp_prins_to_remove
Name    Mtu     Network    Address     ...  
lo0     16896   link#1     ...  
lo0     16896   127        loopback     ...  
lo0     6896    ::1        ...  
en0     1500    link#2     10.0.5a.fa.13.af ...  
en0     1500    192.168.3   sp3n05      ...  
css0    65520   link#3     ...  
css0    65520   192.168.13 sp3sw05      ...
```

8.12.2 Unconfigure the node (admin-only)

As cell_admin on the control workstation, perform the following steps to unconfigure the node (admin-only):

1. Unconfigure the PSSP additions to DCE for the node, such as the SP Trusted Services principals, groups, organizations, and accounts:

```
sp3en0/ # dce login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep Global
    Global Principal: /.../sp_cell/cell_admin
sp3en0/ # rm_spsec -v -t admin sp3n05
...
```
The DCE hostname in `rm_spsec` must appear as it does in the DCE database.

2. Unconfigure the node in DCE:

   ```bash
   sp3en0/ # unconfig.dce -config_type admin all \n   > -dce_hostname sp3n05 -host_id sp3n05
   ... Unconfiguration completed successfully.
   ```

3. Remove from DCE the host and ftp principals for the additional network interfaces on the nodes:

   ```bash
   sp3en0/ # dcecp -c prin cat | grep -E "host|ftp" | grep 05
   /.../sp_cell/host/sp3sw05
   /.../sp_cell/ftp/sp3sw05
   sp3en0/ # dcecp -c prin delete { /.:host/sp3sw05 /.:ftp/sp3sw05 }
   sp3en0/ #
   ```

   **Note**
   They are listed in the `.tmp/host_and_ftp_prins_to_remove` file you created on the node.

8.12.3 Reconfigure the node (admin only)

As `cell_admin` on the control workstation, perform the following steps to reconfigure the node (admin-only):

1. Re-create the DCE hostname for the node and verify that they were added to the SDR:

   ```bash
   sp3en0/ # dcecp -c prin cat | grep -E "host|ftp" | grep 05
   /.../sp_cell/host/sp3sw05
   /.../sp_cell/ftp/sp3sw05
   sp3en0/ # dcecp -c prin delete { /.:host/sp3sw05 /.:ftp/sp3sw05 }
   sp3en0/ #
   ```
2. As root on the control workstation, reconfigure the node in DCE (DCE admin config only). You would be prompted for the DCE cell administrator’s password. In the following command output screen, the password prompt is not shown.

```
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/cell_admin
sp3en0/ # create_dcehostname
sp3en0/ # splistdata -n | pg
   List Node Configuration Information
   node# frame# slot# slots initial_hostname reliable_hostname dce_hostname default_route processor_type processors_installed description
   ----- ------ ----- ----- ----------------- ----------------- ----------------- -------------- -------------- -------------------- -------------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- -------------- --------------웅
```

3. As cell_admin on the control workstation, reconfigure the PSSP additions to DCE for the node, such as the SP Trusted Services principals, groups, organizations, and accounts:

```
sp3en0/ # exit
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # setupdce -v
   Configuration of DCE Host, sp3n05, will now begin.
   Configuring the Security client...
   Security client configuration is complete.
   Configuring the Directory client...
   Directory client configuration is complete.
   Configuration of DCE Host, sp3n05, was successful.
   Configuration completed successfully.
```

```
sp3en0/ # dce_login cell_admin
Enter Password:
DCE LOGIN SUCCESSFUL
sp3en0/ # klist | grep Global
   Global Principal: /.../sp_cell/cell_admin
sp3en0/ # splistdata -n | pg
```

4. As root on the control workstation, update the authorization files (.k5login, .klogin, .rhosts) for AIX remote commands on the control workstation:

```bash
sp3en0/ # exit
sp3en0/ # klist | grep Global
    Global Principal: /.../sp_cell/hosts/sp3en0/self
sp3en0/ # updauthfiles
sp3en0/ #
```

**Note**

It is not uncommon for the `updauthfiles` command to take several minutes to execute.

8.12.4 Reconfigure the node (local-only)

5. Reconfigure the node in DCE including the PSSP portion, which includes the keytab objects and the keyfiles:

```bash
sp3n05/ # /usr/lpp/ssp/bin/spauthconfig
sp3n05/ # tail -f /var/adm/SPlogs/auth_install/log
...
```
Appendix A. How Kerberos works

(Reprinted from the redbook The RS/6000 SP Inside Out, SG24-5374)

---

Kerberos

Also spelled Cerberus, the watchdog of Hades, whose duty was to guard the entrance (against whom or what does not clearly appear) ... It is known to have had three heads.

- Ambrose Bierce, The Enlarged Devil's Dictionary

Kerberos is a trusted third-party authentication system for use on physically insecure networks. It allows entities communicating over the network to prove their identity to each other while preventing eavesdropping or replay attacks. Figure 33 on page 485 shows the three parties involved in the authentication. The Kerberos system was designed and developed in the 1980’s by the Massachusetts Institute of Technology (MIT), as part of the Athena project. The current version of Kerberos is Version 5, which is standardized in RFC 1510, The Kerberos Network Authentication Service (V5).

![Figure 33. Partners in a Third-Party Authentication](image)

Kerberos provides two services to Kerberos principals (users or services): an Authentication Service (AS) and a Ticket-Granting Service (TGS). Principals can prove their identity to the AS by a single sign-on, and will get a
Ticket-Granting Ticket (TGT) back from the AS. When one authenticated principal (the client) wants to use services of a second authenticated principal (the server), it can get a Service Ticket for this service by presenting its TGT to the Kerberos TGS. The Service Ticket is then sent from the client to the server, which can use it to verify the client's identity.

This section describes the protocol that Kerberos uses to provide these services, independently of a specific implementation. A more detailed rationale for the Kerberos design can be found in the MIT article Designing an Authentication System: a Dialogue in Four Scenes available from the following URL: ftp://athena-dist.mit.edu/pub/kerberos/doc/dialogue.PS

A.1 Kerberos keys and initial setup

To encrypt the messages which are sent over the network, Kerberos uses a symmetric encryption method, normally the Data Encryption Standard (DES). This means that the same key is used to encrypt and decrypt a message, and consequently the two partners of a communication must share this key if they want to use encryption. The key is called secret key for obvious reasons: it must be kept secret by the two parties, otherwise the encryption will not be effective.

This approach must be distinguished from public key cryptography, which is an asymmetric encryption method. There, two keys are used: a public key and a private key. A message encrypted with one of the keys can only be decrypted by the other key, not by the one which encrypted it. The public keys do not need to be kept secret (hence the name “public”), and a private key is only known to its owner (it is not even to the communication partner as in the case of symmetric cryptography). This has the advantage that no key must be transferred between the partners prior to the first use of encrypted messages.

With symmetric encryption, principals need to provide a password to the Kerberos server before they can use the Kerberos services. The Kerberos server then encrypts it and stores the resulting key in its database. This key is the shared information that the Kerberos server and the principal can use to encrypt and decrypt messages they send each other. Initially, two principals who want to communicate with each other do not share a key, and so cannot encrypt their messages. But since the Kerberos server knows the keys of all the principals, it is called a trusted third party and can securely provide a common session key to the two parties.

Obviously, the initial passwords have to be entered securely, if possible at the console of the Kerberos server machine. They might also be generated by the
Kerberos server (especially if the principal is a host or service). In that case they must be securely transferred to the principal which stores (or remembers) them.

### A.2 Authenticating to the Kerberos server

If a principal (typically a user) wants to use Kerberos services, for example because it wants to use an application service which requires Kerberos authentication, it first has to prove its identity to the Kerberos server. This is done in the following way:

A command to sign on to the Kerberos system is issued on the application client, typically `kinit` or `k4init`. This command sends an authentication request to the Kerberos server, as shown in Figure 34 on page 487. This contains the type of service that is requested (here, the client wants to get service from the Ticket-Granting Service), the client's (principal's) name, and the IP address of the client machine. This request is sent in plain text. Note that the principal's password is not sent in this packet, so there is no security exposure in sending the request in plain text.

![Figure 34. Client's Authentication Request](image)

The request is processed by the Authentication Server. Using the client's name, it looks up the corresponding key in the Kerberos database. It also generates a random session key to be shared by the client and the TGS, which will be used to encrypt all future communication of the client with the TGS. With this information, the AS constructs the Ticket-Granting Ticket for the client, which (as all Kerberos tickets) contains six parts:

1. The service for which the ticket is good (here, the TGS)
2. The client's (principal's) name
3. The client machine's IP address
4. A timestamp showing when the ticket was issued
5. The ticket lifetime (maximum 21.25 hours in MIT K4, 30 days in PSSP K4, configurable in K5)

6. The session key for Client/TGS communications

This ticket is encrypted with the secret key of the TGS, so only the TGS can decrypt it. Since the client needs to know the session key, the AS sends back a reply which contains both the TGT and the session key, all of which is encrypted by the client’s secret key. This is shown in Figure 35 on page 488.

Now the sign-on command prompts the user for the password, and generates a DES key from the password using the same algorithm as the Kerberos server. It then attempts to decrypt the reply message with that key. If this succeeds, the password matched the one used to create the user’s key in the Kerberos database, and the user has authenticated herself. If the decryption failed, the sign-on is rejected and the reply message is useless. Assuming success, the client now has the encrypted Ticket-Granting Ticket and the session key for use with the TGS, and stores them both in a safe place. Note that the authentication has been done locally on the client machine, the password has not been transferred over the network.
A.3 Authenticating to an application server

If the client now wants to access an application service which requires Kerberos authentication, it must get a service ticket from the Ticket-Granting Service. The TGT obtained during the Kerberos sign-on can be used to authenticate the client to the TGS; there is no need to type in a password each time a service ticket is requested.

If the client sent only the (encrypted) TGT to the Kerberos TGS, this might be captured and replayed by an intruder who has impersonated the client machine. To protect the protocol against such attacks, the client also generates an authenticator which consists of three parts:

1. The client’s (principal’s) name
2. The client machine’s IP address
3. A timestamp showing when the authenticator was created

The authenticator is encrypted with the session key that the client shares with the TGS. The client then sends a request to the TGS consisting of the name of the service for which a ticket is requested, the encrypted TGT, and the encrypted authenticator. This is shown in Figure 36.

![Figure 36. Client’s Service Ticket Request](image-url)
The Ticket-Granting Server can decrypt the TGT since it is encrypted with its own secret key. In that ticket, it finds the session key to share with the client. It uses this session key to decrypt the authenticator, and can then compare the client's name and address in the TGT and the authenticator.

If the timestamp that the TGS finds in the authenticator differs from the current time by more than a prescribed difference (typically 5 minutes), a ticket replay attack is assumed and the request is discarded.

If all checks pass, the TGS generates a service ticket for the service indicated in the client's request. The structure of this service ticket is identical to the TGT described in A.2, “Authenticating to the Kerberos server” on page 487. The content differs in the service field (which now indicates the application service rather than the TGS), the timestamp, and the session key. The TGS generates a new, random key that the client and application service will share to encrypt their communications. One copy is put into the service ticket (for the server), and another copy is added to the reply package for the client since the client cannot decrypt the service ticket. The service ticket is encrypted with the secret key of the service, and the whole package is encrypted with the session key that the TGS and the client share. The resulting reply is shown in Figure 37. Compare this to Figure 35 on page 488.

![Figure 37. Ticket Granting Service’s Reply: Service Ticket](image-url)
The client can decrypt this message using the session key it shares with the TGS. It then stores the encrypted service ticket and the session key to share with the application server, normally in the same ticket cache where it already has stored the TGT and session key for the TGS.

To actually request the application service, the client sends a request to that server which consists of the name of the requested service, the encrypted service ticket, and a newly generated authenticator to protect this message against replay attacks. The authenticator is encrypted with the session key that the client and the service share. The resulting application service request is shown in Figure 38. Note the resemblance to the request for Ticket-Granting Service in Figure 36 on page 489.

![Figure 38. Client's Application Service Request](image)

The application server decrypts the service ticket with its secret key, uses the enclosed session key to decrypt the authenticator, and checks the user's identity and the authenticator's timestamp. Again, this processing is the same as for the TGS processing the service ticket request. If all checks pass, the server performs the requested service on behalf of the user.
If the client required mutual authentication (that is, the service has to prove its identity to the client), the server could send back a message which is encrypted by the session key it shares with the client, and an application-dependent contents that the client can verify. Since the service can only know the session key if it was able to decrypt the service ticket, it must have known its secret key and so has proven its identity.

Attention

**Authorization:** Kerberos is only responsible for authenticating the two partners. Any authorization mechanism must be enforced by the application itself.
Appendix B. Security command and tool reference

This appendix provides a quick reference for the security commands and tools discussed in this redbook.

**bffcreate**
Creates installation image files in backup format.

**cdsli -cworld**
Recursively lists the namespace of cells:

- `-c` specifies a clearinghouse
- `-w` specifies a long listing
- `-o` specifies an object
- `-r` indicates to list entries recursively
- `-l` specifies a link
- `-d` specifies a directory

**chauthpar [ -c | -f ] -p <syspar> [k5][k4][std]**
Enables the active remote command authentication methods for a system partition.

**chauthpts [ -c | -f ] -p <syspar> [dce][compat][ ]**
Enables the active trusted services authentication methods for a system partition.

**chauthent [-k5][-k4][-std][ ]**
Changes the configured authentication methods for the rcmds on the system. If none of the flags are set, the rcmds will be disabled from functioning.

**chauths [dce][compat][ ]**
Enables the active authentication methods for trusted services on a host.

**chpesite**
Updates the contents of the /opt/dcelocal/etc/security/pe_site file on a host.

**config_spsec [-c][-v]**
Configures SP Services into the DCE database. Services which use DCE as an authentication method are required to have certain information entered in the CDS clearinhouse and the Security registry to perform client/server authentication.

**create_dcehostname**
Populates the System Data Repository (SDR) with DCE hostnames for each node in a partition set to use DCE.
create_keyfiles [-c][-v]
   Creates DCE keytab objects and stores them into specified keyfiles on the local file system. Services which use DCE as an authentication method will use these key files to log into DCE.

dcecp -c  The DCE Control Program, which is used to perform the following DCE management tasks:

   • cell show - Displays information about the servers and hosts in the cell.
   • account catalog - Lists the accounts in the cell.
   • group catalog - Lists the groups in the cell.
   • keytab catalog - Lists the keytabs in the cell.
   • organization catalog - Lists the organizations in the cell.
   • principal catalog - Lists the principals in the cell.
   • principal delete { /.:host/<node> /.:ftp/<node> } - Removes the host and ftp principals that were configured for each of the additional network adapters in the nodes.

dce_login  Validates the DCE principal identity and obtains the principal network credentials. (In other words, the command used to log in to DCE.)
inutoc      Creates a .toc (table of contents) file for directories that have backup format file install images.

install_cw  Completes the installation of system support programs in the control workstation.

k4destroy  Destroys Kerberos 4 authentication tickets.
kdestroy   Destroys Kerberos 5 authentication tickets.
k4list     Lists currently held Kerberos 4 authentication tickets.
klist      Lists currently held Kerberos 5 authentication tickets.
lsauthent   Lists the authentication methods currently configured on the system.

lsauthpar    Lists the remote command authentication methods that are active for the system partition.

lsauthpts    Lists the trusted services authentication methods that are active for the system partition.

lsauthhts   Lists the authentication methods used by trusted services on the local host.
node_number
Obtains the node number attribute for a node from the ODM.

rm_spsec -v -t admin <nodename>
Removes configuration from DCE service principals and keyfiles. When admin is specified, the flag removes the local keyfiles and keytab objects for DCE servers.

rm_spsec -v -t local
Removes configuration from DCE service principals and keyfiles. When local is specified, the flag removes the local keyfiles and keytab objects for the host.

savebase Saves information about base-customized devices in the Device Configuration database onto the boot device.

SDRArchive
Archives the entire contents of the System Data Repository (SDR), except for the archives directory, for later retrieval.

SDRGetObjects Syspar
Sends contents of attributes in specified object (Syspar) to standard output.

setclock Sets the time and date for a host on a network.

setup_authent
Sets up a workstation to use SP authentication services (Kerberos 4).

setupdce -c cell_admin -l /.:/lan-profile
Creates DCE Security and CDS entries for SP nodes: specifies the DCE ID with cell administration authority and the DCE lan profile path.

setupdce -v -u -s <master_security_server> -d <initial_cds_server>
Creates DCE Security and CDS entries for SP nodes: specifies the DCE Master Security server and Initial CDS server, and updates the SDR with their hostnames.

setup_server
Configures a node or control workstation as a boot/install server.

show.cfg Displays the DCE and DFS components configured on the local machine and returns information about the configuration state and the running state.

spacs_cntrl
Controls interactive access to SP nodes.
spauthconfig
Installs and configures a node based on selected authentication methods. The command (called from /etc/rc.sp and psspfb_script) runs each time a node boots.

spbootins Enters boot/install configuration data for a node or series of nodes in the System Data Repository (SDR)

splstdata -e
Displays the configuration data from the System Data Repository (SDR) or system information for each node. The -e option displays environment information including: DCE cell name, DCE Security Master server name, and DCE CDS Master server name.

splstdata -n
Displays the configuration data from the System Data Repository (SDR) or system information for each node. The -n option displays node information including: DCE hostname for each node.

splstdata -p
Displays the configuration data from the System Data Repository (SDR) or system information for each node. The -p option displays partition information including: SDR Security Attribute settings (auth_install, auth_root_rcmds, ts_auth_methods, and auth_methods).

spnkeymand -l
Queries the SP Per Node Key Management daemon and displays the status of the SP Trusted Services keytab files.

spnkeyman_start
Conditionally starts the SP Per Node Key Management daemon on the control workstation, node, or standalone workstation. The command runs locally on the host where the daemon is to be started.

spsetauth -i -p <syspar> [dce][k4][std]
Sets the authentication methods to be installed on the control workstation and in the partition. The -i option specifies the install setup.

spsetauth -d -p <syspar> [dce][k4][std]
Sets the authentication methods to be installed on the control workstation and in the partition. The -d option specifies the authorization setup.

start.dce Starts the DCE components configured on the local machine.
stop.dce Stops the DCE components configured on the local machine.
sysctl whoami -v
   Returns values of all authenticated identities of the sysctl client.

unconfig.dce -config_type admin all -dce_hostname <node> -host_id <node>
   Stops DCE and unconfigures the specified DCE components. Admin unconfiguration is used by the cell administrator on any machine within the cell to update the CDS namespace and the Security registry about changes in the cell.

unconfig.dce -config_type local all
   Stops DCE and unconfigures the specified DCE components. Local unconfiguration is used by the root user on the machine being unconfigured to stop the daemons and delete the appropriate files.

updauthfiles
   Updates or creates the authorization files (.k5login, .klogin, and .rhosts) on the control workstation and on all the nodes in the system based on the setting of the auth_root_rcmd security attribute.
Appendix C. Using the additional material

This redbook also contains additional material in CD-ROM or diskette format, and/or Web material. See the appropriate section below for instructions on using or downloading each type of material.

C.1 Using the CD-ROM or diskette

The diskette that accompanies this redbook contains the following:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aixsetsec.ksh</td>
<td>This script does some security settings for you. It disables a number of services, depending on the system it runs on. You will need to adapt the script to prevent it from disabling services that you require. It checks for world-writable files and directories.</td>
</tr>
<tr>
<td>cds-replicate.ksh</td>
<td>Replicates an entire CDS server.</td>
</tr>
<tr>
<td>con_user.ksh</td>
<td>Example script to control access for DCE users.</td>
</tr>
<tr>
<td>getmail.ksh</td>
<td>Can be run from cron on the CWS and starts sendmail on the nodes to collect mail. When using this script, you can disable sendmail.</td>
</tr>
<tr>
<td>itcs204.ksh</td>
<td>Prepares a system for the IBM ITCS204 standard.</td>
</tr>
<tr>
<td>runtrip.ksh</td>
<td>A secure way of running tripwire on the nodes. Only use it on the CWS when it runs in secure mode, otherwise you need an external security server.</td>
</tr>
<tr>
<td>script.cust</td>
<td>Example script.cust customization file.</td>
</tr>
</tbody>
</table>

C.1.1 System requirements for using the CD-ROM or diskette

The following system configuration is recommended for optimal use of the diskette:

- **Hard disk space**: 1.5 MB minimum
- **Operating System**: Any UNIX
C.1.2 How to use the diskette

You can extract the contents of the diskette by issuing the following command after inserting the diskette into its drive:

```
tar -xvf /dev/rfd0
```

C.2 Locating the additional material on the Internet

The CD-ROM, diskette, or Web material associated with this redbook is also available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/SG245521
```

Alternatively, you can go to the IBM Redbooks Web site at:

```
ibm.com/redbooks
```

Select the Additional materials and open the directory that corresponds to the redbook form number.
Appendix D. Special notices

This publication is intended to help an RS/6000 SP Specialist who wants to implement the new IBM Parallel System Support Programs (PSSP 3.2) security infrastructure to make an RS/6000 SP system as secure as desirable. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM AIX and PSSP software products. See the Publications section of the IBM Programming Announcement for the products mentioned in this redbook for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been
reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM          IBM.COM
LoadLeveler   Nettinity
OS/2          PAL
Redbooks      Redbooks Logo
RS/6000       SP
System/390    TCS

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries.
licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Lotus Notes is a registered trademark of Lotus Development Corporation

Other company, product, and service names may be trademarks or service marks of others.
Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 509.

- Administering IBM DCE and DFS Version 2.1 for AIX and OS/2 Clients, SG24-4714
- DCE and DFS Performance Tuning and Problem Determination on AIX and OS/2 Warp, SG24-4949
- DCE Cell Design Considerations, SG24-4746
- Elements of Security: AIX 4.1, GG24-4433
- IBM Certification Study Guide: RS/6000 SP, SG24-5348
- Inside the RS/6000 SP, SG24-5145
- RS/6000 SP: Problem Determination Guide, SG24-4778
- RS/6000 SP System Performance Tuning, SG24-5340
- Security on the Web Using DCE Technology, SG24-4949
- The RS/6000 SP Inside Out, SG24-5374

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

<table>
<thead>
<tr>
<th>CD-ROM Title</th>
<th>Collection Kit Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM System/390 Redbooks Collection</td>
<td>SK2T-2177</td>
</tr>
<tr>
<td>IBM Networking Redbooks Collection</td>
<td>SK2T-6022</td>
</tr>
<tr>
<td>IBM Transaction Processing and Data Management Redbooks Collection</td>
<td>SK2T-8038</td>
</tr>
<tr>
<td>IBM Lotus Redbooks Collection</td>
<td>SK2T-8039</td>
</tr>
<tr>
<td>Tivoli Redbooks Collection</td>
<td>SK2T-8044</td>
</tr>
<tr>
<td>IBM AS/400 Redbooks Collection</td>
<td>SK2T-2849</td>
</tr>
<tr>
<td>IBM Netfinity Hardware and Software Redbooks Collection</td>
<td>SK2T-8046</td>
</tr>
<tr>
<td>IBM RS/6000 Redbooks Collection</td>
<td>SK2T-8043</td>
</tr>
<tr>
<td>IBM Application Development Redbooks Collection</td>
<td>SK2T-8037</td>
</tr>
<tr>
<td>IBM Enterprise Storage and Systems Management Solutions</td>
<td>SK3T-3694</td>
</tr>
</tbody>
</table>
E.3 Other resources

The following publications are also relevant as further information sources:

- **Administrative Core Services**, SC23-2730
- **AIX System User's Guide: Communications and Networks**, GC23-2523
- **AIX V4.3 Network Information Services (NIS and NIS+) Guide**, SC23-4310
- **AIX V4.3 System Management Concepts: Operating System and Devices**, SC23-4311
- **AIX V4.3 Technical Reference**, SBOF-1878
- **AIX Version 4.3 System Management Guide: Communications and Networks**, SC23-4127
- **IBM DCE for AIX, Administration Reference**, SC23-2732
- **PSSP Administration Guide**, SA22-7348
- **PSSP: Command and Technical Reference**, SA22-7351
- **PSSP Diagnosis Guide**, GA22-7350
- **PSSP: Installation and Migration Guide**, GA22-7347
- **RS/6000 SP: Planning Volume 2, Control Workstation and Software Environment**, GA22-7281
E.4 Referenced Web sites

The following Web sites are also relevant as further information sources:

- http://www.ssh.org
- http://www.bull.com
- http://www.postfix.org
- http://www.cerias.purdue.edu/coast/satan.html
- http://www.ssh.org
- http://all.net
- http://wuarchive.wustl.edu/packages/security/TAMU/
- IBM Redbooks:
  http://www.redbooks.ibm.com/
- Designing an Authentication System: a Dialogue in Four Scenes:
- Trusted Computing System Evaluation Criteria (TCSEC) Rainbow Series Library:
- Information Technology Security Evaluation Criteria (ITSEC) Assurance Level Criteria:
  http://www.itsec.gov.uk/
- Wietse Venema's (author of tcp_wrapper) tools and papers:
  http://sneezy.ice.ntnu.edu.tw/os/unix/security/
- The Admin guide to cracking by Dan Farmer and Wietse Venema:
  http://www.cerias.purdue.edu/coast/satan-html/docs/admin_guide_to_cracking.html/
- The Internet Assigned Numbers Authority:
  http://www.iana.org/
- Large archive of SMIT installable freeware:
  http://www.bull.de/pub
- Secure Shell homepage:
  http://www.ssh.org
- Computer Emergency Response Team:
  http://www.cert.org
• IBM emergency response team:
  http://www.ers.ibm.com

• Postfix homepage, replacement for sendmail:
  http://www.postfix.org

• Fred Cohen's page, programmer of the Deception Toolkit:
  http://all.net

• Chaos Computer Club (partly German):
  http://www.ccc.de

• Security information, large download area with hacker tools and security tools:
  http://packetstorm.securify.com

• This is a security site, not a hacker site:
  http://www.attrition.org

• Top 50 security sites:

• Kerberos FAQs:
  http://www.faqs.org/faqs/kerberos-faq/general/
How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** [ibm.com/redbooks](http://ibm.com/redbooks)
  
  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**
  
  Send orders by e-mail including information from the IBM Redbooks fax order form to:

  **e-mail address**
  
  In United States or Canada: pubscan@us.ibm.com
  
  Outside North America: Contact information is in the “How to Order” section at this site: [http://www.elink.ibmlink.ibm.com/pbl/pbl](http://www.elink.ibmlink.ibm.com/pbl/pbl)

- **Telephone Orders**
  
  **United States (toll free)**: 1-800-879-2755
  
  **Canada (toll free)**: 1-800-IBM-4YOU
  
  **Outside North America**:
  

- **Fax Orders**
  
  **United States (toll free)**: 1-800-445-9269
  
  **Canada**:
  
  
  **Outside North America**:
  
  Fax phone number is in the “How to Order” section at this site: [http://www.elink.ibmlink.ibm.com/pbl/pbl](http://www.elink.ibmlink.ibm.com/pbl/pbl)

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

---

**IBM Intranet for Employees**


© Copyright IBM Corp. 2000
Glossary

access control. The process of making access to system resources available only to authorized principals.

access control list (ACL). A list of principals and the type of access assigned to each.

ACL. Access Control List.

AFS. A distributed file system that uses Kerberos authentication and includes authentication services that can be used by an SP system.

AIX. Advanced Interactive Executive.

API. Application Programming Interface.

assurance. A measure of confidence that the features and design of a system accurately enforce the system's security policy.

authentication. The process of a principal proving the authenticity of its identity.

authentication database. A set of files containing the names and authentication information of all principals within a realm. An authentication realm has one primary database and may have multiple secondary databases. Secondary databases are backup copies of the primary database and may be provided to improve performance or availability.

authenticator. An authentication protocol string created each time authentication occurs and sent with the ticket to the server. It contains a time-stamp encrypted in the session key that can reliably show that the authentication request actually came from the client identified in the ticket.

authorization. The granting of access rights to a principal.

CDS. Cell Directory Service.

cell. An independently-administered collection of file server and client machines running AFS. An AFS cell is equivalent to a Kerberos realm.

client. A process requesting a service.

CPU. Central Processing Unit.

credentials. A protocol message, or part of one, containing a ticket and an authenticator supplied by a client and used by a server to verify the client's identity. The message can contain additional information used by the server to verify its identity to the client.

CSS. Communication Subsystem (SP Switch).

CWS. Control Workstation.

Data Encryption Standard (DES). The secret-key (also known as "private-key") encryption algorithm that is used by Kerberos.

DCE. Distributed Computing Environment.

DES. Data Encryption Standard.

DFS. Distributed File System.

Distributed Computing Environment (DCE). Software designed and developed through the Open Group. AIX DCE is an implementation of DCE for AIX RS/6000 systems.

discretionary access control. A means of restricting access to objects based on the identity of the principal.

DNS. Domain Name Service.

domain. A collection of systems over which an administrator exercises control.

DTS. Distributed Time Service.

GB. gigabytes.


GSSAPI. Generic Security Service API.

IBM. International Business Machines Corporation.

identification. The process of stating the identity of a principal; no proof as to the authenticity of the identity is implied.

instance. A qualifier for a principal name. For services, an instance represents a particular occurrence of the server. For users, an instance
allows a single user to assume additional (or alternate) roles with different authority.

**ITSO.** International Technical Support Organization.

**Kerberos.** A mechanism for providing mutual authentication of two principals using a trusted third party in a distributed environment.

**Kerberos V4.** The fourth version of the Kerberos authentication mechanism from the Massachusetts Institute of Technology (MIT). Used in pre-PSSP3.2 SP systems for authentication within some administrative services.

**Kerberos V5.** The fifth version of the Kerberos authentication mechanism from MIT. Kerberos V5 is not protocol-compatible with prior versions of Kerberos. Kerberos V5 is used as the basis of distributed service authentication within DCE.

**key.** A value used to encrypt protocol strings used for authentication. The private keys of principals are stored in the authentication database. Session keys are contained (encrypted) in tickets and other protocol strings.

**key management.** The process of periodically changing the key associated with a server’s DCE principal to prevent the key from expiring and, thus, disabling the service.

**Kerberos master key.** The key derived from the Kerberos Master Password supplied initially by the administrator when the primary SP authentication server is created. This key is saved in the /.k file, which the Kerberos daemons read instead of prompting for a password. It can also be read by certain database utility commands for the same purpose.

**Kerberos master password.** The password the administrator supplies when initializing the primary authentication server.

**LAN.** Local Area Network.


**MB.** megabytes.

**MIT.** Massachusetts Institute of Technology.

**MPP.** Massively Parallel Processors.

**mutual authentication.** The process of two principals proving their identities to each other.

**NFS.** Network File System.

**NTP.** Network Time Protocol.

**object.** An entity that contains or receives information; access to an object potentially implies access to the information it contains. Examples of objects are files, jobs, queues, nodes, and users.

**OSF.** Open Software Foundation

**per-node key management.** The process of providing key management on each node in an SP system.

**principal.** A user, an instance of a server, or an instance of trusted client code whose identity is to be authenticated.

**PSSP.** Parallel System Support Programs.

**PV.** Physical Volume.

**realm.** A domain that shares an authentication database and servers. There is a single name space for principal name/instance pairs within a realm. A realm is also a logical collection of clients and servers registered in the database.

**RPC.** Remote Procedure Call.

**RSCT.** RS/6000 Cluster Technology (IBM).

**RVSD.** Recoverable Virtual Shared Disk (IBM).

**SCSI.** Small Computer System Interface.

**SDR.** System Data Repository (IBM SP).

**security policy.** The set of rules that determines how a system manages, protects, and distributes sensitive information.

**server.** A process providing a service - the entity to whom the client's identity must be proven. A server is trusted code. In general, a server runs as a separate process. However, the attribute of being a server is merely a role that a process assumes. In some instances, code that is traditionally part of a server may assume the role of a client.

**service.** The name of a principal whose identity is assumed by a server for purposes of
authentication. Multiple servers can use the same service name.

**server key file** (also known as a srvtab). A file containing the names and private keys of the local instances of services. It is accessible only to processes that run under the UID of the user owning the server daemons. On an SP system, all services run as root.

**session key.** A temporary key supplied by an authentication server to clients and servers and used to encrypt parts of authentication protocol messages. Its lifetime is the same as the ticket with which it is created.

**Smit.** System Management Interface Tool (IBM AIX).

**sniffing.** Listening to traffic on a network.

**spoofing.** A system that falsely presents itself as another system on the network.

**SSA.** Serial Storage Architecture.

**ticket.** An encrypted protocol message used to pass the identity of a user from a client to a server. Tickets are created by the Kerberos authentication server and cached in disk files on the client’s system.

**ticket-granting-ticket.** The initial ticket obtained by a user. It is used by client programs to obtain additional tickets for authentication with application services.

**trusted service.** A service that is responsible for enforcing some part of the system’s security policy.
Symbols

15

*none* 441
$HOME/.k5login 439
$HOME/.Xauthority 207
*-srvtab 216
.forward 222
.k5login 13, 439
.k5login file demystified 454
.klogin 12
.netrc 204, 206
.rhosts 12
.ssh/authorized_keys 215
.Xauthority 207, 208
./k 169
./k5login 14
./klogin 14, 169
./rhosts 14, 206
/etc/aliases 222
/etc/dce/cfg.dat 406, 409
/etc/dce/security/pe_site 396
/etc/environment 49
/etc/exports 203
/etc/filesystems 399
/etc/hosts.allow 220
/etc/hosts.deny 220
/etc/hosts.equiv 206
/etc/inetd.conf 204, 205
/etc/initab 204
/etc/krb.conf 168, 453
/etc/krb.realms 168
/etc/krb5.conf 453
/etc/krb-srvtab 12, 168
/etc/logmgt.acl 29, 292
/etc/rc.sp 15
/etc/rc.tcpip 205
/etc/security/failedlogin 233
/etc/security/passwd 12, 203
/etc/security/sysck.cfg 227
/etc/security/user 205, 225
/etc/sendmail.cf 222
/etc/ssh 214
/etc/sysctl.acl 29, 292
/etc/sysctl.conf 185, 195, 280, 298
/etc/sysctl.haem.acl 194, 292
/etc/sysctl.install.acl 194, 292
/etc/sysctl.mmcmd.acl 29, 292
/etc/sysctl.pman.acl 29, 292
/etc/sysctl.rootcmds.acl 29, 194, 292
/etc/sysctl.vsd.acl 29, 292
/etc/syslog.conf 234
/krb5/ 31
/krb5/v5srvtab 31, 453
/opt/dcelocal/etc/cfgdce.log 238
/opt/dcelocal/etc/passwd_overwrite 224
/opt/dcelocal/etc/security/pe_site 48, 111, 112
/opt/dcelocal/var/security/creds/ 418
/opt/dcelocal/var/security/creds/dcecred_ffffff 419, 439
/opt/dcelocal/var/security/creds/dcecred_XXXXXXXXX 418, 439
/spdata/sys1/install/lppsource 216
/spdata/sys1/install/{default}/lppsource 243
/spdata/sys1/install/pssp 216
/spdata/sys1/install/pssplpp/PSSP-X.Y/ 216
/spdata/sys1/keyfiles/ 32, 384
/spdata/sys1/keyfiles/ssp/ 13
/spdata/sys1/spmon/hmacls 320, 328
/spdata/sys1/spmon/hmndceacs 331
/spdata/sys1/spsec/auth_methods 247, 258
/spdata 258
/tftpboot (except *-srvtab files) 216
/tmp 268
/tmp becomes full 348
/tmp is full 268
/tmp/kerberos.out 176
/tmp/krb5cc_UID 420
/tmp/tkt0 192
/usr/adm/tcheck/databases 228
/usr/adm/tcheck/tw.config 228
/usr/bin/chpesite 48
/usr/lib/libspk4rcmd.a 249, 440
/usr/local/bin/cds-replicate.ksh 109
/usr/local/lib/p-cops.ksh 232
/usr/local/tiger/tiger 233
/usr/local/tiger/tigerrc 232
/usr/lpp/bin/spauthconfig 47
/usr/lpp/bin/spauthdump 176
/usr/lpp/bin/rcmdtgt 192
/usr/lpp/samples/sysctl/firstboot.cmds 196
/usr/lpp/spysysctl/bin/firstboot.cmds 195
/usr/lpp/spysysctl/bin/install.cmds 195
/usr/lpp/spysysctl/bin/switch.cmds 195

© Copyright IBM Corp. 2000
/usr/sbin/savebase 158, 168
/usr/security/bin/tripwire 228
/usr/security/bin/tripwire -initialize 228
/var
    Files
    /var/* 346
/var becomes full 346
/var/adm/SPlogs/ 238
/var/adm/SPlogs/auth_install/log 238
/var/adm/sulog 233
/var/dce 346
    core files in 72, 113, 136, 151
    create as separate filesystem 72, 113, 136, 150
    minimum recommended size 72, 113, 136, 151
    redbook recommended size 72, 113, 136, 151
/var/dce becomes full 346
/var/dce/svc/ 238
/var/kerberos/database 169
/var/spool/mail 222
/var/sysctl/db_acl 297
/var/sysctl/db_name 297
/var/sysctl/db_object 297
/var/syslog/syslog.debug 234
    _other_unauth 194

Numerics
7 digit message codes 442
80/20 rule 3, 201

A
about the AIX authentication methods 438
about the PSSP methods 247
acceptable risk 3
access control lists 220
Access Control Lists (ACLs), DCE
    See DCE Access Control Lists (ACLs)
    access control with spacs_cntrl 205
    access control, users 268
    access, restricting 205
    accessing online help, sysct1 293
    accidentally disabling all logins 158, 168, 172
    acl 297, 331
acladd 292
aclcheck 292, 297
aclcreate 292
acldelte 292
acldestroy 292
acclist 292
accrecreate 292
ACLs
    sysctl 281
    sysctl examples 291
activating the filter mechanism 211
adapter, denying all access to via IPSec 219
adding a node 239
adding security options from top down 17
adding the SP administrator DCE principal to the
    DCE cell 93
additional DCE/k5 errors relating to remote com-
    mands 469
additional tools and techniques 9
administrative domain 25
admin-only configuration 114
admin-only part 47
ADSM 74
Advanced DCE cell
    Example 4 91
AFS 207
AFS authentication server
AIX
    and PSSP remote command debug tips/check-
    list 452
    and PSSP remote commands 435, 442
    changing authentication methods 265, 266, 267
    differences with PSSP authentication methods
    245
disjoint set of authentication methods 443
failover of authentication methods 438
filesets are not installed 243
identity 269
querying authentication methods 265, 266, 267, 438
remote command message summary 441
user ID managing 270
AIX 4.3.3, NIS+ 204
AIX and PSSP remote command debug tips/check-
    list 452
AIX and PSSP remote commands 184, 435, 442
AIX authentication methods 248, 259
AIX authentication methods cannot be enabled 243
AIX file permissions 33
AIX filesets are not installed 243
AIX remote command message summary 441
AIX remote commands 12, 15
AIX/PSSP Kerberos V4 integrated login 270, 440
All PSSP services appear to be hung 342
allow user access to nodes 205
Am I authorized? 271
AND COMPAT 223
Andrew File System (AFS)
   See AFS
APARs 226
arthur 112
asymmetric encryption 486
Athena project 485
attack, denial of service 7
attacks, internal 209
attribute settings 17
attribute, rlogin 223
attribute, SYSTEM 225
atypical credentials states 429
AUTH_DES 203, 204
auth_install 14, 15, 17
auth_install, defined 15
auth_methods 14, 16, 17, 248, 438
auth_methods, defined 15
auth_root_rcmd 14, 15, 17, 184, 195
auth_root_rcmd, defined 15
AUTH_SYS 203
AUTH_UNIX 203
Authentication
   Service 485
   System 486
authentication 11, 13
authentication and authorization 11
authentication keys 30
authentication methods, AUTH_UNIX, AUTH_SYS,
and AUTH_DES 203
authentication service 30, 34
authentication, defined 11
authenticator 489
authorization 12, 13
authorization and authentication 11
authorization, defined 11
authorized_keys 215
automatic login authorization 206
automatic root access 13, 14
automounter 215
avoid supper 217

B
backup system, centralized 208
backup tapes offline 202
backups, securing 208
base security option, defined 15
base security options 15, 17
bffcreate 76
binding 49
blocking port 2049 212
bnc cabling 226
boot logical volume 158, 168
boot/install servers 14
Boot/Install Servers with RRA 187
bootp server 225
bootstrap file
   See DCE Security server bootstrap file
bootstrap file, /opt/dcelocal/etc/security/pe_site
   111, 112
bos.net.ipsec.rte 211
broadcast 225
business trade-offs 3

can not create Kerberos V4 ticket
   example 349
can not obtain DCE credentials 346
   examples 347
can not obtain Kerberos V4 credentials 346
CDE
CDS database
   See clearinghouse
CDS namespace replication 108
CDS servers cannot be reached 239
cdsadv 79
cdsclerk 79
cdsd 79
cdsli 241
cdsli -cworld 80
cell
   See DCE, cell
Cell Directory Service (CDS)
   See DCE, CDS
cell_admin 26
cell-administrator principal
   See DCE cell-administrator principal
central point of control 4
centralized backup system 208
Cerberus 485
chargen 204
chauthent 15, 158, 168, 172, 245
chauthpar 245, 267
chauthpts 246, 256
chauths 246
checkauth 286
checksum, double 228
choosing the right security technology for your environment 6
chpesite 111
croot 222
circular dependency 51, 160
circumventing added security layers 28
cl_setup_kerberos 186, 187
class, SP_Restricted 184
clean_up.dce 399
cleaning up a failed DCE customization 478
cleaning up a failed DCE node customization 478
cleaning up a failed DCE node installation 478
clear text 213
clearinghouse 25
clock source 223
cloning
See mksysb creation in a DCE environment
coexistence, RRA 186
Cohen, Fred 230
combination security option, defined 15
combination security options 17
Command
  services_config 215
Commands
  /etc/rc.sp 15
  /spdata/sys1/spsec/auth_methods 258
  /usr/bin/chpesite 48
  /usr/local/bin/cds-replicate.ksh 109
  /usr/local/tiger/cds-replicate.ksh 233
acl 297, 331
acladd 292
aclcheck 292, 297
aclcreate 292
acldel 292
acldestroy 292
acllist 292
accrecreate 292
bffcreate 76
cdsli 241
cdsli -cworld 80
chaoutent 15, 158, 168, 172, 245
chaouthpar 245, 267
chaoutpts 246, 256
chaouthts 246
checkauth 286
chpesite 111
croot 222
cl_setup_kerberos 186, 187
clean_up.dce 399
config_spsec 241, 388
crack 233
create_keyfiles 386
CSS_test 270
dce_login 241
dce_login_noexec 347
dcecp 241
dcecp -c cell show 141
dsh 245, 262, 263, 264
Eclock 188
Efence 188
egrep 400
Eprimary 188
Equiesce 188
Estart 188
Eunfence 188
Eunpartition 188
expfilt 213
firstboot.cust 196
ftp 245
gauth 258
hmadm 328
hmckacl 328, 331
hmdevobj 331, 336
hmgetacls 328, 331
impfilt 213
k4init 423
klist 27
ksrvtgt 423, 440
lppdiff 188
lsauthent 245, 249, 438
lsauthpar 245, 249, 438
lsauthpts 246, 247, 251
lsauthths 246, 247
lsfs 400
lsof 211
lssrc 344
lssrc -s spnkeyman 39
mult_senders_test 188
netstat 241
netstat -i 139
pcar 188
pcops 232
pcp 188, 245
pexec 188
pfind 188
pkill 189
pls 189
pmv 189
ppred 189
pps 189
prm 189
pssp_script 60
psspfb_script 73, 91, 106, 129
ptest 189
rc.sp 225
rcmdtgt 423, 440
rcp 245
rlogin 245
rmxcred 348, 422
rsh 245, 262, 263, 264
rusers 204
rwall 204
s1term 206, 225
savebase 158, 168
script.cust 73, 216
SDR_config 196
SDRGetObjects SP_ports 212
setup_server 176, 215
setupdce 86, 100, 123, 241
show.cfrg 141, 241
spacind 189
spacl 297, 308, 331
spacs_ctl 205, 223
spauthconfig 91, 106, 129, 244
spchgvobj 196
spgetdesc 189
splstdata 189, 247, 253, 438
spmkuser 189
spmkvobj 196
spnmon 193
spnkeyman_start 91, 105, 106, 128, 150
spnkeymand 374
spnkeymand -l 37
spray 204
spstenv 184, 186, 190, 196
ssh-keygen 214
startsrc -s sysctld 195
stopsrc -s sysctld 195
su 205
supper 49, 217
sysctl 183, 247, 252, 279
sysctl whoami 290
syspar_ctrl 215
talk 204
tcbck 227, 228
telnet 189, 245
tigexp 233
upauthfiles 15, 184
whatacl 292
whatacls 297
whoami 290
wrap_test 188
xauth 206, 207, 208
xhost 206
Common Desktop Environment (CDE)
   See CDE
Common periods for problems 238
common problems 237
compat 247
Compatibility 16, 17
compatibility mode 50
Compatibility Security
   Example 2 66
      four different Kerberos V4 configurations 66
      single Kerberos realm 67
Compatibility security state 17
Compatibility to DCE
   DCE version 3.1 filesets required on the CWS
   76, 137
      final verification 159
      further considerations 160
      high-level outline 134
      moving from 133
      optional additional DCE configuration 159
      planning considerations 134
      step-by-step procedure 135
Compatibility to Minimal
   final verification 169
      high-level outline 161
      moving from 160
      planning considerations 161
      removing Kerberos V4 161
      step-by-step procedure 162
complex and error-prone 187
Computer Oracle and Password System (COPS)
   See COPS
config_spsec 241, 388
config_spsec did not run successfully 241
configuring /etc/syslog.conf to forward to another
server 234
configuring the admin-only part of the DCE client
115
configuring the local-only part of the DCE client 117
conflicting settings of security attributes 16
Control Workstation
PSSP DCE keyfile differences with that of the
nodes 385
Control Workstation (CWS)
DCE hostname via netstat -i listing 139
required DCE version 3.1 filesets 76, 115, 137
Control Workstation only PSSP DCE keyfiles 386
cookbook 6
cookie 206, 207
COPS 227, 231, 232
COPS command, pcos 232
COPS directory, /usr/local/lib/p-cops/ 232
COPS, perl version 232
copy_env_files 196
correcting SDR security attributes 243
corrective actions 237
countermeasures 226
crack 233
create_keyfiles 386
creating /var/dce as separate filesystem with
script.cust 73, 113
credentials cache locations 417
credentials states, atypical 429
credentials states, typical 426
css 40
css keyfile 33
CSS_test 188
customizing 216
CWS
See Control Workstation
CWS as relay host, do not do it 222
CWS is a member of each and every partition 18
CWS, partitioning, and security 52
D
daemon
defined 4
daemons
dnsp 225
cdsadv 79
cdscldr 79
cdded 79
cdc 12, 34, 35, 79
dcunixd 223
hardmon 248, 320
ineld 221
kermon 12
Per Node Key Management 38
postfix 221
querying inetd status 344
srdr 32
scd 12, 79
sendmail 217
sshd 208, 214
sysctld 185, 218
syslogd 234
Dan Farmer 229, 232
Data Encryption Standard (DES) 486
Data Fellows Inc. 214
data, malformed 377, 379
database, read-only copies 25
database, read-write copies 25
daytime 204
DCD
DTS 24
DCE 5, 7, 12, 16, 17, 239
/opt/dcelocal/etc/cfgdce.log 238
/opt/dcelocal/etc/security/pe_site bootstrap file
111, 112
/usr/local/bin/cds-replicate.ksh 109
Access Control Lists (ACLs) 13
ACL permissions, viewing 34
ACL types 271
ACLs 33, 271
adding replica servers to improve availability 92
adding the SP administrator principal to the cell 93
additional servers as first node on each ethernet
segment 50
admin log 238
administrative principals 26
admin-only configuration 114
and Kerberos V4 credentials 416
architecting the correct solution 160
attributes and PNKM 368
attributes that impact PNKM 369
Authentication Service 30
binding 49
bypassing CDS lookup 48
cannot obtain credentials 346
CDS
CDS client configuration failure 410
CDS servers cannot be contacted 391
CDS/RPC entries 299
cell 223
cell reconfiguration 239, 353
cell_admin principal 27
cell_admin principal, SP recommendation 28
cell-administrator principal
cell-administrator principal defined 26
cell_administrator principal 27
chpsite command 111
circular dependency with PSSP on CWS 51
cleaning up a failed node customization 478
cleaning up a failed node installation 478
client does not start successfully 381
client machine checking 392
client machine checkpoints 392
client recovery from a full filesystem 348
client/server problem determination 390
components, illustrated 24
concept of a cell 25
configuration listing 141
configuring the admin-only part of the client 115
configuring the CWS as a DCE client 114
configuring the local-only part of the client 117
configuring, things to check first 72, 113, 134
considerations when Masters are on nodes 160
Control Workstation only PSSP DCE keyfiles 386
core files 72, 113, 136, 151
creating a cell 135
creating the /var/dce filesystem 72, 113, 136, 150
credentials cache locations 418
daemons, cdsadv 79
daemons, cdsclerk 79
daemons, cbsd 79
daemons, dced 79
daemons, secd 79
dced (the self-host principal) keyfile is /krb5/31
dceunixd 223
defined
deleting credentials 421
DFS LPP 8
directory service 8
distributed file service 8
Distributed Management Environment 23
DTS
entry-level 50
examples of cache locations 419
examples of Registry Status 243
expired PSSP DCE keyfile 376, 378
filesets are not installed 243
forwardable ticket attribute 312
forwarding credentials 310
full client configuration 114
fully configured client 407
GDS
general checkpoints 405
group membership 273, 321
hostname 139
identity 269
improving availability in the cell 92
incomplete configuration 471
incorrect level 241
incorrect mount points (/var/dce locations) 399
installp fails 244
integrated login 223
its /etc/dce/cfg.dat file 406
joining an existing cell in the future 160
Kerberos V5 5, 7, 12
key management service 35
key versions 35
keyfile management 35
keyfile, changing a password in 34
keyfile, physical location 31
keyfiles were not created 243
keyfiles, how they differ on the CWS and nodes 32
keys (aka passwords) 35
keytab files 13
keytab list, CWS example 30
keytab list, node example 30
keytab object 30
local configuration -- unable to get key version 412
local-only configuration 114
missing interface names in self-host keyfile 412
missing PSSP DCE keyfile 377
mksysb image creation 73
nopolicy 372, 373
not configured for PSSP 375
not configured properly 243
not installed or configured 375
optional SP administrator principal(s) 27
optional SP administrator principal(s) defined 26
example SP-Administrator principal(s)
options for setting it up  49
partially configured client  408
password expiration date (pwdexpdate)  368
password lifetime (pwdlife)  368
pe_site file parsing  112
performance and load balancing with the pe_site file  112
preconfiguration checklist  72, 113, 134
Primary CDS server
primary components  23
principals  26
PSSP 3.2 software dependencies  160
PSSP group requirements for hardmon  321
PSSP group, haem-users  28
PSSP group, hm-admin  28
PSSP group, hm-control  28
PSSP group, hm-monitor  28
PSSP group, sdr-admin  28
PSSP group, sdr-restricted  191
PSSP group, sdr-system-class-admin  28
PSSP group, sdr-system-class-write  28
PSSP group, sdr-write  28
PSSP group, sspec-admin  29
PSSP group, switchtbld-clean  29
PSSP group, switchtbld-load  29
PSSP group, switchtbld-status  29
PSSP group, sysctl-cwssrte  29
PSSP group, sysctl-default  29
PSSP group, sysctl-logmgt  29
PSSP group, sysctl-mmcmd  29
PSSP group, sysctl-pman  29
PSSP group, sysctl-vsd  29
PSSP groups  27
PSSP Keyfiles  29
PSSP Trusted Services keyfiles are under /sp-data/sys1/keyfiles/  8
purpose of the cell in PSSP 3.2  133
rcmd principal  193
reconfiguring a PSSP DCE keyfile  383, 384
recovering failed client config due to time skew  351
redbook recommendation  8
reflecting group membership  277
refreshing credentials  420
refreshing self-host credentials  273
Registry  40, 370
removing credentials  421
replicating CDS with /usr/local/bin/cds-repli-
dce.client 76, 115, 138
  dce.client.core.rte 77, 115, 138
  dce.client.rte 77, 115, 138
  dce.client.rte.admin 76, 115, 138
  dce.client.rte.cds 76, 115, 138
  dce.client.rte.config 76, 115, 138
  dce.client.rte.pthreads 77, 115, 138
  dce.client.rte.rpc 76, 115, 138
  dce.client.rte.security 76, 115, 138
  dce.client.rte.time 77, 115, 138
  dce.client.rte.web 77, 115, 138
  dce.client.rte.zones 77, 115, 138
  dce.client.smit 77, 115, 138
  dce.security 77, 137
  dce.security.rte 77, 137
  dce.security.smit 77, 137
  dce.xdsxom 77, 115, 138
  dce.xdsxom.rte 77, 115, 138
  dce_login 241
  dce_login_noexec 347
  dcecp 241
  dcecp -c cell show 141
  dced 12, 34, 35, 79
  dceunixd 223
  deadlock 160
  dealing with mksysb images in a DCE environment 73
  deception program 230
  Deception Toolkit (DTK) 230
  Deception Toolkit and TCP Wrappers 231
  Deception Toolkit, Fred Cohen 230
  deceptive ports 230
  default Kerberos V4 realm name 170
  default partition 32
  define a security policy 201
  defined policy 291, 327
  definitions, filter 210, 213
  deleting a node 239
  deleting DCE credentials 421
  deleting Kerberos V4 credentials 425
  denial of service attack 7
  deny access to the system by default 205
  deny all traffic on the adapter rule, placement of 220
  deny user access to nodes 205
  denying all access to a certain adapter 219
  denying all ports 219
  denying all traffic with filter rules 210
  DES

See Data Encryption Standard
DES-based encryption 204
DES-encrypted 12
detecting intruders 199, 200
DFS 8, 207, 215, 229
different security policies in different partitions 18
disabling all logins 158, 168, 172
disabling NFS services 212
disabling services 204
disabling services, how to 204
disabling unnecessary services 202, 204
disaster recovery plan 209
disaster recovery services 208, 209
discard 204
disconnecting a node from the external network 226
disjoint security settings 245
disjoint set of AIX authentication methods 443
disjoint set of methods 259
displaying default hardmon DCE ACLs 321
Distributed Computing Environment (DCE)
  See DCE
Distributed File System (DFS)
  See DFS
Distributed Management Environment
  See DCE, Distributed Management Environment
Distributed Time Service (DTS)
  See DCE, DTS
do not allow automatic login authorization 206
do not use the CWS as a relay host 222
domain, administrative 25
double checksum 228
Doug Schales 232
dpid2 205
DR
  See disaster recovery services
dsh 188, 245, 262, 263, 264
  robust 310
dtspcd 204

each SP partition has its own security state 17
eavesdropping 485
echo 204
Eclock 188
Ecommands with RRA 188
ecuring your backups 208
Efence 188
egrep 400
Elements of security
AIX 4.1 199
enabling technology 23
encrypt 7
encrypted 209
encrypted authentication process 7
encrypted file 30
encrypted login sessions 209
encrypted login sessions, using 213
encrypted X traffic 208
encrypted, authenticated tunnel 214
encryption 7
encryption via hardware card 213
encryption, DES-based 204
encryption, performance impact 213
enhance filtering by IPsec 218
entry-level DCE 50
environment variable,
RPC_UNSUPPORTED_NETADDRS 49
environment variable,
RPC_UNSUPPORTED_NETIFS 49
environment variable, TRY_PE_SITE 48
Eprimary 188
Equiesce 188
establishing higher granularity of object protection 73
Estart 188
e etc 303
ten/logmgt.acl 296
ten/sysctl.acl 296
ten/sysctl.haem.acl 296
ten/sysctl.install.acl 296
ten/sysctl.mmmcmd.acl 296
ten/sysctl.pman.acl 296
ten/sysctl.rootcmds.acl 296
ten/sysctl.vsd.acl 296
Eunfence 188
Eunpartition 188
Event Management 5
Example 5 112
example security scenarios 199
examples of changing AIX authentication methods 265, 266, 267
examples of changing PSSP authentication methods 255, 257
examples of correcting hardmon DCE ACL database problems 339
examples of correcting sysctl DCE ACL database problems 315
examples of DCE cache locations 419
examples of DCE Registry organization, account attributes 371
examples of DCE Registry Status 243
examples of hardmon authorization failures 322
examples of hardmon DCE ACL management/query command 334
examples of managing DCE ACLs and ACL content 300
examples of querying AIX authentication methods 265, 266, 267
examples of querying authorization requirements of sysctl commands 287
examples of querying PSSP authentication methods 255, 257
examples of rmxcred 423
examples of robust dsh 310
examples of s1term success/failure 323
examples of shutdown ACL 272
examples of successful rsh/dsh 262, 263, 264
examples of successful sysctl authentication 252
examples of successful/unsuccessful SDR write access 274
examples of sysctl ACL query via a script 313
examples of sysctl ACLs 291
examples of sysctl authentication failure 281, 286
examples of sysctl DCE ACL permissions 305
examples of sysctl DCE ACL types 305
examples of unsuccessful rsh/dsh 262, 263, 264
examples of unsuccessful sysctl authentication 252
examples of verifying PSSP authentication methods 251
Existing DCE cell 112
expfilt 213
expired PSSP DCE keyfile 376, 378
exploits 226
export filesystems read-only 203

F
failover of authentication methods 15
AIX 248
PSSP 247
faking programs 230
Farmer, Dan 229
COPS, Dan Farmer 232
file collections 49
Files
$HOME/.Xauthority 207
-srvtab 216
.forward 222
.k5login 13, 439
.klogin 12
.netc 204, 206
.rhosts 12
.ssh/authorized_keys 215
.Xauthority 207, 208
.k5login 14
/.klogin 14
/.rhosts 14, 206
/etc/aliases 222
/etc/dce/cfg.dat 406, 409
/etc/dce/security/pe_site 396
/etc/environment 49
/etc/exports 203
/etc/filesystems 399
/etc/hostos.allow 220
/etc/hostos.den y 220
/etc/hostos.equiv 206
/etc/inetd.conf 204, 220, 230
/etc/inittab 204
/etc/krb.conf 453
/etc/krb5.conf 453
/etc/krb-srvtab 12
/etc/logmgt.acl 29, 292
/etc/rc.tcpip 205
/etc/security/failedlogin 233
/etc/security/passwd 12, 203
/etc/security/sysck.cfg 227
/etc/security/user 205, 225
/etc/sendmail.cf 222
/etc/ssh 214
/etc/sysctl.acl 29, 292
/etc/sysctl.conf 185, 195, 280, 298
/etc/sysctl.haem.acl 194, 292
/etc/sysctl.install.acl 194, 292
/etc/sysctl.mmcmd.acl 29, 292
/etc/sysctl.pman.acl 29, 292
/etc/sysctl.rootcmds.acl 29, 194, 292
/etc/sysctl.vsd.acl 29, 292
/etc/syslog.conf 234
/krb5/ 31
/krb5/v5srvtab 31, 453
/opt/dcelocal/etc/confdce.log 238
/opt/dcelocal/etc/passwd_overwrite 224
/opt/dcelocal/etc/security/pe_site 48, 111, 112
/opt/dcelocal/var/security/creds/* 418
/opt/dcelocal/var/security/creds/dcecred_ffffffff 419
/opt/dcelocal/var/security/creds/dcecred_XXXXXXXXX 418, 439
/spdata/* 258
/spdata/sys1/install/lppsource 216
/spdata/sys1/install/[default]/lppsource/* 243
/spdata/sys1/install/pssp 216
/spdata/sys1/install/pssplpp/PSSP-X.Y/ 216
/spdata/sys1/keyfiles/ 32
/spdata/sys1/keyfiles/* 384
/spdata/sys1/keyfiles/ssp// 13
/spdata/sys1/spmon/hmcals 320, 328
/spdata/sys1/spmon/hmcdeacals 331
/spdata/sys1/spspauth/methods 247
/tftpboot 216
/tmp/* 268
/tmp/krb5cc_UID 420
/tmp/tkt0 192
/usr/adm/tcheck/databases 228
/usr/adm/tcheck/tw.config 228
/usr/lib/lbspk4rcmd.a 249, 440
/usr/local/lib/p-cops/ 232
/usr/local/tiger/tiger 233
/usr/local/tiger/tigerc 232
/usr/lpp/spp/bin/spauthconfig 47
/usr/lpp/spsp/cmd/bin/rcmdtgt 192
/usr/lpp/spsp/samples/sysctl/firstboot.cmds 196
/usr/lpp/spsp/sysctl/bin/firstboot.cmds 195
/usr/lpp/spsp/sysctl/bin/install.cmds 195
/usr/security/bin/tripwire 228
/var/adm/SPlogs/* 238
/var/adm/SPlogs/auth/insta ll/log 238
/var/adm/sulog 233
/var/dce/* 346
/var/dce/svc/* 238
/var/spool/mail 222
/var/sysctl/db_acl 297
/var/sysctl/db_name 297
/var/sysctl/db_object 297
/var/syslog/syslog.debug 234
/var/syslog/syslog.debug 234
authorized_keys 215
e tc/sysctl.vsd.acl 296
hosts.equiv 12
identity.pub 215
pe_site 381
filter definitions 210, 213
filter mechanism, activating 211
filter rule 219
filter rule evaluation order 220
filter rule parsing 212
filter rule processing 210
filter rules 210
filter rules to allow traffic in one direction 219
filter rules, denying all traffic 210
filter rules, permitting all traffic 210
filter rules, setting up 211
filtering 210
final verification
  Compatibility to DCE 159
  Compatibility to Minimal 169
  Minimal to Compatibility 181
firewalls 3
firstboot.cust 196
firstboot-cust procedure 196
flooding 7
forwardable credentials 312
forwardable DCE ticket attribute 312
forwarding DCE credentials 310
four different Kerberos V4 configurations with Compatibility security 66
four security attributes 15, 16
fping 229
fraud, network
  See network fraud
Fred Cohen 230
ftp 245
full client configuration 114
fully configured DCE client 407
further considerations
  Compatibility to DCE 160

G
general DCE checkpoints 405
General Parallel File System (GPFS) 5
general security considerations 3
generating public/private key pairs with ssh-keygen 214
Generic Security Services API (GSSAPI) 8
getauth 286
GID 225
Global Directory Service (GDS)  

See DCE, GDS
global SP principal root.SPbgAdm 185
good idea 185
GPFS 33, 186, 191
  See General Parallel File System
GPFS not supported in RRA 14
Group Services 5
GSmonitor 40, 42, 45, 46
GSSAPI
  See Generic Security Services API

H
hacker 226, 228
hackers 217, 226
hackers, Internet-based 209
hackers, professional 209
HACMP 4.3.1 186
HACMP ODM 187
HACMP topology 186
HACMP with RRA 186
HACMP workaround 187
HACMP/ES 14
HACMP/ES 4.3.0 186
HACWS restrictions and limitations 187
HACWS with RRA 187
HAEM 36
haem 5
haem-users 28
HAGS 36
hags 5
hardmon 4, 193, 248, 320
  access control via ACLs and groups 320
  ACL and group access controls 324
  ACL permissions 327
  common DCE ACL entry types 307
  DCE ACL database problems/corrective actions 338
  DCE ACL entry types 306
  DCE ACL manager 338
  DCE ACLs 331
  DCE CDS/RPC entries 331
  examples of authorization failures 322
  examples of correcting DCE ACL database problems 339
  hierarchical DCE objects and ACLs 333
  high-level DCE ACLs 337
  Kerberos V4 ACL 328
  Kerberos V4 ACL permissions 323
IPSec, allowing traffic in one direction 219
IPSec, denying all access to a certain adapter 219
isjoint 245
ISO date stamp 40, 43
ISO time stamp format 37
itso_cell 112

J
joining an existing DCE cell, things to consider 160
jueneman 25

K
k4 195, 196, 440
k4init 423
k5 439
KSMUTE 454, 465
KDC 382
kerberized commands 202
kerberized commands do not encrypt the data 213
Kerberos 245
how it works 9
kerberos daemon 12
Kerberos master security server 12
Kerberos V4 5, 7, 12, 184, 186, 245, 249
AIX/PSSP integrated login 440
and DCE credentials 416
and root cache locations 424
cannot obtain credentials 346
database 169
default realm name 170
deleting credentials 425
hardmon ACL 328
hardmon ACL permissions 323
identity 269
integrated login with AIX/PSSP 270
managing hardmon ACL 328
miscellaneous remnant removal 168
not configured properly 243
removing credentials 425
synchronizing with DCE ACLs 319
sysctl ACL 291
user ID managing 270
Kerberos V4 encrypted authentication process 7
Kerberos V4 not configured properly 243
Kerberos V4 realm 7
Kerberos V5 5, 7, 12, 184, 245
Kerberos V5 improvements over Kerberos V4 7
Kerberos Version 4 50
Kerberos, internal workings of 9
key
private 204
public 204
key management service
See DCE, key management service
key pairs 214
key system, public and private 204
keyfile 30, 368, 370, 377
Control Workstation only PSSP DCE keyfiles 386
missing interface names in DCE self-host keyfile 412
recovering a PSSP DCE keyfile 383, 384
keyfile contains principal keys 30
keyfiles 385
Keyfiles for PSSP
See DCE, PSSP Keyfiles
keytab files, DCE
See DCE keytab files
kinit 12
klist 27
KRB5CCNAME 282, 418, 419, 439, 440, 454
KRBTOKFILE 282, 423, 440, 454
ksrvtgt 423, 440

L
lab environment
illustrated 58, 132
system diagram 132
latest security fixes 226
least secure partition 18
LED u95 91, 106, 129
level of security 6
limitations, RRA 186
list open files (lsof) 211
little paranoid 229
LoadLeveler 5, 33
local host settings 247, 248, 438
local-only 47
local-only configuration 114
log file monitoring 227
Log Files
/etc/security/failedlogin 233
/var/adm/sulog 233
syslog daemon log files 234
log files you should check 233
login sessions, encrypted 209

529
trol Workstation 385
using s1term to login to a node 461
node access 205
node cloning procedure with script.cust 74
node install customization 196
node installation, disconnecting from external net-
work 226
node, adding 239
node, deleting 239
node, re-installing 239
none 247, 248
nopolicy 372, 373
novel approach to intrusion detection 230
ntalk 204
NTP

O
offline backups 202
offline storage 209
one PSSP daemon appears to be hung 342
one root user prior to RRA 183
open environment 200
Open Software Foundation (OSF) 23
optional SP-administrator principal(s)
See DCE optional SP-administrator principal(s)
options for setting up DCE 49
OSF
See Open Software Foundation
other Kerberos V4 cleanup 168
our lab environment
See lab environment

P
packet filtering 209, 217, 218, 220
paging space, low 345
Parallel Environment 5
Parallel System Support Program (PSSP)
See PSSP
paranoid 229
parsing, filter rule 212
parsing, pe_site file 112
partially configured DCE client 408
partition level non-matching methods 256
partition level settings 247, 248, 438
partition name 251, 257
partition, default 32
partition, single 250, 259
partitioning 239
partitioning an SP 17
partitioning warning 18
partitioning, an important note 52
partitioning, security, and the CWS 52
partitions 216
partitions, different security policies in each 18
passphrase 215
password checking with crack 233
passwords, weak 233
patches 216
pcat 188
pcnfsd 204
pcop 232
pcp 188, 245
pdnzd 204
pe_site 381, 396
pe_site file parsing 112
pe_site, performance and load balancing with 112
Per Node Key Management 37, 40
See PNKM
Per Node Key Management daemon 38
Per Node Key Management log 38
Per Node Key Management, CWS example 37
Per Node Key Management, node example 37
performance impact with encryption 213
PERL 229
perl 232
perl-version of COPS 232
permitting all traffic with filter rules 210
pexec 188
pexescr 188
pfind 188
physical access controls
See Security servers, physical access controls
physical keyfile 31
pkill 189
planning considerations
Compatibility to DCE 134
Compatibility to Minimal 161
Minimal to Compatibility 170
pls 189
pmand 5
pmv 189
PNKM 368
and DCE attributes 368
DCE attributes that impact PNKM 369
factors that impact PNKM 370
using to diagnose problems 373, 375
POE 33

531
policy, defined  291, 327
port 2049, blocking NFS via  212
port 25  221
ports lower than 1024  209
    See trusted ports
ports, trusted
postfix  221
postfix (aka IBM Secure Mailer)  222
postfix and sendmail.cf  222
postfix and set-uid  222
postfix daemons  222
postfix does not use sendmail.cf  222
postfix, Wietse Venema  222
ppred  189
pps  189
Practical Unix and Internet Security  199, 201
Pre-DCE verification checklist  72, 113, 134
Pre-PSSP 3.2 default security state  66
Primary CDS server
    See DCE Primary CDS server
principal  485
principal keys, contained in keyfile  30
private and public keys for SSH servers  214
private key  486
private keys, generating with ssh-keygen  214
private/public key scheme with secure shell  214
prm  189
problem determination
    See Problem isolation and resolution
Problem isolation and resolution  237
    common periods for  238
    DCE client/server  390
Problem Management  5, 342
professional hackers  209
protecting the security registry  92, 114
PSSP
    and AIX remote commands  442
    authentication methods cannot be enabled  243
    DCE keyfiles were not created  243
    logs  238
    See PSSP 3.2
PSSP 3.2
    ACL control  272
    all services appear to be hung  342
    and AIX remote command debug tips/checklist  452
    and AIX remote commands  435
    authentication methods  246
    changing authentication methods  255, 257
Control Workstation only DCE keyfiles  386
DCE Masters cannot be on nodes  160
DCE not configured for it  375
DCE software dependencies  160
DCE version 3.1 filesets required on the CWS  76, 115, 137
differences with AIX authentication methods  245
disjoint authentication methods  250
expired DCE keyfile  376, 378
methods  247
missing DCE keyfile  377
one daemon appears to be hung  342
PSSP defined
purpose of the DCE cell  133
querying authentication methods  255, 257
recovering a PSSP DCE keyfile  383, 384
removal of DCE version2.x filesets  75, 137
running without Kerberos V4  161
security installation and configuration log  238
security installation, fastest method  60
security installation, good starting point  60
security installation, untrusted environment  61
security mechanisms supported by  7
simplest security state offered  60
verifying authentication methods  251
PSSP 3.2 and DCE software dependencies  160
PSSP ACL control  272
PSSP Authentication Methods  246
PSSP authentication methods cannot be enabled  243
PSSP DCE ACL object
    etc/logmgt.acl  296
    etc/sysctl.acl  296
    etc/sysctl.haem.acl  296
    etc/sysctl.install.acl  296
    etc/sysctl.mmcmd.acl  296
    etc/sysctl.pman.acl  296
    etc/sysctl.rootcmds.acl  296
    etc/sysctl.vsd.acl  296
PSSP DCE group, sdr-restricted  191
PSSP DCE groups
    See DCE, PSSP groups
PSSP DCE keyfile differences  385
PSSP DCE keyfiles  33, 34, 37
PSSP DCE keyfiles were not created  243
PSSP Keyfiles
    See DCE, PSSP Keyfiles
PSSP logs  238
PSSP partitions 32
PSSP Per Node Key Management 37
PSSP security installation and configuration log 238
PSSP services 192
pssp_script 60
psspfb_script 73, 91, 106, 129
pptest 189
PTFs 216, 226
public and private key system 204
public key 486
public key file, identity.pub 215
public keys, generating with ssh-keygen 214
public/private key pairs 214
purpose of the DCE cell in PSSP 3.2 133
pwdexpdate 368

Q
querying and managing hardmon ACLs 327
querying and managing sysctl ACLs 291
querying inetd status 344
querying the AIX methods 249, 438
querying the PSSP methods 247

R
randomized value 38
rc.dt 204
rc.nfs 204
rc.sp 225
rcmd principal 193
rcmd service tickets 177
rcmdtgt 423, 440
rcp 245
read permission for the world 216
read-only copies of DCE databases 25
read-only filesystem export 203
read-write copies of DCE databases 25
realm 7
recording passwords 207
recovering a PSSP DCE keyfile 383, 384
recovering failed DCE client config due to time skew 351
recovery DCE client from a full filesystem 348
reducing the security state 161
reflecting DCE group membership 277
refreshing DCE credentials 420
refreshing the DCE self-host credentials 273
Registry 38
registry 25
re-installing a node 239
relay host 222
relay host, do not use the CWS 222
remote command client and server must share a protocol 443
remote command execution 5
remote command facility 5
Remote Procedure Call (RPC)
See DCE, RPC
removing DCE credentials 421
removing Kerberos V4 161
removing Kerberos V4 credentials 425
removing security options from bottom up 17
replay attacks 485, 490
replica servers 92
replicating the DCE CDS namespace 108
required software not installed 258, 268
Research Center, Watson 222
restoring or installing a node in a safe way 225
restrict access to the system by setting up IP Security 210
restrict_root_rcmd 184, 190, 196, 197
Restricted Root Access
minimum required PSSP level 186
Restricted Root Access (RRA)
See RRA
restricting use of su command 205
restricting user access 205
restrictions, RRA 183
reverse name lookup 200
rexec 204
RFC 1510 485
risk, acceptable 3
rlogin 245
rlogin attribute 223
rmxcred 348, 422
robust dsh, example 310
root
and Kerberos V4 cache locations 424
root and Kerberos V4 cache locations 424
root umask settings 215
root user holding the DCE credentials 273
root.SPbgAdm 185
root@node_host_name 194
RPC_UNSUPPORTED_NETADDRS environment variable 49
RPC_UNSUPPORTED_NETIFS
non-existent interfaces 136
RPCTE_UNSUPPORTED_NETIFS environment variable 49
/etc/sysctl.conf file on CWS different than on nodes 185
_other_unauth warning 195
activation failure with VSD/GPFS 186
activation, immediate 191
additional security implications 189
AIX and PSSP remote commands 184
Boo t/Install Server changes to /etc/sysctl.conf 195
Boo t/Install Server on CWS 187
Boo t/Install Servers 187
c1_setup_kerberos 186, 187
coe xistence 186
credentials warning 191
customer entries not touched 191
Ecommands 188
enablement -- general steps 189
Event Management 194
firstboot.cust 196
GPFS not supported 14
GPFS warning 186, 189
HACMP 186
HACMP 4.3.1 186
HACMP workaround 187
HACMP/ES 4.3.0 186
HACWS 187
HACWS restrictions and limitations 187
how does it work 184
immediate activation 191
limitations 186
manual updating of kerberos database 187
migrations to PSSP 3.2 190
multiple Boot/Install Server recommendation 195
Multiple Boot/Install Servers 195
new PSSP installs 189
new root principal, steps to configure 193
new root principal, use of 193
node install customization 196
not a silver bullet 189
not recommended in none/std environment 195
one root user prior to 183
PSSP 3.2 node requirement 186
PSSP 3.2 requirements 189
recycling sysctld 195
remote commands, three scenarios 192
restrict_rootrcmd attribute 184
restrictions 183
root administrator principal exposure 193
root.SPbgAdm, global SP principal 185
SDR indicator 184
SDR Site Environment settings SDR Site Environ-
ment settings for RRA 196
security risk 193
security state Compatibility 192
security state DCE 191
security state Minimal 194
SP_Restricted class 184
spstenv 190
spstenv failure message 191
supper warning 185
supported security configurations 183
Switch Management 194
sysctl ACL files 185
sysctl, critical part 185
sysctld, recycling 195
system management commands 188
three instances when sysctl access required 194
troubleshooting tips 196
user management, adapting 189
VSD adapter warning 186, 189
VSD not supported 14
VSD/GPFS 186
when not to use 195
with Minimal security state 183
RRA considerations 14
RSCT 35, 37
RSCT services 34
RSCT subsystems 36
rsh 245, 262, 263, 264
rstatd 204
rules, filter 210
running Tiger 233
rusers 204
rusersd 204
rwall 204
rwalld 204
s1term 206, 225
examples of success or failure 323
using it to login to a node 461
illustrated 131
security strategies 4
security technology 6
choosing the right one for your environment 6
security terminology and concepts 11
security through confusion 230
security, IP 209
security, partitioning, and the CWS 52
self 26
self-host 394, 419, 439
missing interface names in DCE self-host keyfile
412
self-host principal 273
See DCE self-host principal
sendmail 205, 217, 221
sendmail configuration file 222
sendmail daemon 222
sendmail, port 25 221
sendmail, tcp-wrapped 217
server consolidation 183
service access groups 29
service principals 29
service ticket 486
Services
chargen 204
cmsd 204
daytime 204
discard 204
dpid2 205
dtspcd 204
echo 204
httpdlite 204
infod 204
ntalk 204
pcnfsd 204
pdnsd 204
rc.dt 204
rc.nfs 204
rexec 204
rstd 204
rusersd 204
rwalld 204
sendmail 205
snmpd 205
sprayd 204
time 204
services_config 215
session key 487
set up DCE so the PSSP daemons are properly au-
thenticated 218
setting up a DCE client on the CWS 114
setting up a DCE Security Replica Server on a node 110
setting up a Secondary CDS Server on a node 106
setting up filter rules 211
setting up filtering 210
setting up the CWS as DCE server 74
set-uid 222
setup_authent 189
setup_server 176, 215
setupdce 86, 100, 123, 241
setupdce did not run successfully 241
devcal 393
seven step security installation template 60
show.cfg 141, 241
Simple DCE cell
CWS as potential single point of failure 72
does not provide a highly available environment
72
Example 3 71
limited DCE environment 72
simplest security state offered 60
single partition 250, 259
SMTP protocol 222
SNG 210
sniffers 209
sniffing 6, 213
defined 7
SNMP Manager keyfile 34
snmpd 205
sockets 211
software dependencies 160
SP
identity 269
sp 312
SP administrator DCE principal, adding it to the DCE
cell 93
SP central point of management 5
SP files with world read permissions 216
SP partitioning 17
SP security attributes 14
SP security concepts and terminology 11
SP security design 4
SP security states 16
SP system
altering security configuration of 8
newly installed 8
recently migrated to PSSP 3.2 8
security beyond PSSP 9
troubleshooting security configuration problems 9
SP TaskGuides 14
SP Trusted Services 4, 5, 12, 15, 27, 245
defined 4
haem 5
hags 5
hardmon 4
hats 4
pmand 5
sdrd 4
sysctl 4
sp_admin@ 194
SP_NAME 251, 257, 454
SP_Restricted class 184
spacctnd 189
spacl 297, 308, 331
spacs_cntrl 205, 223
spauthconfig 91, 106, 129, 244
spbgroot 195
spbgroot principal 191
spchgv newObj 196
special considerations for high-security environment 244
special DCE cache locations 419
spgetdesc 189
spimage 227
split configuration 114
split configuration for DCE clients 47
spflstdata 247, 253, 438
spflstdata (-d, -h, -i options only) 189
spmkuser 189
spmkv newObj 196
spmon 193
spnkeyman_start 91, 105, 106, 128, 150
spnkeymand 91
spnkeymand -I 37
spoofing 6, 203, 206
defined 7
spray 204
sprayd 204
spsec_defaults 370
spsec_overrides 370
spsec-admin 29
spsec-services DCE organization 368
spsec-services organizational attributes 37
spstenv 184, 186, 190, 196
spstenv and RRA 190
spying, recording passwords 207
SSH 209, 214
ssh 208, 214
ssh client 208
SSH server private and public keys 214
sshd 214
ssh daemon 208
ssh-keygen 214
ssp.clients files 202
ssp/.../spbgroot principal 185
standard 195, 196
Standard AIX 7, 12, 50, 184, 245
standard commercial systems 201
start the IPSec device 212
start the IPSec filtering 212
startsrc -s sysctld 195
std 441
steady-state 238, 244
step-by-step procedure
  Compatibility to DCE 135
  Compatibility to Minimal 162
  Minimal to Compatibility 171
stopsrc -s sysctld 195
storage, offline 209
su 205
suggested reading for DCE 52
Sun RPC
  AUTH_DES 203
  AUTH_SYS 203
  AUTH_UNIX 203
Sun's Remote Procedure Call (RPC) protocol 203
supper 49, 203, 217, 223
suroot group 205
switch management commands 188
switchtbd-clean 9
switchtbd-load 29
switchtbd-status 29
symmetric encryption 486
synchronized time
  See DCE, synchronized time
synchronizing hardmon Kerberos V4 and DCE
ACLs 341
synchronizing sysctl Kerberos V4 and DCE ACLs
319
sysctl 4, 183, 247, 252, 279
  accessing online help 293
  ACLs 281
  authentication failure examples 281, 286
  authorization callbacks 280
things to consider when joining an existing DCE cell
three security states 16, 17
ticket cache 491
Ticket-Granting Service 485, 487
Ticket-Granting Ticket 486, 487
Tiger 227, 232, 233
Tiger configuration file, /usr/local/tiger/tigerrc 232
Tiger explanations, tigexp 233
Tiger, Doug Schales 232
Tiger, how often to run 233
Tiger, running 233
Tiger, TAMU's 232
tigexp 233
time 204
time master 25
time server 25
time skew too great 349
  example 350
Tool Control Language 185, 196, 279
Toolkit, Deception 230
tools and techniques, additional 9
Topology Services 4
trade-offs 3
traffic, allowing only one direction with IPsec 219
traffic, denying all with filter rules 210
traffic, permitting all with filter rules 210
transitional security state
  defined 131
do not bypass 135, 161, 171
Tripwire 217, 227
triplow 227, 228
tripwire configuration file 228
tripwire database 228
tripwire program file 228
troubleshooting
  See Problem isolation and resolution
Troubleshooting tips 196, 237
trust 3
target level 6
trusted 227
trusted ports 211
Trusted Services
  See SP Trusted Services
trusted shell 227
trusted third-party 485
TRY_PE_SITE environment variable 48
ts_auth_methods 14, 16, 17, 247
ts_auth_methods, defined 15
tunnel, encrypted and authenticated 214
  tunnels 210
tunnels, IP 213
tw.db_ 228
twenty-four hour cycle 38
twenty-four hour period 38
typical credentials states 426

U
UID 225
umask 022 215
umask 027 209, 215
umask 077 215
umask settings for root 215
SYSTEM=DCE OR DCE 223
Uniform Resource Locator
  See URL
unintentional exposures 203
Universal Coordinated Time (UTC) 25
updating DCE pe_site 396
updauthfiles 15, 184
URL
  Kerberos V5 8
user access on nodes 205
user access, restricting it 205
user identity and access control 268
user IDs
  managing 270
user.admin collection 49
using DCE as the only authentication method 225
using encrypted login sessions 213
using lssrc 344
using PNKM to diagnose problems 373, 375
using s1term to login to a node 461
using TCP Wrappers to restrict access to services 220
utp cabling 226

V
Venema, Wietse 220, 222, 229
verify secval is running 393
verify the pe_site file exists and is correct 396
viewing port ownership with Isol 211
VSD 186, 191
VSD adapters 186, 189
VSD not supported in RRA 14
W

Warnings
  accidentally disabling all logins 158, 168, 172
  backup tapes offline 202
  bypassing transitional security states 135, 161, 171
  create /var/dce as separate filesystem 72, 113, 136, 150
  DCE Masters cannot be on nodes 160
  DCE version 2.x fileset removal 75, 137
  disable unnecessary services 202, 204
  do not use the CWS as a relay host 222
  export filesystems read-only 203
  Minimal security in untrusted environment 61
  protect the security registry 92, 114
  reducing the security state 161
  Simple DCE cell with CWS as single point of failure 72
  unintentional exposures 203
  when not to use supper 217

Watson Research Center 222
WCOLL 251, 454
weak passwords 233
what is my identity? 269
whatacl 292
whatacls 297
which xauth 208
which xhost 207
whoami 290
Wietse Venema 220, 222, 229
working collective 251, 454
world read permission 216
wrap_test 188
write operations to the SDR fail 241

X

X client, defined 206
X server, defined 206
X Windows 206
X11 202
X11 connections, securing 206
X11.apps.config 206
xauth 207
xauth info 208
xauth list 208

Z

zero length authorization files 456
IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.

- Use the online Contact us review redbook form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

<table>
<thead>
<tr>
<th>Document Number</th>
<th>SG24-5521-00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redbook Title</td>
<td>Exploiting RS/6000 SP Security: Keeping It Safe</td>
</tr>
</tbody>
</table>

Review

What other subjects would you like to see IBM Redbooks address?

Please rate your overall satisfaction: O Very Good  O Good  O Average  O Poor

Please identify yourself as belonging to one of the following groups:

- O Customer
- O Business Partner
- O IBM, Lotus or Tivoli Employee
- O Solution Developer
- O None of the above

Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.

O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.

Questions about IBM’s privacy policy? The following link explains how we protect your personal information. [ibm.com/privacy/yourprivacy/](http://ibm.com/privacy/yourprivacy/)
Exploiting RS/6000 SP Security: Keeping It Safe
Exploiting RS/6000 SP Security: Keeping It Safe

Learn PSSP 3.2 security features step by step

Discover time-saving troubleshooting tips

Explore additional security tools

This IBM Redbook explains how to exploit the enhanced security features of PSSP 3.2. It walks you through installing, configuring, administering, and troubleshooting security technologies ranging from standard AIX, Kerberos 4, and DCE/Kerberos 5 to Restricted Root Access and beyond.

The PSSP security services were designed and implemented to minimize the complexities often associated with security features, especially during their installation and configuration phases. Given the nature of security, and considering its pervasiveness and global scope, it is critical to get it right. To that end, this redbook provides extensive coverage of problem determination and troubleshooting techniques to help you ensure that your implementation is sound.

In addition to the new PSSP 3.2 security features, this redbook discusses additional tools and techniques with which to enhance the security of your SP environment.