

# Simplify Management of Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments

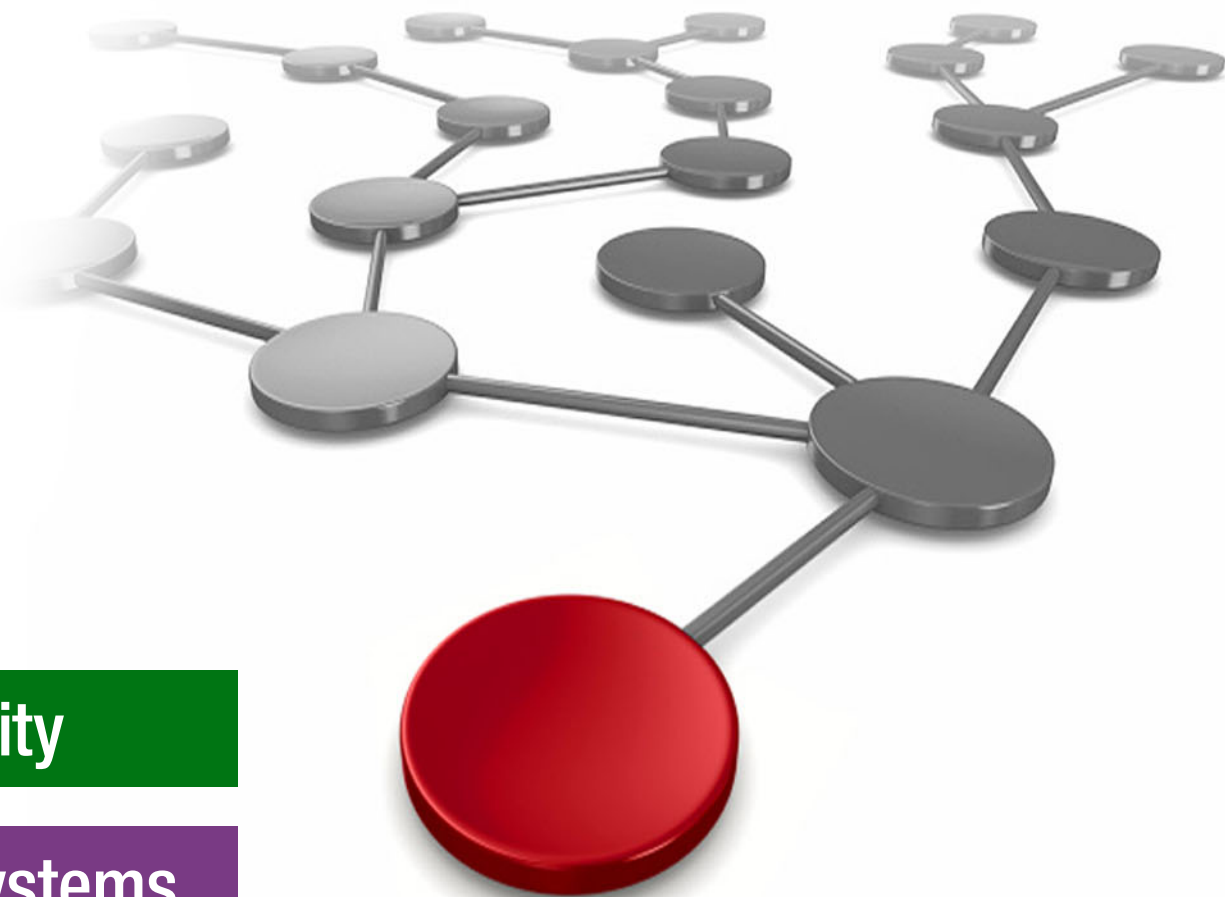
Dino Quintero

Faraz Ahmad

Stephen Dominguez

David Pontes

Cesar Rodriguez



**Security**

**Power Systems**







International Technical Support Organization

**Simplify Management of Security and Compliance with  
IBM PowerSC in Cloud and Virtualized Environments**

September 2019

**Note:** Before using this information and the product it supports, read the information in “Notices” on page ix.

## **Second Edition (September 2019)**

This edition applies to:

IBM PowerSC v1.2

IBM PowerSC Trusted Network Connect and Patch Management v1.2.0.0

IBM AIX v7.2

Red Hat Enterprise Linux Server release v7.4

**© Copyright International Business Machines Corporation 2019. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	ix
Trademarks .....	x
 <b>Preface</b> .....	 xi
Authors .....	xi
Now you can become a published author, too! .....	xii
Comments welcome .....	xii
Stay connected to IBM Redbooks .....	xiii
 <b>Chapter 1. IT security and Compliance Management</b> .....	 1
1.1 Business context for IT security .....	2
1.2 Influential factors for IT security .....	2
1.2.1 Business factors that influence security .....	3
1.2.2 IT factors that influence security .....	4
1.3 IBM Security Framework .....	7
1.4 IBM Security Blueprint .....	14
1.5 Security and Compliance Management .....	16
1.5.1 Audit reports .....	16
1.6 Summary .....	18
 <b>Chapter 2. IBM PowerSC GUI Server</b> .....	 19
2.1 Component architecture .....	20
2.2 Installing IBM PowerSC GUI server .....	21
2.2.1 AIX .....	21
2.2.2 Red Hat Enterprise Linux .....	26
2.2.3 SUSE Linux Enterprise Server .....	29
2.3 GUI administration .....	32
2.3.1 Endpoint administration .....	33
2.3.2 Manage users and groups .....	34
2.3.3 PowerSC GUI login .....	37
2.4 Installing the UIAgent .....	40
2.4.1 Installing UIAgent on AIX .....	40
2.4.2 Installing UIAgent on RHEL .....	42
2.5 Endpoint administration .....	43
2.5.1 Generate keystore .....	43
2.5.2 Security certificate expiration dates .....	46
2.5.3 IBM PowerVC integration .....	48
2.6 Managing groups in IBM PowerSC GUI .....	55
2.6.1 Creating groups .....	55
2.6.2 Renaming groups .....	58
2.6.3 Editing groups .....	58
2.6.4 Cloning groups .....	59
2.6.5 Deleting a group .....	59
2.7 IBM PowerSC GUI server features .....	60
2.7.1 Home tab .....	60
2.7.2 Compliance tab .....	60
2.7.3 Security tab .....	61
2.7.4 Reports tab .....	62
2.7.5 Profile Editor tab .....	73

<b>Chapter 3. Compliance automation</b>	75
3.1 IBM PowerSC compliance automation overview	76
3.1.1 Business challenge	76
3.1.2 Security and compliance automation concepts	76
3.2 Installation	77
3.2.1 Operating system prerequisites	77
3.3 Profiles	79
3.3.1 Payment Card Industry Data Security Standard (PCI) v3	79
3.3.2 General Data Protection Regulation	79
3.3.3 Test scenarios with GDPR and PCIV3	80
3.4 Applying a profile	80
3.4.1 Using the GUI	82
3.5 Checking compliance	85
3.5.1 Checking against applied profile	85
3.5.2 Simulate option	87
3.6 UNDO	91
3.7 Custom profile	92
3.8 Importing custom profiles not created with IBM PowerSC	103
3.9 Applying the PCIV3 profile to an AIX LPAR	106
3.9.1 Simulate first	108
<b>Chapter 4. Real-Time File Integrity Monitoring</b>	111
4.1 PowerSC Real-Time Compliance	112
4.1.1 Detailed implementation	113
4.1.2 Deployment considerations	114
4.1.3 Installation	115
4.1.4 Configuration steps	117
4.1.5 RTC configuration files	119
4.1.6 Adding a new file to RTC file monitoring list	124
4.1.7 Local logging	127
4.1.8 SNMP traps	127
4.1.9 RTC debug mode	128
4.2 AIX Trusted Execution	128
4.2.1 Components of Trusted Execution	129
4.2.2 Trusted Execution modes	131
4.2.3 Trusted Execution integration with PowerSC GUI	135
4.2.4 System integrity check with PowerSC GUI	137
4.2.5 Online Check with PowerSC GUI	139
4.2.6 TSD customization with PowerSC GUI	142
4.2.7 Best practice to enable TE in online mode	144
4.2.8 Updating an application that is integrated with TE	145
4.3 Linux auditd	147
4.3.1 Prerequisites	147
4.3.2 Configuration	149
4.3.3 Add file for monitoring	149
4.3.4 View FIM alerts	151
4.4 FIM reporting with PowerSC GUI	153
4.4.1 Dashboard view of FIM events	153
4.4.2 Reporting of FIM events	153
<b>Chapter 5. PowerSC Trusted Network Connect and Patch Management v1.2.0.0</b>	155
5.1 Introduction	156
5.2 Component architecture	157
5.3 Simplifying management of security and compliance by using TNC	160

5.4	Deployment considerations . . . . .	162
5.4.1	Disk and memory requirements . . . . .	162
5.4.2	Requirements to install software . . . . .	162
5.4.3	Host installation matrix for TNC components . . . . .	163
5.4.4	Syslog configuration . . . . .	164
5.5	Installing TNCPM . . . . .	164
5.5.1	Networking requirements for TNCPM internet connections . . . . .	164
5.5.2	Configuring the TNCPM . . . . .	165
5.5.3	Configuring the Trusted Network Connect Server . . . . .	174
5.5.4	Configuring the Trusted Network Connect Client . . . . .	181
5.5.5	Configuring Trusted Network Connect Server email. . . . .	182
5.6	Working with Trusted Network Connect and Patch Management. . . . .	182
5.6.1	Verifying the Trusted Network Connect Client . . . . .	182
5.6.2	Viewing the Trusted Network Connect Server logs. . . . .	189
5.6.3	Viewing the verification results of the TTNCCs. . . . .	189
5.6.4	Updating the Trusted Network Connect Client . . . . .	189
5.6.5	Updating and verifying by using PowerSC GUI 1.2.0.0 . . . . .	192
5.6.6	New TNC functions provided in PowerSC GUI 1.2.0.1. . . . .	195
5.6.7	Update logs. . . . .	196
5.7	Troubleshooting . . . . .	196
5.7.1	Check syslog. . . . .	196
5.7.2	Verify your configuration files . . . . .	196
5.7.3	Update operation fails while AIX Trusted Execution is enabled . . . . .	196
5.7.4	Refreshing the daemons to correct anomalies . . . . .	197
5.7.5	Enabling TNCS verbose logging. . . . .	197
5.7.6	More information. . . . .	197
	<b>Chapter 6. Trusted Logging . . . . .</b>	<b>199</b>
6.1	Component architecture . . . . .	200
6.1.1	Built on virtual SCSI foundations . . . . .	200
6.1.2	Virtual Log devices . . . . .	202
6.1.3	Virtual logs . . . . .	202
6.1.4	Virtual log directory and file structure . . . . .	204
6.1.5	Virtual log repositories . . . . .	206
6.1.6	Shared storage pools . . . . .	207
6.2	Deployment considerations. . . . .	208
6.2.1	Deploying Trusted Logging on a dedicated Virtual I/O Server . . . . .	208
6.2.2	Securing the Virtual I/O Server . . . . .	209
6.2.3	Local virtual logs or shared storage pools. . . . .	209
6.2.4	Where to store local virtual logs . . . . .	210
6.3	Detailed implementation . . . . .	211
6.3.1	Virtual log target devices. . . . .	211
6.3.2	Virtual log devices. . . . .	212
6.3.3	Messages that are written to the state files. . . . .	213
6.3.4	Multipath presentation on the client LPAR . . . . .	215
6.3.5	Workload partitions . . . . .	215
6.3.6	Performance . . . . .	215
6.4	Installation . . . . .	218
6.4.1	Installing the Client LPAR component. . . . .	218
6.4.2	Verifying the version of the Virtual I/O Server. . . . .	219
6.5	Working with Trusted Logging. . . . .	219
6.5.1	Changing the local virtual log repository file system . . . . .	219
6.5.2	Creating a virtual log on a single Virtual I/O Server . . . . .	220

6.5.3	Accessing virtual log data on the Virtual I/O Server . . . . .	220
6.5.4	Configuring shared storage pools . . . . .	222
6.5.5	Demonstrating multipath failover. . . . .	227
6.5.6	Configuring AIX auditing to use a virtual log . . . . .	230
6.5.7	Configuring syslog to use a virtual log . . . . .	232
6.5.8	Backing up Trusted Logging data on the Virtual I/O Server . . . . .	234
6.5.9	Deleting virtual logs and virtual log target devices . . . . .	245
6.6	Troubleshooting . . . . .	246
6.7	Conclusion . . . . .	250
<b>Chapter 7.</b>	<b>Trusted Boot . . . . .</b>	<b>251</b>
7.1	Overview . . . . .	252
7.2	Component architecture . . . . .	253
7.2.1	Trusted Boot technical overview . . . . .	253
7.3	Detailed implementation . . . . .	255
7.4	Installation . . . . .	256
7.4.1	Installing the collector . . . . .	257
7.4.2	Installing the verifier . . . . .	259
7.5	Working with Trusted Boot . . . . .	259
7.5.1	Configuring SSH . . . . .	259
7.5.2	Enabling Virtual Trusted Platform Module (vTPM) . . . . .	261
7.5.3	Enrolling a system. . . . .	262
7.5.4	Attesting a system. . . . .	263
7.5.5	Attesting multiple systems. . . . .	263
7.5.6	Simulating a failure . . . . .	264
7.6	Troubleshooting . . . . .	268
7.6.1	Common problems . . . . .	268
7.6.2	Diagnosis . . . . .	269
7.7	Conclusion . . . . .	272
<b>Chapter 8.</b>	<b>Trusted Firewall . . . . .</b>	<b>273</b>
8.1	Component architecture . . . . .	274
8.1.1	Firewall technologies . . . . .	274
8.1.2	Deny and permit . . . . .	275
8.1.3	Packet filtering rules . . . . .	276
8.1.4	Security policies . . . . .	276
8.2	Detailed implementation . . . . .	280
8.3	Deployment considerations. . . . .	288
8.4	Installation . . . . .	288
8.4.1	Trusted Firewall installation. . . . .	288
8.4.2	Verifying the Trusted Firewall installation . . . . .	292
8.5	Working with Trusted Firewall . . . . .	292
8.5.1	Configuring the Secure Virtual Machine . . . . .	292
8.5.2	Configuring the filter rules. . . . .	299
8.5.3	Removing Trusted Firewall . . . . .	311
8.6	Troubleshooting Trusted Firewall . . . . .	312
8.7	Conclusion . . . . .	313
<b>Appendix A.</b>	<b>Trusted Firewall addendum . . . . .</b>	<b>315</b>
	ICMP codes . . . . .	316
	ICMPv6 codes . . . . .	318
<b>Related publications</b>	<b>. . . . .</b>	<b>321</b>
IBM Redbooks	. . . . .	321

Online resources .....	321
Help from IBM .....	321





# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	POWER Hypervisor™	PowerVM®
FileNet®	Power Systems™	Redbooks®
IBM®	POWER7®	Redbooks (logo)  ®
IBM Watson®	PowerHA®	Watson™
POWER®	PowerSC™	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

This IBM® Redbooks® publication provides a security and compliance solution that is optimized for virtualized environments on IBM Power Systems™ servers, running IBM PowerVM® and IBM AIX®. Security control and compliance are some of the key components that are needed to defend the virtualized data center and cloud infrastructure against ever evolving new threats. The IBM business-driven approach to enterprise security that is used with solutions, such as IBM PowerSC™, makes IBM the premier security vendor in the market today.

The book explores, tests, and documents scenarios using IBM PowerSC that leverage IBM Power Systems servers architecture and software solutions from IBM to help defend the virtualized data center and cloud infrastructure against ever evolving new threats.

This publication helps IT and Security managers, architects, and consultants to strengthen their security and compliance posture in a virtualized environment running IBM PowerVM.

## Authors

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

**Dino Quintero** is an IT Management Consultant Project Leader and an IBM Level 3 Senior Certified IT Specialist with IBM Redbooks in Poughkeepsie, New York. Dino shares his technical computing passion and expertise by leading teams developing technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing, cloud computing, artificial intelligence (including machine and deep learning), and cognitive solutions. He also is a Certified Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist.

**Faraz Ahmad** is an IBM Power Systems solution architect working in IBM Lab Services, India. Faraz has over 15 years of experience in various areas of IT, including software development, solution designing, and IT consulting. He specializes in cyber security and in his current role, he designs security solutions for IBM customers. He is also a geography lead and mentors security consultants in the Central and Eastern Europe, Middle East, and Africa regions. His other areas of expertise includes IBM PowerHA®, AIX, Linux, Networking, and virtualization. He is author of multiple patents and recognized as an Invention Plateau holder. He has a degree in Computer Science from Birla Institute of Technology, Ranchi, India.

**Stephen Dominguez** is the worldwide AIX security lead for IBM Systems Lab Services. He has worked for IBM for over 20 years. He has been delivering AIX Security consulting services for 10 years. He spent his initial 10 years in IBM UNIX Product Test organization testing the HMC and AIX Security components. He is a Java certified programmer.

**David Pontes** has been working for more than 20 years at IBM, where he has worked in several areas, from support, management, and transformation projects. David has worked for the past seven years in security for Power Systems, and part of those years in the role of consultant for the IBM Lab Services Cloud Team.

**Cesar Rodriguez** is a Member of IBM Academy of Technology and a Master Inventor who works at the IBM Cyber Security Project Office. He has submitted more than 50 patent applications to the US Patent Office. His patents cover several topics, including Cyber Security, Cognitive Computing, IoT, and autonomous vehicles. He also holds several certifications in Project Management including PMP, Scrum Master, Scrum Developer, Scrum Product Owner, Agile Expert, and Scrum Trainer. Additionally, Cesar holds a Master of Science degree in Cyber Security that he received with the highest Honors (Summa Cum Laude). He also teaches Cyber Security in several universities at the bachelor's and Master's degree levels.

Thanks to the following people for their contributions to this project:

Wade Wallace  
IBM Redbooks, Austin Center

Xiohan Qin  
IBM USA

Petra Buehrer  
IBM Germany

Tim Hill  
Rocket Software

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099

2455 South Road  
Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:  
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:  
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:  
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:  
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:  
<http://www.redbooks.ibm.com/rss.html>





# IT security and Compliance Management

This chapter describes the overall business context for information technology (IT) security and Compliance Management. This chapter examines the factors that influence why and how IT security and Compliance Management must be conducted in a certain business context, and as an introduction to the IBM Security Framework and the IBM Security Blueprint.

This chapter also describes the business requirements for an IT security and Compliance Management solution.

This chapter contains the following sections:

- ▶ Business context for IT security
- ▶ Influential factors for IT security
- ▶ IBM Security Framework
- ▶ IBM Security Blueprint
- ▶ Security and Compliance Management
- ▶ Summary

## 1.1 Business context for IT security

Organizations rely on information security systems more than ever to prevent threats to intellectual property, corporate sensitive information, Sensitive Personal Information (SPI), damage to the reputation, and privacy.

Organizations often adopt a business, compliance, or monetary-driven approach to IT security. Using this approach alone does not provide sufficient protection for business processes and assets against the increasing number of IT systems risks because it might overlook key, cross-discipline aspects.

As the pace of globalization continues and new technologies emerge, traditional boundaries between organizations continue to disappear. The ideal response involves planning and assessment to identify risks across key business areas, including people, processes, data, and technology throughout the entire organization. It is important to take a holistic approach that can facilitate a business and compliance-driven security strategy that can act as an effective defense for the entire organizational data and systems.

Organizations must build business processes, policies, and services that are *secure by design*, meaning that security is intrinsic to their business processes, product development, and daily operations. Security must be factored into the initial design, not bolted on afterward.

Additionally, companies must adopt a policy of *compliance by design* to ensure that all services and IT systems that are supporting the operations are aligned with the required regulatory requirements.

This approach enables an organization to securely and safely adopt new forms of technology, such as cloud computing and mobile device management, and business models, such as telecommuting and outsourcing, which can be more safely used for cost benefit, innovation, and shorter time to market.

## 1.2 Influential factors for IT security

Nowadays, most companies or organizations require some kind of IT infrastructure to efficiently run their business. This brings a new set of risks to the companies that are related to that IT infrastructure. The following main factors influence those risks:

- Business factors

Business factors measure value, risk, and economic costs that influence an organization's approach to IT security. The value determines the worth of IT asset systems to the business. Risk involves compliance, corporate structure, corporate image, and the risk appetite and tolerance of the company. Economic determines productivity impact, competitive advantage, and IT systems cost.

Business factors also represent issues and consequences that are of significance to the stakeholders of the managed IT systems. This set of factors might vary from industry to industry, from organization to organization in the same industry, and even from different business units in an organization.

- IT factors

IT factors represent operational constraints in the general IT environment. For example, the complexity of an IT system, including its environment, that is exposed to internal and external threats, which creates risks that the organization must address.



IT factors also represent technical considerations that affect the trustworthiness of the IT systems and likely the IT environment as a whole. The combination of business and IT factors represents the foundation for security management.

### **1.2.1 Business factors that influence security**

Business factors represent a relationship between the IT organization and the rest of the business. This reference is about the business factors that must be supported by IT systems and therefore must be secured to reduce the associated risks.

#### **Correct and reliable operation**

Correct and reliable operation is the degree to which the business must be accurate and consistent in its operation. Correct operation means that the operations perform the proper response or function with no errors. Reliable means that the same result occurs all of the time. Any IT system must consistently provide stakeholders with the expected results.

Security events and incidents might impact the correct and reliable operation of these business processes. It might also affect the underlying IT infrastructure or upstream and downstream business processes. The consequences of a defective service (incorrect or varying results over time) might be significant to the consumer of the service, and therefore to the provider of the service.

In addition to affecting the correct and reliable operation of the business processes, security capabilities themselves must adhere to metrics around correct and reliable operation. Logs and reports must be correct and reliable, security features must keep the number of false positives and false negatives to a minimum, and security software and appliances must strive for defect-free operation with a mean time-to-failure that meets an organization's requirements.

#### **Service-level agreements**

This factor applies to circumstances where security threats and threat agents can impact the availability of IT systems and therefore the organization's ability to conduct business or provide services. Service-level agreements (SLAs) define acceptable conditions of operation in an organization or service. SLAs might vary from IT system to IT system or from application to application.

The availability of IT systems, data, and processes are conditions that are commonly referenced in SLAs. SLAs are also commonly associated to contractual penalties. Therefore, companies must concentrate efforts to ensure the optimal availability of their IT systems.

#### **IT system value**

From a business perspective, the IT system value directly relates to the value of the business transactions that it supports. These IT system values might be tangible or intangible. For an e-retailer, these IT system values are tangible assets.

For a financial services company, the asset might be the client information or other data that is used in transactions of the system. Another important consideration is that intangible assets are sometimes not clearly identified, and these assets are normally the most valuable assets of the company.

#### **Protection of the IT systems value and brand image**

All companies must invest in performing the required efforts to protect their image and reputation. A security incident or attack likely has a direct effect on the company, their reputation, and value.

Therefore, the security measures and investment are likely to be proportional to the consequences. Reputation damage that is caused by an IT security incident is not reversible (and sometimes fatal); therefore, preventing it must be a corporate priority to minimize that risk.

### **Legal and regulatory compliance**

Legal and regulatory compliance is a set of externally imposed rules or guidelines that must be followed by an organization. Violations of legal and regulatory compliance can result in legal consequences, including federal fines. Because of the increasing number of regulations and the associated complexity, organizations must find a system that enables a simplified way to apply, deploy, enforce, manage, and update those regulations across their IT systems.

### **Contractual obligation**

Security and risk management for IT system are likely to be proportional to the consequences that are can occur when the business encounters contractual liability from a security incident. For example, a security incident that affects the availability of a service can prevent an organization from complying with some contractual obligations, which is typically tied to some penalties and sanctions.

### **Financial loss and liability**

A security incident likely causes a direct or indirect financial loss to the organization. Examples of direct loss include theft of intellectual capital, theft of clients, theft of data, and impact to the availability of services, processes, billing systems, and fraud.

Some examples of indirect loss include civil or criminal process, fines, penalties, loss of credibility, and damage to the corporate image.

The investment in security mechanisms that enhance, standardize, and facilitate the security and compliance of IT systems must be proportional to the consequences if any of these risks are materialized.

### **Critical infrastructure**

Critical infrastructure is associated to the systems that support a process or service. Typically, any impact to those systems affects the capacity to provide services to all or most of the users. Examples include telecommunications, electrical power grids, transportation systems, and computer networks.

The loss of the IT systems that support the critical infrastructure likely has a ripple effect, which causes secondary losses and increases the impact to the organization. The identification of the systems that support the critical infrastructure is key during the *Risk Analysis process* because of the effect of the related incidents.

### **Safety and survival**

Security incidents likely have a major affect on human life, government function, and socioeconomic systems. For example, the affect on the operations of a hospital, medical center, public infrastructure, and public services that are caused by a security incident on the IT systems can cause a negative or fatal impact on public safety and health.

## **1.2.2 IT factors that influence security**

In this section, we provide an overview of a group of IT factors that must be considered during the risk analysis process to reduce the risks and consequences associated.

## **Internal threats and threat agents**

Security-related incidents on corporate IT systems can be caused by Internal threats and threat agents that are found within the physical and logical boundaries of the organization or enterprise and these threats and threat agents are normally related to people.

An example of an internal threat agent is a person who uses their ability or influence to access an IT system to carry out a malicious activity.

## **External threats and threat agents**

External threats and threat agents that are outside the physical and logical boundaries of the organization or company can also trigger or cause IT security incidents affecting the IT systems.

*External threats* are single points of failure that are outside the organization boundaries, including a power system grid or the outside internet connection.

An example of an *external threat agent* is a hacker or a squirrel that accidentally cause a disruption on the power or communications cables.

These threats and threat agents likely affect any of the three main security components: Confidentiality, Integrity, and Availability.

## **IT Service Management**

The incorrect management or operation of the IT systems likely affect the system's availability, which introduces several risks to the business that if materialized can result in financial losses.

The organization must ensure that service delivery commitments are achieved to reduce the risk of not meeting an SLAm, which can lead to a penalty.

For example, an attacker can target a company with a DDos attack that is aimed to prevent the company from achieving an SLA by affecting the availability of a system.

A simplified management of the IT systems often improves the two pillars of IT Service Management: Service Support (Manage of the Incident, Problem, and Change process), and Service Delivery (SLAs, Disaster Recovery, and Availability Management).

A company that properly implement and manages IT Service Management drastically reduces the risks that are associated with the business and IT security.

## **IT environment complexity**

The complexity of the IT environment increases or decreases the risk over the IT systems and their data. The IT environment reflects the infrastructure on which the IT systems are placed, including systems, networks, the policies and procedures that are associated, and others.

For example, most IT environments often must be aligned with a plurality of regulations across all their systems and the complexity that is associated to achieve the required level of compliance is exponentially associated to the complexity of the environment.

## **Business environment complexity**

The business can include another set of complexity factors because of the market, regulations, and others. Also, the management system, hierarchy, and the governance model that is used by the company can contribute to the complexity of the business, which adds a set of related risks and complexity to the IT systems.

## **Audit and traceability**

IT audits are a key element on IT security. Logs are important because they enable the traceability that is required to support the audits. This also supports several important policies that are related to IT security, such as separation of duties, and are a vital component to support other IT security processes, such as forensics and compliance.

## **IT vulnerabilities**

In this section, we describe several important IT vulnerabilities.

### ***Configuration***

Misconfiguring IT systems can produce many vulnerabilities on the IT systems. Therefore, companies must assign the required resource to ensure the optimal configuration of the systems.

Additionally, the use of a system to standardize and simplify the configuration of IT systems typically reduces the risks that are associated and at the same time improves the use of resources that are needed to support it.

### ***Flaws***

A flaw on IT systems must be considered at least as a high probability risk that can be used by an attacker to damage the confidentiality, integrity, and availability of the corporate data.

The biggest danger of those flaws is when the manufacturer is unaware of the flaw and the attacker can use it freely until a patch or work-around is designed, developed, and applied. Those vulnerabilities are known as *Zero-day Vulnerabilities* and are one of the biggest threats on IT systems.

After the patch is available, companies must ensure that it is deployed and applied as soon as possible to reduce the risk of an attack. Therefore, organizations must have a reliable, standard, and robust system to distribute and deploy those patches securely.

### ***Exploits***

An exploit is commonly referred to as a known script or vulnerability that is used to attack a set of IT systems.

Sometimes, exploits take advantage of some features or functions within a system to escalate permissions, perform unauthorized actions, disrupt the functions of the system, and many other attacks.

Next, we introduce the IBM Security Framework that was created to help organizations with their security challenges.

The main goal of the IBM Security Framework is to create a bridge to address the communication gap between the business and technical perspectives of security to enable simplification of thought and process.

## 1.3 IBM Security Framework

Today, business initiatives often are guided by the principles of Governance, Risk, and Compliance (GRC). However, these terms are broad and often have different meanings to different stakeholders in an organization.

Each chief experience officer (CXO) often attempts to mitigate risks for their division's domain; therefore, they have different priorities and points of view when it comes to risk management, including the following examples:

- ▶ The Chief Risk Officer (CRO) looks at the organization's overall risk profile and where the organization is most vulnerable to an unexpected loss.
- ▶ The Chief Financial Officer (CFO) must ensure that the necessary controls are in place to have accurate financial statements.
- ▶ The Chief Information Security Officer (CISO) ensures that the IT infrastructure and systems support the overall business and the organization. The CISO must minimize the risk of the IT environment and IT systems. Additionally, the CISO must assess and communicate the effect of those risks to the overall organization from a GRC perspective.

Regardless of the organizational perspective of risk management, processes and IT controls must be established and aligned to obtain a complete picture of the organization's *risk posture*.

Additionally, the establishment of IT security systems to control, monitor, and mitigate the risks are critical capabilities for any IT department.

IBM created the IBM Security Framework (see Figure 1-1 on page 8) to help ensure that every IT security aspect can be properly addressed when a holistic approach is used with business-driven security.

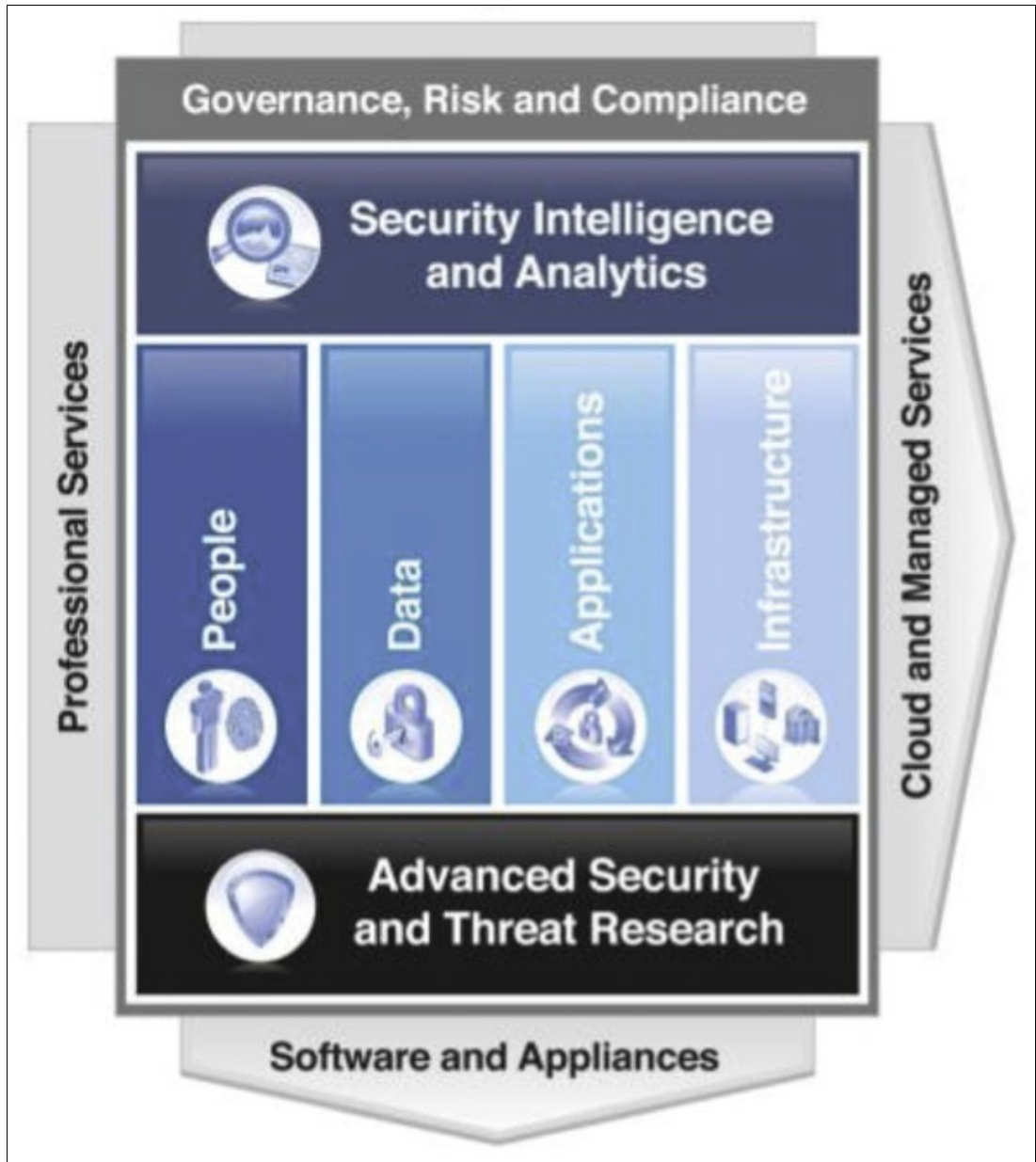


Figure 1-1 IBM Security Framework

The capabilities that are described by the IBM Security Framework are based on Security Intelligence and Threat Research capabilities.

Additionally, the solutions that are provided within the *security domains* and more layers can be delivered through software, hardware (appliances), or as Managed Services or Cloud offerings.

The IBM Security Framework is a layered model that is composed of the following domains:

► Infrastructure

IT systems must be secured to be aligned with all required regulations to achieve the level of compliance that is required by the organization. Therefore, a robust Security system is required to support the organization's IT systems to stay ahead of emerging threats that can adversely affect system components, the people, and business processes that they support.

Organizations are increasingly using virtualization technology to support their goals of delivering services in less time, with greater agility, and at lower cost. However, those environments must preemptively and proactively monitor the operation of the IT systems infrastructure while looking for threats and vulnerabilities to avoid or reduce the probability of security breaches. This domain covers securing that IT system infrastructure against all emerging threats (see Figure 1-2).

**Important:** IBM PowerSC is a robust, integrated, and simplified standard solution that supports organizations to increase their security and simplify compliance for the organization.

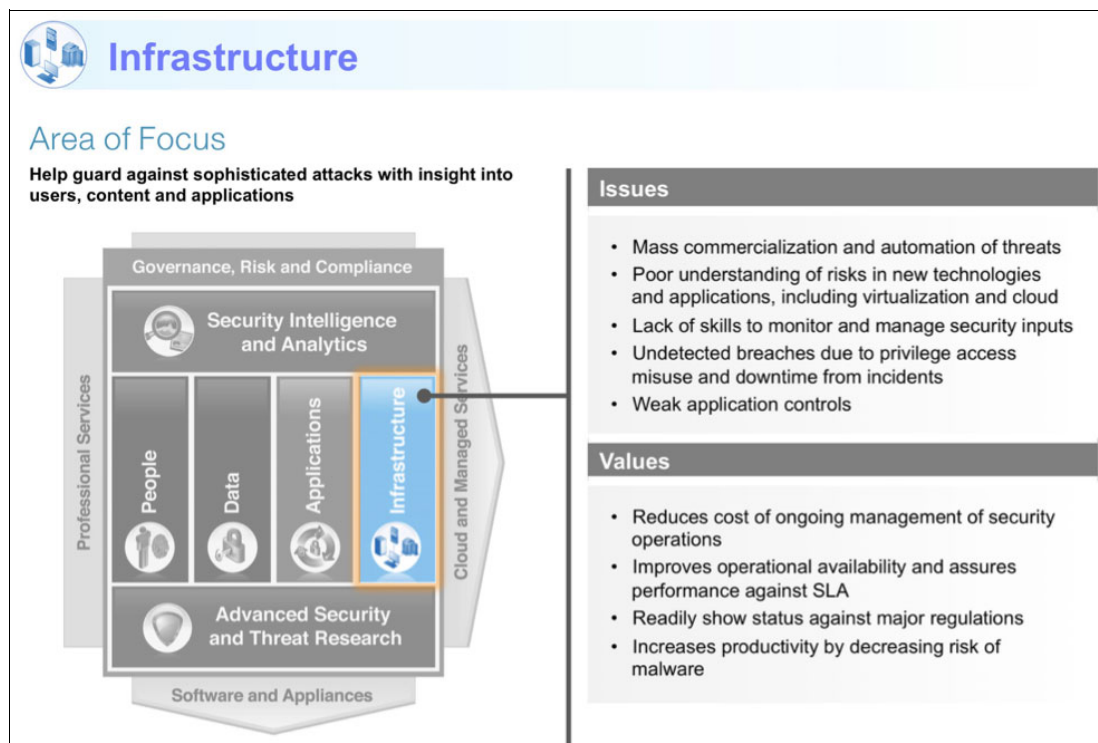


Figure 1-2 Summary and other aspects to be addressed within the infrastructure domain

► Advanced Security and Threat Research

Because the threat landscape continues to evolve and attacks continue to grow in number and complexity, an advanced research solution is required to stay ahead of the threats, as shown in Figure 1-3.

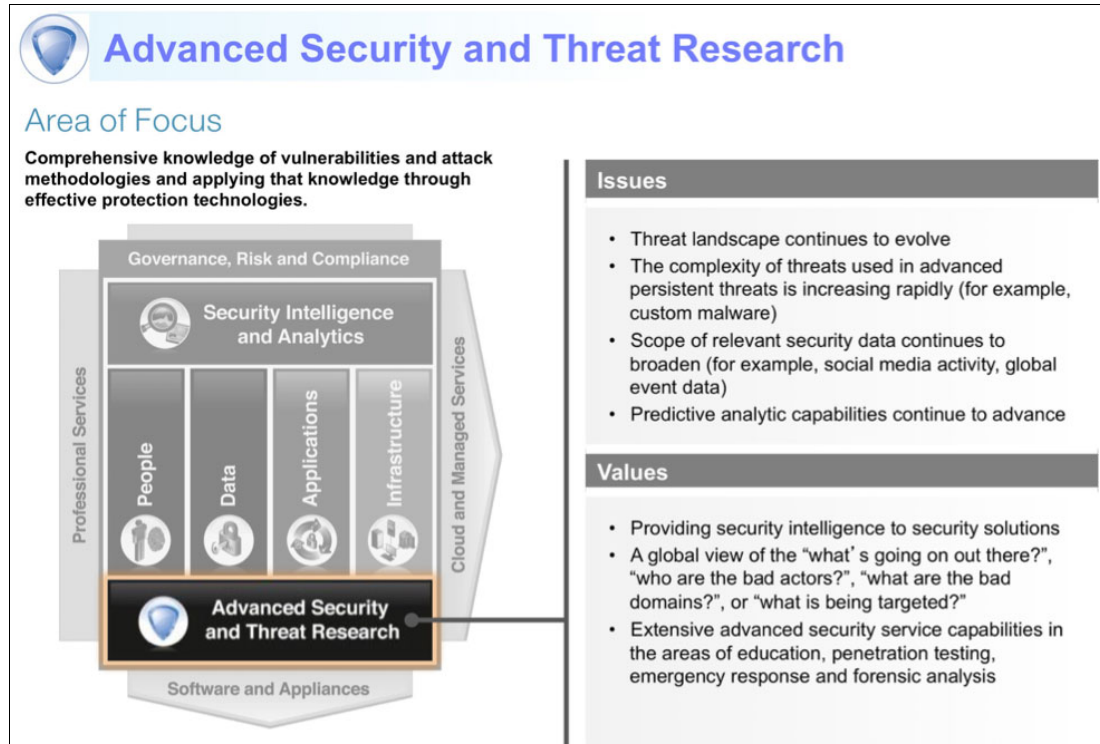


Figure 1-3 Advanced Security and Threat Research domain



► People

As shown in Figure 1-4, this domain covers aspects about how to ensure that the correct people have access to the correct assets at the correct time, which is known as *identity management and access control*.

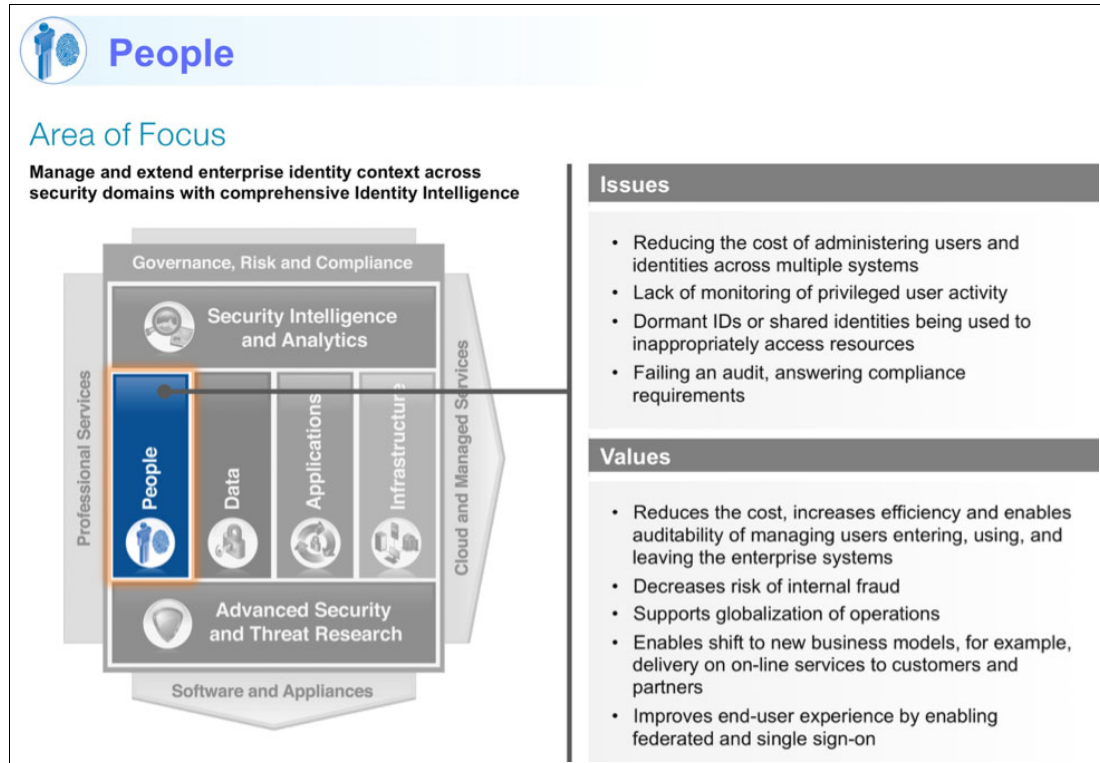


Figure 1-4 People domain

► Data

Data is one of the most valuable assets for organizations and this domain (see Figure 1-5) covers aspects about how to protect critical data in transit, in use, or at rest across the organization.

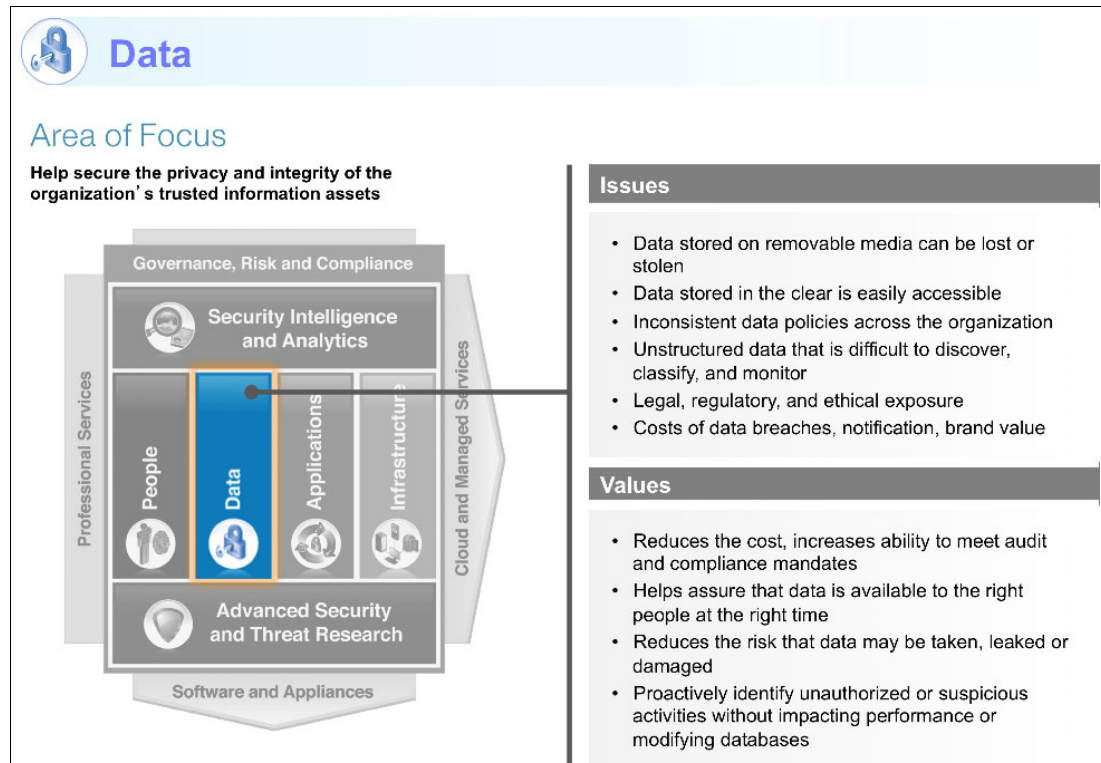


Figure 1-5 Data domain

► Applications

One of the most effective ways to avoid a breach is by ensuring that the applications your users are accessing are designed and implemented securely (see Figure 1-6 on page 13). Therefore, organizations must proactively protect their business-critical applications from external and internal threats throughout their entire lifecycle, from design to development, test, and production.

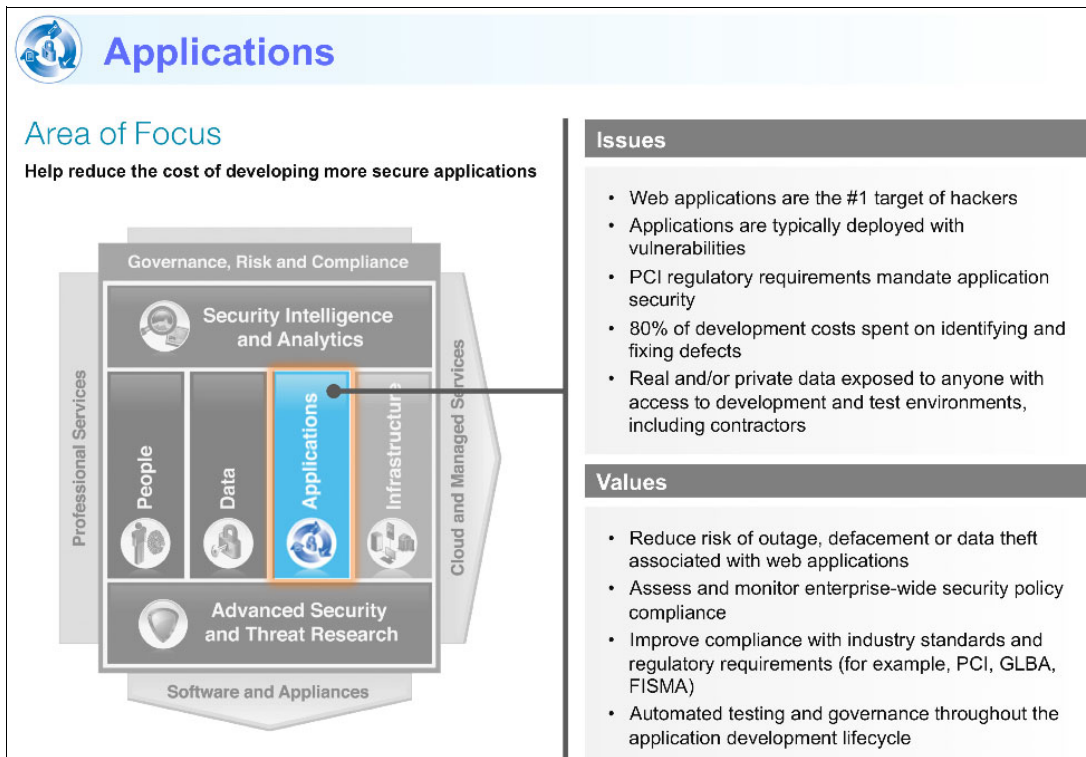


Figure 1-6 Application and Process domain

#### ► Security Intelligence and Analytics

Security Intelligence and Analytics provides a layer of discovery and reporting on top of the security domains. This layer is achieved with a control center for logging, viewing, analyzing, alerting, and reporting on events *across*, rather than *within* domains (see Figure 1-7 on page 14).

It provides a unified system for collection, aggregation, and analysis of application logs, security events, vulnerability data, identity and access management data, configuration files, and network information from security applications and devices throughout the security domains. In addition, the Security Intelligence and Analytics layer provides a common platform for all searching, filtering, rule writing, and reporting functions, including a user interface for log management, risk modeling, vulnerability prioritization, incident detection, and impact analysis tasks.

The comprehensive nature of the security intelligence architecture shows the vast amount of real-time and historic data for alerting and forensic analysis. The reduction of potentially billions of items of security data into a manageable set of actionable items, and the identification of patterns of behavior, are the foundation of the analytics within this layer.

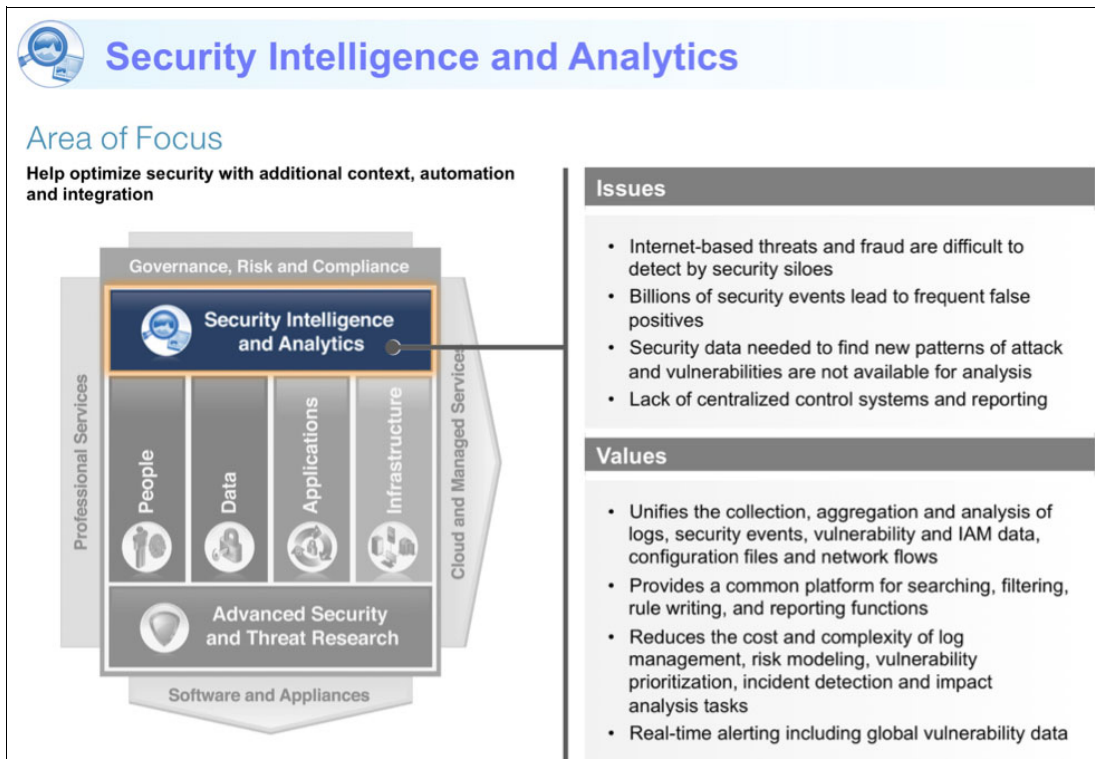


Figure 1-7 Security Intelligence and Analytics layer

**Note:** IBM provides the full breadth and depth of solutions and services that enable organizations to take this business-driven, secure-by-design approach to security that aligns with the IBM Security Framework.

## 1.4 IBM Security Blueprint

The IBM Security Framework divides the area of business-oriented IT security into four major security domains and three support layers.

Next, these domains and layers are broken down into greater detail to work toward a common set of core security capabilities that are needed to help your organization securely achieve its business goals. These core security capabilities are called the *IBM Security Blueprint*.

The IBM Security Blueprint uses a product-agnostic and solution-agnostic approach to categorize and define security capabilities and services that are required to answer the business concerns in the IBM Security Framework.

The IBM Security Blueprint was created after researching many client-related scenarios, which focused on how to build IT solutions. The intention of the blueprint is to support and assist in designing and deploying security solutions in your organization.

Building a specific solution requires a specific architecture, design, and implementation. The IBM Security Blueprint can help you evaluate those aspects, but does not replace them. Using the IBM Security Blueprint in this way can provide a solid approach to consider the security capabilities in an architecture or solution.

IBM chose to use a high-level service-oriented perspective for the blueprint, which is based on the IBM service-oriented architecture (SOA) approach.

Services use and refine other services (for example, policy and access control components affect almost every other infrastructure component).

The IBM Security Blueprint is shown in Figure 1-8.

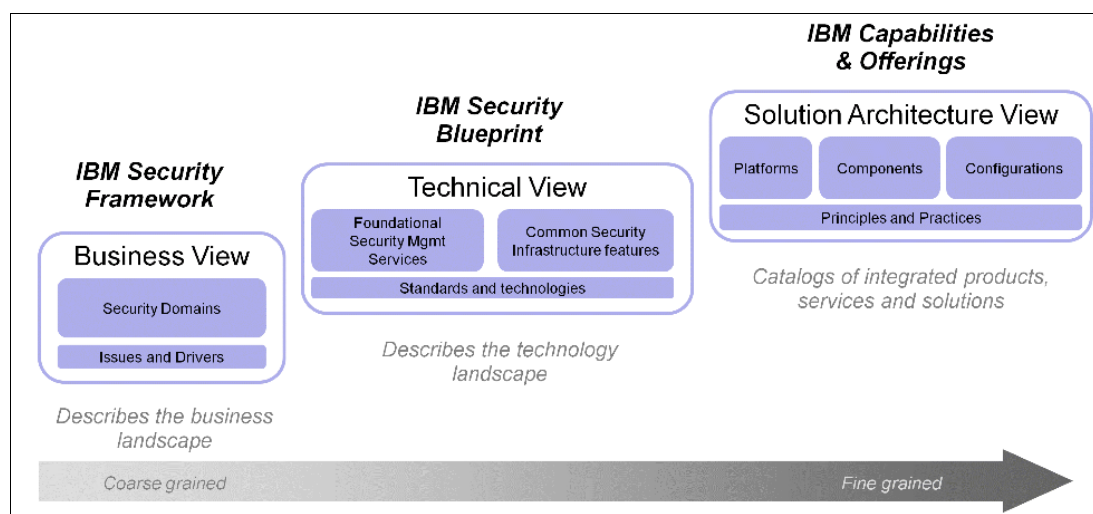


Figure 1-8 IBM Security Blueprint positioning

The left portion of Figure 1-8 represents the IBM Security Framework, which describes and defines the security domains from a business perspective.

The middle portion in Figure 1-8 represents the IBM Security Blueprint, which describes the IT security management and IT security infrastructure capabilities that are needed in an organization.

The capabilities that are described in the IBM Security Blueprint are presented in product and vendor-neutral terms.

The right portion of Figure 1-8 represents the solution architecture views, which describe specific deployment guidance that is related to a specific IT environment and the current maturity of the organization in the security domains. The solution architecture views also provide details about specific products, solutions, and their interactions.

Figure 1-9 on page 16 shows the components and subcomponents of the IBM Security Blueprint that must be examined for every solution in the Infrastructure security domain. In addition to the Foundational Security Management Services, you can use the IBM Security Blueprint to determine the Security Services and Infrastructure components by reviewing the component catalogs for these Foundational Security Management Services. Each of these components can then be assessed by determining whether each infrastructure component is required to make a Foundational Security Management service functional so that it can address the issues or provide a value that is associated with the particular business security domain (in this case, Infrastructure).

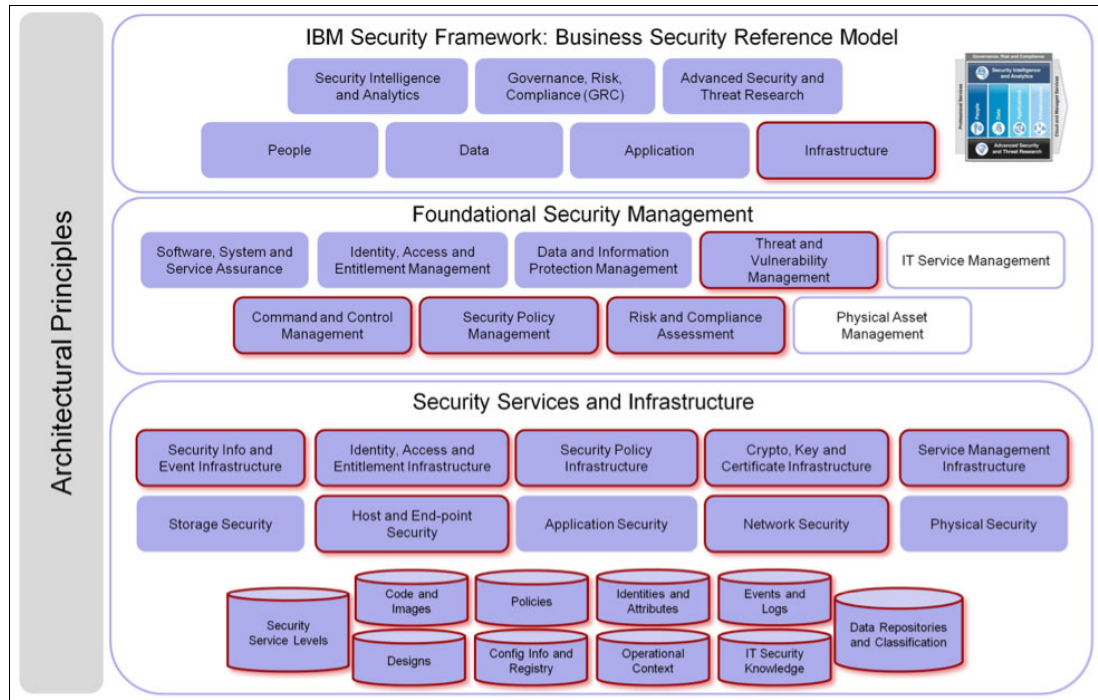


Figure 1-9 IBM Security Blueprint components for the infrastructure solution pattern

For more information about the IBM Security Framework, IBM Security Blueprint, and all related security components and domains (including infrastructure), see *Using the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, [SG24-8100](#).

## 1.5 Security and Compliance Management

Compliance Management covers all activities that are related to the alignment of the operation to a standard as mandated by internal policies, external regulations, and laws such as the following examples:

- ▶ Payment Card Industry Data Security Standard (PCI DSS)
- ▶ Sarbanes-Oxley
- ▶ Health Insurance Portability and Accountability Act (HIPAA)
- ▶ Security Technical Implementation Guide for the Department of Defense
- ▶ COBIT best practices

### 1.5.1 Audit reports

Audit reports help to identify, document, and verify the level of compliance to any internal policy, external regulation, or applicable law.

The management of virtual machines on the enterprise domain is a complex task that likely requires different configurations and settings. Also, more configuration can be required on each machine to ensure alignment with a set of regulations. Therefore, the efficient gathering of audit reports from that plurality (and variety) of IT systems often results on a complex task.

Enforcing the configuration for a particular virtual machine requires a flexible monitoring mechanism that constantly evaluates the state of these settings. This mechanism must identify current patches and security configuration levels, along with hardware configuration, and compare them against defined policies. It can then produce a high-level picture of the infrastructure through reports and graphics that identifies gaps that might exist in the security and compliance of this IT infrastructure.

**Compliance versus control:** If you are audited (or if you audited someone else), you probably know that a difference exists between being in compliance and being in control. Consider the following points:

- ▶ When you are in compliance, all of your systems and processes are operated and delivered according to the security policies and standards (and you have evidence for compliance).
- ▶ When you are in control, you know what is in compliance and what is not, you know why, and you create an action plan (and you have evidence for control).

Now, which is more important? Being in control is more important because you can be in compliance by accident. Furthermore, if you are compliant but not in control, chances are high that you cannot stay compliant for long.

If you are in control, you are compliant eventually, or at least you have it on record why you are not compliant.

In addition, if you are not compliant and not in control, gaining control must be your primary goal, which is why more often regulations shift from compliance to control objectives.

Addressing the security needs of virtual machines must be a holistic approach that starts with gaining visibility into these virtual machines within the infrastructure. The saying that “you cannot manage what you cannot see” can be translated in this context as “you cannot secure and control what you cannot see”. To properly remediate vulnerabilities and enforce configurations, you must first know which virtual machines are at risk and which regulations, policies, and restrictions are applied on each of them.

Many failed audits result from poor security management of server vulnerabilities because of configuration drift, or the inability to rapidly deploy (and confirm) the application of patches and updates, security settings, and virtual machine misconfiguration.

A unified solution that incorporates the capability to detect and identify audit gaps at a specific time can help organizations move to a unified management approach, which enhances visibility, management, and control. Also, this process enables the link between the establishment of security strategy and policy, execution of that strategy, real-time reporting, and security and compliance reporting.

Virtual machine security and Compliance Management are vital to IT security management. The ideal response involves a level of planning and assessment to identify risks across key business areas, including people, processes, data, and technology throughout the entire business. It is important to plan a holistic approach that can facilitate a business-driven security blueprint and strategy that can act as an effective shield of defense for the entire organization.

We think that organizations must build services that are *secure by design*, meaning that security is intrinsic to their business processes, product development, and daily operations. It means that this is factored from the initial design and not added afterward. This methodology allows an organization to securely and safely adopt new forms of technology (innovation) that runs on new virtual machines that are in cloud computing.



## 1.6 Summary

In this chapter, we examined the business and technology factors that influence security in organizations. We then examined the approach and tools that are available to consider a holistic approach to secure the IT operations of an organization.

Then, we described how the IBM Security Framework and IBM Security Blueprint can help avoid misalignment of priorities between IT and the business strategy. These tools aim to ensure that every necessary IT security domain is properly addressed when you take a holistic approach to business-driven security.

Finally, per the scope of this book, we highlighted the importance of security and Compliance Management on virtualized environments.





# IBM PowerSC GUI Server

In this chapter, we describe the main steps that must be performed and the options that are available to install the IBM PowerSC GUI server, administer the account, and work with compliance levels and profiles.

This chapter contains the following sections:

- ▶ 2.1, “Component architecture” on page 20
- ▶ 2.2, “Installing IBM PowerSC GUI server” on page 21
- ▶ 2.3, “GUI administration” on page 32
- ▶ 2.4, “Installing the UIAgent” on page 40
- ▶ 2.5, “Endpoint administration” on page 43
- ▶ 2.6, “Managing groups in IBM PowerSC GUI” on page 55
- ▶ 2.7, “IBM PowerSC GUI server features” on page 60

## 2.1 Component architecture

IBM PowerSC GUI Server was introduced in IBM PowerSC version 1.1.5. The UI Server made a significant difference in the usability of IBM PowerSC features. Feature enhancements are available in every new release of IBM PowerSC. IBM PowerSC 1.2 can also manage Linux endpoints in addition to AIX endpoints.

The IBM PowerSC GUI provides a centralized management console for visualization of endpoints and their status; applying, undoing, or checking compliance levels; grouping systems for the application of compliance level actions; and viewing and customizing compliance configuration profiles. The IBM PowerSC GUI also provides extensive profile editing and reporting capabilities.

The GUI interface also includes File Integrity Monitoring (FIM). FIM includes Real Time Compliance (RTC) and Trusted Execution (TE) for AIX and Audit for Linux endpoints. The IBM PowerSC GUI can be used to configure RTC, TE, Auditd, and view real-time events. For more information, see Chapter 4, “Real-Time File Integrity Monitoring” on page 111.

The IBM PowerSC GUI consists of the following main parts, as shown in Figure 2-1:

- ▶ UI Server  
AIX/Linux LPAR for IBM PowerSC GUI server
- ▶ UI Endpoint Agent  
AIX/Linux endpoints that are managed by IBM PowerSC GUI server
- ▶ Browser  
User interaction

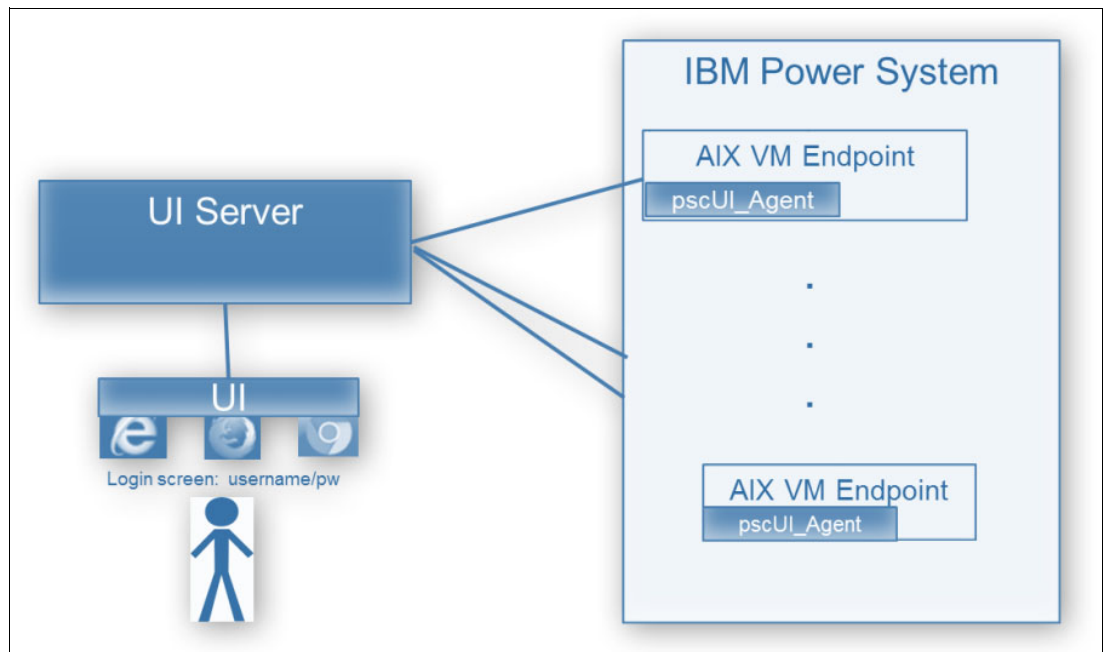


Figure 2-1 IBM PowerSC GUI

IBM PowerSC GUI communication adheres to the following security standards:

- ▶ The HTTPS communication between the IBM PowerSC server and the IBM PowerSC GUI agents on each of the AIX or Linux endpoints are bidirectional and uses industry-standard technology (such as SSL Certificates), and other application-specific technology.
- ▶ All communication between the IBM PowerSC GUI agents and the IBM PowerSC GUI server is encrypted by using protocols and cipher suites that are consistent with the security requirements of the protected systems.
- ▶ IBM PowerSC GUI uses TLS 1.2 protocol level to interact with all the IBM PowerSC GUI agents and with all the IBM PowerSC GUI users.
- ▶ The IBM PowerSC GUI server access supports LDAP or local accounts and allows management of access and endpoint-control authority by using AIX and Linux group membership.
- ▶ After the first contact is established between the IBM PowerSC GUI server and the IBM PowerSC GUI agents, a one-time agent-server security handshake is performed.

## 2.2 Installing IBM PowerSC GUI server

It is recommended to use a separate LPAR for the IBM PowerSC GUI server.

With IBM PowerSC 1.2, you can run the GUI server on AIX or Linux. At the time of this writing, IBM PowerSC supports the following Linux versions:

- ▶ Red Hat Enterprise Linux (RHEL)
- ▶ SUSE Linux Enterprise Server (SLES)

Next, we describe how to install IBM PowerSC GUI Server on three operating systems: AIX, RHEL, and SLES.

### 2.2.1 AIX

In our scenario, we used an AIX 7.2 LPAR and the IBM PowerSC Standard Edition ISO file to perform the installation.

**Note:** The installation media can be downloaded from this [web page](#).

Complete the following steps:

1. At the server, confirm the AIX version by using the **oslevel -s** command:

```
# oslevel -s
7200-01-01-1642
```

2. Copy the ISO file in some path (for example, /software) and use the **loopmount** command to mount the ISO file, as shown in Figure 2-2 on page 22.

```

# cd /software
#
# ls -l
total 2318912
-rw-r----- 1 root      system  1187282944 Sep 12 11:27 POWERSC_STD_EDITION_V1.2_62018.iso
drwxr-xr-x   2 root      system      256 Sep 12 11:23 lost+found
#
# loopmount -i POWERSC_STD_EDITION_V1.2_62018.iso -o "-V cdrfs -o ro" -m /mnt
#
# df -k
Filesystem      1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4         524288      347416   34%    11517    13% /
/dev/hd2        2621440     144876   95%    47958    58% /usr
/dev/hd9var      1048576     961316    9%      817     1% /var
/dev/hd3         1048576     1047152    1%       52     1% /tmp
/dev/hd1         262144     261740    1%        7     1% /home
/dev/hd11admin   262144     261744    1%        5     1% /admin
/proc            -           -     -         -         - /proc
/dev/hd10opt     1048576     675424   36%    11281     7% /opt
/dev/livedump    262144     261776    1%        4     1% /var/adm/ras/livedump
/dev/fslv00     10485760     9324376   12%        5     1% /software
/dev/loop0      1158918         0  100%   579459   100% /mnt
#
# cd /mnt
#
# ls -l
total 24
-rw-rw-r-- 1 4000      4000          42 May 31 09:21 .Version
drwxrwxr-x 3 4000      4000    2048 May 31 09:21 RPMS
drwxrwxr-x 3 4000      4000    2048 May 31 09:20 installp
drwxrwxr-x 2 4000      4000    2048 May 31 09:20 rhel
drwxrwxr-x 2 4000      4000    2048 May 31 09:21 sles
drwxrwxr-x 3 4000      4000    2048 May 31 09:21 usr
#

```

Figure 2-2 Copy and mount the location of the ISO file

3. After mounting the ISO file, go to the `installp/ppc` directory, as shown in Figure 2-3.

```

# ls -l
total 24
-rw-rw-r-- 1 4000      4000          42 May 31 09:21 .Version
drwxrwxr-x 3 4000      4000    2048 May 31 09:21 RPMS
drwxrwxr-x 3 4000      4000    2048 May 31 09:20 installp
drwxrwxr-x 2 4000      4000    2048 May 31 09:20 rhel
drwxrwxr-x 2 4000      4000    2048 May 31 09:21 sles
drwxrwxr-x 3 4000      4000    2048 May 31 09:21 usr
#
#
# cd installp
#
# ls -l
total 4
drwxrwxr-x 2 4000      4000    2048 May 31 09:21 ppc
#
# cd ppc
#
# ls -l
total 1099860
-rw-rw-r-- 1 4000      4000    29190 May 31 09:21 .toc
-rw-r--r-- 2 4000      4000   879104 May 31 09:20 openpts.verifier
-rw-r--r-- 2 4000      4000  9142784 May 31 09:20 powerscStd.ice
-rw-r--r-- 2 4000      4000  437760 May 31 09:20 powerscStd.license
-rw-r--r-- 2 4000      4000  7330304 May 31 09:20 powerscStd.msg
-rw-r--r-- 2 4000      4000  133632 May 31 09:20 powerscStd.rtc
-rw-r--r-- 2 4000      4000  484352 May 31 09:20 powerscStd.svm
-rw-r--r-- 2 4000      4000  8919040 May 31 09:20 powerscStd.tnc_commands
-rw-r--r-- 2 4000      4000  57313792 May 31 09:20 powerscStd.tnc_lib
-rw-r--r-- 2 4000      4000   263168 May 31 09:20 powerscStd.tnc_pm
-rw-r--r-- 2 4000      4000 174404096 May 31 09:20 powerscStd.uiAgent
-rw-r--r-- 2 4000      4000 303695360 May 31 09:20 powerscStd.uiServer
-rw-r--r-- 2 4000      4000   81408 May 31 09:20 powerscStd.vlog
#

```

Figure 2-3 Move the ISO file to the installation location

4. You can use command line or SMIT to install IBM PowerSC filesets. In our example, we used SMIT for the installation. You can run **smitty installp** fast path to install the filesets, as shown in Figure 2-4. For the IBM PowerSC GUI Server, we need the following filesets:
- ▶ powerscStd.license
  - ▶ powerscStd.uiserver

```
# smitty installp

Install and Update Software

Move cursor to desired item and press Enter.

Install Software
Update Installed Software to Latest Level (Update All)
Update Installed Software to Latest Level (Live Update)
Install Software Bundle
Update Software by Fix (APAR)
Install and Update from ALL Available Software
```

Figure 2-4 Using smitty installp to install IBM PowerSC filesets

Figure 2-5 shows the input pane for the location of the installation filesets.

```
Install Software

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

* INPUT device / directory for software [Entry Fields]
```

Figure 2-5 Specifying the location of the filesets

Figure 2-6 shows the IBM PowerSC GUI server installation filesets.

```
Install Software

Type or select values in entry fields.
Press Enter AFTER making all desired chan|
SOFTWARE to install
Move cursor to desired item and press F7. Use arrow keys to scroll.
ONE OR MORE items can be selected.
Press Enter AFTER making all selections.

* INPUT device / directory for software | [TOP]
* SOFTWARE to install
PREVIEW only? (install operation will N|
COMMIT software updates?
SAVE replaced files?
AUTOMATICALLY install requisite softwar|
EXTEND file systems if space needed?
OVERWRITE same or newer versions?
VERIFY install and check file sizes?
Include corresponding LANGUAGE filesets|
DETAILED output?
Process multiple volumes?
ACCEPT new license agreements?
Preview new LICENSE agreements?
INVOKE live update?
Requires /var/adm/ras/liveupdate/lvupda|
WPAR Management
Perform Operation in Global Environ|
Perform Operation on Detached WPARs|
Detached WPAR Names
Remount Installation Device in WPAR|
Alternate WPAR Installation Device

openpts.verifier ALL
+ 1.0.0.1 Open Platform Trust Services - verifier
powerscStd.ice ALL
+ 1.2.0.0 IBM PowerSC Standard Profile
powerscStd.license ALL
+ 7.1.2.0 PowerSC Standard Edition
powerscStd.rtc ALL
+ 1.2.0.0 Real-Time Compliance
powerscStd.svm ALL
+ 1.2.0.0 Secure Virtual Machine
powerscStd.tnc_commands ALL
+ 1.2.0.0 Trusted Network Connect Commands
powerscStd.tnc_lib ALL
+ 1.2.0.0 Trusted Network Connect Libraries
powerscStd.tnc_pm ALL
+ 1.2.0.0 Trusted Network Connect for Patch Management
powerscStd.uiAgent ALL
+ 1.2.0.0 PowerSC User Interface Agent
powerscStd.uiserver ALL
+ 1.2.0.0 PowerSC User Interface Server
[MORE...11]

F1=Help F2=Refresh F3=Cancel F4=List
F5=Reset F7=Select F8=Image F10=Exit F8=Image
F9=Shell F6| Enter=Do /=Find n=Find Next
F1+-----
```

Figure 2-6 Select the filesets

**Note:** Select the **powerscStd.ice** fileset to harden the IBM PowerSC GUI server. Select **powerscStd.rtc** to use the Real Time Compliance feature on the IBM PowerSC GUI server.

Figure 2-7 shows the pane to accept the license agreement.

Install Software

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

	[Entry Fields]
* INPUT device / directory for software	.
* SOFTWARE to install	<b>powerscStd.license</b>
PREVIEW only? (install operation will NOT occur)	no
COMMIT software updates?	yes
SAVE replaced files?	no
AUTOMATICALLY install requisite software?	yes
EXTEND file systems if space needed?	yes
OVERWRITE same or newer versions?	no
VERIFY install and check file sizes?	no
Include corresponding LANGUAGE filesets?	yes
DETAILED output?	no
Process multiple volumes?	yes
<b>ACCEPT new license agreements?</b>	<b>yes</b>
Preview new LICENSE agreements?	no
INVOKE live update?	no
Requires /var/adm/ras/liveupdate/lvupdate.data.	
<b>WPAR Management</b>	
Perform Operation in Global Environment	yes
Perform Operation on Detached WPARs	no
Detached WPAR Names	[_all_wpars]
Remount Installation Device in WPARs	yes
Alternate WPAR Installation Device	[]

Figure 2-7 Accept the license

Figure 2-8 shows the installation completion message pane without errors.

```

Command: OK          stdout: yes          stderr: no
Before command completion, additional instructions may appear below.

[MORE...89]

. . . . . << Copyright notice for powerscStd.msg >> . . . . .
Licensed Materials - Property of IBM

5765PSE00
Copyright International Business Machines Corp. 2015, 2016.

All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
. . . . . << End of copyright notice for powerscStd.msg >>. . . .

Finished processing all filesets. (Total time: 1 mins 1 secs).
Finished processing all filesets. (Total time: 1 mins 1 secs).

+-----+
| Summaries: |
+-----+

Installation Summary
-----
Name                                Level      Part      Event      Result
-----
powerscStd.license                  7.1.2.0    USR        APPLY      SUCCESS
powerscStd.license                  7.1.2.0    ROOT       APPLY      SUCCESS
powerscStd.uiServer.rte             1.2.0.0    USR        APPLY      SUCCESS
powerscStd.uiServer.rte             1.2.0.0    ROOT       APPLY      SUCCESS
powerscStd.msg.en_US                1.2.0.0    USR        APPLY      SUCCESS

File /etc/inittab has been modified.

One or more of the files listed in /etc/check_config.files have changed.
See /var/adm/ras/config.diff for details.

```

Figure 2-8 Installation that is completed without errors

- After completing the installation through the smitty, go to the command line and run the **lslpp -L** command to verify the installation:

```

# lslpp -L powerscStd.uiServer.rte
Fileset              Level  State  Type  Description (Uninstaller)
-----
powerscStd.uiServer.rte  1.2.0.0  C      F      PowerSC User Interface Server

```

The installation adds an entry in the `/etc/inittab` file to start the IBM PowerSC Server at start time. Use the **lsitab** command to verify the entry:

```

# lsitab pscuiserver
pscuiserver:2:wait:/usr/bin/startsrc -s pscuiserver > /dev/console 2>&1

```

IBM PowerSC GUI Server installation automatically installs the `pscuiserver` service. This installation can be verified by running the **lssrc** command:

```

# lssrc -s pscuiserver
Subsystem      Group          PID           Status
pscuiserver    pscuiserver    9503060       active

```

By default, IBM PowerSC GUI Server uses port 80 (for Non-SSL) and 443 (for SSL). Although the IBM PowerSC GUI Server opens port 80 for non-SSL, the traffic is redirected to port 443 to ensure that it is always encrypted. You can change the default port by running the **pscuiserverctl** command:

```
# netstat -an | grep LISTEN | egrep "80|443"
tcp      0      0 *.80          *.*           LISTEN
tcp      0      0 *.443         *.*           LISTEN
```

The IBM PowerSC GUI server listens on TCP port 443 for all communication from the PowerSC GUI agent, or from any web browser. The IBM PowerSC GUI agent that is running on each endpoint listens on TCP port 11125 for all communication from the IBM PowerSC GUI server.

## 2.2.2 Red Hat Enterprise Linux

With PowerSC 1.2, you can run the IBM PowerSC GUI Server on Red Hat Enterprise Linux (RHEL) running on Power Systems. Similar to the steps for AIX, the following procedure demonstrates how to install and configure the UI Server on RHEL.

**Note:** We are not showing how to install Red Hat Enterprise Linux on IBM Power System in this section. This process can be done by using the manuals that are available at this [web page](#).

You can find the RHEL version by viewing the `/etc/redhat-release` file as shown in the following example:

```
#cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.4 (Maipo)
```

By using a similar procedure for AIX, create a directory that is named `/software`, transfer the ISO file to the server, and mount it to have access to the rpm files, as shown in Figure 2-9.

```
#
# cd /software
#
# ls -l
total 1159456
-rw-r----- 1 root root 1187282944 Sep 12 13:14 POWERSC_STD_EDITION_V1.2_62018.iso
#
# mount -t iso9660 -o loop /software/POWERSC_STD_EDITION_V1.2_62018.iso /mnt
mount: /dev/loop0 is write-protected, mounting read-only
#
# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/vg_root-lv_root 40G       3.0G    38G   8% /
devtmpfs                   776M       0    776M   0% /dev
tmpfs                      811M       0    811M   0% /dev/shm
tmpfs                      811M     16M    796M   2% /run
tmpfs                      811M       0    811M   0% /sys/fs/cgroup
/dev/mapper/vg_root-lv_home 1014M     33M    982M   4% /home
/dev/sdh2                  497M     171M    326M  35% /boot
tmpfs                      163M       0    163M   0% /run/user/0
/dev/loop0                 1.2G     1.2G       0 100% /mnt
#
```

Figure 2-9 Accessing the RPM files



After mounting the ISO file, find a directory that is named /rhel, which includes three scripts that you can use for RHEL in later steps, as shown in Figure 2-10.

```
# cd /mnt
#
# ls -l
total 10
drwxrwxr-x 3 4000 4000 2048 May 31 15:20 installp
drwxrwxr-x 2 4000 4000 2048 May 31 15:20 rhel
drwxrwxr-x 3 4000 4000 2048 May 31 15:21 RPMS
drwxrwxr-x 2 4000 4000 2048 May 31 15:21 sles
drwxrwxr-x 3 4000 4000 2048 May 31 15:21 usr
#
# cd rhel
#
# ls -l
total 310030
-rw-r--r-- 1 4000 4000 3509441 May 31 15:20 comply-1.0-rhel6.ppc64.sh
-rw-r--r-- 1 4000 4000 3930161 May 31 15:20 powersc-pscexpert-1.2.0.0-rhel7.ppc64le.sh
-rw-r--r-- 1 4000 4000 132218910 May 31 15:20 powersc-uiAgent-1.2.0.0-rhel7.ppc64le.sh
-rw-r--r-- 1 4000 4000 177810872 May 31 15:21 powersc-uiServer-1.2.0.0-rhel7.ppc64le.sh
#
```

Figure 2-10 Scripts under the rhel directory to be used for the GUI installation

To install the IBM PowerSC GUI server, run the script as shown in Figure 2-11.

```
# bash ./powersc-uiServer-1.2.0.0-rhel7.ppc64le.sh
A JVM must be installed and on the path
#
```

Figure 2-11 Script to install the IBM PowerSC GUI server

We encountered a Java virtual machine (JVM) related error. To correct this error, you must check whether JAVA was installed, and that the PATH variable is updated with the path where JAVA is installed, as shown in Figure 2-12. (A few other prerequisites might exist for installation.)

```
# which java
/usr/bin/java
#
# java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 8.0.5.15 - pxl6480sr5fp15
-20180502_01(SR5 FP15))
IBM J9 VM (build 2.9, JRE 1.8.0 Linux ppc64le-64 Compressed Refe
rences 20180425_385365 (JIT enabled, AOT enabled)
OpenJ9      - a7ffbfe
OMR         - a531219
IBM         - 59ef3dc)
JCL - 20180425_01 based on Oracle jdk8u171-b11
#
```

Figure 2-12 Checking JAVA is installed and the variable PATH is valid

The RPMs that are shown in Example 2-1 must be met.

#### Example 2-1 RPM prerequisites

```
Prereq : 'gettext' 'openssl' 'perl'); java
# rpm -q --requires powersc-uiServer-1.2.0.0-rhel7.ppc64le
/bin/sh
```

```

/bin/sh
/bin/sh
/bin/sh
gettext
openssl
perl
redhat-release-server >= 7.4
rpmLib(CompressedFileNames) <= 3.0.4-1
rpmLib(FileDigests) <= 4.6.0-1
rpmLib(PayloadFilesHavePrefix) <= 4.0-1
rpmLib(PayloadIsXz) <= 5.2-1

```

---

**Note:** It is recommended that your media repository is up to date to avoid any problem with rpm dependencies.

After completing and resolving any dependency, run again the script to install the GUI, as shown in Figure 2-13.

```

# bash ./powersc-uiServer-1.2.0.0-rhel7.ppc64le.sh
x - created lock directory _sh31162.
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN
"ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND
PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, or "99" to go back to the previous screen.
1
x - removed lock directory _sh31162.
Preparing... ##### [100%]
Updating / installing...
  1:powersc-uiServer-1.2.0.0-rhel7 ##### [100%]
/opt/powersc/uiServer /
logonGroupList=security
security=*
Certificate stored in file </etc/security/powersc/uiServer/psc_server_cert.pem>
Certificate was added to keystore
Copy /etc/security/powersc/uiServer/endpointTruststore.jks to /etc/security/powe
rsc/uiAgent/endpointTruststore.jks on every endpoint.
Certificate stored in file </etc/security/powersc/uiServer/psc_signing_cert.pem>
Certificate was added to keystore
httpPort=80
httpsPort=443
Created symlink from /etc/systemd/system/multi-user.target.wants/powersc-uiServe
r.service to /usr/lib/systemd/system/powersc-uiServer.service.

```

Figure 2-13 Running the script again to install the GUI

Accept the license agreement. After few minutes, you see output that is similar to the output that is shown in Figure 2-13 on page 28. You can run the **systemctl** command to verify that the powersc-uiscrver service is running, as shown in Figure 2-14.

```
# systemctl status powersc-uiServer.service
â powersc-uiServer.service - PowerSC UI Server
   Loaded: loaded (/usr/lib/systemd/system/powersc-uiServer.service; enabled; ven
dor preset: disabled)
   Active: active (running) since Wed 2018-09-12 13:49:39 EDT; 4h 44min ago
 Main PID: 22715 (uiServer.sh)
   CGroup: /system.slice/powersc-uiServer.service
           ââ22715 /bin/sh /opt/powersc/uiServer/bin/uiServer.sh
           ââ22760 /opt/powersc/uiServer/bin/uiscrver /opt/powersc/uiServer/bi...

Sep 12 13:49:39 p52n76 systemd[1]: Started PowerSC UI Server.
Sep 12 13:49:39 p52n76 systemd[1]: Starting PowerSC UI Server...
Sep 12 13:49:39 p52n76 uiServer.sh[22715]: Starting PowerSC UI server with ma....
Sep 12 13:49:42 p52n76 uiServer.sh[22715]: log file: /var/log/powersc/uiServe...g
Sep 12 17:18:16 p52n76 uiscrver[22760]: IBM Java[22760]: pam_securetty(login...ty
Sep 12 17:18:16 p52n76 uiscrver[22760]: IBM Java[22760]: pam_unix(login:auth...wn
Sep 12 17:18:16 p52n76 uiscrver[22760]: IBM Java[22760]: pam_unix(login:auth...t=
Sep 12 17:18:26 p52n76 uiscrver[22760]: IBM Java[22760]: pam_securetty(login...ty
Sep 12 17:18:26 p52n76 uiscrver[22760]: IBM Java[22760]: pam_unix(login:auth...wn
Sep 12 17:18:26 p52n76 uiscrver[22760]: IBM Java[22760]: pam_unix(login:auth...t=
Hint: Some lines were ellipsized, use -l to show in full.
```

Figure 2-14 Check the status of the IBM PowerSC GUI server

Run the **netstat** command to verify that the ports are opened, as shown in Figure 2-15.

```
# netstat -an | grep LISTEN | egrep "80|443"
tcp6      0      0  :::80          :::*           LISTEN
tcp6      0      0  :::443         :::*           LISTEN
```

Figure 2-15 Verify the open ports

## 2.2.3 SUSE Linux Enterprise Server

With IBM PowerSC 1.2 release, you can run the IBM PowerSC GUI Server on SUSE Linux Enterprise Server (SLES) running on Power Systems. Similar to the steps for AIX, the following procedures demonstrate how to install and configure GUI Server on SLES.

**Note:** This section does not show you how to install SLES. This process can be done by using your procedures or following the instructions that are described in the manual that is available at [this web page](#).

By using similar steps as with AIX, create a directory that is named /software, transfer the ISO file to the server, and mount it to have access to the rpm files, as shown in Figure 2-16.

```
# cd /software
#
# ls -l
total 1159488
-rw-r----- 1 root root 1187282944 Sep 12 17:10 POWERSC_STD_EDITION_V1.2_62018.iso
#
# mount POWERSC_STD_EDITION_V1.2_62018.iso -o loop /mnt
mount: /dev/loop0 is write-protected, mounting read-only
#
# cd /mnt
#
# ls -l
total 12
-rw-rw-r-- 1 4000 4000 42 May 31 14:21 .Version
drwxrwxr-x 3 4000 4000 84 May 31 14:21 RPMS
drwxrwxr-x 3 4000 4000 84 May 31 14:20 installp
drwxrwxr-x 2 4000 4000 344 May 31 14:20 rhel
drwxrwxr-x 2 4000 4000 356 May 31 14:21 sles
drwxrwxr-x 3 4000 4000 84 May 31 14:21 usr
#
```

Figure 2-16 Preparing the repository

After mounting the ISO file, you find a directory that is named sles, which includes three scripts that you can use for installation, as shown in Figure 2-16.

The sles directory contains three scripts, as shown in Figure 2-17.

```
# cd sles
#
# ls -l
total 296532
-rw-r--r-- 1 4000 4000 3732689 May 31 14:21 comply-1.0-sles11.ppc64.sh
-rw-r--r-- 1 4000 4000 3845147 May 31 14:21 powersc-pscxpert-1.2.0.0-sles12.ppc64le.sh
-rw-r--r-- 1 4000 4000 125424036 May 31 14:21 powersc-uiAgent-1.2.0.0-sles12.ppc64le.sh
-rw-r--r-- 1 4000 4000 170641870 May 31 14:21 powersc-uiServer-1.2.0.0-sles12.ppc64le.sh
```

Figure 2-17 ISO installation scripts

To install the IBM PowerSC GUI server, run the script as shown in Figure 2-18.

```
# bash ./powersc-uiServer-1.2.0.0-sles12.ppc64le.sh
A JVM must be installed and on the path
```

Figure 2-18 Script to install PowerSC GUI

We encountered a JVM-related error, as shown in Figure 2-19.

```
# which java
which: no java in (/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/b
in:/bin:/usr/bin/X11:/usr/games)
#
# java -version
If 'java' is not a typo you can use command-not-found to lookup the package that c
ontains it, like this:
__cnf java
```

Figure 2-19 Error that is encountered with JAVA installation

To correct this error, you must check that JAVA was installed, and that the PATH variable is updated with the path where JAVA is installed, as shown in Figure 2-20 on page 31. (A few prerequisites might exist for the installation.)

```
# which java
/opt/ibm/java-ppc64le-80/jre/bin/java
#
# java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 8.0.5.21 - pxl6480sr5fp21-20180830_01(SR5 F
P21))
IBM J9 VM (build 2.9, JRE 1.8.0 Linux ppc64le-64-Bit Compressed References 2018082
9_395745 (JIT enabled, AOT enabled)
OpenJ9      - e82188c
OMR         - eea30e
IBM         - 98805ca)
JCL        - 20180821_01 based on Oracle jdk8u181-b12
```

Figure 2-20 Checking that JAVA was installed and its path

**Note:** It is recommended that you have your media repository up to date to avoid any problem with rpm dependencies.

After completing and resolving any dependency, run again the script to install the GUI, as shown in Figure 2-21.

```
# bash ./powersc-uiServer-1.2.0.0-sles12.ppc64le.sh
x - created lock directory _sh07800.
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN
"ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND
PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, or "99" to go back to the previous screen.
1
x - removed lock directory _sh07800.
Preparing... ##### [100%]
Updating / installing...
  1:powersc-uiServer-1.2.0.0-sles12 ##### [100%]
/opt/powersc/uiServer /
logonGroupList=security
security=*
Certificate stored in file </etc/security/powersc/uiServer/psc_server_cert.pem>
Certificate was added to keystore
Certificate was added to keystore
Copy /etc/security/powersc/uiServer/endpointTruststore.jks to /etc/security/powersc/uiAgent
/endpointTruststore.jks on every endpoint.
Certificate stored in file </etc/security/powersc/uiServer/psc_signing_cert.pem>
Certificate was added to keystore
httpPort=80
httpsPort=443
Created symlink from /etc/systemd/system/multi-user.target.wants/powersc-uiServer.service t
o /usr/lib/systemd/system/powersc-uiServer.service.
#
```

Figure 2-21 Run the script to install the GUI

Accept the license agreement. After few minutes, you see an output that is similar to the output that is shown in Figure 2-21.



You can run the **systemctl** command to verify whether the powersc-uiServer service is running, as shown in Figure 2-22.

```
# systemctl status powersc-uiServer.service
â powerSC UI Server
  Loaded: loaded (/usr/lib/systemd/system/powersc-uiServer.service; ena
bled; vendor preset: disabled)
  Active: active (running) since Wed 2018-09-12 18:21:00 EDT; 10min ago
    Main PID: 13835 (uiServer.sh)
      Tasks: 87 (limit: 512)
     CGroup: /system.slice/powersc-uiServer.service
             ââ13835 /bin/sh /opt/powersc/uiServer/bin/uiServer.sh
             ââ13883 /opt/powersc/uiServer/bin/uisever /opt/powersc/ui...

Sep 12 18:21:00 p52n73 systemd[1]: Started PowerSC UI Server.
Sep 12 18:21:00 p52n73 uiServer.sh[13835]: Starting PowerSC UI server...
Sep 12 18:21:02 p52n73 uiServer.sh[13835]: log file: /var/log/powersc...
Hint: Some lines were ellipsized, use -l to show in full.
```

Figure 2-22 Check the status of the uiServer service

Run the **netstat** command to verify that the ports are opened, as shown in Figure 2-23.

```
# netstat -an | grep LISTEN | egrep ":80|:443"
tcp        0      0      0 :::80          :::*           LISTEN
tcp        0      0      0 :::443         :::*           LISTEN
```

Figure 2-23 Checking that the ports are opened

## 2.3 GUI administration

After you install and start the IBM PowerSC GUI Server, you can access it by using any browser, such as Internet Explorer, Mozilla, or Chrome. Before you access the PowerSC GUI, you must set up the accounts of users who can log in to the GUI.

More security control is provided by using UNIX Groups. Any LDAP users or local users who are defined by the AIX or Linux operating system must be a member of the security group to have the endpoint management into the PowerSC GUI. The administrator must set or change group membership by using the **pscuiserverctl** command.

After you are logged in, you might still be restricted to view-only mode. You can use the user authority function to perform actions against endpoints that are controlled by UNIX group membership. To perform any actions, you must be a member of a UNIX group that has permission to manage the endpoint.

By default, any user who is a member of the security group can manage every endpoint that is visible in the IBM PowerSC GUI. The IBM PowerSC administrator can restrict user access to the individual endpoint level by using the **pscuiserverctl** command with the **setgroup** parameter. For example, **pscuiserverctl setgroup <group name> <comma- or space-separated list of host names>**.

The IBM PowerSC GUI supports the following access roles:

- Login access: This access is required to log in to the IBM PowerSC GUI. You can assign login access to a group or multiple groups by setting the `loginGroupList` by using the **pscuiserverctl** command. For more information, see 2.3.3, “PowerSC GUI login” on page 37.

- **Administrative access:** This access is required to perform administrative functions by using the IBM PowerSC GUI. You can assign administrative access to a group or multiple groups by setting the `administratorGroupList` by using the `pscuiserverctl` command. For more information, see 2.3.3, “PowerSC GUI login” on page 37.

**Note:** To get administrative access, you must assign the group to be part of `logonGroupList` and `administratorGroupList` so that the group members can log in and perform administrative tasks.

Because of security recommended practices, we created separate users and groups in our scenario to demonstrate how it is possible to configure the access to GUI for login only and administrative purposes.

Normally, you can run the `pscuiserverctl` command as a root user to set the access control. Create a user and group and set the password. We demonstrate how we use it for AIX and Linux systems in 2.3.2, “Manage users and groups” on page 34.

### 2.3.1 Endpoint administration

The IBM PowerSC GUI Server provides another access control mechanism through which you can restrict the groups to manage specific endpoints. After logging in to the IBM PowerSC GUI, a user can view only the status of endpoints if their user account is a member of a UNIX (AIX or Linux) group that is allowed to manage the endpoint.

Run the following script to specify the AIX groups in which a user must be a member to run commands on specific endpoints. You must provide fully qualified host names of the endpoints. The groups that you specify are written to the `/etc/security/powersc/uiServer/groups.txt` file:

```
pscuiserverctl setGroups <group name> <comma separated list of host names>
```

By default, any user who is a member of the security group can manage every endpoint that is visible in the IBM PowerSC GUI.

In the IBM PowerSC GUI, you must pay attention to and consider the following relationship between endpoints and groups:

- One UNIX group can be associated with many endpoints
- One endpoint can be associated with many UNIX groups

When a user is logged in to the IBM PowerSC GUI, group associations are used to determine whether a user is allowed to run commands to specific endpoints, or whether the user is allowed only to view endpoint status. Consider the following points:

- If the user must run commands against a specific endpoint by using the IBM PowerSC GUI, the user must be associated with one of the groups that is associated with the endpoint.
- The group membership for users that are logging in to the IBM PowerSC GUI is compared with the set of groups that are associated with each endpoint. If the user's group membership matches groups that are associated with each endpoint, the user can run commands, such as **Apply profiles**, **Undo**, and **Check** against that endpoint.
- If the user's group membership does not match any groups that are associated with each endpoint, the user can view only the status for that endpoint.

## 2.3.2 Manage users and groups

This section creates users and groups on AIX and Linux and demonstrates the IBM PowerSC GUI Server's login process.

### AIX

On AIX, you can use the **mkgroup** command to create a group and the **mkuser** command to create a user. As shown in Figure 2-24, we create a group **psclog** and add a user **psclogin** to this group. Then, we run the **pscuerverctl** command to set the **logonGroupList** to the **psclog** group. This process allows all users that are part of the **psclog** group (in our case, the **psclogin** user) to log in to the IBM PowerSC GUI server.

```
# mkgroup psclog
#
# mkuser -a pgrp=psclog groups=psclog psclogin
#
# lsuser -a pgrp psclogin
psclogin pgrp=psclog
#
# /opt/powersc/uiServer/bin/pscuerverctl set logonGroupList psclog
logonGroupList=psclog
#
# /opt/powersc/uiServer/bin/pscuerverctl set logonGroupList
psclog
#
```

Figure 2-24 Creating group and adding a user to it

Next, we create a group **pscadm** and add user **pscadmin** to this group, as shown in Figure 2-25.

```
# mkgroup pscadm
#
# mkuser -a pgrp=pscadm groups=pscadm pscadmin
#
# lsuser -a pgrp pscadmin
pscadmin pgrp=pscadm
#
# pscuerverctl set logonGroupList
psclog
#
# pscuerverctl set logonGroupList psclog,pscadm
logonGroupList=psclog,pscadm
#
# pscuerverctl set administratorGroupList pscadm
administratorGroupList=pscadm
#
# pscuerverctl set administratorGroupList
pscadm
#
```

Figure 2-25 Create group and adding the user to the group



Then, we run the **pscuiserverctl** command to set the `logonGroupList` and the `administratorGroupList` to include the `pscadm` group. This process allows all users that are part of the `pscadm` group (in our case, the `pscadmin` user) to log in to the IBM PowerSC GUI server.

## Linux

On Linux, you can use the **groupadd** command to create a group, the **useradd** command to create a user, and the **usermod** command to modify the user's group.

As shown in Example 2-2, we create a group `psclog` and added a user `psclogin` to this group. Then, we run the **pscuiserverctl** command to set the `logonGroupList` to the `psclog` group. This process allows all users that are part of the `psclog` group (in our case, the `psclogin` user) to log in to the IBM PowerSC GUI server.

### *Example 2-2 Creating group and adding the user to the group*

---

```
Linux:
[root@p52n76 ~]# useradd pscadmin
[root@p52n76 ~]#
[root@p52n76 ~]# passwd pscadmin
Changing password for user pscadmin.
New password:
BAD PASSWORD: The password is shorter than 7 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@p52n76 ~]#
[root@p52n76 ~]#
[root@p52n76 ~]# groupadd pscadm
[root@p52n76 ~]#
[root@p52n76 ~]# usermod -G pscadm pscadmin
[root@p52n76 ~]#
[root@p52n76 ~]#
[root@p52n76 ~]# groups pscadmin
pscadmin : pscadmin pscadm
[root@p52n76 ~]#
[root@p52n76 ~]#
[root@p52n76 ~]# pscuiserverctl set administratorGroupList pscadmin
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
administratorGroupList=pscadm
[root@p52n76 ~]# pscuiserverctl set administratorGroupList
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
pscadmin
```

```

[root@p52n76 ~]#
[root@p52n76 ~]#
[root@p52n76 ~]# pscuiserverctl set logonGroupList psclog,pscadmin
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
logonGroupList=psclog,pscadmin
[root@p52n76 ~]# pscuiserverctl set logonGroupList
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
psclog,pscadmin
[root@p52n76 ~]# pscuiserverctl set logonGroupList psclog,pscadm
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
logonGroupList=psclog,pscadm
[root@p52n76 ~]# pscuiserverctl set logonGroupList
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
psclog,pscadm
[root@p52n76 ~]# pscuiserverctl set logonGroupList
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
psclog,pscadm
[root@p52n76 ~]# pscuiserverctl set administratorGroupList
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:

```

```
LANGUAGE = (unset),
LC_ALL = (unset),
LC_CTYPE = "UTF-8",
LANG = "en_US.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
pscadmin
[root@p52n76 ~]# pscuiserverctl set administratorGroupList pscadm
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
LANGUAGE = (unset),
LC_ALL = (unset),
LC_CTYPE = "UTF-8",
LANG = "en_US.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
administratorGroupList=pscadm
[root@p52n76 ~]#
(...)
```

---

### 2.3.3 PowerSC GUI login

We first tried logging in to the GUI by using the `pscllogin` user account. To access the GUI, use any browser and enter the hostname or IP address of the PowerSC GUI server:

`https://<hostname of PowerSC GUI server>`

By default, IBM PowerSC GUI uses port 443 for SSL-based communication, which can be modified by using the `pscuiserverctl` command. If you use any other port, you must specify the port number in the URL:

`https://<hostname of PowerSC GUI server>:<port>`

The IBM PowerSC GUI login page opens, as shown in Figure 2-26 on page 38. We log in by using the `pscllogin` user account.

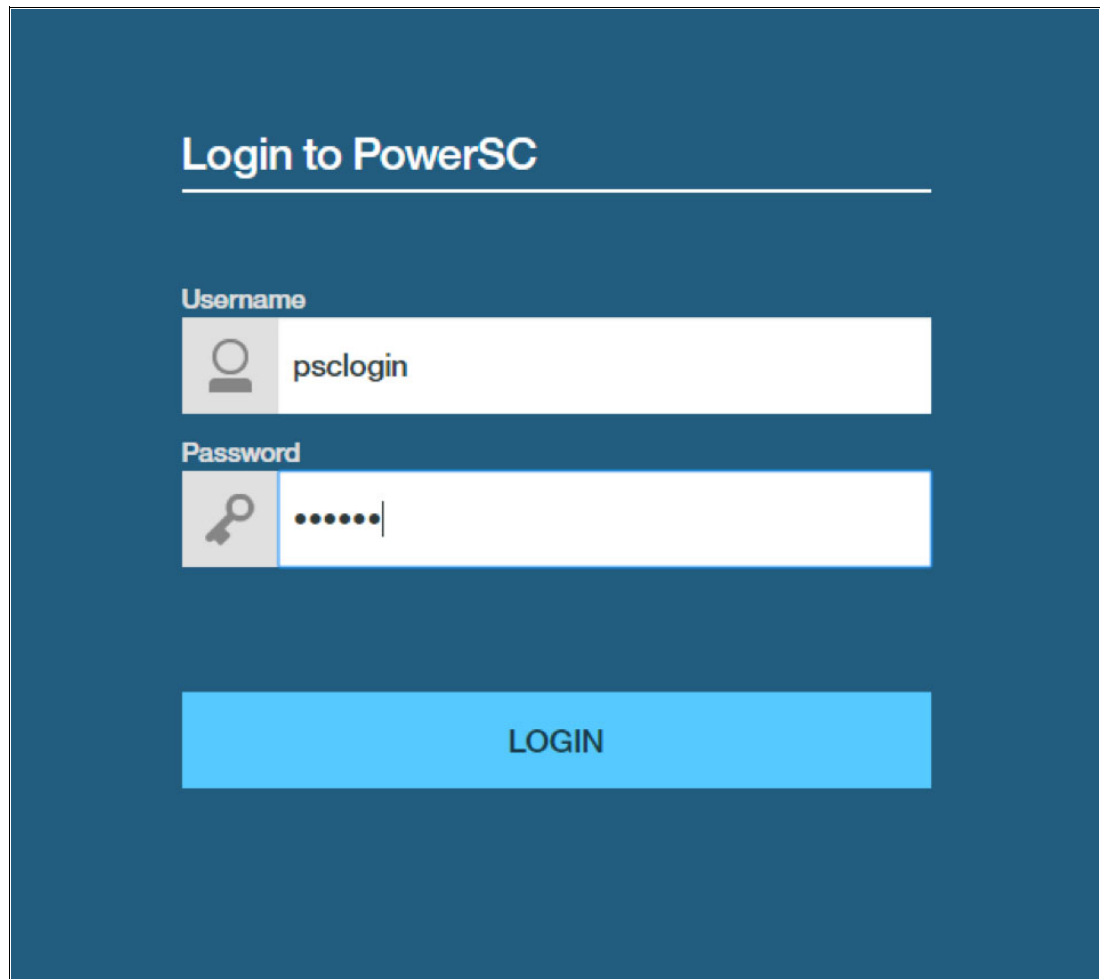


Figure 2-26 PowerSC login page

Because the psclogin user account does not have administrative access, you see many of the GUI options grayed out or unavailable, as shown in Figure 2-27.

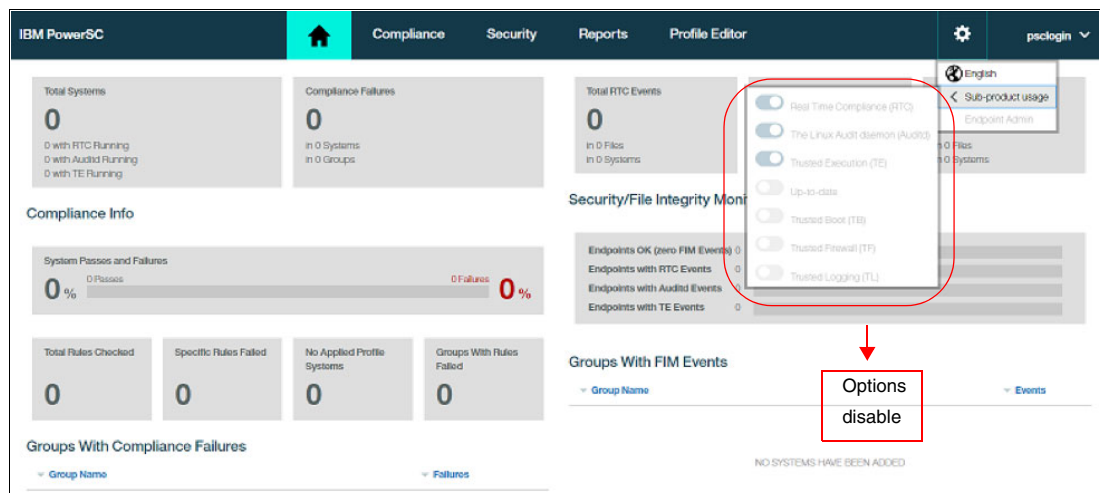
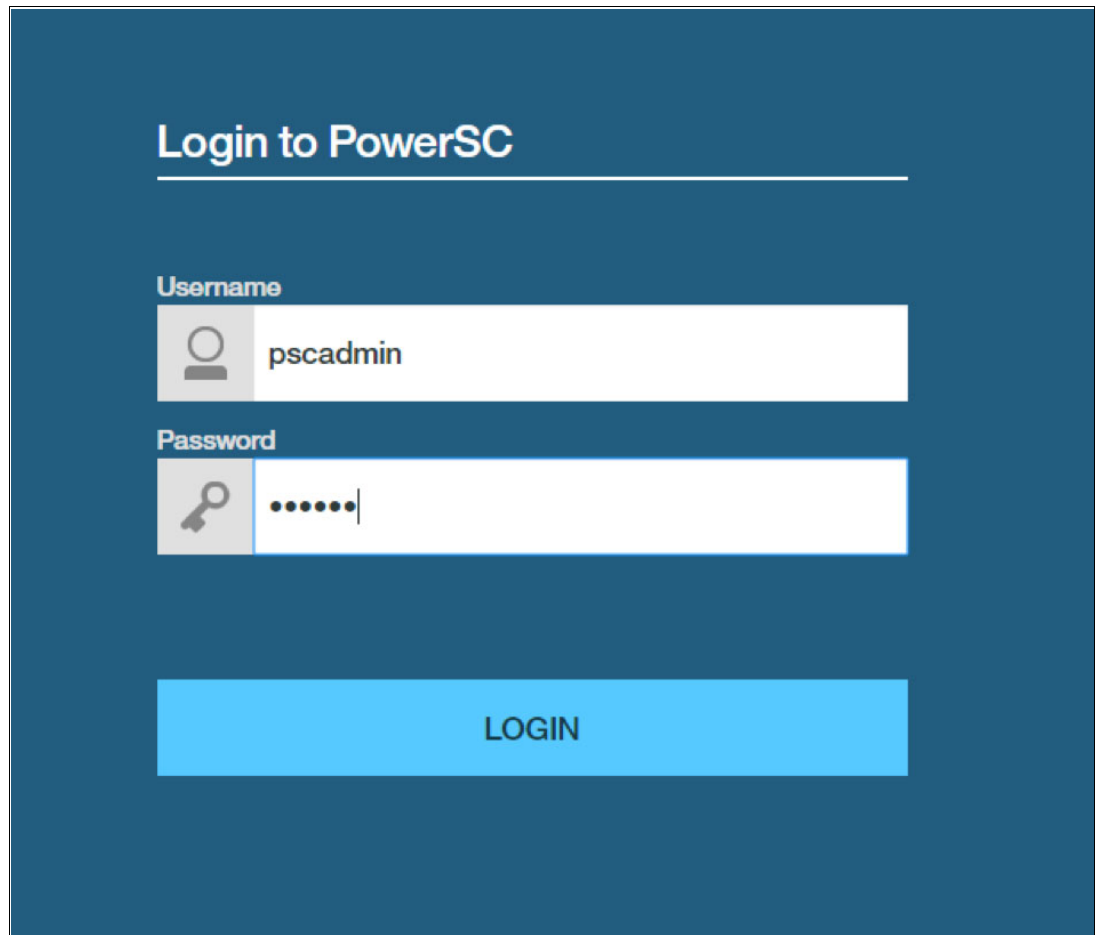


Figure 2-27 GUI home overview for psclogin account

Next, login by using the pscadmin account, as shown in Figure 2-28.



The image shows the 'Login to PowerSC' interface. It has a dark blue background. At the top, the title 'Login to PowerSC' is in white. Below it, there are two input fields: 'Username' with a person icon and 'Password' with a key icon. The username field contains the text 'pscadmin' and the password field contains masked dots. A large blue 'LOGIN' button is at the bottom.

Figure 2-28 PowerSC login with pscadmin

The pscadmin account features all administrative access rights, as shown in Figure 2-29.

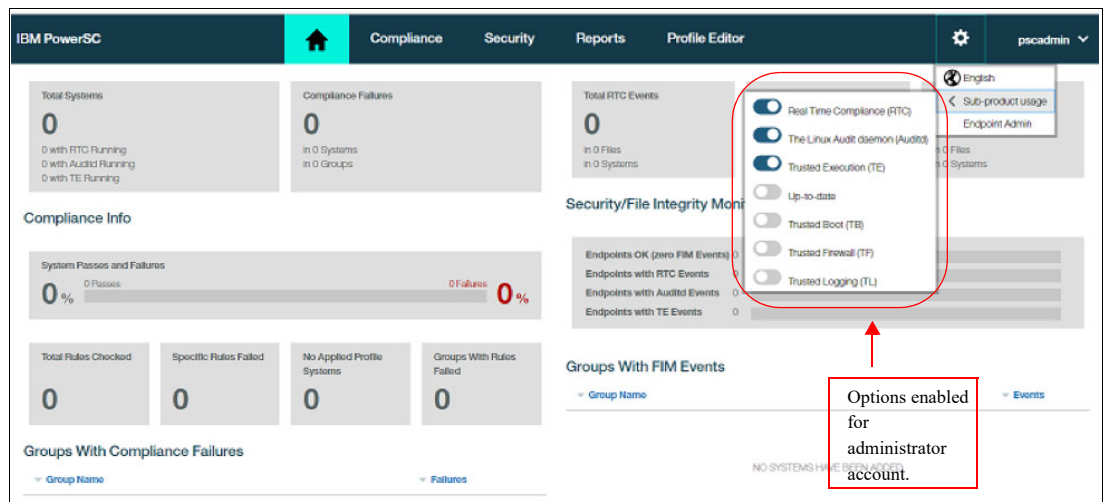


Figure 2-29 GUI home overview for pscadmin account

Next, we describe how you can install and configure UIAgent on the endpoints so that IBM PowerSC GUI server can manage them.

## 2.4 Installing the UIAgent

You must install UIAgent on the LPAR so that it can communicate with the IBM PowerSC GUI server. As of this writing, IBM PowerSC GUI agent can be installed on AIX, RHEL, and SUSE endpoints.

### 2.4.1 Installing UIAgent on AIX

The IBM PowerSC GUI agent filesets are provided with the IBM PowerSC Standard Edition. It is not part of the base AIX operating system.

Complete the following steps to install UIAgent on an AIX LPAR:

1. Copy the PowerSC Standard Edition ISO file to LPAR and loopmount it:

```
loopmount -i /software/POWERSC_STD_EDITION_V1.2_62018.iso -o "-V cdrfs -o ro"
-m /mnt
```

2. Go to the installation path (/mnt/installp/ppc), as shown in Figure 2-30.

```
# cd /mnt/installp/ppc
#
# ls
.toc                                powerscStd.rtc                    powerscStd.uiAgent
openpts.verifier                   powerscStd.svm                    powerscStd.uiServer
powerscStd.ice                     powerscStd.tnc_commands          powerscStd.vlog
powerscStd.license                 powerscStd.tnc_lib
powerscStd.msg                     powerscStd.tnc_pm
#
```

Figure 2-30 Installation path and filesets

### 3. Install UIAgent and license filesets by using smitty, as shown in Figure 2-31:

smitty installp

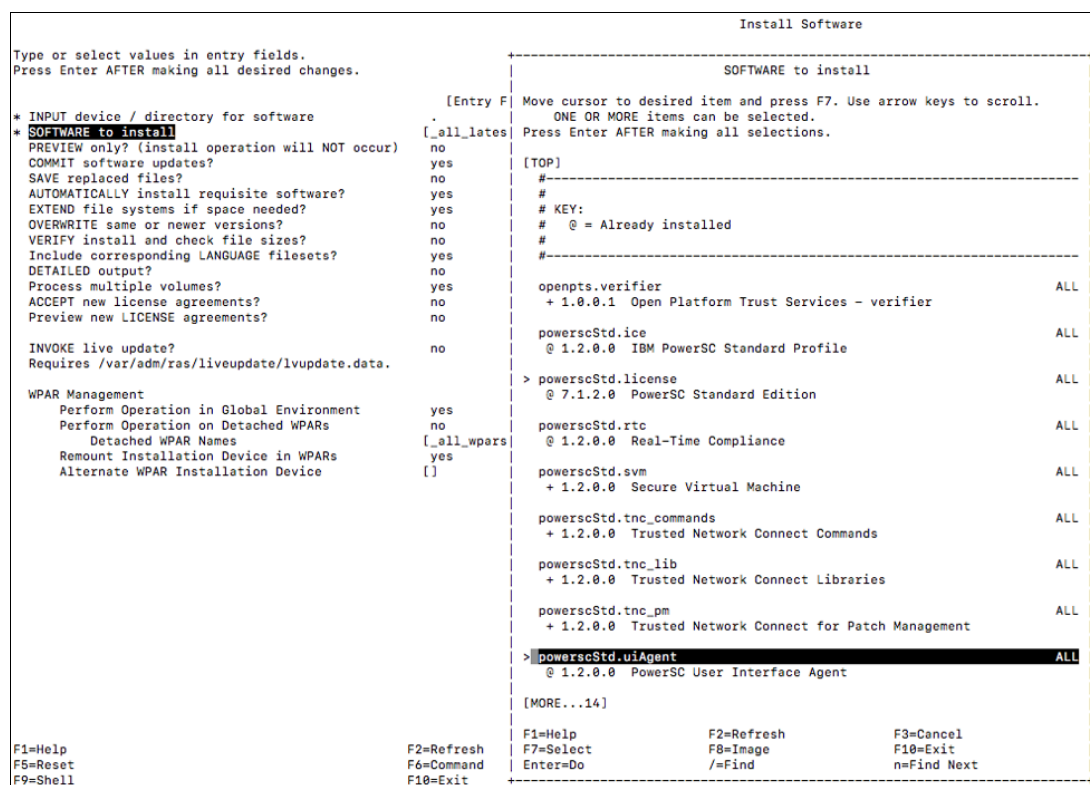


Figure 2-31 Install the UIAgent and license filesets

### 4. Verify the installation:

```
# ls1pp -L powerscStd.uiAgent.rtc
Fileset          Level State Type Description (Uninstaller)
-----
powerscStd.uiAgent.rtc 1.2.0.0 C    F    PowerSC User Interface Agent
```

The installation process automatically attempts to start the UIAgent process and creates an entry in /etc/inittab file:

```
# lsitab pscuiagent
pscuiagent:2:wait:/usr/bin/startsrc -s pscuiagent > /dev/console 2>&1
```

If the IBM PowerSC GUI server is not reachable or the server keystore is not present on the endpoint, the UIAgent start process fails and automatically stops.

For more information about generating the keystore, see 2.5.1, “Generate keystore” on page 43.

## 2.4.2 Installing UIAgent on RHEL

In this section, we describe how to install UIAgent on RHEL.

The following prerequisites must be met before IBM PowerSC GUI agent is installed:

- ▶ Install the IBM PowerSC pscxpert before installing the GUI agent. For more information about installing pscxpert, see step 3 next.
- ▶ The UIAgent needs redhat-lsb-core.

Complete the following steps to install UIAgent on RHEL:

1. Copy the PowerSC Standard Edition ISO file to the LPAR and loopmount it:

```
loopmount -i /software/POWERSC_STD_EDITION_V1.2_62018.iso -o "-V cdrfs -o ro"  
-m /mnt
```

2. Change the directory to /mnt/rhel:

```
# cd /mnt/rhel  
# ls  
comply-1.0-rhel6.ppc64.sh powersc-pscxpert-1.2.0.0-rhel7.ppc64le.sh  
powersc-uiAgent-1.2.0.0-rhel7.ppc64le.sh  
powersc-uiServer-1.2.0.0-rhel7.ppc64le.sh
```

3. Install pscxpert by running the powersc-pscxpert-1.2.0.0-rhel7.ppc64le.sh script, as shown in Figure 2-32.

```
# bash ./powersc-pscxpert-1.2.0.0-rhel7.ppc64le.sh  
x - created lock directory _sh29816.  
x - removed lock directory _sh29816.  
Preparing... #####  
[100%]  
Updating / installing...  
 1:powersc-pscxpert-1.2.0.0-rhel7 #####  
[100%]
```

Figure 2-32 Install pscxpert

4. Install UIAgent by running the powersc-uiAgent-1.2.0.0-rhel7.ppc64le.sh script, as shown in Figure 2-33.

```
# bash ./powersc-uiAgent-1.2.0.0-rhel7.ppc64le.sh  
x - created lock directory _sh30487.  
x - removed lock directory _sh30487.  
Preparing... ##### [100%]  
Updating / installing...  
 1:powersc-uiAgent-1.2.0.0-rhel7 ##### [100%]  
/opt/powersc/uiAgent /  
Created symlink from /etc/systemd/system/multi-user.target.wants/powersc-uiAgent.  
service to /usr/lib/systemd/system/powersc-uiAgent.service.
```

Figure 2-33 Script to install the UIAgent

5. Verify the installation by using the rpm command:

```
# rpm -qa | grep powersc  
powersc-uiAgent-1.2.0.0-rhel7.ppc64le  
powersc-pscxpert-1.2.0.0-rhel7.ppc64le
```



## 2.5 Endpoint administration

This section describes endpoint administration management.

### 2.5.1 Generate keystore

To work with the endpoints, some steps must be performed between the IBM PowerSC GUI server and IBM PowerSC GUI agent to communicate and discover the compliance level for AIX systems; for example, low, medium or high.

First, the truststore security certificate must be distributed to the endpoints. You or the system administrators must deploy the truststore security certificate on all endpoints.

Normally, a truststore file is created during the installation of the UI Server and all endpoints running the UI agent must be copied. The name of the truststore file is `endpointTruststore.jks`. The file is placed in the `/etc/security/powersc/uiServer/` directory on the UI Server.

**Note:** It is important to have the `/etc/hosts` or DNS setup configured correctly.

The following steps show how you must place the `endpointTruststore.jks` file on each endpoint for the IBM PowerSC GUI agent on that endpoint to make contact with the IBM PowerSC GUI server, and to start the process that results in the creation of the keystore on the endpoint.

It is possible to distribute the truststore file by using one of the following methods:

- ▶ Manually copy the `endpointTruststore.jks` file to each endpoint.
- ▶ If IBM PowerVC (or another virtualization manager) is used in your environment, the `endpointTruststore.jks` file can be put onto the IBM PowerVC image. When the IBM PowerVC image is deployed to an endpoint, the IBM PowerSC GUI agent and the truststore file are included. For more information, see 2.5.3, “IBM PowerVC integration” on page 48.

Complete the following steps by using the IBM PowerSC GUI server:

1. Go to the `/etc/security/powersc/uiServer` directory, as shown in Figure 2-34.

```
# cd /etc/security/powersc/uiServer
#
# ls
endpointTruststore.jks      serverKeystore.jks          uiServer.conf.properties
groups.txt                  serverTruststore.jks        uiServer.conf.properties~
passwords                    signingKeystore.jks
#
```

Figure 2-34 *uiServer* filesets

2. Copy the `endpointTruststore` file by using the `scp` command to the endpoint agent LPAR at the `/etc/security/powersc/uiAgent` directory. Then, start the agent at the endpoint.

AIX is shown in Figure 2-35.

```
# lssrc -s pscuiagent
Subsystem      Group          PID           Status
pscuiagent
#
# startsrc -s pscuiagent
0513-059 The pscuiagent Subsystem has been started. Subsystem PID is 17498530
#
# lssrc -s pscuiagent
Subsystem      Group          PID           Status
pscuiagent     17498530      active
```

Figure 2-35 Agent endpoint start for AIX

Linux is shown in Figure 2-36.

```
# systemctl start powersc-uiAgent.service
#
# systemctl status powersc-uiAgent.service
â powersc-uiAgent.service - PowerSC UI Agent
   Loaded: loaded (/usr/lib/systemd/system/powersc-uiAgent.service; enabled; vend
or preset: disabled)
   Active: active (running) since Thu 2018-09-13 16:57:50 EDT; 16s ago
   Main PID: 29757 (uiAgent.sh)
   CGroup: /system.slice/powersc-uiAgent.service
           ââ29757 /bin/sh /opt/powersc/uiAgent/bin/uiAgent.sh
           ââ29775 /opt/powersc/uiAgent/bin/uiAgent /opt/powersc/uiAgent/bin/j...

Sep 13 16:57:50 p52n76 systemd[1]: Started PowerSC UI Agent.
Sep 13 16:57:50 p52n76 systemd[1]: Starting PowerSC UI Agent...
Sep 13 16:57:50 p52n76 uiAgent.sh[29757]: Starting PowerSC UI Agent, and redi....
Hint: Some lines were ellipsized, use -l to show in full.
#
```

Figure 2-36 Agent endpoint start for Linux

The first time an endpoint starts running, the IBM PowerSC GUI agent uses the truststore file to determine where the IBM PowerSC GUI server is running. Then, the IBM PowerSC GUI agent sends a message to the IBM PowerSC GUI server with a request to join the list of available monitored endpoints.

3. After starting the agent, check the log file, as shown in Figure 2-37.

```
# tail -f /var/log/powersc/uiAgent/pscuiagent_2018-09-13_20-19.0.log
J9CL - 20180202_377933)
* AIX 7.2
*
*****
[20:19:31.925/000] <retrieveHostNameFromCertificate> : Hostname: p52n75.pbm.ihost
.com parsed from certificate with alias name psc_server in Store /etc/security/po
wersc/uiAgent/endpointTruststore.jks
[20:19:31.991/000] <main> : uname -n : p52n75
[20:19:32.003/000] : No /etc/security/powersc/uiAgent/endpointKeystore.jks file f
ound, requesting from UI Server
[20:19:33.647/000] <KeyStoreRequester.requestKeyStoreFile> : get current token of
requesting keystore: 0E48A3A6F2CA38B5
[20:19:34.049/000] : keystore file is not generated..
```

Figure 2-37 Check the log file

- Run the `pscuiserverctl setgroup pscadm "*"` command to add the UNIX groups that can manage the endpoints, as shown in Figure 2-38.

```
# pscuiserverctl setgroup pscadm "*"
pscadm=*
#
# cat /etc/security/powersc/uiServer/groups.txt
security=*
pscadm=*
#
```

Figure 2-38 Add the UNIX groups to manage the endpoints

It is important to know after you start the agent that the keystore request is sent to the GUI Server. The GUI Server administrator must generate and confirm the keystore for the endpoint. Complete the following steps:

- Go to IBM PowerSC GUI server, click the **Languages and Settings** icon in the menu bar of the main page.
- At the Configuration tab, click **Manage the Endpoints**, check the Keystore Requests, check the connection of the endpoint, and delete.

It is possible to verify the communication between discovered endpoints and the IBM PowerSC GUI server (see Figure 2-39).

**Important:** For each endpoint, you must verify that a keystore request is valid. If it is valid, you can generate a keystore for the endpoint.

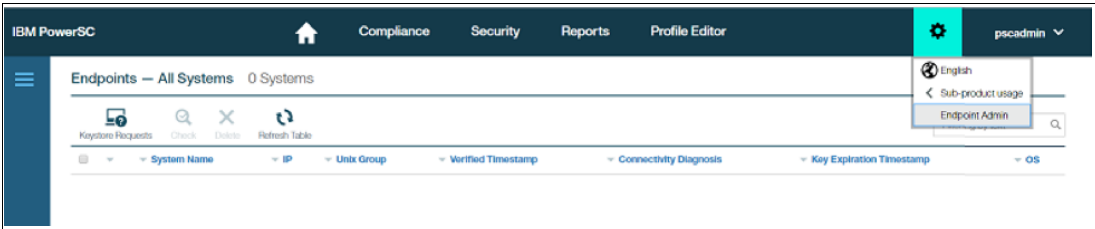


Figure 2-39 Check for all endpoints available for monitoring

- Select the endpoint, as shown in Figure 2-40.



Figure 2-40 Selecting the endpoint to request the keystore

4. Click **Proceed** to generate the endpoint keystore, as shown in Figure 2-41.

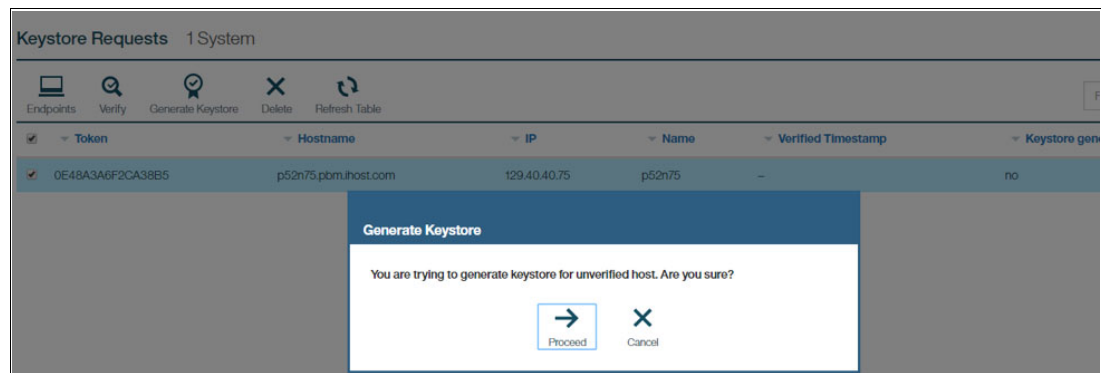


Figure 2-41 Generating the endpoint keystore

Confirm that the GUI is used and check that “yes” is displayed in the keystore generated field, as shown in Figure 2-42.

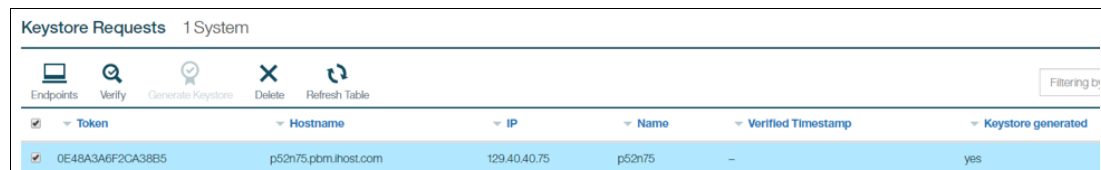


Figure 2-42 Checking for keystore generation for the endpoint

## 2.5.2 Security certificate expiration dates

The security certificate expiration dates can be checked for each endpoint. You can use the Endpoint Administrator tab to view the date that a security keystore certificate for an endpoint expires, as shown in Figure 2-43.

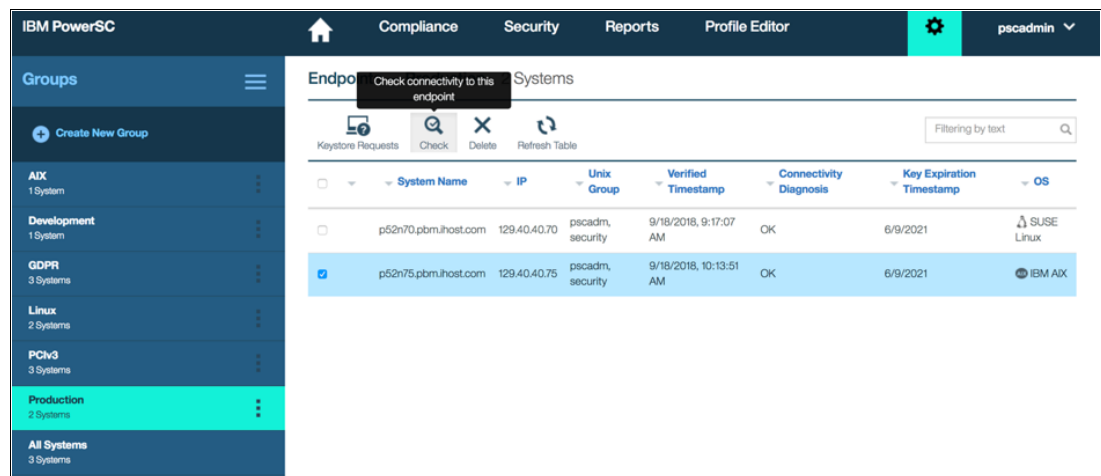


Figure 2-43 PowerSC shows the key expiration timestamp dates for the endpoints

Go to the Languages and Settings icon in the menu bar of the main page. Then, click the **Endpoint Admin** tab.

You can check that the date the security certificate expires is displayed in the Key Expiration Timestamp column, as shown in Figure 2-44. After the expiration date passes, it automatically generates a request for a new security certificate.

System Name	IP	Unix Group	Verified Timestamp	Connectivity Diagnosis	Key Expiration Timestamp	OS
p52n70.pbm.ihost.com	129.40.40.70	pscadm, security	9/18/2018, 2:00:18 PM	OK	6/9/2021	SUSE Linux
p52n75.pbm.ihost.com	129.40.40.75	pscadm, security	9/18/2018, 1:55:22 PM	OK	6/9/2021	IBM AIX

Figure 2-44 Endpoint production details

In this view, you also can confirm and verify whether the generated keystore request is valid. If it is not valid, you can generate a keystore for the endpoint.

**Note:** Before the expiration date is reached, you can preemptively delete the current certificate for the endpoint and restart the IBM PowerSC GUI agent on that endpoint.

By deleting the certificate for the endpoint, when IBM PowerSC GUI agent is restarted, it appears that the endpoint is new, and as such, IBM PowerSC automatically generates a request for a new security certificate.

By using the Endpoint Administrator Keystore Requests page, you can verify that the generated keystore request is valid. If it is valid, you can generate a keystore for the endpoint.

For more information about each option, see the [Administering endpoint and server communication page](#) of IBM Knowledge Center.

## Creating more security certificates

By using the IBM PowerSC GUI server, you can use shell scripts to create or import security certificates that can be found in the `/opt/powersc/uiServer/bin/` directory:

```
generate_server_keystore_uiServer.sh
generate_signing_keystore_uiServer.sh
generate_endpoint_keystore_uiServer.sh
import_well_known_certificate_uiServer.sh
convertProfileToBean.sh
```

For more information about each script, see [this website](#).

The following keystores are required and are created by one or more of the shell scripts that are run during the installation process or by the IBM PowerSC administrator:

- ▶ endpointKeystore.jks
- ▶ endpointTruststore.jks
- ▶ serverKeystore.jks

- ▶ serverTruststore.jks
- ▶ signingKeystore.jks

For more information about the shell scripts that are provided by PowerSC to generate the certificate, [this page](#) of IBM Knowledge Center.

You also can purchase your certificate so that you can use the script `import_well_known_certificate_uiServer.sh`. Run this script only if you are providing your own well-known certificate.

If you have a certificate .pem file from a well-known certificate authority, you can run this script to create the endpoint truststore, import that certificate, and create the GUI server truststore, the truststore, and the GUI server keystore.

### 2.5.3 IBM PowerVC integration

It is common that companies have some kind of cloud project or are preparing to have a private cloud. Therefore, the system administrators can use a virtualization manager, such as IBM PowerVC, to copy the truststore file onto each new endpoint by using an image that contains the IBM PowerSC agent and the truststore file.

To integrate IBM PowerVC, you must complete some steps on the IBM PowerVC and on the IBM PowerSC. Therefore, you might ask for help from the IBM PowerVC administrator.

First, you must copy the truststore file that is created and must be used by all endpoints. The name of the file is `endpointTruststore.jks` and it is in the `/etc/security/powersc/uiServer/` directory.

The `endpointtruststore.jks` file must be placed on each endpoint so that the IBM PowerSC GUI agent on that endpoint to make contact with the IBM PowerSC GUI server and to start the process that results in the creation of the keystore on the endpoint.

As the example and for our scenario, we copied the endpoint truststore `/etc/security/powersc/uiServer/endpointTruststore.jks` file to the IBM PowerVC image. Then, we deployed the IBM PowerVC image as a new endpoint.

For more information about the variable `OS_AUTH_URL` that appears at the file `/opt/ibm/powervc/powervcrc` file, contact your IBM PowerVC administrator. You can have something similar to our environment:

```
https://powervc2.pbm.ihost.com:5000/v3/auth
```

Then, go to the IBM PowerSC Server and run the following commands:

```
p52n75:/opt/powersc/uiServer/bin #
p52n75:/opt/powersc/uiServer/bin # ./pscuiserverctl set powervcKeystoneUrl
p52n75:/opt/powersc/uiServer/bin #
p52n75:/opt/powersc/uiServer/bin # ./pscuiserverctl set powervcKeystoneUrl
https://powervc2.pbm.ihost.com:5000/v3/auth/
powervcKeystoneUrl=https://powervc2.pbm.ihost.com:5000/v3/auth/
p52n75:/opt/powersc/uiServer/bin # ./pscuiserverctl set powervcKeystoneUrl
https://powervc2.pbm.ihost.com:5000/v3/auth/
p52n75:/opt/powersc/uiServer/bin #
```

# IBM PowerVC GUI

Using the IBM PowerVC GUI and with administrator or deployment permissions as shown in Figure 2-45, we created a gold image with the IBM PowerSC agents files, and the endpointTruststore.jks file on it. (This process is not described in this book.)

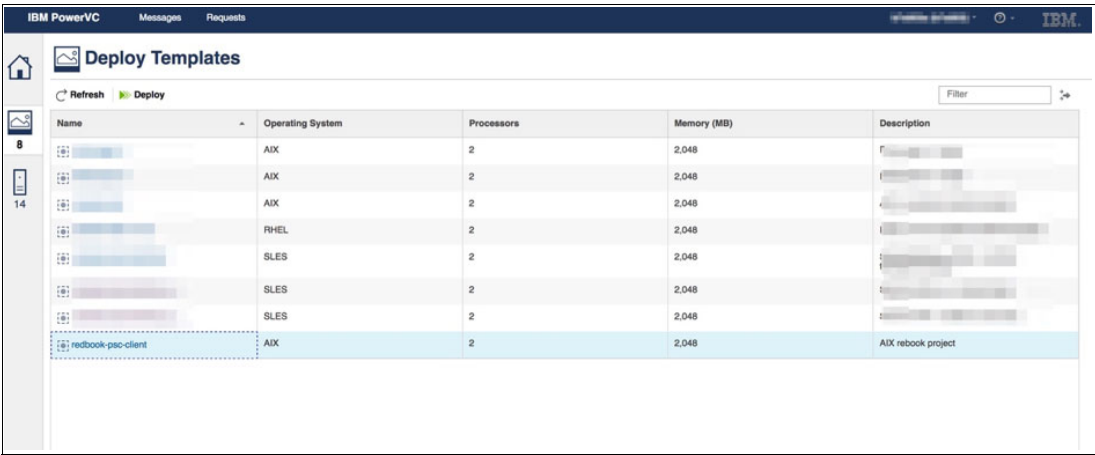


Figure 2-45 IBM PowerVC GUI


In our scenario, we created an image with AIX 7.2 called redbook-psc-client. Selecting this image, click the **Deploy** option to start the process through the IBM PowerVC GUI.

You must complete the information that you want for your new machine. Enter the name of the virtual machine. In our case, we entered `redbook-psc-client_deploy` and one instance by adding a single description, as shown in Figure 2-46.


IBM PowerVC

Messages

Requests




8



14

Deploy Templates

Deploy redbook-psc-client



Deploy redbook-psc-client

Deploy Template Specifications

Operating system:	AIX
Processors:	2
Processing units:	0.2
Memory (MB):	2,048

Specifications

\* Virtual machine name:

redbook-psc-client\_deploy

\* Instances:

1

Key pair: ?

None

Virtual machine description:

Deploy PowerSC Agent Client

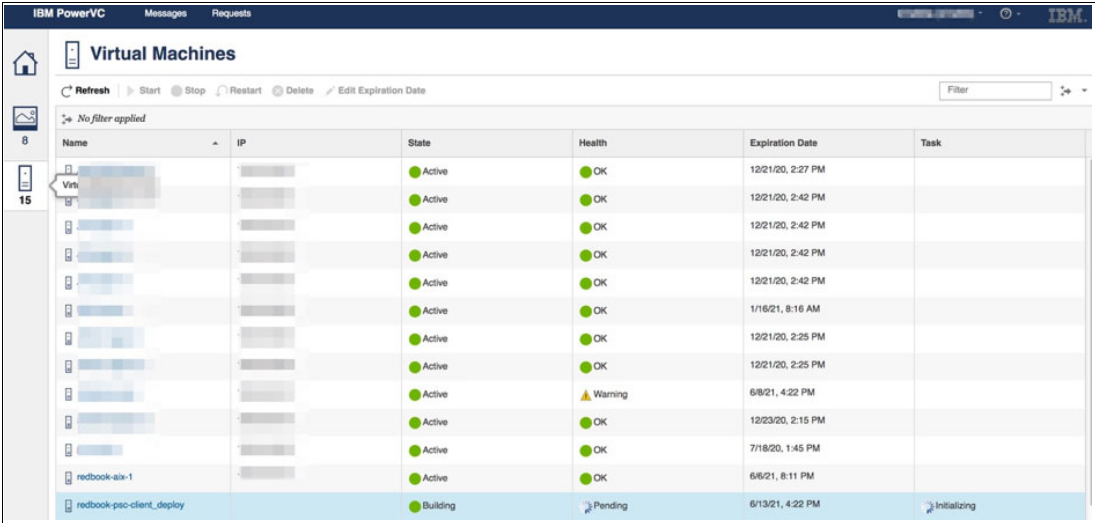
Message to administrator: ?

Figure 2-46 IBM PowerVC GUI creating the deployment image

50 Simplify Management of Security and Compliance with IBM PowerSC in Cloud and Virtualized Environments



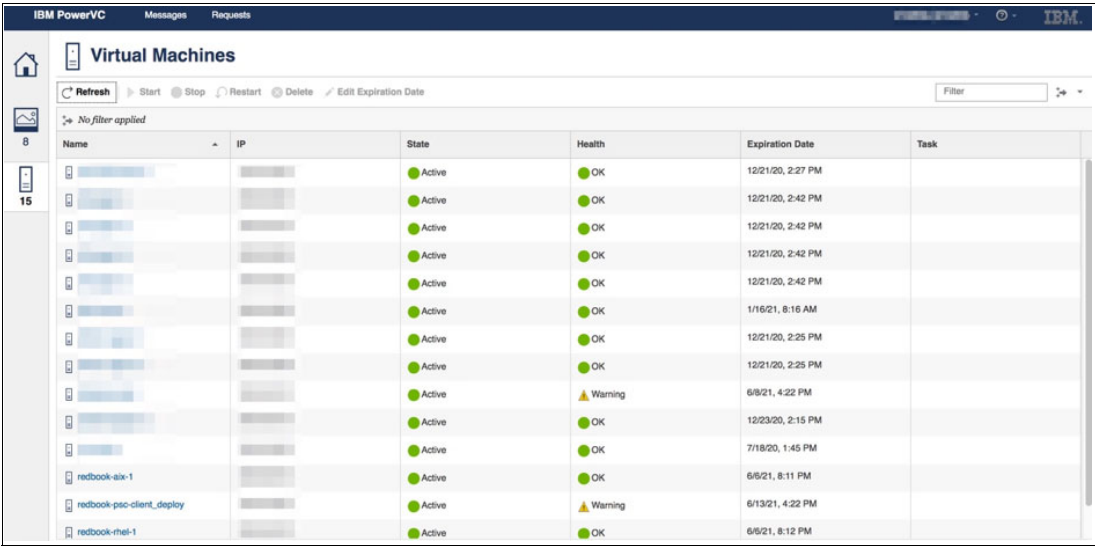
After starting the process, you can see in the Virtual Machines tab that our new machines appear as “Building” state, as shown in Figure 2-47.



IBM PowerVC Messages Requests						
Virtual Machines						
Refresh Start Stop Restart Delete Edit Expiration Date Filter						
No filter applied						
Name	IP	State	Health	Expiration Date	Task	
Virt...		Active	OK	12/21/20, 2:27 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	1/16/21, 8:16 AM		
		Active	OK	12/21/20, 2:25 PM		
		Active	OK	12/21/20, 2:25 PM		
		Active	Warning	6/8/21, 4:22 PM		
		Active	OK	12/23/20, 2:15 PM		
		Active	OK	7/18/20, 1:45 PM		
redbook-ala-1		Active	OK	6/6/21, 8:11 PM		
redbook-pso-client_deploy		Building	Pending	6/13/21, 4:22 PM	Initializing	

Figure 2-47 Virtual machines states

This process normally takes 8 - 10 minutes to complete. You can check that the new machine now shows as Active, as shown in Figure 2-48.



IBM PowerVC Messages Requests						
Virtual Machines						
Refresh Start Stop Restart Delete Edit Expiration Date Filter						
No filter applied						
Name	IP	State	Health	Expiration Date	Task	
		Active	OK	12/21/20, 2:27 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	12/21/20, 2:42 PM		
		Active	OK	1/16/21, 8:16 AM		
		Active	OK	12/21/20, 2:25 PM		
		Active	OK	12/21/20, 2:25 PM		
		Active	Warning	6/8/21, 4:22 PM		
		Active	OK	12/23/20, 2:15 PM		
		Active	OK	7/18/20, 1:45 PM		
redbook-ala-1		Active	OK	6/6/21, 8:11 PM		
redbook-pso-client_deploy		Active	Warning	6/13/21, 4:22 PM		
redbook-rhel-1		Active	OK	6/6/21, 8:12 PM		

Figure 2-48 Virtual machines shows active state

In the Configuration tab of IBM PowerSC Server GUI, the new endpoint is created through the IBM PowerVC as Keystore Request. Here, we can see the token ID, hostname, IP address, and if the keystore was granted, as shown in Figure 2-49.

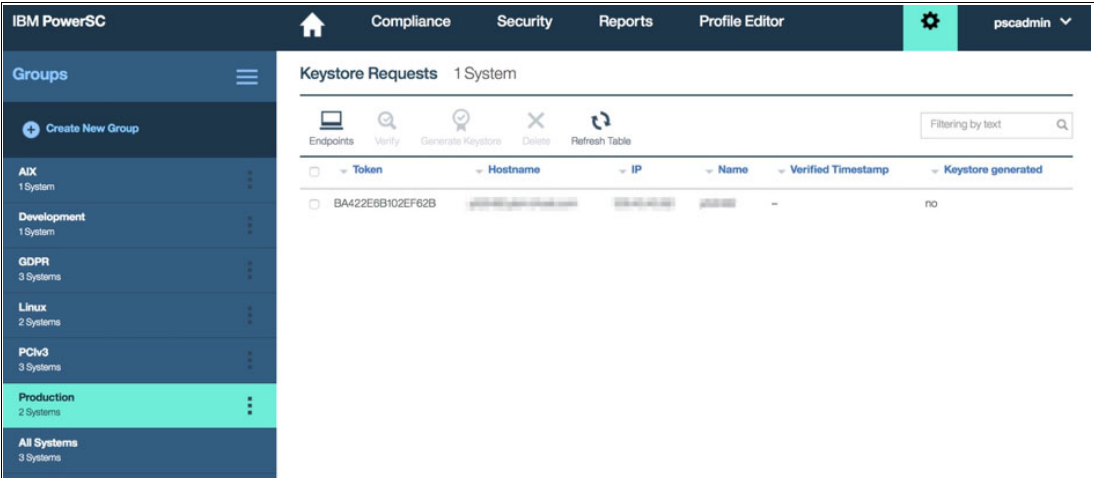


Figure 2-49 PowerSC GUI new endpoint created

By selecting the machine, the Verify, Generate Keystore, Delete, and Refresh Table options become available. Then, click **Verify** to confirm that these machines are created by IBM PowerVC, as shown in Figure 2-50.

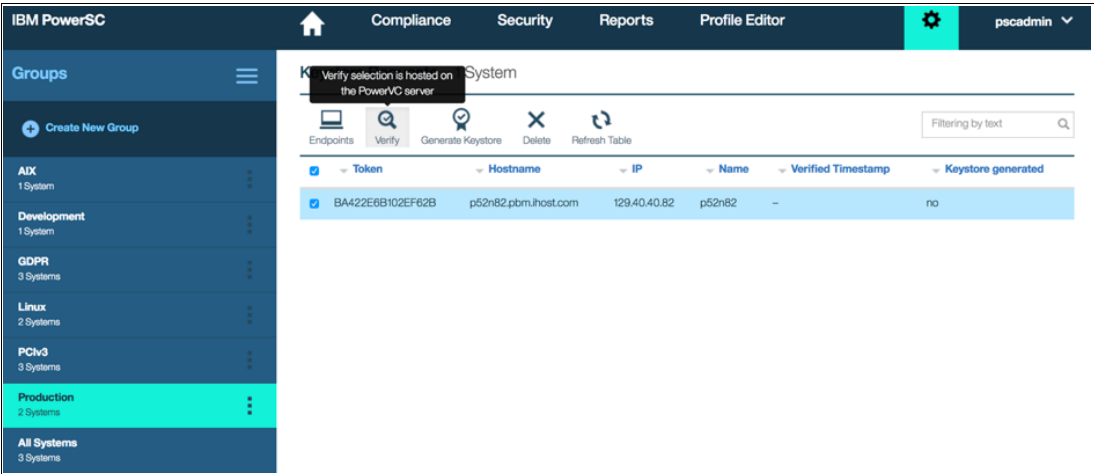


Figure 2-50 PowerSC GUI machine verification pane

When you use this option, the IBM PowerVC Administrator credentials must be entered, as shown in Figure 2-51.

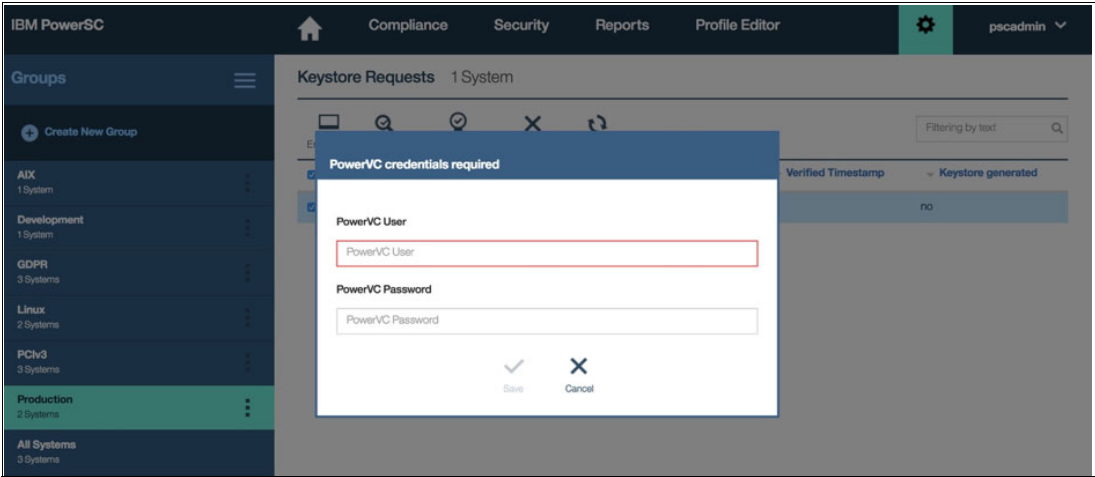


Figure 2-51 IBM PowerSC GUI entering the IBM PowerVC credentials

After confirming that the machines are created by IBM PowerVC, the keystore to send the GUI Server must be generated, as shown in Figure 2-52.

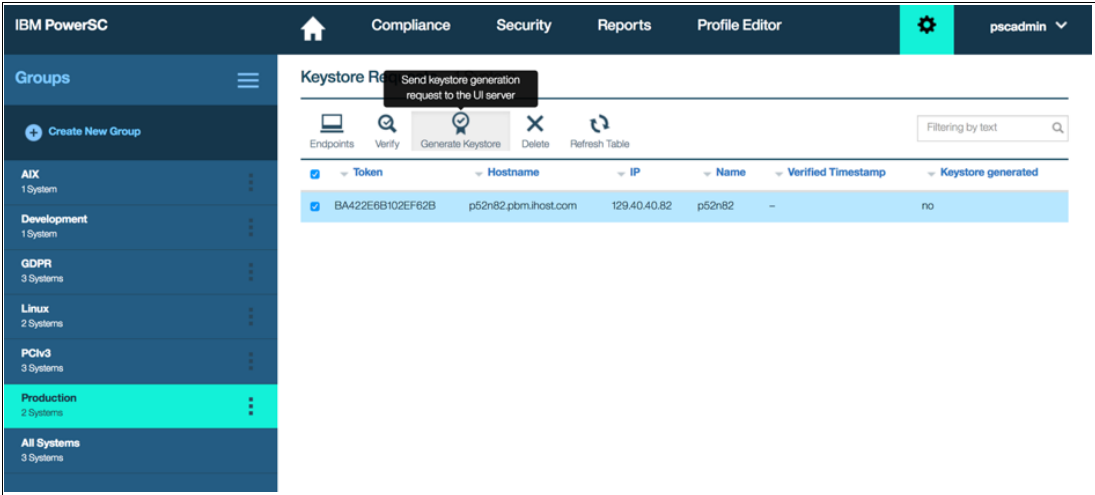


Figure 2-52 IBM PowerSC GUI generating the Keystore to send to UI server

This operation always is necessary if the endpoint is not discovered automatically by the IBM PowerSC Server, as shown in Figure 2-53.

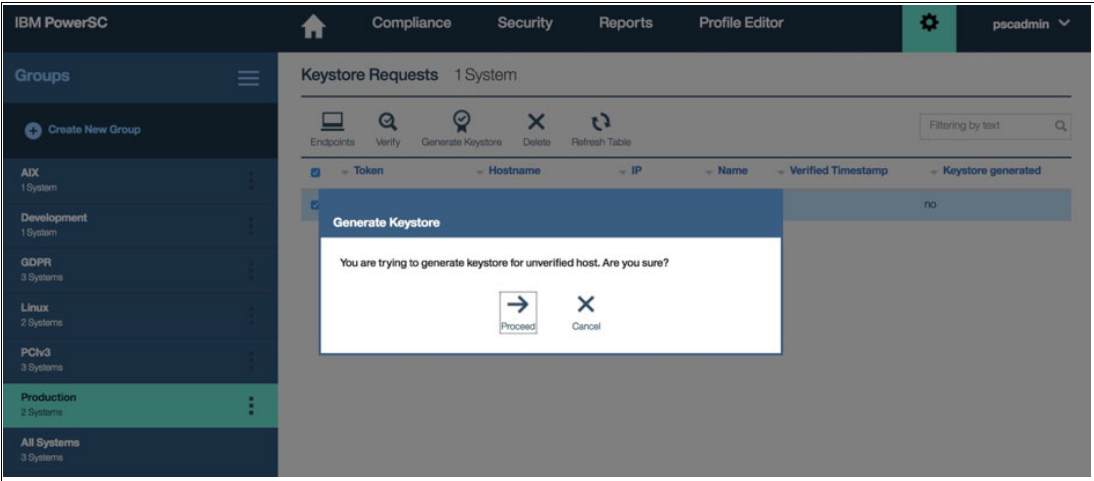


Figure 2-53 Generating Keystore verification

You can see now that the keystore must be generated after confirming that this machine was created by IBM PowerVC. The Keystore Generate tab changed to Yes, as shown in Figure 2-54.

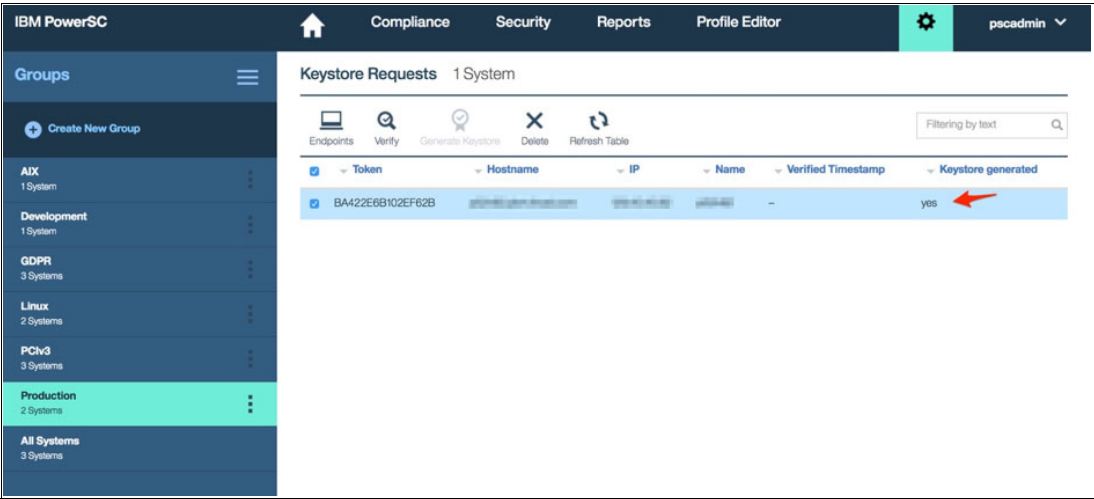


Figure 2-54 IBM PowerSC pane view showing the Keystore generated

Also, the status of the new endpoint features the updated timestamp, as shown in Figure 2-55.

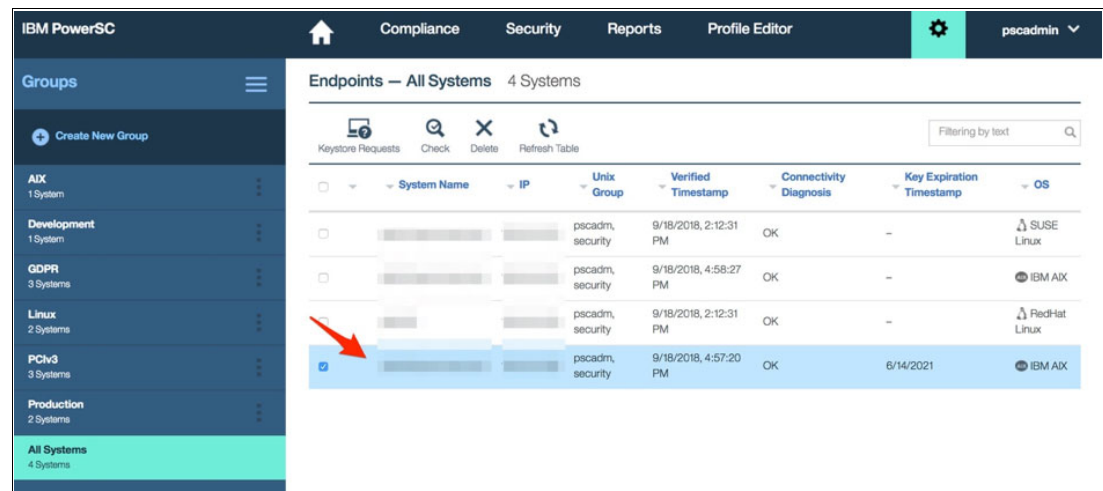


Figure 2-55 IBM PowerSC GUI shows endpoint timestamp updated

## 2.6 Managing groups in IBM PowerSC GUI

With IBM PowerSC GUI, you can categorize endpoints into groups. This feature is useful when you want to segregate endpoints that are based on workload types (such as application and database), or if you want to segregate based on environment type, such as production, quality, and testing.

This section describes how to create and manage groups by using IBM PowerSC GUI.

### 2.6.1 Creating groups

To create a group for administration purposes, go to the Compliance tab and click the horizontal line ellipse in the navigation pane to open the Group Editor. You see all endpoints that were discovered (in our example, three).

Also, the Group Editor can be accessed by using the Compliance and Security tabs, as shown in Figure 2-56.

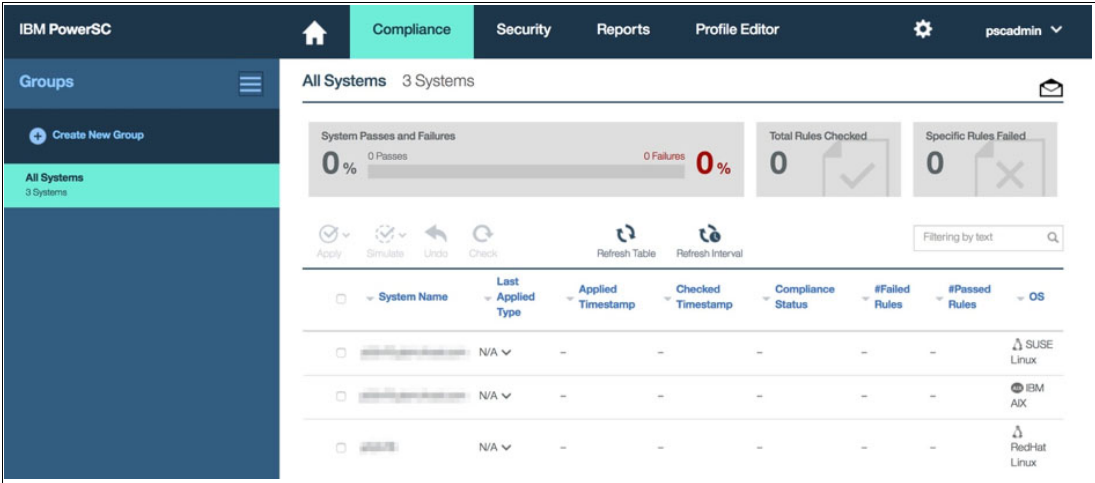


Figure 2-56 PowerSC Compliance tab - Create New Group

To create a custom group, click the **Plus** sign and enter the name of the Group. In our example, we created two groups (AIX and Linux) to manage the AIX and Linux endpoints.

**Note:** The name of group must be unique and can be up to 128 characters.

Creating an AIX group

Select the machine, move to the group, and save, as shown in Figure 2-57.

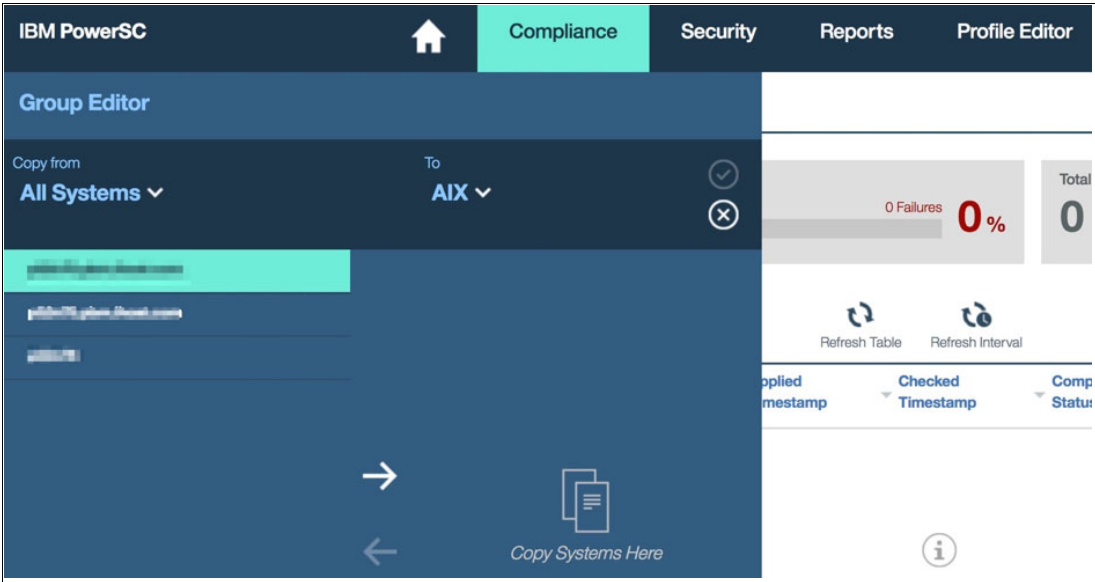


Figure 2-57 Creating the AIX group

After it is saved, you see that the group was created, as shown in Figure 2-58.

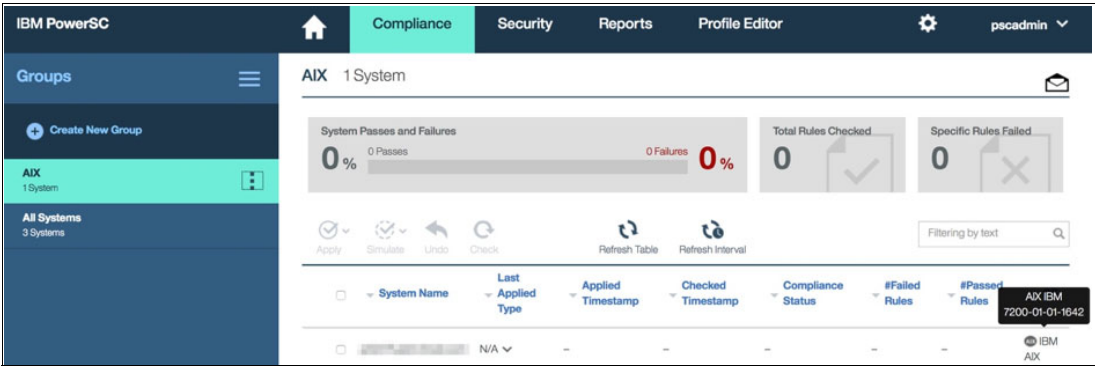


Figure 2-58 AIX group created

Creating a Linux group

Repeat the same steps to create the Linux group. Figure 2-59 shows the created Linux group.

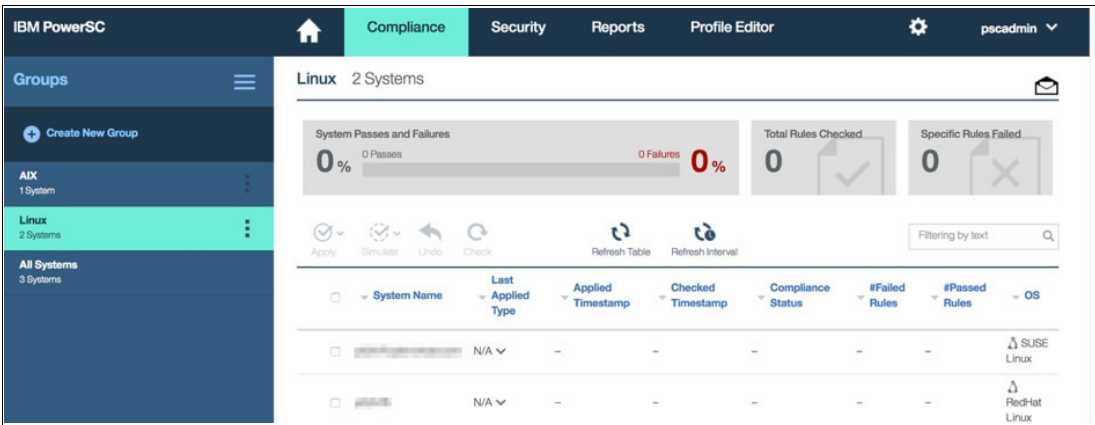


Figure 2-59 Linux group created

Also, you can create as many groups as needed to fulfill your requirements. To demonstrate our scenario, we created a few more groups: Development, Production, PCIv3, and GDPR.

One endpoint can be part of more than one group at a time. Figure 2-60 shows all of the groups that were created.

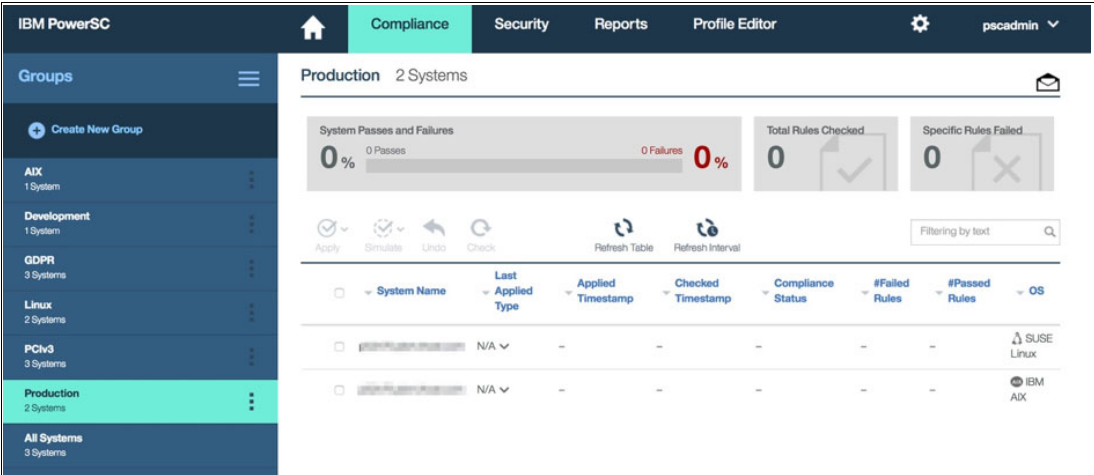


Figure 2-60 More groups created

### 2.6.2 Renaming groups

The rename group option is used only when you want to change the name of a group. Select the **Groups** tab and select the ellipse to the right of the group that you want to rename. Click **Rename Group**, as shown in Figure 2-62 on page 59. Specify the new name for the group in the Group name field.

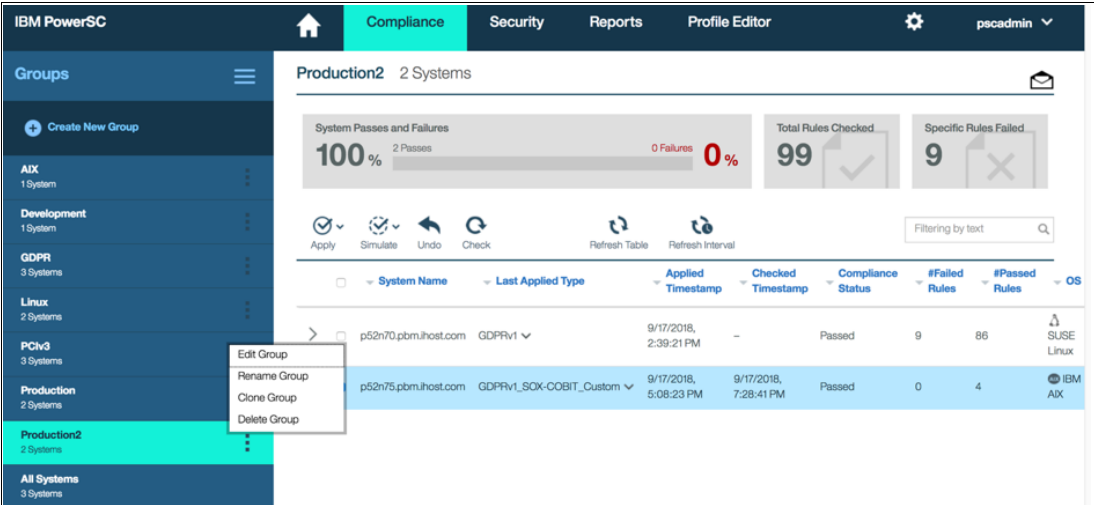


Figure 2-61 Editing or renaming a group

### 2.6.3 Editing groups

Using the Edit Group option, you can add or remove endpoints from a group. Select the group that you want to edit, for example:

- To add an endpoint system to the group, select the system from the All Systems list and click the right arrow.



- To remove an endpoint from the group, select the system from the GroupName list and click the left arrow (see Figure 2-62).

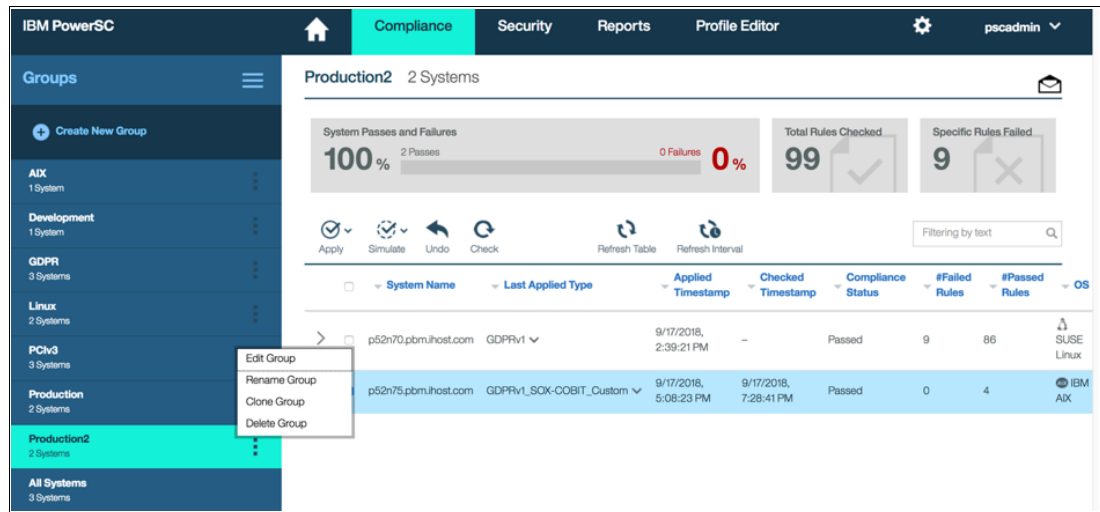


Figure 2-62 Editing or renaming a group

After making the changes, click the **Save group changes** option to save your changes, as shown in Figure 2-63.

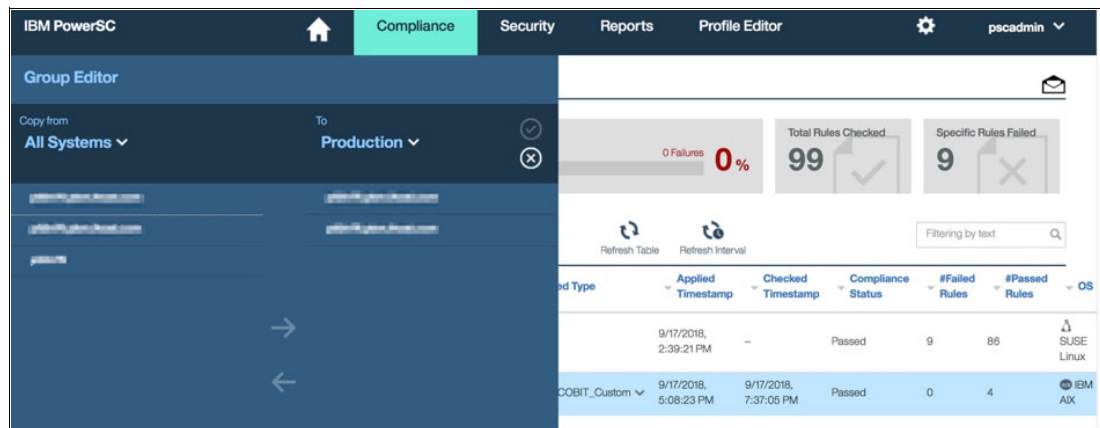


Figure 2-63 Save the group changes pane

## 2.6.4 Cloning groups

You can use this option to clone a group to create a duplicate with the same endpoints and a new name.

## 2.6.5 Deleting a group

You can use this option to delete groups that are no longer applicable (see Figure 2-62).

Go to the Group Editor from the Security or Compliance tabs. Select the group that you want to delete. Click **Delete Group**. The group is deleted and removed from the list of groups in the Groups tab.

## 2.7 IBM PowerSC GUI server features

This section describes various features that are available in the IBM PowerSC GUI server.

### 2.7.1 Home tab

The Home tab of the IBM PowerSC GUI provides a high-level view of the compliance and FIM events, as shown in Figure 2-64. The GUI provides a quick view of the following information:

- ▶ How many systems are enrolled with IBM PowerSC GUI server
- ▶ How many endpoints include compliance failures
- ▶ How many compliance rules checked and failed
- ▶ How many systems are without any profile applied
- ▶ Total number of RTC, Audit, and TE events

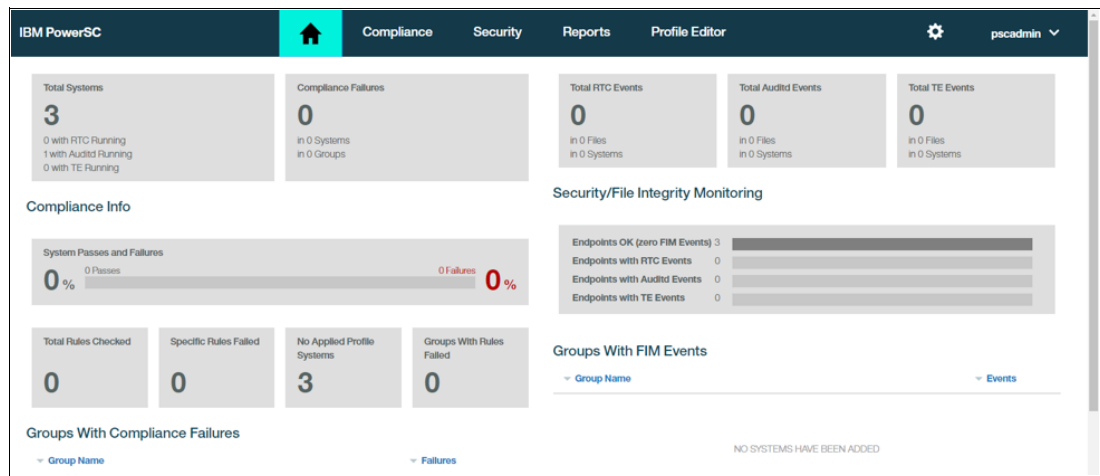


Figure 2-64 PowerSC GUI Home tab

### 2.7.2 Compliance tab

The Compliance tab provides options to manage hardening the enrolled endpoints. You can perform the following tasks:

- ▶ Apply a profile to the endpoint
- ▶ Undo a profile
- ▶ Check compliance against a selected profile by using the Simulate option
- ▶ Check compliance against an applied profile by using the Check option

To perform a task, you can select the endpoints and click **Apply**, **Simulate**, **Undo**, or **check**, as shown in Figure 2-65.

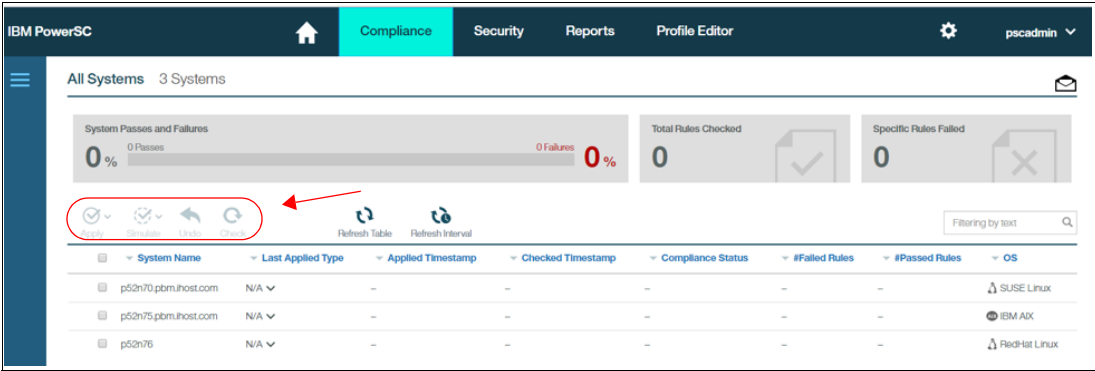


Figure 2-65 PowerSC Compliance tab

For more information about the Compliance tab, see Chapter 3, “Compliance automation” on page 75.

### 2.7.3 Security tab

The Security tab shows more information about FIM events. You find four columns that provide information about each endpoint. The first three columns are for the following FIM events, as shown in Figure 2-66:

- ▶ RTC
- ▶ Auditd
- ▶ Trusted Execution

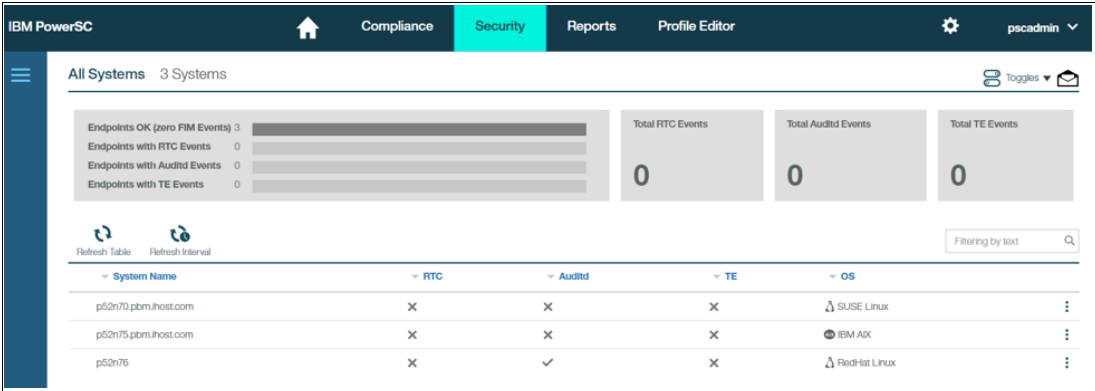


Figure 2-66 PowerSC Security tab

The fourth column shows the operating system name. For AIX endpoints, the FIM events are shown under the RTC and the TE columns. For Linux endpoints, the FIM events are shown under the Auditd column.

You can use this page to view FIM alerts or configure RTC, Auditd, and TE for the endpoints. For more information about configuring FIM, see Chapter 4, “Real-Time File Integrity Monitoring” on page 111.

## 2.7.4 Reports tab

IBM PowerSC GUI provides an excellent reporting facility. You can configure separate email IDs for overview emails or detail emails. The overview emails contain only high-level information that can be relevant to managers. The detail emails include low-level information that can be relevant to the system administrators.

Next, we describe the options that are available in the Reports tab.

### Compliance Overview

The Compliance Overview section can be configured to receive reports that are related to high-level overview compliance, as shown in Figure 2-67.

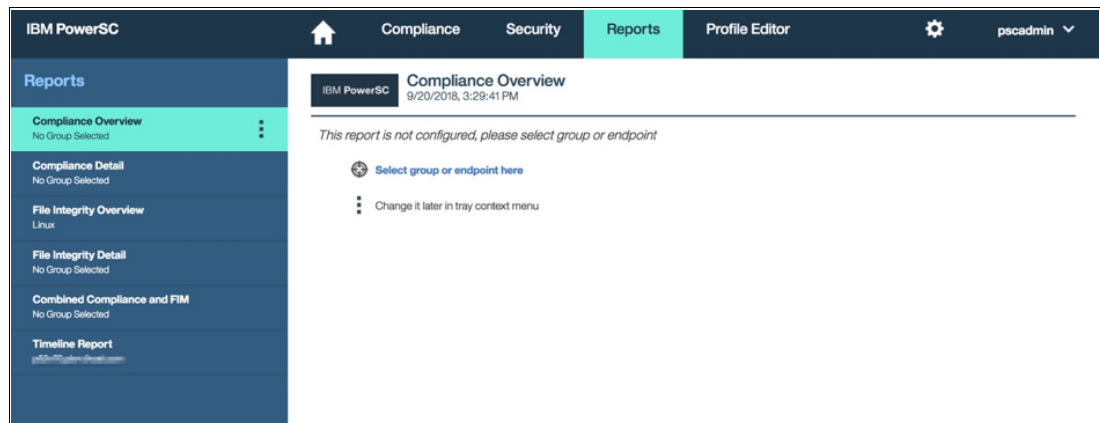


Figure 2-67 PowerSC Reports Compliance Overview pane

To configure this report, click **Compliance Overview** and select the group or endpoints for which you want to configure the report, as shown in Figure 2-68.

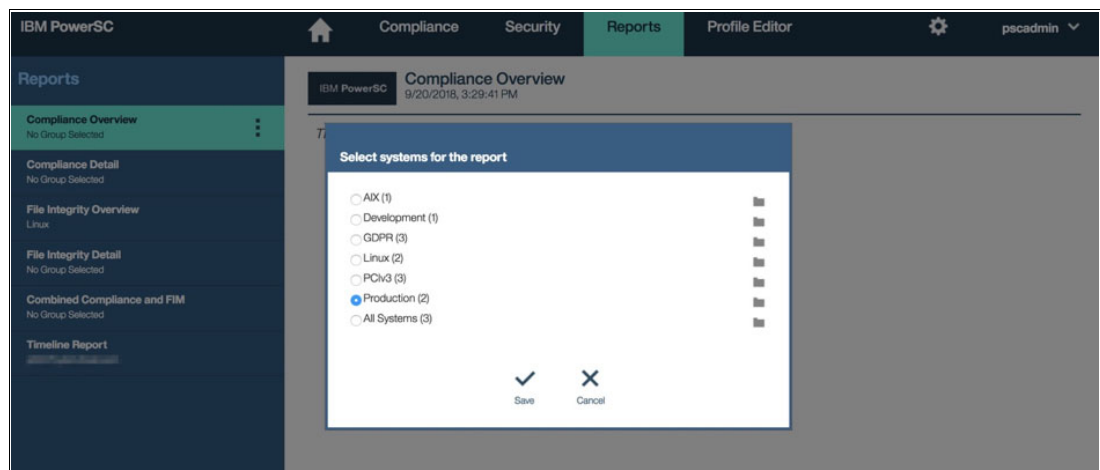


Figure 2-68 Configuring the report

Figure 2-69 shows the report for the Production group.

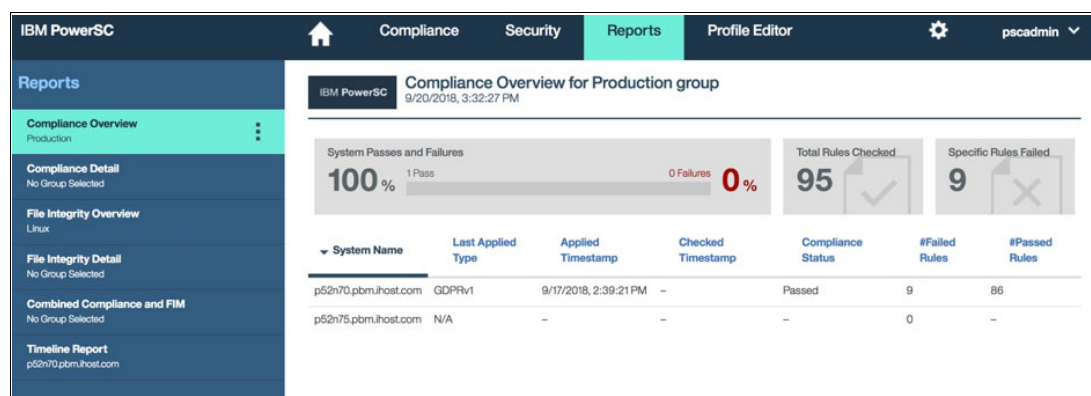


Figure 2-69 Compliance Overview pane for Production group

The following options are available to configure email-based reporting:

- ▶ Email options: Send daily emails
- ▶ Send immediately: Send immediate email

Click **Email options**, as shown in Figure 2-70.

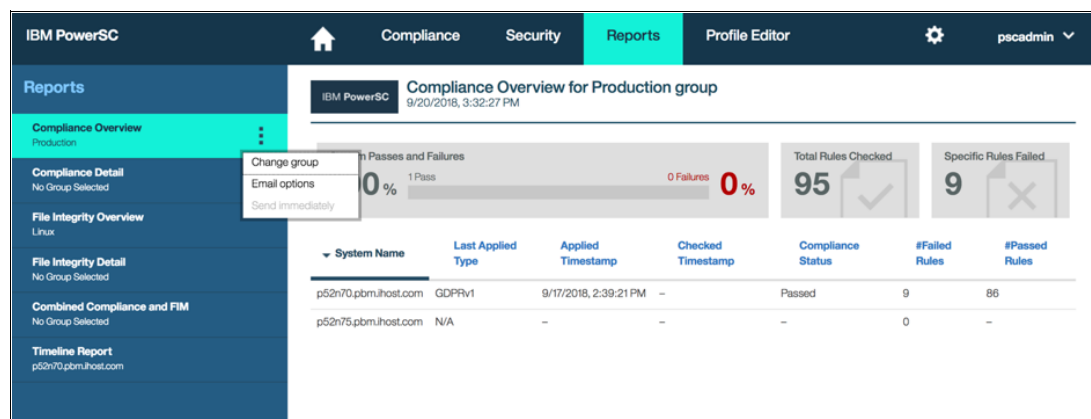


Figure 2-70 Email options pane

This tab shows the email configuration pane. The following options are available:

- ▶ Send me e-mails: Select this option.
- ▶ Addresses [comma separated]: Specify email address to where you want to receive the reports. To specify more than one email address, specify multiple email IDs that must be comma-separated.
- ▶ Subject: Specify the subject for the email.
- ▶ Send every day at: Specify the hours and minutes when you want to receive the email.

After completing this information, click **Save**. Daily emails are then configured, as shown in Figure 2-71 on page 64.

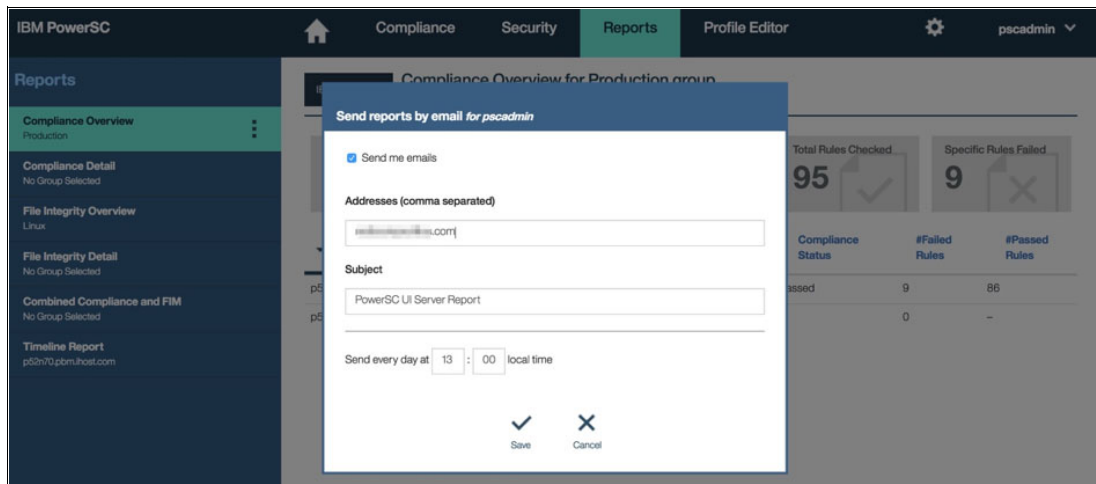


Figure 2-71 Send report by email pane

Figure 2-72 shows a sample Compliance Overview email.

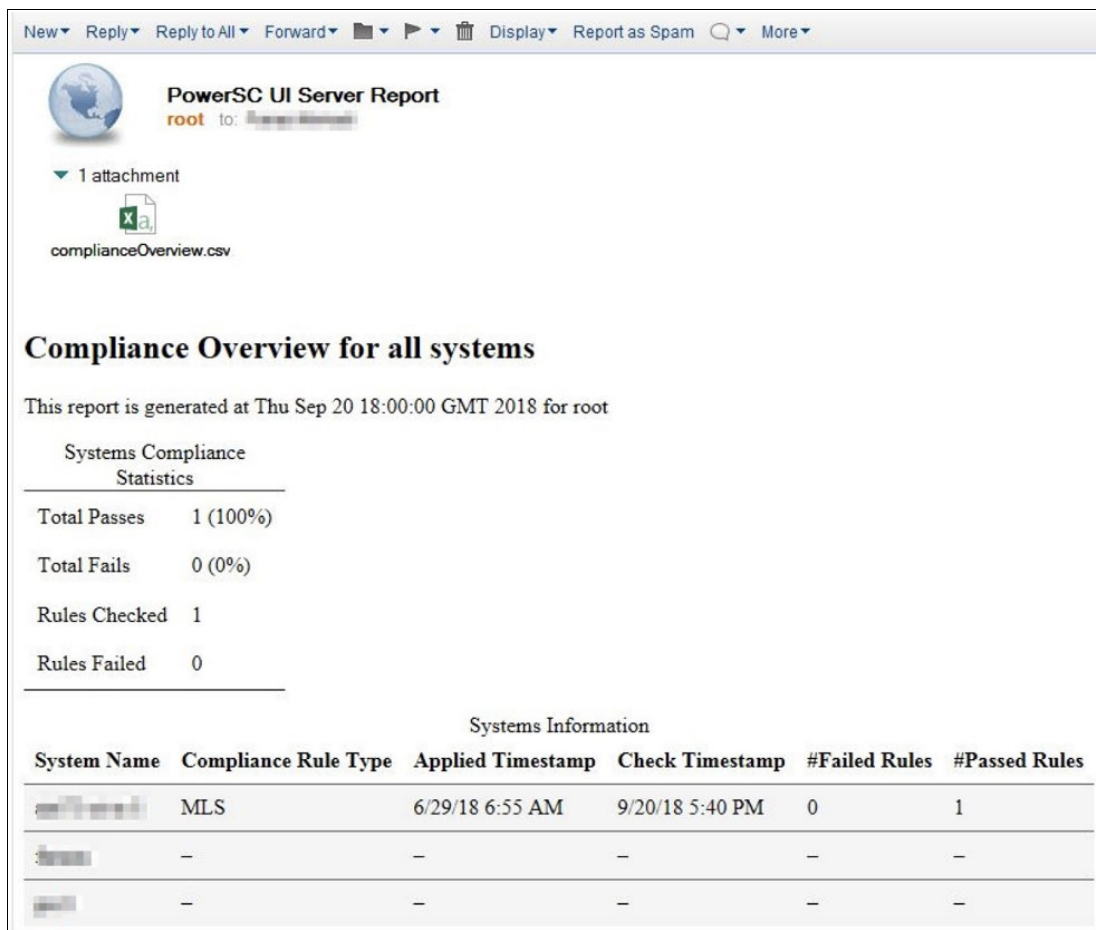


Figure 2-72 Compliance Overview email for all systems

## Compliance detail

You can configure the Compliance Detail reporting to get low-level details about compliance failures. As with the Compliance Overview, you can specify the group and configure automated email options for this group, as shown in Figure 2-73.

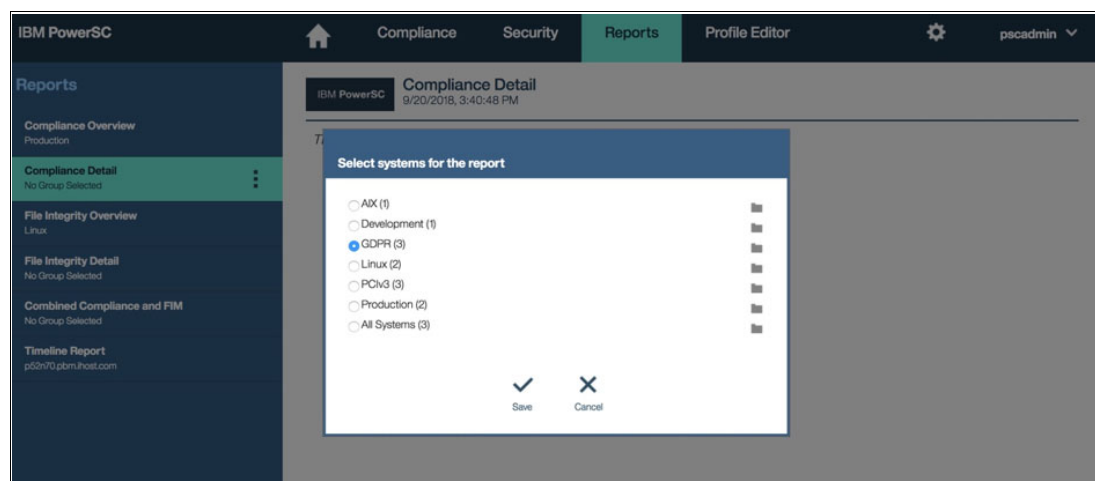


Figure 2-73 Selecting GDPR for the report

Figure 2-74 shows the compliance report for the GDPR group.

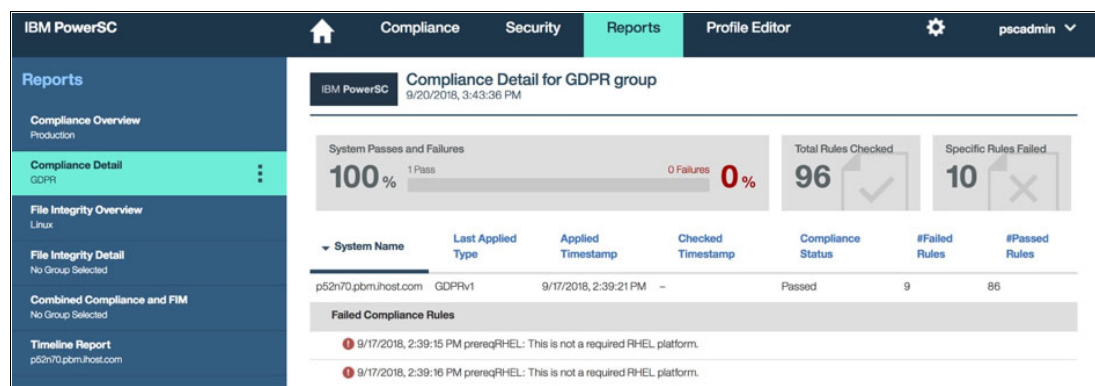


Figure 2-74 Compliance detail report for GDPR group

## FIM integrity overview

You can configure the File Integrity Overview reporting to get high-level information about FIM events. As with the Compliance Overview, you can specify the group and configure automated email options for this group, as shown in Figure 2-75.

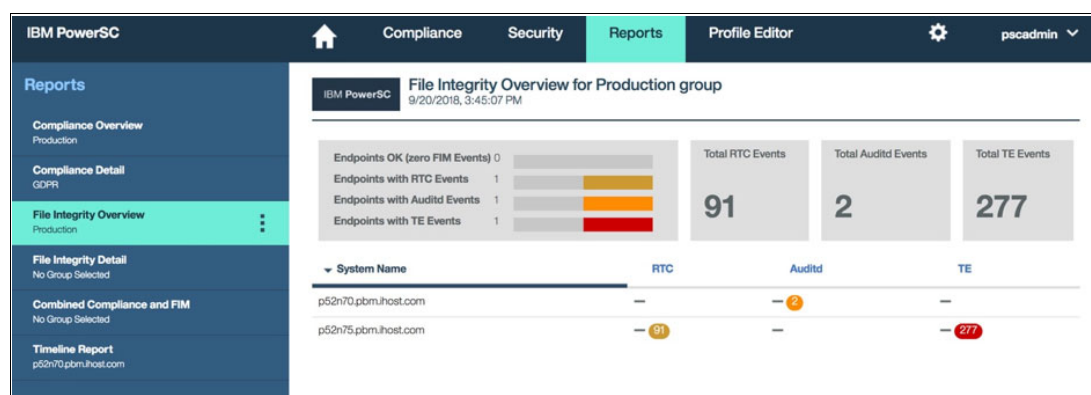


Figure 2-75 PowerSC File Integrity Overview for Production group

## FIM integrity detail

You can configure the File Integrity Detail reporting to get low-level information about FIM events. As with the Compliance Overview, you can specify the group and configure automated email options for this group, as shown in Figure 2-76.

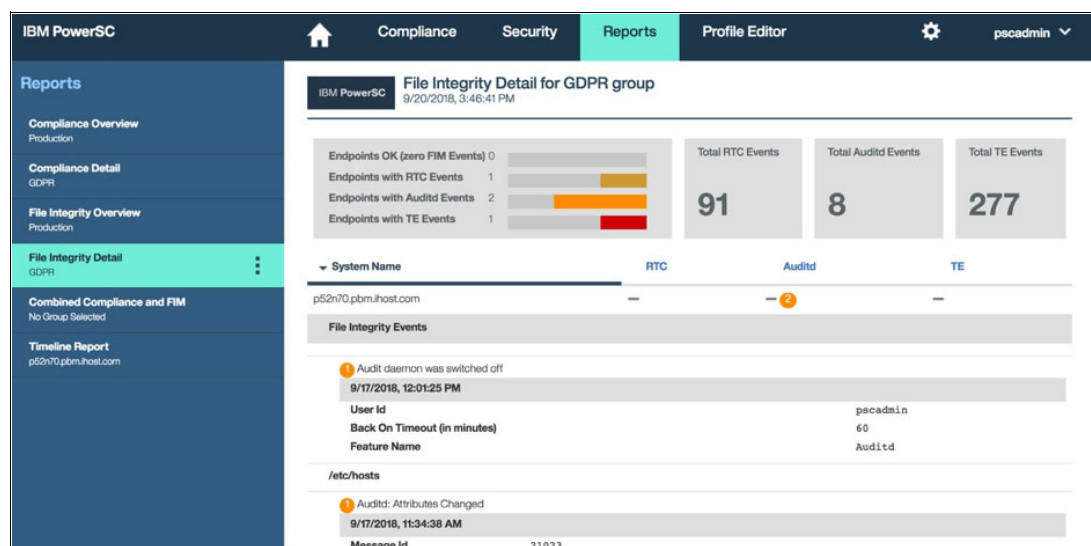


Figure 2-76 File Integrity Detail for GDPR group



## Combined Compliance and FIM

You can combine compliance and FIM reporting in one location by using this section. As with the Compliance Overview, you can specify the group and configure automated email options for this group, as shown in Figure 2-77.

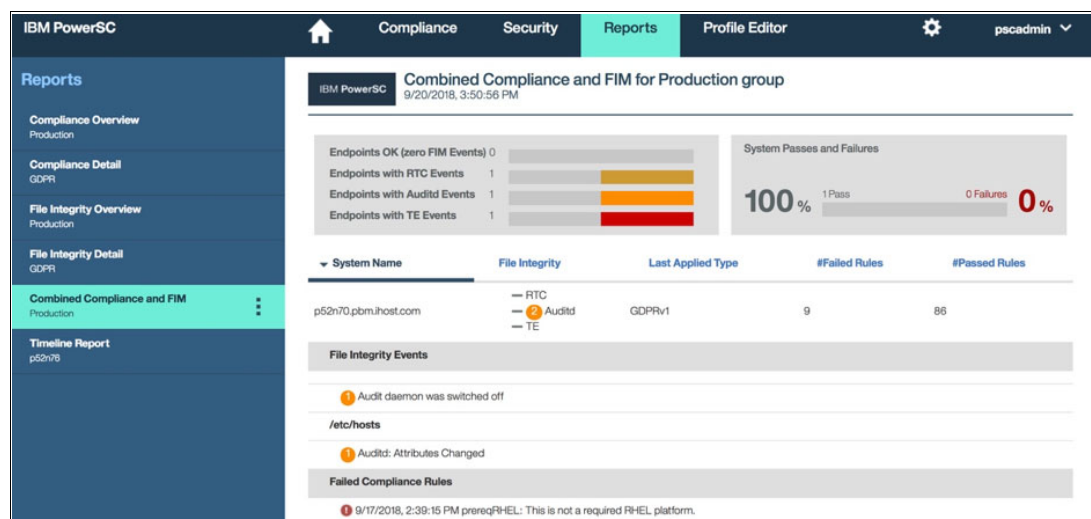


Figure 2-77 Combined Compliance and FIM for Production group

## Timeline Report

The Timeline Report section provides an excellent way to report compliance and FIM events in a monthly, daily, or hourly view. Click **Timeline Report** and select the endpoint for which you want to see the report. The total number of events is shown at the top of Figure 2-78.

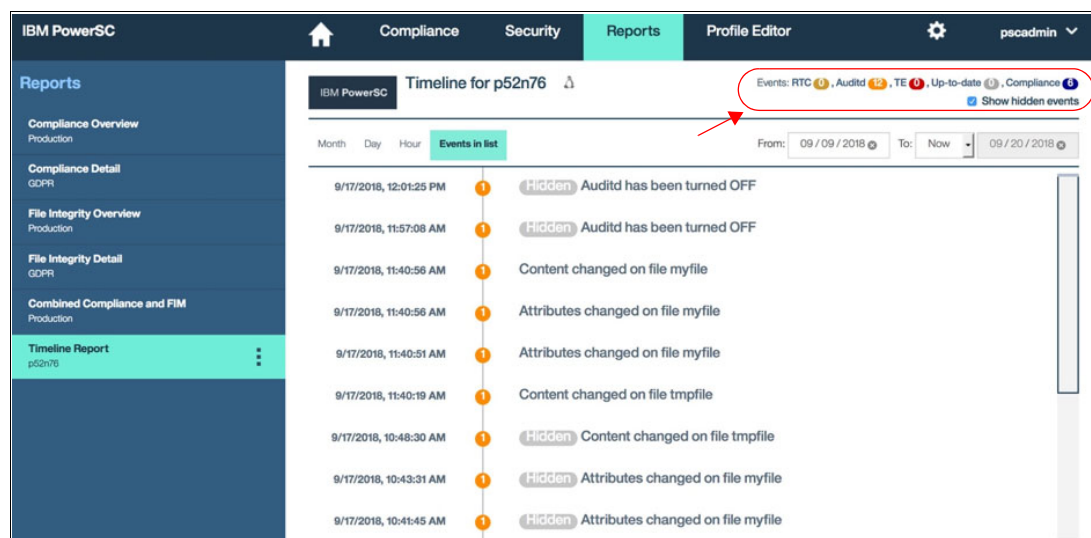


Figure 2-78 Timeline report

You also can specify a date to filter the events for the specific date range, as shown in Figure 2-79.

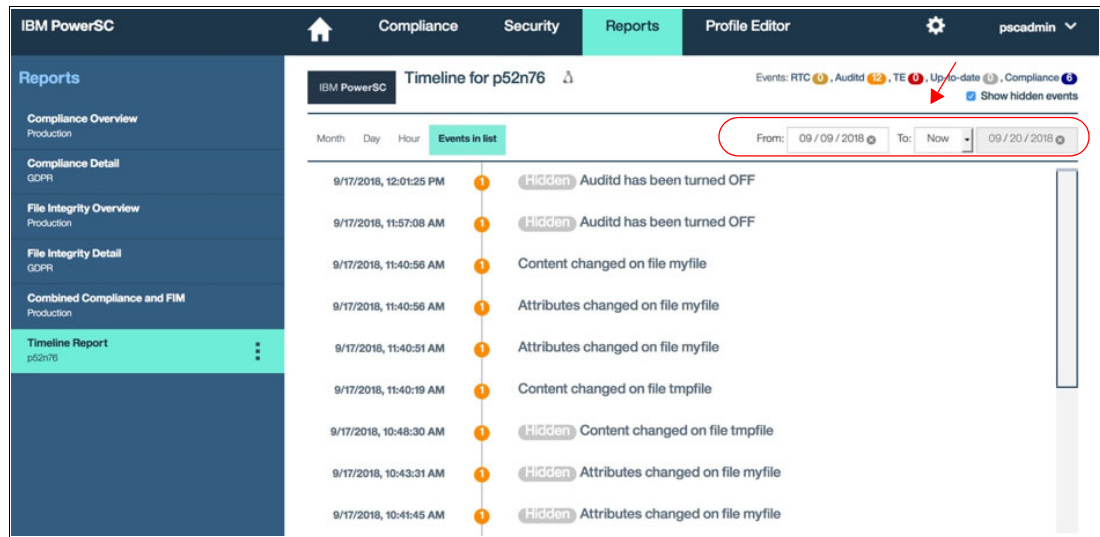


Figure 2-79 Timeline report with date range

Click **Month** to view the timeline report by month, as shown in Figure 2-80.

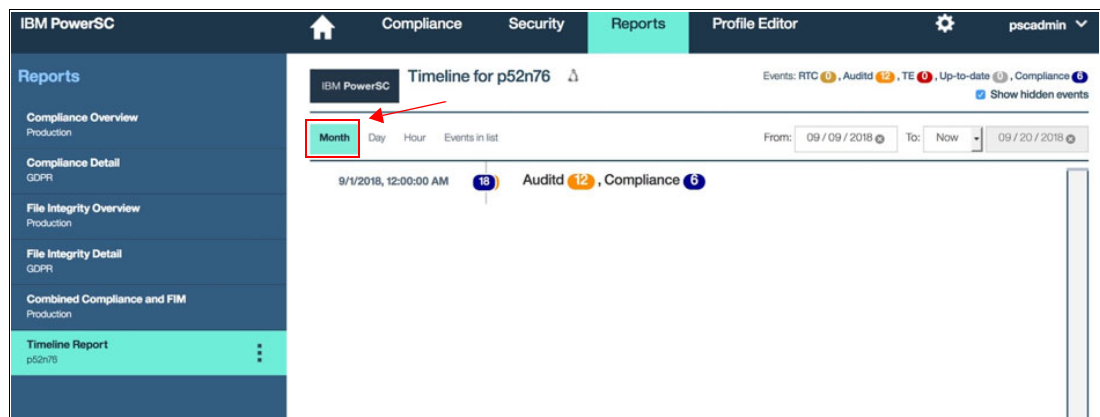


Figure 2-80 Timeline report monthly view

Click **Day** to view the timeline report by day, as shown in Figure 2-81.

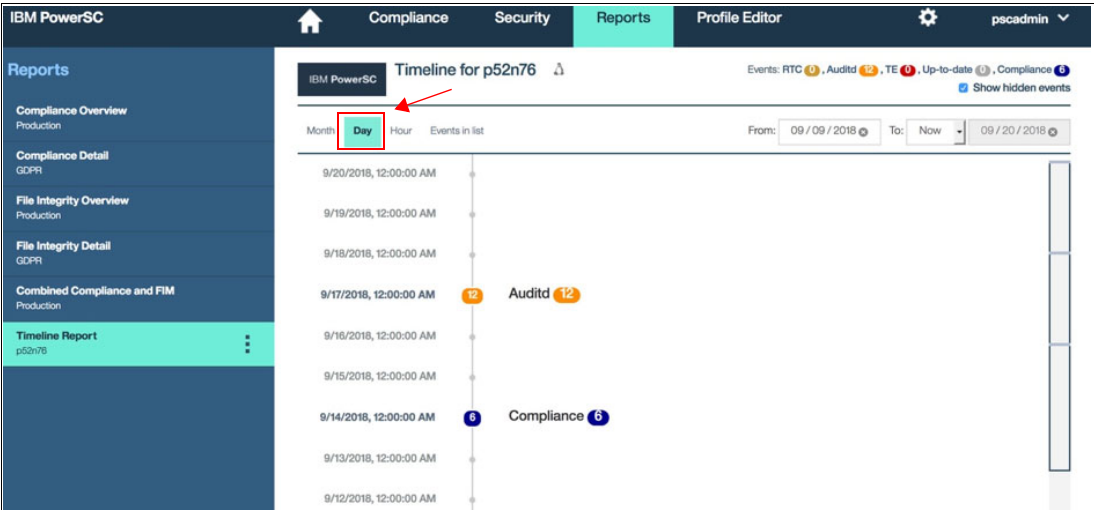


Figure 2-81 Timeline report Day view

Click **Hour** to view the timeline report by hour, as shown in Figure 2-82.

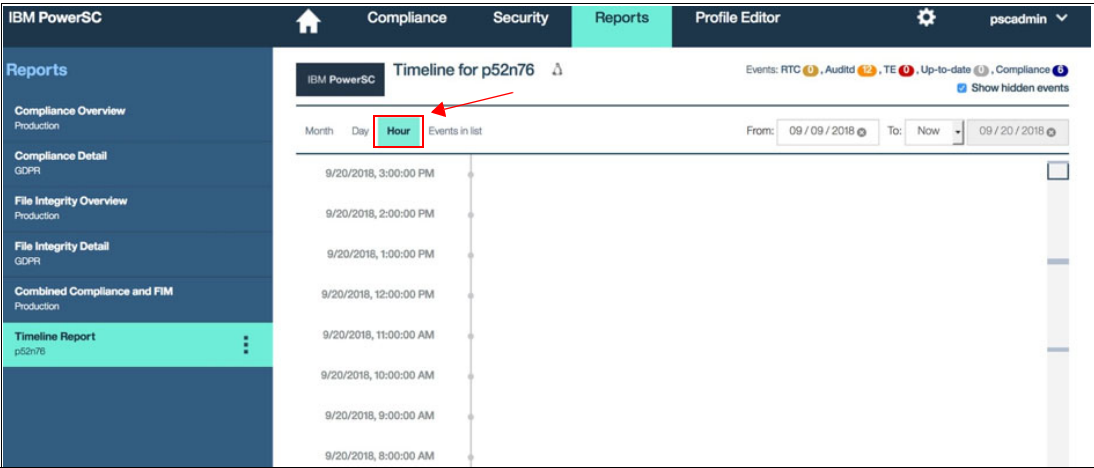


Figure 2-82 Timeline report Hourly view

Also, you can select the Event list menu to show or hide the types of vents that are displayed in the timeline, as shown in Figure 2-83.

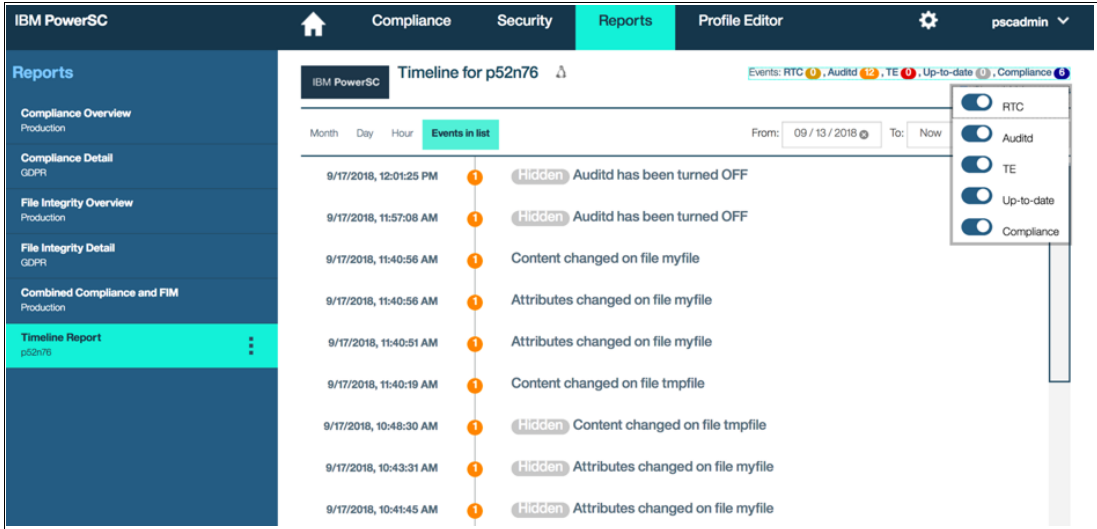


Figure 2-83 Timeline report using the event list

You also can select the **Change endpoint** option or **Configure and email immediately** option, as shown in Figure 2-84.



Figure 2-84 Timeline report by changing the endpoint

Figure 2-85 shows how to change the endpoint to generate a report.

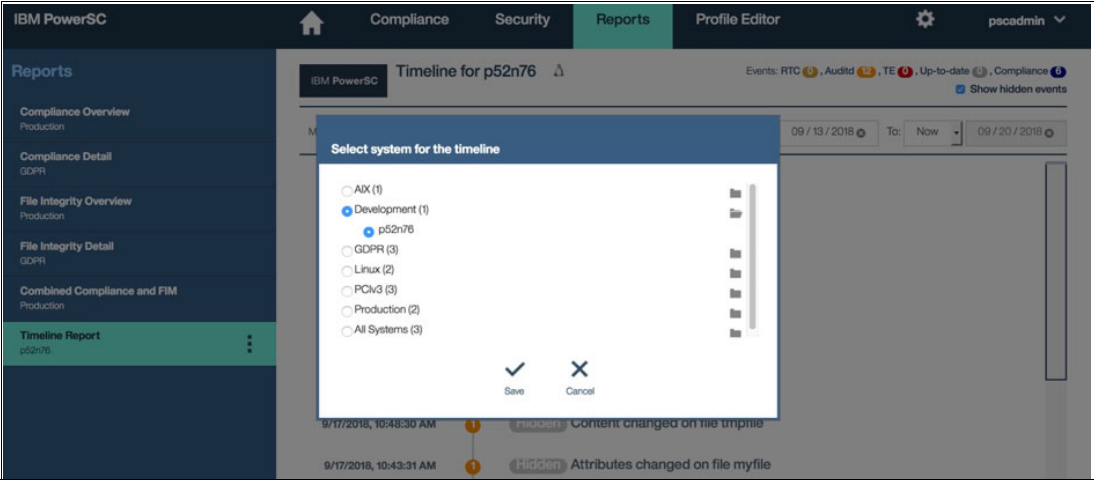


Figure 2-85 Selecting endpoint for the timeline report

Figure 2-86 shows sending the timeline report by using email.

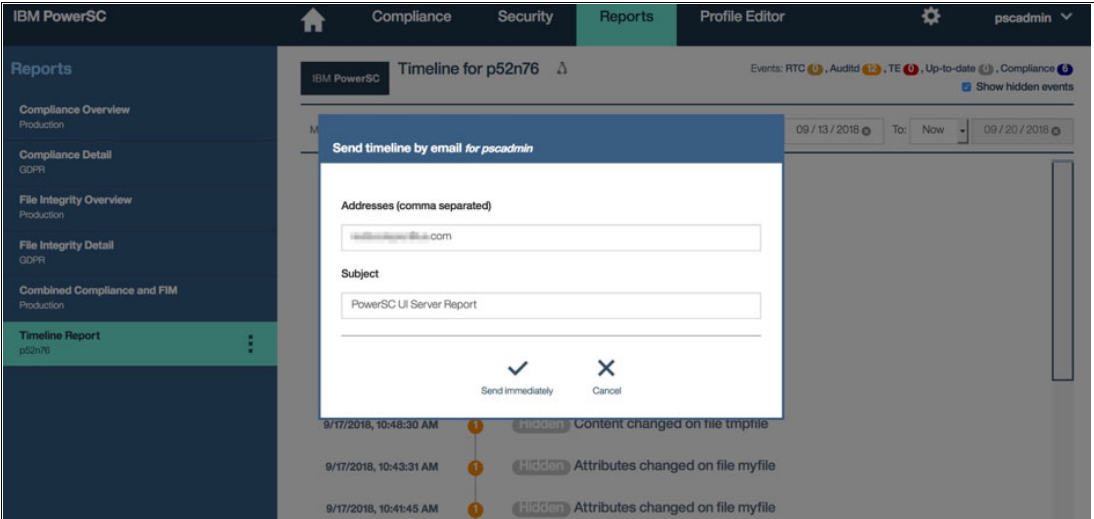


Figure 2-86 Timeline report by email

You can get more information if you select one of the events, as shown in Figure 2-87.

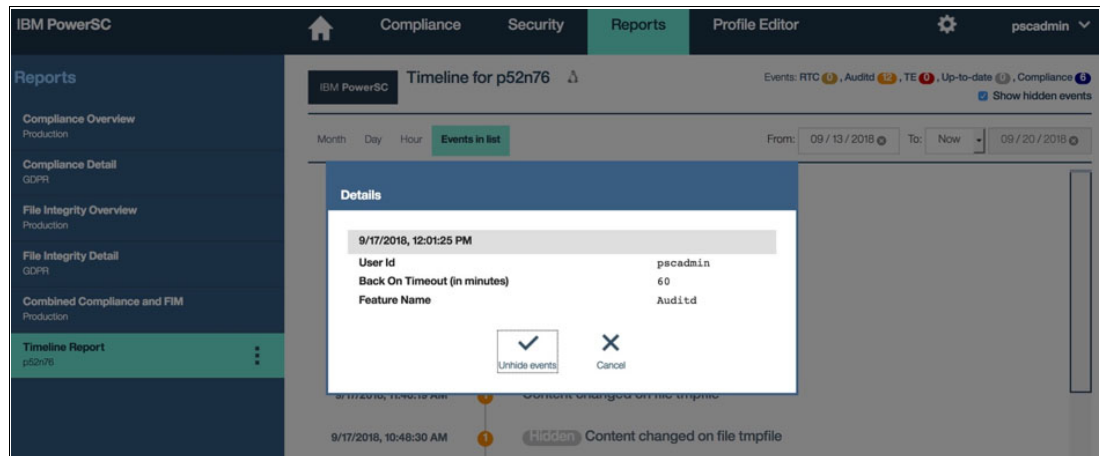


Figure 2-87 Select one of the events in the list to report more details

You can unhide an event if it was deleted from the report, as shown in Figure 2-88.

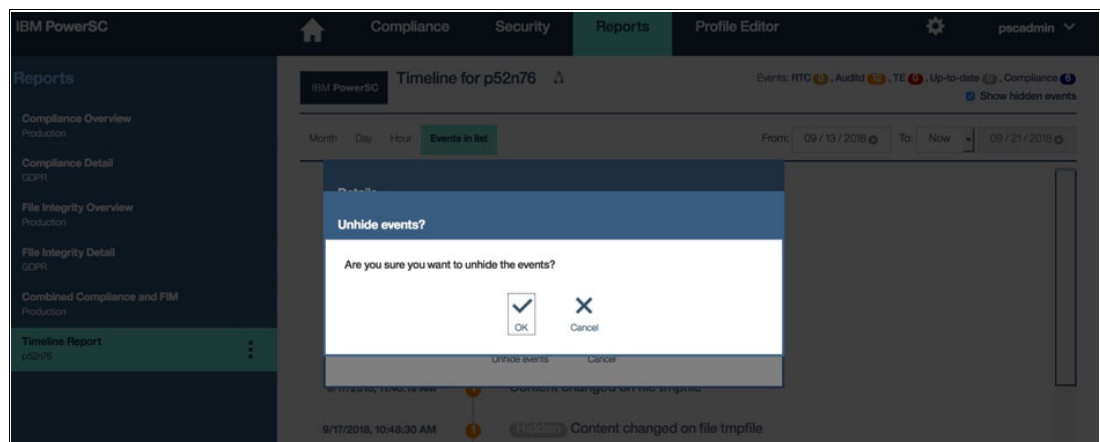


Figure 2-88 Unhide the events pane selection

In the File Integrity Detail menu option, selecting an event shows more information about the specific event, as shown in Figure 2-89 on page 73.



Figure 2-89 Timeline report selecting an event

## 2.7.5 Profile Editor tab

The Profile Editor tab provides the option to view and customize in-built compliance profiles, as shown in Figure 2-90.

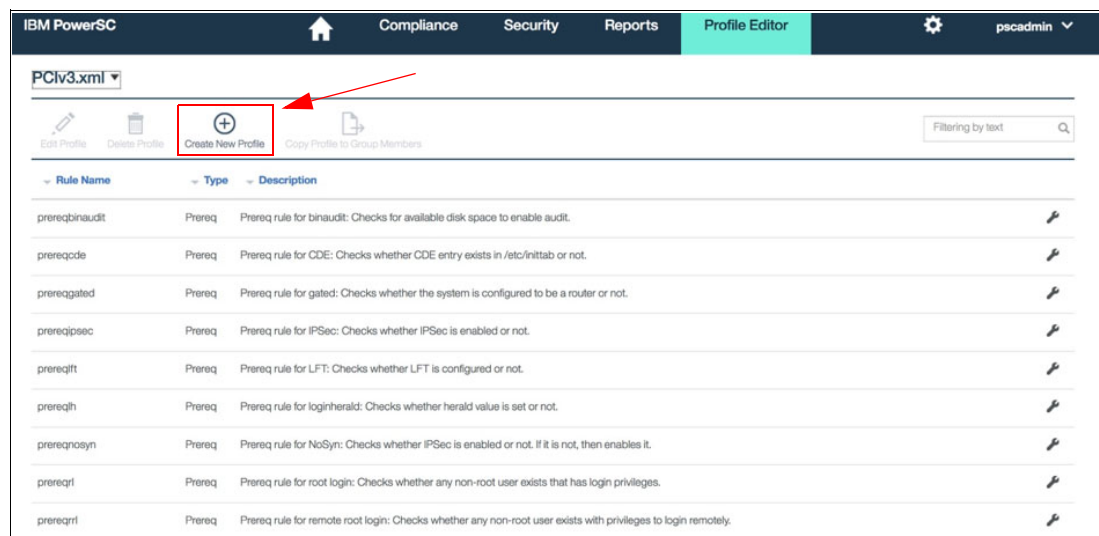


Figure 2-90 PowerSC Profile Editor pane

By using the drop-down menu, you can select any of the built-in profiles or custom profiles. You can view the security rules along with the description. The same page provides an option to create a profile by using the Create New Profile option.







## Compliance automation

Meeting compliance is a major challenge for most organizations. Many organizations still use a manual process to harden their systems, which is time-consuming and error-prone.

IBM PowerSC provides a way to automate the compliance process by providing built-in security profiles that are based on various security standards, such as PCI-DSS, HIPAA, GDPR, and DOD.

This chapter describes IBM PowerSC compliance features and contains the following sections:

- ▶ 3.1, “IBM PowerSC compliance automation overview” on page 76
- ▶ 3.2, “Installation” on page 77
- ▶ 3.3, “Profiles” on page 79
- ▶ 3.4, “Applying a profile” on page 80
- ▶ 3.5, “Checking compliance” on page 85
- ▶ 3.6, “UNDO” on page 91
- ▶ 3.7, “Custom profile” on page 92
- ▶ 3.8, “Importing custom profiles not created with IBM PowerSC” on page 103
- ▶ 3.9, “Applying the PClv3 profile to an AIX LPAR” on page 106

## 3.1 IBM PowerSC compliance automation overview

This section describes IBM PowerSC compliance automation for regulatory compliance.

### 3.1.1 Business challenge

Regulatory compliance requires setting security on systems in a uniform manner. Understanding and applying a particular standard is tedious, time consuming and error prone.

#### Solution

Security compliance automation provides pre-built profiles to support industry standards, including the following examples:

- ▶ Payment Card Industry Data Security Standard (PCI) v3
- ▶ Health Insurance Portability and Accountability Act Privacy and Security Rules (HIPAA)
- ▶ North American Electric Reliability Corporation compliance (NERC)
- ▶ Department of Defense Security Technical Implementation Guide for UNIX (DOD STIG)
- ▶ Control Objectives for Information and related Technology (COBIT)
- ▶ General Data Protection Regulation (GDPR) (Regulation (EU) 2016/679)

PSCxpert (enhanced version of AIXpert) is the underlying mechanism to apply policy settings and check for compliance.

### 3.1.2 Security and compliance automation concepts

This section describes a few concepts regarding security and compliance automation. Consider the following points:

- ▶ By using IBM PowerSC, Security and Compliance Automation concepts can be used by an automated method to configure and audit AIX systems in accordance with the US Department of Defense (DoD) Security Technical Implementation Guide (STIG), the Payment Card Industry (PCI) data security standard (DSS), the Sarbanes-Oxley act, COBIT compliance (SOX/COBIT), and the Health Insurance Portability and Accountability Act (HIPAA).
- ▶ IBM PowerSC also can help to automate the configuration and monitoring of systems that must be compliant by using an accurate and complete method of security configuration automation that is used to meet the IT compliance requirements for the most important standards, such as the DoD UNIX STIG, the PCI DSS, the Sarbanes-Oxley act, COBIT compliance (SOX/COBIT), and the Health Insurance Portability and Accountability Act (HIPAA).
- ▶ IBM PowerSC uses preconfigured compliance profiles to reduce the administrative workload of interpreting compliance documentation and implementing the standards as specific system configuration parameters. This technology also can help your company to reduce the cost of compliance configuration and auditing by automating the processes.

For the information about IBM PowerSC, see [IBM Knowledge Center](#).

## 3.2 Installation

To use the security and compliance automation feature, it is necessary to install the fileset powerscSTD.ice fileset.

This fileset must be installed on AIX and Linux systems that require the security and compliance automation feature.

### 3.2.1 Operating system prerequisites

Using the same directory or location from the installation media, check all filesets that are available for PowerSC Standard Edition, as shown in Figure 3-1.

```
# pwd
/mnt/installp/ppc
#
# ls
.toc                powerscStd.rtc      powerscStd.uiAgent
openpts.verifier    powerscStd.svm      powerscStd.uiServer
powerscStd.ice       powerscStd.tnc_commands powerscStd.vlog
powerscStd.license   powerscStd.tnc_lib
powerscStd.msg       powerscStd.tnc_pm
```

Figure 3-1 Checking the installation media for PowerSC Standard Edition

The installation can be done by using the command line, but this example uses SMIT, as shown in Figure 3-2.

```
# smitty installp

Move cursor to desired item and press Enter.

Install Software
Update Installed Software to Latest Level (Update All)
Update Installed Software to Latest Level (Live Update)
Install Software Bundle
Update Software by Fix (APAR)
Install and Update from ALL Available Software
```

Figure 3-2 SMIT panel for PowerSC Standard Edition installation

Select the powerscStd.ice fileset for installation, as shown in Figure 3-3.

INPUT device / directory for software	Move cursor to desired item and press F7. Use arrow keys to ONE OR MORE items can be selected. Press Enter AFTER making all selections.
<b>SOFTWARE to install</b>	
PREVIEW only? (install operation will NOT occur)	[TOP]
COMMIT software updates?	#-----
SAVE replaced files?	#
AUTOMATICALLY install requisite software?	#
EXTEND file systems if space needed?	# KEY:
OVERWRITE same or newer versions?	# @ = Already installed
VERIFY install and check file sizes?	#
Include corresponding LANGUAGE filesets?	#-----
DETAILED output?	
Process multiple volumes?	openpts.verifier
ACCEPT new license agreements?	+ 1.0.0.1 Open Platform Trust Services - verifier
Preview new LICENSE agreements?	> <b>powerscStd.ice</b>
INVOKE live update?	@ 1.2.0.0 IBM PowerSC Standard Profile
Requires /var/adm/ras/liveupdate/lvupdate.data.	> powerscStd.license
WPAR Management	@ 7.1.2.0 PowerSC Standard Edition
Perform Operation in Global Environment	powerscStd.rtc
Perform Operation on Detached WPARs	@ 1.2.0.0 Real-Time Compliance
Detached WPAR Names	
Remount Installation Device in WPARs	powerscStd.svm
Alternate WPAR Installation Device	+ 1.2.0.0 Secure Virtual Machine
	powerscStd.tnc_commands
	+ 1.2.0.0 Trusted Network Connect Commands
	powerscStd.tnc_lib
	+ 1.2.0.0 Trusted Network Connect Libraries
	powerscStd.tnc_pm
	+ 1.2.0.0 Trusted Network Connect for Patch Management

Figure 3-3 Selecting the software to install

Figure 3-4 shows how to accept the license agreements.

	[Entry Fields]
* INPUT device / directory for software	.
* <b>SOFTWARE to install</b>	[ <b>powerscStd.ice</b> ]
PREVIEW only? (install operation will NOT occur)	no
COMMIT software updates?	yes
SAVE replaced files?	no
AUTOMATICALLY install requisite software?	yes
EXTEND file systems if space needed?	yes
OVERWRITE same or newer versions?	no
VERIFY install and check file sizes?	no
Include corresponding LANGUAGE filesets?	yes
DETAILED output?	no
Process multiple volumes?	yes
<b>ACCEPT new license agreements?</b>	<b>yes</b>
Preview new LICENSE agreements?	no
INVOKE live update?	no
Requires /var/adm/ras/liveupdate/lvupdate.data.	
WPAR Management	
Perform Operation in Global Environment	yes
Perform Operation on Detached WPARs	no
Detached WPAR Names	[_all_wpars]
Remount Installation Device in WPARs	yes
Alternate WPAR Installation Device	[ ]

Figure 3-4 Accept the license agreements

After completing the installation, confirm it by using the `ls1pp` command, as shown in Figure 3-5.

```
# ls1pp -l powerscStd.ice
```

Fileset	Level	State	Description
-----			
Path: /usr/lib/objrepos powerscStd.ice	1.2.0.0	COMMITTED	IBM PowerSC Standard Profile
Path: /etc/objrepos powerscStd.ice	1.2.0.0	COMMITTED	IBM PowerSC Standard Profile

Figure 3-5 Confirming the installation of PowerSC Standard Edition

### 3.3 Profiles

After installing the filesets for security and compliance automation, you can find them in the `/etc/security/aixpert/custom` directory on AIX systems, and the `/etc/security/pscxpert/custom.` directory for Linux

For example, an AIX system is checked in Figure 3-6 for the installed filesets.

```
# cd /etc/security/aixpert
#
# ls
README.ICEexpress  core          ldap          undo
bin                custom        log
check_report.txt   dictionary    tmp
#
# cd custom
#
# ls
DataBase.xml      Hipaa.xml      NERCv5_to_AIXDefault.xml
DoDv2.xml         NERC.xml      PCIv3.xml
DoDv2_to_AIXDefault.xml  NERC_to_AIXDefault.xml  PCIv3_to_AIXDefault.xml
GDPRv1.xml        NERCv5.xml    SOX-COBIT.xml
```

Figure 3-6 Checking the installed filesets for an AIX system

#### 3.3.1 Payment Card Industry Data Security Standard (PCI) v3

The Payment Card Industry - Data Security Standard (PCI - DSS) categorizes IT security into 12 sections, which are called the 12 requirements and security assessment procedures. For more information, see [IBM Knowledge Center](#).

#### 3.3.2 General Data Protection Regulation

General Data Protection Regulation (GDPR) is designed to unify data privacy requirements across the European Union (EU). If your company uses any type of commercial market or processes the information of EU Data Subjects, you might need to be GDPR compliant.

GDPR requires several levels of compliance. The levels of compliance that can be interpreted as requirements for server and network configuration are addressed with the IBM PowerSC product. For more information, see [IBM Knowledge Center](#).

### 3.3.3 Test scenarios with GDPR and PCIv3

The Security and Compliance Automation component of PowerSC Standard Edition can help you deploy and monitor your AIX and Linux systems in your environments. Depending on your requirements, you can use certified, predefined policies or customized policies. To customize the policies, you change the configurable elements in the regulatory standards and directives.

For this IBM Redbooks publication, we tested two scenarios by using the GDPR and PCIv3 and applying them to AIX and Linux systems.

**Note:** Always test first. It is recommended to implement the component on a test environment. Changes that are made with the Security and Compliance Automation component affect the systems, and applications might not work as designed.

For your operating system's default settings, remember that applying security profiles can change the default settings on your operating system. Ensure that the rules do not prevent you from accessing your system after implementation. Even root can be locked out when the settings are applied.

## 3.4 Applying a profile

A profile can be applied by using the command line (which was available until version 1.1.4 of IBM PowerSC) or by using the GUI.

Each of the IBM PowerSC built-in profiles includes rules that must be applied to an endpoint to meet security requirements. You can create a custom profile when you need to apply only a subset or a different combination of these rules or customize compliance levels.

Figure 3-7 on page 81 shows the steps for applying the profile for PCIv3, which can be found in the directory `/etc/security/aixpert/custom` in the AIX environment.

To apply a security profile, use the following command:

```
pscxpert -f <Path to Profile> -p
```

```
# pscxpert -f /etc/security/aixpert/custom/PCiv3.xml -p
Processing prereqbinaudit :cached
Processing prereqcode :cached
Processing prereqgated :cached
Processing prereqipsec :cached
Processing prereqlft :cached
Processing prereqlh :cached
Processing prereqnosyn :cached
Processing prereqrl :cached
Processing prereqrll :cached
Processing prereqsed :cached
Processing prereqmontcb :cached
Processing prereqda :cached
Processing prereqTE :cached
Processing prereqssh :cached
Processing pciv3_ipsecshunhost .....:done.
Processing pciv3_ipsecshunports .....:done.
Processing pciv3_tcptr :done.
Processing pciv3_SecureLPM ...: warning.
do_action(): Warning: Prereq failed for ios.cli.rte
Processing pciv3_ipsecpermit .....:done.
Processing pciv3_maxexpired .....:done.
Processing pciv3_maxrepeats .....:done.
Processing pciv3_netstat .....:done.
Processing pciv3_disssnmpdmn .....: failed.
Processing pciv3_disssnmpmibddmn .....: failed.
Processing pciv3_disaixmibddmn .....:done.
Processing pciv3_hostmibddmn .....:done.
Processing pciv3_dislpd .....:done.
Processing pciv3_discde .....:done.
Processing pciv3_distimedmn .....:done.
Processing pciv3_disrwhoddmn .....:done.
```

Figure 3-7 Applying a profile using the CLI

Figure 3-8 shows the final execution results after the profile is applied.

```
Processing pciv3_removequest .....:done.
Processing pciv3_pciaudit :done.
Processing pciv3_entpdmn .....:done.
Processing pciv3_rootrlogin .....:done.
Processing pciv3_chetcftpusers .....:done.
Processing pciv3_sedconfig .....:done.
Processing pciv3_rootpwdintchk .....:done.
Processedrules=105      Passedrules=98  PrereqFailedrules=5      Failedrules=2  L
evel=PCiv3
      Input file=/etc/security/aixpert/custom/PCiv3.xml
```

Figure 3-8 Completing the execution of the profile

Applying a profile can take few minutes to complete and you can check all rules that are being processed. In the end, you can see how many rules were processed (in our case 105), how many rules passed, how many rules failed, and how many rules include prerequisite failures. Also, these rules can be checked on /etc/security/aixpert//check\_report.txt.

If you have prerequisites rules or failure rules, we recommend reviewing them and checking whether that prerequisite applies to your environment. The system can have missing installation prerequisites or other issues that require attention from the administrator.

After determining the underlying command of the failed rule, examine the system to understand the configuration command that is failing. One alternative can be to create your custom PCI profile version. For more information about creating custom profiles, see 3.7, “Custom profile” on page 92. Then, a custom security profile must be created.

By using the command **pscxpert -t**, you can confirm that the latest profile is applied, as shown in Figure 3-9.

```
# pscxpert -t
Applied Profiles:PCIv3
```

Figure 3-9 Latest profile applied

### 3.4.1 Using the GUI

To apply profiles from the GUI, log in to the GUI by using a browser with a user that has administrator privileges (in our example, as pscadmin user). Then, on the left side of the Compliance tab, apply a compliance level or profile to one or more endpoints in a selected group.

The levels and profiles that can be applied to all the endpoints are listed. Profiles that cannot be applied to all the selected endpoints are displayed with a gray color. The profile in gray mean this profile is not copied on the endpoint. If required, you can copy the profile to the endpoint from the Profile Editor Tab.

For our example, we selected the GDPR group and the Linux\_GDPRv1 profile, as shown in Figure 3-10.

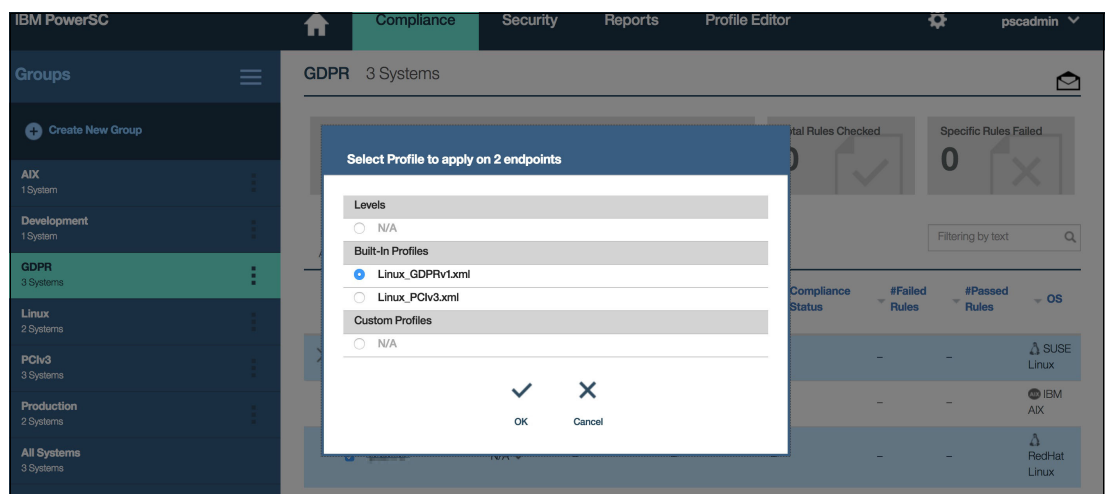


Figure 3-10 Selecting the profile to apply to two endpoints



Also, select the endpoints with Red Hat or SUSE and click **OK**, as shown in Figure 3-11.

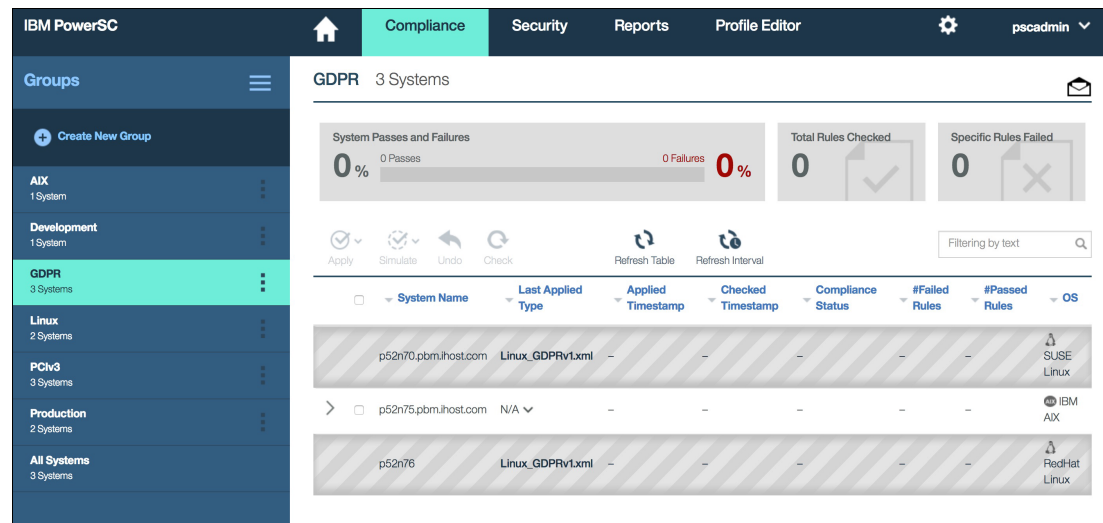


Figure 3-11 Selecting the endpoints

You can check the applying process has started and after some minutes, you can see the results, as shown in Figure 3-12.

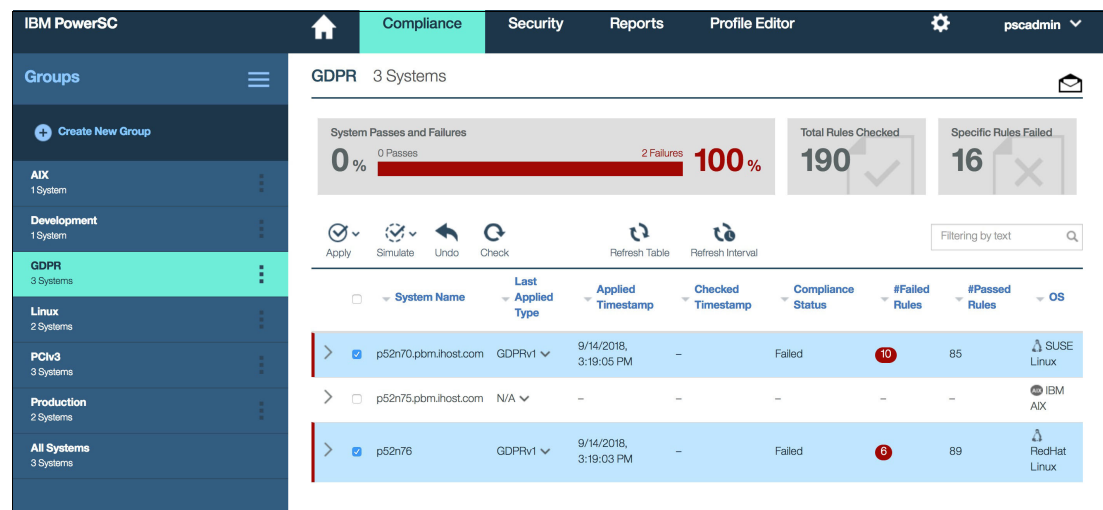


Figure 3-12 Process started and its results are shown

In this scenario, if one or more rules cannot be applied after applying the profile, the rules are considered failed. More information is available about how many rules were checked for both servers, and the specific rules that failed.

If one or more rules fail, the endpoint is marked with a red bar. You also can check the reason for the rule failure (see Figure 3-13).

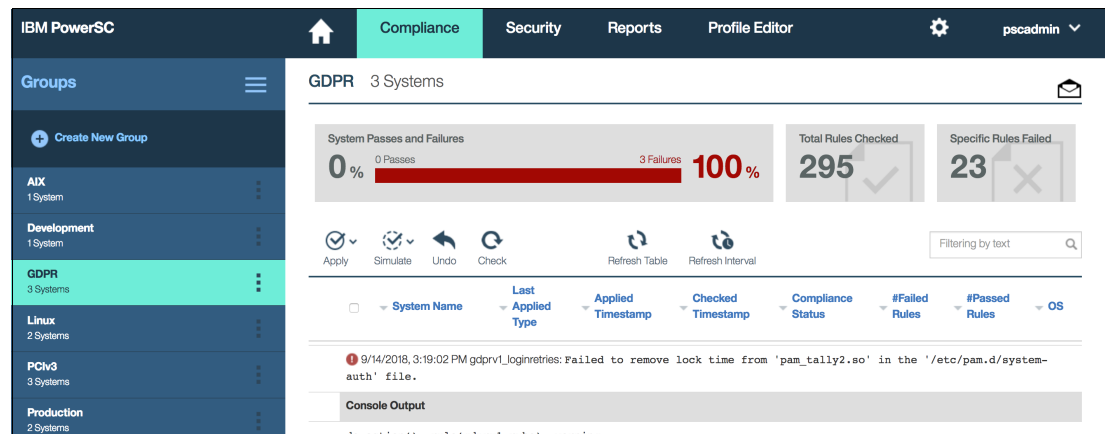


Figure 3-13 Checking the status of the applied profiles

Failed rules can be associated with many causes; for example pre-requisites failures, or that some rules do not apply to your environment. Therefore, we recommend verifying and analyzing the results before applying any profile to perform a simulate process. This process can be better understood at the Simulate Option session.

Also, you can adjust the rules that are applied by creating a custom profile or by editing a custom profile (see Figure 3-14).

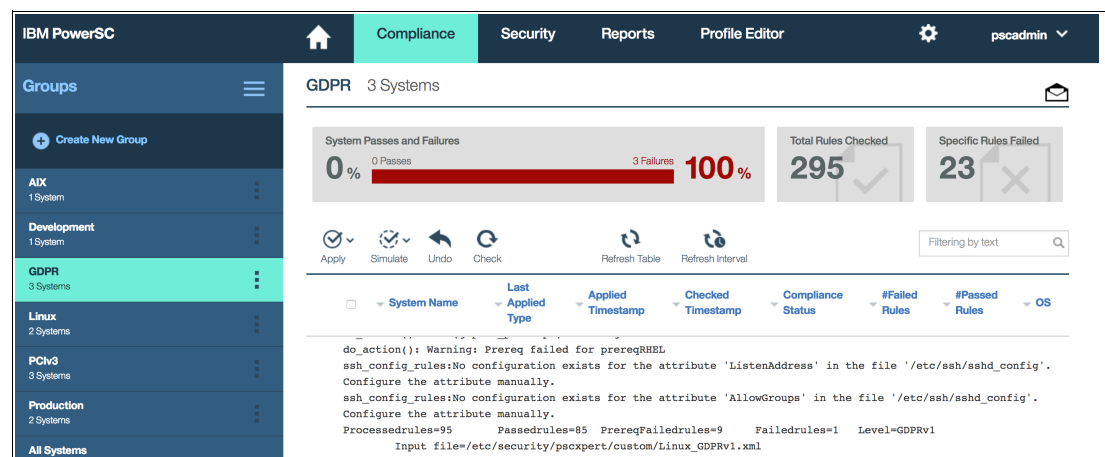


Figure 3-14 Checking for messages after applying profiles

It is important to understand what each rules does and what changes it is making to the environment. It is possible that some rules cannot be applied because of the characteristics of the server.

In most environments and situations, it is recommend that the administrators review and edit compliance files to remove problem rules. After compatibility checks are completed, the compliance rule files can be considered stable and used to be deployed onto production servers.

## 3.5 Checking compliance

In this section, we describe how to set up and check for compliance.

### 3.5.1 Checking against applied profile

You can use the cron job or a script to perform and schedule regular checks by way of the command line by using the **pscexpert -c** command, as shown in Figure 3-15.

```
# pscexpert -c
Processedrules=98      Passedrules=91  Failedrules=7   Level=PCIv3
      Input file=/etc/security/aixpert/core/appliedaixpert.xml
```

Figure 3-15 Checking the status of the applied profiles

One file called `check_report.txt` was created at `/etc/security/aixpert`, as shown in Figure 3-16.

```
# cat /etc/security/aixpert/check_report.txt

**** p52n75.pbm.ihost.com : Sep 14 16:25:13 ****

cominetdconf: Service ftp using protocol tcp should be disabled, however it is
now enabled.
sshPCIconfig: No configuration exists for the attribute ListenAddress in the
file /etc/ssh/sshd_config. Configure the attribute manually.
sshPCIconfig: No configuration exists for the attribute AllowGroups in the file
/etc/ssh/sshd_config. Configure the attribute manually.
EnableRbac: RBAC users are not properly created. To create these RBAC users run
the script /etc/security/pscexpert/bin/RbacEnablement.
```

Figure 3-16 Output file with a status report

The checking process can be done to confirm that the last applied compliance level and profile were applied.

From the GUI main page, select the **Compliance** tab → **At Groups** on left side and then, elect the group. In our example, GDPR with Linux endpoints can be checked for compliance levels, and the GDPR profile applied, as shown in Figure 3-17.

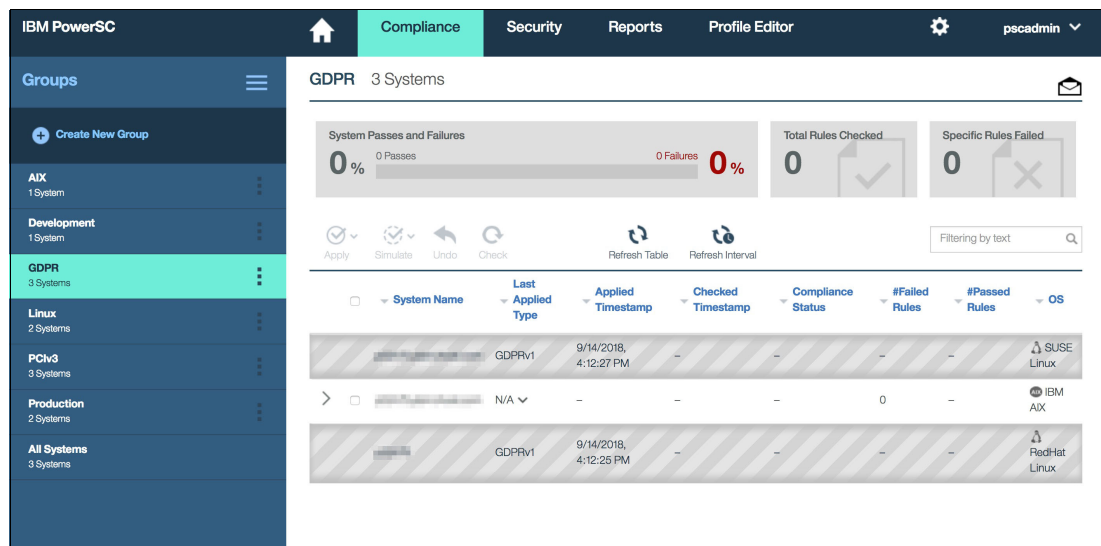


Figure 3-17 Profiles applied to GDPR

Figure 3-18 shows the PowerSC compliance for GDPR after the profiles are applied.

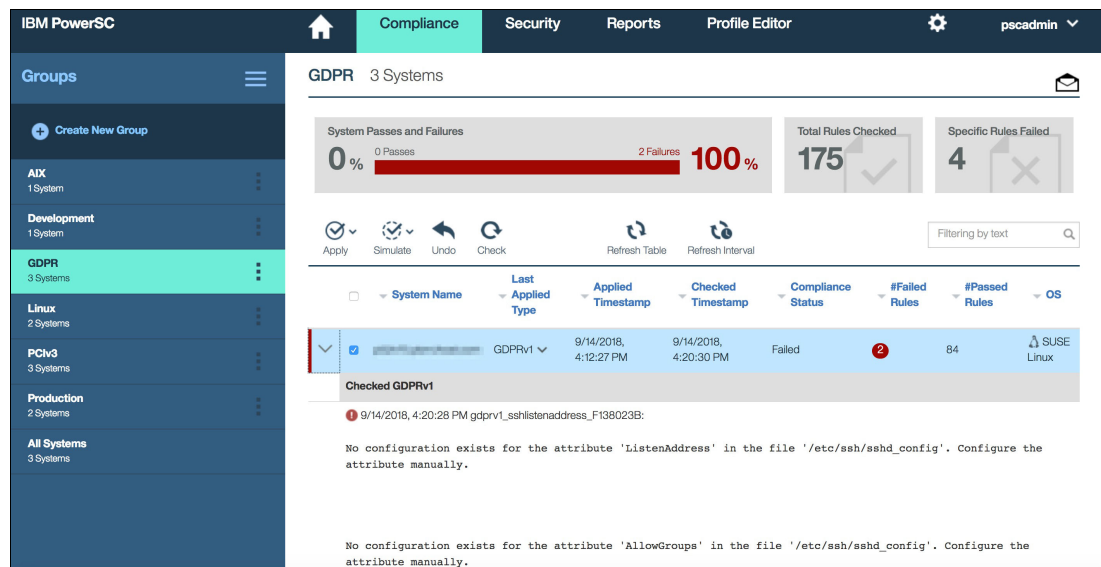


Figure 3-18 Checking compliance after applying profiles

During the check process, the endpoint is checked to see whether the rules that are in the compliance level or profile can be applied. The endpoints are not updated. If any rules cannot be applied, it is considered that they fail when they are applied. If one or more rules fail, the endpoint is shown with a red bar and the text “Failed” is displayed in the #Failed Rules column. In our example, we can see that some failed rules exist, which must be analyzed.

From the #Failed Rules list for each endpoint that is marked with red, you can view the message that indicates why the rule failed. You can adjust the rules that are applied by creating a custom profile.

If you want to check a compliance level or profile that was not applied to one or more endpoints in a selected group, you can repeat the steps. When you open the Last Checked Type drop list, select one of the following options:

- ▶ All available levels: Displays a list of all the available levels that you can check against an endpoint.
- ▶ All available profiles: Displays a list of all the available profiles that you can check against an endpoint.

Then, select the level or profile that want to check against an endpoint.

### 3.5.2 Simulate option

The simulate option can be a best practice to use during the implementation process. It is possible to simulate the action of applying any profile against a selected endpoint. This process helps you simulate and analyze the impact of applying a profile to your environment and the impact to the policies that are defined.

You can perform the simulate process by using the command line as well. At the prompt of your endpoint, run the following command:

```
#pscxpert -c -P /etc/security/aixpert/custom/GDPRv1.xml -p
```

In our example, we use the GDPRv1.xml to simulate against the endpoint, as shown in Figure 3-19 and Figure 3-20 on page 88.

```
# psctxpert -c -P /etc/security/aixpert/custom/GDPRv1.xml -p
Processing gdprv1_ipsecshunhost : failed.
Processing gdprv1_ipsecshunports :done.
Processing gdprv1_toptr : failed.
Processing gdprv1_SecureLPM : failed.
Processing gdprv1_ipsecpermit :done.
Processing gdprv1_maxexpired : failed.
Processing gdprv1_maxrepeats : failed.
Processing gdprv1_netstat :done.
Processing gdprv1_dissnmpdmn : failed.
Processing gdprv1_dissnmpmibddmn : failed.
Processing gdprv1_disaixmibddmn :done.
Processing gdprv1_hostmibddmn :done.
Processing gdprv1_dislpd :done.
Processing gdprv1_discde :done.
Processing gdprv1_distimedmn :done.
Processing gdprv1_disrwhoddmn :done.
Processing gdprv1_disdpid2dmn :done.
Processing gdprv1_disdhcpserver :done.
Processing gdprv1_disdhcpagent :done.
Processing gdprv1_shell : failed.
Processing gdprv1_rlogin : failed.
Processing gdprv1_rexecd : failed.
Processing gdprv1_comsat :done.
Processing gdprv1_fingerd :done.
Processing gdprv1_systat :done.
```

Figure 3-19 Simulating the process

```

Processing pciv3_pwdalgchk_2DCB19AF :done.
Processing pciv3_minlen_2DCB19AF :done.
Processing pciv3_minalpha_2DCB19AF :done.
Processing pciv3_minother_2DCB19AF :done.
Processing pciv3_maxage_2DCB19AF :done.
Processing pciv3_histexpire_2DCB19AF :done.
Processing pciv3_histsize_2DCB19AF :done.
Processing pciv3_pciaudit_2DCB19AF :done.
Processing pciv3_entpdmn_2DCB19AF :done.
Processing pciv3_rootrlogin_2DCB19AF :done.
Processing pciv3_chetcftpusers_2DCB19AF :done.
Processing pciv3_sedconfig_2DCB19AF :done.
Processing pciv3_rootpwdintchk_2DCB19AF :done.

Processing gdprv1_lockacc_rlogin : failed.
Processedrules=107      Passedrules=43  Failedrules=64  Level=GDPRv1
      Input file=/etc/security/aixpert/custom/GDPRv1.xml

```

Figure 3-20 Simulating the process continues

In our example, if this profile is applied to this endpoint, you can see that only 43 rules passed out of 107, as shown in Figure 3-21.

```

# cat check_report.txt

***** p52n75.pbm.ihost.com : Sep 14 16:55:23 *****

ipsecshunhosths: Port 11 is not being filtered for IPsec V4.
ipsecshunports: TCP Port 11 is not being filtered for IPsec V4.
ipsecshunports: TCP Port 19 is not being filtered for IPsec V4.
ipsecshunports: TCP Port 43 is not being filtered for IPsec V4.
ipsecshunports: TCP Port 63 is not being filtered for IPsec V4.
ipsecshunports: TCP Port 67 is not being filtered for IPsec V4.

```

Figure 3-21 Checking the report after process has run

At the same directory, you can validate the check\_report.txt file and see which rules failed and the reason for that failure, as shown in Figure 3-22.

```

retcpip: The status for daemon xntpd should be enabled in /etc/rc.tcpip file,
however it is disabled.
chuserstanza: User attribute rlogin in stanza root should have value false but
it is .
chetcftpusers: The /etc/ftpusers file does not have root entry.
sedconfig: Stack Execution Disable feature is not enabled for setidfiles.
chuserstanza: User attribute dictionlist in stanza root should have value /etc
/security/aixpert/dictionary/English but it is .
lockacc_rlogin: The user accounts daemon bin sys adm uuap lpd lp nobody invsc
out uuap are not locked.

Processedrules=107      Passedrules=43  Failedrules=64  Level=GDPRv1
      Input file=/etc/security/aixpert/custom/GDPRv1.xml

```

Figure 3-22 Checking the failing rules

Also, you can use the following command to generate a report in a csv format file:

```
# pscxpert -c -r -P /etc/security/aixpert/custom/GDPRv1.xml -p
```



or

```
# pscxpert -c -R -P /etc/security/aixpert/custom/GDPRv1.xml -p
```

The use of this command creates the `/etc/security/aixpert/check_report.csv` file, as shown in Figure 3-23.

Admin:root					
Report date and Time:Sep 14 17:09:24					
Report Version 1.0					
HostName	IP Address	Description	Command Arguments	Result	Reason for failure
p52n75.pbm.ihost.com	129.40.40.75	Shun host for 3 minutes: Shuns the hosts that try to access unused ports for 3 minutes.	/etc/security/pscxpert/bin/ipsecshunhost osthis.gdprv1_ipsecshunhost	FAIL	Port 11 is not being filtered for IPsec V4.
p52n75.pbm.ihost.com	129.40.40.75	Guard host against port scans: Shuns vulnerable ports for 3 minutes to guard the host against port scans.	/etc/security/pscxpert/bin/ipsecshunports.gdprv1_ipsecshunport	PASS	
p52n75.pbm.ihost.com	129.40.40.75	TCP Traffic Regulation High: Enforces denial-of-service mitigation on popular ports.	/etc/security/pscxpert/bin/tcptra_aixpert.gdprv1	FAIL	TCP TR has not been activated.
p52n75.pbm.ihost.com	129.40.40.75	Enable automated IPSEC Tunnel creation between VIO servers during live partition migration.	/etc/security/pscxpert/bin/cfgsecmig on	FAIL	This command can be run only on VIOS systems.
p52n75.pbm.ihost.com	129.40.40.75	Allow the packets from HMC restricting inbound and outbound traffic to that which is necessary for the cardholder data environment.	/etc/security/pscxpert/bin/ipsecpermihostorport.PermitHost_IPSEC	PASS	
p52n75.pbm.ihost.com	129.40.40.75	Always change vendor-supplied defaults before installing a system on the network. Specifies the maximum time (in weeks)	/etc/security/pscxpert/bin/chusrattr_maxexpired=8 ALL.gdprv1_maxexpired	FAIL	The attribute maxexpired for user root should

Figure 3-23 `/etc/security/aixpert/check_report.csv` file format

## Using the GUI

Similar to the check process, go to the Compliance tab, select the Group, and select the endpoint. Then, go to the Simulate tab, select the profile (in this case, PCIV3) and click **OK** (see Figure 3-24).

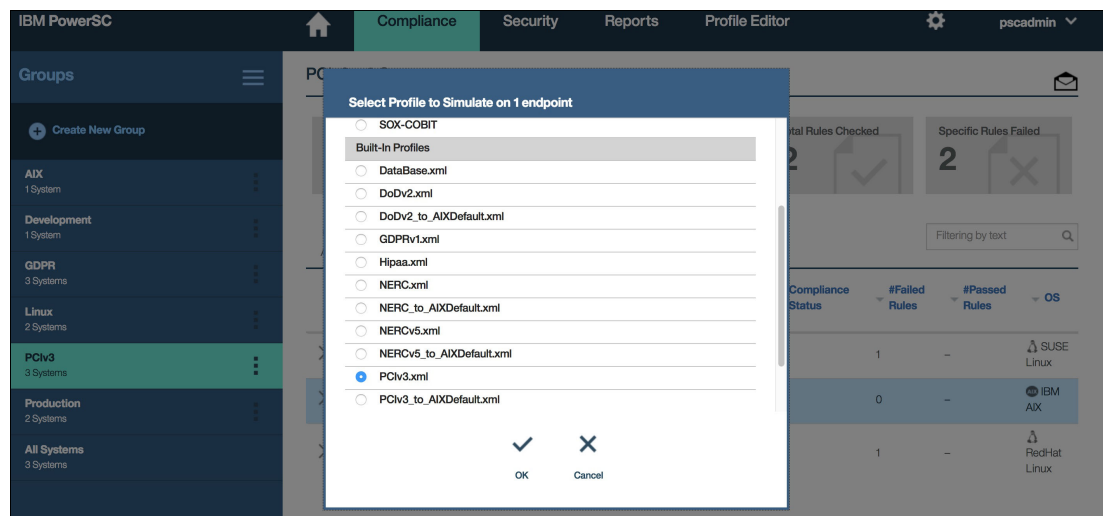


Figure 3-24 Selecting the profile to simulate

The process starts, as shown in Figure 3-25.

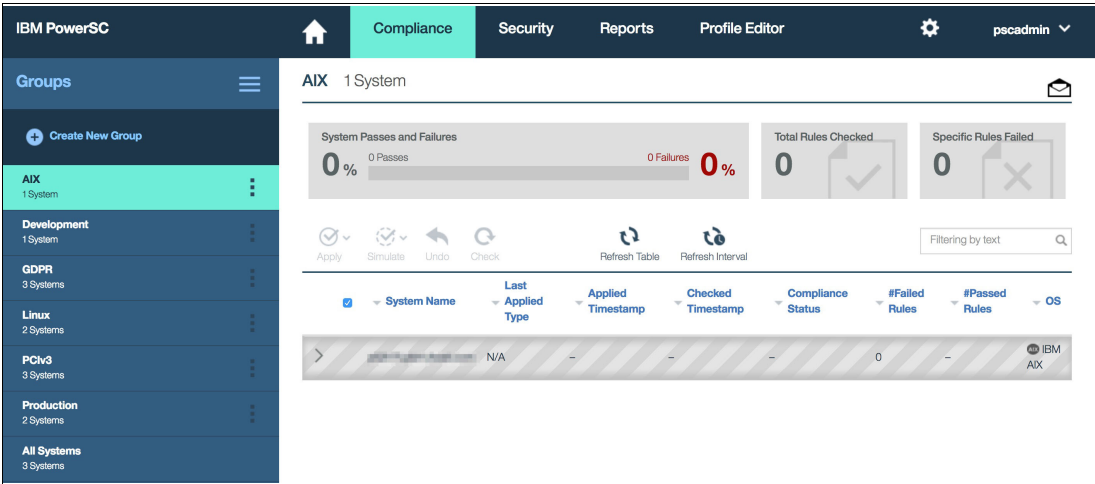


Figure 3-25 Simulation process starts

Figure 3-26 shows the results of the simulation.

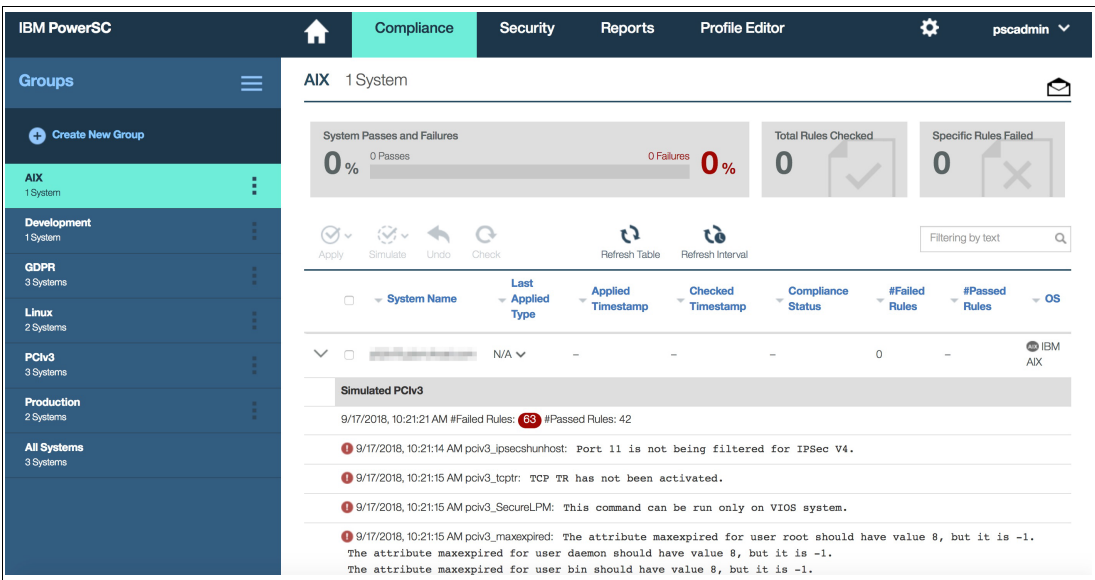


Figure 3-26 Simulation results

By using the command line process, you can review the results to analyze which rules failed and if any adjustment is necessary.

**Note:** Always test it first. It is highly advised to implement the component on a test environment. Changes that are made with the Security and Compliance Automation component affect the systems, and applications might not work as designed.



## 3.6 UNDO

In some cases, it is necessary to undo any compliance level profile applied. This process is done by selecting one or more endpoints through the GUI or command line.

When the command line is used (see Figure 3-27), you run the **pscxpert -u -p** command.

```
# psccxpert -u -p
Processing pciv3_ipsecshunhost_2DCB19AF :done.
Processing pciv3_ipsecshunports_2DCB19AF :done.
Processing pciv3_tcptr_2DCB19AF :done.
Processing pciv3_ipsecpermit_2DCB19AF :done.
Processing pciv3_maxexpired_2DCB19AF :done.
Processing pciv3_maxrepeats_2DCB19AF :done.
Processing pciv3_netstat_2DCB19AF :done.
Processing pciv3_shell_2DCB19AF :done.
Processing pciv3_rlogin_2DCB19AF :done.
Processing pciv3_rexecd_2DCB19AF :done.
```

Figure 3-27 Running the undo profile command

The UNDO process works recursively, and the UNDO rules are built dynamically.

**Tip:** The **psccxpert UNDO** command must run twice to return AIX to the default settings if you are using more than one profile.

### Using the GUI

If you must return to the previous state, use the UNDO option, which performs a back operation to the last profile that was applied.

At the Compliance tab, click **Groups** and select the endpoint that you want to undo. Click **UNDO**, as shown in Figure 3-28.

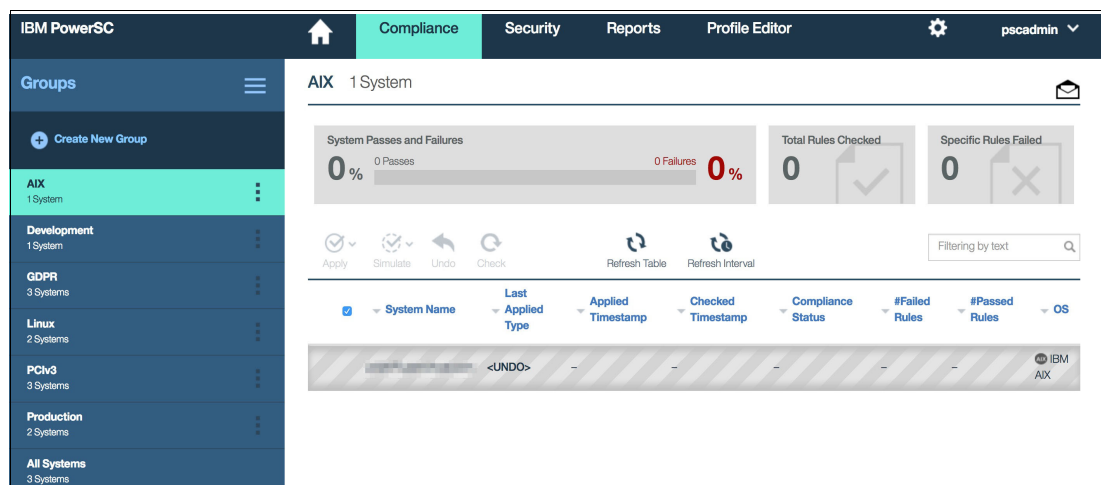


Figure 3-28 Process returning to the previous applied state

The UNDO process completion panel is shown in Figure 3-29.

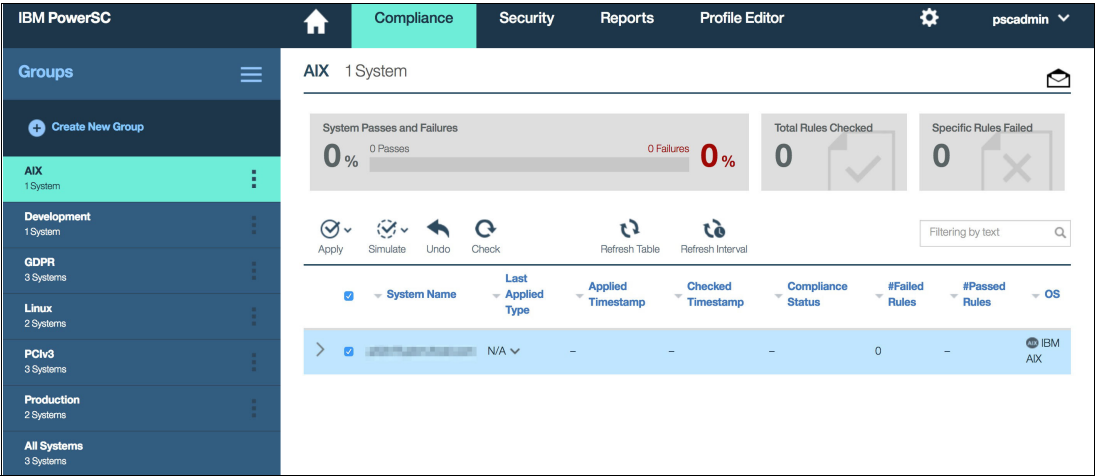


Figure 3-29 Undo process completion

After the process finishes, you see that no information about the compliance level is shown on the endpoint.

### 3.7 Custom profile

In this section, we apply the PCiv3 profiles. Because some rules do not necessarily apply to this environment, this section shows how to create a custom profile that is based on the PCiv3 in the Profile Editor tab.

Go to the Profile Editor and select the PCiv3 profile, as shown in Figure 3-30.

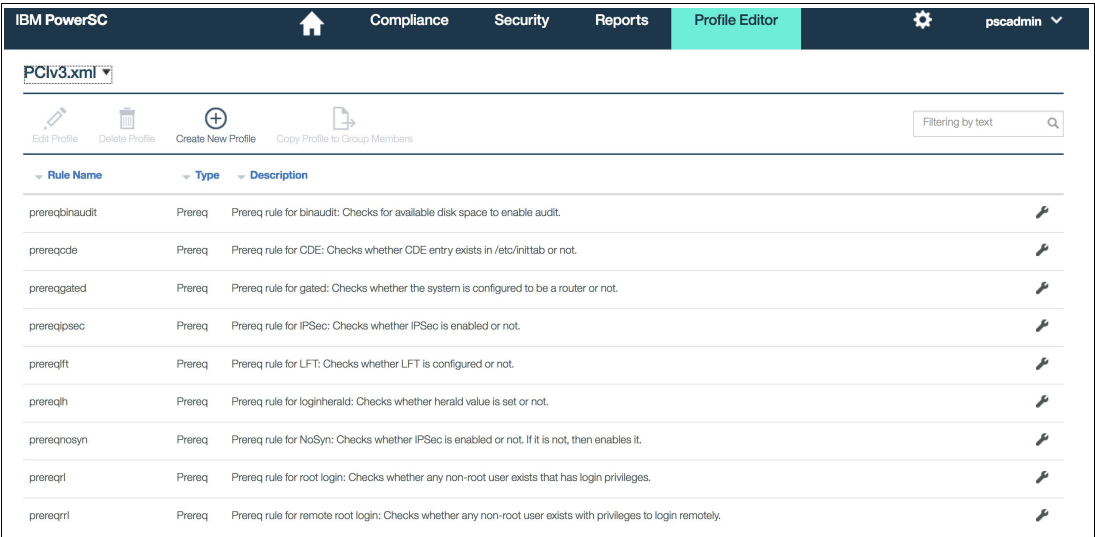


Figure 3-30 Selecting the profile

Click **Create a New Profile**, enter the new profile name (in our example, PCiv3\_Custom\_RedBook) and the type (PCiv3\_Custom), as shown in Figure 3-31. Then, click **Confirm** to continue.

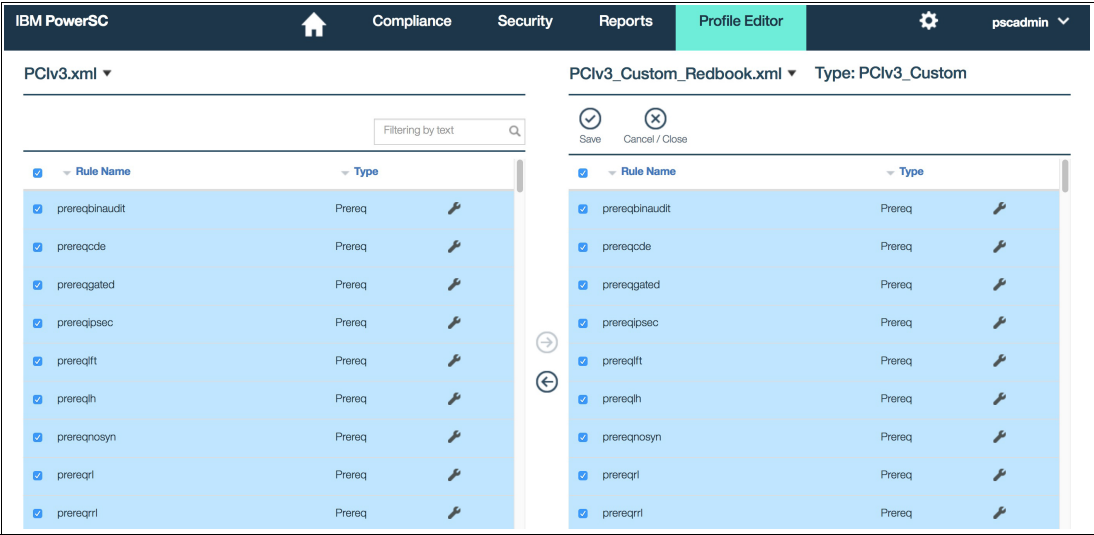


Figure 3-31 Customizing the rules

You can then select the rules individually and use the arrows to select your custom profile or select all. Then, you can edit your policies and save them, as shown in Figure 3-31 and Figure 3-32.

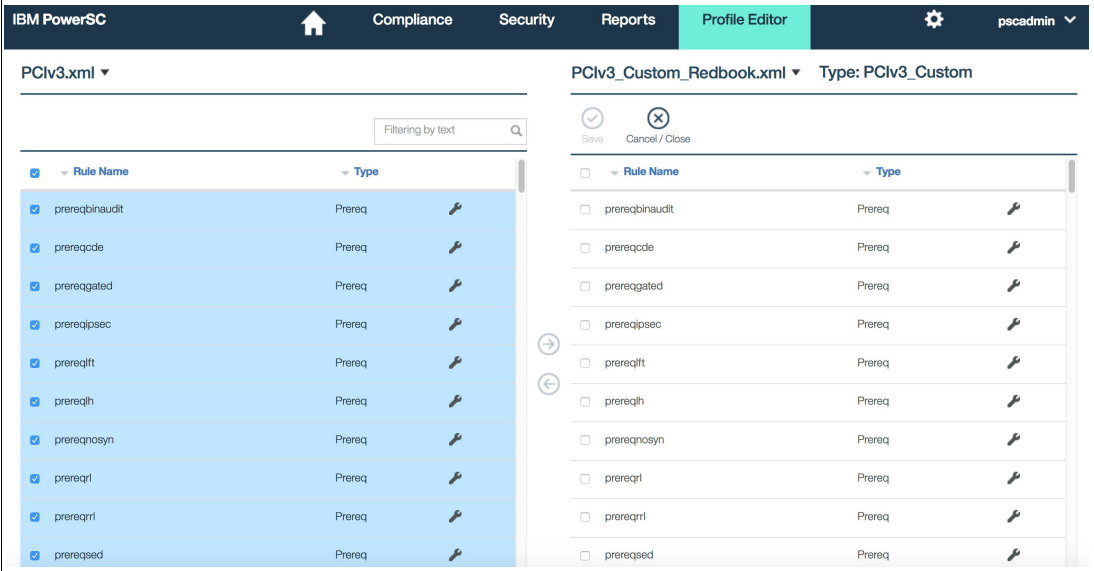


Figure 3-32 Adding rules to the new profile

After saving the new profiles, you can go through the Custom Profile again and select the rules by clicking the arrow to remove the rules that do not apply to the environment or policy, as shown in Figure 3-33.

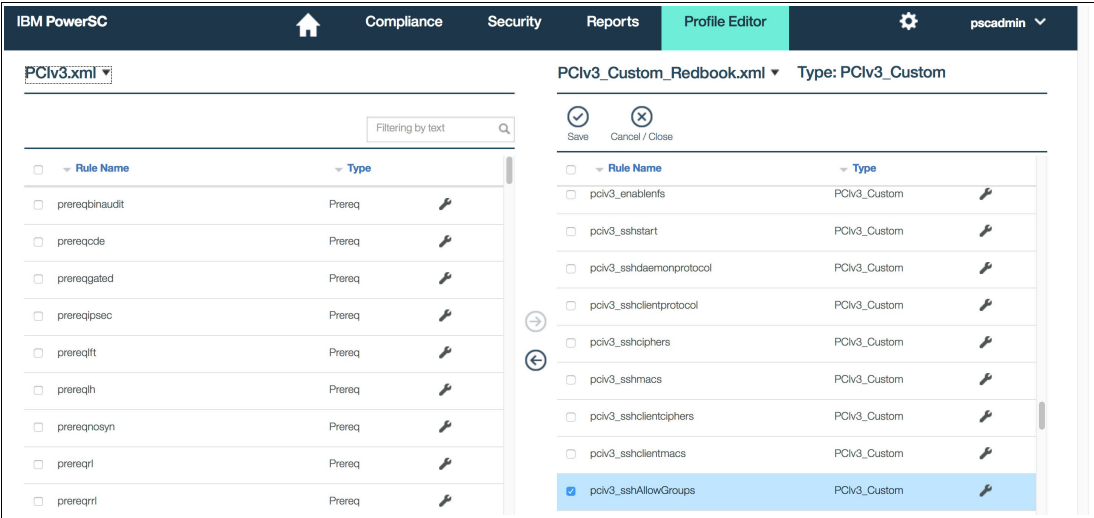


Figure 3-33 Checking and customizing the new profile

### Editing rule arguments for custom profiles

Click the wrench icon that is associated with the custom profile for which you want to view or edit the rules. The Edit rule arguments window opens.

You can edit the following fields for the rule:

- ▶ **Name**  
Specifies the name of the rule.
- ▶ **Description**  
Specifies a description of the rule.
- ▶ **Script**  
Specifies the name of the script that applies the rule.
- ▶ **Arguments**  
Specifies the arguments that are included in the rule. Each argument is separated by a space.

**Warning:** You should change rule arguments only if you are familiar with the script. Specifying incorrect values prevents the rule from working properly. Carefully examine the script before making any changes.

If a rule is not applicable to the specific system environment, most compliant organizations permit documented exceptions. Also, you can change the value of the content for specific rules as in this scenario changes the value for PCiv3\_maxage.

Complete the following steps:

1. Select the custom profile PCiv3\_Custom.
2. Click **Edit**.

3. Select the rule that you want to change, as shown in Figure 3-34 and Figure 3-35.

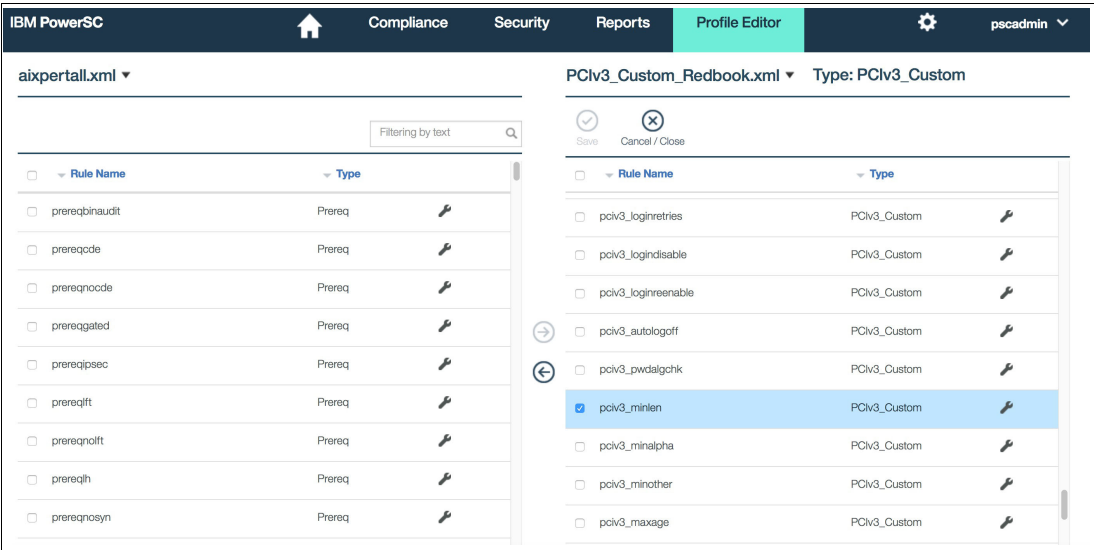


Figure 3-34 Changing the rules

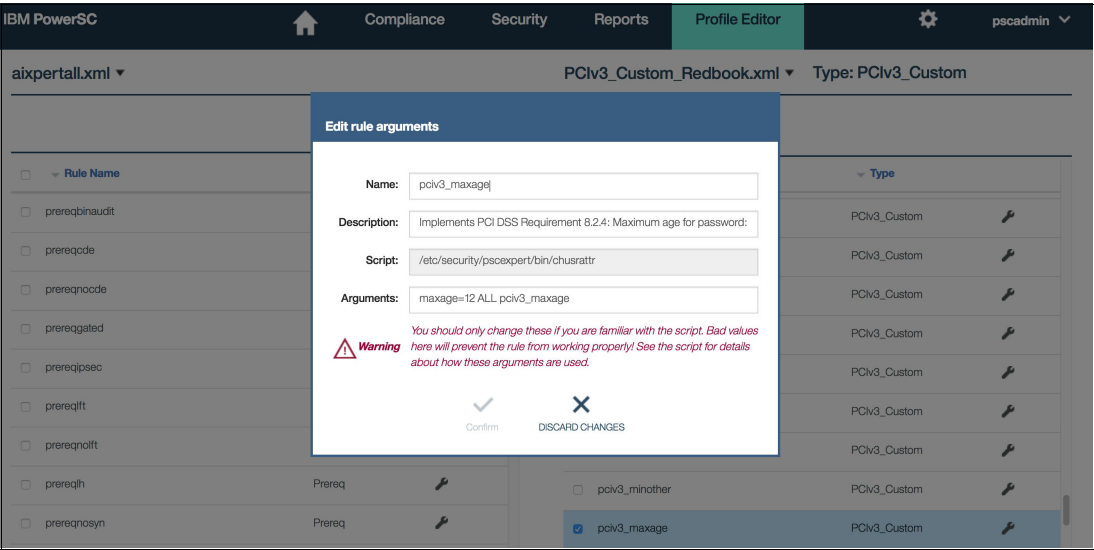


Figure 3-35 Editing the rule arguments

The value of maxage is changed from 12 to 10.

4. Click **Confirm**.

5. Save the profile, as shown in Figure 3-36.

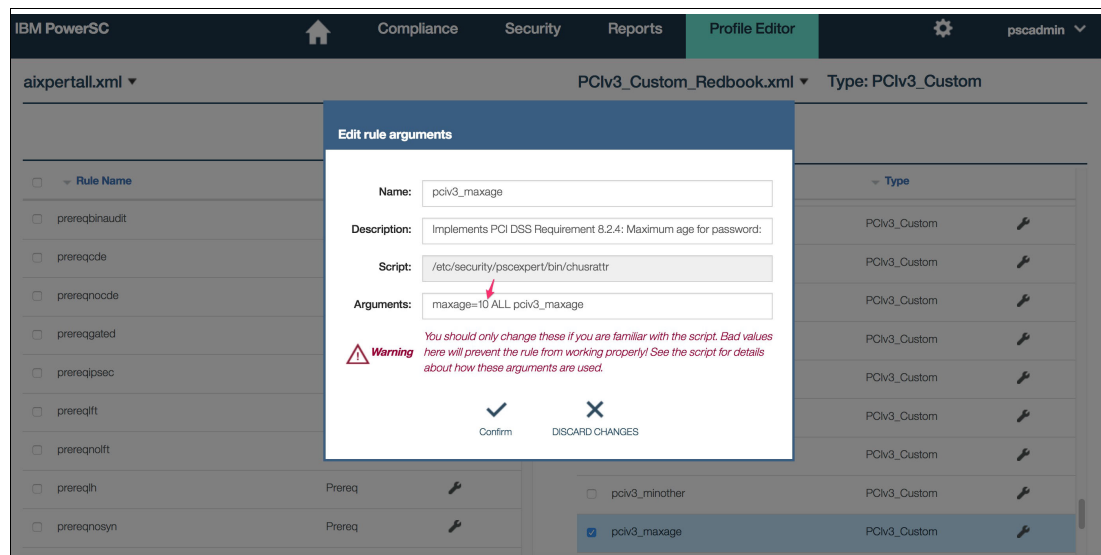


Figure 3-36 Changing the value of maxage

After completing all of the changes in the Custom Profile, the profile must be copied to all existing groups and the machines that you need to apply this profile. In this case, the AIX machine belongs to more than one group.

6. Select the groups to copy the profile to and click **OK**, as shown in Figure 3-37.

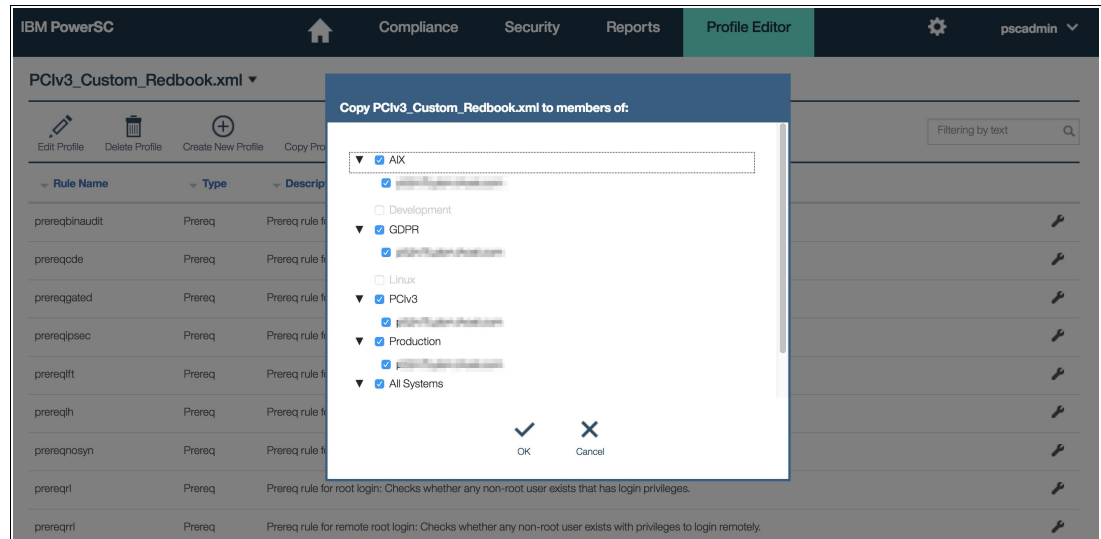


Figure 3-37 Copying the profile to groups and machines

7. In the Compliance tab, select the profile and click **OK**, as shown in Figure 3-38.

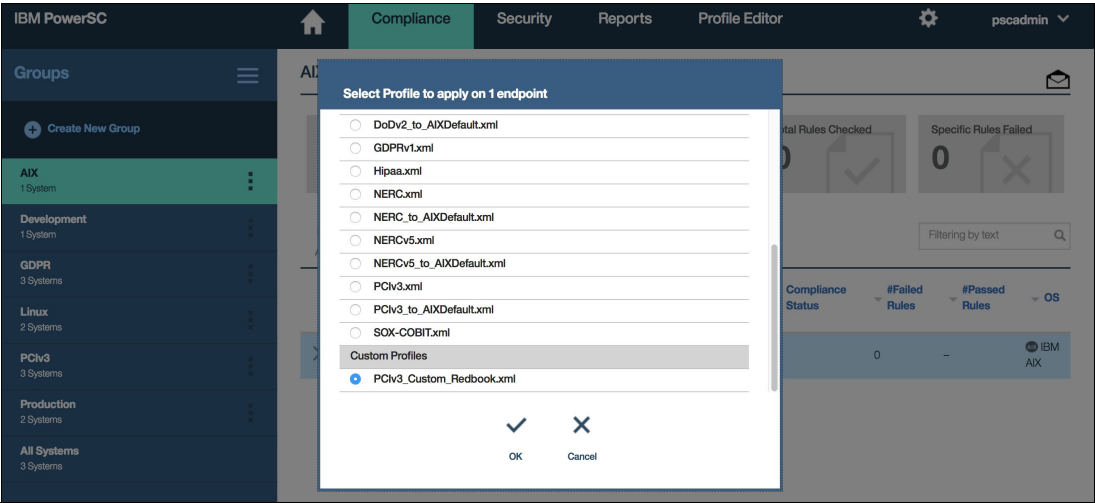


Figure 3-38 Selecting the profile to apply

Figure 3-39 shows the AIX system receiving the profile.

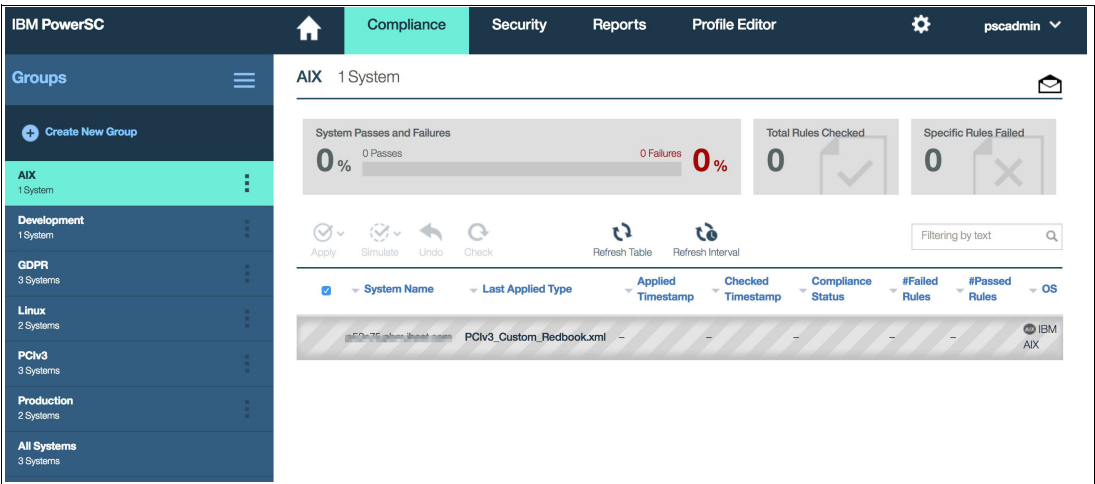


Figure 3-39 Applying the profile

After applying the new custom profile, you can see some rules failed, but you can run the check process to confirm if everything is in place, as shown in Figure 3-40.

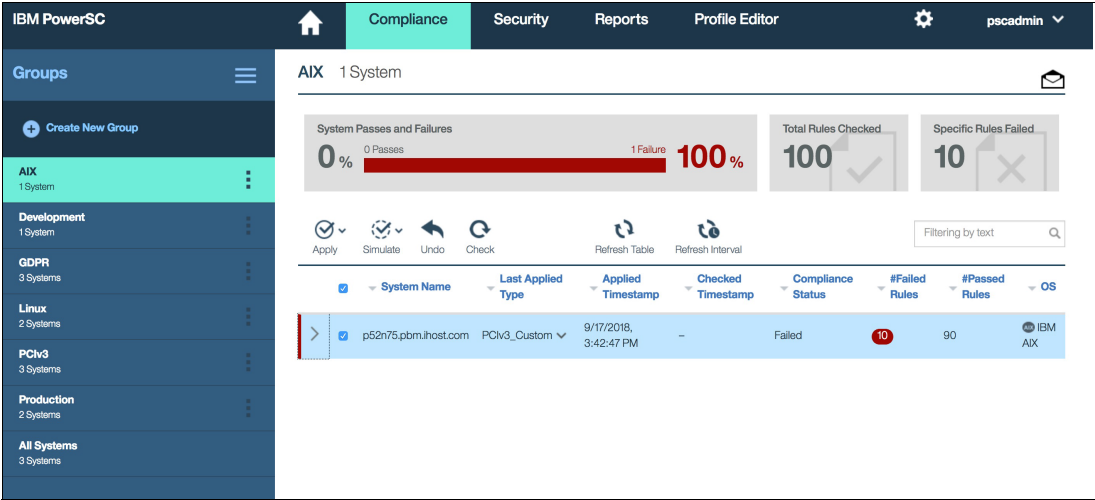


Figure 3-40 Report after applying the new profile

After reviewing the process, you can see that all rules passed, as shown in Figure 3-41.

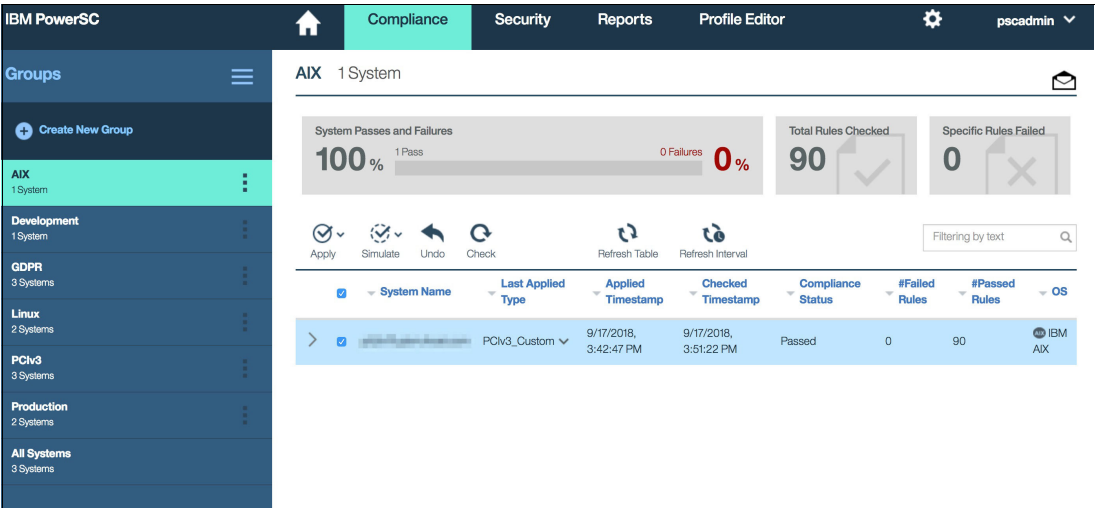


Figure 3-41 Applied profile passed

Also, a custom profile can be created to use rules from more than one profile. In this example, another custom profile is created that uses the GDPR and SOX-COBIT profiles.



Complete the following steps:

- 1. In the Profile Editor, select the GDPR profile, as shown in Figure 3-42.

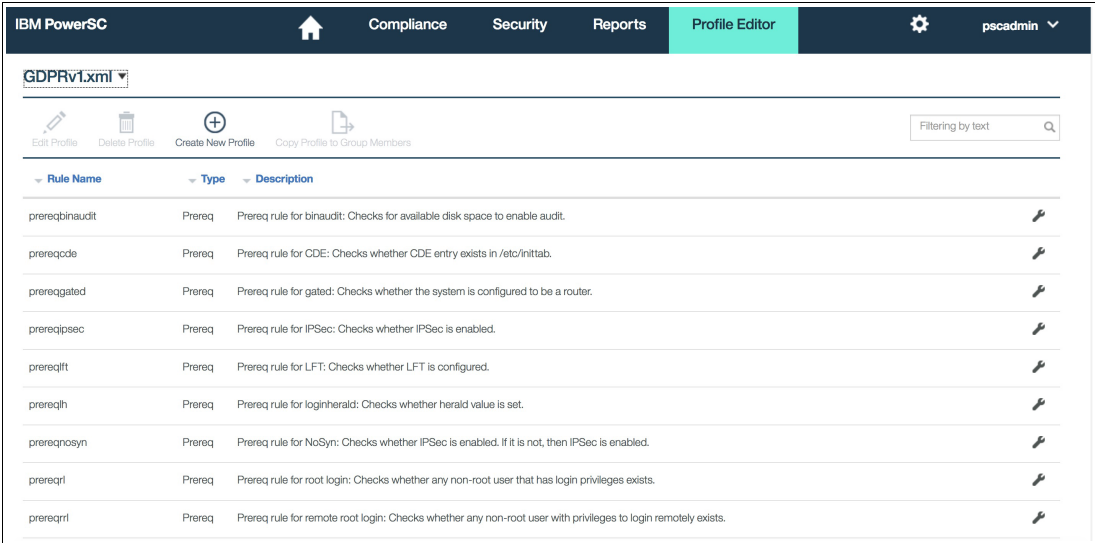


Figure 3-42 Combining profiles

- 2. Click **Create New Profile** and add the name and type. A new profile is created that is named GDPRv1\_S0X-COBIT\_Custom, as shown in Figure 3-43.

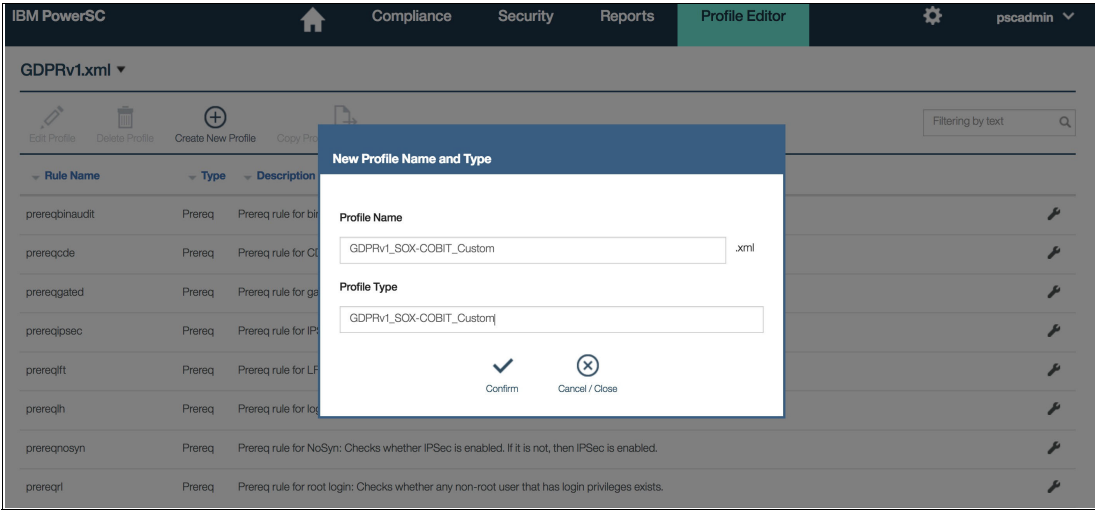


Figure 3-43 Creating a profile

3. Select the rules from GDPR that applies to your environment and move to the new custom profile, as shown in Figure 3-44.

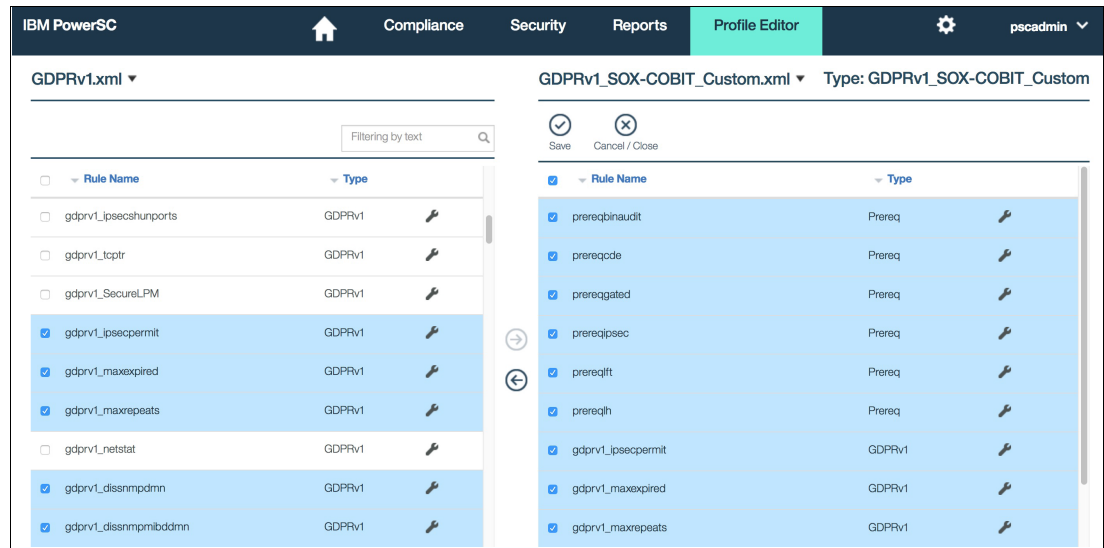


Figure 3-44 Selecting the rules to be apply to the new profile

4. You can now add rules from the SOX-COBIT profile. When you are finish, click **Save** to save the profile, as shown in Figure 3-45.

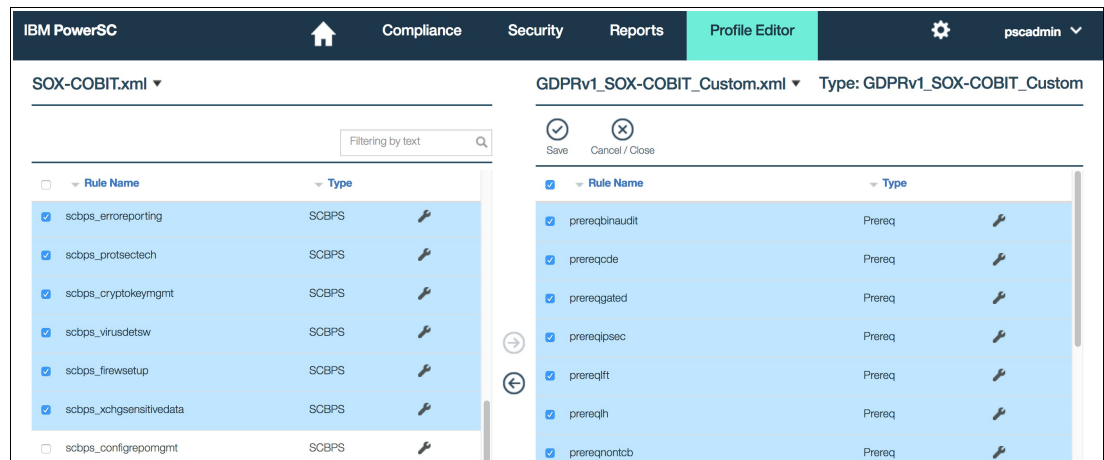


Figure 3-45 Selecting rules from SOX-COBIT

5. Copy the new profile to the groups and machines, as shown in Figure 3-46 on page 101.

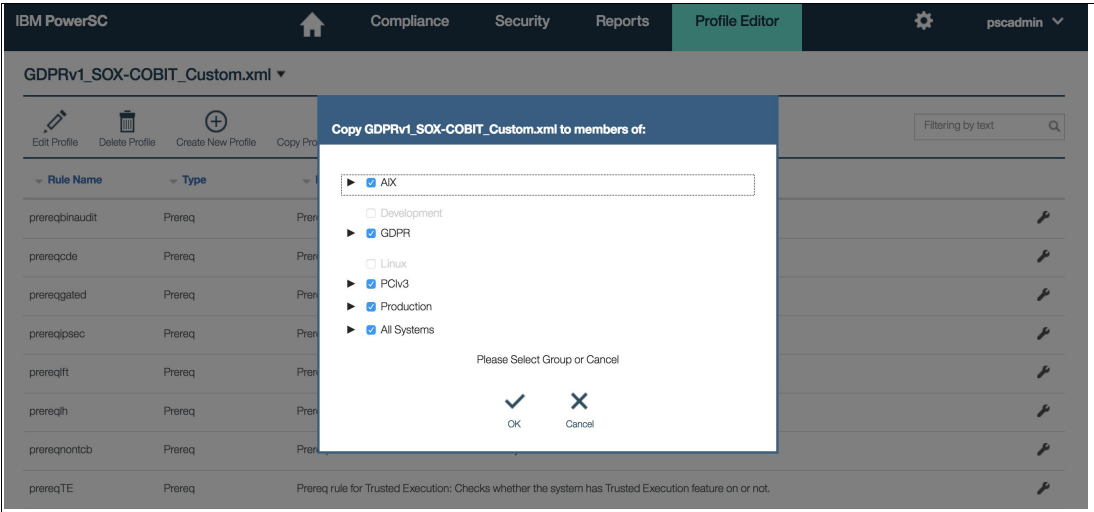


Figure 3-46 Distributing the new profile

6. In the Compliance page, select the group (our example uses the production group). Select the AIX machine and the custom profile GDPRv1\_SOX-COBIT\_Custom, and apply it, as shown in Figure 3-47.

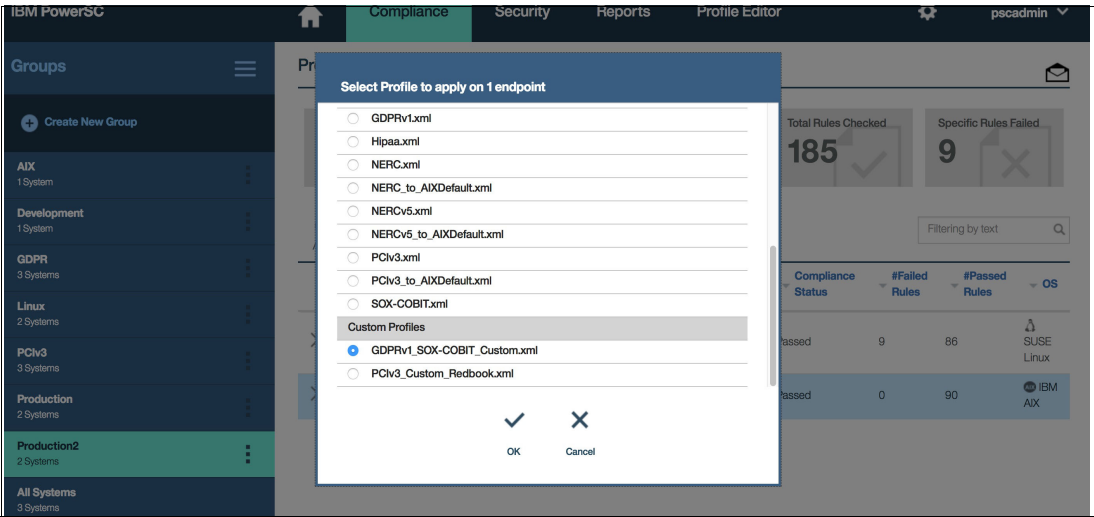


Figure 3-47 Applying the profile to endpoint

Figure 3-48 shows that the profile successfully applied to the endpoint.

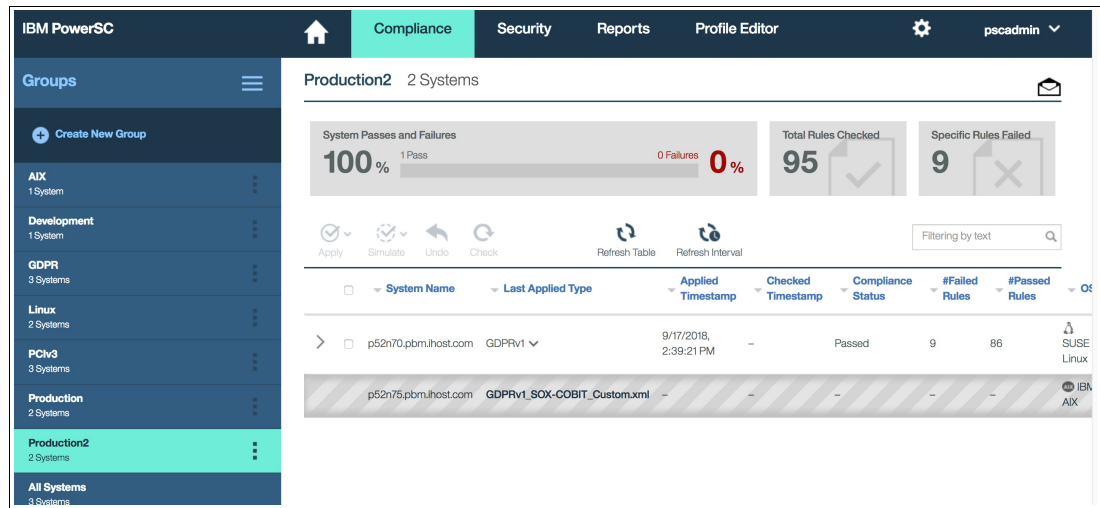


Figure 3-48 Profile applied completed successfully

After the profile is applied, you can see that some rules failed for this particular profile, as shown in Figure 3-49. You can use the process of applying to review the pre-requisites and refine the custom profile before applying it again. You also can use the Simulate process against the machine to re-test the profile while looking for a successful completion.

**Note:** Use a test or sandbox machine to test the profiles before going into production.

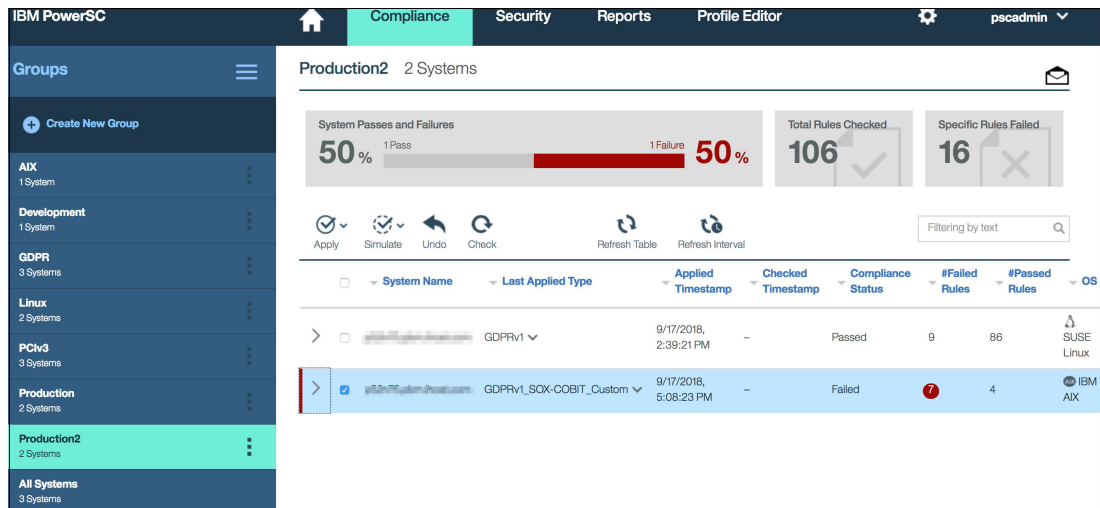


Figure 3-49 Report of the profiles applied

- Run the process again and check the report pane. You see the process completed successfully and all rules passed, as shown in Figure 3-50.

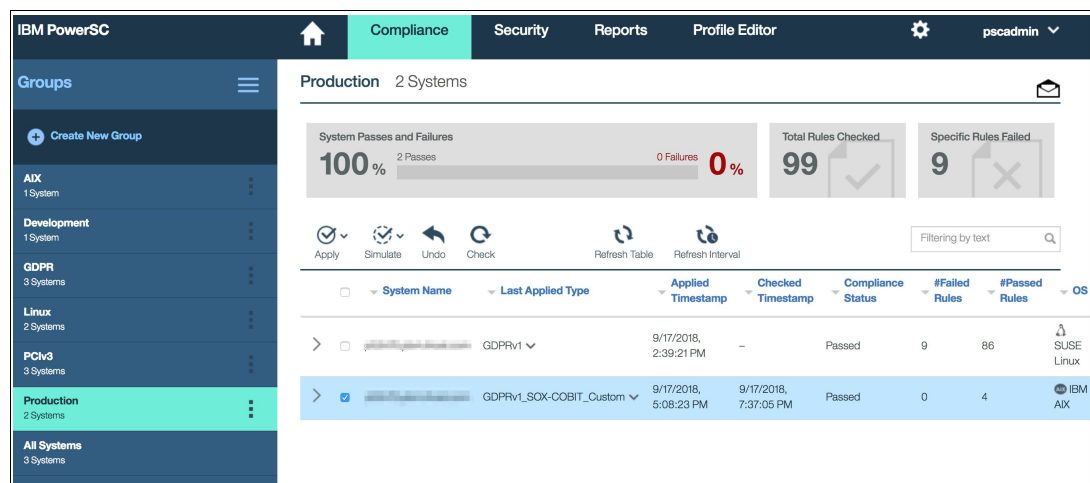


Figure 3-50 Success profile completion report

## 3.8 Importing custom profiles not created with IBM PowerSC

For more information about how to import custom files, see [IBM Knowledge Center](#).

Normally, all profiles are stored in the `/etc/security/aixpert/custom` directory. If you have a profile or want to create a profile by using the command line, complete the following steps to import it in the IBM PowerSC GUI:

- In the `/etc/security/aixpert/custom` directory, select the profile that want to use and copy it to a new profile. In our example, we used `PCIv3_Custom_Redbook.xml`, as shown in Example 3-1.

Example 3-1 Copying the profiles into a new profile

```
p52n75:/etc/security/aixpert/custom #
p52n75:/etc/security/aixpert/custom # ls -al
total 1336
drwxr-xr-x  2 bin      bin          4096 Sep 17 17:36 .
drwxr-xr-x 10 bin      bin          4096 Sep 17 16:34 ..
-r-x-----  1 root     system       60367 Sep 13 16:10 DataBase.xml
-r-x-----  1 root     system       67741 Sep 13 16:10 DoDv2.xml
-r-x-----  1 root     system       35913 Sep 13 16:10 DoDv2_to_AIXDefault.xml
-r-x-----  1 root     system       76571 Sep 13 16:10 GDPRv1.xml
-rw-r--r--  1 root     system       11099 Sep 17 17:36
GDPRv1_SOX-COBIT_Custom.xml
-r-x-----  1 root     system       21660 Sep 13 16:10 Hipaa.xml
-r-x-----  1 root     system       41142 Sep 13 16:10 NERC.xml
-r-x-----  1 root     system       34651 Sep 13 16:10 NERC_to_AIXDefault.xml
-r-x-----  1 root     system       36942 Sep 13 16:10 NERCv5.xml
-r-x-----  1 root     system       31684 Sep 13 16:10 NERCv5_to_AIXDefault.xml
-r-x-----  1 root     system       79253 Sep 13 16:10 PCIv3.xml
-rw-r--r--  1 root     system       75242 Sep 17 16:34 PCIv3_Custom_Redbook.xml
-r-x-----  1 root     system       58525 Sep 13 16:10 PCIv3_to_AIXDefault.xml
-r-x-----  1 root     system       12436 Sep 13 16:10 SOX-COBIT.xml
```

```
p52n75:/etc/security/aixpert/custom #
p52n75:/etc/security/aixpert/custom # cp PCiv3_Custom_Redbook.xml
PCiv3_Custom_Production.xml
p52n75:/etc/security/aixpert/custom #
p52n75:/etc/security/aixpert/custom #
```

---

2. In /opt/powersc/uiServer/bin, use the convertProfileToBean.sh script to create an IBM PowerSC GUI version of your custom profile XML file. The converted file is saved in the original directory, with the original name of the file, and an .xml, as shown in Example 3-2.

*Example 3-2 Running the script to create an IBM PowerSC version of the profile*

---

```
p52n75:/opt/powersc/uiServer/bin # ./convertProfileToBean.sh
/etc/security/aixpert/custom/PCiv3_Custom_Production.xml
/opt/powersc/uiServer/bin/uisserv
OUTPLAY=
HOME=/
USER=root
JBOSS_HOME=
TEMP=
TMP=
_STARTED=1
LIBPATH=/opt/powersc/uiServer/bin/jre/bin/default:/opt/powersc/uiServer/bin/jre/li
b/ppc64:/opt/powersc/uiServer/bin/jre/lib/ppc64/default:/opt/powersc/uiServer/bin/
jre/bin:
running class com.rocketsoft.nm.vertical.powersc.ProfileToBeanConverter..
```

---

3. Move the \*.xml.xml file into the /opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles/ directory on the IBM PowerSC UI server as shown in Example 3-3. Then you need to restart the PowerSC UI server to read the new profile and display it in the GUI.

*Example 3-3 Move the file to the PowerSC UI server*

---

```
p52n75:/etc/security/aixpert/custom # ls
DataBase.xml          GDPRv1_SOX-COBIT_Custom.xml    NERCv5.xml
PCiv3_Custom_Production.xml.xml
DoDv2.xml             Hipaa.xml
NERCv5_to_AIXDefault.xml    PCiv3_Custom_Redbook.xml
DoDv2_to_AIXDefault.xml    NERC.xml                      PCiv3.xml
PCiv3_to_AIXDefault.xml
GDPRv1.xml            NERC_to_AIXDefault.xml
PCiv3_Custom_Production.xml    SOX-COBIT.xml

p52n75:/etc/security/aixpert/custom # mv PCiv3_Custom_Production.xml.xml
/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles/

p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles # ls -al
total 752
drwxr-xr-x  2 root    security    4096 Sep 18 13:21 .
drwxr-xr-x 33 root    security    4096 May 30 08:19 ..
-rw-r--r--  1 root    system      7063 Sep 17 17:34
GDPRv1_SOX-COBIT_Custom.xml.jxo
-rw-r--r--  1 root    system      23756 Sep 17 17:34
GDPRv1_SOX-COBIT_Custom.xml.xml
-rw-r--r--  1 root    system     146050 Sep 18 13:14
PCiv3_Custom_Production.xml.xml
```

```

-rw-r--r--    1 root    system      45714 Sep 18 10:56
PCIv3_Custom_Redbook.xml.jxo
-rw-r--r--    1 root    system      146270 Sep 18 10:56
PCIv3_Custom_Redbook.xml.xml
p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles #

p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles #
p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles # stopsrc -s
pscuiserver
0513-044 The pscuiserver Subsystem was requested to stop.
p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles #
p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles # startsrc
-s pscuiserver
0513-059 The pscuiserver Subsystem has been started. Subsystem PID is 17367480.
p52n75:/opt/powersc/uiServer/knowledge/site/powerscui/aixpertProfiles #

```

Figure 3-51 shows how to import and use the custom profile created.

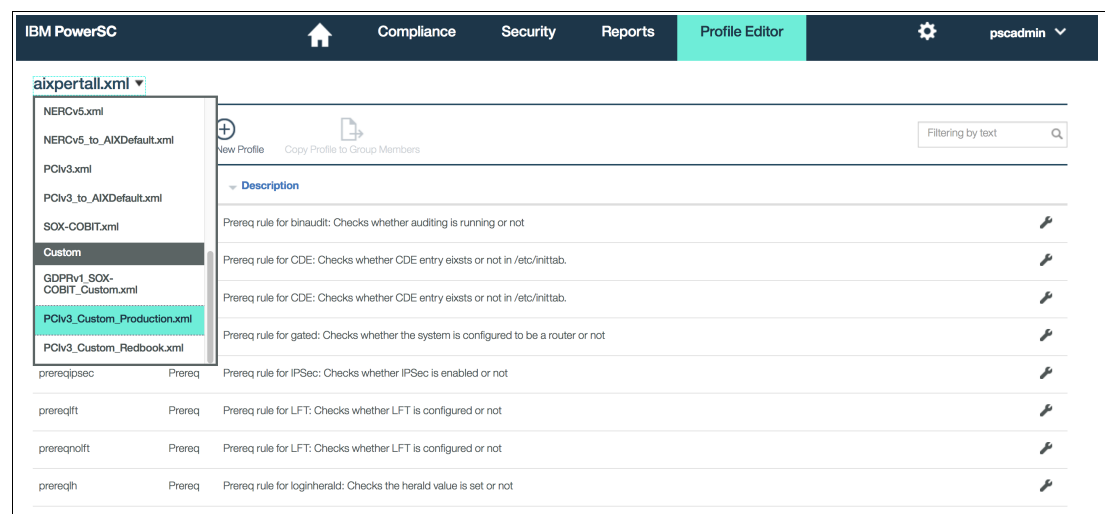


Figure 3-51 Using the custom profile created

## 3.9 Applying the PCIV3 profile to an AIX LPAR

Complete the following steps to apply the profile by using the IBM PowerSC GUI:

1. Select the group and the machine to which you want to apply the profile. In our example, the group AIX and one machine are selected, as shown in Figure 3-52.

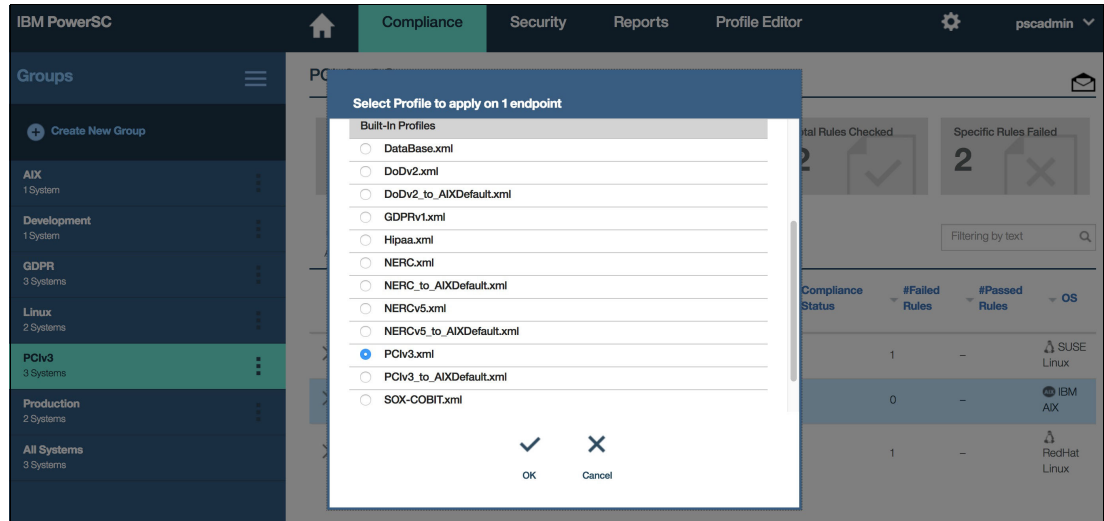


Figure 3-52 Selecting the group and machine

2. Select the profile (see Figure 3-53).

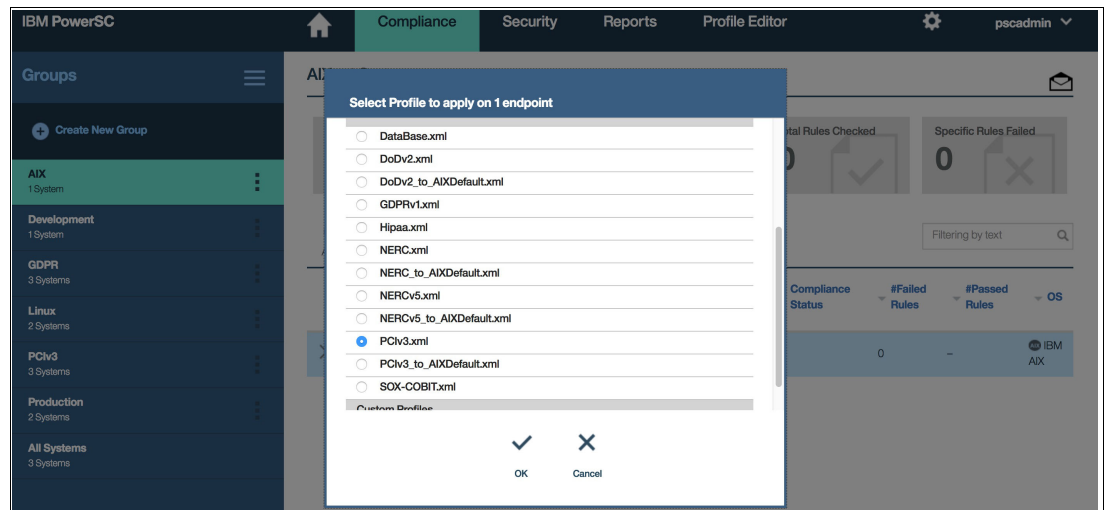


Figure 3-53 Selecting the profile to apply to endpoints



3. Click **OK** to start applying the new profile, as shown in Figure 3-54.

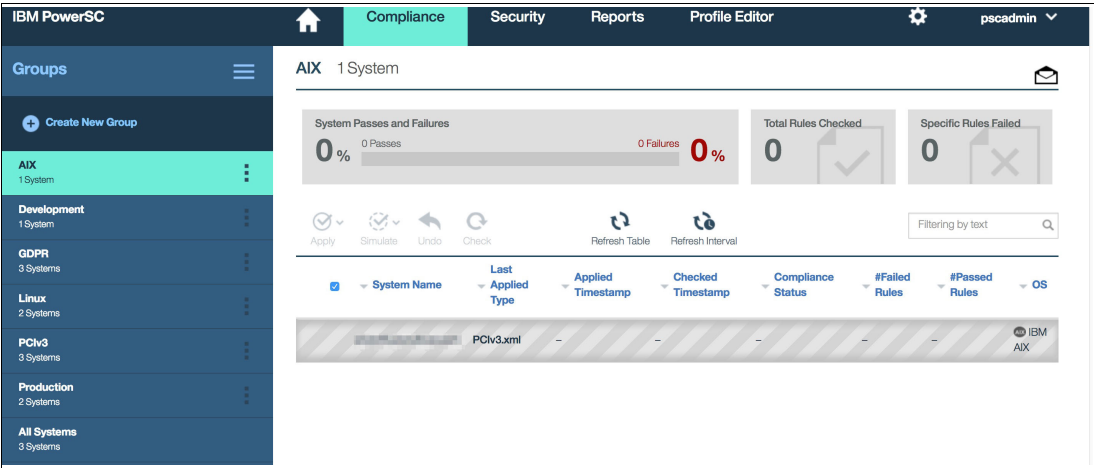


Figure 3-54 Applying the profile to the selected endpoints

After a few minutes, you can see the completion results, as shown in Figure 3-55. Depending on the machine, this process can take approximately 5 minutes to complete. You can check the status of the process by using the command line.

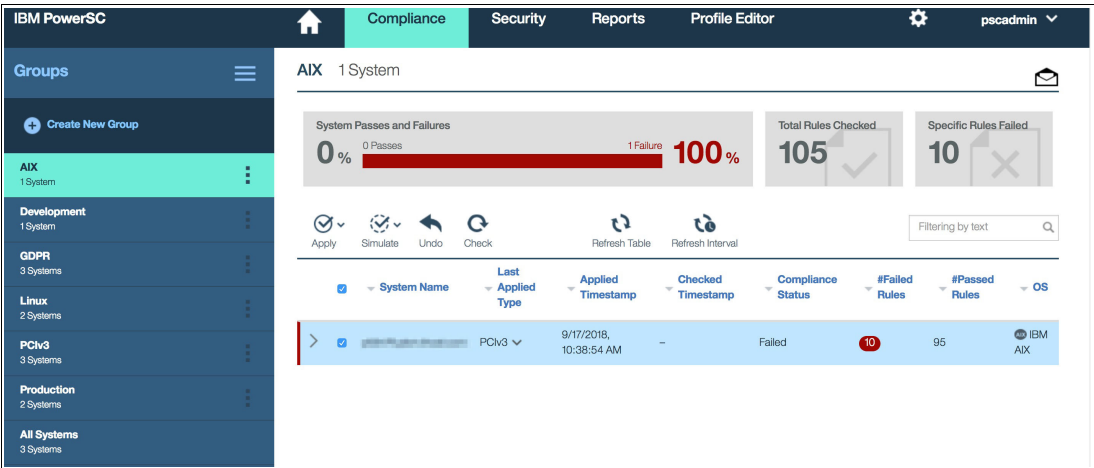


Figure 3-55 Report of applying the profile to endpoints

Figure 3-56 on page 108 shows the results of the number of Passed and Failed rules. The Compliance Status shows as Failed because you must review the failed rules and attempt to fix them.

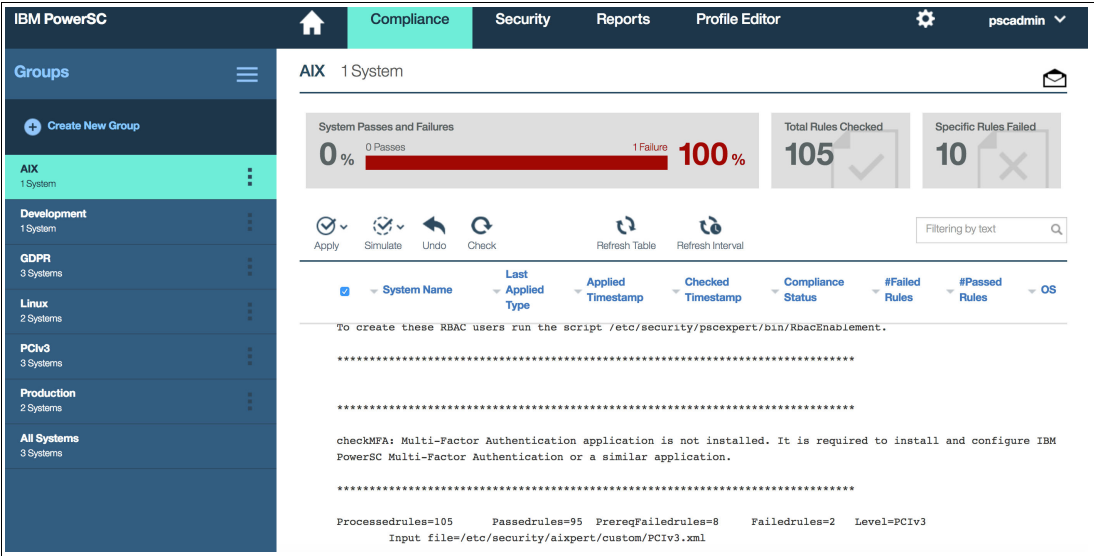


Figure 3-56 Profile application status passed and failed rules

### 3.9.1 Simulate first

Complete the following steps to use the IBM PowerSC GUI to simulate the application of a profile to selected endpoints:

1. Select the machine.
2. In the Simulate tab, select the profile that you and simulate (in our example, PCiv3).
3. Click **OK**, as shown in Figure 3-57.

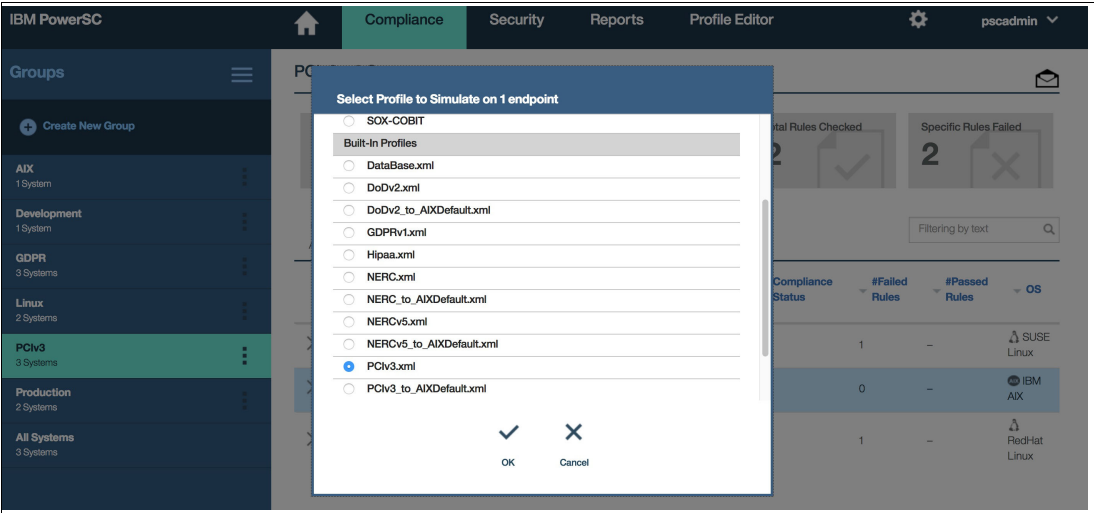


Figure 3-57 Selecting profile to simulate

The simulation process starts, as shown in Figure 3-58.

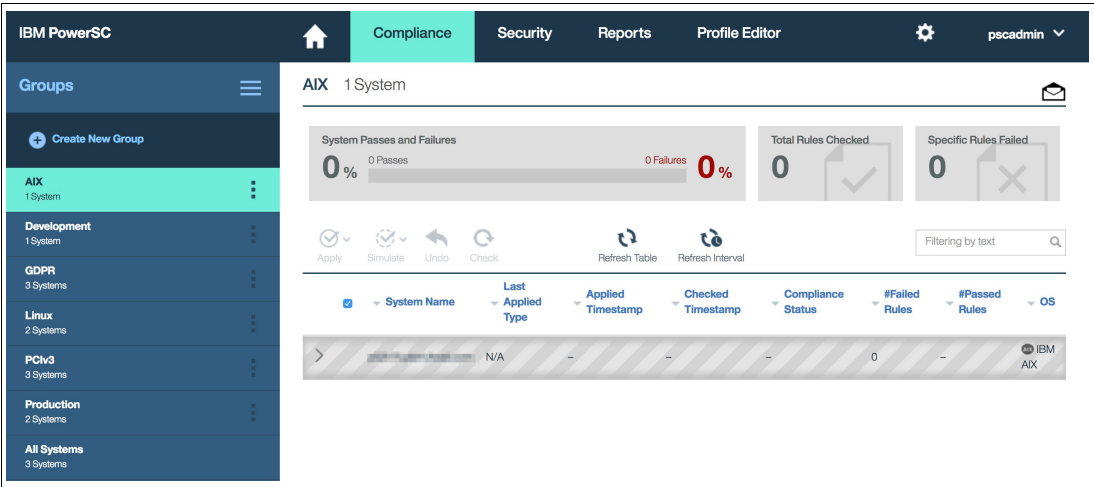


Figure 3-58 PowerSC GUI Compliance pane - Simulation starts

The results of the simulation are shown in Figure 3-59.

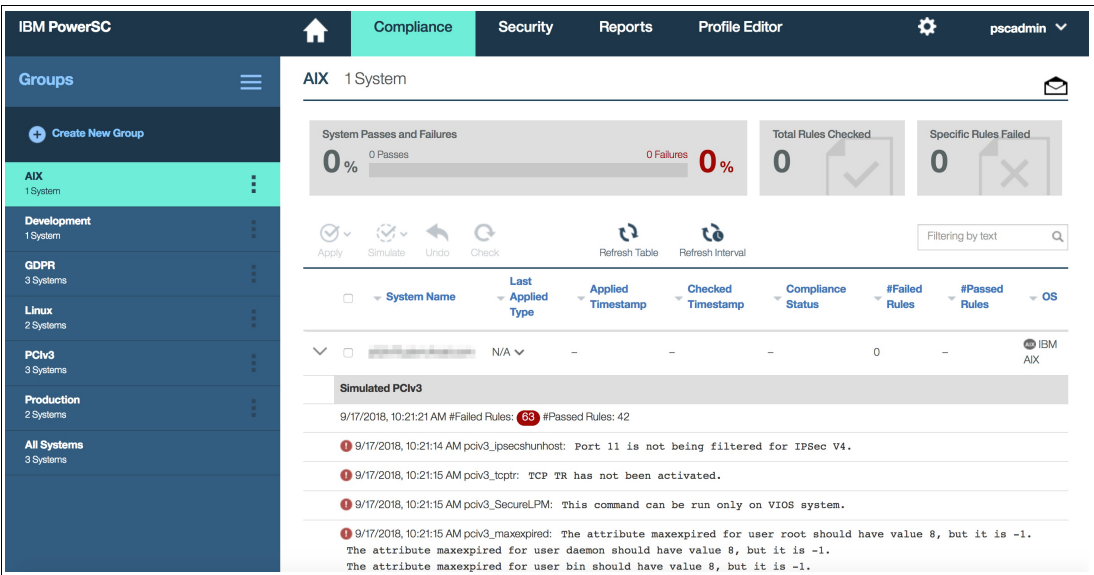


Figure 3-59 PowerSC GUI Compliance pane - Simulation results

For all simulated and processed rules processes, you can analyze, review, and adjust what must be changed for all of the rules in the profile to be applied successfully.





# Real-Time File Integrity Monitoring

Many critical files exist on a system that contain sensitive data, such as configuration details and user information. From a security perspective, it is important to monitor changes that are made to these sensitive files. File Integrity Monitoring (FIM) is a method that can detect modification of critical files.

A File Integrity Monitoring solution must have the following characteristics:

- ▶ The alerts are in real time.
- ▶ A centralized way is used to monitor FIM alerts of multiple endpoints.
- ▶ Provides important details, such as who modified the file, when was it modified, and how was it modified.

This chapter introduces the PowerSC File Integrity Monitoring (FIM) component, which consists of real-time monitoring for IBM AIX and Linux systems to ensure that these systems are configured correctly and are consistently in a compliant state.

This chapter contains the following topics:

- ▶ 4.1, “PowerSC Real-Time Compliance” on page 112
- ▶ 4.2, “AIX Trusted Execution” on page 128
- ▶ 4.3, “Linux auditd” on page 147
- ▶ 4.4, “FIM reporting with PowerSC GUI” on page 153

## 4.1 PowerSC Real-Time Compliance

The PowerSC Real-Time Compliance (RTC) component can be used to generate real-time alerts whenever a monitored file is modified. By using the AHAFS<sup>1</sup> event monitoring technology, PowerSC RTC monitors a list of files. When any file modification is detected, it generates alert in the following ways:

- ▶ Email alerts
- ▶ Log message to a file
- ▶ SNMP message to your monitoring server
- ▶ Alert to PowerSC GUI server

PowerSC RTC works with the PowerSC Compliance Automation to provide automatic compliance check feature. Whenever a monitored file is changed, RTC runs the **pscexpert** command to capture and notify if any compliance violations occurred. For more information about the PowerSC compliance automation feature, see Chapter 3, “Compliance automation” on page 75.

A high-level overview of the RTC architecture is shown in Figure 4-1.

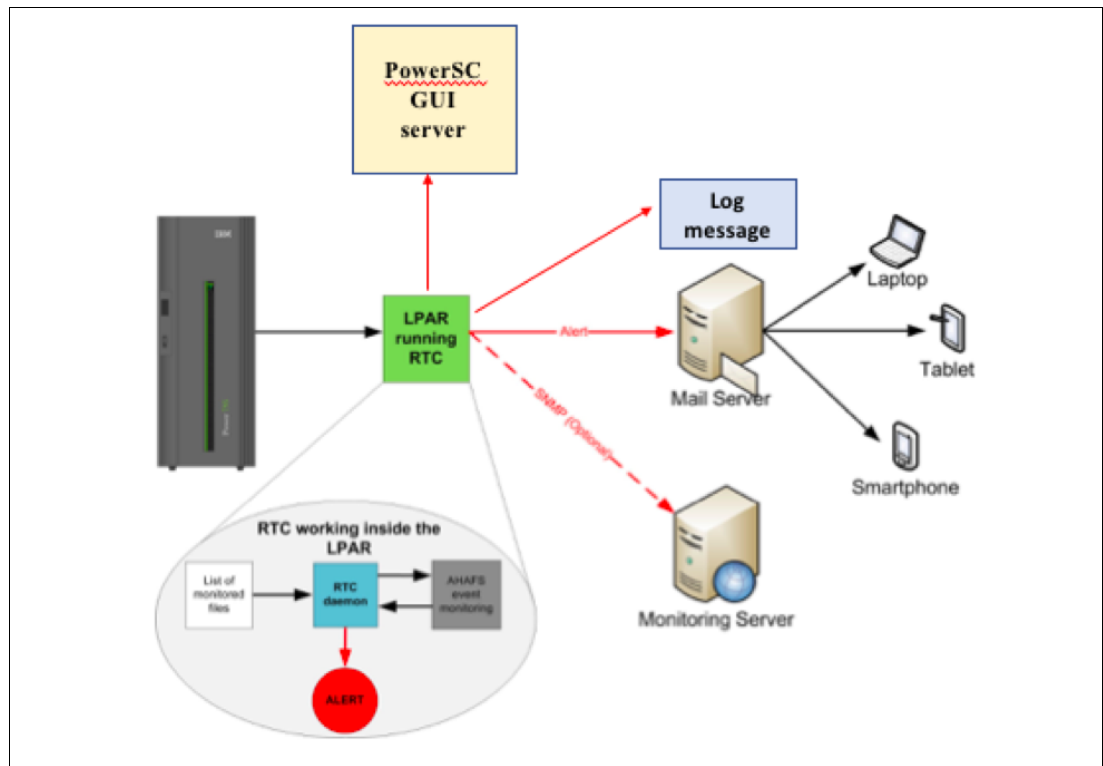


Figure 4-1 High-level RTC architecture

<sup>1</sup> At the core of the AIX Event Infrastructure is a pseudo-filesystem: Autonomic Health Advisor File System (AHAFS), which is implemented as a kernel extension. AHAFS mainly acts as a mediator to take the requests of event registration, monitoring, and unregistering from the processes interested in monitoring for events. It forwards the requests to the corresponding event producers (code responsible for triggering the occurrence of an event) in the kernel space, processes the callback functions when the event occurs, and notifies the registered users or processes with useful information.

## 4.1.1 Detailed implementation

RTC can act as a stand-alone component for monitoring the configured files in the predefined file list and notifying the administrators, in real time, of any changes to these files. RTC can also work with the Compliance Automation to ensure that all policies are adhered to and no unauthorized changes go unnoticed to the administrators.

RTC provides the following monitoring configuration options:

- ▶ Content monitoring: Checks if the content of the file was modified
- ▶ Attribute monitoring: Verifies whether the file permissions or ownership were changed

It is possible to configure both monitoring functions for the same file. You also can change the type of monitoring of a file at any time. RTC can also monitor directories to notify if any new files or directories are created or deleted inside the monitored directory.

RTC uses the AIX AHAFS feature for the detection of any potential configuration change, which reduces monitoring overhead to a bare minimum. No scheduled monitoring jobs are run because configuration changes are captured in real time by this technology.

When installed, the software provides a default list of files to monitor for changes, and this list can be modified by adding or removing files and directories as needed. RTC is configured by specifying the alert notification emails. This feature can also be extended to support various options through a configuration file.

If the PowerSC GUI agent is configured on the LPAR where RTC is deployed, the agent automatically sends RTC alerts to the PowerSC GUI server. This feature helps to monitor RTC events centrally on PowerSC GUI server.

RTC provides the following benefits:

- ▶ Monitors file modifications of a predefined list of files by using the AIX event infrastructure (AHAFS)
- ▶ Checks for compliance violations on file changes by using the **pscxpert** command
- ▶ Sends alerts by using email, log messages, or SNMP traps on compliance violations and file modification events
- ▶ Integrates with the PowerSC GUI server for centralized management and reporting
- ▶ Sends emails to administrators by way of their hand-held devices or any SNMP server
- ▶ Sends administrators daily FIM reports through PowerSC GUI automated reporting capabilities

After finishing the initial installation and configuration, the RTC daemon `rtcd` uses the monitoring capabilities of the AHAFS and validates if any of the files that are contained in the predefined list were changed. If a violation occurred, an alert is sent to a configured list of emails and the monitoring continues.

The RTC functional flow is shown in Figure 4-2.

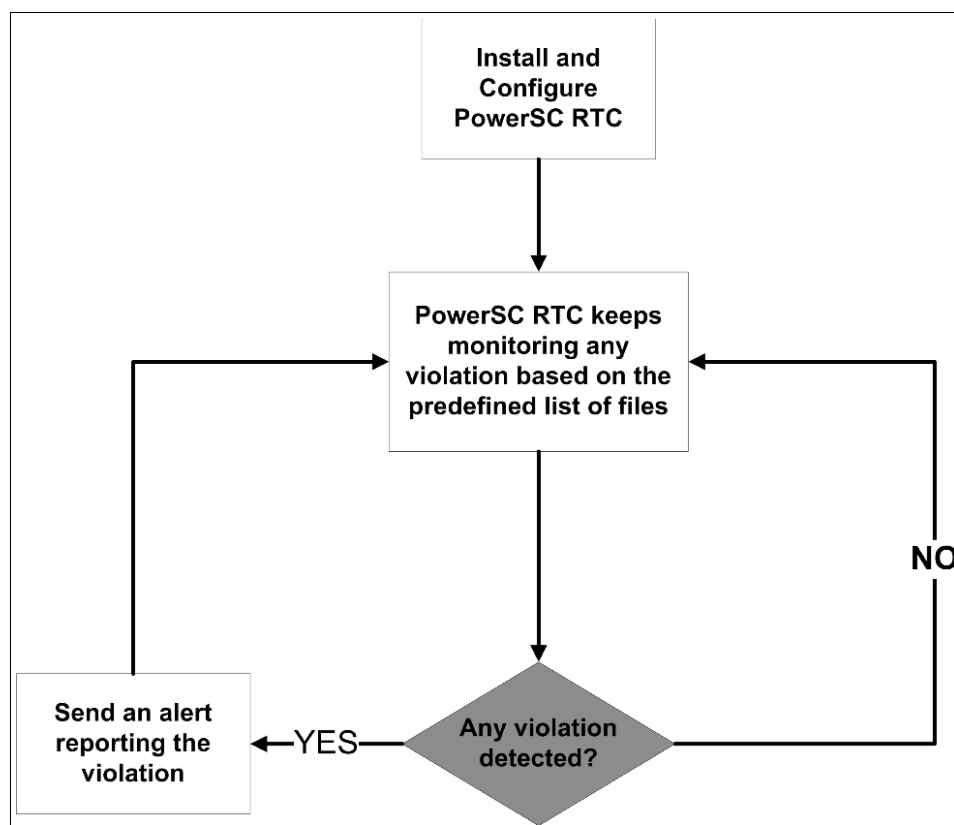


Figure 4-2 RTC functional flow

## 4.1.2 Deployment considerations

Before RTC is deployed, review the following considerations, which are described next:

- ▶ Software installation requirements
- ▶ Necessary file sets

### Software installation requirements

Ensure that you are running one of the following AIX operating systems on the system where you are installing the RTC component:

- ▶ IBM AIX 6.1 with Technology Level 7, or later, with AIX Event Infrastructure for AIX and AIX Clusters (bos.ahafs 6.1.7.0), or later
- ▶ IBM AIX 7.1 with Technology Level 1, or later, with AIX Event Infrastructure for AIX and AIX Clusters (bos.ahafs 7.1.1.0), or later
- ▶ IBM AIX 7.2, or later, with AIX Event Infrastructure for AIX and AIX Clusters (bos.ahafs 7.2.0.0), or later



To ensure that you meet all the software requirements, run the commands that are shown in Example 4-1.

*Example 4-1 Ensuring your configuration meets the requirements*

```
# oslevel -s
7200-01-01-1642
root@rtc-server> # lslpp -L bos.ahafs
```

Fileset	Level	State	Type	Description (Uninstaller)
-----	-----	-----	-----	-----
bos.ahafs	7.2.1.0	C	F	Aha File System

Example 4-1 shows that you deployed an IBM AIX 7.2 with Technology Level 1 and bos.ahafs version 7.2.1.0.

For more information about IBM AIX Technology Level or to download the necessary fixes, see this IBM Fix Central [web page](#).

## Necessary file sets

To install RTC, you need the following file sets, which can be found in the PowerSC Standard Edition:

- ▶ powerscStd.rtc
- ▶ powerscStd.license

## 4.1.3 Installation

The RTC component is provided in the PowerSC Standard Edition. It is not part of the base AIX operating system. PowerSC RTC is installed directly on the host and does not require any configuration on PowerVM.

Complete the following steps to install RTC:

1. Copy the PowerSC Standard Edition ISO file to the LPAR and loopmount it:

```
loopmount -i /software/POWERSC_STD_EDITION_V1.2_62018.iso -o "-V cdrfs -o ro"
-m /mnt
```

2. Go to the installation path (/mnt/installp/ppc), as shown in Figure 4-3.

```
# pwd
/mnt/installp/ppc
#
# ls
.toc                powerscStd.rtc      powerscStd.uiAgent
openpts.verifier    powerscStd.svm      powerscStd.uiServer
powerscStd.ice       powerscStd.tnc_commands powerscStd.vlog
powerscStd.license  powerscStd.tnc_lib
powerscStd.msg       powerscStd.tnc_pm
#
```

*Figure 4-3 Installation path*

3. Install RTC and license filesets by using `smitty installp`, as shown in Figure 4-4.

```
* INPUT device / directory for software
* SOFTWARE to install
PREVIEW only? (install operation will N
COMMIT software updates?
SAVE replaced files?
AUTOMATICALLY install requisite softwar
EXTEND file systems if space needed?
OVERWRITE same or newer versions?
VERIFY install and check file sizes?
Include corresponding LANGUAGE filesets
DETAILED output?
Process multiple volumes?
ACCEPT new license agreements?
Preview new LICENSE agreements?

INVOKE live update?
Requires /var/adm/ras/liveupdate/lvupda

WPAR Management
  Perform Operation in Global Environ
  Perform Operation on Detached WPARs
    Detached WPAR Names
  Remount Installation Device in WPAR
  Alternate WPAR Installation Device

Move cursor to desired item and press F7. Use arrow keys to
ONE OR MORE items can be selected.
Press Enter AFTER making all selections.

[TOP]
#-----
#
# KEY:
# @ = Already installed
#-----
openpts.verifier
+ 1.0.0.1 Open Platform Trust Services - verifier

powerscStd.ice
@ 1.2.0.0 IBM PowerSC Standard Profile

> powerscStd.license
@ 7.1.2.0 PowerSC Standard Edition

> powerscStd.rtc
@ 1.2.0.0 Real-Time Compliance

powerscStd.svm
+ 1.2.0.0 Secure Virtual Machine

powerscStd.tnc_commands
+ 1.2.0.0 Trusted Network Connect Commands

powerscStd.tnc_lib
+ 1.2.0.0 Trusted Network Connect Libraries

powerscStd.tnc_pm
+ 1.2.0.0 Trusted Network Connect for Patch Management
```

Figure 4-4 Pane to install RTC and filesets

4. Verify the installation:

```
# lsipp -L powerscStd.rtc.rtc
Fileset                                Level State Type Description (Uninstaller)
-----
powerscStd.rtc.rtc                    1.2.0.0 C F Real-Time Compliance
```

You can also list the contents of the file set as shown in Example 4-2.

Example 4-2 Listing the content of the package

Fileset	File
-----	
Path: /usr/lib/objrepos	
powerscStd.rtc.rtc 1.2.0.0	
	/usr/sbin/rtcd
	/usr/sbin/mkrtc
Path: /etc/objrepos	
powerscStd.rtc.rtc 1.2.0.0	
	/etc/security/pscexpert/bin/rtc_lku
	/etc/security/rtc/rtcd.conf
	/etc/security/rtc/rtcd_policy.conf

Four main objects are installed with PowerSC RTC, as listed in Table 4-1.

Table 4-1 RTC files and functions

File name	Functions
usr/sbin/rtcd	RTC daemon: <ul style="list-style-type: none"><li>▶ Monitors file changes</li><li>▶ Starts <b>pscxpert</b> command to check for violations</li><li>▶ Sends alerts</li></ul>
/usr/sbin/mkrtc	The command to set up RTC subsystem.
/etc/security/rtc/rtcd.conf	The configuration file for the rtcd daemon.
/etc/security/rtc/rtcd_policy.conf	The policy file contains files and directories to be monitored by the RTC subsystem.

4.1.4 Configuration steps

After verifying the prerequisites and the installation of the software, configure PowerSC RTC by using one of the following methods:

- ▶ The command line with the **mkrtc** command
- ▶ Smitty

To avoid mistakes, use the top-level menu smitty to configure the software. The fast path to run this configuration is **smitty RTC**, as shown in Figure 4-5.

```
# smitty RTC
                                                                    Configure R
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[Entry Fields]
* Email Address (comma separated) [ ]
  Alert Information Level          [1]
  Alert Style                      [once]
  Alert Email Subject              [ ]
  Debug                           [off]
```

Figure 4-5 Configuration fast path to the RTC pane

Enter information, such as email address and email subject, as shown in Figure 4-6.

```
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[Entry Fields]
* Email Address (comma separated) [root@localhost]
  Alert Information Level          [1]
  Alert Style                      [always]
  Alert Email Subject              [RTC alert]
  Debug                           [off]
```

Figure 4-6 Entering information in the pane fields

Figure 4-7 shows the command completion and that the configuration was set up correctly.

```
Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

0513-059 The rtcd Subsystem has been started. Subsystem PID is 7078326.
3008-033 The RTC subsystem has been configured successfully.
```

Figure 4-7 Completing the changes

You can verify the status of rtcd daemon by running the `lssrc` command, as shown in Figure 4-8.

```
# lssrc -s rtcd
Subsystem          Group          PID          Status
rtcd               19267910      active
```

Figure 4-8 Verify the status of the rtcd daemon

The rtcd daemon is a subsystem under the AIX System Resource Controller (SRC); therefore, you must use the following SRC commands to manage the rtcd subsystem:

- ▶ To check the subsystem use:  
`lssrc -s rtcd`
- ▶ To stop the subsystem use:  
`stopsrc -s rtcd`
- ▶ To start the subsystem use:  
`startsrc -s rtcd`

You can also configure RTC by using the command line with the `mkrtc` command. The syntax of the `mkrtc` command is shown in Example 4-3.

Example 4-3 Displaying different options to configure RTC with `mkrtc` command

```
mkrtc -e <email,email,...> [-a <alertStyle>] [-d <debug>] [-i <infoLevel>] [-s
<emailSubject>] [-c <minchecktime>]
```

where:

<email>: Email address where alerts are sent to  
 <alertStyle>: Takes 1 of 3 values: once, event, and always. Default is once  
 <debug>: Takes 1 of 2 values: on or off. Default is off  
 <infolevel>: Takes 1 of 3 values: 1, 2, and 3. Default is 1  
 <emailSubject>: The text for the Subject-line for the email alert  
 (minchecktime): Minimum interval time that RTCD uses for compliance checks.  
 Default is 30 min. 0 indicates never.

**Note:** Use one of the following methods to uninstall RTC, if needed:

```
mkrtc -u
smitty RTC - Unconfigure Real-Time Compliance Subsystem
```

The RTC feature can also be managed from the PowerSC GUI server. For this process, the PowerSC GUI agent must be configured on this LPAR. For more information about how to configure PowerSC GUI agent, see 2.4, “Installing the UIAgent” on page 40.

After you start RTC on the LPAR, it appears as running (shown as tick mark) in the PowerSC GUI Security window, as shown in Figure 4-9.

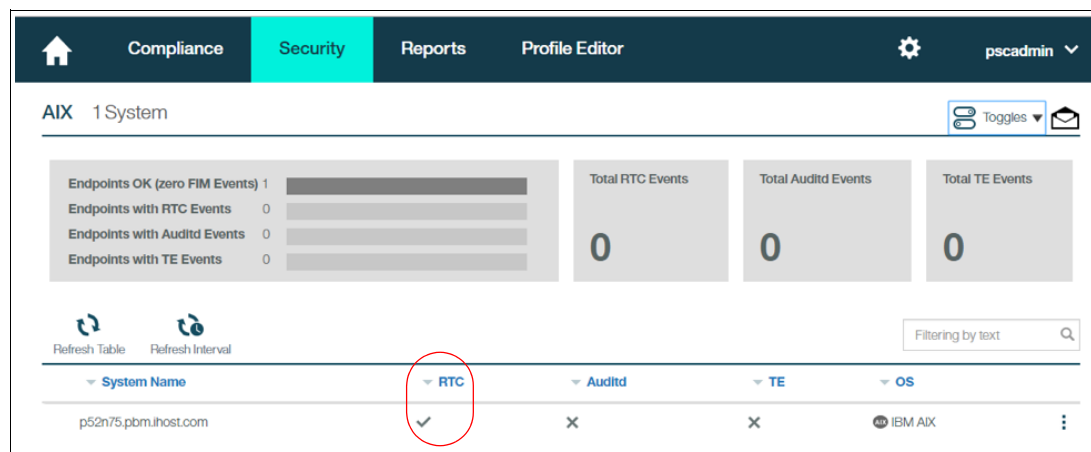


Figure 4-9 RTC running in the LPAR

## 4.1.5 RTC configuration files

The following main RTC configuration files are placed in the `/etc/security/rtc` directory:

- ▶ `rtcd.conf`
- ▶ `rtcd_policy.conf`

### rtcd.conf file

The `rtcd.conf` file defines the RTC configuration details. Various fields in this file can be configured to define the RTC configuration. The options to configure the RTC subsystem fields are listed in the Table 4-2.

Table 4-2 Options to configure the RTC subsystem

Field	Options
email	List of email addresses to which to send alerts.
emailSubject	Specifies the subject of the alert email.
infolevel	Specifies the information level of file modifications. Valid values are 1, 2, and 3
alertStyle	Specifies alert style. Valid values are: <ul style="list-style-type: none"> <li>▶ Once: Alerts once for the same set of compliance violations (default setting)</li> <li>▶ Event: Alerts once for the same set of compliance violations, but continues to alert for each file changes</li> <li>▶ Always: Alerts compliance violations and file changes on any file change event</li> </ul>
debug	Specifies whether to turn on debug. Valid values are on and off.

Field	Options
alertMsgSize	Specifies the alert message size. Valid values are: <ul style="list-style-type: none"> <li>► Verbose: Provides the entire message (default setting).</li> <li>► limited: Limits the size of the alert message to the first violation and the first event. If more than one violation or event occurs, it is indicated in the message.</li> </ul>
snmptrap_enable	Specifies to enable or disable SNMP trap. Valid values are yes or no.
snmptrap_host	Host that is receiving the SNP traps.
snmptrap_community	SNMP community name.
snmptrap_oid	Specifies OID for the trap message.
locallogfile	Specifies to enable local logging for RTC alerts.
minCheckTime	Specifies the minimum amount of time between compliance verifications in minutes. This setting ensures compliance checks regularly (even without file modification triggers) to catch user file creations that can have compliance implications; for example, rhost file in users' home directory. The default minimum time is 30 minutes. A value of 0 indicates never.
complianceCheck	Specifies whether the rtd calls pscxprt to do compliancy checks. Valid values are on and off.

You can manually edit these options in the configuration file or use the PowerSC GUI.

To configure RTC by using the PowerSC GUI, go to the Security page of the PowerSC GUI server, select the LPAR, and click **Configure RTC**, as shown in Figure 4-10.

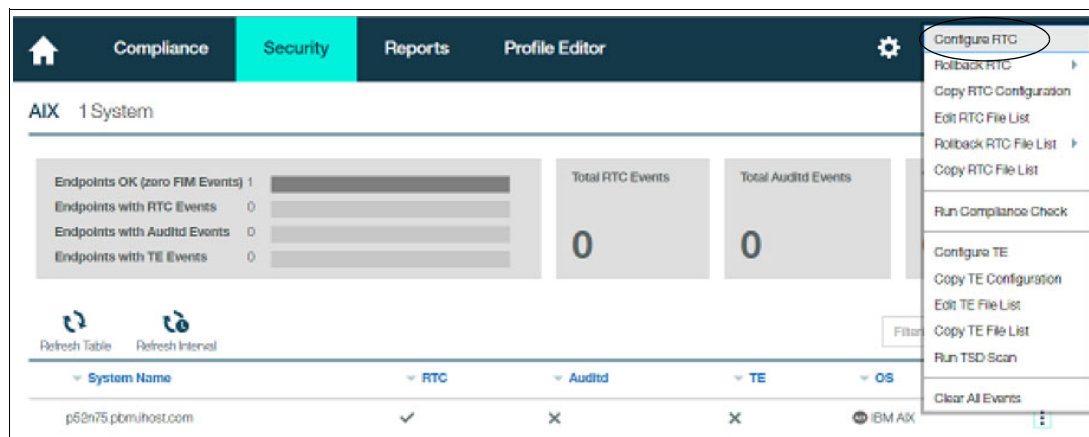


Figure 4-10 Security configuration pane

Figure 4-11 shows all of the fields that are available to edit.

**RTC Policies Configuration**

**complianceCheck:** ☒ on ☐ off

Specifies whether the rtd should call pscxpert to do compliancy checks. If complianceCheck is set to on, then pscxpert will be called from the rtd. If complianceCheck is set to off, then pscxpert will never be called. The default setting is on.

**debug:** ☐ off ☒ on

Specifies whether to turn on debug messages on. The valid values are on and off. The default value is off.

**email:**

Specifies the comma-separated list of email addresses to which the alerts will be sent.

**emailSubject:**



 Save  Cancel

Figure 4-11 RTC policies configuration pane

The PowerSC GUI server automatically creates a backup of the `rtd.conf` file whenever it is modified. This backup provides rollback options if you want to return to any of the previous configuration settings. The PowerSC GUI server also provides a Copy RTC configuration option that can be used if you want to copy the same RTC configuration to other LPARs.

### **rtd\_policy.conf**

The `rtd_policy.conf` file is responsible for storing the list of files and directories to be monitored and specifying the type of monitoring. After installing RTC, this file includes all of the predefined configurations, which can be altered by the administrator as needed.

The contents of `/etc/security/rtc/rtd_policy.conf` follows the same stanza that is shown in Figure 4-12 on page 122.

```

/etc/security/login.cfg:
    eventtype = modFile

/etc/security/audit/config:
    eventtype = modFile

/etc/security/audit/events:
    eventtype = modFile

/opt/IBMinvscout/bin/invscoutClient_PartitionID:
    eventtype = modFileAttr

/opt/IBMinvscout/bin/invscoutClient_VPD_Survey:
    eventtype = modFileAttr

/sbin/helpers/jfs2/backbyinode:
    eventtype = modFileAttr

/sbin/helpers/jfs2/diskusg:
    eventtype = modFileAttr

```

Figure 4-12 Sample content from `rtcd_policy.conf` file

The available options to define the type of monitoring are shown in Table 4-3.

Table 4-3 Options to specify type of monitoring

Attribute	Type of monitoring
modFile	Monitors file content changes.
modFileAttr	Monitors file permission or ownership changes.
modDir	This attribute is for directories. It monitors whether a new file or directory is created or deleted inside the monitored directory.

If you want to add or remove any file from the monitoring, you can do so directly by editing the `/etc/security/rtd/rtcd_policy.conf` file, by using the **chsec** command, or by using the PowerSC GUI.

### Adding new files to monitoring using chsec

To monitor the `/tmp/myfile` for file attribute changes use the following command:

```
chsec -f /etc/security/rtd/rtcd_policy.conf -s /tmp/myfile -a \
eventtype=modFileAttr
```

### Changing the monitoring of a file using chsec

To change the `/tmp/myfile` monitoring for file attribute changes and content changes, use the following command:

```
chsec -f /etc/security/rtd/rtcd_policy.conf -s /tmp/myfile -a \
eventtype=modFileAttr,modFile
```



## Removing files from monitoring using chsec

To remove /tmp/myfile from the monitoring list, use the following command:

```
chsec -f /etc/security/rtc/rtcd_policy.conf -s /tmp/abc -a eventtype=
```

You can also use the PowerSC GUI server to manage the `rtcd_policy.conf` file. Go to PowerSC GUI server Security page, select the LPAR, and click **Edit RTC File List**, as shown in Figure 4-13.

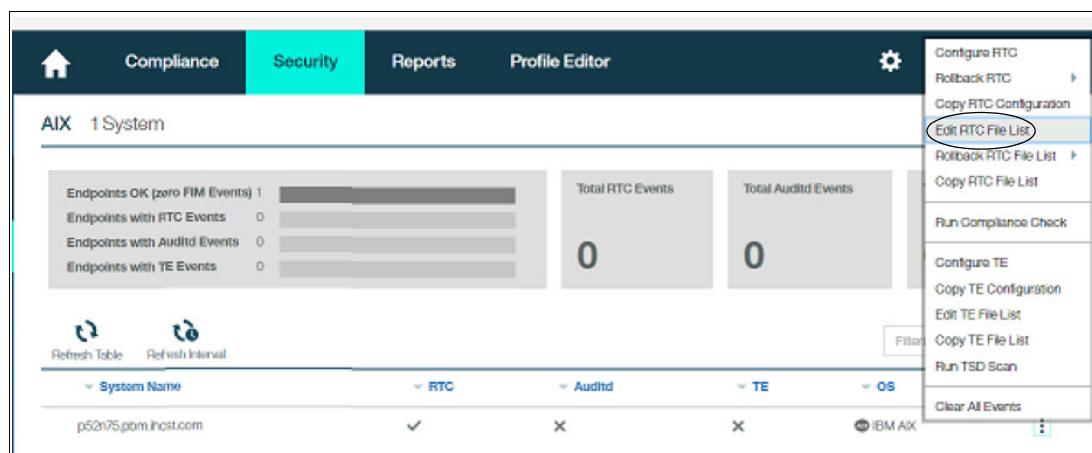


Figure 4-13 Managing `rtcd_policy.conf` file from the PowerSC GUI

The action lists all the files present in the AIX system. The list shows two options for each file: Content and Attribute.

If an option is selected, the file is being monitored for that attribute. You can clear the attribute if you want to remove this file from monitoring.

Similarly, you can select the option to add any file for RTC monitoring (see Figure 4-14).

Name	Content	Attributes
<input checked="" type="checkbox"/> ..	<input type="checkbox"/>	<input type="checkbox"/>
.	<input type="checkbox"/>	<input type="checkbox"/>
bincmds	<input type="checkbox"/>	<input type="checkbox"/>
config	<input checked="" type="checkbox"/>	<input type="checkbox"/>
events	<input checked="" type="checkbox"/>	<input type="checkbox"/>
objects	<input type="checkbox"/>	<input type="checkbox"/>
streamcmds	<input type="checkbox"/>	<input type="checkbox"/>

Figure 4-14 RTC File List Configuration pane

**Note:** For monitoring a directory, you must browse the directory and find a filename as dot (.), which indicates current directory. You must select the option for dot (.) to enable monitoring the directory.

### 4.1.6 Adding a new file to RTC file monitoring list

This section shows how to add a new file (/confidential.txt) to the RTC file monitoring list by using the PowerSC GU. The following process is used:

1. Go to PowerSC GUI - Security page.
2. Select the LPAR.
3. Click **Edit RTC File List**.

4. Browse to the `/confidential.txt` file and select the **Content** and **Attribute** options (see Figure 4-15).

**RTC File List Configuration**

Directory Path  
/

Name	Content	Attributes
.	<input type="checkbox"/>	<input type="checkbox"/>
.kshrc	<input type="checkbox"/>	<input type="checkbox"/>
.profile	<input type="checkbox"/>	<input type="checkbox"/>
.sh_history	<input type="checkbox"/>	<input type="checkbox"/>
.vi_history	<input type="checkbox"/>	<input type="checkbox"/>
.Xdefaults	<input type="checkbox"/>	<input type="checkbox"/>
confidential.txt	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
pcmsrv.lock	<input type="checkbox"/>	<input type="checkbox"/>
smit.log	<input type="checkbox"/>	<input type="checkbox"/>

Save Cancel

Figure 4-15 Adding the `confidential.txt` file to the monitoring list

This step automatically adds an entry of the `/confidential.txt` file in the `/etc/security/rtc/rtcd_policy.txt` file on the AIX LPAR, as shown in Figure 4-16.

```
# cat /etc/security/rtc/rtcd_policy.conf

/confidential.txt:
    eventtype = modFile,modFileAttr
```

Figure 4-16 Displaying content of `rtcd_policy.conf` file

You now modify the content of the file as follows:

```
# echo "my text" >> /confidential.txt
```

An RTC alert is generated in the PowerSC GUI server, as shown in Figure 4-17 on page 126.

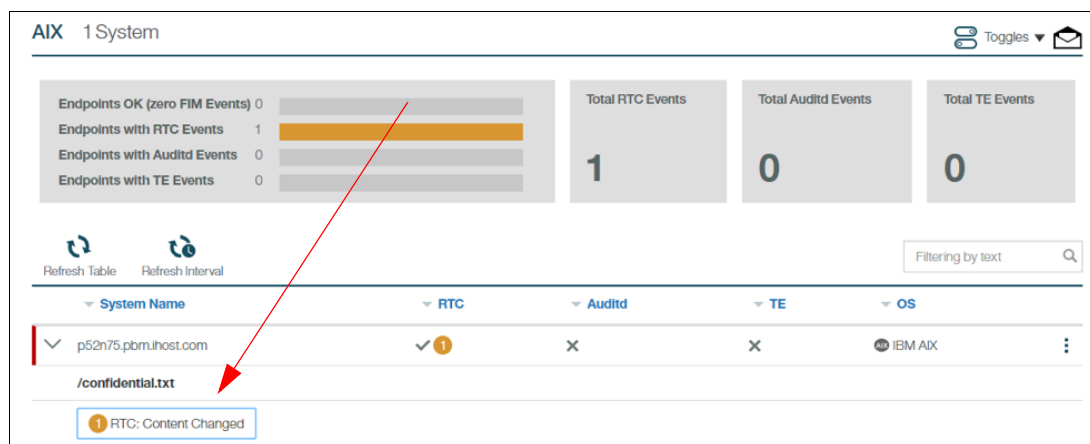


Figure 4-17 Alert generated

If you click the alert message, it displays more information, as shown in Figure 4-18.

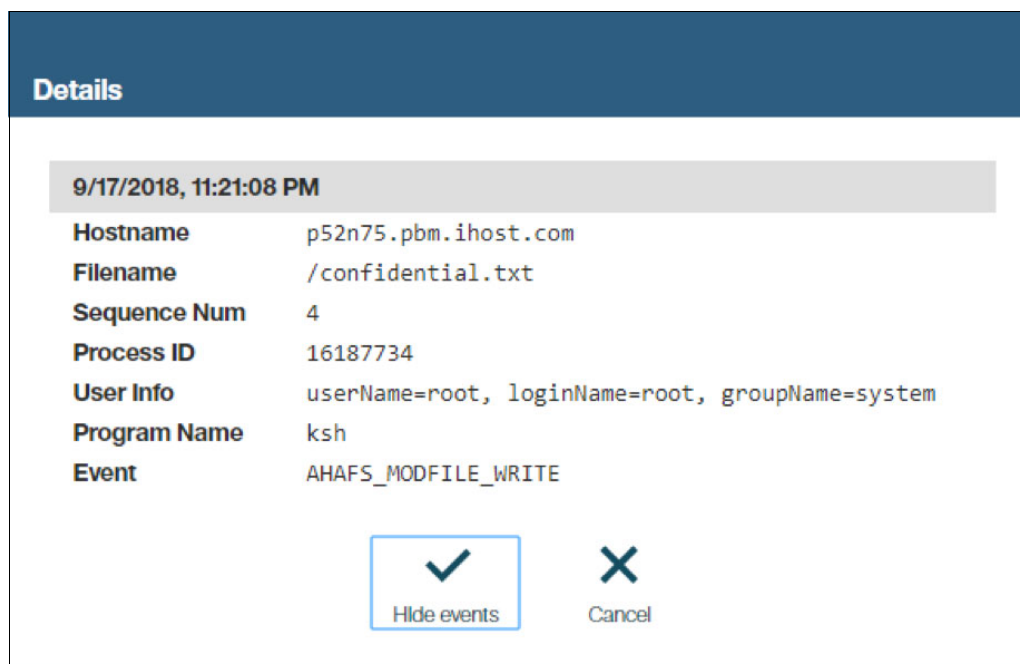


Figure 4-18 Details of the alert generated

Next, modify the permission of the file as follows:

```
# chmod 777 /confidential.txt
```

A new alert is generated in the PowerSC GUI server, as shown in Figure 4-19.

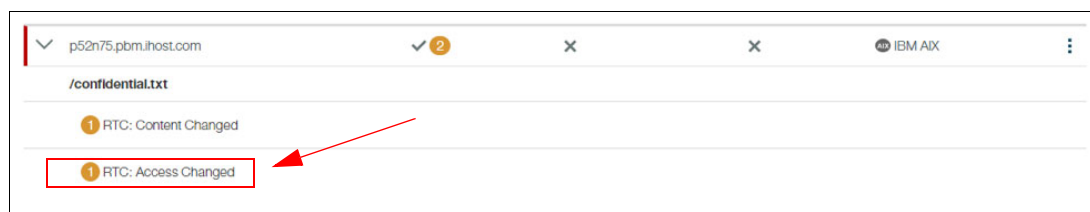


Figure 4-19 Another alert generated after the access was changed

Figure 4-20 shows the details of the alert about the access modification on the file.

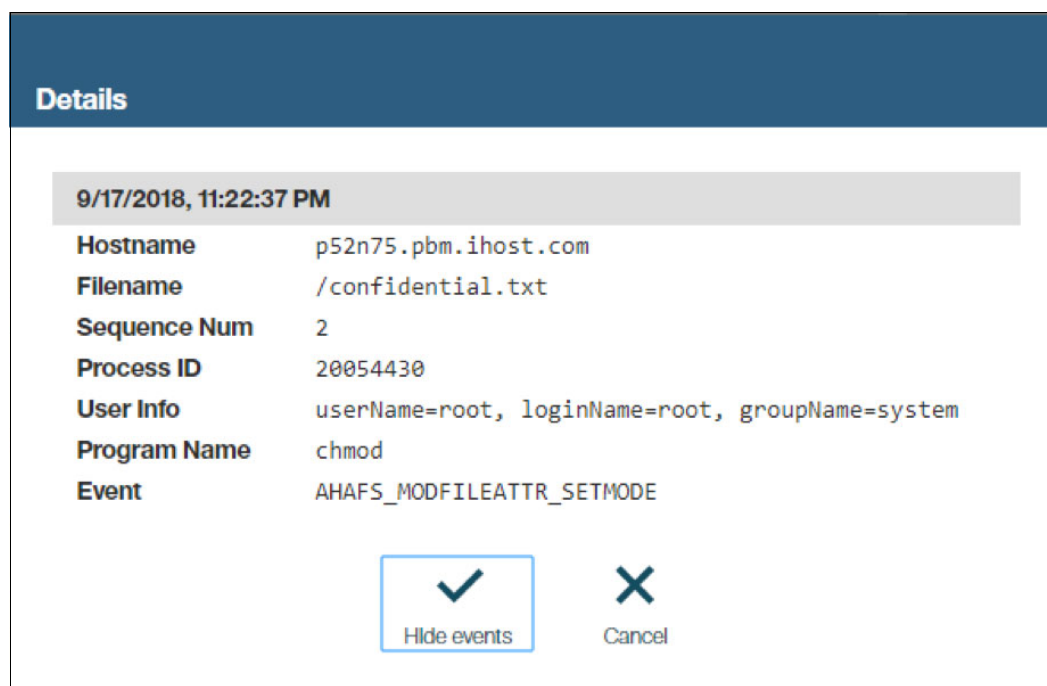


Figure 4-20 Alert details

### 4.1.7 Local logging

You can configure PowerSC RTC to log alerts to a local file in the LPAR. This configuration is done by defining the path of the local file in `/etc/security/rtc/rtcd.conf` file. How to output RTC alerts to the `/var/log/rtc.log` file is shown in the following example:

```
vi /etc/security/rtc/rtcd.conf
...
locallogfile: /var/log/rtc.log
...
```

### 4.1.8 SNMP traps

Simple Network Management Protocol (SNMP) is an Internet Protocol that is used to manage devices over the network. An SNMP trap is the capability that a network device has of sending messages to a network management server. For example, an AIX machine running RTC with the SNMP trap function enabled can send an SNMP message to another machine or an SNMP server when one of its monitored files is modified.

To enable the SNMP function, edit the file `/etc/security/rtc/rtcd.conf` directly by modifying some parameters and performing a stop/start on the `rtcd` daemon as shown in Example 4-4 by using parameters from Table 4-2 on page 119.

*Example 4-4 Enabling SNMP traps, user input in bold*

```
# vi /etc/security/rtc/rtcd.conf
# Specifies the parameters for snmptrap notification. Refer to
# the snmptrap documentation on setting the parameters. Set
# snmptrap_enable to yes to enable snmptrap alert, set to no to
# disable.
```

```
# Note: The snmpd daemon on the host specified by snmptrap_host
# must be running
snmptrap_enable:yes
snmptrap_host:rtcsnmpserver
snmptrap_community:rtccommunity
#snmptrap_oid:my0id
# stopsrc -s rtd
0513-044 The rtd Subsystem was requested to stop.
# startsrc -s rtd
0513-059 The rtd Subsystem has been started. Subsystem PID is
15335608.
```

---

After enabling the SNMP trap capability as shown in Example 4-4 on page 127, the rtd daemon starts sending SNMP traps for each detected violation to the SNMP server by using the `snmptrap` command.

### 4.1.9 RTC debug mode

When set to active, the debug mode on RTC starts sending the rtd debug logs to syslog along with alert information for the monitored files. For this reason, it is recommended to use the debug mode for debug purposes only to avoid filling up your syslog with duplicate information.

**Prerequisite:** This configuration requires the syslog to be configured and working.

## 4.2 AIX Trusted Execution

Trusted Execution (TE) is a feature that was introduced in AIX version 6.1. This AIX built-in security feature does not require installing any other software.

This feature helps to monitor the system for integrity violations and enforce runtime policies to disable execution of binaries, shell scripts, libraries, and kernel extensions if they are tampered with. It can help you prevent against malware attacks by allowing only white listed commands to run.

TE refers to a collection of features that are used to verify the integrity of the system's trusted computing base, which in the context of TE is called *Trusted Signature Database* (TSD). In addition, TE implements advance security policies, which together can be used to enhance the trust level of the complete system.

The usual way for a malicious user to harm the system is to access the system and then install trojan horses, rootkits, or tamper with some security critical files such that the system becomes vulnerable and exploitable.

The central idea behind the set of features under TE is to prevent such activities or in a worst-case scenario, identify if any such thing occurs to the system. Using the functionality provided by TE, the system administrator can decide on the actual set of executables that are allowed to run or the set of kernel extensions that are allowed to be loaded.

TE can also be used to audit the security state of the system and identify files that changed, which increases the trusted level of the system and makes it more difficult for the malicious user to harm the system. The set of features under TE can be grouped into the following categories:

- ▶ Managing the TSD
- ▶ Auditing the integrity of the TSD (system integrity check)
- ▶ Configuring security policies (runtime integrity check)
- ▶ TE Path, Trusted Library Path, Trusted Shell, and Secure Attention Key

PowerSC GUI can be used to centrally manage the TE feature of multiple AIX endpoints. For this feature, the PowerSC GUI agent must be configured on the AIX endpoint. For more information about setting up the PowerSC GUI, see 2.2, “Installing IBM PowerSC GUI server” on page 21.

For more information s about the AIX TE feature, see [IBM Knowledge Center](#).

## 4.2.1 Components of Trusted Execution

TE features the following main components:

- ▶ Trusted Signature Database

TSD is a database that is used to store critical security parameters of trusted files that are on the system. This database is in `/etc/security/tsd/tsd.dat` and includes any AIX media. In TE's context, *trusted files* are files that are critical from the security perspective of a system and if compromised can jeopardize the security of the entire system.

Every trusted file has an associated stanza or a file definition that is stored in the TSD. A file can be marked as trusted by adding its definition in the TSD by using the **trustchk** command. This command can be used to add, delete, or list entries from the TSD. If wanted, the TSD can be locked so even root cannot write to it any longer. Locking the TSD becomes immediately effective. To unlock the TSD, a system restart is required.

Example 4-5 shows a TSD stanza.

*Example 4-5 Shows stanza in TSD*

---

```
/usr/bin/ls:
    owner = bin
    group = bin
    mode = 555
    type = FILE
    hardlinks =
    symlinks =
    size = 29101
    cert_tag = 49424d4149583a31324331342d33314332303a324b3a41
    signature =
9b2e3d74ea578c78df6ab41ce5d23046a795aa4b468c20610bfd752505db9112e81cfe8678ab57b493
6ad550c050be5804ea09f21ab306119186e0eeaa8d74e1fdf1d91427f16ed4e36906c06dfc0c354619
af27ae3730fab60fd41b
6b1a0f133277955d52658da05d44d25166268374dc2c9828fbefeb105f64cfd1204920c790d45e313
633ed18761bb43fabcf9da515e62ca24197fd9502aead2ff6a9a7188ac0ae96d004b98de578c26b80d
bebaa11503ded9b15c4fe3109ac3977e89ddb3d2
6de5b47d496c2e9566ee8c6fb324bca839619144569d3b70642959ccafb9d2786bc1eedf6ebfeefd9c
33badfbee7c53e0c8919761023f01a9f90b820ec8d
    hash_value =
833dd2e0a3f429441e5b2b1534bfbcbcd941e0adda2c365b50cb8148a430ef33
    minslabel =
```

```

maxlabel =
intlabel =
accessauths = aix.fs.object.list
innateprivs = PV_DAC_R,PV_DAC_X
inheritprivs =
authprivs =
secflags = FSF_EPS
t_innateprivs = PV_MAC_R,PV_MIC

```

Table 4-4 lists the attributes of the Trusted Signature Database.

*Table 4-4 Trusted Signature Database attributes*

Attribute	Description
owner	Owner of the file. This value is computed by the <b>trustchk</b> command when the file is added to the TSD
group	Owner of the file. This value is computed by the <b>trustchk</b> command when the file is added to the TSD
mode	Comma-separated list of values. This value is computed by the <b>trustchk</b> command. The permissible values are SUID, SGID, SVTX, and TCB. The file permissions must be the last value and can be specified as an octal value. For example, for a file that is set uid and has permission bits as rwxr-xr-x, the value for the mode is SUID,755
type	Type of the file. This value is computed by the <b>trustchk</b> command. The possible values are FILE, DIRECTORY, MPX_DEV, CHAR_DEV, BLK_DEV, and FIFO.
hardlinks	List of hardlinks to the file. Because this value cannot be computed by the <b>trustchk</b> command, it must be supplied by the user at the same time when a file is added to the database.
symlinks	List of symlinks to the file. Because this value cannot be computed by the <b>trustchk</b> command, it must be supplied by the user when a file is added to the database.
size	Defines the size of the file. This value is computed by the <b>trustchk</b> command. A value of VOLATILE means that the file is changed frequently.
cert_tag	This value is computed by the <b>trustchk</b> command when the file is added to the TSD. The field maps the digital signature of the file with the associated certificate that can be used to verify the file's signature. (At the time of this writing, the certificate's ID is also its file name in /etc/security/certificates, but this might change in future releases.)
signature	The digital signature of the file. VOLATILE means that the file is changed frequently. This field is computed by the <b>trustchk</b> command.
hash_value	Cryptographic hash of the file. This value is computed by the <b>trustchk</b> command. VOLATILE means that the file is changed frequently.
minlabel	Defines the minimum Sensitivity Label for the object (when running Trusted AIX).
maxlabel	Defines the maximum Sensitivity Label for the object (when running Trusted AIX). This attribute is not applicable to regular files and FIFO.
intlabel	Defines the integrity label for the object (when running Trusted AIX).
accessauths	Defines the access authorization on the object (used in RBAC).
innateprivs	Defines the innate privileges for the file (used in RBAC).
inheritprivs	Defines the inherit privileges for the file (used in RBAC).
authprivs	Defines the privileges that are assigned to the user if they are authorized (used in RBAC).
secflags	Defines the file security flags that are associated with the object (used in RBAC). The FSF_TLIB flag also is available. It marks the object as part of the Trusted Library.



t_accessauth	Defines the extra Trusted AIX-specific access authorizations.
t_innateprivs	Defines the extra Trusted AIX-specific innate privileges for the file.
t_authprivs	Defines the extra Trusted AIX-specific privileges that are assigned to the user if they are authorized.
t_secflags	Defines the extra Trusted AIX-specific file security flags that are associated with the object.

► Certificates

TE uses certificates to verify the integrity of a file. The location of certificate is /etc/security/certificates. The default entries in the TSD database are signed by IBM private key and certificates are provided to verify the integrity. To add new files to TSD, administrators can provide their own private key/certificate pair. OpenSSL can be used to generate the keys and sign the file-related hashes.

► **trustchk** command

The **trustchk** command is the main command that can be used to manage various functions of TE. The command provides the following functions:

- Verifies system integrity
- Sets runtime policies
- Manages the TSD database

## 4.2.2 Trusted Execution modes

TE can be used in the following modes for performing integrity checks:

► System (offline mode)

In the offline mode, you can run the **trustchk** command to verify the integrity of trusted files. In this mode, the administrator can identify if any of the trusted files are tampered. Figure 4-21 shows the TE in system mode.

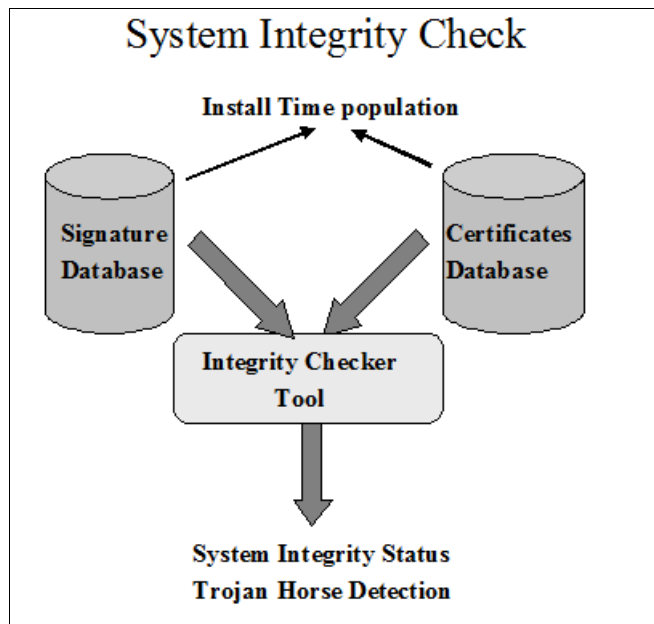


Figure 4-21 Trusted execution in system mode

Example 4-6 shows how **trustchk** can be used for system integrity check.

*Example 4-6 Shows commands for system integrity check*

---

To verify integrity of a particular file:

**trustchk -n <filename>**

To verify integrity of all files:

**trustchk -n ALL**

To correct integrity of file:

**trustchk -y <filename>** (without prompt)

**trustchk -t <filename>** (with prompt)

To query the TSD definition of a file:

**trustchk -q <filename>**

---

► Run-time (online mode)

In the online mode, TE can protect the execution of tampered files. When a file is marked as trusted (by adding its definition to the TSD), the TE feature can be used to monitor its integrity on every access. TE can continuously monitor the system and hence can detect tampering of any trusted file (by a malicious user or application) that is on the system at run time (that is, at load time). If the file is tampered with, TE can take corrective actions based on pre-configured policies. If a file is being opened or run and has an entry in the TSD, TE behaves as described next.

Before loading the binary, the component responsible for loading the file (system loader) starts the TE subsystem, which calculates the hash value by using the SHA-256 algorithm. This runtime calculated hash value is matched with the value that is stored in the TSD. The binary can be opened and run only if the values match (see Figure 4-22).

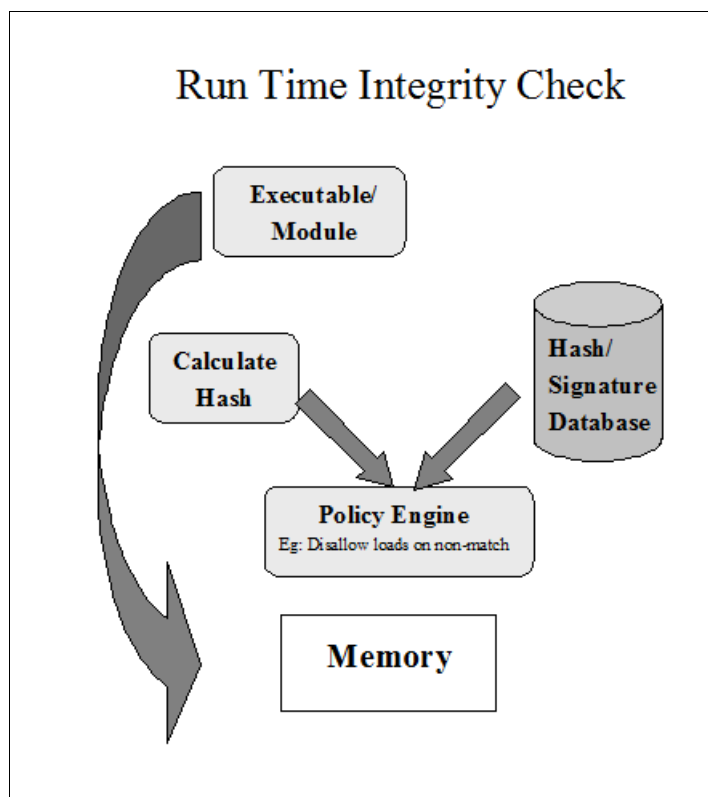


Figure 4-22 Shows Trusted Execution in runtime mode

The **trustchk** command can be used to display or configure runtime policies. Table 4-5 lists various runtime policies.

Table 4-5 Shows Trusted Execution runtime policies

Policy	Action
CHKEXEC	Checks the integrity of trusted executables before loading them in memory for execution.
CHKSHLIB	Checks the integrity of trusted shared libraries before loading them in memory for execution.
CHKSCRIPT	Checks the integrity of trusted shell scripts before loading them in memory.
CHKKERNEXT	Checks the integrity of kernel extensions before loading it in memory.
STOP_UNTRUSTD	Stops loading of files that are not trusted; that is, only files belonging to the TSD are loaded. This policy works with any of the CHK* policies. For example, if CHKEXEC=ON and STOP_UNTRUSTD=ON, any executable binary that does not belong to the TSD is blocked from execution.
STOP_ON_CHKFAIL	Stops loading of trusted files that fail the integrity check. This policy also works in combination with CHK* policies. For example, if CHKSHLIB=ON and STOP_ON_CHKFAIL=ON, any shared library that does not belong to the TSD is blocked from being loaded into memory for use.

TSD_LOCK	Lock the TSD so that it is not available for modification by using the <b>trustchk</b> command. Enabling this policy immediately locks the TSD; disabling it requires a system restart (as any change to an active policy).
TSD_FILES_LOCK	Lock trusted files. No change to any TSD files is allowed.
TEP	Sets the value of the Trusted Execution Path, and enables or disables it. The TEP consists of a list of colon-separated absolute paths, SUCH AS /usr/bin:/usr/sbin. When this policy is enabled, the files that belong to only these directory paths can be run. Any executable program that requests to be loaded that does not belong to the TEP is blocked.

You can run the **trustchk -p** command to display the current TE policies, as shown in Figure 4-23.

```
# trustchk -p
TE=OFF
CHKEXEC=OFF
CHKSHLIB=OFF
CHKSCRIPT=OFF
CHKKERNEXT=OFF
STOP_UNTRUSTD=OFF
STOP_ON_CHKFAIL=OFF
LOCK_KERN_POLICIES=OFF
TSD_FILES_LOCK=OFF
TSD_LOCK=OFF
TEP=OFF
TLP=OFF
```

Figure 4-23 The **trustchk -p** command

To change a Trusted Execution policy, use the following command:

```
trustchk -p <attribute=ON|OFF>
```

For example: **trustchk -p CHKEXEC=ON**.

### 4.2.3 Trusted Execution integration with PowerSC GUI

The PowerSC GUI server can centrally manage the AIX TE feature. For this management, you must configure the PowerSC GUI agent on the AIX LPAR.

The PowerSC GUI Security page displays the TE status for the LPAR, as shown in Figure 4-24.

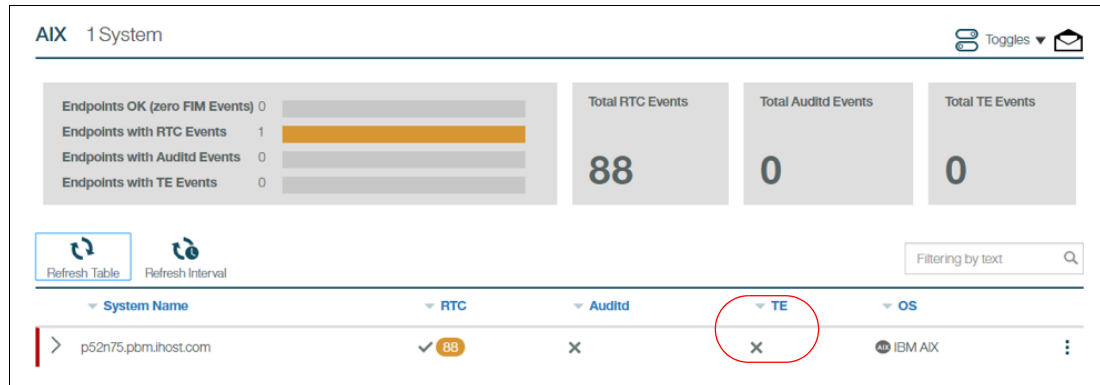


Figure 4-24 PowerSC GUI Security window

Figure 4-24 shows an X where TE is not enabled, a tick (✓) sign where TE enabled, and a dash (-) where it cannot detect TE status.

You can select the LPAR and go to Configure TE option to enable or disable TE runtime policies, as shown in Figure 4-25.

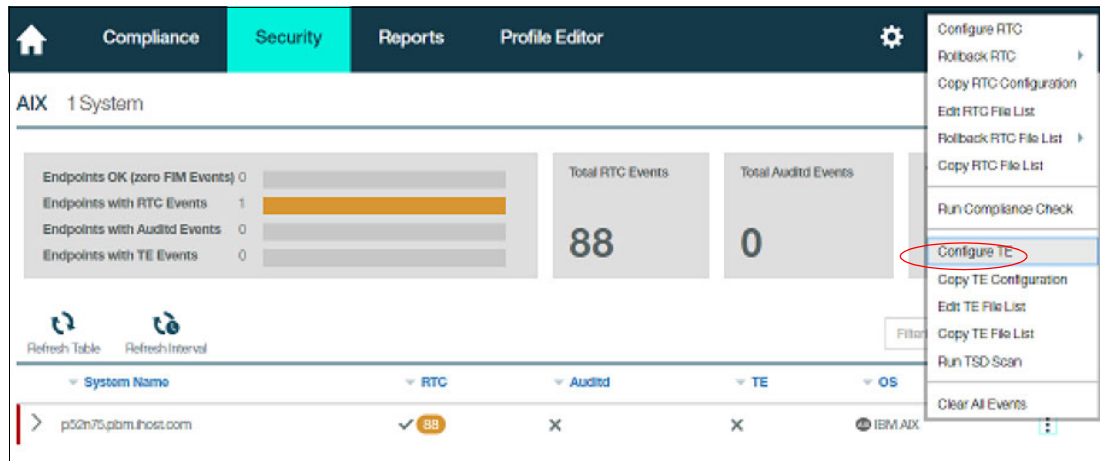
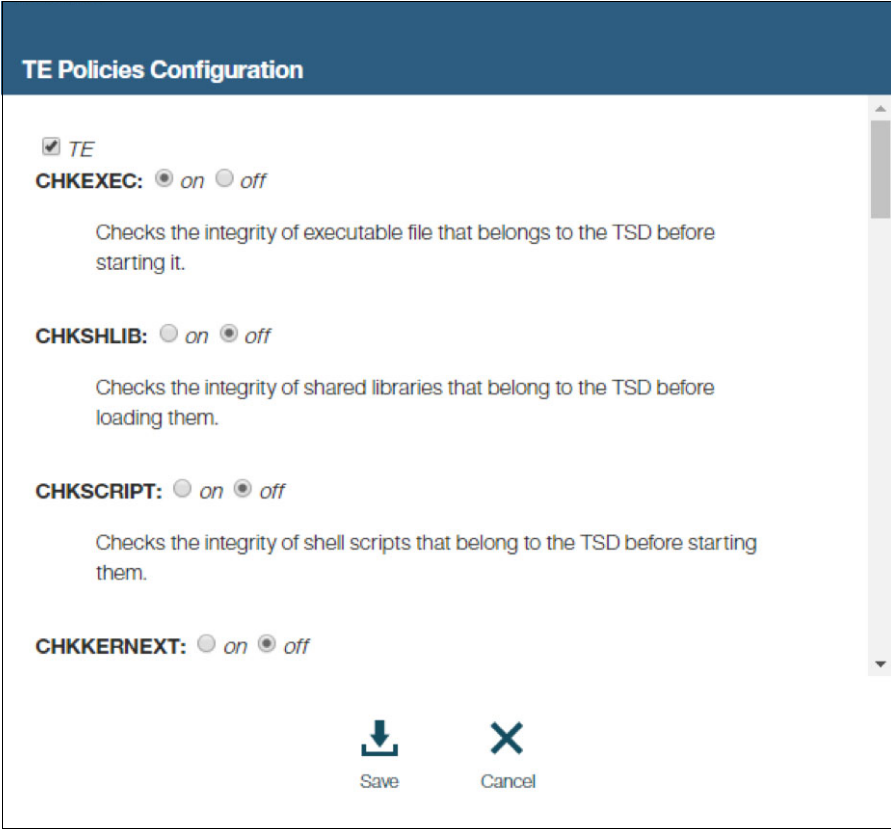


Figure 4-25 Configuring the TE options

Figure 4-26 shows various TE policies that can be enabled or disabled.



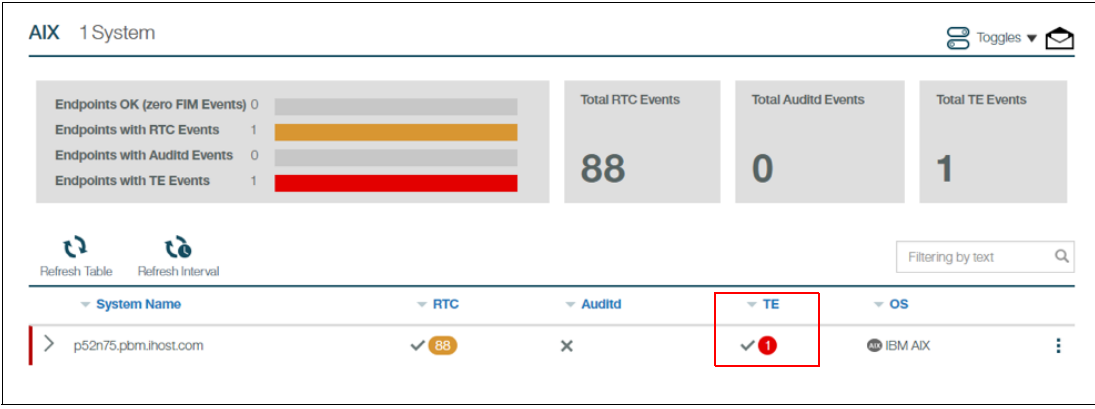
The image shows a configuration pane titled "TE Policies Configuration". It contains four policy settings, each with a description and radio buttons for "on" and "off".

- TE** (checked):
  - CHKEEXEC:** ☒ on ☐ off. Checks the integrity of executable file that belongs to the TSD before starting it.
  - CHKSHLIB:** ☐ on ☒ off. Checks the integrity of shared libraries that belong to the TSD before loading them.
  - CHKSCRIPT:** ☐ on ☒ off. Checks the integrity of shell scripts that belong to the TSD before starting them.
  - CHKKERNEXT:** ☐ on ☒ off.

At the bottom, there are "Save" and "Cancel" buttons.

Figure 4-26 TE Policies Configuration pane

After you enable TE, the PowerSC GUI indicates that TE is enabled, as shown in Figure 4-27.



The image shows the PowerSC GUI for an AIX system. It displays a summary of events and a table of system details. The "TE" column in the table is highlighted with a red box, showing a checkmark and a red circle with the number 1, indicating that TE is enabled.

System Name	RTC	Auditd	TE	OS
p52n75.pbrn.ihost.com	✓ 88	✗	✓ 1	IBM AIX

Figure 4-27 PowerSC GUI showing TE enabled

PowerSC GUI shows a TE alert, which indicates that TE is switched on. If you click the alert, more information displayed, as shown in Figure 4-28.

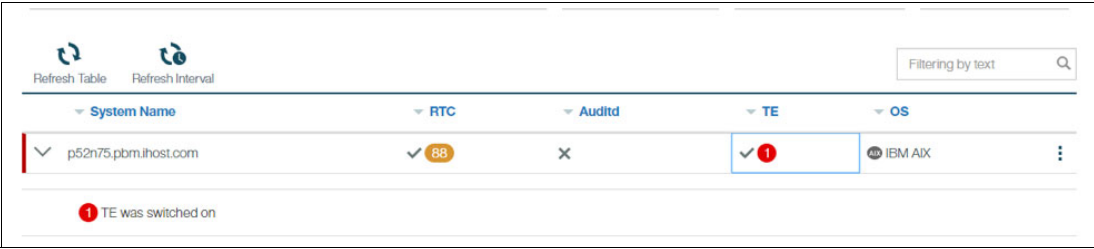


Figure 4-28 PowerSC GUI with TE enabled

If you run the **trustchk** command on the LPAR, you can see that the policies you changed from the PowerSC GUI server are enabled on the LPAR, as shown in Example 4-7.

Example 4-7 Show TE policies enabled on the LPAR

```
# trustchk -p
TE=ON
CHKEXEC=ON
CHKSHLIB=ON
CHKSCRIPT=OFF
CHKKERNEXT=OFF
STOP_UNTRUSTD=OFF
STOP_ON_CHKFAIL=OFF
LOCK_KERN_POLICIES=OFF
TSD_FILES_LOCK=OFF
TSD_LOCK=OFF
TEP=OFF
TLP=OFF
```

4.2.4 System integrity check with PowerSC GUI

You can perform TE system integrity check (offline) from the PowerSC GUI, as shown in Figure 4-29. At the PowerSC GUI Security page, select the LPAR and then, click **Run TSD Scan**.

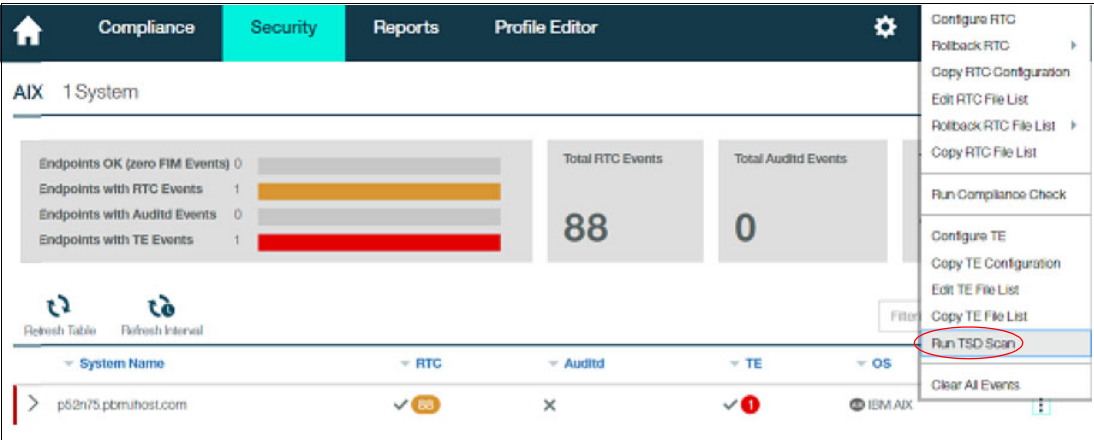


Figure 4-29 Offline system integrity check with PowerSC GUI

The offline integrity check starts, as shown in Figure 4-30.

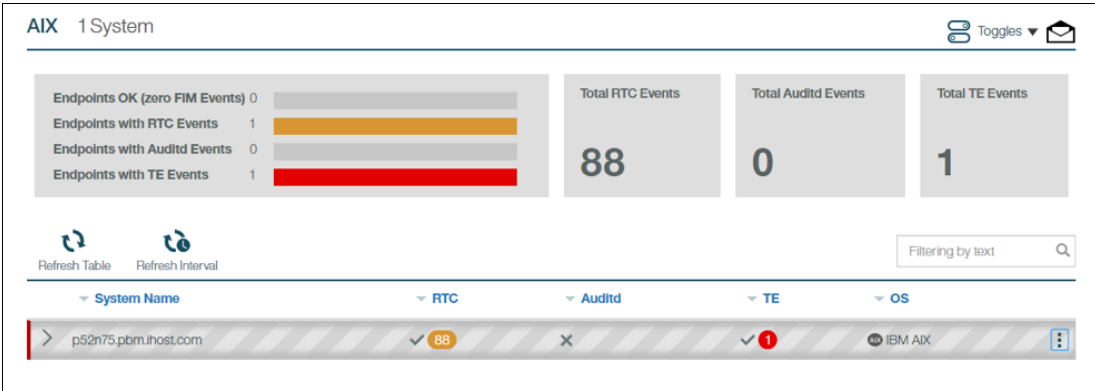


Figure 4-30 Offline integrity check

After the scan is complete, you can view the result by clicking the **TE: Concern** alert, as shown in Figure 4-31.

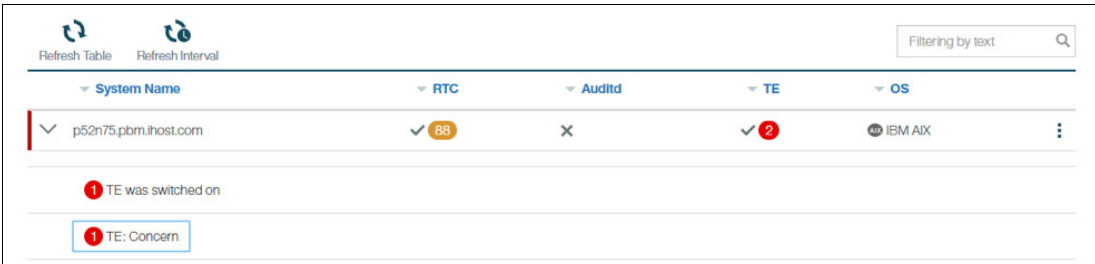


Figure 4-31 Shows results of the scan



Figure 4-32 shows information from the TE Concern alert.

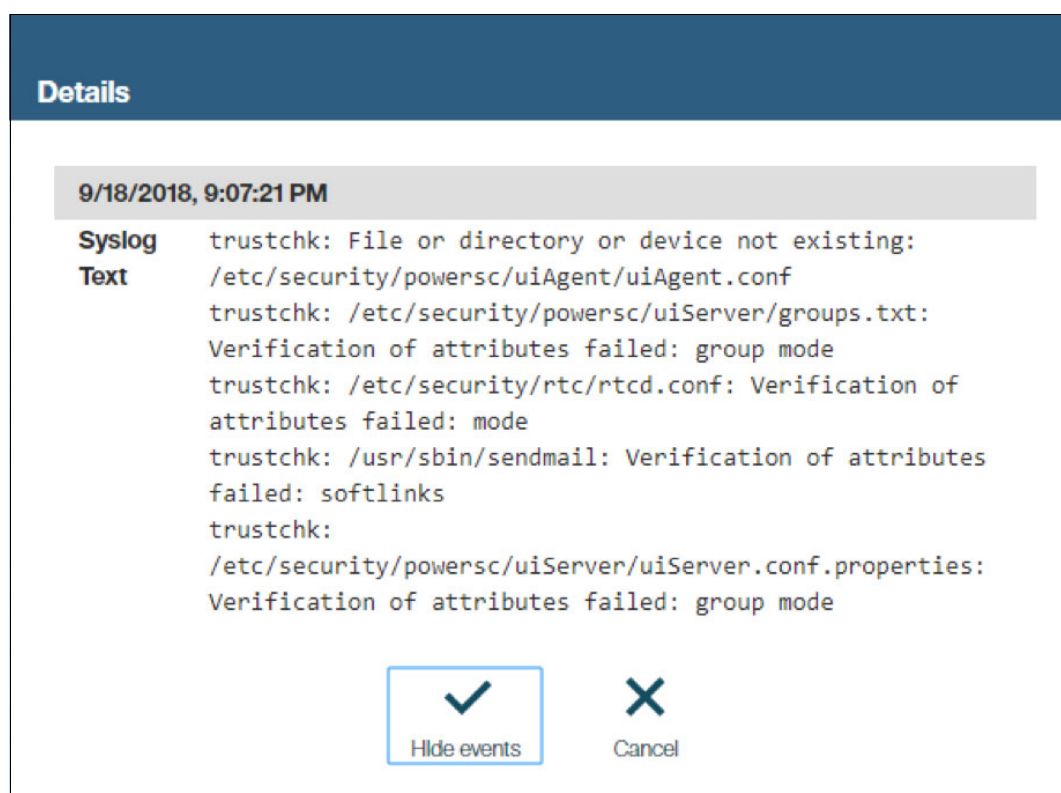


Figure 4-32 Details from the TE Concern alert

The same result can be viewed by running the **trustchk -n ALL** command on the LPAR, as shown in Figure 4-33.

```
# trustchk -n ALL
trustchk: File or directory or device not existing: /etc/security/powersc/uiAgent/uiAgent.conf
trustchk: Verification of attributes failed: /etc/security/powersc/uiServer/groups.txt
: group mode
trustchk: Verification of attributes failed: /etc/security/rtc/rtcd.conf
: mode
trustchk: Verification of attributes failed: /usr/sbin/sendmail
: softlinks
trustchk: Verification of attributes failed: /etc/security/powersc/uiServer/uiServer.conf.properties
: group mode
```

Figure 4-33 Showing TE system integrity check result

As you can see, PowerSC GUI makes it easier to manage TE on multiple endpoints through a single console.

## 4.2.5 Online Check with PowerSC GUI

The PowerSC GUI can be used to manage runtime TE policies of multiple endpoints. In this section, we describe how to verify the integrity of the trusted executables and block the execution if an integrity mismatch occurs.

For this scenario, enable the following TE policies:

```
TE=ON
CHKEXEC=ON
STOP_ON_CHKFAIL=ON
```

Log in to PowerSC GUI Security page and select the LPAR. Click **Configure TE**. Then, enable three policies, as shown in Figure 4-34.

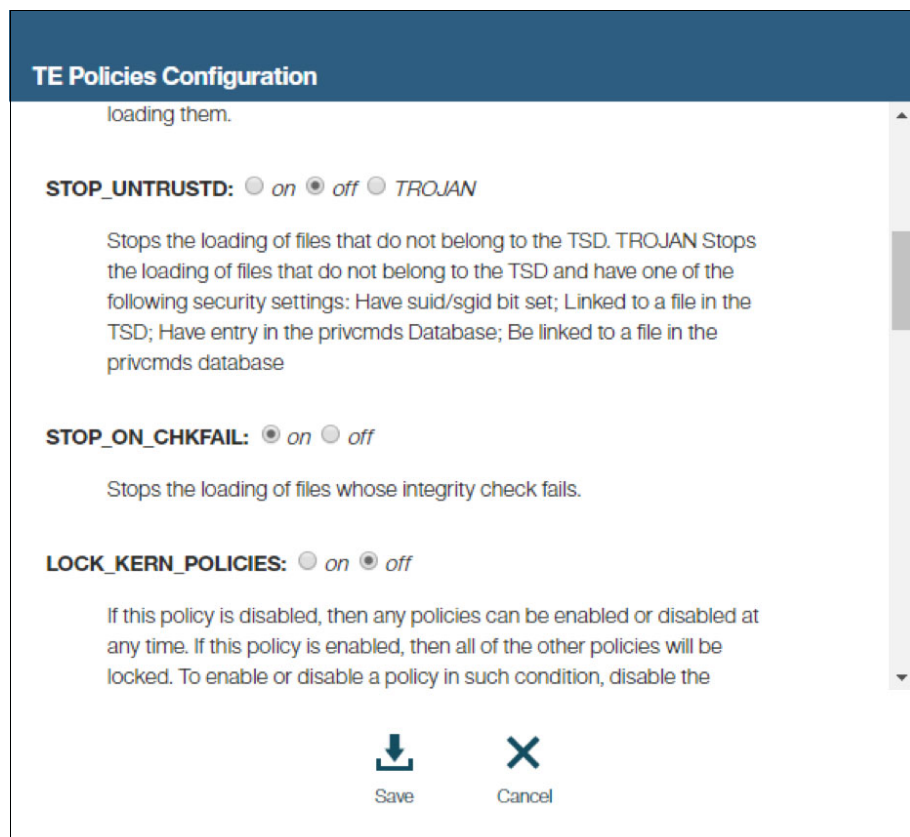


Figure 4-34 Enabling policies in the TE Policies Configuration

On the LPAR, three policies are enabled, as shown in Example 4-8.

Example 4-8 Displaying TE policies

```
# trustchk -p
TE=ON
CHKEXEC=ON
CHKSHLIB=OFF
CHKSCRIPT=OFF
CHKKERNEXT=OFF
STOP_UNTRUSTD=OFF
STOP_ON_CHKFAIL=ON
LOCK_KERN_POLICIES=OFF
TSD_FILES_LOCK=OFF
TSD_LOCK=OFF
TEP=OFF
TLP=OFF
```

**Note:** To get TE runtime alerts, you must check that syslog is enabled in the LPAR. If syslog is disabled, you must enable it and restart the PowerSC GUI agent.

TE blocks the execution of a command if an integrity mismatch occurs in a trusted file; for example, *chfs*.

The `/usr/bin/chfs` command is a trusted file. Without any tampering, you can run the command as shown in Figure 4-35.

```
# ls -l /usr/sbin/chfs
-r-xr-xr-x    5 root      system      108009 Mar 08 2017  /usr/sbin/chfs
#
# chfs -a size=+1M /tmp
Filesystem size changed to 2621440
```

Figure 4-35 Running the command without any tampering

After modifying the permission of the `chfs` file, attempt run it again. This time, TE detects the modification and blocks the execution, as shown in Example 4-9.

Example 4-9 File permission changed, and TE blocks execution

```
# chmod 777 n/usr/sbin/chfs
#
# chfs -a size=+1M /tmp
ksh: chfs: 0403-006 Execute permission denied.
```

The TE alert can be seen in the PowerSC GUI server, as shown in Figure 4-36.



Figure 4-36 PowerSC GUI TE alert

The details of the alert can be viewed by clicking the alerts, as shown in Figure 4-36. The details of the alert are shown in Figure 4-37 and Figure 4-38 on page 142.

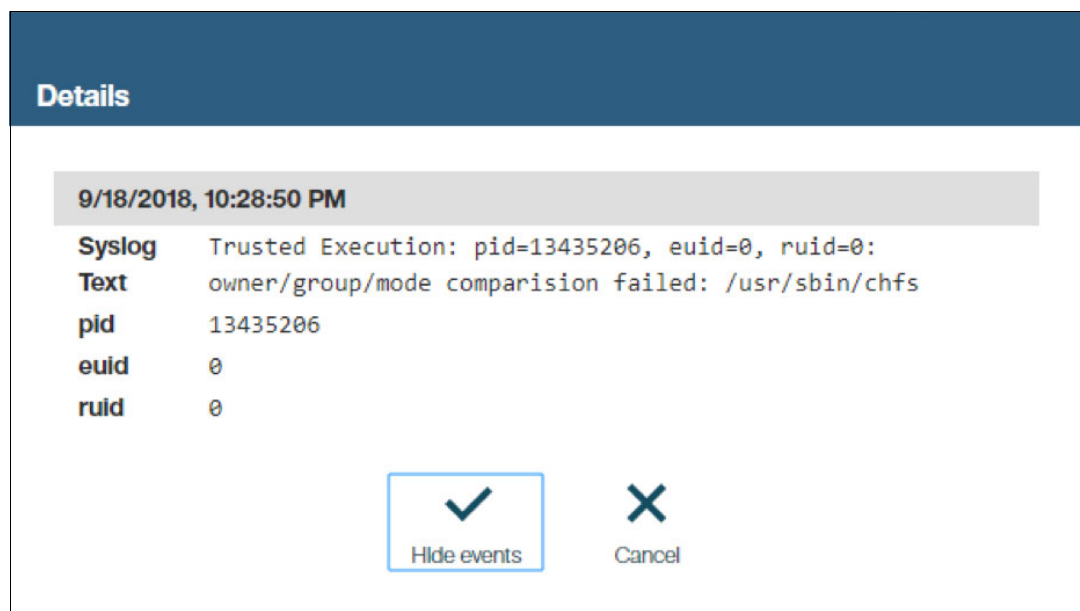


Figure 4-37 Trusted Execution file changed report

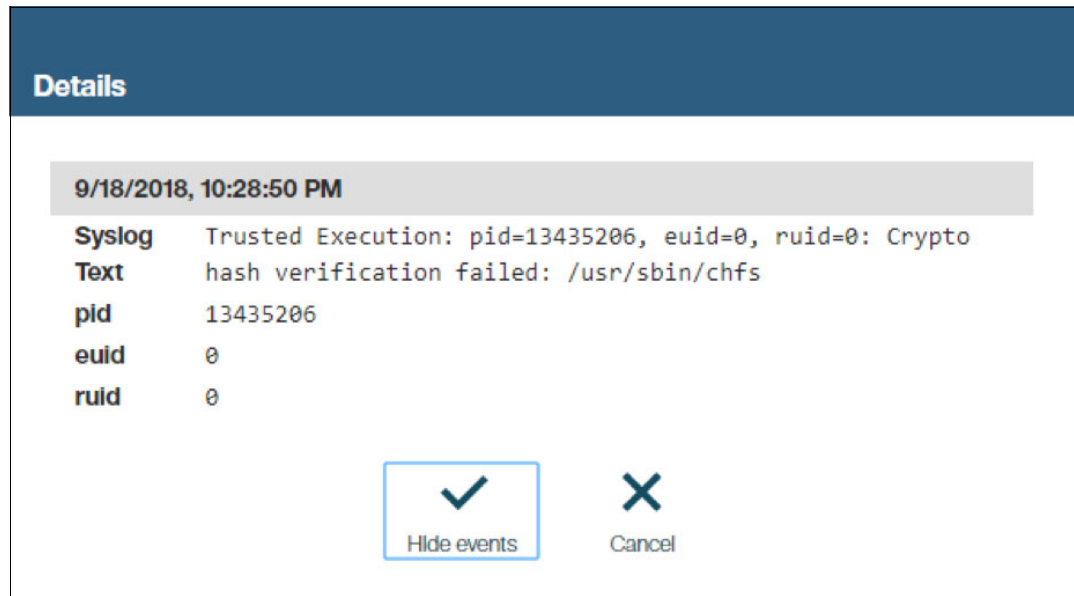


Figure 4-38 Trusted Execution command blocked report

You can fix the problem by running the **trustchk -t <filename>** command.

## 4.2.6 TSD customization with PowerSC GUI

To add a new file to TSD, the **trustchk** and **openssl** commands are used. OpenSSL is required to create the private key and the certificate pair to sign the hash value of the file.

PowerSC GUI simplifies this process because the key management is done by the PowerSC GUI server. If you add a new file to TSD by using the PowerSC GUI, you are not required to run **openssl** commands to create keys. This process is internally managed by the PowerSC GUI server.

To add a new file to TSD, go to PowerSC GUI Security page and select the LPAR. Click the **Edit TE File List** option, as shown in Figure 4-39.

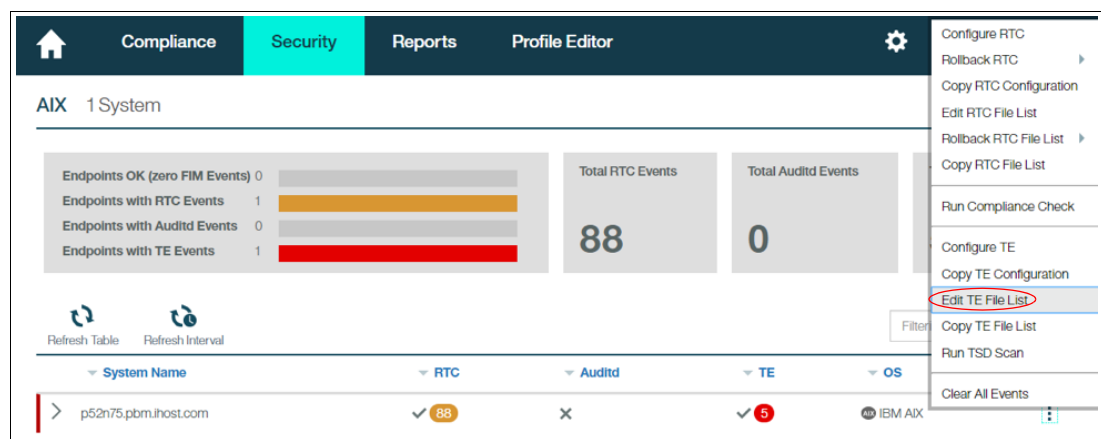


Figure 4-39 Using the PowerSC GUI to add a new file to TSD

Browse to select the file. In this case, add the `/home/config.ksh` file to TSD, as shown in Figure 4-40 on page 143.

TE File List Configuration

Directory Path

/home

Name	TE	Volatile
..		
lost+found		
psadmin		
pslogin		
srvproxy		
test1		
config.ksh	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Save

Cancel

Figure 4-40 TE File List Configuration pane

Click **Save** to add the file to TSD.

You can verify the new entry by running the **trustchk -q** command on the LPAR, as shown in Figure 4-41.

```
# trustchk -q /home/config.ksh
/home/config.ksh:
    type = FILE
    owner = root
    group = system
    mode = 644
    size = 0
    hash_value = e3b0c44298fc1c149afb4c8996fb92427ae41e4649f
    cert_tag = 008935178761efb5cc
    signature = 67c81af27086bde377fe523cf07288274ed3d6176de2!
7980783bdefb649801b726a172791c94dfdb5f9cd3af7e0f3be193202b1c56b0:
683e27d48f389253a4c0bd44049181729643a54d11c290b642e830d3728fe350:
bd7846bc95cf8edf1a3ddb1ec9ac1303774acd63f9d0d80369add5d8b5
```

Figure 4-41 Running the **trustchk** command to query TSD for the **/home/config.ksh** file

**Note:** At the time of this writing, the PowerSC GUI does not automatically load the new information from TSD into the kernel. You must restart TE for this load to take effect. This process can be done by running the **trustchk -p TE=ON** command on the endpoint.

## 4.2.7 Best practice to enable TE in online mode

The TE runtime check is one of the best ways to protect your system against malware attacks. You can enforce strong runtime policies to allow only **whitelisted** commands to run.

However, if runtime policies are enforced without proper planning, legitimate commands might be blocked by TE, which can affect the usability and availability of the system. For example, if you did not add the application or middleware commands to TSD and enable the runtime policy to block untrusted commands, the application commands do not run.

To prevent such a scenario, you can initially enable TE in log only mode and review your system for few days. After you are satisfied with the configuration by reviewing the log, you can enable TE in enforcement mode.

Complete the following steps to enable TE without causing disruption to the usability of the system:

1. Enable TE in log only mode. In this mode, TE logs only the integrity mismatch errors, and does not block the execution. For this process, you must enable the following TE flags:

```
TE=ON
CHKEEXEC=ON
CHKSHLIB=ON
CHKSCRIPT=ON
CHKKERNEXT=ON
STOP_UNTRUSTD=OFF
STOP_ON_CHKFAIL=OFF
```

2. Review TE logs directly on the LPAR or by using the PowerSC GUI.
3. After reviewing the alerts, identify the commands that are legitimate and can be added to TE database.
4. Add the commands to TE database. You can use command line option or use PowerSC GUI.
5. Keep monitoring to ensure no new TE alerts occur on the system. After you confirm that no new alerts are present, enable TE in enforcement mode. In this mode, TE blocks the execution. For this process you must enable the following TE flags:

```
TE=ON
CHKEEXEC=ON
CHKSHLIB=ON
CHKSCRIPT=ON
CHKKERNEXT=ON
STOP_UNTRUSTD=ON
STOP_ON_CHKFAIL=ON
```

6. Continue monitoring the alerts.

**Note:** As with any other security feature, we strongly recommend enabling TE in a test environment first. After you test the settings thoroughly, enable TE in the production environment.

## 4.2.8 Updating an application that is integrated with TE

If you want to update an application that is integrated with TE, you must update the TSD database after the application update is completed. This process is required; otherwise, the TSD entries for the application files do not match with the files on the system.

Complete the following steps:

1. When TE is enabled, it does not allow you to install or update anything. Therefore, you must temporarily disable TE. This process can be done by using the command line or from the PowerSC GUI:
  - From command line run the following command:  

```
trustchk -p TE=OFF
```
  - From the PowerSC GUI, go to the Toggles option, which is on PowerSC GUI Security page, as shown in Figure 4-42.

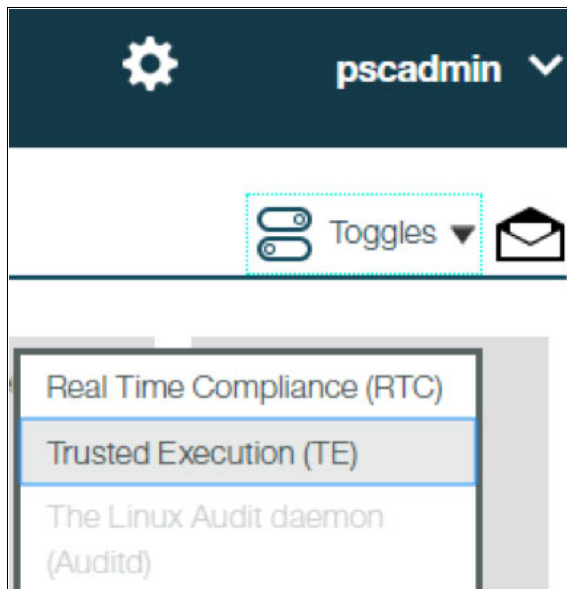


Figure 4-42 PowerSC GUI Security pane

2. Select the **Turn off** option to disable TE, as shown in Figure 4-43 on page 146.

### Trusted Execution (TE)

☐

Turn ON for all endpoints in the current group

☒

Turn OFF for all endpoints in the current group

Automatically turn back ON after:

☒

1 Hour

☐

5 Hours

☐

1 Day

☐

1 Week

☐

Never

Apply

Cancel

Figure 4-43 TE pane

**Tip:** The PowerSC GUI provides an option to automatically turn on TE after a specific period. You can choose this period from the available options. This feature is useful so that you do not forget to turn on TE after the installation is finished.

3. Update the application.
4. Update the TSD entries for the application files (which are changed) by completing the following steps:
  - a. Delete the previous entry.
  - b. Add the new entry.



5. Enable TE by using the command line or from the PowerSC GUI, as shown in Figure 4-44.

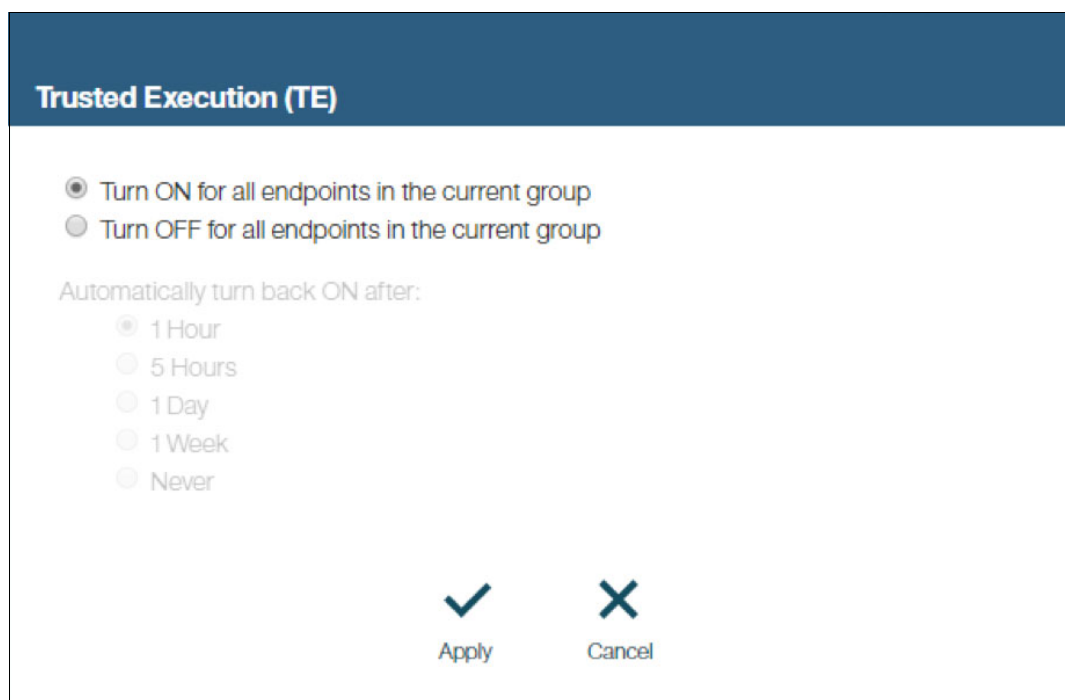


Figure 4-44 Enabling TE

## 4.3 Linux auditd

PowerSC 1.2 extends the FIM functionality to Linux endpoints as well. The following Linux distros are supported in PowerSC 1.2:

- ▶ Red Hat Enterprise Linux (RHEL)
- ▶ SUSE Linux Enterprise Server (SLES)

PowerSC uses the Linux auditd feature to track changes to monitored files.

### 4.3.1 Prerequisites

The following prerequisites must be met enable FIM on Linux endpoints:

- ▶ Enable auditing on Linux endpoints by starting the auditd subsystem by using the **systemctl** command.

To enable auditing from the command line, use the following command:

```
systemctl start auditd
```

To check the status of audit, use the following command:

```
systemctl status auditd
```

Figure 4-45 shows how you can start auditd on Linux endpoints.

```
# systemctl start auditd
#
# systemctl status auditd
â auditd.service - Security Auditing Service
   Loaded: loaded (/usr/lib/systemd/system/auditd.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-09-17 12:13:43 EDT; 2 s ago
     Process: 8691 ExecStartPost=/sbin/auditctl -R /etc/audit/audit.rules (code=exited, status=0/SUCCESS)
    Main PID: 8690 (auditd)
      Tasks: 2 (limit: 512)
     CGroup: /system.slice/auditd.service
            ââ8690 /sbin/auditd -n

Sep 17 12:13:43 p52n70 systemd[1]: Starting Security Auditing...
Sep 17 12:13:43 p52n70 auditd[8690]: Started dispatcher: /sbin...
Sep 17 12:13:43 p52n70 systemd[1]: Started Security Auditing ...
Sep 17 12:13:43 p52n70 auditctl[8691]: No rules
Sep 17 12:13:43 p52n70 auditctl[8691]: AUDIT_STATUS: enabled=...
Sep 17 12:13:43 p52n70 auditd[8690]: Init complete, auditd 2....
Hint: Some lines were ellipsized, use -l to show in full.
```

Figure 4-45 Running systemctl command to start auditd

**Note:** If you MUST disable auditd, use the following command:

```
systemctl stop auditd
```

#### 6. Install pscxpert and PowerSC GUI agent.

You MUST install PowerSC pscxpert and UIAgent filesets. For more information about The installing these filesets, see 2.4, “Installing the UIAgent” on page 40.

You can verify the installation by running the **rpm** command, as shown in Example 4-10.

Example 4-10 Verifying PowerSC filesets installation

```
# rpm - qa |grep - i powersc
powersc-pscxpert-1.2.0.0-sles12.ppc63le
powersc-uiAgent-1.2.0.0-sles12.ppc64le
#
```

## 4.3.2 Configuration

To configure FIM on Linux endpoint, you must configure the PowerSC GUI agent on the Linux LPAR. For more information, see 2.4, “Installing the UIAgent” on page 40.

The auditd appears enabled in the PowerSC GUI as indicated by a tick mark (see Figure 4-46).

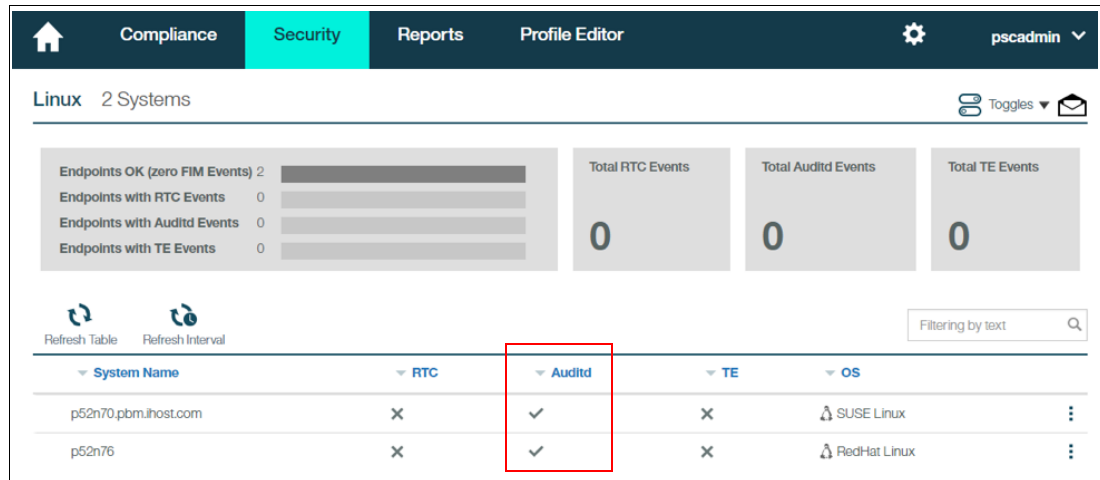


Figure 4-46 PowerSC GUI showing auditd enabled

## 4.3.3 Add file for monitoring

You can add any file for integrity monitoring by using the PowerSC GUI server. At the PowerSC GUI server security page, select the LPAR. Then, select **Edit Auditd File List**, as shown in Figure 4-47.

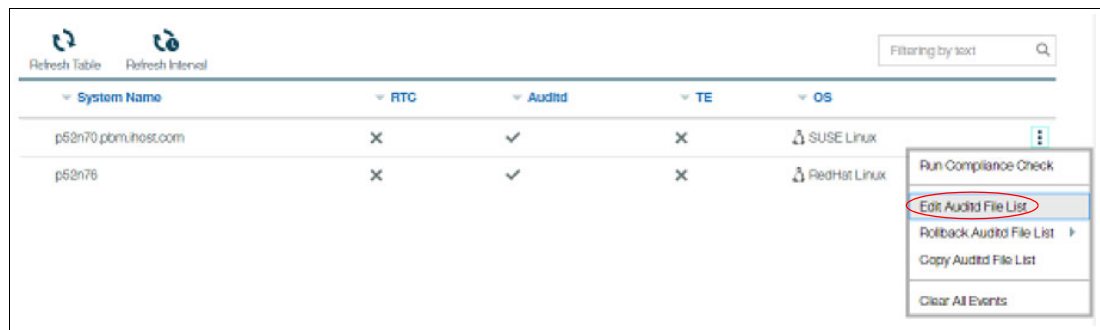


Figure 4-47 PowerSC GUI security page pane

Browse to the file that you want to add, as shown in Figure 4-48.

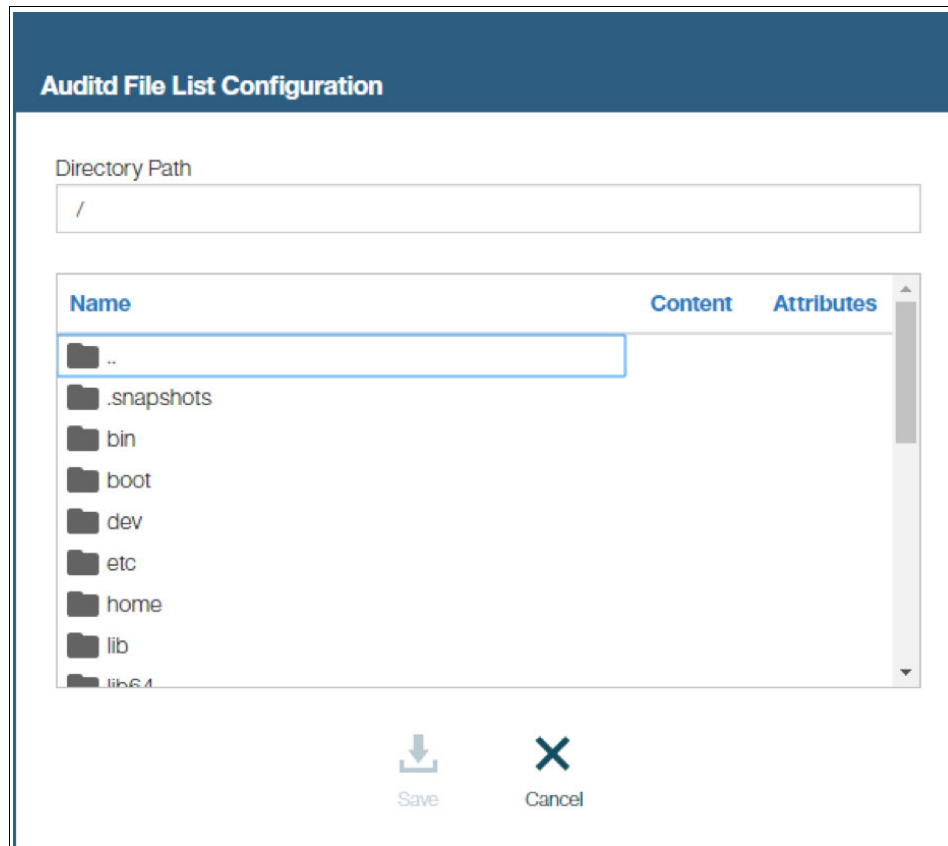


Figure 4-48 Auditd File List Configuration pane

In this example, we add the /etc/hosts file, as shown in Figure 4-49.

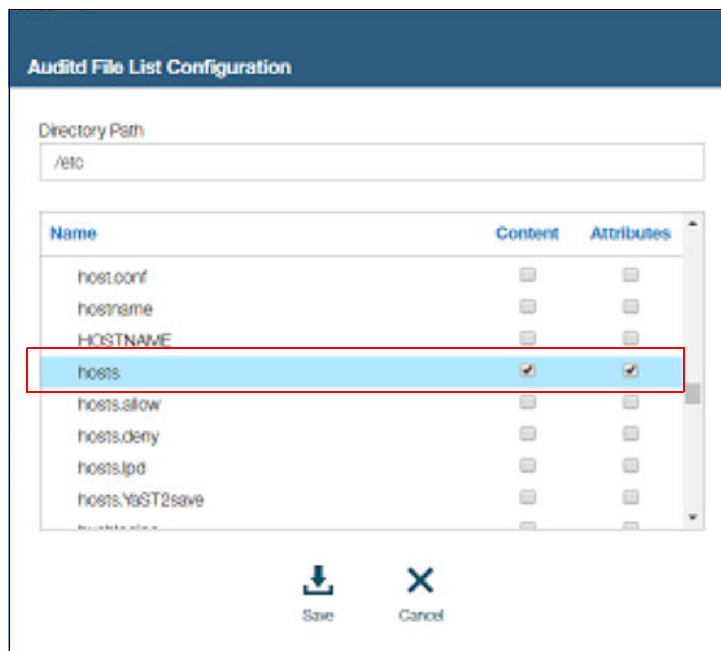


Figure 4-49 Auditd File List Configuration

In the LPAR, you can verify that the file is being monitored by auditd, as shown in Example 4-11.

Example 4-11 Verifying the file /etc/hosts is monitored by auditd

```
# auditctl -l
-w /etc/hosts -p w -k powersc-mod-file
-w /etc/hosts -p a -k powersc-mod-file-attr
```

### 4.3.4 View FIM alerts

PowerSC GUI server shows you FIM alerts if the monitored file is modified. Figure 4-50 displays how PowerSC GUI server shows alerts if the permission of monitored file is modified.

```
# ls -l /etc/hosts
-rw-r--r-- 1 root root 641 Sep 12 16:06 /etc/hosts
#
#
#
# chmod g+w /etc/hosts
#
# ls -l /etc/hosts
-rw-rw-r-- 1 root root 641 Sep 12 16:06 /etc/hosts
#
```

Figure 4-50 Modifying permission of /etc/hosts file

The file modification action generates an alert in the PowerSC GUI, as shown in Figure 4-51.

System Name	RTC	Auditd	TE	OS	
✓ p52n70.pbmihost.com	✗	✓ 1	✗	SUSE Linux	⋮
/etc/hosts					
1 Auditd: Attributes Changed					
p52n76	✗	✓	✗	Red-Hat Linux	⋮

Figure 4-51 PowerSC GUI file change alert

If you want more information about the alert, click the alert message and more information is presented, as shown in Figure 4-52.

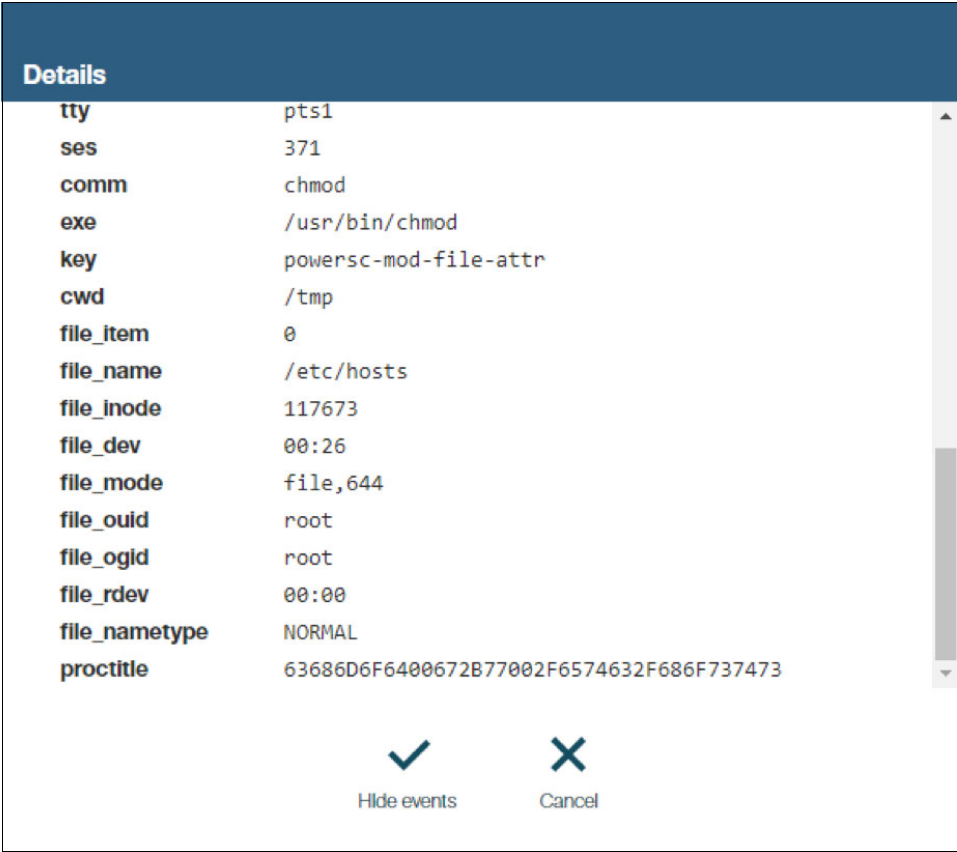


Figure 4-52 PowerSC GUI alert details

## 4.4 FIM reporting with PowerSC GUI

PowerSC GUI provides excellent reporting feature for viewing FIM events centrally.

### 4.4.1 Dashboard view of FIM events

PowerSC GUI provides a dashboard view of all the FIM-related events at one place. You can select the group for which you want to view the FIM alerts. The dashboard view is useful to get a high-level view of FIM events, as shown in Figure 4-53.



Figure 4-53 PowerSC GUI dashboard

### 4.4.2 Reporting of FIM events

PowerSC GUI server provides a reporting feature for FIM events. You can configure the following reporting options in PowerSC GUI server at the Reports page:

- **File Integrity Overview:** This option provides an overview of FIM events, as shown in Figure 4-54. You can configure email address to receive periodic emails.

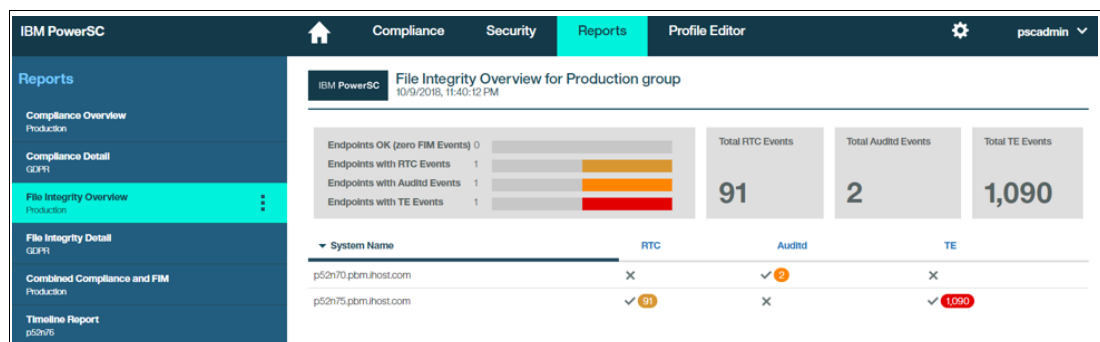


Figure 4-54 PowerSC GUI Reports pane

- **File Integrity Detail:** This option provides detail reporting of FIM events, as shown in Figure 4-55. You can configure email address to receive periodic emails.

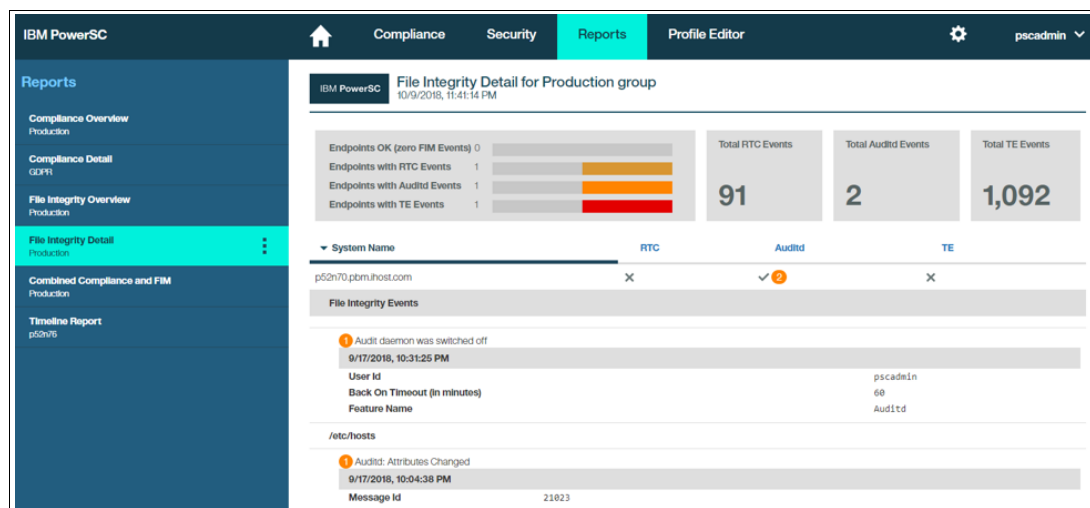


Figure 4-55 PowerSC GUI File Integrity Details view

- **Timeline Report:** The timeline report provides FIM alerts by month, day, or hour, as shown in Figure 4-56.

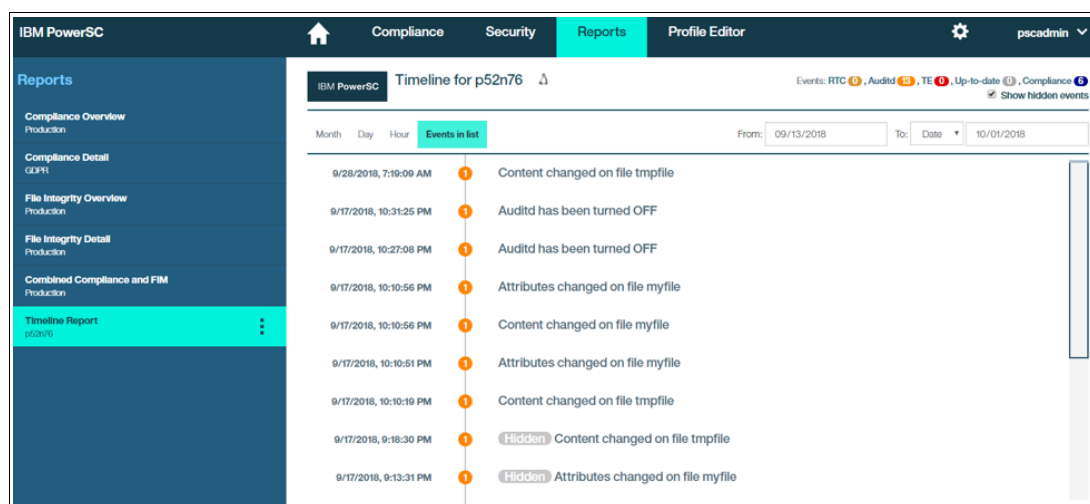


Figure 4-56 PowerSC GUI Timeline Report view

For more information about PowerSC reporting features, see 2.7.4, “Reports tab” on page 62.





# **PowerSC Trusted Network Connect and Patch Management v1.2.0.0**

This chapter describes PowerSC trusted network connect and patch management and contains the following topics:

- ▶ 5.1, “Introduction” on page 156
- ▶ 5.2, “Component architecture” on page 157
- ▶ 5.3, “Simplifying management of security and compliance by using TNC” on page 160
- ▶ 5.4, “Deployment considerations” on page 162
- ▶ 5.5, “Installing TNCPM” on page 164
- ▶ 5.6, “Working with Trusted Network Connect and Patch Management” on page 182
- ▶ 5.7, “Troubleshooting” on page 196

## 5.1 Introduction

Unfortunately, several recent headlines involved organizations experiencing high-profile security breaches that were primarily caused by attackers who were leveraging vulnerabilities on systems that were not properly patched. One of the most common tactics of cyber attackers is to identify software or hardware vulnerabilities for exploitation based on publicly known vulnerabilities.

Numerous security defenses are needed to properly secure any type of computer environment. Some security defenses, such as employee security education, are not even considered technical security controls.

One of the most fundamental and important cyber defenses is vulnerability management because when vulnerabilities are published, attackers are informed of these vulnerabilities. Attackers can use this new information to enhance their ability to leverage vulnerabilities against the organizations they target.

Vulnerability management provides on-going deployment and maintenance of up-to-date patching. Properly patching your systems can require significant time and resources. The following significant challenges must be overcome to achieve proper patching:

- Verify relevance

When an advisory is published, the details of the patch must be understood before deciding to patch. Nuances or details of the patch can exist that can indicate that the patch is not relevant. Therefore, it is important first fully understand the details of the advisory before deploying the patch.

- Determine priority and scheduling

If a patch is determined relevant to your environment, the risk of the security vulnerability and the business application effect to applying the update must be understood to define a proper plan for patching.

- Test updates

Good practice recommends testing any type of update in a test environment before applying the update to production. Therefore, it is important to allocate resources and define a plan for testing the update before applying the update to production.

Fortunately, PowerSC provides Trusted Network Connect (TNC) and Patch Management as a key solution to use when implementing your vulnerability management cyber defenses.

TNC and Patch Management can query AIX or VIOS hosts to determine whether they are properly patched. This process of querying a host to determine whether a host meets compliance requirements is referred to as *verification*. TNC provides automated and manual methods for specifying the criteria that determines whether a host is classified as compliant or non-compliant to the patch policy.

TNC and Patch Management can retrieve relevant AIX and VIOS updates from IBM Fix Central and other sites for open source packages in installation and rpm format (for more information, see this [web page](#)). This process allows TNC to automatically download updates and provide email messaging to notify you when new updates are downloaded and available for administrator-initiated deployment. Considering TNC is primarily a command line-based solution, it can facilitate security patching monitoring and deployment automation.

## 5.2 Component architecture

TNC is a set of network security specifications that are recommended by Trusted Computing Group (TCG). TCG is a consortium of multiple organizations, with IBM as one of its key promoters.

The goal of TCG is to develop, define, and promote open standards for trusted computing and security technologies. TCG specifications include hardware building blocks and software interfaces across multiple platforms, peripherals, and devices. These specifications enable secure computing environments with the primary goal of helping users to protect their informational assets from compromise because of software attacks.

A subgroup of TCG, TNC defined this solution architecture that can help the administrators to enforce policies to effectively control access to the network infrastructure. The end points that are requesting access are assessed based on the integrity measurements of critical components that can affect its operational environment. The integrity verification can be strengthened by using capabilities of Trusted Platform Module (TPM) and Trusted Software Stack (TSS). The TNC architecture is built on the well-established security standards, such as 802.1X, EAP, RADIUS, IPsec, and TLS/SSL.

Figure 5-1 shows the topology of TNC and Patch Management.

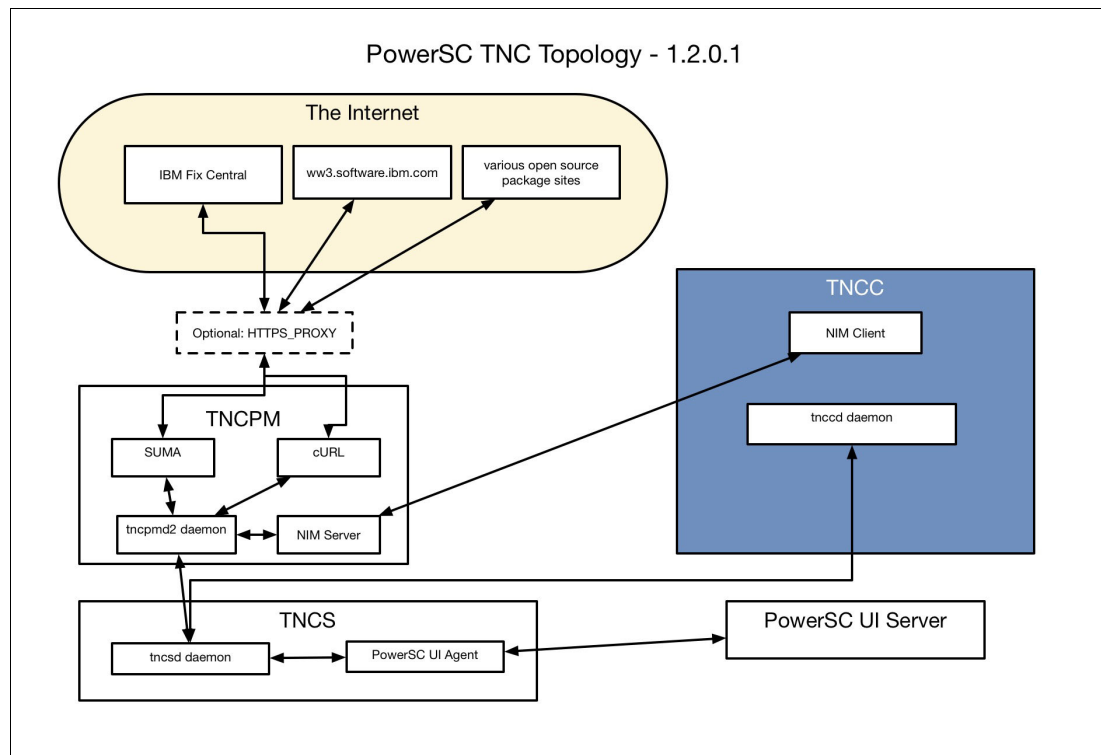


Figure 5-1 PowerSC TNC topology

Figure 5-1 also shows TNC and Patch Management serving as the patch repository. The TNC and Patch Management daemon, `tncpmd2`, uses the AIX Service Update Management Assistant (SUMA) and `cURL` to download AIX Technology Service Packs, interim fixes, and open source packages from the different sites shown in Figure 5-1.

This retrieval of software can be done by using a direct connection to the internet or an http or https proxy. The TNCS is the PowerSC component that issues verify and update operations against TNC Clients.

The verification of the TNCC occurs through a network connection that is established between the TNCS daemon (tncsd) and the TNCC daemon (tnccd). When the TNCS issues an update operation against a TNC Client, the operation is communicated to the TNC and Patch Management by using a network connection that is established between the TNCS daemon and the TNCPM daemon (tncpmd2). When the TNC and Patch Management daemon receives the update operation request from the TNCS, it instructs the Network Installation Manager (NIM) server on the local host to perform the TNC update by way of NIM.

The TNC model consists of the following components:

► **Trusted Network Connect Client**

The Trusted Network Connect Client (TNCC) is implemented by way of the daemon, tnccd. The TNCC runs on your standard AIX host. Its primary function is to respond to verify operations that are started from the TNCS.

When the TNCS issues a verify operation against the TNCC, the TNCC replies with an inventory of filesets and fixes that are applied on the local TNCC host. When the TNCS retrieves this inventory, it determines whether the TNCC is compliant, based on the TNC Patch Policy requirements.

For more information about how compliance is determined, see 5.6.1, “Verifying the Trusted Network Connect Client” on page 182.

**Important:** As of PowerSC 1.2.0.0, a Virtual I/O Server can now be configured as a TNCC.

► **Trusted Network Connect Server**

The Trusted Network Connect Server (TNCS) is implemented by way of the daemon tncsd. TNCS is the primary administrative component for your standard AIX and VIOS Trusted Network Connect Clients (TNCCs). The TNCS starts verification and update operations against TNCCs. These functions are the two most critical functions that the TNCS provides.

**Important:** Both of these operations can also be issued against a TNC ipgroup, which is a TNC-defined group of hosts.

The TNCS verify operation process consists of the following steps:

1. The TNCS requests the target TNCC to generate an inventory of filesets, ifixes, and open source packages that are installed on the local host.
2. The TNCC sends the inventory to the TNCS.
3. The TNCS analyzes the TNCC inventory to determine whether the TNCC is classified as COMPLIANT, depending on the TNC Patch Policy that corresponds to the TNCC.

The TNCS update operation process consists of the following steps:

1. The TNCS requests the Trusted Network Connect Patch Management server (TNCPM) to update a particular TNCC with a particular set of updates. For an AIX TNCC, the update can be an interim fix, APAR, service pack, or technology level, or open source package. For a VIOS TNCC, the update can be only an interim fix or an open source package group.

2. The TNCPM takes the TNCS request and submits it for execution by using the underlying NIM server that is installed on the local host of the TNCPM server.
3. A NIM operation performs the update against the targeted TNCC by using the NIM client on the local host of the TNCC.
4. At the conclusion of the NIM operation, the results are returned to the TNCS.
5. The TNCS reports the status of the update operation at the conclusion of the update operation. This result can be viewed from the command-line interface. The details of the results also are stored on the local TNCS.

### **Trusted Network Connect Patch Management server**

The Trusted Network Connect Patch Manager (TNCPM) server is implemented by way of daemon `tncpmd2`. The TNCPM server provides the following critical functions to TNC:

- ▶ Interface with SUMA and cURL to retrieve updates from IBM Electronic Customer Care (ECC), IBM Fix Central, and other open source-related sites.
- ▶ Implement a fix repository for all the types of updates that are retrieved from the internet by using SUMA and cURL.
- ▶ The TNCPM server requires that a NIM server is installed on the same host on which the TNCPM server is installed. The TNCPM server applies updates to TNCCs by way of NIM when instructed to by the TNCS.
- ▶ Interface with the TNCS to keep the TNCS in sync with the latest list of updates that were downloaded by the TNCPM.
- ▶ Provides a TNCPM Scheduler that provides the ability for the TNC administrator to specify how often the TNCPM contacts IBM Fix Central to see whether new or updated security updates exist. Whenever the TNCPM Scheduler checks IBM Fix Central, it notifies the TNCS of any changes that were made to the TNCPM fix repository. Also, the `psconf pull` operation can be used to receive the changes that are made to the TNCPM repository. For more information, see “Sync TNCS with TNCPM” on page 179.

### **Trusted Network Connect secure communication**

The TNC daemons communicate over the encrypted channels that are enabled by Transport Layer Security (TLS) or Secure Sockets Layer (SSL). The secure communication ensures that the data and commands that flow in the network are authenticated and secure. Each system must have its own key and certificate, which are generated when the initialization command for the components is run. This process is not apparent to the administrator and requires less involvement from the administrator.

### **Trusted Network Connect protocol**

The TNC protocol is used with the TNC framework to maintain network integrity. TNC provides specifications to verify the endpoint integrity. The endpoints are assessed based on the integrity measurements of critical components that can affect their operational environment.

The TNC framework enables administrators to monitor the integrity of the systems in the network. The TNC implementation on AIX is integrated with the AIX patch distribution infrastructure to build a complete patch management solution.

TNC specifications must satisfy the requirements of AIX and IBM POWER® family system architecture. The components of TNC provide a complete patch management solution on the AIX operating system.

This configuration enables administrators to efficiently manage the software configuration on AIX deployments. It also provides tools to verify the patch levels of the systems and generate a report on the clients that are not compliant. Also, patch management simplifies the process of downloading and installing the patches.

## 5.3 Simplifying management of security and compliance by using TNC

TNC simplifies the management of security and compliance through the following features:

- Point and click management support by way of the graphical user interface (GUI).

After the TNC infrastructure is fully and properly configured by your TNC administrator, the PowerSC GUI can be used to centrally verify and deploy patches by way of a simple point and click interface. The PowerSC GUI administrator can easily run the most essential tasks of verifying and updating patches without having to understand the internals of TNC configuration or AIX patching.

- Automation of update retrieval.

TNC automates the retrieval of all AIX and VIOS security updates to the TNCPM server. After the TNC administrator specifies the set of technology levels the TNCPM services, all new fixes and service packs that are published automatically are downloaded to the TNCPM.

- Notification of new security updates.

Administrators receive email notifications when new AIX service packs and interim fixes were downloaded to the TNCPM.

- Complete flexibility in defining patch policy.

Your AIX environment that is managed by TNC can contain many different types of AIX endpoints with different install and patch requirements. TNC allows you to specify multiple criteria that constitute compliance, so you can specify patch policy for any type of complexity your AIX environment contains.

This flexibility in patch policy is made possible by TNC's modular method for defining patch policy. The modular approach to defining TNC patch policy can use one or more of the following constructs:

- TNC IP groups
- TNC fileset policies
- TNC patch policies
- TNC ifix groups
- TNC ifix lists
- TNC apar groups
- TNC apar lists
- TNC open package groups

- Flexible reporting options.

TNC allows you to group subsets of your AIX partitions in what are called *TNC IP groups*. TNC IP groups allow you to set up email reporting based on an IP group. For example, this feature allows a production administrative group to receive only TNC notifications regarding production partitions because email reporting was configured to send notifications to the production email distribution list only for production-related notifications. Email reporting options allow you to apply more filtering by specifying that only errors or only success events must be reported.

- Simple reverification or update for new security updates.

When the TNCPM downloads a new security update, the *default policy* can be used to quickly verify your TNC clients including the new ifix. The new ifix also can be immediately deployed to one or more TNC clients.

For more information, see 5.6.1, “Verifying the Trusted Network Connect Client” on page 182.

- Correlation of patches with endpoint service pack level.

An AIX ifix typically corresponds to one specific AIX service pack level. Without a patch management solution, you must maintain the mappings of several different ifixes for the different service pack levels of your environment. TNC eliminated this complexity by maintaining this ifix mapping for you. For example, if you verify a TNC client and TNC reports an ifix is missing, you can be sure that the reported ifix corresponds to the specific service pack level of the TNC client that was verified.

- Patch recommendations that are made upon the filesets that are installed on the TNCC.

When a TNC Client is verified, the filesets that are installed on the system are inventoried. TNC uses this inventory to provide ifix installation recommendations that are based on the actual filesets that are installed on the specific endpoint. If ifixes correspond to a specific service pack level but are not needed because the corresponding filesets are not installed, TNC identifies this detail and report this distinction without identifying it as a missing ifix. Therefore, you see only compliance failures for vulnerabilities that correspond to the filesets that are installed on your endpoints.

- Extensive installation support, including open source packages in rpm and installp format.

PowerSC v1.1.6.0 introduced the ability to download open source packages. This new functionality allows you to define open package groups so you can use the standard verify and update operations with open source packages. With this new extension of open source packages, TNC provides a comprehensive software patching solution that meets any type of software update requirements for AIX systems.

- Light-weight component architecture that provides excellent performance.

PowerSC TNC is a solution specifically for AIX environments. Because it specializes in AIX endpoints, it was implemented with a small code base that provides fast performance for update and verification operations. This small code base also integrates multi-threaded support for all verify operations, so the performance of these operations against groups of systems is as efficient as possible.

- Provides restart details about update operations.

Most security ifixes do not require a restart after being applied, but some do. TNC eliminates the question of whether an update requires a restart by displaying a restart required field for updates that require a restart after being applied.

- Flexible command line functions that facilitate automation.

PowerSC TNC implemented its functionality by using standard UNIX command line conventions. This feature is conducive for integration with solutions that provide automation, such as in-house scripting or third-party automation solutions.

- Automatic updating of patch repository that includes updating ifixes with superseding versions.

When a vulnerability is first identified, an initial security ifix is published to IBM Fix Central. However, sometimes newer versions for the same vulnerability are later published. When this issue occurs, the old version of the ifix must be replaced with the newer superseding version.

PowerSC TNC addresses this requirement by not only downloading the newer version of the ifix to the TNCPM, but also by having the TNCS verify operation check if an ifix is superseded by another ifix that is based on an ifix number and related files information. If the TNCS does detect that an ifix is superseded by a newer version, the TNCS uses the superseding version for the verify operation.

## 5.4 Deployment considerations

This section describes issues that you must consider before deploying the solution.

### 5.4.1 Disk and memory requirements

This section describes the disk and memory requirements for the solution.

#### Minimum requirements

Figure 5-1 on page 157 shows general minimal recommendations. Requirements can vary depending on the details of the number of AIX technology levels and service packs you must support and the number of TNC clients.

**Important:** The resource sizes that are described in this section are dedicated to TNC. This sizing does not consider non-TNC elements that are running on the same host on which the TNC component is running.

Table 5-1 General recommendations

	Disk	Memory
TNCPM	25 G	5 GB
TNCS	2 G	1 GB
TNCC	1 G	500 MB

### 5.4.2 Requirements to install software

Table 5-2 lists the general support and recommended PowerSC TNC component installation matrix.

Table 5-2 PowerSC TNC component installation matrix

	VIOS	AIX 6.1	AIX 7.1	AIX 7.2
TNCPM	N/A	Not supported	Not recommended	Recommended
TNCS	N/A	All	All	All
TNCC	2.2 versions or later	All	All	All



Table 5-3 lists the requirements matrix for TNCPM.

*Table 5-3 PowerSC TNCPM requirement matrix*

AIX	SUMA	OpenSSL	Notes
7.2 TL 1	7.2.1.0	1.0.2	Supplied with the operating system
7.2 TL 0	7.2.1.0	1.0.2	SUMA and Java might need to be installed separately
7.1 TL 4	7.2.1.0	1.0.2	SUMA and Java might need to be installed separately

Table 5-4 lists the fileset installation matrix for TNC components.

*Table 5-4 PowerSC TNC fileset install matrix*

Fileset name	Requirement for TNCPM	Requirement for TNCS	Requirement for TNCC	Requirement for IPREF
openpts.verifier	No	No	No	No
powerscStd.ice	No	No	No	No
powerscStd.license	Yes	Yes	Yes	Yes
PowerscStd.rtc	No	No	No	No
powerscStd.svm	No	No	No	Yes
powerscStd.tnc_commands	Yes	Yes	Yes	Yes
powerscStd.tnc_lib	Yes	Yes	Yes	Yes
powerscStd.tnc_pm	Yes	No	No	No
powerscStd.uiAgent	No	No	Optional	No
powerscStd.uiServer	No	No	No	No
powerscStd.vlog	No	No	No	No
ca-certificates	Yes	No	No	No
curl	Yes	No	No	No
libgcc	Yes	No	No	No

**Important:** If you want to add the ability for the PowerSC GUI to perform verification or update operations against a TNC Client, install the TNCC as a GUI-managed endpoint by using the powerscStd.uiAgent fileset and corresponding configuration procedure.

### 5.4.3 Host installation matrix for TNC components

Table 5-5 lists the TNC components that can be installed on the same host.

*Table 5-5 PowerSC TNC components on the same host*

	TNCC	TNCS	TNCPM
TNCC installed on AIX host	N/A	No	No
TNCC installed on VIOS host	N/A	No	No

TNCS installed on AIX host	No	N/A	Yes
TNCPM installed on AIX host	No	Yes	N/A

**Note:** For a security best practice, it is recommended that the TNCS and TNCPM be installed on separate hosts. The rationale for this suggestion is that if an attacker accesses the system that contains the TNCPM or TNCS, they can access only one PowerSC component, not both. However, the matrix in Table 5-5 shows that the TNCS and the TNCPM properly function when they are installed on the same host.

#### 5.4.4 Syslog configuration

For all TNC component systems, enable \*.debug syslog to capture TNC messages, as shown in Example 5-1. (TNC sends syslog messages to user.info).

*Example 5-1 Sample configuration*

```
# echo "*.debug /var/log/debug.log rotate size 32m files 8 compress" \
>> /etc/syslog.conf
# touch /var/log/debug.log
# refresh -s syslogd
```

### 5.5 Installing TNCPM

This section describes installing TNCPM.

#### 5.5.1 Networking requirements for TNCPM internet connections

By using TNCPM, various types of security updates can be automatically and manually downloaded.

The following networking information is needed for communicating through a firewall or configuring an http or https proxy with the specific servers and ports that are used by TNCPM. (For more information about network-related security recommendations for TNCPM, see “Recommendations for securing the TNCPM” on page 173):

- Technology Level and Service Pack download requirements.

Table 5-6 lists the server information for software downloads.

*Table 5-6 Server information for software download requirements*

Server	IP address	Protocol/port	Protocol/port
esupport.ibm.com	192.42.56.189	http/80	https/443
esupport.ibm.com	192.42.60.189	http/80	https/443
esupport.ibm.com	129.42.54.189	http/80	https/443

For more information about how to restrict SUMA to use only https for all communication, see 5.5.2, “Configuring the TNCPM” on page 165.

- Ifix download requirements.

Table 5-7 lists the server information for ifix downloads.

Table 5-7 Server information for ifix software download

Server	Protocol/port
www3.software.ibm.com	https/443

**Important:** TNC uses only https for ifix download.

- Open source package download requirements.

By using TNCPM, open source packages can be downloaded in rpm and installp formats. These packages are always downloaded manually.

Several open source package sites are used for these downloads. The file `/etc/tncpm_openpack_sitelist.conf` lists the various sites that can be used when querying and downloading these packages.

You can edit this file to customize the set of protocols and download servers that are used when querying and downloading open source packages.

If you want to use scp, sftp, or https when querying or downloading from open source package sites, you must obtain the site's certificate and add it to the `/etc/security/certificates/tnc` directory. TNC populated the certificate for retrieving rpm and installp packages from this [IBM Software web page](#).

## 5.5.2 Configuring the TNCPM

Complete the following steps to configure PowerSC TNCPM:

1. Ensure sufficient file system space.

The TNCPM install uses mostly `/var` and `/tmp` for its configuration. Ensure that at least 10 GB available is in `/var` and 500 MB available in `/tmp`.

2. Configure the Primary Service Configuration.

Complete the following steps to configure AIX to use a proxy for http or https communication by using smitty:

- a. As root on the TNCPM, run the `smitty srv_conn` command.
- b. Highlight **Create/Change Primary Service Configuration**.
- c. Press Enter.
- d. Highlight **Connection Type** and press the Tab key to select **HTTP\_PROXY**.
- e. Specify the port number of your proxy.
- f. Use the Test service configuration option to verify your configuration.
- g. After successfully testing the configuration, you can set the test option back to No and apply the configuration.

**Important:** Contact your network administrator for help or more information about specifying the correct configuration for your proxy.

3. Complete the following steps to configure SUMA:

a. Set Download Protocol to HTTPS.

The default configuration of SUMA specifies HTTP for the download protocol. HTTPS must be used for the download protocol. Run the following command on the TNCPM:

```
# suma -c -a DOWNLOAD_PROTOCOL=HTTPS
```

The configuration is listed, as shown in Example 5-2 on page 166.

*Example 5-2 Listing the configuration*

---

```
root@lbtnc1> suma -c
FIXSERVER_PROTOCOL=https
DOWNLOAD_PROTOCOL=https
DL_TIMEOUT_SEC=180
DL_RETRY=1
HTTP_PROXY=
HTTPS_PROXY=
SCREEN_VERBOSE=LVL_INFO
NOTIFY_VERBOSE=LVL_INFO
LOGFILE_VERBOSE=LVL_VERBOSE
MAXLOGSIZE_MB=1
REMOVE_CONFLICTING_UPDATES=yes
REMOVE_DUP_BASE_LEVELS=yes
REMOVE_SUPERSEDE=yes
TMPDIR=/var/suma/tmp
```

---

b. Verify that SUMA can receive data from IBM Fix Central and IBM Electronic Customer Care (ECC), as shown in the following working example (condensed using "... etc ..."):

```
# suma -x -a Action=Preview -a RqType=Latest
*****
Performing preview download.
*****
Partition id was unassigned; will attempt to assign it.
Partition id assigned value 12
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/7200-01-04-1806.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/U877996.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/U877995.bff
... etc ...
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/U872710.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/U872706.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/U872704.bff
Total bytes of updates downloaded: 2011392512
Summary:
    271 downloaded
    0 failed
skipped
root@lbtnc1>
```

c. Verify that SUMA can download data from IBM Fix Central and IBM ECC.

This second test is a deeper verification that your TNCPM can download updates from the internet, as shown in the following working example:

```
root@lbtnc1> suma -x -a Action=Download -a RqType=PTF -a RqName=U813941 \
-a FilterML=6100-01 -a DLTarget=/tmp
Partition id was unassigned; will attempt to assign it.
Partition id assigned value 12
```

```
Download SUCCEEDED: /tmp/installp/ppc/U813941.bff
Total bytes of updates downloaded: 1331200
Summary:
```

```
    1 downloaded
    0 failed
    0 skipped
```

```
root@lbtnc1>
```

- d. Verify curl can interact with IBM Fix Central:

```
# curl --silent --cacert /etc/security/certificates/tnc/IBM_IFIX_cert.pem
--list-only https://www3.software.ibm.com/aix/efixes/security/
```

- e. Initialize TNCPM.

This step is the preliminary step in configuring TNCPM. The TNCPM downloads the latest service pack and the latest ifixes relative to the latest service pack in question.

The command uses the following syntax:

```
pmconf init -i <download interval> -l <TL List> -A [ -P <download path>] [
-x <ifix interval>] [ -K <ifix key>]
```

The following example is an actual working example (condensed using "... etc ..."):

```
root@lbtnc1> pmconf init -i d1:h8:m0 -l 7200-01 -A -x 60
```

```
New ifix interval check set to 60
```

```
accept_all_licenses for TNC Clients update set to yes
```

```
New Service Pack interval check set to d1:h8:m0
```

```
Initializing 7200-01
```

```
Downloading Metadata for 7200-01
```

```
Platform Extension: information for proxy SAS not found in repository
```

```
Partition id was unassigned; will attempt to assign it.
```

```
Partition id assigned value 12
```

```
Storing auth proxy creds for SAS
```

```
successfully stored auth proxy creds for SAS
```

```
Storing auth proxy creds for PROFILE_URIS_LENGTH
```

```
successfully stored auth proxy creds for PROFILE_URIS_LENGTH
```

```
Storing auth proxy creds for PROFILE_URI_0
```

```
successfully stored auth proxy creds for PROFILE_URI_0
```

```
Download SUCCEEDED:
```

```
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/7200-01-04-180
6.dd.xml
```

```
Download SUCCEEDED:
```

```
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/7200-01-04-180
6.install.tips.html
```

```
Download SUCCEEDED:
```

```
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/U875433.pd.sdd
```

```
Download SUCCEEDED:
```

```
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/U875433.dd.xml
```

```
Download SUCCEEDED:
```

```
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/aix_7200-01-01
.special.note.txt
```

```
Total bytes of updates downloaded: 544178
```

```
Summary:
```

```
    17 downloaded
    0 failed
    0 skipped
```

```
Latest Service Pack is 7200-01-04
```

```
Downloading Service Pack 7200-01-04
```

```

Partition id was unassigned; will attempt to assign it.
Partition id assigned value 12
New ifix interval check set to 60
accept_all_licenses for TNC Clients update set to yes
New Service Pack interval check set to d1:h8:m0
Initializing 7200-01
Downloading Metadata for 7200-01
Platform Extension: information for proxy SAS not found in repository
Partition id was unassigned; will attempt to assign it.
Partition id assigned value 12
Storing auth proxy creds for SAS
successfully stored auth proxy creds for SAS
Storing auth proxy creds for PROFILE_URIS_LENGTH
successfully stored auth proxy creds for PROFILE_URIS_LENGTH
Storing auth proxy creds for PROFILE_URI_0
successfully stored auth proxy creds for PROFILE_URI_0
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/7200-01-04-180
6.dd.xml
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/7200-01-04-180
6.install.tips.html
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/7200-01-04-180
6.pd.sdd
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/U875433.pd.sdd
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/U875433.dd.xml
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/metadata/installp/ppc/aix_7200-01-01
.special.note.txt
Total bytes of updates downloaded: 544178
Summary:
    17 downloaded
    0 failed
    0 skipped
Latest Service Pack is 7200-01-04
Downloading Service Pack 7200-01-04
Partition id was unassigned; will attempt to assign it.
Partition id assigned value 12
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/SPs/7200-01-04/installp/ppc/7200-01-
04-1806.bff
NOTE: ... removing actual output to condense text...
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/SPs/7200-01-04/installp/ppc/U872704.
bff
Total bytes of updates downloaded: 2011392512
Summary:
    271 downloaded
    0 failed
    0 skipped
Preparing to copy install images (this will take several minutes)...
Now checking for missing install images...

```

```

Checking for ifixes...
Checking website for advisories...
Scanning advisories for consistency...
Searching advisories for applicable ifixes (this will take several
minutes)...
Updating the cache for Product: [aixbase]
Downloading ifixes for Product: [aixbase]
Updating the cache for Product: [bellmail]
... note: removing some actual text to condense output ...
Updating the cache for Product: [wpar]
Downloading ifixes for Product: [wpar]
83 new ifixes downloaded.
Applying ifix(es) (this will take several minutes)...
Processing [aixbase_advisory]
Processing [bellmail_advisory]
Processing [bellmail_advisory2]
Processing [nettcp_advisory]
Processing [nettcp_advisory2]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 102m_ifix.180105.epkg.Z
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 fips_102m.180105.epkg.Z
Processing [openssl_advisory26]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 102ma_ifix.180410.epkg.Z
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 fips_102ma.180410.epkg.Z
Processing [pconsole_advisory]
Processing [pconsole_advisory2]
Processing [powerha_advisory]
Processing [rc4_advisory]
Processing [rmsock_advisory2]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 IJ06907sla.180607.epkg.Z
Processing [sendmail_advisory]
Processing [tcpdump_advisory3]
Processing [tftp_advisory]
Processing [variant4_advisory]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-04 IJ05820m4a.180423.epkg.Z
Processing [wpar_advisory]
root@l1bstnc1>

```

f. Initialize after a failed initialization.

Suppose that during your initialization, some type of failure caused the initialization to fail. This failure can be because of various factors, such as network, disk, or power failure. If a failure occurs, retry the initialization. TNC automatically deletes files that were downloaded in the failed initialization.

g. Add lower-level service packs.

After you initialize the TNCPM, it is a good practice (but not required) to download all service packs that are related to the technology levels you initialized against. For example, suppose you initialized the TNCPM against 7200-01 and the initialization downloaded 7200-01-04 as the latest service pack. You add service packs 1 - 3 for 7200-01.

**Note:** Check you have sufficient disk space throughout the download process, otherwise, your installation can fail.

The command uses the following syntax:

```
pmconf add -p <SP List> [ -U <user-defined SP path> ]
```

The following example is an actual working example (condensed using "... etc ..."):

```
# pmconf add -p 7200-01-01
partition id assigned value 12
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/SPs/7200-01-01/installp/ppc/U875433.
bff
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/SPs/7200-01-01/installp/ppc/U872703.
bff
... etc ...
Download SUCCEEDED:
/var/tnc/tncpm/fix_repositories/7200-01/SPs/7200-01-01/installp/ppc/U872693.
bff
Total bytes of updates downloaded: 87010816
Summary:
    9 downloaded
    0 failed
    0 skipped
Preparing to copy install images (this will take several minutes)...
Now checking for missing install images...
Checking for ifixes...
Checking website for advisories...
Scanning advisories for consistency...
Searching advisories for applicable ifixes (this will take several
minutes)...
No new ifixes downloaded.
Applying ifix(es) (this will take several minutes)...
Processing [aixbase_advisory]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-01 IJ02828s1a.171221.epkg.Z
Processing [bellmail_advisory]
NEW IFIX
New Ifix registered with TNCPM for 7200-01-01 IV91011s1a.161125.epkg.Z
Processing [bellmail_advisory2]
... etc ...
NEW VIOS IFIX
New VIOS Ifix registered with TNCPM for 2.2.4.60 IJ05824sBa.180426.epkg.Z
NEW VIOS IFIX
New VIOS Ifix registered with TNCPM for 2.2.6.0 IJ05824mAa.180501.epkg.Z
Processing [wpar_advisory]
root@lbtnc1>
```

#### h. Configure TNCPM.

This task sets up the TNCPM to listen on its own port and establishes connection information for its TNCS.

The command uses the following syntax:

```
pmconf mktncpm [ pmpport=<port> ] tncserver=ip | hostname : <port>
```



The following output is produced:

```
root@lbtnc1> pmconf mktncpm pmpport=20000 tncserver=10.3.126.34:10000
Starting component :TNCMP
tncpmd daemon started successfully pid 12255622
root@lbtnc1>
```

i. Query open source sites for packages.

The TNCMP queries several sites for open source packages. Use this function to determine what packages you can might download to your TNCMP.

The command uses the following syntax:

```
pmconf get -L -o <package> -V <version | all> -T <installp | rpm>
```

The following output is produced:

```
root@lbtnc1> pmconf get -o lsof -T rpm -L -V all
Attempting to obtain list for [lsof] from
[https://www3.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to obtain list for [lsof] from
[http://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to obtain list for [lsof] from
[ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to obtain list for [lsof] from [ftp://www.oss4aix.org/RPMS/lsof]
The following packages were found:
lsof-4.85-1.aix6.1.ppc.rpm
lsof-4.85-1.aix7.1.ppc.rpm
lsof-4.86-1.aix6.1.ppc.rpm
lsof-4.86-1.aix7.1.ppc.rpm
lsof-4.87-1.aix6.1.ppc.rpm
lsof-4.87-1.aix7.1.ppc.rpm
lsof-4.88-1.aix6.1.ppc.rpm
lsof-4.88-1.aix7.1.ppc.rpm
lsof-4.88-1.aix7.2.ppc.rpm
lsof-4.89-1.aix6.1.ppc.rpm
lsof-4.89-1.aix7.1.ppc.rpm
lsof-4.89-1.aix7.2.ppc.rpm
root@lbtnc1>
```

j. Download open source packages.

This function allows you to download a particular open source package.

The command uses the following syntax:

```
pmconf get -o <package> -V <version> -T <installp | rpm> -D <download
directory>
```

The following example output is produced:

```
root@lbtnc1> pmconf get -o lsof -V 4.89-1.aix7.2 -T rpm -D /lsof
```

```

Attempting to download [lsof] from
[https://www3.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:download_package: Unable to execute the download!
Status=[1]
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to download [lsof] from
[http://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:download_package: Unable to execute the download!
Status=[1]
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to download [lsof] from
[ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/lsof]
No matches found
tncpm_openpack_download:download_package: Unable to execute the download!
Status=[1]
tncpm_openpack_download:process_sitelist_entry: Error [1] attempting to
locate package(s) from this site
Attempting to download [lsof] from [ftp://www.oss4aix.org/RPMS/lsof]
Download Complete. File(s) saved to /lsof/lsof_4.89-1.aix7.2
root@lbtnc1>

```

k. Register an open source package to the TNCPM.

This operation updates the TNCPM with a open source package it has downloaded.

The command uses the following syntax:

```
pmconf add -o <package name> -V <version> -T [installp|rpm] -D <User defined path>
```

Before you register the open package with the TNCS, verify that the package count uses **psconf pull**. In the following output, you can see that our TNCS do not include any open-source packages registered:

```

root@lbtnc1> psconf pull
debug1: [psconf] 11141504:00000001 TNCS_pull(): Command /usr/sbin/tnc
--pull=10.3.126.34:20000
Running transaction
Transaction Summary
Pulled      4 SPs    71 Apars    27 Advisories    239 Ifixes    563 Filesets
0 Packages
Total transaction size: 590821 byte(s)
Total transaction time: 0.01 sec(s)
Transaction succeeded
root@lbtnc1>

```

Register the open source package, as shown in the following example:

```

root@lbtnc1> pmconf add -o lsof -V 4.89-1.aix7.2 -T rpm -D
/lsof/lsof_4.89-1.aix7.2
/var/tnc/tncpm/fix_repositories/packages
Preparing to copy install images (this will take several minutes)...
Now checking for missing install images...
Package [lsof] Version [4.89-1.aix7.2] added to repository and registered
with TNC

```

```
root@lbtnc1>
```

Now, restart the TNCPM and perform a **psconf** pull operation to see if a new open source package is registered to the TNCs. In the following output, you can see that “1 Packages” is now being reported:

```
root@lbtnc1> pmconf stop
tncpmd daemon stopped successfully
root@lbtnc1> pmconf start
Starting component :TNCPM,SERVER
tncsd daemon started successfully pid 9830702
root@lbtnc1> psconf pull
debug1: [psconf] 11141522:00000001 TNCs_pull(): Command /usr/sbin/tnc
--pull=10.3.126.34:20000
Running transaction
Transaction Summary
Pulled      4 SPs    71 Apars    27 Advisories    239 Ifixes    563 Filesets
1 Packages
Total transaction size: 591181 byte(s)
Total transaction time: 0.01 sec(s)
Transaction succeeded
root@lbtnc1>
```

I. Enable TNCPM to use non-security ifixes.

TNC automatically downloads security fixes. You can add HIPER, IBM FileNet® Process Engine, and Enhancement to the default configuration.

After this step is performed, you must add these types of ifixes as stand-alone fixes. TNC 1.2.0.0 supports only the automatic download of security ifixes.

The command uses the following syntax:

```
pmconf modify -t <APAR type list>
```

The following output is produced:

```
# pmconf modify -t HIPER,PE,Enhancement
# pmconf stop
# pmconf start
```

Recommendations for securing the TNCPM

To provide the best security for your TNCPM implementation, complete the following steps:

- a. Instead of using a direct connection to the internet, connect through an http proxy.  
A secure proxy implementation uses a whitelisting approach for authorizing which connections are allowed. For a secure whitelisting proxy configuration, you need the proxy to authorize the https ports of the servers, as described in 5.5.1, “Networking requirements for TNCPM internet connections” on page 164.
- b. Ensure SUMA is using only HTTPS. For more information, see 5.5.2, “Configuring the TNCPM” on page 165.
- c. Configure the `/etc/tncpm_openpack_sitelist.conf` to contact servers with the sftp, scp, or https protocols only.

### 5.5.3 Configuring the Trusted Network Connect Server

Complete the following steps to configure the TNCS:

1. Use the following command to configure the TNCS. This command configures the TNCS daemon, tnccd:

```
psconf mkserver [ tncport=<port> ] pmserver=<host:port> [tsserver=<host>]
[recheck_interval=<time_in_minutes> | d (days): h (hours) : m (minutes) ]
[dbpath = <user-defined directory> ][default_policy=<yes | no > ]
[clientData_interval=<time_in_minutes> | d (days) : h (hours) : m (minutes) ] [
clientDataPath=<Full_path >]
```

The following output is produced:

```
# psconf mkserver tncport=10000 pmserver=10.3.126.45:20000 recheck_interval=60
```

2. Start and stop the TNCS.

When a configuration change is applied to the TNCS, you often must bounce the server by using the following command:

```
psconf { start | stop | restart } server
```

The following output is produced:

```
root@l1bstnc1> psconf stop server
tnccsd daemon stopped successfully
root@l1bstnc1> psconf start server
root@l1bstnc1>
```

3. Create a TNC group, which is a set of partitions, by using the following command:

```
psconf add { -G <ipgroupname> ip=[1]<host1, host2...> | {-A<apargrp>
[aparlist=[1]apar1, apar2...|
{-V <ifixgrp> [ifixlist=[+|-]ifix1,ifix2...]
```

The following output is produced:

```
# psconf add -G tncgrp ip=lbsaix1,lbstds3,lbsaix2
```

4. List the members of a TNC group by using the following command:

```
psconf list { -S | -G <ipgroupname | ALL > | -F <FSPolicyname | ALL > | -P <
policyname | ALL > | -r <buildinfo | ALL > | -I -i <ip | ALL > | -A <apargrp
| ALL > | -V <ifixgrp> | -O <openpkggrp|ALL>} [-c] [-q]
```

The following output is produced:

```
root@l1bstnc1> psconf list -G tncgrp
#ipgroupname      ip                  policyname          EMAILID
EMAILTYPE
tncgrp            lbsaix1.aus.stglabs.ibm.com  tncpol              ....
....
tncgrp            lbstds3.aus.stglabs.ibm.com  tncpol              ....
....
root@l1bstnc1>
```

5. Create a filesset policy.

A filesset policy is a set of filesets that meet security policies. A filesset policy is defined by an AIX service pack level and can include apar groups, ifix groups, and open package groups.

**Important:** The fileset policy cannot be used with a VIOS TNC Client. Only the default policy can be used when verifying a VIOS TNC Client.

The command uses the following syntax:

```
psconf add -F <FSPolicyname> -r <buildinfo> [apargrp= [1]<apargrp1, apargrp2..>] [ifixgrp=[+|-]<ifixgrp1,ifixgrp2...>]
```

The following output is produced:

```
# psconf -F fspol -r 7100-03-06
```

#### 6. Create a TNC policy.

When you create a TNC policy, you must define two parts to the TNC policy. One part defines which TNC Groups belong to the TNC Policy. The second part defines what filesset policies belong to the TNC Policy. Perform both of the following operations to create a TNC Policy:

- Map a set of IP groups to a TNC Policy.

A TNC policy is a combination of one or more IP groups and one or more filesset policies. A TNC policy determines for a set of IP groups what filesset policies achieve patch compliance for the IP group members. Therefore, one or more IP groups map to a single TNC policy, and one or more filesset policies map to a TNC policy.

The command uses the following syntax:

```
psconf add -P <policyname> { fspolicy=[1]<f1,f2...> | ipgroup=[1]<g1,g2...> }
```

The following output is produced:

```
# psconf -P tncpol ipgroup=tncgrp
```

- Map a set of filesset policies to a TNC policy.

A TNC policy is a combination of one or more IP groups and one or more filesset policies. A TNC policy determines for a set of IP groups what filesset policies achieve patch compliance for the IP group members. Therefore, one or more IP groups map to a single TNC policy, and one or more filesset policies map to a TNC policy.

The command uses the following syntax:

```
psconf add -P <policyname> { fspolicy=[1]<f1,f2...> | ipgroup=[1]<g1,g2...> }
```

The following output is produced:

```
# psconf -P tncpol fspolicy=fspol
```

#### 7. List the service packs that are registered to the TNCS.

The TNCPM is the component that downloads all updates. For TNCS to issue an update operation to a client, it must refer to an update that was registered to the TNCS by the TNCPM.

The command uses the following syntax:

```
psconf list { -S | -G <ipgroupname | ALL > | -F <FSPolicyname | ALL > | -P <policyname | ALL > | -r <buildinfo | ALL > | -I -i <ip | ALL > | -A <apargrp | ALL > | -V <ifixgrp> | -O <openpkggrp|ALL>} [-c] [-q]
```

The following output is produced:

```
root@l1bstnc1> psconf list -r ALL
Lslpp SP:
#Release TL      SP
7200      1      1
```

```

7200    1      2
7200    1      3
7200    1      4
Apar SP:
#Release TL      SP
7200    1      1
7200    1      2
7200    1      3
7200    1      4

```

**root@lbtnc1>**

8. List the ifixes and apars that are registered to the TNCS for a service pack.

The TNCPM is the component that downloads all updates. For TNCS to issue an update operation to a client, it must refer to an update that was registered to the TNCS by the TNCPM.

The command uses the following syntax:

```

psconf list { -S | -G < ipgroupname | ALL > | -F < FSPolicyname | ALL > | -P <
policyname | ALL > | -r < buildinfo | ALL > | -I -i < ip | ALL > | -A <
apargrp | ALL > | -V < ifixgrp > | -O < openpkggrp | ALL > } [-c] [-q]

```

The following output is produced, which is a working example (condensed using "... etc ..."):

**root@lbtnc1> psconf list -r 7200-01-04**

#ifix	Release	TL	SP	cve	cvss
fileset	vrnf				
102j_ifix.170207.epkg.Z	7200	1	4	....	....
openssl.base	20.13.102.1000				
102j_ifix.170207.epkg.Z	7200	1	4	....	....
openssl.base	1.0.2.1000				
102m_ifix.180105.epkg.Z	7200	1	4	....	....
openssl.base	20.13.102.1300				
102m_ifix.180105.epkg.Z	7200	1	4	....	....
openssl.base	1.0.2.1300				
102ma_ifix.180410.epkg.Z	7200	1	4	....	....
openssl.base	20.13.102.1300				
102ma_ifix.180410.epkg.Z	7200	1	4	....	....
openssl.base	1.0.2.1300				
517_ifix.170113.epkg.Z	7200	1	4	....	....
openssl.base	20.13.101.500				
517_ifix.170113.epkg.Z	7200	1	4	....	....
openssl.base	1.0.1.517				
... etc ...					

#aparname	Release	TL	SP	Apar_type	Fileset
abstract					
IJ02828	7200	1	4	Security	
bos.cluster.rte 7.2.1.3, A potential security issue exists					
IJ03035	7200	1	4	Security	
bos.pmapapi.pmsvcs 7.2.1.3,bos.mp64 7.2.1.5, A POTENTIAL SECURITY ISSUE EXISTS					
IV96310	7200	1	4	Security	
bos.net.tcp.ntpd 7.2.1.2, A potential security issue exists					
IV97898	7200	1	4	Security	bos.acct
7.2.1.2, A potential security issue exists					
IV97901	7200	1	4	Security	bos.acct
7.2.1.2, A potential security issue exists					

```

IV97958          7200    1      4      Security
bos.rte.archive 7.2.1.2, A potential security issue exists
IV98298          7200    1      4      Security          bos.rte.lvm
7.2.1.3, A potential security issue exists
IV98830          7200    1      4      Security
bos.net.tcp.bind_utils 7.2.1.3, A potential security issue exists
IV99499          7200    1      4      Security
bos.net.tcp.client_core 7.2.1.3, A potential security issue exists
IV99552          7200    1      4      Security          bos.rte.lvm
7.2.1.3, A potential security issue exists
#

```

#### 9. List an ifix by name.

A detailed listing of a security advisor can be obtained by referencing an ifix name by using the following command:

```
psconf report -v ALL -o TEXT | grep -p <name of ifix>
```

The following output is produced:

```

# psconf report -v ALL -o TEXT | grep -p IV96310m2a.170519.epkg.Z
AIX Advisory: ntp_advisory9.asc
Abstract: ....
Reboot: ....
Workaround: ....
  CVE: CVE-2017-6464
  CVSS: 4.2 4.2
  CVE: CVE-2017-6462
  CVSS: 1.6 1.6
  CVE: CVE-2017-6458
  CVSS: 4.2 4.2
  CVE: CVE-2017-6451
  CVSS: 1.8
  Ifix: IV96306m9a.170519.epkg.Z
    Release: 537214528-537214704-537214816
    APAR: IV96306
    APAR Release: 6100-09-10
    Fileset: bos.net.tcp.client
    VRMF: 6.1.9.201
  Ifix: IV96310m2a.170519.epkg.Z
    Release: 7200-01-01
    APAR: IV96310
    APAR Release: 7200-01-04
    Fileset: bos.net.tcp.ntpd
    VRMF: 7.2.1.1
    Fileset: bos.net.tcp.ntp
    VRMF: 7.2.1.0
  Ifix: IV96312m5a.170518.epkg.Z
    Release: 7200-01-01
    APAR: IV96312
    APAR Release: 7200-01-03
    Fileset: ntp.rte
    VRMF: 7.1.0.9
#

```

#### 10. List all security advisories that are known to the TNCS.

Perform this action if you want a detailed listing of all the security advisories that are registered to the TNCS.

The command uses the following syntax:

```
psconf report -v <CVEid | ALL> -o <TEXT | CSV>
```

The following output is produced, which is a working example (condensed using "... etc ..."):

```
root@l1bstnc1> psconf report -v ALL -o TEXT
Report Date: Tue Sep 18 20:43:14 2018
Version: 1.0
Advisories:
AIX Advisory: variant4_advisory.asc
Abstract: ....
Reboot: ....
Workaround: ....
  Ifix: IJ05820m2a.180430.epkg.Z
    Release: 7200-01-02
    APAR: IJ05820
    APAR Release: 7200-01-05
    Fileset: bos.mp64
    VRMF: 7.2.1.5
  Ifix: IJ05820m3a.180430.epkg.Z
    Release: 7200-01-03
    APAR: IJ05820
    APAR Release: 7200-01-05
  Ifix: IJ05820m4a.180423.epkg.Z
    Release: 7200-01-04
    APAR: IJ05820
    APAR Release: 7200-01-05
  Ifix: IJ05824m9a.180501.epkg.Z
    Release: 537213360-537213536-537213712
    APAR: IJ05824
    APAR Release: 6100-09-12
  Ifix: IJ05824m9b.180502.epkg.Z
    Release: 537213360-537213536-537213712
    APAR: IJ05824
    APAR Release: 6100-09-12
  Ifix: IJ05824mAa.180501.epkg.Z
    Release: 537213360-537213536-537213712
    APAR: IJ05824
    APAR Release: 6100-09-12
  Ifix: IJ05824sBa.180426.epkg.Z
    Release: 537213360-537213536-537213712
    APAR: IJ05824
    APAR Release: 6100-09-12
... etc ...
#
```

11. List a security advisory by CVEID if you want a detailed listing of a particular security advisor by referencing its CVEID.

The command uses the following syntax:

```
psconf report -v <CVEid | ALL> -o <TEXT | CSV>
```



The following output is produced:

```
root@l1bstnc1> psconf report -v CVE-2018-0739 -o TEXT
Report Date: Sun Jul 22 19:35:16 2018
Version: 1.0
Advisories:
AIX Advisory: openssl_advisory26.asc
Abstract: ....
Reboot: TNCC
Workaround: ....
    CVE: CVE-2018-0739
    CVSS: 5.3
    Ifix: 102ma_ifix.180410.epkg.Z
        Release: 537210400-537210576-537210752
        APAR: N/A
        APAR Release: N/A
        Fileset: openssl.base
            VRMF: 20.13.102.1300
    Ifix: fips_102ma.180410.epkg.Z
        Release: 537232160-537232336-537232512
        APAR: N/A
        APAR Release: N/A
```

```
root@l1bstnc1>
Sync TNCS with TNCPM
```

When using the **psconf** command on the TNCS, you cannot reference the name of a service pack, apar, ifix, or open source package unless it was first downloaded to the TNCPM and then registered to the TNCS.

To register everything that was downloaded on the TNCPM by using the TNCS, you must perform the **psconf pull** operation.

The following output indicates that 4 AIX service packs, 20 AIX apars, 24 advisories, 200 ifixes, 493 filesets, and no open source packages were downloaded to the TNCPM and these updates are registered to the TNCS:

```
root@l1bstnc1> psconf pull
debug1: [psconf] 16515360:00000001 TNCS_pull(): Command /usr/sbin/tnc
--pull=10.3.126.34:20000
Running transaction
Transaction Summary
Pulled      4 SPs      20 Apars      24 Advisories      200 Ifixes      493 Filesets
0 Packages
Total transaction size: 530381 byte(s)
Total transaction time: 0.01 sec(s)
Transaction succeeded
root@l1bstnc1>
```

## 12. Create an ifix group.

When you want your systems to deploy a set of ifixes, you must create an ifix group and then map that group to a fileset policy. The fileset policy is mapped to a TNC policy.

The command uses the following syntax:

```
psconf add { -G <ipgroupname> ip=[1]<host1, host2...> |
{ -A<apargrp>[aparlist=[1]apar1, apar2... |
{ -V <ifixgrp> [ifixlist=[+|-] ifix1, ifix2...]}
```

The following output is produced:

```
root@l1bstnc1> psconf add -V ifixgrp1 ifixlist=IV96310m2a.170519.epkg.Z
```

```
root@lbstnc1>
```

### 13. Create a fileset policy with an ifix group.

An ifix group does not take effect until you map it to a Fileset Policy, which must be mapped to a TNC policy.

The command uses the following syntax:

```
psconf add -F <FSPolicyname> -r <buildinfo> [apargrp= [1]<apargrp1, apargrp2..>] [ifixgrp=[+|-]<ifixgrp1, ifixgrp2...>]
```

The following output is produced:

```
root@lbstnc1> psconf add -F fspol -r 7200-01-02 ifixgrp=ifixgrp1
```

### 14. Create an apar group.

When you want your systems to deploy a set of apars, you must create an apar group and then map that group to a fileset policy. The fileset policy is then mapped to a TNC policy.

The command uses the following syntax:

```
psconf add { -G <ipgroupname> ip=[1]<host1, host2...> |  
{-A<apargrp>[aparlist=[1]apar1, apar2... |  
{-V <ifixgrp> [ifixlist=[+|-] ifix1, ifix2...}]}
```

The following output is produced:

```
root@lbstnc1> psconf add -A apargrp1 aparlist=IV60303  
root@lbstnc1>
```

### 15. Create a fileset policy with an apar group.

An apar group does not take effect until you map it to a fileset Policy, which must be mapped to a TNC policy.

The command uses the following syntax:

```
psconf add -F <FSPolicyname> -r <buildinfo> [apargrp= [1]<apargrp1, apargrp2..>] [ifixgrp=[+|-]<ifixgrp1, ifixgrp2...>]
```

The following output is produced:

```
root@lbstnc1> psconf add -F fspol2 -r 7100-03-03 apargrp=apargrp1
```

### 16. Create an open package group.

When you want your systems to deploy a set of open source packages, you must create an open package group and then map that to a fileset policy. The fileset policy is then mapped to a TNC policy.

The command uses the following syntax:

```
psconf add -O <openpkggrp> <openpkgname:version>
```

The following output is produced:

```
root@lbstnc1> psconf add -O opengrp1 lsof:4.89-1.aix7.2  
Successfully added the attribute OpenPackage Group  
Successfully added the attribute OpenPackage  
root@lbstnc1>
```

### 17. Create a fileset policy with an open package group.

You can map an open package group to a fileset policy. After it is attached to a fileset policy, a verification operation that is run by using the fileset policy verifies that the open package is installed on the TNCC.

As shown in Figure 5-2 on page 181, the fileset policy is shown without an open package group, the open package groups are listed, the open package group is added to the fileset policy, and then, the fileset policy is listed showing the added open group.

The command uses the following syntax:

```
psconf add -O <openpkggrp> fspolicy=<fspolicy name>
```

Figure 5-2 shows the command listing the fileset policies.

```

root@l1bstnc1> psconf list -F ALL
#fspolicyname      Release TL      SP      AparGroup      IfixGroup      AutoUpdate      OpenPackageGroup
fspol              7200      1      2      ....      ....      ....
root@l1bstnc1> psconf list -O ALL
#OpenPackageGroup  name      PackVersion      TNCMPVersion      InstallType      Reboot
opengrp1          lsof      4.89-1.aix7.2      1.0      rpm      no
root@l1bstnc1> psconf add -O opengrp1 fspolicy=fspol
root@l1bstnc1> psconf list -F ALL
#fspolicyname      Release TL      SP      AparGroup      IfixGroup      AutoUpdate      OpenPackageGroup
fspol              7200      1      2      ....      ....      ....
root@l1bstnc1>

```

Figure 5-2 Listing the fileset policies

## 5.5.4 Configuring the Trusted Network Connect Client

Complete the following steps to configure the TNC:

1. Configure the NIM client.

TNC uses a NIM client on the TNCC to install updates. Run the following command to configure your TNCC as a NIM client:

```
nimit{-a name=Name -a pif_name=Pif -a master=Hostname} [ -a
master_port=PortNumber ] [ -a registration_port=PortNumber ] [ -a
cable_type=Type | -a ring_speed=Speed ] [-a iplrom_emu=Device ] [ -a
platform=PlatformType ] [ -a netboot_kernel=NetbootKernelType ] [-a
adpt_add=AdapterAddress] [ -a is_alterate= yes | no ] [ -a connect=value ] [
-a vlan_tag=value] [-a vlan_pri=value]
```

The following output is produced:

– Not using Nimsh:

```
# rm /etc/niminfo
```

```
# nimit -a master=<short hostname of TNCPM> -a name=<short hostname of TNCC>
```

– Using Nimsh:

```
nimit -a master=<short hostname of TNCPM> -a name=<short hostname of TNCC> -a
\ pif_name=en0 a connect=nimsh
```

2. Configure the TNC client daemon (tnccd) that communicates with the TNCS.

The command uses the following syntax:

```
psconf mkclient [ tncport=<port> ] tncserver=<host:port>
```

The following output is produced:

```
root@l1bsaix1> psconf mkclient tncport=10000 tncserver=l1bstnc1:10000
root@l1bsaix1>
```

3. Check the status. If you are seeing a problem on the client, you can query its status. (A working client is shown in the following example.)

The command uses the following syntax:

```
psconf status
```

The following output is produced:

```

root@lbsaix1> psconf status
component = CLIENT
tncport = 10000
tncserver = lbstnc1
trustmode = false
tnccd daemon pid 10158392
root@lbsaix1>

```

## 5.5.5 Configuring Trusted Network Connect Server email

This section shows how to configure the TNCS email.

1. Configure email for TNCPM downloads notifications.

When the TNCPM downloads a new service pack or ifix, this notification is logged to /var/spool/mail/root. You can use the mail command to configure the system to drnf these notifications by way of email as well.

2. Configure email for verification operation results.

In PowerSC 1.2.0.2, TNC can send emails that provide the results of verification operations. Emails can also be sent when verifications succeed or fail. If you want to limit notifications, you can filter your notifications by using IP group. You can also filter email messages by type by using option -E.

The command uses the following syntax:

```
psconf add -e emailid [-E FAIL | COMPLIANT | ALL ] [ipgroup= [1]<g1,g2...>]
```

The following output is produced:

```
root@lbstnc1> psconf add -e sdoming@us.ibm.com
```

## 5.6 Working with Trusted Network Connect and Patch Management

This section describes how to work with the TNC and Patch Management.

### 5.6.1 Verifying the Trusted Network Connect Client

This section describes verifying the TNCC.

#### Default Policy

The term *default policy* refers to the set of ifixes and apars for any service pack that was downloaded by the TNCPM and registered to the TNCS. When a verification operation occurs against a TNC Client and no matching fileset policy exists, the default policy is used to determine compliance.

**Important:** The VIOS TNC Client can use only the default policy for verification operations.

#### *Understanding the verify operation output.*

In the following example, the meaning of the different sections that are reported by a TNCS verify operation are clarified:

```
root@lbstnc1> psconf verify -i lbsaix1
```

Running transaction 10.3.126.48

*# note here that no fileset policy maps to the TNCC so the  
# default policy is used:*

Running verification based on default policy 7200-01-04  
# The ifixhigher.rpt is a listing of ifixes that are not needed on the  
# TNCC because the fileset version that is installed on the TNCC is higher than  
# what the listed ifixes were published to fix. This report is not very  
# important. Future releases will only provide this report when the  
# verbose option is activated.

**#(ifixhigher.rpt)**  
# ifixes with fileset that is at a higher level than maximum  
# Ifix\_Level: Ifix\_Name: Fileset\_Name: Ifix\_Version: Client\_Version  
7.2.1.4: 517\_ifix.170113.epkg.Z: openssl.base: 1.0.1.517: 1.0.2.800  
7.2.1.4: IV83169m9a.160401.epkg.Z: openssl.base: 1.0.2.500: 1.0.2.800  
7.2.1.4: IV83169m9a.160401.epkg.Z: openssl.base: 1.0.1.515: 1.0.2.800  
7.2.1.4: IV83169m9a.160401.epkg.Z: openssl.base: 0.9.8.2506: 1.0.2.800  
7.2.1.4: IV83169m9b.160401.epkg.Z: openssl.base: 1.0.2.500: 1.0.2.800  
7.2.1.4: IV83169m9b.160401.epkg.Z: openssl.base: 1.0.1.515: 1.0.2.800  
7.2.1.4: IV83169m9b.160401.epkg.Z: openssl.base: 0.9.8.2506: 1.0.2.800  
7.2.1.4: IV83169m9c.160401.epkg.Z: openssl.base: 1.0.2.500: 1.0.2.800  
7.2.1.4: IV83169m9c.160401.epkg.Z: openssl.base: 1.0.1.515: 1.0.2.800  
7.2.1.4: IV83169m9c.160401.epkg.Z: openssl.base: 0.9.8.2506: 1.0.2.800  
7.2.1.4: IV83169s9d.160401.epkg.Z: openssl.base: 1.0.2.500: 1.0.2.800  
7.2.1.4: IV83169s9d.160401.epkg.Z: openssl.base: 1.0.1.515: 1.0.2.800  
7.2.1.4: IV83169s9d.160401.epkg.Z: openssl.base: 0.9.8.2506: 1.0.2.800

# The ifixfilesetnotinst.rpt indicates that the following ifixes are  
# not needed because the filesets that correspond to the ifix aren't  
# installed on the TNCC

**#(ifixfilesetnotinst.rpt)**  
# ifix with filesets that are not installed on the other system  
# Ifix\_Level: Ifix\_Name: Fileset\_Name: Fileset\_Version  
7.2.1.4: IV83983s5a.160602.epkg.Z: ntp.rte: 7.1.0.5  
7.2.1.4: IV87279s7a.160901.epkg.Z: ntp.rte: 7.1.0.7  
7.2.1.4: IV92126m3a.170106.epkg.Z: ntp.rte: 7.1.0.7  
7.2.1.4: fips\_ifix.170113.epkg.Z: openssl.base: 1.0.1.517: 1.0.2.800

# the missingifixes.rpt is very important. It is listing the  
# ifixes that should be evaluated for deployment on the TNCC.  
# You should evaluate each of these ifixes for relevance to your  
# environment and determine when and if you will apply the ifix  
# to your TNC Clients that need the patch.

**#(missingifixes.rpt)**  
# ifixes that are not installed on the other system  
# Ifix\_Level: Ifix\_Name: Apar\_Name  
7.2.1.4: 102j\_ifix.170207.epkg.Z: N/A  
7.2.1.4: 102m\_ifix.180105.epkg.Z: N/A  
7.2.1.4: 102ma\_ifix.180410.epkg.Z: N/A  
7.2.1.4: 102oa\_ifix.180906.epkg.Z: N/A  
7.2.1.4: 6202\_ifix.160830.epkg.Z: N/A

```

7.2.1.4:6203_ifix.170124.epkg.Z: N/A
7.2.1.4:IJ05820m4a.180423.epkg.Z:IJ05820
7.2.1.4:IJ06400s9a.180514.epkg.Z:IJ06400
7.2.1.4:IJ06655m2a.180527.epkg.Z:IJ06655
7.2.1.4:IJ06907s1a.180607.epkg.Z:IJ06907
7.2.1.4:IJ07501m4a.180716.epkg.Z:IJ07501
7.2.1.4:fips_102j.170207.epkg.Z: N/A
7.2.1.4:fips_102m.180105.epkg.Z: N/A
7.2.1.4:fips_102ma.180410.epkg.Z: N/A
7.2.1.4:fips_102oa.180910.epkg.Z: N/A

```

```

# the missingapars.rpt is important. It is listing the
# apars that should be evaluated for deployment on the TNCC.
# You should evaluate each of these apars for relevance to your
# environment and determine when and if you will apply the apar
# to your TNC Clients that need the patch.

```

```

#(missingapars.rpt)
#Missing Apars that are not on the other system
#Apar_Level:Apar_Name:Fileset_Name
7.2.1.4:IJ01423:devices.pciex.df1060e214103404.com 7.2.1.3,
7.2.1.4:IJ01426:devices.common.IBM.xhci.rte 7.2.1.2,devices.common.IBM.usb.rte
7.2.1.1,
7.2.1.4:IV96360:devices.pciex.df1060e214103404.com 7.2.1.3,

```

```

# the clientstatus.rpt is provides the results of the verification. The results
# are detailed in one line that lists the TNCC ip address, the service pack level
# of the TNCC, the service pack level of the fileset policy used to execute the
# verification, the number of apars installed on the TNCC, the number of
# apars not found on the TNCC that should be evaluated for possible install
# on the TNCC, the number of ifixes not found on the TNCC that should be
# evaluated for possible deployment on the TNCC, the number of open source
# packages that are defined in the fileset policy that are missing on the
# endpoint, and lastly the compliance status for the TNCC
#(clientstatus.rpt)
#status info of the other system
#Client_IP:Client_Level:Policy_Level:Client_Apars:Apars:Ifixes:Packages:Status
10.3.126.48:7.2.1.4:7.2.1.4:1445:3:15:0:NON-COMPLIANT

```

```

Transaction elapsed: 2.10 secs, size: 104354 bytes
Transaction succeeded 10.3.126.48
root@lbstnc1>

```

### ***Verifying the operation when a corresponding fileset policy exists***

The TNC server verifies the client by using an existing fileset policy that corresponds to the TNC client.

The command uses the following syntax:

```

psconf mkserver [tnoport=<port>] pmservice=<host:port>
[tsserver=host][recheck_interval=<time in mins> | d(days):h(hours):m(minutes)]
[dbpath=<userdefined directory>] [verify_policy=<auto|manual>]
[clientDataPath=<userdefined directory for clientData>] [clientData_interval=<time
in mins> | d(days):h(hours):m(minutes)]

```

The following output is produced:

```
# The following shows that only lbsaix1 is mapped to a TNC policy, tncpol.
# This TNC policy contains the fileset policy, fspol.
```

```
root@lbstnc1> psconf list -P ALL
#policy          groupname          subpolicy
  tncpol          tncgrp1            fspol
root@lbstnc1> psconf list -G tncgrp1
#ipgroupname      ip                  policyname          EMAILID
EMAILTYPE
  tncgrp1          lbsaix1.aus.stglabs.ibm.com tncpol              ....
....
```

```
# I am now going to verify lbsaix1, which is mapped to a fileset policy, fspol
```

```
root@lbstnc1> psconf verify -G tncgrp1
```

```
Running transaction 10.3.126.48
OSLevel 7200-01-02 exact match FS Policy <fspol>
Running policy checks for 7.2.1.2:7.2.1.2:fspol
Running verification based on policy fspol
```

```
 #(clientstatus.rpt)
#status info of the other system
#Client_IP:Client_Level:Policy_Level:Client_Apars:Apars:Ifixes:Packages:Status
10.3.126.48:7.2.1.2:7.2.1.2:1058:0:0:0:COMPLIANT
```

```
Transaction elapsed: 2.02 secs, size: 86136 bytes
Transaction succeeded 10.3.126.48
# now verifying the client that is not mapped to a TNC policy
root@lbstnc1> psconf verify -i lbstds3
Running transaction 10.3.126.46
```

```
# Even though lbstds3 is not mapped to a TNC Policy it finds that there is a
# fileset policy, fspol,
# that corresponds to the service pack of lbstds3, so it verifies the client
# using this fileset policy
```

```
# policy:
OSLevel 7200-01-02 exact match FS Policy <fspol>
Running policy checks for 7.2.1.2:7.2.1.2:fspol
Running verification based on policy fspol
```

```
 #(clientstatus.rpt)
#status info of the other system
#Client_IP:Client_Level:Policy_Level:Client_Apars:Apars:Ifixes:Packages:Status
10.3.126.46:7.2.1.2:7.2.1.2:1058:0:0:0:COMPLIANT
```

```
Transaction elapsed: 1.32 secs, size: 85659 bytes
Transaction succeeded 10.3.126.46
root@lbstnc1> psconf list -s ALL -i ALL
#ip              Release TL      SP      status          time
trustlevel
10.3.126.48      7200    1        2      COMPLIANT      2018-07-25
15:37:49      ....
  10.3.126.46    7200    1        2      COMPLIANT      2018-07-25
15:38:28      ....
```

```
root@l1bstncl>
```

### ***Verifying the operation by using the default policy***

In the following example, the TNC clients are not assigned to a TNC policy. Therefore, the TNC clients are verified against all ifixes and apars that map to their particular service pack.

The command uses the following syntax:

```
psconf mkserver [tncport=<port>] pmservice=<host:port>
[tsserver=host][recheck_interval=<time in mins> | d(days):h(hours):m(minutes)]
[dbpath=<userdefined directory>] [verify_policy=<auto|manual>]
[clientDataPath=<userdefined directory for clientData>] [clientData_interval=<time
in mins> | d(days):h(hours):m(minutes)]
```

The following output is produced:

```
root@l1bstncl> psconf delete -P tncpol
root@l1bstncl> psconf delete -F fspol
root@l1bstncl> psconf verify -G tncgrpl
Running transaction 10.3.126.48
Running verification based on default policy 7200-01-02

#(instifixes.rpt)
#ifixes that are installed on the other system
#Ifix_Level:Ifix_Name:Apar_Name
7.2.1.2:IV96310m2a.170519.epkg.Z:IV96310
7.2.1.2:IV96310m2a.170519.epkg.Z:IV96310

#(ifixhigher.rpt)
#ifixes with fileset that is at a higher level than maximum
#Ifix_Level:Ifix_Name:Fileset_Name:Ifix_Version:Client_Version
7.2.1.2:517_ifix.170113.epkg.Z:openssl.base:1.0.1.517:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800

#(ifixfilesetnotinst.rpt)
#ifix with filesets that are not installed on the other system
#Ifix_Level:Ifix_Name:Fileset_Name:Fileset_Version
7.2.1.2:IV83983s5a.160602.epkg.Z:ntp.rte:7.1.0.5
7.2.1.2:IV87279s7a.160901.epkg.Z:ntp.rte:7.1.0.7
7.2.1.2:IV92126m3a.170106.epkg.Z:ntp.rte:7.1.0.7
7.2.1.2:IV96312m5a.170518.epkg.Z:ntp.rte:7.1.0.9
7.2.1.2:fips_ifix.170113.epkg.Z:openssl.base:1.0.1.517:1.0.2.800

#(missingifixes.rpt)
#ifixes that are not installed on the other system
#Ifix_Level:Ifix_Name:Apar_Name
```



```

7.2.1.2:102j_ifix.170207.epkg.Z: N/A
7.2.1.2:102m_ifix.180105.epkg.Z: N/A
7.2.1.2:102ma_ifix.180410.epkg.Z: N/A
7.2.1.2:6202_ifix.160830.epkg.Z: N/A
7.2.1.2:6203_ifix.170124.epkg.Z: N/A
7.2.1.2:IJ02828s2a.171221.epkg.Z:IJ02828
7.2.1.2:IJ02919s1a.180108.epkg.Z:IJ02919
7.2.1.2:IJ03035m2a.180118.epkg.Z:IJ03035
7.2.1.2:IJ05820m2a.180430.epkg.Z:IJ05820
7.2.1.2:IJ06907s1a.180607.epkg.Z:IJ06907
7.2.1.2:IV94723m3a.171009.epkg.Z: N/A
7.2.1.2:IV94723s2a.170414.epkg.Z:IV94723
7.2.1.2:IV97811s2a.170712.epkg.Z:IV97811
7.2.1.2:IV97898s2a.171201.epkg.Z:IV97898
7.2.1.2:IV97901s2a.171201.epkg.Z:IV97901
7.2.1.2:IV97958s0b.171205.epkg.Z:IV97958
7.2.1.2:IV98830m1a.170809.epkg.Z:IV98830
7.2.1.2:IV99499m3a.171115.epkg.Z:IV99499
7.2.1.2:IV99552m3a.171031.epkg.Z:IV99552
7.2.1.2:fips_102j.170207.epkg.Z: N/A
7.2.1.2:fips_102m.180105.epkg.Z: N/A
7.2.1.2:fips_102ma.180410.epkg.Z: N/A

#(clientstatus.rpt)
#status info of the other system
#Client_IP:Client_Level:Policy_Level:Client_Apars:Apars:Ifixes:Packages:Status
10.3.126.48:7.2.1.2:7.2.1.2:1058:0:22:0:NON-COMPLIANT

Transaction elapsed: 1.23 secs, size: 86136 bytes
Transaction succeeded 10.3.126.48
root@lbtstnc1> psconf verify -G tncgrp2
Running transaction 10.3.126.46
Running verification based on default policy 7200-01-02

#(instifixes.rpt)
#ifixes that are installed on the other system
#Ifix_Level:Ifix_Name:Apar_Name
7.2.1.2:IV96310m2a.170519.epkg.Z:IV96310
7.2.1.2:IV96310m2a.170519.epkg.Z:IV96310

#(ifixhigher.rpt)
#ifixes with fileset that is at a higher level than maximum
#Ifix_Level:Ifix_Name:Fileset_Name:Ifix_Version:Client_Version
7.2.1.2:517_ifix.170113.epkg.Z:openssl.base:1.0.1.517:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9a.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9b.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800
7.2.1.2:IV83169m9c.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800
7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:1.0.2.500:1.0.2.800
7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:1.0.1.515:1.0.2.800

```

```

7.2.1.2:IV83169s9d.160401.epkg.Z:openssl.base:0.9.8.2506:1.0.2.800

#(ifixfilesetnotinst.rpt)
#ifix with filesets that are not installed on the other system
#Ifix_Level:Ifix_Name:Fileset_Name:Fileset_Version
7.2.1.2:IV83983s5a.160602.epkg.Z:ntp.rte:7.1.0.5
7.2.1.2:IV87279s7a.160901.epkg.Z:ntp.rte:7.1.0.7
7.2.1.2:IV92126m3a.170106.epkg.Z:ntp.rte:7.1.0.7
7.2.1.2:IV96312m5a.170518.epkg.Z:ntp.rte:7.1.0.9
7.2.1.2:fips_ifix.170113.epkg.Z:openssl.base:1.0.1.517:1.0.2.800

#(missingifixes.rpt)
#ifixes that are not installed on the other system
#Ifix_Level:Ifix_Name:Apar_Name
7.2.1.2:102j_ifix.170207.epkg.Z: N/A
7.2.1.2:102m_ifix.180105.epkg.Z: N/A
7.2.1.2:102ma_ifix.180410.epkg.Z: N/A
7.2.1.2:6202_ifix.160830.epkg.Z: N/A
7.2.1.2:6203_ifix.170124.epkg.Z: N/A
7.2.1.2:IJ02828s2a.171221.epkg.Z:IJ02828
7.2.1.2:IJ02919s1a.180108.epkg.Z:IJ02919
7.2.1.2:IJ03035m2a.180118.epkg.Z:IJ03035
7.2.1.2:IJ05820m2a.180430.epkg.Z:IJ05820
7.2.1.2:IJ06907s1a.180607.epkg.Z:IJ06907
7.2.1.2:IV94723m3a.171009.epkg.Z: N/A
7.2.1.2:IV94723s2a.170414.epkg.Z:IV94723
7.2.1.2:IV97811s2a.170712.epkg.Z:IV97811
7.2.1.2:IV97898s2a.171201.epkg.Z:IV97898
7.2.1.2:IV97901s2a.171201.epkg.Z:IV97901
7.2.1.2:IV97958s0b.171205.epkg.Z:IV97958
7.2.1.2:IV98830m1a.170809.epkg.Z:IV98830
7.2.1.2:IV99499m3a.171115.epkg.Z:IV99499
7.2.1.2:IV99552m3a.171031.epkg.Z:IV99552
7.2.1.2:fips_102j.170207.epkg.Z: N/A
7.2.1.2:fips_102m.180105.epkg.Z: N/A
7.2.1.2:fips_102ma.180410.epkg.Z: N/A

#(clientstatus.rpt)
#status info of the other system
#Client_IP:Client_Level:Policy_Level:Client_Apars:Apars:Ifixes:Packages:Status
10.3.126.46:7.2.1.2:7.2.1.2:1058:0:22:0:NON-COMPLIANT

Transaction elapsed: 1.73 secs, size: 85659 bytes
Transaction succeeded 10.3.126.46
root@lbstnc1>
# since both clients don't map to a fileset policy, all ifixes
# were checked, and both TNC clients are marked as non-compliant

root@lbstnc1> psconf list -s ALL -i ALL
#ip                Release TL      SP      status          time
trustlevel
10.3.126.48        7200    1        2        NON-COMPLIANT   2018-07-25
15:40:50    ....
  10.3.126.46        7200    1        2        NON-COMPLIANT   2018-07-25
15:41:11    ....

```

```
root@lbtnc1>
```

## 5.6.2 Viewing the Trusted Network Connect Server logs

The following files are used to view the TNCs logs:

- ▶ `/etc/tncs.conf`: This configuration file lists the attributes of the TNC components that are installed on the local host.
- ▶ `/etc/tncpm.conf`: This file lists the configuration settings for the TNCPM.

## 5.6.3 Viewing the verification results of the TTNCCs

You can query the status of a single system or all systems by using the following command:

```
psconf list { -H | -s <COMPLIANT | IGNORE | FAILED | ALL> } -i <host | ALL> [-c] [-q]
```

The following output is produced:

```
root@lbtnc1> psconf list -s ALL -i ALL
#ip          Release TL    SP    status          time
trustlevel
 10.3.126.46    7200    1      2      COMPLIANT      2018-07-20
01:17:20 ....
 10.3.126.48    7200    1      2      COMPLIANT      2018-07-21
20:30:24 ....
root@lbtnc1>
```

## 5.6.4 Updating the Trusted Network Connect Client

This section describes updating the TNCC.

### Ifix update operation

You can update a single client or an ipgroup at the same time by using the following command:

```
psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a
<apar1, apar2,apargrp1,apargrp2,...> |
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}
```

The following output is produced:

```
root@lbtnc1> psconf update -i lbsaix1 IV96310m2a.170519.epkg.Z
Running transaction
Running policy check
10.3.126.48:7.2.1.2:7.2.1.2:NON-COMPLIANT
Running policy based update
Running ifix install operation
10.3.126.48:7.2.1.2:IV96310m2a.170519.epkg.Z:INSTALL-REQUEST
10.3.126.48:7.2.1.2:IV96310m2a.170519.epkg.Z:INSTALL-SUCCESS
Transaction succeeded
root@lbtnc1>
```

## Ifix preview update operation

You can update a single client or an ipgroup at the same time by using the following command:

```
psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a  
<apar1, apar2,apargrp1,apargrp2,...> |  
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}
```

The following output is produced:

```
root@lbtstnc1> psconf update -p -i lbtstds3 -v IV96310m2a.170519.epkg.Z  
Running transaction  
Running policy check  
10.3.126.46:7.2.1.2:7.2.1.2:NON-COMPLIANT  
Running user defined update  
Running ifix install operation  
10.3.126.46:7.2.1.2:IV96310m2a.170519.epkg.Z:PREVIEW_INSTALL-REQUEST  
10.3.126.46:7.2.1.2:IV96310m2a.170519.epkg.Z:PREVIEW_INSTALL-SUCCESS  
Transaction succeeded  
root@lbtstnc1>
```

## Removing an ifix

You can update a single client or an ipgroup at the same time by using the following command:

```
psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a  
<apar1, apar2,apargrp1,apargrp2,...> |  
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}
```

The following output is produced:

```
root@lbtstnc1> psconf update -i lbtstds3 -u -v IV96310m2a.170519.epkg.Z  
Running transaction  
Running policy check  
10.3.126.46:7.2.1.2:7.2.1.2:NON-COMPLIANT  
Running user defined update  
Running ifix uninstall operation  
10.3.126.46:7.2.1.2:IV96310m2a:UNINSTALL-REQUEST  
10.3.126.46:7.2.1.2:IV96310m2a:UNINSTALL-SUCCESS  
Transaction succeeded  
root@lbtstnc1>
```

## Updating an operation by using ifix group

You can update a single client or an ipgroup at the same time by using the following command:

```
psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a  
<apar1, apar2,apargrp1,apargrp2,...> |  
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}
```

The following output is produced:

```
root@lbtstnc1> psconf update -i lbtstds3 -v ifixgrp1  
Running transaction  
Running policy check  
10.3.126.46:7.2.1.2:7.2.1.2:NON-COMPLIANT  
Running user defined update  
Running ifix install operation
```

```

10.3.126.46:7.2.1.2:IV97811s2a.170712.epkg.Z:INSTALL-REQUEST
10.3.126.46:7.2.1.2:IV97811s2a.170712.epkg.Z:INSTALL-SUCCESS
10.3.126.46:7.2.1.2:IV96310m2a.170519.epkg.Z:INSTALL-REQUEST
10.3.126.46:7.2.1.2:IV96310m2a.170519.epkg.Z:INSTALL-SUCCESS
Transaction succeeded
root@lbstnc1>

```

## Updating apar

You can deploy an apar to a TNCC by using the following command:

```

psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a
<apar1, apar2,apargrp1,apargrp2,...> |
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}

```

The following output is produced:

```
# psconf update -i 10.3.126.48 -a IV60303
```

## Opening a package update

You can update a single client or an ipgroup at the same time by using the following command:

```

psconf update [-p] {-i<host>| -G <ipgroup, ipgroup2,...>[-r <buildinfo> | -a
<apar1, apar2,apargrp1,apargrp2,...> |
[-u] -v<ifix1, ifix2,ifixgrp1,ifixgrp2,...> | -O <openpkggrp1, openkggrp2,...>}

```

The following output is produced:

```

root@lbstnc1> psconf update -i lbsaix1 -O opengrp1
Running transaction
Running policy check
10.3.126.48:7.2.1.4:7.2.1.4:NON-COMPLIANT
Running user defined update
Running open package install operation
10.3.126.48:7.2.1.4:rpm:lsf:4.89-1.aix7.2:INSTALL-REQUEST
10.3.126.48:7.2.1.4:rpm:lsf:4.89-1.aix7.2:INSTALL-SUCCESS
Transaction succeeded
root@lbstnc1>

```

## Updating service packs

In the following example, the TNCC is updated from 7200-01-02 to 7200-01-03. In the output, you see the message INSTALL-FAILURE towards the end of the output. This message appears because the NIM client generated an error code to indicate a restart was required.

To verify that the installation was successful, go to the updated log directory for this TNC Client and view the resulting installation log that is created after the update is completed. This log provides the NIM installation information that resulted from the update operation.

After you verify the operation completed successfully with the installation log, start the system per standard AIX best practice after a service pack update, and reverify the TNC Client after restart as shown in the following example:

```

root@lbstnc1> psconf update -i 10.3.126.48 -r 7200-01-03
Running transaction
Running policy check
10.3.126.48:7.2.1.2:7.2.1.2:NON-COMPLIANT
Running update of system 10.3.126.48 level: 7200-01-02 to release: 7200-01-03
10.3.126.48:7.2.1.2:7.2.1.3:INSTALL-REQUEST

```

```
10.3.126.48:7.2.1.2:7.2.1.3:INSTALL-FAILURE
Nothing to update
Transaction succeeded
root@lbstnc1>
```

After rebooting the system and verifying the TNCC again, the client shows as properly updated to the 7200-01-03 service pack level:

```
root@lbstnc1> psconf list -s ALL -i ALL
#ip          Release TL      SP      status          time
trustlevel
 10.3.126.46      7200    1        2      COMPLIANT      2018-09-15
11:25:54 ....
 10.3.126.48      7200    1        3      COMPLIANT      2018-09-15
18:06:43 ....
root@lbstnc1>
```

### Technology level update

PowerSC 1.2.0.0 introduced the ability to run an update operation that changes the AIX technology level of a client. If your NIM Server supports the technology level update per NIM requirements, TNC can perform the operation.

## 5.6.5 Updating and verifying by using PowerSC GUI 1.2.0.0

PowerSC 1.2.0.0 added support for the TNCS verify and update operations to be issued by using the PowerSC GUI. The 1.2.0.0 release depends on the configuration of TNC components to be done first at the command line. These two functions are run by using the GUI.

Complete the following steps,

1. Activate the subproduct usage feature Up-to-date, as shown in Figure 5-3.

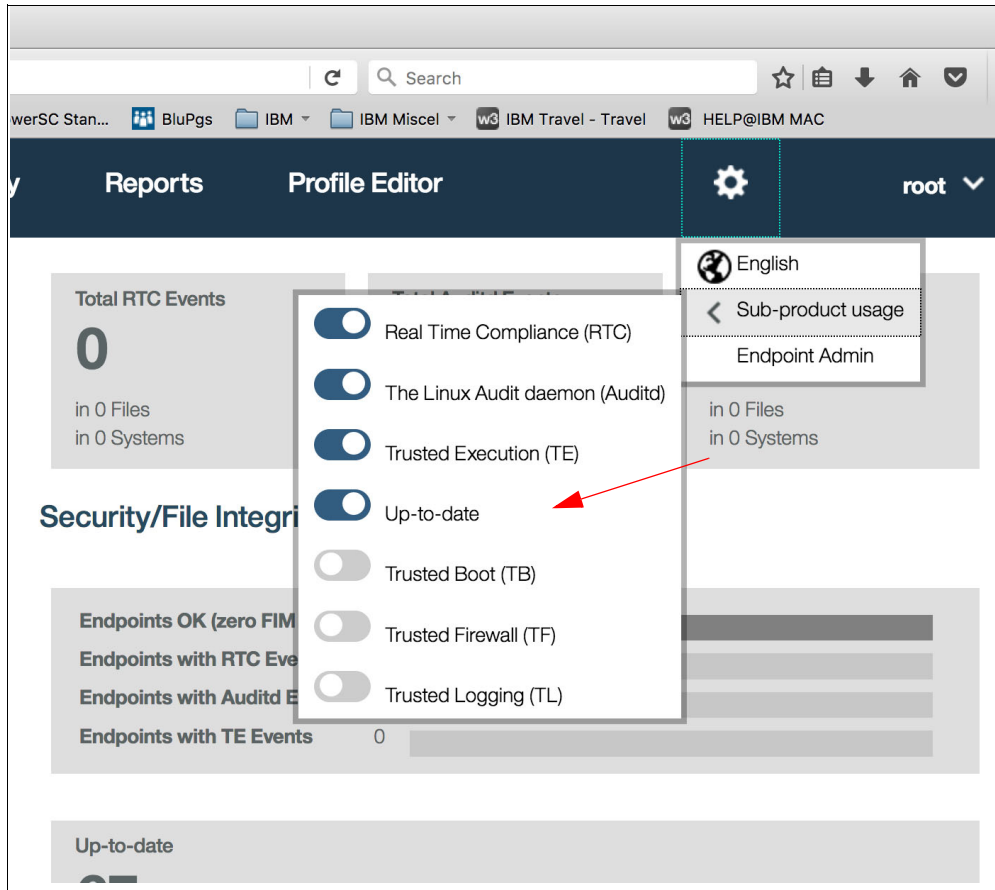


Figure 5-3 Using the PowerSC GUI

2. After the Up-to-date subproduct is activated, the window that is shown in Figure 5-4 opens.

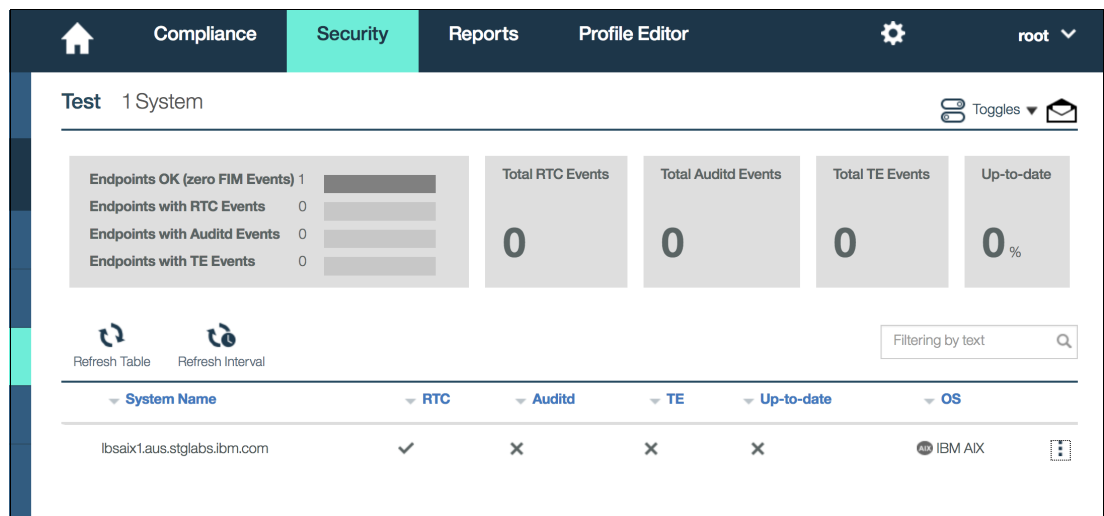


Figure 5-4 PowerSC GUI security pane

3. Complete verify and update operations on the TNC Client by clicking the three dots that are to the far right of the TNC client name (see Figure 5-5).

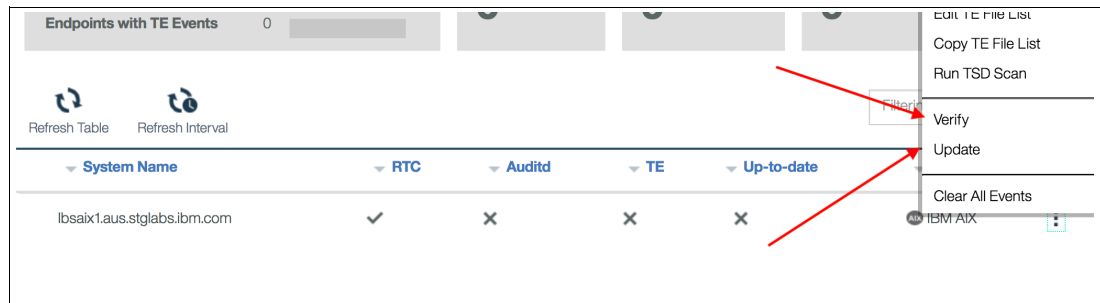


Figure 5-5 Selecting to perform operations on the TNC Client by using the PowerSC GUI

- When an update operation occurs by using the PowerSC GUI, the fileset policy that corresponds to your TNC client is used. For more predictable results, define a fileset policy and specify the precise set of security updates with which you want the client to be updated.
- Issue the Update operation by using the PowerSC GUI.
- After the update operation finishes, issue a Verify operation by using the GUI to get the up-to-date status of the TNC Client.

Figure 5-6 shows the TNC Client is not reported as compliant because one of the patches applied required a restart of the system.

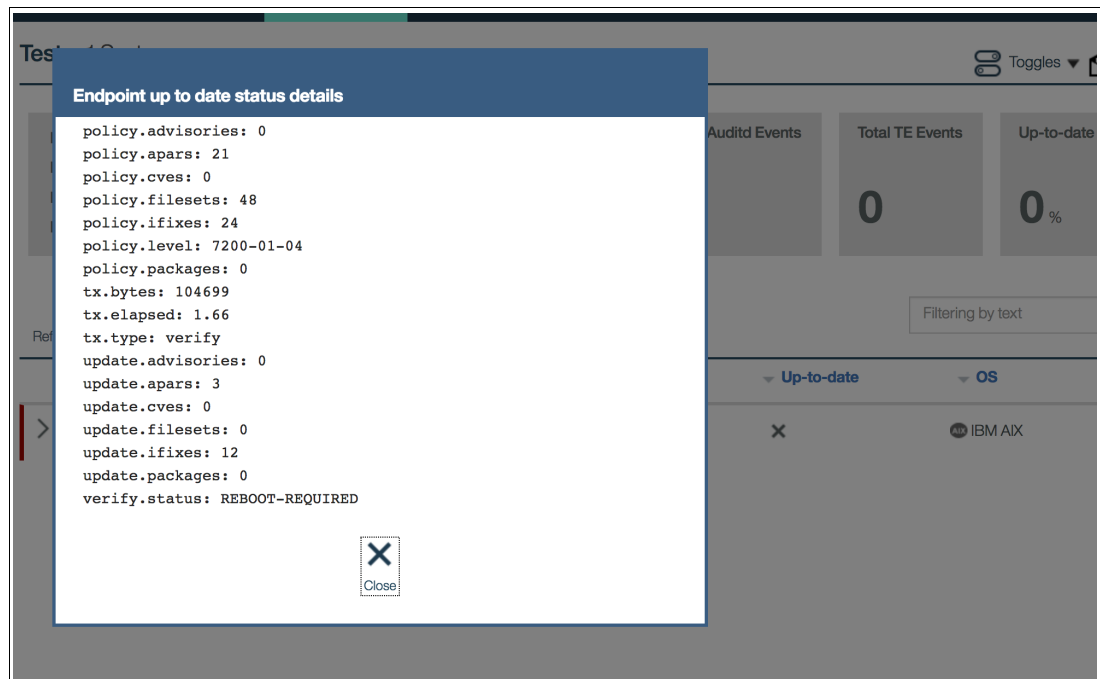


Figure 5-6 PowerSC GUI status details pane shows a required action

- If your TNC client is successfully updated, that status is reflected in the command line interface and the PowerSC GUI (see Figure 5-7).



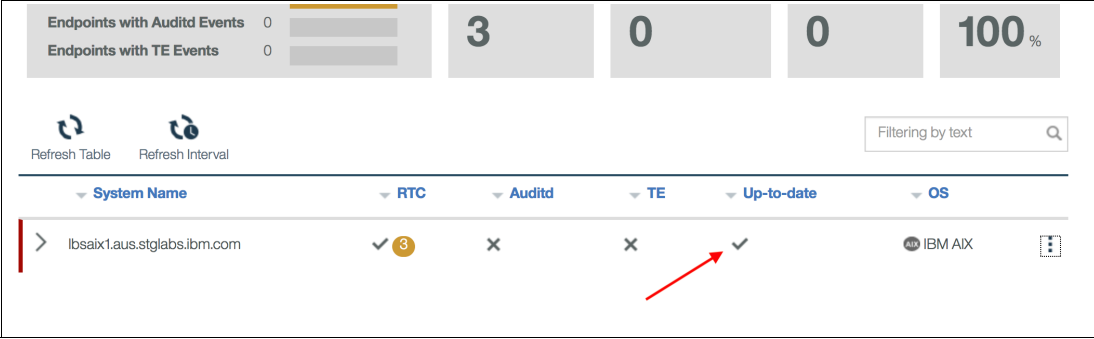


Figure 5-7 PowerSC GUI status details of system showing status compliant

### 5.6.6 New TNC functions provided in PowerSC GUI 1.2.0.1

New options are available in the GUI of 1.2.0.1. One new function is the ability to run the `psconf pu11` operation on a TNCS, as shown in Figure 5-8.

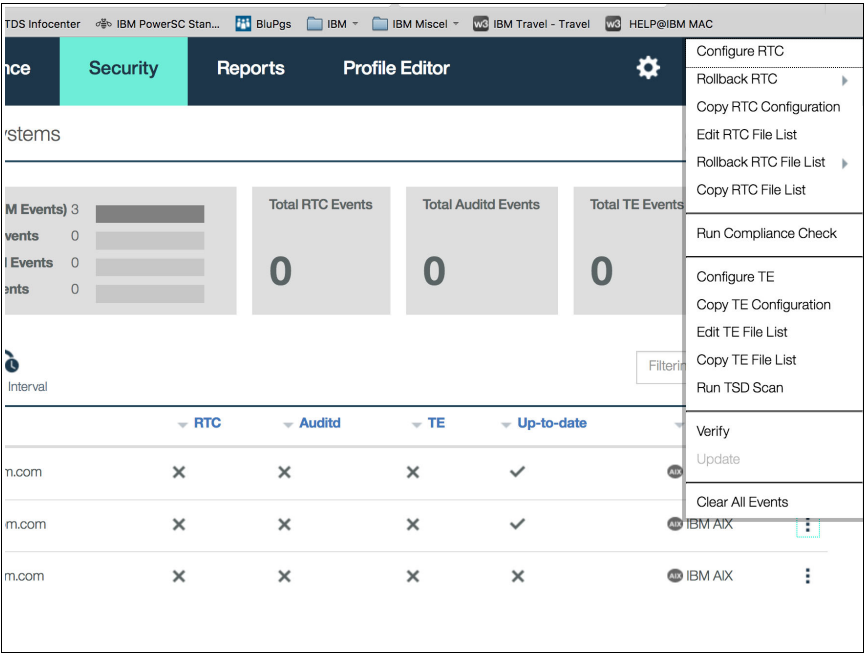


Figure 5-8 PowerSC GUI v1.2.0.1

The second new option updates the TNCC to a higher service pack level or different technology level, as shown in Figure 5-9.

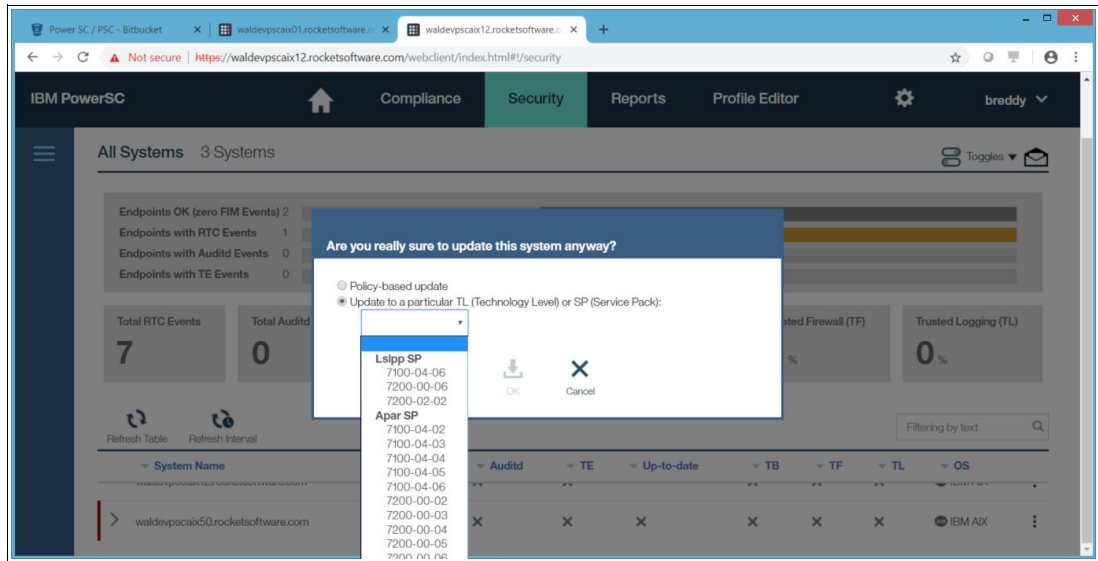


Figure 5-9 PowerSC GUI v1.2.0.1 - updating the TNCC

## 5.6.7 Update logs

After an update operation completes, you can view details of the update operation. The log can be found in the following directory that corresponds to the TNCC that was updated:

`/var/tnc/tncpm/log/update/<TNCC hostname>`

## 5.7 Troubleshooting

This section provides a few troubleshooting techniques.

### 5.7.1 Check syslog

Ensure that you enabled syslog on the TNCC, TNCS, and TNCPM. The \*.debug setting provides the superset of all syslog information. TNC components report to syslog by using user.info.

### 5.7.2 Verify your configuration files

Verify your daemon configuration files do not include an incorrect value. When the TNC daemons start, they use the settings that are defined in the following local files:

```
/etc/tncs.conf
/etc/tncpm.conf
```

### 5.7.3 Update operation fails while AIX Trusted Execution is enabled

The example in this section shows the failure of an update operation when AIX Trusted Execution is enabled:

```
root@lbstnc1> cd /var/tnc/tncpm/log/update/lbsaix1.aus.stglabs.ibm.com
root@lbstnc1> cat 09-15-18:17:35:38
```

IP:10.3.126.48 Name:lbsaix1 Fileset:all lpp\_resource:tncpm\_7200-01-03\_lpp

Installation cannot proceed as Trusted Execution Checks are enabled.

Disable the TE, TEP and TLP policies on system and try installing

0042-001 nim: processing error encountered on "master":

0042-001 m\_cust: processing error encountered on "lbsaix1":

0042-175 c\_script: An unexpected result was returned by the  
"lbstncl.aus.stglabs.ibm.com:/export/nim/scripts/lbsaix1.script" command:

0042-175 c\_installp: An unexpected result was returned by the "/usr/sbin/installp"  
command:

See the log file:

/var/adm/ras/nim.installp

for details or use the "showlog" operation.

### 5.7.4 Refreshing the daemons to correct anomalies

In some circumstances, refreshing the TNC daemons can correct various types of error conditions. Refresh the daemons in the following order to ensure the daemons are properly refreshed:

1. Stop TNCCs.
2. Stop TNCS.
3. Stop TNCMP.
4. Start TNCS.
5. Start TNCCs.
6. Start TNCMP.

### 5.7.5 Enabling TNCS verbose logging

If you want to enable more detail in the logging, activate the following setting on the TNCS:

```
# export TNC_VERBOSE=4
# Psconf stop server
# Psconf start server
```

### 5.7.6 More information

For more information about SUMA, see the following IBM Knowledge Center web pages:

- ▶ <https://ibm.co/2DuG82T>
- ▶ <https://ibm.co/2Mu33OV>
- ▶ <https://ibm.co/2B3Rf1f>





# Trusted Logging

The Trusted Logging component of IBM PowerSC allows a Virtual I/O Server to become a secure repository for log data that is generated by other logical partitions (client LPARs) on the system. This facility is provided in such a way that no network configuration is required and the client LPARs cannot modify or remove any of the data after it is written.

The Virtual I/O Server administrator can provision any number of write-only virtual log devices for a client LPAR, which can be used for any purpose. The content of each of these logs are in a directory in the Virtual I/O Server's file system.

This chapter describes Trusted Logging, and covers the purpose and architecture of the component. It also includes detailed, hands-on examples of installation, configuration, and management tasks.

This chapter contains the following topics:

- ▶ 6.1, "Component architecture" on page 200
- ▶ **6.2, "Deployment considerations" on page 208**
- ▶ 6.3, "Detailed implementation" on page 211
- ▶ 6.4, "Installation" on page 218
- ▶ 6.5, "Working with Trusted Logging" on page 219
- ▶ 6.6, "Troubleshooting" on page 246
- ▶ 6.7, "Conclusion" on page 250

## 6.1 Component architecture

Trusted Logging is implemented partly within the Virtual I/O Server and partly within the client LPARs. Trusted Logging uses virtual SCSI technology to allow the client LPARs to pass log data to the Virtual I/O Server in a secure fashion.

This section describes the architecture of Trusted Logging. We introduce the important concepts that are required to develop an intuitive understanding of what Trusted Logging can do and how it works.

### 6.1.1 Built on virtual SCSI foundations

Trusted Logging passes data from client LPARs to Virtual I/O Servers by using the virtual SCSI infrastructure that existed in PowerVM since its inception. Virtual SCSI provides secure one-to-one conduits between client LPARs and Virtual I/O Servers. It was originally used for the provision of virtual disk, tape, and optical media to client LPARs.

The virtual SCSI channels of communication pass through the PowerVM hypervisor. The channels are reliable (messages cannot get lost) and secure (the traffic is not visible to any LPAR other than the one participating in the specific client LPAR-to-Virtual I/O Server relationship).

The SCSI protocol does not treat the two endpoints of a connection in the same way. In SCSI terminology, the client LPAR is an *Initiator* and the Virtual I/O Server is a *Target*. The SCSI protocol allows Initiators only to read and write data from the Target; the Target cannot make any request of the Initiator. Therefore, it is impossible for the Virtual I/O Server to insert or extract data from the client LPARs; instead, they must explicitly transmit that data to the Virtual I/O Server.

Virtual SCSI connections are provisioned by creating *virtual SCSI client adapters* on the client LPARs and *virtual SCSI server adapters* on the Virtual I/O Servers by using the Hardware Management Console (HMC). Trusted Logging can use virtual SCSI adapters that are in place for the provision of virtual disk, tape, or optical media resources to a client LPAR.

For more information about the configuration of virtual SCSI adapters, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

Figure 6-1 shows a simple virtual SCSI configuration, in which two client LPARs include virtual SCSI client adapters (named vscsi0 on each) that are connected to virtual SCSI server adapters on a single Virtual I/O Server (named vhost0 and vhost1).

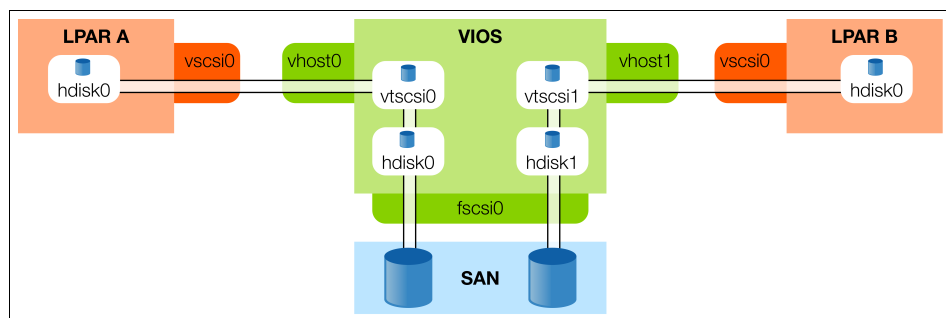


Figure 6-1 Virtual SCSI with two client LPARs and a single Virtual I/O Server

The Virtual I/O Server is presenting two virtual SCSI disks (vtscsi0 and vtscsi1), which are backed by physical disks (hdisk0 and hdisk1) from a Fibre Channel (FC) adapter (fscsi0) that is attached to a storage area network (SAN). One of the disks is presented to each client LPAR, where the disks appear as devices named hdisk0.

A more complex but common deployment model is to use two Virtual I/O Servers, which provides multiple paths (“multipath”) to the same physical resource. This model allows Virtual I/O Servers to be individually upgraded without loss of service to the client LPARs. Figure 6-2 shows a simple multipath configuration that provides the same access from the two SAN-backed disks to client LPARs, but now with redundant data paths.

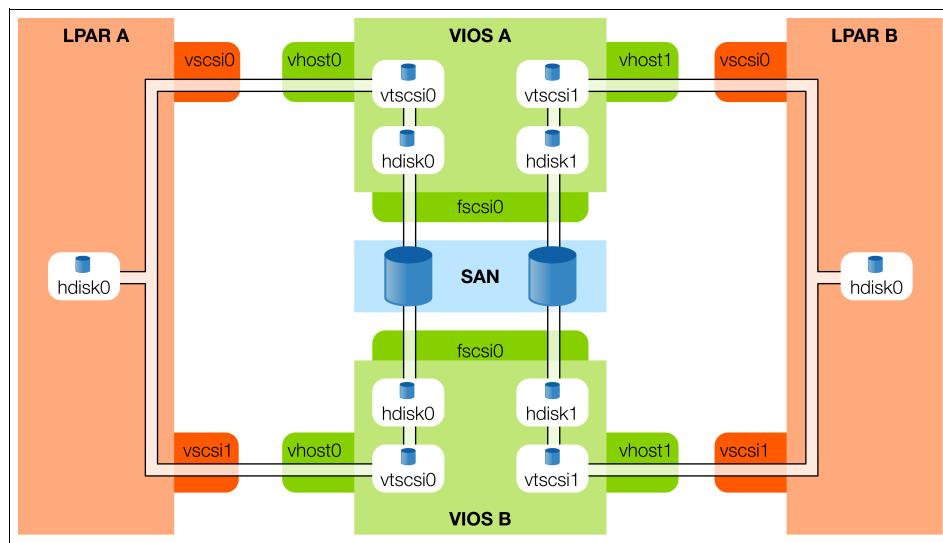


Figure 6-2 Virtual SCSI multipath configuration

**Note:** For Trusted Logging to support multipath configurations, *shared storage pools* must be deployed on the Virtual I/O Servers. For more information about deploying Trusted Logging with shared storage pools, see 6.5.4, “Configuring shared storage pools” on page 222.

Shared storage pools provide a means by which many Virtual I/O Servers (a *cluster*) can coordinate concurrent access to *individual files* on SAN storage. Shared storage pools are the foundation of Trusted Logging’s multipathing support. The combination of Trusted Logging with shared storage pools has advantages beyond multipath. The Virtual I/O Servers in a shared storage pool cluster do not need to all be on a single physical system. Therefore, the following benefits are possible:

- ▶ All log data that is collected by the Virtual I/O Servers in a shared storage pool can be accessed by any Virtual I/O Server in the shared storage pool. Therefore, the number of management touchpoints that are needed to back up and analyze log data across a large hardware estate is reduced.
- ▶ PowerVM Live Partition Mobility can be performed from one physical system to another with no impact to the log data. The log data seamlessly continues to be written to the same log file on the shared storage. PowerVM Live Partition Mobility *can* be performed between Virtual I/O Servers that do not use shared storage pools. However, a new log file is created on the destination Virtual I/O Server. The old log file on the original Virtual I/O Server remains because no shared file system is in place between the Virtual I/O Servers.

## 6.1.2 Virtual Log devices

Trusted Logging builds on virtual SCSI by introducing a new type of virtual SCSI device (the *virtual log device*) which is analogous to the virtual disk device that is familiar to PowerVM users. However, a virtual log device is unusual. Rather than providing random-access read/write capability, as a virtual disk provides, it instead provides a restricted *concatenate-only* capability. Therefore, a client LPAR can provide data to be written to a log file, but it cannot control where in the file that data is written (it is always written at the end). Nor can it retrieve the data after it is written.

As with all other virtual SCSI devices, the following procedure is used for creating a virtual log device and presenting it to a client LPAR:

1. Create a virtual log on the Virtual I/O Server.
2. Attach the virtual log to a virtual SCSI server adapter on the Virtual I/O Server by creating a *virtual log target device*.
3. Detect new virtual SCSI devices on the client LPAR.
4. Verify that the client LPAR detected the virtual log device on its corresponding virtual SCSI client adapter.
5. Configure operating system services to use the newly detected virtual log device.

For more information about this process, see 6.5, “Working with Trusted Logging” on page 219.

## 6.1.3 Virtual logs

This section describes the terminology and concepts that relate to virtual logs, which represent the individual logs that the Trusted Logging component creates and manages.

A virtual log is an entity that is created and managed on the Virtual I/O Server. It is important to understand the difference between the *virtual log* and the *virtual log target device*. The virtual log device also exists on the Virtual I/O Server and is the means by which virtual logs are exposed to client LPARs. Consider the following points:

- ▶ The *virtual log target device* connects a specific virtual log to a specific virtual SCSI server adapter. This relationship is analogous to how a virtual optical target device connects an optical media image to a virtual SCSI server adapter for use by a client LPAR.
- ▶ The *virtual log* represents the log file on the Virtual I/O Server, together with some configuration properties, and it is *not* a device. It is possible to create a virtual log without creating an associated virtual log target device, although the virtual log is not accessible by a client LPAR. Unattached virtual logs can be attached to virtual SCSI server adapters at a later point by creating a virtual log target device.

**Virtual logs:** A virtual log can be connected to at most one virtual log target device at any time, so virtual logs cannot be concurrently shared between several client LPARs.

Because virtual logs are not devices, they cannot be uniquely identified by device names. Instead, a virtual log is assigned a random Universally Unique Identifier (UUID) when it is created. The UUID is a 32-character hexadecimal number, as shown in the following example:

00000000000000005b3f6b7cfcec4c67



This UUID is unique within the Virtual I/O Server, or within the Virtual I/O Server cluster if shared storage pools are used. When a virtual log is moved to another Virtual I/O Server with Live Partition Mobility, the UUID moves with it.

Figure 6-3 shows the relationship between virtual logs, virtual log target devices, virtual log devices, virtual SCSI client adapters, and virtual SCSI server adapters. It shows a virtual log (00000000000000005b3f6b7cfcec4c67) that is presented to client LPAR A as vlog0 by attaching it as a device vtlog0 to the virtual SCSI server adapter vhost0. The virtual SCSI server adapter vhost0 is connected to virtual SCSI client adapter vscsi0 on the client LPAR.

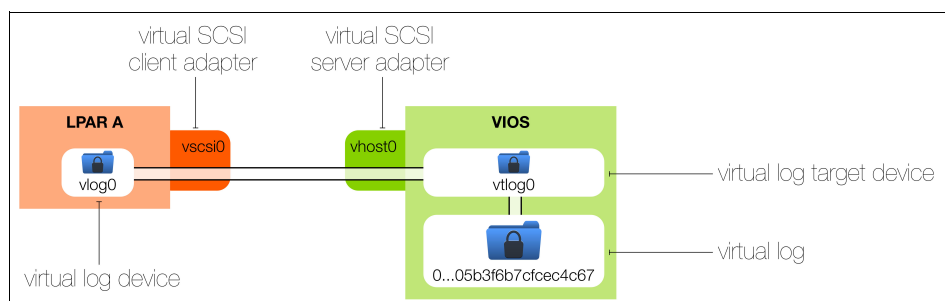


Figure 6-3 Trusted Logging virtual SCSI components

Because managing and tracking the purpose of virtual logs by using only the UUID can be laborious, virtual logs also possess two properties to make management easier: the *client name* and the *log name*.

Both of these properties are modifiable by the Virtual I/O Server administrator. They can be inspected (but not modified) by the client LPAR to which a virtual log is connected. Although the Virtual I/O Server administrator can modify these properties to contain any value that they like, convention is that the properties are used in the following way:

- |                    |   |
|--------------------|---|
| <b>Client name</b> | Indicates the name of the client LPAR to which this log is to be attached. Typically, all virtual logs that are intended for a particular client LPAR are assigned the same client name. If a virtual log is created and attached to a virtual SCSI server adapter in a single operation, the Virtual I/O Server attempts to obtain the host name of the associated client LPAR. The Virtual I/O Server uses that name as the client name if it is not specified on the command line. The client name can be up to 96 characters in length. |
| <b>Log name</b>    | Indicates the purpose of a virtual log. This property can be assigned any value by the Virtual I/O Server administrator and must be provided when a new virtual log is created. For example, you can create two virtual logs named <code>audit</code> (for the collection of audit data) and <code>syslog</code> (for the collection of syslog data) for a certain client LPAR. The log name can be up to 12 characters in length.  |

On the client LPAR, these properties can be inspected by using the `lsattr -El` command. The log name is also used to name the device in the client LPAR's `/dev` file system, as described in 6.3.2, "Virtual log devices" on page 212.

Understanding the purpose of the UUID, client name, and log name is sufficient to start creating virtual logs. A virtual log can be created, and then attached to a virtual SCSI server adapter by using two separate invocations of the `mkvlog` command on the Virtual I/O Server, as shown in Example 6-1 on page 204.

Example 6-1 also shows the Virtual I/O Server commands to create a virtual log with a manually specified client name. The first invocation creates a virtual log with the specified client name and log name. The second invocation attaches the virtual log to the specified virtual SCSI server adapter.

*Example 6-1 Virtual I/O Server commands to create a virtual log (manual client name)*

---

```
$ mkvlog -client LPAR2.01 -name syslog
Virtual log 0000000000000000f8546e995c208cbe created
$ mkvlog -uuid 0000000000000000f8546e995c208cbe -vadapter vhost0
vtlog0 Available
```

---

Alternatively, the virtual log and virtual log target device can be created in a single operation, with the client name automatically assigned, as shown in Example 6-2. Example 6-2 shows the use of the **mkvlog** command to create the virtual log. Then, Example 6-2 shows the use of the **lsvlog** command to display the virtual log properties, which shows the automatic assignment of the client name.

*Example 6-2 Virtual I/O Server commands to create virtual log (automatic client name)*

---

```
$ mkvlog -name syslog -vadapter vhost1
Virtual log 0000000000000000a11af0a9ac388216 created
vtlog1 Available
$ lsvlog -dev vtlog1
```

Client Name	Log Name	UUID	VTD
LPAR2.01	syslog	0000000000000000a11af0a9ac388216	vhost1/vtlog1

---

**Important:** Automatic client name assignment requires that the client LPAR is active and its operating system is fully operational. If these conditions are not met, the command might fail with the following message:

```
mkvlog Error:
      Client LPAR is not accessible for VSCSI adapter vhost1. Use -client
option to specify a client name for the new Virtual Log.
```

## 6.1.4 Virtual log directory and file structure

On the Virtual I/O Server, every virtual log stores its associated log data in its own directory. All of these directories exist in subtrees of a *virtual log repository root directory*. This directory is `/var/vio/vlogs` by default for non-shared virtual logs, but it can be modified by using the **chvlrepo** command. Virtual logs that use shared storage pools have a more complicated directory structure, as described in 6.5.4, “Configuring shared storage pools” on page 222.

Within the virtual log repository root directory, every virtual log exists in a *clientname/logname/* subdirectory. Example 6-3 shows the resulting directory structure when three virtual logs are created with three separate invocations of the **mkvlog** command. (Two virtual logs are created for a client LPAR named `s1` that is attached to `vhost0`. One virtual log is created for another client LPAR named `s2` that is attached to `vhost1`.) A **find** command is run to locate all the directories in the virtual log repository root.

*Example 6-3 Directory structure of the local virtual log repository with three virtual logs*

---

```
$ mkvlog -vadapter vhost0 -name syslog
Virtual log 0000000000000000b224bb0dfb1030bf created
vtlog0 Available
```

---

```

$ mkvlog -vadapter vhost0 -name audit
Virtual log 00000000000000004e42f98eed1c6a02 created
vtlog1 Available
$ mkvlog -vadapter vhost1 -name syslog
Virtual log 0000000000000000fe72d7b80c0394a9 created
vtlog2 Available
$ find /var/vio/vlogs -type d
/var/vio/vlogs
/var/vio/vlogs/config
find: 0652-081 cannot change directory to </var/vio/vlogs/config>:
: The file access permissions do not allow the specified action.
/var/vio/vlogs/s1
/var/vio/vlogs/s1/audit
/var/vio/vlogs/s1/syslog
/var/vio/vlogs/s2
/var/vio/vlogs/s2/syslog

```

---

The following observations are from the output of the **find** command that is shown in Example 6-3 on page 204:

- ▶ A config subdirectory is not accessible from the Virtual I/O Server command line, and it contains no client LPAR-generated data.
- ▶ The virtual logs automatically detected the s1 and s2 client LPAR names by querying the vhost0 and vhost1 virtual SCSI adapters.
- ▶ Each virtual log has a corresponding *clientname/logname/* subdirectory.

Within each of the leaf subdirectories, log files are stored. The following types of data are generated by Trusted Logging:

- ▶ *Log data* is a byte-for-byte copy of the logs, as written by the client LPAR.
- ▶ *State data* consists of informational messages that concern the operation of Trusted Logging. Some of these messages are generated by the client LPAR, and some of these messages are generated by the Virtual I/O Server. For more information about the contents of these files, see 6.3.3, “Messages that are written to the state files” on page 213.

To reduce the possibility of the log files causing the Virtual I/O Server file system to fill up, both of these log types store their data in a series of rotating log files. The number of files and the maximum size of each file are configurable by the Virtual I/O Server administrator.

As an example, consider a virtual log that is configured to use five 2 MB files for log data. Data is initially written to the first file until the next write causes the file size to exceed 2 MB. At this point, a second file is created, and data is written to that new file, and so on. When the maximum file count is reached, the next write causes the first file to be truncated to zero bytes, and writes continue to the newly emptied file.

To configure these settings, the following virtual log properties can be specified when a virtual log is created by using the **mkvlog** command, or modified after the virtual log is created with the **chvlog** command:

<b>Log files</b>	The number of files to use for client-generated log data,
<b>Log file size</b>	The maximum size of each client-generated log data file,
<b>State files</b>	The number of files to use for virtual log state data,
<b>State file size</b>	The maximum size of each virtual log state data file,

The created files are named in the following way:

- ▶ For log files, files are named *clientname\_logname.index*, where *clientname* and *logname* are the properties of the virtual log, and *index* starts at 000 and increments when each new log file is created. For example, a virtual log with client name *host01* and log name *syslog*, which is configured to generate at most three log files, eventually (given enough data from the client LPAR) creates the following set of log files:

```
host01_syslog.000  
host01_syslog.001  
host01_syslog.002
```

- ▶ For state files, files are named *clientname\_logname.state.index*, where *clientname* and *logname* are the properties of the virtual log, and *index* starts at 000 and increments when each new log file is created. For example, a virtual log with client name *host01* and log name *syslog*, which is configured to generate at most three state files, eventually is expected to create the following set of state files:

```
host01_syslog.state.000  
host01_syslog.state.001  
host01_syslog.state.002
```

## 6.1.5 Virtual log repositories

Every virtual log is stored in a *virtual log repository*. One virtual log repository exists on each Virtual I/O Server for non-shared virtual logs. Another virtual log repository exists if shared storage pools are configured on the Virtual I/O Server.

A virtual log repository is responsible for the following tasks:

- ▶ Defining default values for the following properties of new virtual logs:
  - Log files
  - Log file size
  - State files
  - State file size
- ▶ Enabling or disabling the use of virtual logs at a repository-wide level
- ▶ For the local virtual log repository, defining the directory in to which virtual logs and their configuration are placed

Virtual log repository properties are modified by using the **chv1repo** command.

**Changing the repository root or state:** If virtual logs exist, you cannot perform the following tasks:

- ▶ Change the repository root directory.
- ▶ Change the state of the repository from enabled to disabled.

Therefore, it is important that the Virtual I/O Server administrator selects the repository root directory early in the deployment process. For more information, see **6.2, “Deployment considerations” on page 208**.

## 6.1.6 Shared storage pools

Shared storage pools are the means by which Trusted Logging allows a cluster of Virtual I/O Servers to concurrently access log file data that is generated by virtual log devices.

The original use of shared storage pools was to allow a single SAN-backed disk to be split into several smaller volumes and presented to client LPARs as virtual disks with advanced functions, such as thin provisioning, seamless Live Partition Mobility, and cluster-wide management.

However, for Trusted Logging, the important function that is provided by shared storage pools is the provision of a cluster-wide file system. This cluster-wide file system is stored on the SAN and accessible from every Virtual I/O Server in the cluster. By storing virtual log data in the shared storage pool file system, multiple paths to the same log file can be provided, which facilitates multipath support for virtual logs and seamless Live Partition Mobility.

The construction of a shared storage pool cluster requires that all member Virtual I/O Servers can access the same set of Fibre Channel SAN disks and communicate with each other by way of TCP/IP.

Figure 6-4 shows the architecture of a Trusted Logging deployment that uses shared storage pools. In this example, two Virtual I/O Servers form a shared storage pool by way of Fibre Channel and Ethernet connectivity, presenting a common shared file system to both Virtual I/O Servers. Virtual logs that are created within this shared file system are visible to both Virtual I/O Servers and therefore, can be presented to a client LPAR in a multipath arrangement.

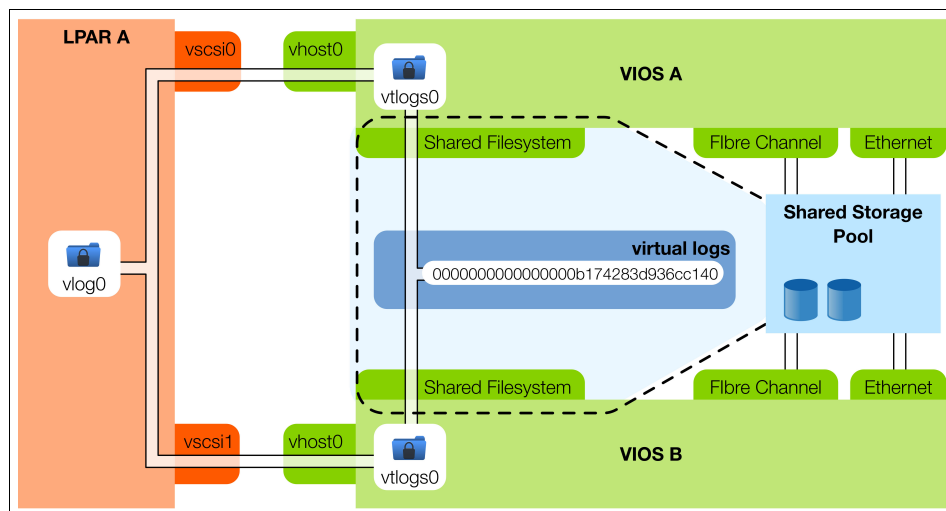


Figure 6-4 Trusted Logging that uses shared storage pools for multipath virtual logs

For more information about the use of Trusted Logging with shared storage pools, see the following sections:

- ▶ 6.5.2, “Creating a virtual log on a single Virtual I/O Server” on page 220
- ▶ “Creating a single-path virtual log in a shared storage pool” on page 224
- ▶ “Creating a multipath virtual log using shared storage pools” on page 225

## 6.2 Deployment considerations

This section describes and guides the decisions that must be made before Trusted Logging is deployed. When you consider the deployment of Trusted Logging to your virtualized infrastructure, you must understand what decisions are difficult to change after virtual logs are created and running. You also need to know what changes are easy and nondisruptive for the users of Trusted Logging.

The following configuration decisions are difficult to reverse:

- ▶ Whether to use local virtual logs or virtual logs in a shared storage pool
- ▶ If you use local virtual logs, the location in the file system of the log data

In both cases, the virtual log configurations cannot be preserved. Also, the virtual logs must be deleted and then new virtual logs must be created.

The following virtual log properties can be modified at the same time the virtual logs are attached to virtual SCSI server adapters and performing I/O, without any noticeable effect on client LPARs. However, because these properties define the directory into which the logs are written, the log and state files effectively become “split”. The old data is preserved in the old location, and new log and state messages are written to the new location. You can modify the following virtual log properties without affecting the client LPARs:

- ▶ Client name
- ▶ Log name

The following virtual log maximum number and maximum size properties can be modified during the time the virtual logs are attached to virtual SCSI server adapters and performing I/O, without any noticeable effect on client LPARs:

- ▶ Log files
- ▶ State files

Next, we guide you through the major decisions that are required when you plan your Trusted Logging deployment.

### 6.2.1 Deploying Trusted Logging on a dedicated Virtual I/O Server

Although you can deploy Trusted Logging on the Virtual I/O Servers, deploying it on a separate Virtual I/O Server results in the following security advantages:

- ▶ You can specify a password for the `padmin` user that differs from other Virtual I/O Servers. With this separate password, you effectively can restrict access to the contents of the virtual logs to a small group of people. You can select a subset of the people who might need access to Virtual I/O Servers for day-to-day maintenance operations.
- ▶ When you use Trusted Logging to create local virtual logs (as opposed to virtual logs in shared storage pools), no network connectivity is required on the Virtual I/O Server because communication from the client LPAR to the Virtual I/O Server uses the virtual SCSI infrastructure. Provisioning a dedicated Trusted Logging Virtual I/O Server with no network adapters and therefore access only by way of the HMC console removes the need to configure and maintain satisfactorily secure network connectivity.

However, this approach makes it difficult to extract the virtual log data regularly. Therefore, this approach is better-suited to deployments in which logs must be retained securely for audit purposes, but not regularly analyzed or moved to another location. With no network connectivity, backup and disaster recovery can be performed by using SAN-backed disks and the mirroring and snapshot capabilities of the SAN controller.

Therefore, the decision to deploy a dedicated Virtual I/O Server for Trusted Logging is essentially a tradeoff. You choose the convenience of a single management point and network-based backup options or the security of restricted passwords and no required network.

## 6.2.2 Securing the Virtual I/O Server

If Trusted Logging is deployed on a Virtual I/O Server with configured network adapters, be careful to ensure that it is sufficiently protected from intrusion. Use the **viosecur**e Virtual I/O Server command, which can be used to automatically apply firewall rules and improved security settings.

A full discussion of **viosecur**e is outside the scope of this chapter. For more information, see the following sections of the Virtual I/O Server documentation:

- ▶ Configuring Virtual I/O Server firewall settings:  
<https://ibm.co/2Z0UzUu>
- ▶ Configuring Virtual I/O Server system security hardening:  
<https://ibm.co/318nA2q>

**Note:** Do not use **viosecur**e when shared storage pools are active. The use of a firewall can disrupt the operation of a shared storage pool cluster.

## 6.2.3 Local virtual logs or shared storage pools

As described in 6.1.6, “Shared storage pools” on page 207, shared storage pools provide a means by which a cluster of Virtual I/O Servers can share a common file system into which virtual log data can be placed.

After virtual logs are created, they cannot be migrated easily from the local virtual log repository to a shared storage pool, or vice versa. Therefore, it is important to decide early in your deployment planning whether you intend to use shared storage pools.

Virtual logs in shared storage pools offer the following advantages:

- ▶ Multiple paths to a single virtual log can be provided to the client LPAR, which makes the virtual log infrastructure tolerant to loss of service from a particular Virtual I/O Server.
- ▶ PowerVM Live Partition Mobility can be performed between Virtual I/O Servers in the shared storage pool cluster at the same time maintaining a single virtual log file.
- ▶ Every Virtual I/O Server in the cluster can see every virtual log. Therefore, the examination and backup of the data is easier because they can be performed for all virtual logs in a cluster from a single Virtual I/O Server.

However, a shared storage pool deployment of Trusted Logging also has the following requirements and behaviors; therefore, it might not be suitable for your environment:

- ▶ The time to complete a write to a virtual log device on the client LPAR is approximately doubled if the virtual log is in a shared storage pool, when compared with a local virtual log that is stored on the same disk infrastructure. For more information, see 6.3.6, “Performance” on page 215.
- ▶ The shared storage pool data must be on SAN-backed disks, which must be accessible from all Virtual I/O Servers in the cluster.

- TCP/IP networking is required between all Virtual I/O Servers in the cluster, which limits your ability to provide secure Virtual I/O Servers with no network adapters, as discussed in 6.2.1, “Deploying Trusted Logging on a dedicated Virtual I/O Server” on page 208.

If your infrastructure is capable, and performance is acceptable, the use of shared storage pools might make sense because of the additional function that it provides. However, in deployments where the performance of log writes is paramount, or the Virtual I/O Servers must be segregated from the network, shared storage pools might not be the best choice.

## 6.2.4 Where to store local virtual logs

By default, the local virtual log repository places its log data in the `/var/vio/vlogs` directory. You can modify this location by using the procedure that is described in 6.5.1, “Changing the local virtual log repository file system” on page 219. However, because this procedure cannot be performed after virtual logs are created, it must be considered as part of your deployment planning.

When you decide where to place the local virtual log repository, the following useful possibilities are available:

- Keep the local virtual log repository in `/var/vio/vlogs`
- Move the local virtual log repository to a dedicated file system

If you use virtual logs in shared storage pools, the decision of where to place the log data is not yours. The logs are stored on the SAN-backed shared storage pool disk that is specified when the Virtual I/O Server cluster is created. However, you still need to read the remainder of this section. The factors that are discussed can help to define how your SAN disks are configured.

Consider the following key factors when you decide where your virtual log data must be stored:

- Disk space

Although an individual virtual log can be bound by the disk space that it can use (see 6.1.5, “Virtual log repositories” on page 206), an overall cap does not exist on the total amount of space that can be used by the virtual log repository as a whole. The repository can contain hundreds or thousands of virtual logs.

Therefore, it is important to ensure that the virtual log repository is not responsible for filling up the `/var` file system, which can result in the loss of availability of other Virtual I/O Server functions. The easiest way to avoid this issue is to create a dedicated file system for virtual logs, which ensures that if that file system fills up, other services are not affected.

- Performance

For the best performance, virtual logs must be written to dedicated physical disks so that other I/O operations do not affect write latency. The best performance might be achieved by using a SAN-backed disk for the virtual log repository. The RAID level (the way that data is striped and mirrored across an array of physical disks) also affects the performance that can be obtained.

- Resilience

You might have availability requirements for your virtual log data beyond the requirements for the rest of your Virtual I/O Server file system. For example, you might require your virtual log data to use a more fault-tolerant RAID level, or to be synchronously or asynchronously mirrored to another site by using the built-in capabilities of your SAN controller.



These capabilities are typically provided at a per-disk granularity. Therefore, implementing a specific policy for virtual log data is likely requires that a separate disk with the appropriate qualities is presented to the Virtual I/O Server, and a separate file system must be deployed within it.

Except for small test environments, it is highly likely that the default location for your virtual log data in `/var/vio/vlogs` is not suitable because of some or all of the factors that are described in this section. The deployment planning process must consider the quality of service (performance and resilience) that is required of these disks, and the total amount of required space.

## 6.3 Detailed implementation

PowerSC Trusted Logging is deployed by creating virtual logs and associated virtual log target devices on a Virtual I/O Server, attaching these devices to a virtual SCSI server adapter, and detecting the resulting virtual log device on the corresponding virtual SCSI client adapter on the client LPAR. After it is created, writes to these devices from within the client LPAR are transmitted to the Virtual I/O Server. The writes are written to a series of rotating log files in a dedicated directory structure.

This section describes some of the useful implementation details of these subcomponents.

### 6.3.1 Virtual log target devices

The virtual log target devices are created on the Virtual I/O Server and are analogous to virtual disk, optical, or tape target devices.

By default, these devices are named `vtlogn` for local virtual logs, and `vtlogsn` for shared storage pool virtual logs, where *n* is a number that is unique to the specific device, starting with 0. They are child devices of a virtual SCSI server adapter.

Example 6-4 shows the expected output from the `lsmap` command for a virtual SCSI server adapter with one virtual log, one file-backed disk, and one optical media device attached.

Example 6-4 Using `lsmap` to view virtual log target devices on the Virtual I/O Server

\$ lsmap -vadapter vhost0		
SVSA	Physloc	Client Partition ID
-----		
vhost0	U8205.E6C.06A22ER-V1-C13	0x00000003
VTD	vtlog0	
Status	Available	
LUN	0x8300000000000000	
Backing device	vlog:000000000000000075f175982f4d10d9	
Physloc		
Mirrored	N/A	
VTD	vtopt0	
Status	Available	
LUN	0x8200000000000000	
Backing device	cd0	
Physloc	U78AA.001.WZSHN02-P2-D9	
Mirrored	N/A	

VTD	vtscsi0
Status	Available
LUN	0x8100000000000000
Backing device	/var/vio/storagepools/fbpool/lp1fb1
Physloc	
Mirrored	N/A

---

The Backing device field of the virtual log target device (vtlog0 in Example 6-4 on page 211) corresponds to the UUID of the virtual log.

Each virtual log target device runs as its own kernel process, so its resource usage can be monitored with standard commands, such as **ps**, **topas**, and **nmon**.

### 6.3.2 Virtual log devices

The virtual log devices are created on the client LPAR when it detects a new virtual log on one of its virtual SCSI client adapters by using the **cfgmgr** command.

By default, these devices are named **vlog<sub>n</sub>**, where *n* is a number unique to the particular device, starting with **vlog0**. They are child devices of a virtual SCSI client adapter. Example 6-5 shows the expected **lsdev** output on a client LPAR with two virtual logs present.

*Example 6-5 Using the lsdev command to view virtual log devices on the client LPAR*

---

```
$ lsdev -t vlog
vlog0 Available Virtual Log
vlog1 Available Virtual Log
```

---

Each device appears in **/dev** as two equivalent files, with identical major and minor numbers. One of the files is named per the device name. The other file incorporates the name of the log, as specified when the virtual log is created on the Virtual I/O Server. Example 6-6 shows the expected representation of two virtual logs that are named **syslog** and **audit** in the client LPAR's **/dev** file system.

*Example 6-6 Contents of /dev with two virtual logs named "syslog" and "audit"*

---

```
$ ls -l /dev/vl*
crw----- 1 root system 37, 0 Sep 13 05:32 /dev/vlsyslog0
crw----- 1 root system 37, 1 Sep 13 05:32 /dev/vlaudit1
crw----- 1 root system 37, 0 Sep 13 05:32 /dev/vlog0
crw----- 1 root system 37, 1 Sep 13 05:32 /dev/vlog1
```

---

Detailed properties of the virtual log can be inspected from within the client LPAR by using the **lsattr -El** command, as shown in Example 6-7.

*Example 6-7 Detailed virtual log information by using the lsattr command on the client LPAR*

---

```
$ lsattr -El vlog0
PCM                                Path Control Module                False
UUID      0000000000000000b174283d936cc140 Unique id for virtual log device    False
client_name  sl                      Client Name                        False
device_name  vlsyslog0                       Device Name                        False
log_name     syslog                          Log Name                          False
max_log_size 2097152                         Maximum Size of Log Data File      False
max_state_size 2097152                     Maximum Size of Log State File     False
```

---

pvid	none	Physical Volume Identifier	False
------	------	----------------------------	-------

It is common to want to see a simple mapping of the virtual log device name to the virtual log's log name, which is specified on the Virtual I/O Server. Combine the `lsdev`, `lsattr`, and `xargs` commands to create a one-line command. This command produces a summary of the log name that is associated with each virtual log device, as shown in Example 6-8.

*Example 6-8 Displaying the mapping from device name to log name*

---

```
$ lsdev -Fname -tvlog | xargs -L1 lsattr -Ea log_name -F"name value" -l
vlog0 audit
vlog1 syslog
```

---

To write to these virtual log devices, open the appropriate file in `/dev` and perform a write to it. Virtual log devices can be used in place of log files for many applications, including `syslog`, which is described in 6.5.7, “Configuring syslog to use a virtual log” on page 232.

**Important:** Only one process on the client LPAR can have a specific virtual log device open at any time. If a second process attempts to open the device, an error is returned, which can manifest on a command line as the following message:

```
$ echo "Test" > /dev/vlog0
The requested resource is busy.
ksh: /dev/vlog0: 0403-005 Cannot create the specified file.
```

Each write to the virtual log device is atomically written to the appropriate set of log files on the Virtual I/O Server; no single write is ever truncated or split across multiple log files. If the write cannot be performed on the Virtual I/O Server because of insufficient disk space or the deletion of the corresponding virtual log target device, the write on the client LPAR fails. It returns the `ENOSPACE` error code if the Virtual I/O Server disk fills up, or error code `EIO` for all other errors.

The largest log message that can be written is 32 KB.

### 6.3.3 Messages that are written to the state files

As described in 6.1.4, “Virtual log directory and file structure” on page 204, *state files* collect log data that is generated by the Trusted Logging components, alongside the *log files* that store the log data that is generated by the client LPAR. This section describes each of the messages that can be written to the state files, and explains their purpose.

All state messages are of the following form:

```
[timestamp] [Virtual I/O Server hostname] message...
```

The first field is the POSIX timestamp at the point that the state messages are written, according to the Virtual I/O Server. The second field is the hostname of the Virtual I/O Server that generated the message, which is useful when you analyze behavior in multipath and live migration-capable environments.

The following messages are shown in their abstract form, followed by an example and a description of what the message represents:

- *virtual log target device* initialized
 

```
[1336172435] [vios1] vtlog0 initialized
```

This message is generated when the virtual log target device on the Virtual I/O Server is initialized, which occurs at start time and when the device is initially created (by using the **mkvlog** command).

- ▶ *virtual log target device shutting down*

```
[1336172435] [vios1] vtlog0 shutting down
```

This message is generated when the virtual log target device is shut down by using the **rmvlog** command on the Virtual I/O Server.

- ▶ *virtual log target device by using path:*

```
[1336172435] [vios1] vtlog0  
using /var/vio/vlogs/powerscl/testlog//powerscl_testlog.state.000
```

```
[1336172555] [vios1] vtlog0 using  
/var/vio/vlogs/powerscl/testlog//powerscl_testlog.000
```

These messages are generated whenever a virtual log target device writes to a log file for the first time. When a device is first initialized, messages are written to the state file. However, the log file is not created until the first message is received from the client LPAR.

When a state or log file reaches its size limit and the next file in the sequence is created, a new message is emitted. Example 6-9 shows a typical sequence of messages for a virtual log that is configured to use three log files.

*Example 6-9 State messages for a virtual log with three log files*

---

```
[1347535902] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.000  
[1347535910] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.001  
[1347535918] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.002  
[1347535925] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.000  
[1347535933] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.001  
[1347535940] [vios1] vtlog0 using /vlogs/s1/logA//s1_logA.002
```

---

- ▶ *virtual log target device reconfigured*

```
[1336172435] [vios1] vtlog0 reconfigured
```

These messages are generated whenever the configuration of a virtual log is changed while virtual log is connected to a virtual log target device. The changes (for example, to the log name or the file sizes) take effect immediately by informing the Virtual I/O Server kernel driver of the new configuration. This message is generated when that new configuration takes effect.

- ▶ *Client process **pid** (**name**) by using path **index** (**devno=***major,minor*;**lua=***lua*)*

```
[1336172555] [vios1] Client process 7012592 (ksh) using path 0  
(devno=17,0;lua=0x8200000000000000)
```

When a process on the client LPAR opens a virtual log device, the client LPAR's virtual log device driver emits this message. It shows the process ID and name of the process that opened the virtual log device, and details of the virtual SCSI client adapter device that is now used.

The *index* is an internal identifier that shows which of the available virtual SCSI client adapters is used. This index is always 0 in single path configurations.

The *devno* field indicates the major and minor version number of the virtual SCSI client adapter (not of the virtual log device).

The *lua* field uniquely identifies the virtual log device among other virtual SCSI devices on the same virtual SCSI client adapter.

- Closed by client process *pid (name)*

[1336172555] [vios1] Closed by client process 7012592 (ksh)

This message is emitted when the process on the client LPAR that previously opened a virtual log device closes it.

- Client switch from path *index(devno=major,minor,lua=lua)* to path *index(devno=major,minor,lua=lua)*

When multiple paths are configured for a virtual log (see “Creating a multipath virtual log using shared storage pools” on page 225), and the virtual log device driver experiences an I/O error when writing to the current path, it searches for an alternative path. It searches by attempting to send a message of this form. After one of these messages is successfully transferred, the path is used for all future messages that originate from that virtual log device.

### 6.3.4 Multipath presentation on the client LPAR

Multipath virtual logs require shared storage pools to be configured on the Virtual I/O Server. For more information about an architectural overview, see 6.1.6, “Shared storage pools” on page 207. For more information about a hands-on example, see 6.5.4, “Configuring shared storage pools” on page 222.

The client LPARs identify multiple routes to the same virtual log by looking for virtual logs with the same UUID and presenting those different paths as a single virtual log device.

Virtual logs use a multipathing algorithm that is much simpler than the multipathing algorithms in use for virtual disks. All writes to the virtual log device are sent down the first detected path. If an I/O error occurs when the write is performed, the virtual log device attempts to send the same message down the other paths until it finds a path that does not return an error. It then continues to use that new path for all log traffic.

Although the `lspath` command shows the Virtual SCSI client adapter providing the multiple routes to a certain virtual log, paths with failed I/O are not explicitly marked offline. A health check interval does not need to be configured to ensure that paths with failed I/O are tried again.

For an example of the multipath failover capability in practice, see 6.5.5, “Demonstrating multipath failover” on page 227.

### 6.3.5 Workload partitions

Workload partitions (WPARs) are virtualized operating system environments within a single instance of the AIX operating system. WPARs secure and isolate the environment for the processes and signals that are used by enterprise applications. For more information about the definitive WPAR reference, see this IBM Knowledge Center [web page](#).

Trusted Logging does not support exporting virtual log devices to WPARs on the client LPAR.

### 6.3.6 Performance

The performance of Trusted Logging is highly dependent on your precise deployment environment. However, it is possible to make some general observations about the relative performance of Trusted Logging in various configurations.

First, it is important to understand that writes are processed synchronously for any specific virtual log. The write to the virtual log device on the client LPAR does not return control to the generating application until it receives acknowledgment from the Virtual I/O Server that the message is successfully written to disk. Each log message must take the following path:

1. The write is performed by the client LPAR.
2. The data is transmitted through the virtual SCSI infrastructure.
3. The Virtual I/O Server receives the data and writes it to disk.
4. The disk commits the write and reports successful completion to the Virtual I/O Server.
5. The Virtual I/O Server reports successful completion to the client LPAR.

This process guarantees that the messages are received and written on the Virtual I/O Server in the correct order. Any failed write (for example, because of a lack of disk space on the Virtual I/O Server) can return an error message to the application.

Messages that are written into different virtual logs are not ordered with respect to each other. Messages can be written to several different virtual logs concurrently.

Because of the implementation of the Virtual SCSI infrastructure, the write time observed for small (for example, single byte) writes is almost identical to the write time for 4 KB writes. Beyond 4 KB, writes take longer because more data is transferred to the Virtual I/O Server and written to disk.

The processing of virtual log messages uses Virtual I/O Server CPU resources and I/O operations on the disk that contains the virtual log. Use the **nmon** command, which is accessible from the **oem\_setup\_env** shell, to monitor these properties of the Virtual I/O Server for any bottlenecks that might affect Trusted Logging performance.

The use of the **nmon** command collects and displays metrics from throughout the system on a single page. For Trusted Logging, review the following metrics:

- ▶ CPU utilization
- ▶ Disk utilization
- ▶ Disk adapter utilization

When started with the NMON=twDa environment variable set, **nmon** displays this information automatically. Figure 6-5 shows a **nmon** display that is captured from a Virtual I/O Server with six virtual logs active.

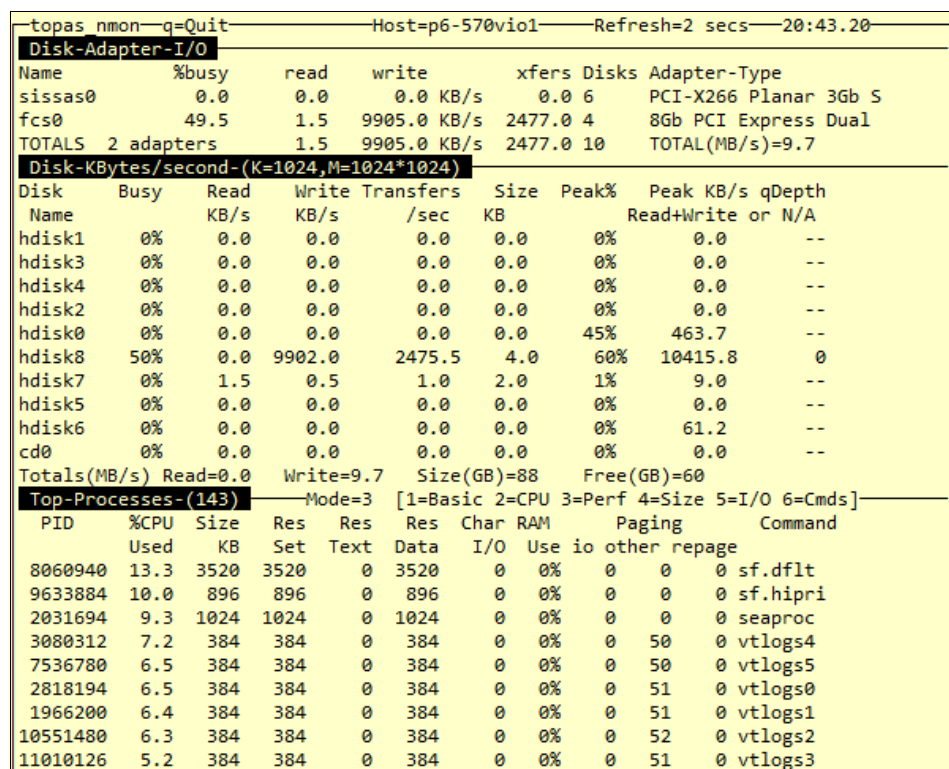


Figure 6-5 The nmon display when started by using the NMON=twda nmon command

If a specific disk is busy and a likely bottleneck to virtual log performance results, the following options are available that might improve performance:

- Move to a SAN-backed volume if your Virtual I/O Server uses internal disks for its file systems.
- Move the virtual log repository to a dedicated file system on a dedicated disk. This procedure is described in 6.5.1, “Changing the local virtual log repository file system” on page 219. However, this procedure works on empty virtual log repositories only. You must remove all your virtual logs and re-create them after the virtual log repository is moved. For this reason, the location of your virtual log repository is a key deployment consideration, as discussed in 6.2.4, “Where to store local virtual logs” on page 210.

If the performance bottleneck appears to be the CPU, CPUs can be added to the Virtual I/O Server. However, remember that the work of each virtual log target device is serial in nature, and each virtual log therefore cannot use more than a single thread of execution.

It is expected that assigning more processors to the Virtual I/O Server realizes the most benefit in environments in which many virtual logs are deployed. This environment provides the most opportunity for parallel work that can use those extra processors.

## 6.4 Installation

Trusted Logging requires that a device driver from the PowerSC Standard installation media is installed on all the client LPARs to which virtual logs are available. It also requires Virtual I/O Server version 2.2.1.0 (and later), which includes the Trusted Logging commands by default.

Therefore, no action is required to enable Virtual I/O Servers for Trusted Logging. Installation of the client LPAR virtual log device is described in 6.4.1, “Installing the Client LPAR component” on page 218. For more information about verifying the version of your Virtual I/O Servers, see 6.4.2, “Verifying the version of the Virtual I/O Server” on page 219.

### 6.4.1 Installing the Client LPAR component

The Trusted Logging component is `powerscStd.vlog` on the PowerSC Standard installation CD. This package must be installed on all client LPARs, but it is not necessary to install it on the Virtual I/O Servers.

Example 6-10 shows the required commands to install the Trusted Logging virtual log device driver with the PowerSC Standard installation media that is mounted in `/cdrom`. The first invocation of `installp` displays the license agreement; the second invocation accepts the license agreement and performs the installation. It is also possible to install Trusted Logging by using the `smitty installp` menu-based interface by selecting the `powerscStd.vlog` package.

---

*Example 6-10 Installation of Trusted Logging device drivers on the client LPAR*

---

```
> installp -aEgd /cdrom powerscStd.vlog.rte
> installp -agXYd /cdrom powerscStd.vlog.rte
```

---

Example 6-11 shows how the `ls1pp` command can be used to verify that Trusted Logging is correctly installed on the client LPAR.

---

*Example 6-11 Verifying Trusted Logging installation on the client LPAR*

---

```
> ls1pp -i powerscStd.vlog.rte
      Vendor
Fileset      Code   Product Id  Feature Id  Package Name
-----
Path: /usr/lib/objrepos
  powerscStd.vlog.rte 1.1.2.0
                        5765-PSE00                powerscStd.vlog

Path: /etc/objrepos
  powerscStd.vlog.rte 1.1.2.0
                        5765-PSE00                powerscStd.vlog
```

---



## 6.4.2 Verifying the version of the Virtual I/O Server

The Virtual I/O Server component of Trusted Logging is available and installed by default on Virtual I/O Server versions 2.2.1.0 and later. The Virtual I/O Server version can be verified by using the **ioslevel** command on the Virtual I/O Server command line, as shown in Example 6-12.

*Example 6-12 Verifying Trusted Logging capability on the Virtual I/O Server*

---

```
$ ioslevel
2.2.1.4
```

---

## 6.5 Working with Trusted Logging

This section provides hands-on examples of common Trusted Logging operations.

### 6.5.1 Changing the local virtual log repository file system

By default, the local virtual log repository is configured to store log data in `/var/vio/vlogs`, which is contained within the file system that is mounted in `/var`. Instead, you might want to provide a file system solely to store virtual log data. With this file system, you can separate disk space allocation and backup policies between the virtual logs and the other data that is stored in `/var`.

**Important:** Changing the virtual log repository file system requires that virtual logs are not present in the virtual log repository.

To change the virtual log repository file system, complete the following steps:

1. In the setup environment (by using the **oem\_setup\_env** command), create the file system by using the **crfs** command. Ensure that it is configured to be remounted on startup. Use the **mount** command to ensure that the file system is available for use. Use the **chmod** command to make the directory group-writable. Example 6-13 shows the required commands to create a 2 GB file system that is mounted as `/vlogs`.

*Example 6-13 Creation of a file system for the local virtual log repository*

---

```
$ oem_setup_env
# crfs -g rootvg -m /vlogs -v jfs2 -A yes -p rw -a size=2G
File system created successfully.
2096884 kilobytes total disk space.
New File System size is 4194304
# mount /vlogs
# chmod g+rwX /vlogs
# exit
```

---

2. Reconfigure the local virtual log repository to store virtual logs in that new location. Example 6-14 shows how the **chv1repo** command is used to update the configuration to store virtual logs in the newly created **/vlogs** file system.

*Example 6-14 Changing the path to the local virtual log repository*

---

```
$ chv1repo -root /vlogs
Updated repository.
```

---

Virtual logs now store their log data in the **/vlogs** file system tree, as described in 6.1.4, “Virtual log directory and file structure” on page 204.

## 6.5.2 Creating a virtual log on a single Virtual I/O Server

Creating a virtual log is straightforward. As described in 6.1.3, “Virtual logs” on page 202, each virtual log includes a log name and a client name. Each virtual log features properties that control the size and number of log and state files. Each virtual log also can be attached to a Virtual SCSI server adapter by using a virtual log target device.

The simplest way to create a virtual log is to specify only the log name and the Virtual SCSI server adapter to which it connects. This task produces a new virtual log, which obtains its client name by querying the client LPAR. It inherits the default log and state file sizes and numbers that are specified in the local virtual log repository.

A new virtual log target device is created, which connects the new virtual log to the specified Virtual SCSI server adapter and to a client LPAR. Example 6-15 shows the use of **mkvlog** with the **-vadapter** option to specify Virtual SCSI server adapter **vhost0** and the **-name** option to specify **syslog** as the log name.

*Example 6-15 Simple creation of a virtual log using mkvlog*

---

```
$ mkvlog -vadapter vhost0 -name syslog
Virtual log 00000000000000000506d8da7fa28d7fe created
vtlog0 Available
```

---

It is also possible to override the local virtual log repository default properties by using more command-line arguments. Example 6-16 shows that invocations of **mkvlog** can include the following information:

- ▶ The **-lf** option to specify the number of log files
- ▶ The **-lfs** options to specify the size of each of those log files
- ▶ The **-sf** option to specify the number of state files
- ▶ The **-sfs** options to specify the size of each of those state files

*Example 6-16 Creating a virtual log using mkvlog and overriding default properties*

---

```
$ mkvlog -vadapter vhost1 -name audit -lf 10 -lfs 10M -sf 4 -sfs 200K
Virtual log 000000000000000005661f7f13dea7100 created
vtlog1 Available
```

---

## 6.5.3 Accessing virtual log data on the Virtual I/O Server

As described in 6.1.4, “Virtual log directory and file structure” on page 204, each virtual log stores its log and state files in a directory within the virtual log repository. To identify the directory in which a particular virtual log stores its data, use the **lsvlog -detail** command.

Example 6-17 shows the **lsvlog** command to display information for the virtual log that is associated with virtual log target device **vtlog0**. The Log Directory field shows the full path to the files of the virtual log.

*Example 6-17 lsvlog -detail that is used to locate the Log Directory*

---

```
$ lsvlog -detail -dev vtlog0
```

```
Client Name: s2
```

```
Log Name:                syslog
UUID:                   00000000000000001ef302550a66eed
Virtual Target Device:   vtlog0
Parent Adapter:         vhost1
Vlog State:             enabled
Device Status:          available
Logical Unit Address:    8200000000000000
Storage Pool:
Log Directory:          /var/vio/vlogs/s2/syslog/
Maximum Log Files:      1
Maximum Log File Size:  104857600
Maximum State Files:    2
Maximum State File Size: 1048576
```

---

The contents of the directory can be viewed by using the **ls** command, as shown in Example 6-18, where **ls -l** is used to display detailed information for each file. In this example, a single log file that is named **s2\_syslog.000** and a single state file that is named **s2\_syslog.000** are included.

*Example 6-18 Viewing files associated with a virtual log using the ls -l command*

---

```
$ ls -l /var/vio/vlogs/s2/syslog/
```

```
total 16
```

```
-rw-r----- 1 root staff      4 Sep 19 03:07 s2_syslog.000
-rw-r----- 1 root staff 378 Sep 19 03:07 s2_syslog.state.000
```

---

## Plain text log files

If the log files contain plain text, the contents of the files are best viewed by using the **cat** and **tail** commands. Example 6-19 shows the **tail** command that is used with the **-n** option to display the specified number of lines from the end of the file.

*Example 6-19 Using the tail command to view the most recent ten entries in a virtual log*

---

```
$ tail -n4 /var/vio/vlogs/s2/syslog/s2_syslog.000
```

```
Sep 21 08:05:40 s2 auth|security:info sshd[9371764]: Failed password
    for root from 172.16.254.6 port 64104 ssh2
Sep 21 08:05:40 s2 auth|security:info syslog: ssh: failed login attempt
    for root from 172.16.254.6
Sep 21 08:05:41 s2 auth|security:info sshd[9371764]: Failed password
    for root from 172.16.254.6 port 64104 ssh2
Sep 21 08:05:41 s2 auth|security:info syslog: ssh: failed login attempt
    for root from 172.16.254.6
```

---

## Log files that contain audit records

If the contents of the log file are generated by the AIX auditing subsystem (for more information, see 6.5.6, “Configuring AIX auditing to use a virtual log” on page 230), the log records are stored in a binary format. The log records cannot be viewed by using commands, such as **ls** and **cat**. Instead, the **auditpr** command must be used.

**Important:** You must use the **-r** option with the **auditpr** command.

By default, the **auditpr** command converts user and group IDs in the audit records into user and group names. However, it uses the user and group files on the Virtual I/O Server to perform this mapping. This mapping is invalid because the records are generated on a client LPAR with potentially different users and groups. The **-r** command option suppresses this conversion.

Example 6-20 shows the use of the **auditpr -r** command to examine the contents of a virtual log that is generated by the AIX auditing subsystem on a client LPAR.

*Example 6-20 Viewing AIX audit records from a client LPAR on the Virtual I/O Server*

```
$ auditpr -r -i /var/vio/vlogs/s2/audit/s2_audit.000
```

event	login	status	time	command	wpar name
FS_Chdir	0	OK	Thu Sep 20 05:46:17 2012	ps	Global
FS_Chdir	0	OK	Thu Sep 20 05:47:09 2012	bash	Global
FS_Mkdir	0	FAIL	Thu Sep 20 05:47:13 2012	java	Global
FILE_Unlink	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FILE_Rename	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FS_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FS_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
S_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FILE_Unlink	0	OK	Thu Sep 20 05:47:15 2012	rm	Global
FS_Chdir	0	OK	Thu Sep 20 05:47:17 2012	ps	Global

## 6.5.4 Configuring shared storage pools

If shared storage pools are configured on the Virtual I/O Server, no further action is needed and virtual logs can be created with the **mkvlog** command. Use the **-sp storage\_pool\_name** command option to indicate that the virtual log needs to be created in a shared storage pool. For more information about examples of how to create virtual logs in shared storage pools, see the following sections:

- ▶ “Creating a single-path virtual log in a shared storage pool” on page 224
- ▶ “Creating a multipath virtual log using shared storage pools” on page 225

For more information about configuring shared storage pools, see section 2.6 in *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

For convenience, the remainder this section summarizes the steps that must be taken to configure a shared storage pool. Shared storage pools provide a shared file system that is accessible from a cluster of up to four Virtual I/O Servers. They use shared disks on a SAN to share data. They communicate by way of a TCP/IP network to coordinate file system operations and ensure that all Virtual I/O Servers see a consistent representation of the file system.

Shared storage pools have the following prerequisites:

- *At least two SAN-backed disks are required.* One disk serves as the *repository disk*, which stores metadata regarding the state of the cluster. At least one other disk stores the user data (in our case, virtual logs). Both of these disks must be at least 10 GB.
- *All Virtual I/O Servers in the cluster must communicate by way of TCP/IP.* Host name resolution (Domain Name System [DNS] or host file-based) must be set up between members of the cluster.

Figure 6-6 shows how a cluster of two Virtual I/O Servers can be formed. Also shown is the hardware configuration and device names.

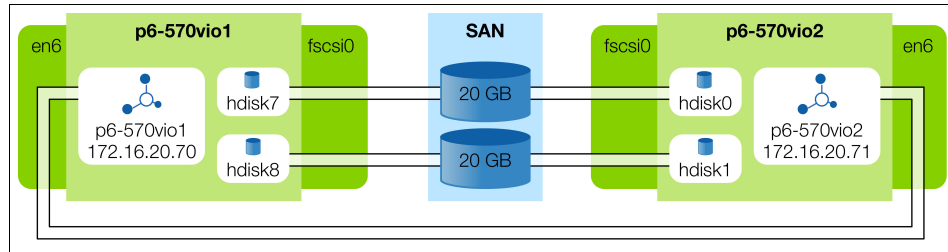


Figure 6-6 A hardware configuration that is suitable for shared storage pools

Configuration of shared storage pools between these two Virtual I/O Servers can be formed as shown in Example 6-21. Both Virtual I/O Servers must first have their Fibre Channel (FC) adapters configured correctly by using two invocations of the **chdev** command. Then, concurrent access to the disks must be enabled by setting the reserve policy attribute of each disk, also by using the **chdev** command.

After the devices are configured, the Virtual I/O Server cluster can be created by running the **cluster -create** command on either of the Virtual I/O Servers. Then, run the **cluster -addnode** command on the same Virtual I/O Server to add the second Virtual I/O Server to the cluster.

Example 6-21 Command sequence to create shared storage pool

```
p5-570vio1  $ chdev -dev fscsi0 -attr dyntrk=yes -perm
             fscsi0 changed
             $ chdev -dev fscsi0 -attr fc_err_recov=fast_fail -perm
             fscsi0 changed
             $ chdev -dev hdisk7 -attr reserve_policy=no_reserve
             hdisk7 changed
             $ chdev -dev hdisk8 -attr reserve_policy=no_reserve
             hdisk8 changed

p5-570vio2  $ chdev -dev fscsi0 -attr dyntrk=yes -perm
             fscsi0 changed
             $ chdev -dev fscsi0 -attr fc_err_recov=fast_fail -perm
             fscsi0 changed
             $ chdev -dev hdisk8 -attr reserve_policy=no_reserve
             hdisk8 changed
             $ chdev -dev hdisk9 -attr reserve_policy=no_reserve
             hdisk9 changed
             $ cluster -create -clustername vlog_cluster -repopvs
             hdisk8 -spname vlog_ssp -sppvs hdisk9 -hostname
             p6-570vio2
             Cluster vlog_cluster has been created successfully.
```

```
$ cluster -addnode -clustername vlog_cluster -hostname
p6-570vio1
```

Partition p6-570vio1 has been added to the vlog\_cluster cluster.

After these commands are run, a shared storage pool cluster is formed. Verify this cluster with the **cluster -status** command, which shows the members of the cluster and by using the **lsvlrepo** command. The **lsvlrepo** command shows a second virtual log repository within the newly created shared storage pool, as shown in Example 6-22.

*Example 6-22 Verifying cluster creation with the cluster -status and lsvlrepo commands*

```
$ cluster -status -clustername vlog_cluster
```

Cluster Name	State
vlog_cluster	OK

Node Name	MTM	Partition Num	State	Pool State
p6-570vio2	9117-MMA02101F170	2	OK	OK
p6-570vio1	9117-MMA02101F170	1	OK	OK

```
$ lsvlrepo
```

Storage Pool	State	Path
	enabled	/var/vio/vlogs
vlog_ssp	enabled	/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/

## Creating a single-path virtual log in a shared storage pool

Creating a single-path virtual log in a shared storage pool first requires that shared storage pools are configured on the Virtual I/O Server on which the virtual log is required.

To create a virtual log in a shared storage pool, the **mkvlog** command is used with the **-sp** option to specify the shared storage pool into which to place the virtual log. Example 6-23 shows the **mkvlog** command that is used to create a virtual log that is named **syslog** in the **vlogssp** shared storage pool that connects that virtual log to the **vhost0** virtual SCSI server adapter at the same time.

*Example 6-23 Creating a virtual log in a shared storage pool and examining its properties*

```
$ mkvlog -sp vlog_ssp -name syslog -vadapter vhost0
```

```
Virtual log 99b977dec96860fba65ab60766a56c11 created
vtlogs0 Available
```

```
$ lsvlog -detail -u 99b977dec96860fba65ab60766a56c11
```

```
Client Name: s1
```

Log Name:	syslog
UUID:	99b977dec96860fba65ab60766a56c11
Virtual Target Device:	vtlogs0
Parent Adapter:	vhost0
Vlog State:	enabled
Device Status:	available
Logical Unit Address:	8100000000000000
Storage Pool:	vlog_ssp
Log Directory:	/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/s1/syslog/
Maximum Log Files:	2
Maximum Log File Size:	1048576
Maximum State Files:	2

Maximum State File Size: 1048576

---

The **lsvlog -detail** command can be used to verify that the virtual log was created. This command can also be used to identify the file system path from which the virtual log's state and log files can be accessed.

After the new virtual log device is created, the client LPAR administrator detects the new virtual log device by running the **cfgmgr** command.

Example 6-24 shows the detection of the new virtual log on the client LPAR by using the **lsdev** command to list the virtual log device and the **lsattr** command to display its properties.

*Example 6-24 Detection of a shared storage pool virtual log on the client LPAR*

---

```
$ cfgmgr
$ lsdev -t vlog
vlog0 Available Virtual Log
$ lsattr -El vlog0
PCM                                Path Control Module                False
UUID          99b977dec96860fba65ab60766a56c11 Unique id for virtual log device False
client_name    s1                      Client Name                        False
device_name    vlsyslog                  Device Name                        False
log_name       syslog                  Log Name                          False
max_log_size   2097152                          Maximum Size of Log Data File     False
max_state_size 2097152                          Maximum Size of Log State File    False
pvid           none                          Physical Volume Identifier        False
```

---

## Creating a multipath virtual log using shared storage pools

The process for creating a multipath virtual log is identical to the process that used to create a single-path virtual log as described in “Creating a single-path virtual log in a shared storage pool” on page 224. The exception is that this process uses another step that is performed on the Virtual I/O Server that provides the second path to the virtual log from the client. The existence of multiple paths to the virtual log can also be verified from the client LPAR by using the **lspath** command.

In the following example, two Virtual I/O Servers (p6-570vio1 and p6-570-vio2) exist in a shared storage pool cluster. Both Virtual I/O Servers have a virtual SCSI server adapter (vhost0 on both Virtual I/O Servers) that provides virtual SCSI connectivity to a client LPAR (that has two virtual SCSI client adapters, vscsi0 and vscsi1) named client1. The objective is to create a virtual log in the shared storage pool and present it to the client LPAR on both paths.

Figure 6-7 on page 226 shows the topology of the system and the paths to the virtual log to be created. The virtual log target devices on both Virtual I/O Servers are called vtlogs0.

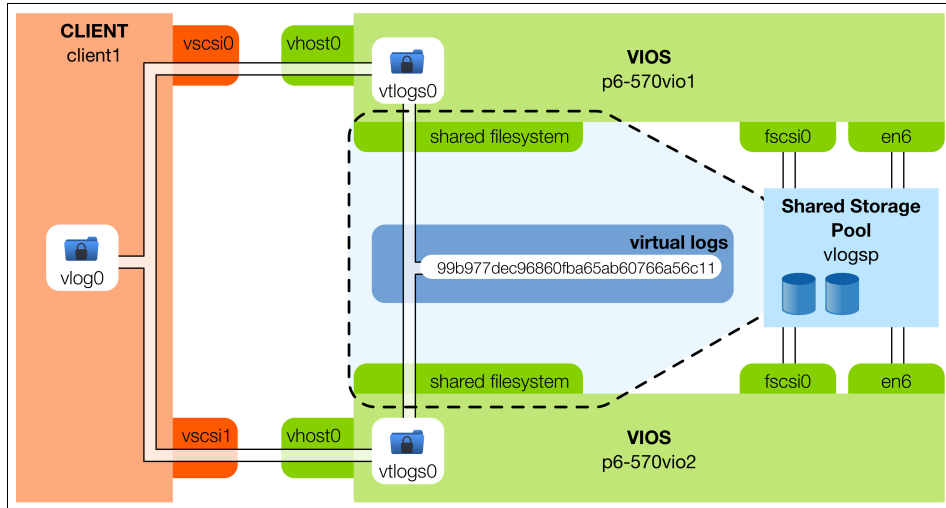


Figure 6-7 Topology of the system for which a multipath virtual log is created

To create a virtual log in a shared storage pool, the **mkvlog** command is used with the **-sp** option to specify the shared storage pool into which to place the virtual log.

Because virtual logs in shared storage pools are visible to all Virtual I/O Servers in the shared storage pool cluster, use the following process to establish a multipath configuration:

1. Create the virtual log on p6-570vio1 and attach it to vhost0.
2. On p6-570vio2, attach the virtual log to vhost0.
3. On the client LPAR, detect new devices and verify that paths to the virtual log device are detected on vscsi0 and vscsi1.

The remainder of this section discusses these steps in more detail.

Example 6-25 shows the **mkvlog** command that is used to create a virtual log that is named **syslog** in the **vlogsp** shared storage pool, which connects that virtual log to the **vhost0** virtual SCSI server adapter at the same time.

*Example 6-25 Creating a virtual log in a shared storage pool on p6-570vio1*

```
$ mkvlog -sp vlogsp -name syslog -vadapter vhost0
Virtual log 99b977dec96860fba65ab60766a56c11 created
vtlogs0 Available
```

Because the virtual log is created in a shared storage pool, it is immediately accessible from the second Virtual I/O Server, p6-570vio2. Example 6-26 shows how the **lsvlog** command can be used to confirm that the virtual log is accessible.

*Example 6-26 Confirming the virtual log and establishing a second path from p6-570vio2*

```
# lsvlog
Client Name  Log Name  UUID                                VTD
client1      syslog    99b977dec96860fba65ab60766a56c11
$ mkvlog -u 99b977dec96860fba65ab60766a56c11 -vadapter vhost0
vtlogs0 Available
```



The VTD field shows that the virtual log is not connected. The field shows only the virtual log target devices on the local Virtual I/O Server, not virtual log target devices that are present elsewhere in the cluster. The **mkvlog** command can then be used to connect this virtual log to the vhost0 virtual SCSI server adapter, which establishes a second path.

After the virtual log device is created, the client LPAR administrator detects the new virtual log device by running the **cfgmgr** command. Example 6-27 shows the detection of the new virtual log on the client LPAR. Example 6-27 also shows the use of the **lsdev** command to list the virtual log device and the **lspath** command to display the virtual SCSI client adapters that provide a path to the virtual log.

---

*Example 6-27 Detection of a multipath shared storage pool virtual log on the client LPAR*

---

```
$ cfgmgr
$ lsdev -t vlog
vlog0 Available Virtual Log
$ lspath -l vlog0
Available vlog0 vscsi1
Available vlog0 vscsi0
```

---

The multipath virtual log is now established, and either of the Virtual I/O Servers can now be deactivated without affecting the availability of the virtual log to the client LPAR. For more information about an example of how multipath failover can be simulated and how the virtual log tracks the change of the path, see 6.5.5, “Demonstrating multipath failover” on page 227.

## 6.5.5 Demonstrating multipath failover

This section describes the failover capability of virtual logs when you use multipathing with shared storage pools. It uses as a basis a virtual log that is established by using the procedure that is described in “Creating a multipath virtual log using shared storage pools” on page 225. This procedure establishes a virtual log on a client LPAR by way of two Virtual I/O Servers: p6-570vio1 and p6-570vio2.

The demonstration is performed by examining the state file of the virtual log, which shows changes in path activity. We then use the **rmvlog** command to disable the current path, which shows failover to the second path. The first path is then reactivated by using the **mkvlog** command, and the second path is disabled, which shows failover back to the first path. During this demonstration, it is assumed that the client LPAR is generating a continuous stream of log messages to its virtual log device.

### Locating the state file

The state file can be viewed on any Virtual I/O Server in the shared storage pool cluster. Locate the directory that contains the state file by using the **lsvlog -detail** command, which can be passed a specific UUID to display data for that virtual log only. Example 6-28 shows the **lsvlog** command that is used to locate the virtual log’s directory. Example 6-28 also shows the **ls -l** command that lists the contents of the directory, and the **cat** command that shows the contents of the state file.

---

*Example 6-28 Locating and examining a shared storage pool virtual log’s state file*

---

```
$ lsvlog -detail -uuid 99b977dec96860fba65ab60766a56c11
Client Name: client1

Log Name:                syslog
UUID:                   99b977dec96860fba65ab60766a56c11
Virtual Target Device:   vtlogs0
```

```

Parent Adapter:          vhost0
Vlog State:              enabled
Device Status:           available
Logical Unit Address:    8200000000000000
Storage Pool:            vlogsp
Log Directory:
/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog/
Maximum Log Files:       2
Maximum Log File Size:   1048576
Maximum State Files:     2
Maximum State File Size: 1048576

$ ls -l
  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog/
total 83768
-rw-r----- 1 root staff          5704 Sep 17 05:02 s1_logA.state.000

$ cat
/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog/s1_logA.state.
000
[1347874203] [p6-570vio1] vtlogs0 using
  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
  client1_syslog.state.000
[1347874203] [p6-570vio1] vtlogs0 initialised
[1347874243] [p6-570vio2] vtlogs0 using
  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
  client1_syslog.state.000
[1347874243] [p6-570vio2] vtlogs0 initialised
[1347874605] [p6-570vio1] Client process 15597740 (syslogd) using path
  0 (devno=17,0;lua=0x8200000000000000)
[1347874605] [p6-570vio1] vtlogs0 using
  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
  client1_syslog.000

```

---

The contents of the state file show that the virtual log device is created and that both Virtual I/O Servers have a device that uses it. The client LPAR has the device open with the syslogd process. The client LPAR uses the path by way of the device with (major,minor) number 17,0, which corresponds to vscsi0 and is confirmed by the Virtual I/O Server p6-570vio1 that emits a message to indicate that it is writing to the log data file.

### Failing over to an alternative path

We now disable the virtual log device on p6-570vio1 and then re-enable it. This process is sufficient to cause the virtual log device on the client LPAR to fail over to an alternative path. We can then reexamine the state file to see what changed.

Example 6-29 shows the **rmvlog** and **mkvlog** commands that are used on p6-560vio1 to disable and re-enable the virtual log target device. The extra state file messages are then examined. They show that p6-570vio2 opened the log file and the client LPAR switched to the new path.

*Example 6-29 Removing the active path and confirming failover to an alternative path*

---

```

$ rmvlog -dev vtlogs0
vtlogs0 Defined

```

```

$ mkvlog -dev vtlogs0
vtlogs0 Available
$ cat
/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog/s1_logA.state.
000
(Lines from last example removed)
[1347875100] [p6-570vio1] vtlogs0 shutting down
[1347875099] [p6-570vio2] vtlogs0 using
    /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
    client1_syslog.000
[1347875099] [p6-570vio2] Client switched from path 0
    (devno=17,0;lua=0x8200000000000000) to path 1
    (devno=17,1;lua=0x8100000000000000)
[1347875101] [p6-570vio1] vtlogs0 using
    /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
    client1_syslog.state.000
[1347875101] [p6-570vio1] vtlogs0 initialised

```

---

As shown, p6-570vio2 opened the log file when the client switched to using device 17,1, which corresponds to vscsi1. The reinitialization of the vtlogs0 device on p6-570vio1 is recorded, but the active path remains on p6-570vio2.

## Failing back to the original path

We now disable the virtual log target device on p6-570vio2 by using the **rmvlog** command and then reexamine the state file to see what changed. Example 6-30 shows the **rmvlog** command that is used on p6-570vio2 to disable the virtual log target device. The extra state file messages are then examined, which shows that p6-570vio1 opened the log file and the client LPAR switched to the new path.

*Example 6-30 Removing the active path and confirming the failover to the original path*

```

$ rmvlog -dev vtlogs0
vtlogs0 Defined
$ cat
/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog/s1_logA.state.
000
(Lines from last example removed)
[1347877578] [p6-570vio2] vtlogs0 shutting down
[1347877578] [p6-570vio1] vtlogs0 using
    /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/client1/syslog//
    client1_syslog.000
[1347877579] [p6-570vio1] Client switched from path 1
    (devno=17,1;lua=0x8100000000000000) to path 0
    (devno=17,0;lua=0x8200000000000000)

```

---

As shown, p6-570vio1 opened the log file when the client switched back to using device 17,0, which corresponds to vscsi0. The client LPAR did not experience any disruption to the virtual log device during these failover operations.

## 6.5.6 Configuring AIX auditing to use a virtual log

The AIX auditing subsystem provides a means to record security-related information. It alerts system administrators of potential and actual violations of the system security policy. Auditing collects the following information:

- ▶ The name of the auditable event
- ▶ The status (success or failure) of the event
- ▶ More event-specific information that relates to security auditing

For more information about AIX auditing and its capabilities, see *Auditing and Accounting on AIX*, SG24-6020.

Without Trusted Logging, these audit records are stored in a text or binary form on the client LPAR that generated them. Trusted Logging also allows the binary versions of these audit records to be transferred by way of a virtual log device to the Virtual I/O Server. On the Virtual I/O Server, the audit records cannot be modified or removed by a malicious user on the client LPAR. To use Trusted Logging with AIX auditing, the following steps must be performed:

1. Create a virtual log on the Virtual I/O Server and attach it to the required Virtual SCSI server adapter.
2. Detect the new virtual log device on the client LPAR.
3. Configure AIX auditing to use the virtual log device.

These steps are described next.

### Creating the virtual log

The virtual log that is used for auditing can be a local virtual log, or it can be in a shared storage pool. For more information about complex configurations of virtual logs, see the following sections:

- ▶ 6.5.2, “Creating a virtual log on a single Virtual I/O Server” on page 220
- ▶ “Creating a single-path virtual log in a shared storage pool” on page 224
- ▶ “Creating a multipath virtual log using shared storage pools” on page 225

For this AIX auditing procedure, Example 6-31 shows the creation of a local virtual log that is named `audit`, which is attached to the Virtual SCSI server adapter `vhost0`. In this example, the new virtual log target device that is created is called `vtlog2`.

*Example 6-31 Creation of a simple virtual log for use by the AIX auditing subsystem*

---

```
$ mkvlog -vadapter vhost0 -name audit
Virtual log 000000000000000060cc6b83263a1143 created
vtlog2 Available
```

---

### Detecting the new virtual log device on the client LPAR

After a virtual log is created and attached to the appropriate Virtual SCSI server adapter, it is detectable by the client LPAR.

New devices are detected by running the `cfgmgr` command after which the virtual log is available for use.

As described in 6.3.2, “Virtual log devices” on page 212, the `lsdev` and `lsattr` commands can be used to identify the log names that are associated with each device. This identification is important. We want to ensure that the AIX auditing subsystem writes its logs to the log that we created and not to some other virtual log that exists.

Example 6-32 shows how new virtual logs can be detected by using the **cfgmgr** command. New virtual logs can be displayed with a combination of the **lsdev**, **xargs**, and **lsattr** commands. In this example, the virtual log that we created on the Virtual I/O Server that is named **audit** is detected as the **vlog0** device on the client LPAR. It is accessible as **/dev/vlog0**.

*Example 6-32 Detecting and identifying the new audit virtual log on the client LPAR*

---

```
$ cfgmgr
$ lsdev -Fname -tvlog | xargs -L1 lsattr -Ea log_name -F"name value" -l
vlog0 audit
vlog1 syslog
```

---

## Configuring AIX auditing to use the virtual log device

After the new virtual log device is detected and identified, it can be linked to the AIX auditing subsystem by modifying the **/etc/security/audit/config** file on the client LPAR. This file controls what events the auditing subsystem captures and where it logs them. For more information about the configuration file, see this IBM Knowledge Center [web page](#).

For this example, the important section in the configuration file is the **bin:** section, which controls the placement of binary audit records. A typical **bin:** section might look like the **bin:** section that is shown in Example 6-33.

*Example 6-33 Example bin: section of an /etc/security/audit/config file*

---

```
bin:
    trail = /audit/trail
    bin1 = /audit/bin1
    bin2 = /audit/bin2
    binsize = 10240
    cmds = /etc/security/audit/bincmds
    freespace = 65536
    backuppath = /audit
    backupsize = 0
    bincompact = off
```

---

To enable output to a virtual log, another **virtual\_log** line must be added to the **/etc/security/audit/config** file to enable Trusted Logging on the AIX auditing subsystem. This added line is shown in Example 6-34. This line must refer to whichever virtual log device is identified as the device to use for audit records.

*Example 6-34 Extra virtual\_log line added to enable Trusted Logging*

---

```
virtual_log = /dev/vlog0
```

---

After this line is added to the **/etc/security/audit/config** file, the auditing subsystem can be restarted by using the commands that are shown in Example 6-35.

*Example 6-35 Commands required to shut down and restart the AIX auditing subsystem*

---

```
$ audit shutdown
auditing reset
$ audit start
```

---

AIX audit messages are now written to the virtual log's directory on the Virtual I/O Server. For more information about how to display the log records on the Virtual I/O Server, see 6.5.3, "Accessing virtual log data on the Virtual I/O Server" on page 220.

This section describes the display of audit records. Example 6-36 shows displaying the contents of the binary audit log on the Virtual I/O Server by using the **auditpr -r** command.

*Example 6-36 Using the auditpr -r command Virtual I/O Server to display the contents of a binary audit log*

---

```
$ auditpr -r -i /var/vio/vlogs/s2/audit/s2_audit.000
```

---

event	login	status	time	command	wpar name
<hr/>					
FS_Chdir	0	OK	Thu Sep 20 05:46:17 2012	ps	Global
FS_Chdir	0	OK	Thu Sep 20 05:47:09 2012	bash	Global
FS_Mkdir	0	FAIL	Thu Sep 20 05:47:13 2012	java	Global
FILE_Unlink	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FILE_Rename	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FS_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FS_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
S_Rmdir	0	OK	Thu Sep 20 05:47:13 2012	java	Global
FILE_Unlink	0	OK	Thu Sep 20 05:47:15 2012	rm	Global
FS_Chdir	0	OK	Thu Sep 20 05:47:17 2012	ps	Global

---

## 6.5.7 Configuring syslog to use a virtual log

The syslog facility collects log messages from a range of applications and system services. It provides centralized control over which of these messages are logged and where they are written. It is available as part of the standard AIX installation.

The syslog facility works by providing a special file that is named `/dev/log`, to which applications and services write messages. The `syslogd` daemon reads the messages that are written to this file and processes them according to the rules that are specified in the `/etc/syslog.conf` file.

You can edit `/etc/syslog.conf` to match log messages based on the following information:

- ▶ Facility (the application or service that generated the message, such as `mail` or `auth`)
- ▶ Priority Level (`alert`, `warn`, `info`, `debug`, and so on)

For more information about the structure of the `/etc/syslog.conf` file, see the entry for `syslog.conf` at this IBM Knowledge Center [web page](#).

A virtual log presents itself on the client LPAR as a file in `/dev`. To use virtual logs as a destination for syslog messages, configure `/etc/syslog.conf` to write the messages that you want to the required virtual log.

To use Trusted Logging with syslog, the following steps must be performed:

1. Create a virtual log on the Virtual I/O Server and attach it to the required Virtual SCSI server adapter.
2. Detect the new virtual log device on the client LPAR.
3. Configure syslog to use the virtual log device.

These steps are described next.

## Creating the virtual log

The virtual log use for syslog can be a local virtual log or it can be in a shared storage pool. For more information about complex configurations of virtual logs, see the following sections:

- ▶ 6.5.2, “Creating a virtual log on a single Virtual I/O Server” on page 220
- ▶ “Creating a single-path virtual log in a shared storage pool” on page 224
- ▶ “Creating a multipath virtual log using shared storage pools” on page 225

For this syslog procedure, Example 6-37 shows the creation of a local virtual log that is named `syslog` and attached to the Virtual SCSI server adapter `vhost0`. In this example, the new virtual log target device that is created is called `vtlog3`.

*Example 6-37 Creation of a simple virtual log for use by the AIX auditing subsystem*

---

```
$ mkvlog -vadapter vhost0 -name syslog
Virtual log 000000000000000045a04622edfc10ad created
vtlog3 Available
```

---

## Detecting the new virtual log device on the client LPAR

After a virtual log is created and attached to the appropriate Virtual SCSI server adapter, it is detectable by the client LPAR.

New devices are detected by running the `cfgmgr` command, after which the virtual log is available for use.

As described in 6.3.2, “Virtual log devices” on page 212, the `lsdev` and `lsattr` commands can be used to identify the log names that are associated with each device. This identification is important. We want to ensure that the syslog subsystem writes its logs to the log that we created, and not some other virtual log that exists.

Example 6-38 shows how new virtual logs can be detected by running the `cfgmgr` command. The example shows how new virtual logs can be displayed with a combination of the `lsdev`, `xargs`, and `lsattr` commands. In this example, the virtual log that we created on the Virtual I/O Server that is named `syslog` is detected as the `vlog1` device on the client LPAR. It is accessible as `/dev/vlog1`.

*Example 6-38 Detecting and identifying the new audit virtual log on the client LPAR*

---

```
$ cfgmgr
$ lsdev -Fname -tvlog | xargs -L1 lsattr -Ea log_name -F"name value" -l
vlog0 audit
vlog1 syslog
```

---

## Configuring syslog to use the virtual log device

After the new virtual log device is detected and identified, it can be linked to the syslog subsystem by modifying the `/etc/syslog.conf` file. Example 6-39 shows a line that, when added to `/etc/syslog.conf`, causes informational messages that relate to authentication to be directed to the syslog virtual log, which we identified as `vlog1`. These messages capture login activity on the client LPAR.

*Example 6-39 The syslog.conf line to direct authentication messages to vlog1 virtual log*

---

```
auth.info      /dev/vlog1
```

---

After the configuration change is made, the **syslogd** process must be instructed to reread its configuration file. Example 6-40 shows how this task can be achieved by using the **refresh** command.

*Example 6-40 Reload of the syslogd service to reread the configuration file*

---

```
$ refresh -s syslogd
```

```
0513-095 The request for subsystem refresh was completed successfully.
```

---

Attempted and successful logins now generate informational messages that are transmitted to the Virtual I/O Server by way of the virtual log, vlog1. On the Virtual I/O Server, the log file can be inspected to view these messages. Example 6-41 shows log contents that are typical of auth.info messages.

*Example 6-41 Viewing the contents of syslog from within the Virtual I/O Server*

---

```
$ cat /var/vio/vlogs/s2/syslog/s2_syslog.000
```

```
Sep 21 08:05:40 s2 auth|security:info sshd[9371764]: Failed password for root from
172.16.254.6 port 64104 ssh2
Sep 21 08:05:40 s2 auth|security:info syslog: ssh: failed login attempt for root from
172.16.254.6
Sep 21 08:06:18 s2 auth|security:info sshd[9371772]: Failed password for testuser
from 172.16.254.6 port 64106 ssh2
Sep 21 08:06:18 s2 auth|security:info syslog: ssh: failed login attempt for testuser
from 172.16.254.6
Sep 21 08:06:22 s2 auth|security:info sshd[9371772]: Accepted password for testuser
from 172.16.254.6 port 64106 ssh2
```

---

## 6.5.8 Backing up Trusted Logging data on the Virtual I/O Server

This section describes how your virtual log configuration on the Virtual I/O Server can be backed up and restored. It is not intended to be a full description of the backup options that are available on the Virtual I/O Server. For more information about those backup capabilities, see section 5.2 of *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

A brief description of where Trusted Logging stores various aspects of its configuration data is required to better understand why certain backup operations are necessary.

### Log file data

For local and shared storage pool virtual logs, a virtual log repository root directory is available into which log data is stored. The **lsvlrepo** command can be used to display the path to each virtual log repository. Example 6-42 shows how to use the **-field** option to produce output that contains the name only of the shared storage pool and the virtual log repository directory, one per line.

*Example 6-42 Using lsvlrepo to show paths to virtual log repository directories*

---

```
$ lsvlrepo -field sp,path
```

```
,/var/vio/vlogs
vlog_ssp,/var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/
```

---

In this example, the first line represents the local virtual log repository (because no shared storage pool is indicated). The second line represents the vlog\_ssp shared storage pool of which the Virtual I/O Server on which this command is run is a member.



## Virtual log repositories

The virtual log repositories specify default values for the log and state file number and maximum sizes. They also specify the path to the virtual log repository root directory, into which log and state data files are written.

For local virtual logs, these properties are stored in the `vlogrepo0` pseudo-device, which does not represent a physical device in the system. It contains the configuration options that are required by a virtual log repository. Because it is a device, its configuration can be backed up and restored with the **`viosbr -backup`** command.

For virtual log repositories in shared storage pools, the configuration of the associated virtual log repository is stored in a database that is part of the shared storage pool subsystem. This database is backed up by using the **`viosbr -backup -clustername`** command.

## Virtual log configuration

For local virtual logs, the configuration of the virtual logs themselves (log name, client name, log and state file number, and maximum sizes) is stored in a `config` subdirectory within the local virtual log repository root directory (for more information, see 6.1.3, “Virtual logs” on page 202).

That directory must be backed up if the configuration of the local virtual logs is retained. By default, this directory is not readable by the `padmin` user. This directory must be made readable from within the **`oem_setup_env`** setup environment before this data can be backed up by the `padmin` user. Use a command sequence, such as the command that is shown in Example 6-43.

*Example 6-43 Making the virtual log configuration accessible by the padmin user*

---

```
$ oem_setup_env
# chmod a+rwX /var/vio/vlogs/config
# exit
```

---

For virtual logs that are in shared storage pools, the configuration of the virtual logs is stored in a database that is part of the shared storage pool subsystem. This database is backed up by using the **`viosbr -backup -clustername`** command.

## Virtual log target devices

For local and shared storage pool virtual logs, the virtual log target devices are stored in the system device database. Back up the database by using the **`viosbr -backup`** command.

## Backing up the contents of virtual logs

No special commands are available to back up this data. However, the Virtual I/O Server **`backup`** command provides the `-i` option, which archives a specified list of files to a removable media device.

Combined with an invocation of the **`find`** command to locate all the files in a specific virtual log repository, a one-line command to back up the virtual log repository can be produced, as shown in Example 6-44.

*Example 6-44 Use of find and backup to store a copy of virtual log data*

---

```
$ find /var/vio/vlogs -print | backup -i -v -q
find: 0652-081 cannot change directory to </var/vio/vlogs/config>:
: The file access permissions do not allow the specified action.
Backing up to /dev/rfd0.
```

---

```
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a      0 /var/vio/vlogs
a      0 /var/vio/vlogs/s2
a      0 /var/vio/vlogs/s2/syslog
a    291 /var/vio/vlogs/s2/syslog/s2_syslog.state.000
The total size is 603 bytes.
Backup finished on Sun Sep 23 18:30:45 EDT 2012; there are 100 blocks on 1
volumes.
```

---

In Example 6-44 on page 235, the **find** command sends a list of the files to the **backup** command, which writes the files to the default removable media device rfd0. The config directory is inaccessible to the padmin user by default and therefore is not backed up.

### Backing up configuration data

The operations that you must perform to obtain a full backup of your virtual log configuration depends on whether you are using only local virtual logs, only shared storage pool virtual logs, or both. The following procedures can be inferred from the descriptions of the various data types and their locations that are provided but are described here with examples:

- ▶ If you use *only* local virtual logs, use the **viosbr -backup** command to back up the virtual log repository and the virtual log target device configuration. Separately copy the config subdirectory of the local virtual log repository root directory, perhaps by using the **backup** command.

Example 6-45 shows this operation when the local virtual log repository is in the /var/vio/vlogs directory. It first runs the **viosbr** command to generate a backup file, which is named /tmp/viosback.tar.gz. It uses the **find** command to produce a list of files that includes /tmp/viosback.tar.gz *and* the contents of the local virtual log repository's config directory. This list of files is sent to the **backup** command, which writes them all to the default removable media device.

**File system permissions:** The padmin user cannot access the virtual log repository's config directory by default. You must set the permissions from the **oem\_setup\_env** setup environment for the user that performs backups before this command sequence functions correctly.

*Example 6-45 Complete backup of local virtual log repository data*

---

```
$ viosbr -backup -file /tmp/viosback
Backup of this node (vios01) successful
$ find /tmp/viosback.tar.gz /var/vio/vlogs/config | backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a    4485 /tmp/viosback.tar.gz
a      0 /var/vio/vlogs/config
a    312 /var/vio/vlogs/config/0000000000000000c61fee707f5b454a.vlog
The total size is 4797 bytes.
Backup finished on Sun Sep 23 19:03:23 EDT 2012; there are 100 blocks on 1
volumes.
```

---

- If you use *only* shared storage pool virtual logs, use the **viosbr -backup -clustername** command to back up the virtual log repository, virtual log, and virtual log target device configuration. This operation is simple, as shown in Example 6-46.

*Example 6-46 Complete backup of shared storage pool virtual log repository data*

---

```
$ viosbr -backup -clustername vlog_cluster -file /tmp/viosback
Backup of node p6-570vio1 successful
Backup of this node (p6-570vio2) successful
$ echo /tmp/viosback.vlog_cluster.tar.gz | backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a      336083 /tmp/viosback.vlog_cluster.tar.gz
The total size is 336083 bytes.
Backup finished on Sun Sep 23 19:13:06 EDT 2012; there are 700 blocks on 1
volumes.
```

---

A single invocation of the **viosbr** command captures the configuration of all Virtual I/O Servers in the specified cluster (**vlog\_cluster** in this example) and writes it to a file, which in this case is called **/tmp/viosback.vlog\_cluster.tar.gz**. The name of this file can be sent to the **backup** command, which writes its contents to the default removable media device.

- If you use *both* local virtual logs and shared storage pool virtual logs, use the **viosbr -backup -clustername** command to back up the virtual log repositories, shared storage pool virtual logs, and all virtual log target device configurations. Separately back up the config subdirectory of the local virtual log repository root directory. Example 6-47 shows this operation when the local virtual log repository is in the **/var/vio/vlogs** directory.

**File system permissions:** The padmin user does not have access to the virtual log repository's config directory by default. You must set the permissions appropriately from the **oem\_setup\_env** setup environment for the user that performs backups before this command sequence functions correctly.

*Example 6-47 Backup of shared storage pool and local virtual log repository data*

---

```
$ viosbr -backup -clustername vlog_cluster -file /tmp/viosback
Backup of node p6-570vio1 successful
Backup of this node (p6-570vio2) successful
$ find /tmp/viosback.vlog_cluster.tar.gz /var/vio/vlogs/config |
  backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a 364477 /tmp/viosback.vlog_cluster.tar.gz
a      0 /var/vio/vlogs/config
a   312 /var/vio/vlogs/config/0000000000000000b9e896af7e5b8377.vlog
The total size is 364789 bytes.
Backup finished on Sun Sep 23 19:23:09 EDT 2012; there are 800 blocks on 1
volumes.
```

---

It first starts the **viosbr** command to capture the configuration of all Virtual I/O Servers in the specified cluster (**vlog\_cluster** in this example) into a backup file. This backup file is named **/tmp/viosback.vlog\_cluster.tar.gz**.

It also uses the **find** command to produce a list of files that includes `/tmp/viosback.vlog_cluster.tar.gz` and the contents of the local virtual log repository's config directory. This list of files is sent to the **backup** command, which writes them all to the default removable media device.

## Restoring from backup

Restoring the virtual log configuration from backup is essentially the reverse of the backup operations. The Virtual I/O Server configurations that are backed up with **viosbr -backup** must be restored with the **viosbr -restore** command. The **restore** command can be used to extract backed-up files from removable media and restore them to the file system.

Next, we show an example restore procedure for the same three backup configurations that we described (local virtual logs only, shared storage pool virtual logs only, and local and shared storage pool virtual logs):

- If you use *only* local virtual logs, first restore the local virtual log repository config subdirectory and backup file by using the **restore** command. Then, use the **viosbr -restore** command to restore the virtual log repository and the virtual log target device configurations from that backup file.

Example 6-48 shows a full backup, modify, and restore procedure.

*Example 6-48 Viewing, backing up, deleting, and restoring local virtual logs*

---

```
$ lsvlrepo -local -detail
Local Virtual Log Repository:
  Repository State:      enabled
  Path:                  /vlog6
  Maximum Log Files:     2
  Maximum Log File Size: 1048576
  Maximum State Files:   2
  Maximum State File Size: 1048576

$ lsvlog
Client Name  Log Name  UUID                                VTD
s1           syslog   000000000000000015a339a71349e5a6  vhost0/vtlog2
s1           audit    0000000000000000d92d8e7cc4d99b1e  vhost0/vtlog1
s2           syslog   0000000000000000b9e896af7e5b8377  vhost1/vtlog0

$ viosbr -backup -file /tmp/viosback
Backup of this node (p6-570vio2) successful

$ find /tmp/viosback.tar.gz /vlog6/config | backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a      4322 /tmp/viosback.tar.gz
a      0    /vlog6/config
a      312 /vlog6/config/00000000000000000015a339a71349e5a6.vlog
a      312 /vlog6/config/000000000000000000b9e896af7e5b8377.vlog
a      312 /vlog6/config/000000000000000000d92d8e7cc4d99b1e.vlog
The total size is 5258 bytes.
Backup finished on Mon Sep 24 04:36:45 EDT 2012; there are 100 blocks on 1 volumes.

$ rmvlog -dbdata -u 00000000000000000015a339a71349e5a6
vtlog2 deleted
Virtual log 00000000000000000015a339a71349e5a6 deleted.
Log files deleted.

$ rmvlog -dbdata -u 000000000000000000d92d8e7cc4d99b1e
vtlog1 deleted
Virtual log 000000000000000000d92d8e7cc4d99b1e deleted.
Log files deleted.

$ rmvlog -dbdata -u 000000000000000000b9e896af7e5b8377
vtlog0 deleted
```

```

Virtual log 000000000000000b9e896af7e5b8377 deleted.
Log files deleted.
$ lsvlog
$ restore -vq
New volume on /dev/rfd0:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Mon Sep 24 04:36:45 EDT 2012
Files are backed up by name.
The user is root.
x      4322 /tmp/viosback.tar.gz
x      0 /vlog6/config
x      312 /vlog6/config/00000000000000015a339a71349e5a6.vlog
x      312 /vlog6/config/000000000000000b9e896af7e5b8377.vlog
x      312 /vlog6/config/000000000000000d92d8e7cc4d99b1e.vlog
The total size is 5258 bytes.
The number of restored files is 5.
$ viosbr -restore -file /tmp/viosback.tar.gz
vtlog2 Available
vtlog1 Available
vtlog0 Available
Backedup Devices that are unable to restore/change
=====

DEPLOYED or CHANGED devices:
=====
Dev name during BACKUP          Dev name after RESTORE
-----
vtlog2                          vtlog2
vtlog1                          vtlog1
vtlog0                          vtlog0
$ lsvlog
Client Name    Log Name    UUID                                VTD
s1             syslog     00000000000000015a339a71349e5a6  vhost0/vtlog2
s1             audit      000000000000000d92d8e7cc4d99b1e  vhost0/vtlog1
s2             syslog     000000000000000b9e896af7e5b8377  vhost1/vtlog0

```

Here, for reference, the current virtual log configuration is displayed with **lsvlrepo** and **lsvlog**. The Virtual I/O Server configuration is backed up by using **viosbr -backup**.

The **backup** command is used to write both the configuration backup and the files in the local virtual log's config subdirectory to removable media. The local virtual logs are then removed by using the **rmvlog** command. The **lsvlog** command is used to show that all virtual logs are removed.

The restoration procedure uses the **restore** command to restore the virtual log repository config subdirectory and backup files. The **viosbr** command is used to restore the virtual log target devices from the restored backup. The **lsvlog** command is then used to show that all virtual log devices are restored.

- If you use *only* shared storage pool virtual logs, use the **viosbr -restore -clustername** command to restore the virtual log repository, virtual log, and virtual log target device configuration.

The restoration procedure uses the **restore** command to restore the backup file from removable media. Then, complete the following process to restore data from the cluster backup file:

- a. Remove all other nodes from the cluster by using the **cluster -rmnode** command because the next step can be performed on a single-node cluster only.
- b. Restore the shared storage pool database by using the **viosbr -recoverdb** command.

- c. Restore the shared storage pool cluster configuration by using the **viosbr -restore -repopvs** command, which reintroduces the Virtual I/O Servers into the cluster that are removed in the first step.
- d. Restore the virtual log target device configuration on the Virtual I/O Server by using the **viosbr -restore -subfile** command.

Before undertaking this recovery procedure, the following information must be recorded:

- The device that serves as the cluster repository disk. This device can be seen in the output of a **lscluster -d** command, which identifies the repository disk with the type **REPDISK**. Example 6-49 shows sample output, in which **hdisk8** can be identified as the repository disk. The disk name is passed as a parameter to the **viosbr -restore -repopvs** command during the restore procedure.

*Example 6-49 Identifying the repository disk using lscluster -d*

---

```
$ lscluster -d
Storage Interface Query

Cluster Name: vlog_cluster
Cluster uuid: f7b7e24e-fe83-11e1-bbce-001125cced9d
Number of nodes reporting = 2
Number of nodes expected = 2
Node p6-570vio2
Node uuid = f7b177a6-fe83-11e1-bbce-001125cced9d
Number of disks discovered = 2
    hdisk9
        state : UP
        uDid  : 3E213600A0B800029AC1200005192504868C60F1815
FASTT03IBMfc
    uUId   : 6a8e8d99-5e8b-27ea-6c7f-2dddfd3fd532
    type  : CLUSDISK
    hdisk8
        state : UP
        uDid  :
        uUId   : 36247c45-0407-6906-06f2-802b1c6d0b82
        type  : REPDISK
```

*Details of other node removed.*

---

- The MTM and Partition Number of the Virtual I/O Server whose configuration is to be restored must be known. View this information by using the **cluster -status** command to show the MTM and Partition Number of the nodes in the cluster. Example 6-50 shows how these values are displayed. These values are passed to the **viosbr -restore -subfile** command during the restore procedure.

*Example 6-50 Using cluster -status to identify MTM and Partition Number*

---

```
$ cluster -status -clustername vlog_cluster
```

Cluster Name	State
vlog_cluster	OK

Node Name	MTM	Partition Num	State	Pool State
p6-570vio2	9117-MMA02101F170	2	OK	OK
p6-570vio1	9117-MMA02101F170	1	OK	OK

---

Example 6-51 shows a full backup, modify, and restore procedure.

*Example 6-51 View, back up, delete, and restore shared storage pool virtual logs*

---

```
$ lsvlrepo
Storage Pool  State    Path
              enabled  /vlog6
vlog_ssp      enabled  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/
$ lsvlog
Client Name    Log Name  UUID                                VTD
s2             syslog   99b977dec96860fb09268ff7eba91a00  vhost1/vtlogs3
s1             syslog   99b977dec96860fb8e7938a15a4a09a2  vhost0/vtlogs1
s1             audit    99b977dec96860fbd3cbfd48a76c44d   vhost0/vtlogs2
$ viosbr -backup -clustername vlog_cluster -file /tmp/viosback
Backup of node p6-570vio1 successful
Backup of this node (p6-570vio2) successful
$ echo /tmp/viosback.vlog_cluster.tar.gz | backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a          306498 /tmp/viosback.vlog_cluster.tar.gz
The total size is 306498 bytes.
Backup finished on Mon Sep 24 05:12:02 EDT 2012; there are 600 blocks on 1 volumes.
$ rmvlog -dbdata -u 99b977dec96860fb09268ff7eba91a00
vtlogs3 deleted
Virtual log 99b977dec96860fb09268ff7eba91a00 deleted.
Fileset access removed. Freeing data owned by fileset...
done
Log files deleted.
$ rmvlog -dbdata -u 99b977dec96860fb8e7938a15a4a09a2
vtlogs1 deleted
Virtual log 99b977dec96860fb8e7938a15a4a09a2 deleted.
Fileset access removed. Freeing data owned by fileset...
done
Log files deleted.
$ rmvlog -dbdata -u 99b977dec96860fbd3cbfd48a76c44d
vtlogs2 deleted
Virtual log 99b977dec96860fbd3cbfd48a76c44d deleted.
Fileset access removed. Freeing data owned by fileset...
done
Log files deleted.
$ lsvlog
$ restore -vq
New volume on /dev/rfd0:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Mon Sep 24 05:12:02 EDT 2012
Files are backed up by name.
The user is root.
x          306498 /tmp/viosback.vlog_cluster.tar.gz
The total size is 306498 bytes.
The number of restored files is 1.
$ cluster -rmnode -clustername vlog_cluster -hostname p6-570vio1
Partition p6-570vio1 has been removed from the vlog_cluster cluster

$ viosbr -recoverdb -clustername vlog_cluster -file /tmp/viosback.vlog_cluster.tar.gz
"Database restore successful.
$ viosbr -restore -clustername vlog_cluster -file /tmp/viosback.vlog_cluster.tar.gz
-repopvs hdisk8

WARNING: Below partitions are added into cluster.
```

Run `viosbr` with `-subfile` and `-skipcluster` on these nodes to restore non SSP devices.  
 p6-570viol

```
$ viosbr -restore -clustername vlog_cluster
  -file /tmp/viosback.vlog_cluster.tar.gz
  -subfile vlog_clusterMTM9117-MMA02101F170P2.xml
```

vtlogs2 Available

vtlogs1 Available

vtlogs3 Available

Backedup Devices that are unable to restore/change

=====

DEPLOYED or CHANGED devices:

=====

Dev name during BACKUP	Dev name after RESTORE
------------------------	------------------------

-----

-----

vtlogs2	vtlogs2
---------	---------

vtlogs1	vtlogs1
---------	---------

vtlogs3	vtlogs3
---------	---------

\$ lsvlog

Client Name	Log Name	UUID	VTD
s2	syslog	99b977dec96860fb09268ff7eba91a00	vhost1/vtlogs3
s1	syslog	99b977dec96860fb8e7938a15a4a09a2	vhost0/vtlogs1
s1	audit	99b977dec96860fbdc3cbfd48a76c44d	vhost0/vtlogs2

For reference, the current virtual log configuration is displayed by using the `lsvlrepo` and `lsvlog` commands. The Virtual I/O Server configuration is backed up by using the `viosbr -backup -clustername` commands.

The **backup** command is used to write the configuration backup to removable media. The virtual logs are then removed by using the `rmvlog` command. The `lsvlog` command is used to show that all virtual logs are removed.

The backup procedure that is described in “Backing up configuration data” on page 236 is performed. By using the steps described previously, the repository disk is identified as `hdisk8`. The Virtual I/O Server MTM is `9117-MMA02101F170`. The Partition Number is 2.

Finally, the `lsvlog` command is used to show that all virtual log devices are restored.

- If you use *both* local virtual logs and shared storage pool virtual logs, use the `viosbr -restore -clustername` command to restore the virtual log repository, virtual log, and virtual log target device configuration.

The restoration procedure uses the **restore** command to restore the backup file and the local virtual log configuration from removable media, followed by a four-step process to restore data from the cluster backup file. The four steps are the same steps as described in the previous scenario, in which shared storage pool virtual logs are restored.

Example 6-52 shows a full backup, modify, and restore procedure.

*Example 6-52 Show the restored virtual log devices*

---

```
$ lsvlrepo -detail
```

Local Virtual Log Repository:

Repository State: enabled

Path: /vlog6

Maximum Log Files: 2

Maximum Log File Size: 1048576

Maximum State Files: 2

Maximum State File Size: 1048576

Virtual Log Repository for Shared Storage Pool vlog\_ssp:

Repository State: enabled



```

Path:
  /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/
Maximum Log Files:      1
Maximum Log File Size:  104857600
Maximum State Files:    2
Maximum State File Size: 1048576
$ lsvlog
Client Name      Log Name  UUID                               VTD
s1               syslog   99b977dec96860fb8e7938a15a4a09a2 vhost0/vtlogs1
s1               audit    99b977dec96860fbdc3cbfd48a76c44d vhost0/vtlogs2
s2               syslog   000000000000000068fe15551286c088 vhost1/vtlog0
s2               audit    0000000000000000a22822a0bb3b67bb vhost1/vtlog1
$ viosbr -backup -clustername vlog_cluster -file /tmp/viosback
Backup of node p6-570vio1 successful
Backup of this node (p6-570vio2) successfu
$ find /tmp/viosback.vlog_cluster.tar.gz /vlog6/config | backup -ivq
Backing up to /dev/rfd0.
Cluster 51200 bytes (100 blocks).
Volume 1 on /dev/rfd0
a      283608 /tmp/viosback.vlog_cluster.tar.gz
a      0 /vlog6/config
a      312 /vlog6/config/000000000000000068fe15551286c088.vlog
a      312 /vlog6/config/0000000000000000a22822a0bb3b67bb.vlog
The total size is 284232 bytes.
Backup finished on Mon Sep 24 06:26:01 EDT 2012; there are 600 blocks on 1 volumes.
$ rmvlog -dbdata -u 99b977dec96860fb8e7938a15a4a09a2
vtlogs1 deleted
Virtual log 99b977dec96860fb8e7938a15a4a09a2 deleted.
Fileset access removed. Freeing data owned by fileset...
done
Log files deleted.
$ rmvlog -dbdata -u 99b977dec96860fbdc3cbfd48a76c44d
vtlogs2 deleted
Virtual log 99b977dec96860fbdc3cbfd48a76c44d deleted.
Fileset access removed. Freeing data owned by fileset...
done
Log files deleted.
$ rmvlog -dbdata -u 000000000000000068fe15551286c088
vtlog0 deleted
Virtual log 000000000000000068fe15551286c088 deleted.
Log files deleted.
$ rmvlog -dbdata -u 0000000000000000a22822a0bb3b67bb
vtlog1 deleted
Virtual log 0000000000000000a22822a0bb3b67bb deleted.
Log files deleted.
$ lsvlog
$ restore -vq
New volume on /dev/rfd0:
Cluster size is 51200 bytes (100 blocks).
The volume number is 1.
The backup date is: Mon Sep 24 06:26:01 EDT 2012
Files are backed up by name.
The user is root.
x      283608 /tmp/viosback.vlog_cluster.tar.gz
x      0 /vlog6/config
x      312 /vlog6/config/000000000000000068fe15551286c088.vlog
x      312 /vlog6/config/0000000000000000a22822a0bb3b67bb.vlog
The total size is 284232 bytes.
The number of restored files is 4.

```

```
$ cluster -rmnode -clustername vlog_cluster -hostname p6-570vio1
Partition p6-570vio1 has been removed from the vlog_cluster cluster
```

```
$ viosbr -recoverdb -clustername vlog_cluster
-file /tmp/viosback.vlog_cluster.tar.gz
```

```
"Database restore successful.
```

```
"
```

```
$ viosbr -restore -clustername vlog_cluster
-file /tmp/viosback.vlog_cluster.tar.gz
-subfile vlog_clusterMTM9117-MMA02101F170P2.xml
```

```
vtlogs1 Available
vtlogs2 Available
vtlog1 Available
vtlog0 Available
Backedup Devices that are unable to restore/change
=====
```

```
DEPLOYED or CHANGED devices:
```

```
=====
```

Dev name during BACKUP	Dev name after RESTORE
-----	-----
vtlogs1	vtlogs1
vtlogs2	vtlogs2
vtlog1	vtlog1
vtlog0	vtlog0

```
$ lsvlog
```

Client Name	Log Name	UUID	VTD
s1	syslog	99b977dec96860fb8e7938a15a4a09a2	vhost0/vtlogs1
s1	audit	99b977dec96860fbdc3cbfd48a76c44d	vhost0/vtlogs2
s2	syslog	000000000000000068fe15551286c088	vhost1/vtlog0
s2	audit	0000000000000000a22822a0bb3b67bb	vhost1/vtlog1

First, the current virtual log configuration is displayed by using the **lsvlrepo** and **lsvlog** commands. The Virtual I/O Server configuration is backed up by using **viosbr -backup -clustername**. The **backup** command is used to write both the configuration backup and the files in the local virtual log's config subdirectory to removable media.

The virtual logs are then removed by using the **rmvlog** command. The **lsvlog** command is used to show that all virtual logs are removed.

The backup procedure that is described in “Backing up configuration data” on page 236 is then performed. By using the steps described previously, the repository disk is identified as hdisk8. The Virtual I/O Server MTM is 9117-MMA02101F170. The Partition Number is 2.

Finally, the **lsvlog** command is used to show that all virtual log devices are restored.

This section described Trusted Logging backup scenarios for local and shared storage pool configurations. The Virtual I/O Server **backup** and **restore** commands were used to place the backups onto removable media. Other options are available, such as mounting a remote Network File System (NFS) export by using the **mount** command and placing the required files onto it.

## 6.5.9 Deleting virtual logs and virtual log target devices

Virtual log devices on the client LPAR can be deleted by using the **rmdev** command in the same manner as any AIX device. Example 6-53 shows a useful combination of commands that can be used to delete all virtual log devices. Query the list of devices by using the **lsdev** command, and use the **xargs** utility to run **rmdev -d** on each device in turn.

*Example 6-53 Use lsdev, xargs, and rmdev to remove virtual log devices on client LPAR*

---

```
$ lsdev -F name -t vlog | xargs -L1 rmdev -d -l
```

---

Virtual logs and virtual log target devices can be deleted from within the Virtual I/O Server with the **rmvlog** command. As described in 6.1.3, “Virtual logs” on page 202, Trusted Logging connects virtual logs to client LPARs by attaching them to virtual log target devices.

The **rmvlog** command features the following modes of operation. Each mode removes a different amount of the specified virtual log infrastructure:

**no option**     Disable virtual log target device

When used without other options beyond the specification of a virtual log target device (with **-dev**) or a virtual log (with **-uuid**), the virtual log target device (or the device that is associated with the specified virtual log) is moved into the Disabled state. Therefore, the client LPAR experience errors if it attempts to write to it. This behavior is consistent with the **rmdev** command when used with no other command options. Specifying a virtual log that does not have an associated target device results in an error.

**-d**             Delete virtual log target device

When used with the **-d** option, the virtual log target device that is specified with the **-dev** option (or associated with the virtual log that is specified with the **-uuid** option) is deleted. Therefore, the client LPAR experiences errors if it attempts to write to it. This behavior is consistent with the **rmdev** command when used with the **-d** command option. Specifying a virtual log that does not have an associated target device results in an error.

**-db**            Delete virtual log and associated target device if it exists

When used with the **-db** option, the virtual log target device that is specified with the **-dev** option (or associated with the virtual log that is specified with the **-uuid** option, if one exists) is deleted, along with the associated virtual log. Therefore, the client LPAR experiences errors if it attempts to write to it, and the properties that are associated with the virtual log (log name, client name, and log and state file counts and sizes) are lost. The directory that contains the log and state files is retained. The **-dbdata** command option must be used instead of the **-db** command option if you want to remove this directory.

**-dbdata**       Delete virtual log, target device, and client data

When used with the **-dbdata** option, the behavior is the same as with the **-db** option, except that the directory that contains the log and state files is also removed.

These four scopes are summarized in Figure 6-8 on page 246, which also shows the components that are removed for each of the four possible removal options.

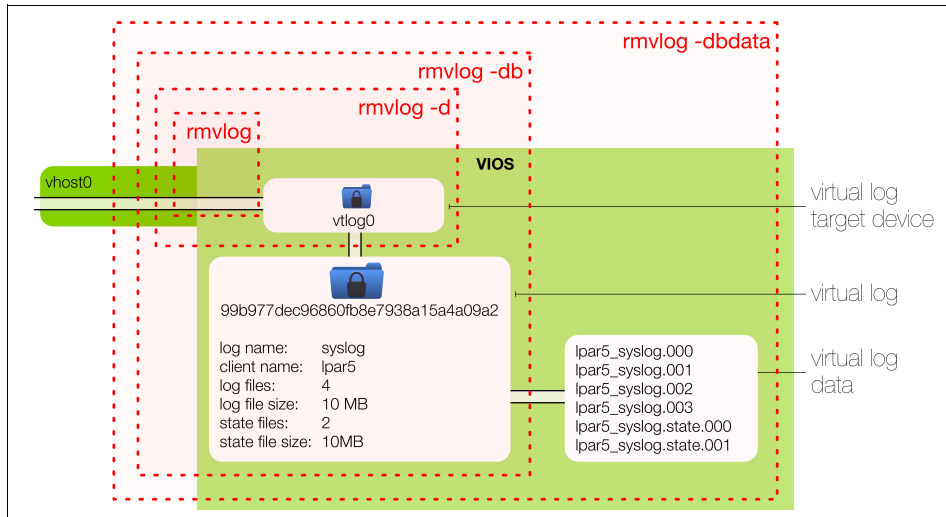


Figure 6-8 The four different scopes of the rmvlog command

## 6.6 Troubleshooting

This section describes the following common incorrect configurations and complex interactions that can occur during the configuration and deployment of Trusted Logging:

- “The following device packages are required” error occurs when a virtual log device is detected on the client LPAR.

If the error message that is shown in Example 6-54 occurs when new virtual log devices are detected, the PowerSC Trusted Logging package is not installed on the client LPAR. For more information about installing the package, see 6.4.1, “Installing the Client LPAR component” on page 218.

*Example 6-54 Error message on the client LPAR with no PowerSC Trusted Logging*

```
lpar2(root)/> cfgmgr
Method error (/usr/lib/methods/cfg_vclient -l vscsi1 ):
    0514-040 Error initializing a device into the kernel.
cfgmgr: 0514-621 WARNING: The following device packages are required for device
support but are not currently installed.
devices.vscsi.tm
```

- Deleting log files on the Virtual I/O Server does not free up as much disk space as you expected.

As described in 6.1.4, “Virtual log directory and file structure” on page 204, virtual logs store client and state data in a series of rotating log files.

When connected to a client LPAR, the virtual log target device driver in the Virtual I/O Server holds the current log file and the current state file open for writing so that incoming messages can be quickly written to disk.

However, file system semantics dictate that when a file is removed, space is not freed back to the file system until all processes close the file. As a result, the removal of log or state files directly with the **rm** command does not free up the space used by the current log and state files until the virtual log target device closes the file. Space that is used by other log and state files is freed immediately, as expected. If a virtual log is not in use by a virtual log target device, this issue does not occur and space is freed immediately.

Furthermore, the virtual log device driver in the Virtual I/O Server continues to write new log messages to the opened file. These new messages are inaccessible because the file is deleted and not visible from the file system.

For these reasons, deleting log and state files manually is not advised. It is better to correctly configure the log and state file counts and sizes to ensure that the disk space usage of a virtual log is acceptable.

However, it is possible to fully realize the free space that you want without causing I/O errors on the client LPAR. Instruct the virtual log target device to close and reopen the log and state files, which cause the file system to reclaim the space. Example 6-55 shows the problem. It then shows how it can be solved by using the **chvlog** command to instruct the virtual log target device to reinitialize, which occurs without failing any I/O on the client LPAR.

*Example 6-55 Using chvlog to free space that is used by opened state and log files*

---

```

$ df -m /vlogs
Filesystem      MB blocks      Free %Used      Iused %Iused Mounted on
/dev/fslv00     2048.00    1614.16    22%         18      1% /vlogs
$ lsvlog -detail -u 00000000000000068fe15551286c088
Client Name: s2

    Log Name:                syslog
    UUID:                    00000000000000068fe15551286c088
    Virtual Target Device:    vtlog0
    Parent Adapter:           vhost1
    Vlog State:               enabled
    Device Status:            available
    Logical Unit Address:      8100000000000000
    Storage Pool:
    Log Directory:            /vlogs/s2/syslog/
    Maximum Log Files:         5
    Maximum Log File Size:     104857600
    Maximum State Files:       2
    Maximum State File Size:   1048576
$ ls -l /vlogs/s2/syslog
total 833736
-rw-r----- 1 root system 104857600 Sep 24 07:42 s2_syslog.000
-rw-r----- 1 root system 104857600 Sep 24 07:42 s2_syslog.001
-rw-r----- 1 root system 104857600 Sep 24 07:42 s2_syslog.002
-rw-r----- 1 root system 89358336 Sep 24 07:42 s2_syslog.003
-rw-r----- 1 root system 104857600 Sep 24 07:42 s2_syslog.004
-rw-r----- 1 root system 1477 Sep 24 07:42 s2_syslog.state.000
$ oem_setup_env
# rm -f /vlogs/s2/syslog/*
# exit
$ df -m /vlogs
Filesystem      MB blocks      Free %Used      Iused %Iused Mounted on
/dev/fslv00     2048.00    1962.10     5%         14      1% /vlogs
$ chvlog -u 00000000000000068fe15551286c088 -state enabled
Updated device.
$ df -m /vlogs
Filesystem      MB blocks      Free %Used      Iused %Iused Mounted on
/dev/fslv00     2048.00    2047.32     1%         13      1% /vlogs

```

---

Example 6-55 on page 247 first uses the **df** command to show 1614.16 MB of available space in the **/vlogs** file system, which is the location of the virtual log repository. We then use the **lsvlog -detail** command to identify the directory for the virtual log that we want to delete. We use the **ls -l** command to show its contents.

We use the **rm** command (from within the setup environment) to delete all the files. We then use the **df** command again to show that all the space is not reclaimed; only 1962.10 MB are available.

We then use the **chvlog** command to enable to virtual log. The virtual log is enabled, so this command does not modify the configuration in any way. However, it also instructs the running virtual log target device to reinitialize itself, which closes and reopens the current state and log files.

We then use the **df** command a final time to show that the expected space is reclaimed in the file system, which now has 2047.32 MB available.

- “The requested resource is busy” error occurs on a client LPAR when opening a virtual log device.

If an error that is similar to the error that is shown in Example 6-56 is seen on the client LPAR, another process opened the virtual log. Multiple concurrent writers are not supported.

*Example 6-56 Multiple writers try to open the same virtual log on the client LPAR*

---

```
The requested resource is busy.  
ksh: /dev/vlog0: 0403-005 Cannot create the specified file.
```

---

If you can access the Virtual I/O Server, the state messages can be inspected to identify which process includes the file open. For more information about how to interpret the state file messages, see 6.3.3, “Messages that are written to the state files” on page 213.

- When the **auditpr** command is used to view audit records on the Virtual I/O Server, the user and group names are incorrect.

Because the **auditpr** command uses the local user and group files to map the IDs in the binary audit logs to string representations, the incorrect strings are used if the Virtual I/O Server has a different set of users and groups to the client LPAR on which the audit log is generated. Use the **-r** command option to the **auditpr** command to suppress the conversion of IDs to names, as described in 6.5.3, “Accessing virtual log data on the Virtual I/O Server” on page 220.

- Changes to a virtual log configuration take effect on the Virtual I/O Server, but the changes are not visible on the client LPAR by using the **lsattr** command.

The **lsattr** command queries only the device for attributes when it is first detected. To refresh this list, remove the virtual log device by using the **rmdev -d -l** command and start a detection of devices again with the **cfigmgr** command.

- Changes to shared storage pool virtual log configuration do not take effect on all Virtual I/O Servers in the cluster.

When the configuration of a virtual log in a shared storage pool is modified by using the **chvlog** command, the change is communicated to virtual log target devices that run on the Virtual I/O Server on which the change is requested. The change is *not* communicated to virtual log target devices that run on other Virtual I/O Servers in the cluster.

Therefore, virtual log target devices that provide multiple paths to a client LPAR can get out of sync, which results in unwanted behaviors, such as a path not respecting an updated log file size change.

The solution is to force a reconfiguration of the device on the Virtual I/O Servers that provide alternative paths for the same virtual log by issuing a **chvlog -state enabled** command. This command does not change any properties of the virtual log. Instead, it instructs running virtual log target devices on the Virtual I/O Server on which the command is run to reload their configuration from the virtual log.

Example 6-57 shows how a configuration change on the `vios1` Virtual I/O Server with **chvlog** (setting the number of log files to 5 by using the **-lfs** command option) is propagated to the `vios2` Virtual I/O Server by using the **chvlog -state enabled** command. As shown in Example 6-57, it also is ensured that a change to a virtual log's configuration is propagated to both paths in a shared storage pool configuration.

---

*Example 6-57 Propagating a change to a virtual log's configuration to both paths*

---

```
vios1 $ chvlog -lfs 5 -u 99b977dec96860fb8e7938a15a4a09a2
      Updated device.

vios2 $ chvlog -state enabled -u 99b977dec96860fb8e7938a15a4a09a2
      Updated device.
```

---

- “Fileset ... is actively being used and cannot be deleted” error occurs when you remove a virtual log from a shared storage pool.

Error messages similar to the messages that are shown in Example 6-58 are shown when the virtual log is in use by another Virtual I/O Server. This situation is not easily detectable because the **lsvlog** command shows only virtual log target devices on the Virtual I/O Server on which the command is run.

---

*Example 6-58 Error message when trying to remove a virtual log that is still in use*

---

```
rmvlog Error:
      Could not remove logs from the repository.
Virtual log 99b977dec96860fb8d4ede0f5be6a540 deleted.
0967-030 Fileset /var/vio/SSP/vlog_cluster/D_E_F_A_U_L_T_061310/vlogs/s1/logB
is actively being used and cannot be deleted.
```

---

Example 6-58 also shows the error message when an attempt is made to remove a virtual log with the **rmvlog** command and the virtual log is still in use by another virtual log target device on a different Virtual I/O Server.

To complete the deletion of the virtual log, locate the virtual log target device that is using the log and remove it with the **rmdev** command.

- “Client LPAR is not accessible for VSCSI adapter” error occurs when you create a virtual log with the **mkvlog** command.

Errors similar to the messages that are shown in Example 6-59 are produced when the client LPAR is not running. Example 6-59 shows the error message when you create a virtual log and attach it to a client LPAR that is not running.

---

*Example 6-59 Error when creating a virtual log and attaching it to inactive client LPAR*

---

```
mkvlog Error:
Client LPAR is not accessible for VSCSI adapter vhost0. Use -client option to
specify a client name for the new Virtual Log.
```

---

Starting the client LPAR and waiting until its operating system is fully started rectifies this problem. Alternatively, use the **-client** command option to specify the client name manually.

- No virtual logs (even local ones) can be seen with **lsvlog** when shared storage pools are configured.

If shared storage pools are configured but they malfunction in some way, a loss of access to all virtual log repositories can result. Rectify the shared storage pool configuration to restore access to virtual logs.

**Note:** The use of firewalls, such as a firewall that is activated with the **viosecure -firewall** command, can cause the shared storage pool cluster to malfunction.

One way to see whether the cluster is functioning properly is to use the **lssp** command to display the shared storage pools that the Virtual I/O Server can access. Example 6-60 shows the behavior of the **lssp** command when the shared storage pool cluster is working properly. It also shows the result when the shared storage pool cluster is impeded by the enablement of the Virtual I/O Server firewall.

*Example 6-60 The lssp command: A shared storage pool cluster and then a firewall*

---

```
$ lssp -clustername vlog_cluster
POOL_NAME:      vlog_ssp
POOL_SIZE:      20352
FREE_SPACE:     19831
TOTAL_LU_SIZE:  0
TOTAL_LUS:      0
POOL_TYPE:      CLPOOL
POOL_ID:        FFFFFFFFAC1014470000000050535277
$ viosecure -firewall on
$ lssp -clustername vlog_cluster
Unable to connect to Database
Unable to connect to Database
```

---

## 6.7 Conclusion

In this chapter, we introduced the Trusted Logging concept and described how it allows log data to be consolidated in a secure fashion with minimal configuration.

Many standards, such as the Payment Card Industry Data Security Standard (PCI DSS), require the secure storage of log data. Trusted Logging is of interest to any organization that is subject to regulatory compliance.

Even if compliance is not a direct concern, the ability of Trusted Logging to consolidate log data within a system (by using local virtual logs) or across a data center (by using shared storage pools) provides enhanced manageability when compared with other means of log collection and analysis.





# Trusted Boot

Although a broad consensus exists globally on the benefits of cloud computing, security is regularly cited as a major inhibitor to its adoption. Having a third-party host your production capabilities or confidential data can lead to fears of service disruption, data loss, or a leak of sensitive information. People fear scenarios, such as a cloud provider that provisions its customers with instrumented virtual machines to spy on their confidential data.

This chapter provides an overview of Trusted Boot, including its reference architecture, how to plan for implementation and installation, and troubleshooting information.

This chapter contains the following topics:

- ▶ 7.1, “Overview” on page 252
- ▶ 7.2, “Component architecture” on page 253
- ▶ 7.3, “Detailed implementation” on page 255
- ▶ 7.4, “Installation” on page 256
- ▶ 7.5, “Working with Trusted Boot” on page 259
- ▶ 7.6, “Troubleshooting” on page 268
- ▶ 7.7, “Conclusion” on page 272

## 7.1 Overview

When guarantees are required about availability, hardware reliability, regulatory compliance, and data privacy, having your hardware, data, and network traffic overseen by a third party is understandably an uncomfortable prospect.

Numerous ways of gaining assurances are available. Regulatory bodies, certifications, and periodic audits or inspections can all be used to help place trust in a cloud provider. However, they cannot provide continual, definitive proof that everything is alright.

Consider the disk image that belongs to your virtual machine (VM). How can you know for certain that your VM boots from the correct device and the disk image is not tampered with? The nature of the cloud means that this question must be answered remotely, which adds the following complications:

- ▶ How can I be sure that I am talking to the correct machine and not a man-in-the-middle machine?
- ▶ How do I know that the response I receive is honest and not manipulated to tell me what I want to hear?

Trusted Boot, which is part of the IBM PowerSC Standard Edition, provides a definitive answer to these questions. Trust Boot is based on the Trusted Computing Group's Trusted Platform Module (TPM) technology. It scrutinizes every step of the boot process, taking secure measurements (cryptographic hashes) of the software and recording them in a virtual TPM (vTPM). Recording data in the vTPM is a one-way-street. After a value is written, it can be retrieved, but it cannot be modified or over-written.

The cryptographic strength of the measurement, coupled with the vTPM capabilities, make it impossible to falsify a measurement. Trusted Boot forms an unbreakable chain of trust for every step of the boot process. For Power Systems, this chain starts at the hypervisor, continues through the partition firmware, and into AIX and the application layer.

Each link in the chain is responsible for measuring the next link and locking this measurement away in the vTPM where it cannot be tampered with. For AIX, it is inspected and analyzed and the measurements are locked away where it cannot touch them before it has a chance to run a single instruction of its own code. If the boot image on the disk is modified, Trusted Boot is aware of this change (see Figure 7-1).

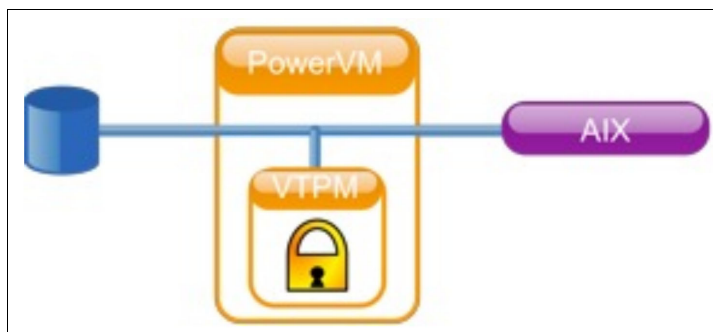


Figure 7-1 PowerVM hypervisor interrogates the AIX boot image before it runs

## 7.2 Component architecture

It is important to understand the integrity of the boot process and how to classify the boot as a trusted boot or a non-trusted boot.

You can configure a maximum of 60 vTPM-enabled logical partitions (LPARs) for each physical system by using the Hardware Management Console (HMC). When configured, the vTPM is unique to each LPAR. When used with the AIX Trusted Execution technology, the vTPM provides security and assurance to the following partitions:

- ▶ Boot image on the disk
- ▶ Entire operating system
- ▶ Application layers

An administrator can view trusted and non-trusted systems from a central console that is installed with the openpts verifier that is available on the AIX expansion pack. The openpts console manages one or more Power Systems servers. It also monitors or attests the trusted state of AIX systems throughout the data center. *Attestation* is the process where the verifier determines (or attests) if a collector performed a trusted boot.

A partition is said to be trusted if the verifier successfully attests the integrity of the collector. The verifier is the remote partition that determines whether a collector performed a trusted boot. The collector is the AIX partition that has a vTPM that is attached and the Trusted Software Stack (TSS) installed. It indicates that the measurements that are recorded within the vTPM match a reference set that is held by the verifier.

A trusted boot state indicates whether the partition booted in a trusted manner. This statement is about the integrity of the system boot process and does not indicate the current or ongoing level of the security of the system.

A partition enters a non-trusted state if the verifier cannot successfully attest the integrity of the boot process. The non-trusted state indicates that some aspect of the boot process is inconsistent with the reference information that is held by the verifier. The possible causes for a failed attestation include booting from a different boot device, booting a different kernel image, and changing the existing boot image.

### 7.2.1 Trusted Boot technical overview

The Trusted Computing Group help set the standard as to what comprises a Trusted Boot. To perform a Trusted Boot, each software element of the boot process, from the firmware through to the operating system, must be tested for authenticity. This test involves measuring (performing a checksum) the software and securely storing the result.

A specialized hardware device that is named TPM is typically used to store these measurements. A chain of trust is built by having each step of the boot process measure the next, starting with the code run at power-on.

After these measurements are made and the operating system is fully booted, a user can request the list of measurements from the VM, which are retrieved from the secure store. The user can then check this list of measurements against a set of values that are known to be good to attest that the software components can all be trusted.

The user must ascertain the authenticity of these measurements (that is, it must be impossible to return a fabricated set of results). To ascertain the authenticity, the secure storage (TPM) encrypts and signs the measurements before transmission (see Figure 7-2).

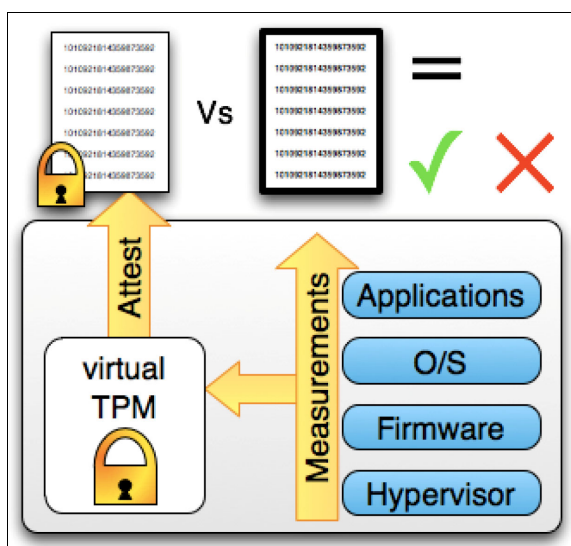


Figure 7-2 Trusted Boot attestation process

The goal of Trusted Boot is to bring this TPM and attestation capability to the POWER platform. POWER systems are not included with a hardware TPM component. The IBM Watson® Research Lab developed software to virtualize the functionality of a TPM. This vTPM software is used to provide the secure storage for the measurements of a POWER LPAR's Trusted Boot.

Each LPAR on a system must measure its boot sequence independently so each LPAR must have its own vTPM device. Adjuncts are used to provide the vTPM devices to the LPAR. Adjuncts provide a reusable, lightweight capability of presenting a hardware device to an LPAR.

The following prerequisites must be met to provide Trusted Boot for POWER LPARs:

- ▶ Modify the entire POWER boot process to perform measurements at every stage.
- ▶ Provide a secure way of storing these measurements.
- ▶ Provide the facility for users to query these measurements.

Some technical considerations influence the high-level design of the solution. One primary consideration is that hardware TPM chips contain an amount of storage space (NVRAM2) that persists across reboots. Therefore, vTPM devices are required to have comparable persistent storage.

POWER systems contain their own piece of NVRAM. The NVRAM is not scalable to provide storage for each possible VM's vTPM. Therefore, a limit of 60 partitions exists on a single machine that can contain a vTPM device.

## 7.3 Detailed implementation

The installation of Trusted Boot involves configuring the collector and the verifier. The hardware and software configurations that are required to install Trusted Boot are shown in Figure 7-3.

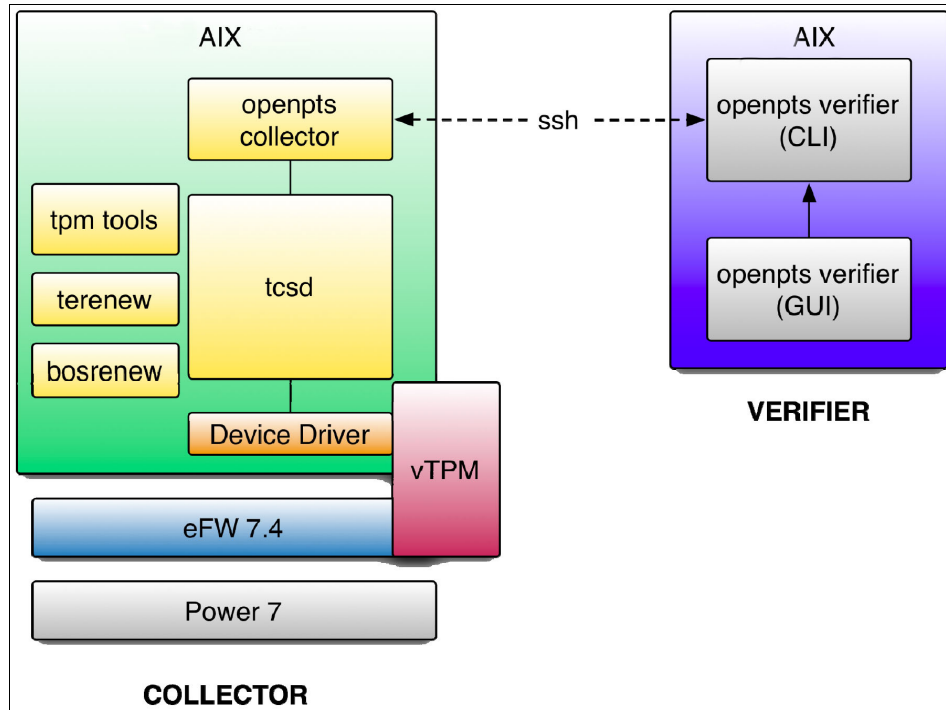


Figure 7-3 Trusted Boot components

The Trusted Boot/vTPM-enabled LPAR is called the *collector*. The LPAR that performs the attestation is called the *verifier*. The collector and the verifier have the following characteristics:

- ▶ Collector:
  - IBM POWER7® hardware that runs on a 740 firmware release.
  - IBM AIX 6 with Technology Level 7 or IBM AIX 7 with Technology Level 1.
  - Hardware Management Console (HMC) version 7.4 or later (not shown in Figure 7-3).
  - The partition is configured with the vTPM and a minimum of 1 GB memory.
  - Secure Shell (SSH) is required, specifically OpenSSH or equivalent.
  - The tcsd daemon must start at boot time.
  - PowerSC vTPM device driver is from the PowerSC Standard Edition CD.
  - Trusted Platform Module (TPM) tools are installed by default by AIX.
  - *terenew* is installed by default by AIX.
  - *bosrenew* is installed by default by AIX.
- ▶ Verifier:
  - SSH, specifically OpenSSH or equivalent.
  - Network connectivity (through SSH) to the collector.
  - Java 1.6 or later to access the openpts console from the graphical interface.

The OpenPTS verifier can be accessed from the command-line interface (CLI) and the graphical user interface (GUI) that is designed to run on a range of platforms.

The AIX version of the OpenPTS verifier ships as part of PowerSC Standard Edition. The other components are in the locations that are specified in Figure 7-4.

Filesets or Programs	Shipping Location
openpts.collector	Base AIX (not installed by default)
openpts.verifier for AIX (CLI and GUI)	PowerSC media Standard Edition
openpts.verifier for Linux (CLI and GUI)	IBM.com download
VTPM device driver	PowerSC media Standard Edition
tcsd	Base AIX Installed by default
tpm tools	Base AIX Installed by default
bosrenew terenew	Base AIX Installed by default

Figure 7-4 Trusted Boot software components

You must consider certain prerequisites before you migrate a partition to the vTPM. An advantage of a vTPM over a physical TPM is that it allows the partition to move between systems at the same time retaining the vTPM. To securely migrate the LPAR, the firmware encrypts the vTPM data before transmission.

To ensure a secure migration, the following security measures must be implemented before migration:

- ▶ Enable the IPSEC on the Virtual I/O Server that performs the migration.
- ▶ Set the trusted system key through the HMC to control the managed systems that can decrypt the vTPM data after migration. The migration destination system must have the same key as the source system to successfully migrate the data.

## 7.4 Installation

In this section, we describe how to install PowerSC Trusted Boot on the collector and the verifier.

**Be aware:** Collector and verifier components cannot be installed on the same system (LPAR or VM).

## 7.4.1 Installing the collector

Complete the following steps to install the collector:

1. Install the `openpts.collector` package from the AIX base CD by using the `smit` or `installp` command, as shown in Figure 7-5. The required user input is shown in bold.

Install and Update from ALL Available Software	
Type or select values in entry fields. Press Enter AFTER making all desired changes.	
	[Entry Fields]
* INPUT device / directory for software	.
* SOFTWARE to install	<b>[openpts.collector]</b> +
PREVIEW only? (install operation will NOT occur)	no +
COMMIT software updates?	yes +
<b>SAVE replaced files?</b>	<b>yes</b> +
AUTOMATICALLY install requisite software?	yes +
EXTEND file systems if space needed?	yes +
OVERWRITE same or newer versions?	no +
VERIFY install and check file sizes?	no +
DETAILED output?	no +
Process multiple volumes?	yes +
<b>ACCEPT new license agreements?</b>	<b>yes</b> +
Preview new LICENSE agreements?	no +
WPAR Management	
Perform Operation in Global Environment	yes +
Perform Operation on Detached WPARs	no +
Detached WPAR Names	<b>[_all_wpars]</b> +
Remount Installation Device in WPARs	yes +
Alternate WPAR Installation Device	<b>[]</b>

Figure 7-5 The `openpts.collector` package from the AIX base CD

2. Install the powerscStd.vtpm from the PowerSC CD by using the **smit** or **installp** command, as shown in Figure 7-6. The required user input is shown in bold.

```

Install and Update from ALL Available Software

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

                                [Entry Fields]
* INPUT device / directory for software      .
* SOFTWARE to install                        [powerscStd.vtpm]
powerscStd.license] +
  PREVIEW only? (install operation will NOT occur)  no +
  COMMIT software updates?                          yes +
  SAVE replaced files?                          yes +
  AUTOMATICALLY install requisite software?         yes +
  EXTEND file systems if space needed?               yes +
  OVERWRITE same or newer versions?                  no +
  VERIFY install and check file sizes?               no +
  DETAILED output?                                   no +
  Process multiple volumes?                          yes +
  ACCEPT new license agreements?                 yes +
  Preview new LICENSE agreements?                    no +

WPAR Management
  Perform Operation in Global Environment            yes +
  Perform Operation on Detached WPARs                no +
    Detached WPAR Names                             [_all_wpars] +
  Remount Installation Device in WPARs               yes +
  Alternate WPAR Installation Device                 []

```

Figure 7-6 The powerscStd.vtpm from the PowerSC CD



## 7.4.2 Installing the verifier

Complete the following steps to install the verifier:

1. Use the fileset from the PowerSC Standard Edition 1.1.2.0 media.
2. Install the `openpts.verifier` by using the `smit` or `installp` command. This package installs the command-line version and graphical interface version of the verifier (see Figure 7-7). The required user input is shown in bold.

Install and Update from ALL Available Software	
Type or select values in entry fields. Press Enter AFTER making all desired changes.	
	[Entry Fields]
* INPUT device / directory for software	.
* SOFTWARE to install	<b>[openpts.verifier]</b> +
PREVIEW only? (install operation will NOT occur)	no +
COMMIT software updates?	yes +
<b>SAVE replaced files?</b>	<b>yes</b> +
AUTOMATICALLY install requisite software?	yes +
EXTEND file systems if space needed?	yes +
OVERWRITE same or newer versions?	no +
VERIFY install and check file sizes?	no +
DETAILED output?	no +
Process multiple volumes?	yes +
<b>ACCEPT new license agreements?</b>	<b>yes</b> +
Preview new LICENSE agreements?	no +
WPAR Management	
Perform Operation in Global Environment	yes +
Perform Operation on Detached WPARs	no +
Detached WPAR Names	<b>[_all_wpars]</b> +
Remount Installation Device in WPARs	yes +
Alternate WPAR Installation Device	<b>[]</b>

Figure 7-7 The `openpts.verifier` from the AIX expansion pack

## 7.5 Working with Trusted Boot

The following sections provide a guide and examples of actions that can be performed with PowerSC Trusted Boot after the base installation is completed.

### 7.5.1 Configuring SSH

The attestation request is sent through SSH. To permit SSH connections without a password, the verifier's certificate must be installed on the collector. To set up the verifier's certificate on the collector's system, complete the following steps:

1. On the verifier, run the following command:

```
ssh-keygen # No passphrase
```

The output is shown in Example 7-1 on page 260.

---

*Example 7-1 The ssh-keygen command execution*

---

```
verifier> # ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (//.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in //.ssh/id_rsa.
Your public key has been saved in //.ssh/id_rsa.pub.
The key fingerprint is:
8f:34:9d:6c:fa:2e:a9:e5:a7:ea:67:be:cf:68:58:29 root@lpar4
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|          o .
|         S.=
|        E.O*
|       ++..
|      .O*+.
|     .+B*B=
|
+-----+
verifier> #
```

---

2. On the verifier, run the following command:

```
scp ~/.ssh/id_rsa.pub <collector>:/tmp
```

The output is shown in Example 7-2.

---

*Example 7-2 The scp ~/.ssh/id\_rsa.pub <collector>:/tmp execution*

---

```
verifier> # scp ~/.ssh/id_rsa.pub collector:/tmp
root@collector's password:
id_rsa.pub
100% 392    0.4KB/s   00:00
verifier> #
```

---

**Important:** Use the **ssh-keygen** command only once on the verifier. Otherwise, the private key (`id_rsa`) and public key (`id_rsa.pub`) are replaced. Use the same public key for all clients.

3. On the collector, run the following command:

```
cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
```

The output is shown in Example 7-3.

---

*Example 7-3 The cat /tmp/id\_rsa.pub >> ~/.ssh/authorized\_keys execution*

---

```
collector> # cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
collector> #
```

---

## 7.5.2 Enabling Virtual Trusted Platform Module (vTPM)

vTPM must be turned on for Trusted Boot to work properly with the collector and verifier. Complete the following steps to enable vTPM:

1. Shut down the LPAR by using the **shutdown** command on the chosen LPAR.
2. Access the LPAR's partition properties by right-clicking the chosen LPAR from the HMC. Click **Properties** to open the window (see Figure 7-8).

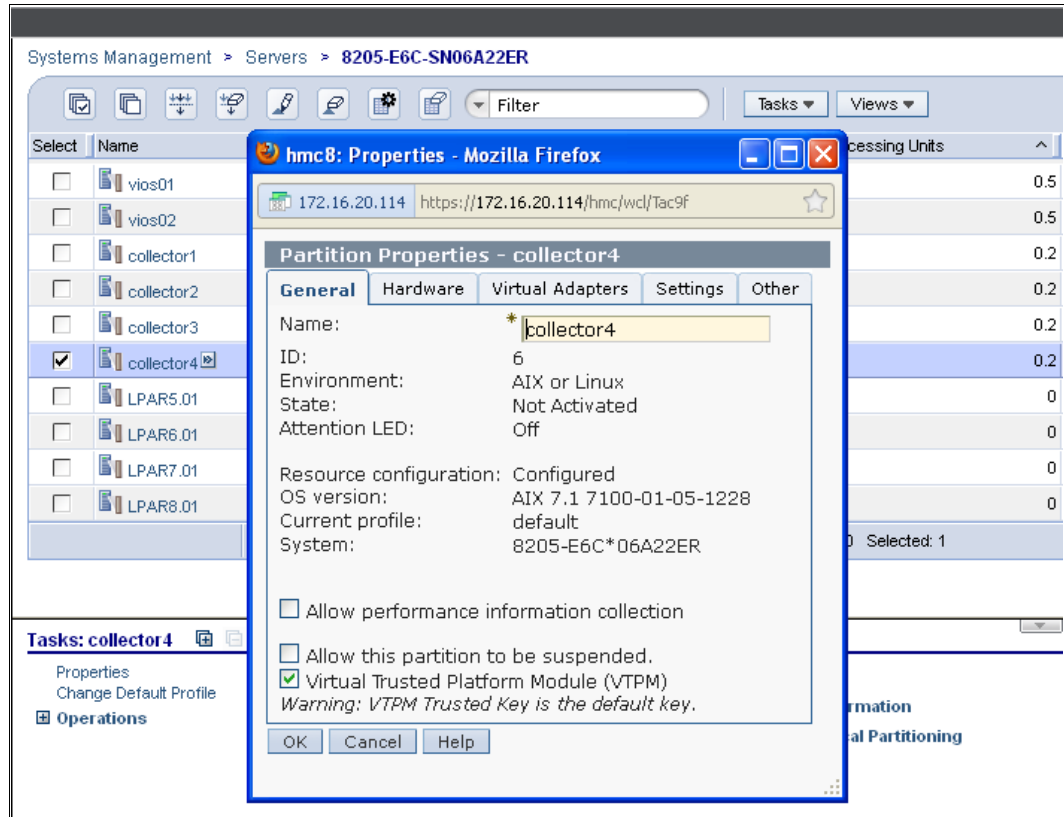


Figure 7-8 LPAR Properties window from the HMC

3. Select the **General** tab.
4. Select **Virtual Trusted Platform Module (vTPM)**.
5. Click **OK**.
6. Power on the LPAR.
7. Verify that vTPM is available on the chosen LPAR by running the following command:

```
lsdev | grep vtpm
```

The output is shown in Example 7-4.

*Example 7-4 Verifying vTPM on the LPAR*

```
collector> # lsdev | grep vtpm
vtpm0      Available      Virtual Trusted Platform Module (VTPM)
collector> #
```

### 7.5.3 Enrolling a system

Enrolling a system is the process of providing an initial set of measurements to the verifier. This set of measurements forms the basis for subsequent attestation requests.

To initialize the openpts collector for the first time, run the **ptsc** command from the collector:

```
ptsc -i
```

The output is shown in Example 7-5.

*Example 7-5 Initializing ptsc on the collector*

---

```
collector> # ptsc -i
Sign key location: SYSTEM
Generate uuid: 8572995a-f865-11e1-a0d7-2ae6a0138902
Generate UUID (for RM): 859909fa-f865-11e1-a0d7-2ae6a0138902
level 0 Reference Manifest :
/var/ptsc/859909fa-f865-11e1-a0d7-2ae6a0138902/rm0.xml
level 1 Reference Manifest :
/var/ptsc/859909fa-f865-11e1-a0d7-2ae6a0138902/rm1.xml

ptscd has successfully initialized!
collector> #
```

---

To enroll a system from the command line, use the following command from the verifier:

```
openpts -i <hostname>
```

The output is shown in Example 7-6.

*Example 7-6 Enrolling a system from command line*

---

```
verifier> # openpts -i collector
Target: collector
Collector UUID: 8572995a-f865-11e1-a0d7-2ae6a0138902
Manifest UUID: 859909fa-f865-11e1-a0d7-2ae6a0138902
Manifest[0]:
//.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902//859909fa-f865-11e1-a0d7-2ae6a0138902/rm0.xml
Manifest[1]:
//.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902//859909fa-f865-11e1-a0d7-2ae6a0138902/rm1.xml
Configuration: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/target.conf
Validation policy: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/policy.conf
verifier> #
```

---

Information about the enrolled partition is in the `$HOME/.openpts` directory. Each new partition is assigned with a unique identifier during the enrollment process. Information that relates to the enrolled partitions is stored in the directory that corresponds to the unique ID.

To enroll a system from the graphical interface, complete the following steps:

1. Start the graphical GUI by using the `/opt/ibm/openpts_gui/openpts_GUI.sh` command.
2. Select **Add & Enroll** from the navigation menu on the left.
3. Enter the host name and the SSH credentials of the system.
4. Click **Add & Enroll**.

## 7.5.4 Attesting a system

To query the integrity of a system boot, use the following command from the verifier:

```
openpts <hostname>
```

The output is shown in Example 7-7.

*Example 7-7 Attesting a system*

---

```
verifier> # openpts collector
Target: collector
Collector UUID: 8572995a-f865-11e1-a0d7-2ae6a0138902 (date: 2012-09-06-20:57:51)
Manifest UUID: 859909fa-f865-11e1-a0d7-2ae6a0138902 (date: 2012-09-06-20:57:51)
username(ssh): default
port(ssh): default
policy file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/policy.conf
property file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/vr.properties
integrity: valid
verifier> #
```

---

To attest a system from the graphical user interface, complete the following steps:

1. Start the GUI by using the `/opt/ibm/openpts_gui/openpts_GUI.sh` command.
2. Select a category from the navigation menu on the left under **All**.
3. Select one or more systems to attest.
4. Click **Attest**.

## 7.5.5 Attesting multiple systems

It is a simple task to attest several machines in one environment. However, if the environment has numerous servers, this task can become laborious. To simplify the attestation of multiple LPARs, you can use a script, as shown in Example 7-8.

*Example 7-8 The tboot\_csv\_output.ksh script*

---

```
#!/bin/ksh
#####
#
# Name of script: tboot_csv.ksh
#
# Info:script to generate a CSV output containing information about the
# attestation of all clients
#
# Author: Fernando Iury Alves Costa
# Creation date: 18/09/2012
#####
echo "HOSTNAME,INTEGRITY";
CLIENTS=$(openpts -D | grep hostname: | awk '{ print $NF }');
for i in $CLIENTS; do
    LINE=$(echo n | openpts -v $i 2> /dev/null | egrep "Target:|integrity:" |
awk '{ print $NF }');
    echo $LINE | awk '{ print $1","$2 }';
done
```

---

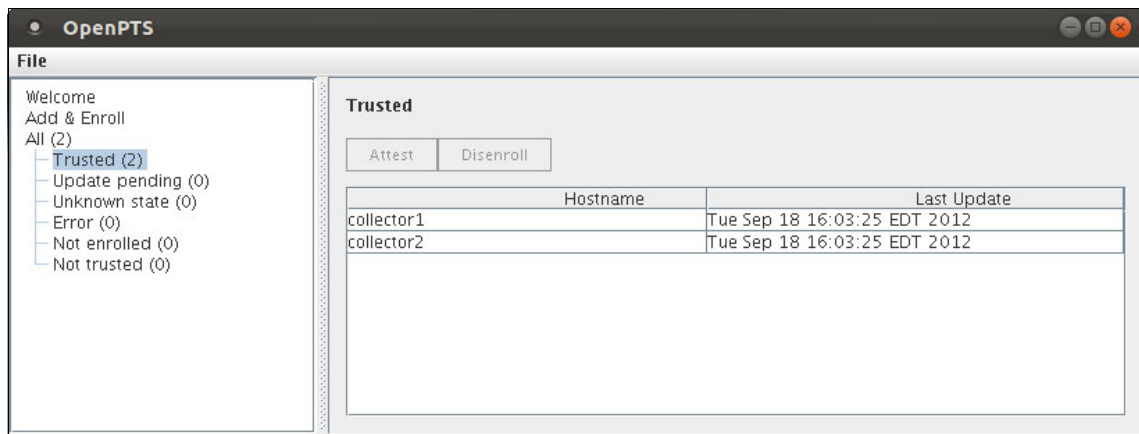
The comma-separated values (csv) format output that is generated by running this script on the *verifier*, as shown in Example 7-9.

*Example 7-9 The tboot\_csv.ksh output sample*

```
verifier> # ./tboot_csv.ksh
HOSTNAME,INTEGRITY
collector1,valid
collector2,invalid
collector3,valid
collector4,valid
verifier> #
```

## 7.5.6 Simulating a failure

We simulate a boot image change in our environment. Figure 7-9 shows all servers as trusted servers in our initial simulated environment.



*Figure 7-9 Initial environment*

Complete the following steps in our simulation:

1. Change the boot image on collector2 by using the **bosboot** command, as shown in Example 7-10.

*Example 7-10 Changing the boot image on collector2*

```
collector2> # bosboot -ad /dev/ipldevice

bosboot: Boot image is 53276 512 byte blocks.
53504+0 records in.
53504+0 records out.
collector2> #
```

2. Check the environment after the change by refreshing the status of all servers. Then, use the openPTS GUI to browse to **File** → **Refresh** from the drop-down menu.

Figure 7-10 shows the updated status of our simulated environment after the change and refresh.

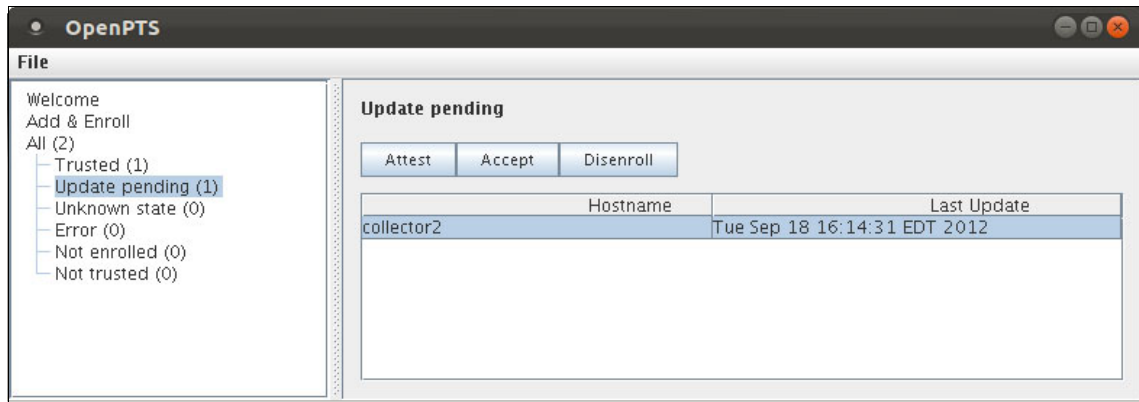


Figure 7-10 The environment after changing the boot image

You can also check the change by using the **openpts** command on the command line, as shown in Example 7-11.

*Example 7-11 Checking the client collector2 by using the command line*

```
verifier> # openpts collector2
Target: collector2
Collector UUID: 8572995a-f865-11e1-a0d7-2ae6a0138902 (date:
2012-09-06-20:57:51)
Manifest UUID: c61a90a4-01d2-11e2-b16d-2ae6a0138902 (date: 2012-09-18-20:52:35)
username(ssh): default
port(ssh): default
policy file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/policy.conf
property file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/vr.properties
integrity: valid
-----
New Manifest UUID: 5hcwTAHXEeKAPyrmoBOJAg== (date: 2012-09-18-21:29:16)
A new reference manifest exists. Update? [Y/n]
n
Keep current manifest
verifier> #
```

3. Reboot collector2 to force a boot with an untrusted boot image. To reboot the client, use the **shutdown** command, as shown in Example 7-12.

*Example 7-12 Rebooting the client to use the changed boot image*

```
collector2> # shutdown -Fr

SHUTDOWN PROGRAM
Tue Sep 18 16:18:22 EDT 2012
Wait for 'Rebooting...' before stopping.
Error reporting has stopped.
Advanced Accounting has stopped...
Process accounting has stopped.
nfs_clean: Stopping NFS/NIS Daemons
0513-004 The Subsystem or Group, nfsd, is currently inoperative.
0513-044 The biod Subsystem was requested to stop.
0513-044 The rpc.lockd Subsystem was requested to stop.
```

```

...
Unmounting the file systems...
Unmounting the file systems...
Bringing down network interfaces: en0 lo0
Sep 18 16:19:58 portmap: terminating on signal.
Rebooting . . .

```

4. Check the environment after the reboot by refreshing the status of all servers. Use the openPTS GUI to browse to **File** → **Refresh** from the drop-down menu.

Figure 7-11 shows the status of our simulated environment after the reboot.

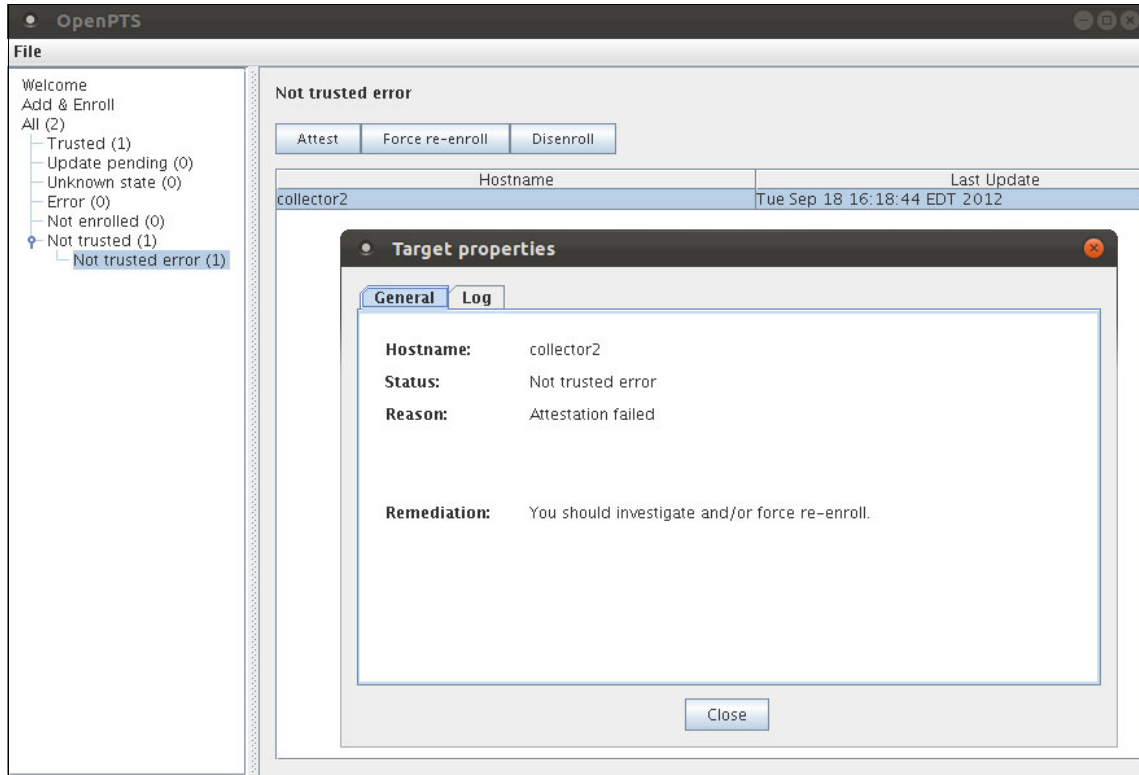


Figure 7-11 The environment after you reboot

You can also check the change by using the command **openpts** on the command line, as shown in Example 7-13.

*Example 7-13 Checking the client after the reboot by using the command line*

```

verifier> # openpts collector2
Target: collector2
Collector UUID: 8572995a-f865-11e1-a0d7-2ae6a0138902 (date:
2012-09-06-20:57:51)
Manifest UUID: 859909fa-f865-11e1-a0d7-2ae6a0138902 (date: 2012-09-06-20:57:51)
username(ssh): default
port(ssh): default
policy file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/policy.conf
property file: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/vr.properties
integrity: invalid
  0 Missing Reference Manifest (RM)
  1 Collector hostname = collector2
  2 Collector UUID = 8572995a-f865-11e1-a0d7-2ae6a0138902

```



```

3 Collector RM UUID = ecc902b2-01cd-11e2-9cbd-2ae6a0138902
4 Missing Reference Manifest directory =
//.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/ecc902b2-01cd-11e2-9cbd-2ae6a01
38902
5 Collector is using another Reference Manifest (RM)
6 Collector hostname = collector2
7 Collector UUID = 8572995a-f865-11e1-a0d7-2ae6a0138902
8 Previous RM UUID = 859909fa-f865-11e1-a0d7-2ae6a0138902, timestamp =
2012-09-06-20:57:51
9 Current RM UUID = ecc902b2-01cd-11e2-9cbd-2ae6a0138902, timestamp =
2012-09-18-20:17:52
10 [RM00-PCR03-PCR3_START] IR validation by RM has failed
11 [RM00-PCR04-EV_EVENT_IPL_LOOP_0] IR validation by RM has failed
12 [QUOTE] verification of PCR Composite has failed, (tskd - bad FSM
configuration in /etc/ptsc.conf)
13 [POLICY-L004] tpm.quote.pcr.3 is G0rSra54/+uuwiS/cI3YJjLiuPs=, not
3ehhn8LqbXgDQ6nP/GdHltKcisw=
14 [POLICY-L014] ibm.pfw.pcr.3.integrity is missing
15 [POLICY-L015] ibm.pfw.pcr.4.integrity is missing
16 [POLICY-L020] tpm.quote.pcrs is invalid, not valid
verifier> #

```

---

##### 5. Make the server trusted again.

After verifying why the server became untrusted and validating that it is still compliant, re-enroll it by completing the following steps:

- Click **Not trusted** from the navigation menu on the left in the openPTS GUI.
- Click the server name from the list of untrusted servers.
- Click **Force re-enroll**.
- Click **OK** to confirm.

You can also make the server trusted again by way of the command line by using the **openpts** command on the verifier, as shown in Example 7-14.

*Example 7-14 Making the server trusted again by using the command line*

---

```

verifier> # openpts -f -i collector2
Target: collector2
Collector UUID: 8572995a-f865-11e1-a0d7-2ae6a0138902
Manifest UUID: e617304c-01d7-11e2-803f-2ae6a0138902
Manifest[0]:
//.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902//e617304c-01d7-11e2-803f-2ae6a0
138902/rm0.xml
Manifest[1]:
//.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902//e617304c-01d7-11e2-803f-2ae6a0
138902/rm1.xml
Configuration: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/target.conf
Validation policy: //.openpts/8572995a-f865-11e1-a0d7-2ae6a0138902/policy.conf
verifier> #

```

---

## 7.6 Troubleshooting

The path to a successful attestation requires all components of Trusted Boot to work in harmony. When something goes wrong, it is relatively easy to discover the root cause of a problem in most cases. For example, the LPAR configuration might change, which leads to outdated reference information.

You must first consult the information center for Trusted Boot, which contains a brief troubleshooting section that covers the most obvious attestation issues. This guide is intended to help you in cases where a deeper analysis might be needed, and the case requires a deeper understanding of what might go wrong with each component.

In general, we can classify attestation problems into two types:

- ▶ Failed attestation is where OpenPTS fails to complete the attestation. Therefore, you cannot verify the integrity of the collector LPAR. For example, one such failed attestation might occur if the vTPM device driver is not installed on the collector.
- ▶ Integrity invalid is where OpenPTS completed the attestation successfully, but reported an invalid integrity. For example, the LPAR configuration might change after it was first enrolled.

This troubleshooting section first describes some of the pitfalls that might be encountered during the configuration and normal operation of a Trusted Boot LPAR and how to avoid them. The following sections describe some of the tools and information to help you successfully debug attestation issues. The term *collector* refers to the Trusted Boot/vTPM-enabled LPAR; the term *verifier* refers to the LPAR that performs the attestation.

### 7.6.1 Common problems

Common problems with attestations are divided into two classes: failed attestations (see Table 7-1) and integrity invalid (see Table 7-2 on page 269). It is assumed that your installed version of AIX supports vTPMs.

Table 7-1 Failed attestations troubleshooting

Root cause	Key indicators	Fix
vTPM missing from LPAR configuration	On the collector, vTPM device is missing from /dev	From the HMC, shut down the LPAR, enable a vTPM, and reboot.
vTPM device driver not installed	On the collector, vTPM device is missing from /dev	Install package powerscStd.vtpm.
vTPM device removed by DLPAR operation	On the collector, vTPM device is missing from /dev	If the LPAR runs on a system that supports vTPM, from the HMC, shut down the LPAR and activate with the profile.
openpts.collector package not installed	On the collector, /usr/bin/ptsc is missing from file system	From AIX Base pack, install the openpts.collector package.

Root cause	Key indicators	Fix
AIX reinstalled on vTPM-enabled LPAR	On the collector, the file <code>/var/tss/lib/tpm/system.data</code> is missing or empty. This file is used by the <code>tcsd</code> daemon to store important keys that are used to communicate with the vTPM. If destroyed, it cannot be re-created without destroying the vTPM.	From the HMC, shut down the LPAR, disable the vTPM, re-enable the vTPM, and activate the LPAR.
SSH not installed on vTPM-enabled LPAR	On the verifier, SSH fails to connect to the collector LPAR.	Install SSH on the collector.
SSH not installed on verifier	On the verifier, SSH fails to connect to the collector LPAR.	Install SSH on the verifier.
Insufficient disk space on the verifier	The home directory that corresponds to the user performing the attestation might not have enough space to store the OpenPTS verifier configuration and log files.	Increase the size of the file system on which the user's home directory resides.

Table 7-2 Integrity invalid troubleshooting

Root cause	Key indicators	Fix
User entered input at the partition firmware prompt before booting	The verifier fails to validate PCR 5	Reboot the collector the same way as you did originally when the original reference information is collected by the verifier. In most cases, reboot normally without entering the SMS menu or the Firmware Prompt.

## 7.6.2 Diagnosis

Many ways are available to try to diagnose the cause of a failed attestation or integrity valid issue, such as checking that important configuration files are in the correct state, viewing the output of various log files, and running various tools on the LPAR.

### Configuration files

In general, failed attestations often occur because of configuration issues where the vTPM is not set up correctly or the correct software packages are not installed. However, occasionally it might be because of an issue with the configuration files on the collector, as shown in Table 7-3 on page 270.

Table 7-3 Configuration files

File	Package	Description
/etc/ptsc.conf	openpts.collector	The configuration file used by the <b>ptsc</b> program used as part of attestation. For example, it provides information to <b>ptsc</b> on where to find the model files that describe how to interpret the contents of each PCR.
/etc/security/tss/tcsd.conf	bos.rte.security	The configuration file used by the <b>tcsd</b> daemon, which is started as part of the AIX boot process. This file includes the list of PCRs about which Trusted Boot is concerned.
/usr/share/openpts/models/bm_power_pcrXX.uml (where XX is a number 0 - 10)	openpts.collector	These configuration files are used by <b>ptsc</b> to describe how to interpret the contents of each PCR that it analyses as part of attestation.

These files are not expected to change during the life of the collector because they are tuned to work on AIX LPARs correctly for all possible LPAR configurations. If these files are not identical to the files that are in the corresponding packages, the attestations might not work correctly.

## Log files

When you debug failed attestation or integrity invalid issues, it is sometimes necessary to investigate the contents of various log files on the collector and verifier (see Table 7-4).

Table 7-4 Log files

File	Location	Description
/var/adm/ras/bootlog	Collector	This file is the AIX initial boot log, which is generated early in the boot process. It shows the vTPM device (for example, /dev/vtpm0) being initialized correctly. If not, a problem likely exists loading the device driver.
/var/adm/ras/conslog	Collector	This file is the output from many of the scripts and programs that are run later in the boot process. This file shows output, such as “Waiting for tcsd to become ready... ok”, which means the tcsd daemon started successfully.
/var/adm/ras/trousers/tcsd.out	Collector	Shows the output from the tcsd daemon.
/var/adm/ras/trousers/tcsd.err	Collector	Shows any errors from the tcsd daemon. If tcsd started successfully, this file is empty.

File	Location	Description
/var/adm/ras/openpts/log	Collector	This log file is generated by <b>ptsc</b> on the first invocation and can be read by using the <b>alog -f /var/adm/ras/openpts/log -o   more</b> command. It is written to whenever a self-test, enrollment, or attestation is performed. It is not possible to describe the format of this log file in detail here. Generally, if the log file contains entries that begin with [ERROR], <b>ptsc</b> found a problem. When problems occur, it might indicate what caused the problem, such as a PCR value that is inconsistent.
/var/adm/ras/openpts/bosupdate.log	Collector	This log file is generated as part of the AIX update process. It is generated by bosboot when the AIX boot image is changed.
\$HOME/.openpts/log (where \$HOME is the home directory of the user that performs the attestations)	Verifier	This log is generated by the <b>openpts</b> executable when you perform enrollments or attestations. On AIX LPARs, this file can be read by using the <b>alog alog -f \$HOME/.openpts/log -o   more</b> command. On Linux LPARs, this file is a normal text file.

## Tools

The following tools are available on the collector that can help you address some of the issues that might occur during attestation:

- TPM version information command:

```
/usr/sbin/tpm_version
```

When run, this command prints version information that is obtained from the vTPM. If this command successfully displays the version information, it means that the collector has a working vTPM and vTPM device driver, and the tcstd daemon is running. Before you debug any issue, the user must run this tool first to ensure that it is possible to communicate with the vTPM.

- Event log:

```
/usr/bin/iml2text
```

This command talks to the tcstd daemon to retrieve the event log, which is a record of all Trusted Boot events that occurred during an AIX boot. The detail that is contained in this log is used to construct the final PCR values that are tested during attestation.

If the command fails to produce any event log output, it might mean that a problem exists communicating with the tcstd daemon, or it might indicate that a problem exists generating the event log at boot time. If the command fails, you can check that the following files are present on the collector:

- /var/adm/ras/trustedboot.log

This file is a softcopy of the event log that is generated as part of the boot process and loaded by tcstd at startup. It is initialized by using the **/usr/lib/tpm/bin/geteventlog** command early in the boot process.

- /var/adm/ras/teboot.log

This file contains the event log entries that correspond to the Trusted Execution database, which are generated by the script **/etc/rc.teboot**.

If either file is missing or empty, it might help to indicate the root of the problem.

## 7.7 Conclusion

This chapter provided an implementation of the Trusted Computing Group *Trusted Boot* mechanism (based around a virtual TPM) that enables a partition owner to prove that their partition boot process is not tampered with.

A TPM is a security device that is defined by the TCG that is used to securely maintain a record of the boot process of a system. This secure record can then be used to conditionally release encryption keys (if the record matches an expected value) and prove to a third party that the boot of a system ran correctly.



## Trusted Firewall

Within the IBM PowerSC Standard Edition, the Trusted Firewall component can provide network firewall services within the local virtualized server to control network traffic between logical partitions (LPARs). LPARs that run on the same server can communicate without going to an external firewall.

Trusted Firewall can help you improve performance and reduce network resource consumption. It also can eliminate the need to use the external network for LPAR-to-LPAR traffic when these LPARs run on the same server. Trusted Firewall offers the following benefits:

- ▶ Saves time and network resources by never going out to a physical network interface controller (NIC) or to an external firewall.
- ▶ Keeps virtual traffic local, which improves performance and saves physical resources.
- ▶ Reduces traffic on the external firewalls significantly.

This chapter describes the Trusted Firewall component architecture, including implementation and deployment, installation details, how-to manage it, and troubleshooting information.

This chapter contains the following topics:

- ▶ 8.1, “Component architecture” on page 274
- ▶ 8.2, “Detailed implementation” on page 280
- ▶ 8.3, “Deployment considerations” on page 288
- ▶ 8.4, “Installation” on page 288
- ▶ 8.5, “Working with Trusted Firewall” on page 292
- ▶ 8.6, “Troubleshooting Trusted Firewall” on page 312
- ▶ 8.7, “Conclusion” on page 313

## 8.1 Component architecture

Trusted Firewall provides a network firewall to filter cross-virtual LAN (VLAN) communications between LPARs on the same server. Trusted Firewall is implemented within the Virtual I/O Server (VIOS). Therefore, it protects AIX, IBM i, and Linux traffic equally.

With Virtual I/O Server Version 2.2.1.4, or later, you can configure and manage the Trusted Firewall feature. By using this feature, LPARs on different VLANs of the same server can communicate through the shared Ethernet adapters. This section describes the architecture of Trusted Firewall in more detail. It introduces the important concepts that are required to build an intuitive understanding of the capabilities of Trusted Firewall and how it works.

### 8.1.1 Firewall technologies

The most basic firewall system separates two IP networks. The configuration of the firewall specifies the connections that are permitted.

The common firewall usage is to control any traffic flow between the secure and non-secure networks. You also can use a firewall to secure one internal network from another on an intranet network.

The following main firewall technologies are available:

- ▶ Packet filter

The firewall inspects the packets and filters them based only on the information that is contained in the packet. The packet information is checked against filtering rules to decide whether the packet is dropped or routed to its destination. The packet firewall ignores the network flow or the connection state.

- ▶ Stateful inspection

This firewall is a step further more secure than packet filter firewall. Packets are recognized as part (or not) of an established authorized connection. Authorized connections as their associated connection state are kept in a state table, which tracks the communication channel.

- ▶ Application-layer firewall

The firewall works at the Application layer of the open systems interconnection (OSI) communication stack. Total knowledge exists of the communication protocol, which authorizes a deep analysis of the packet data content. These firewalls can intercept offensive content, viruses, certain websites, and malformed packets.

With PowerSC Standard Edition, Trusted Firewall Version 1.1.2 provides a packet filter firewall, which is also called a *network firewall*.

Packet firewalls commonly filter traffic by IP address and by TCP or User Datagram Protocol (UDP) ports. They also incorporate the Internet Control Message Protocol (ICMP). Trusted Firewall supports IPv4 and IPv6 protocols. Because the IP packet header is inspected, a *packet filter firewall* works at Layer 3 of the OSI Stack.

Based on the filtering rules that are configured into the firewall, the firewall typically blocks these addresses and ports (Deny) unless they are explicitly allowed (Permit).

In a packet-filtering firewall, the firewall checks usually for five characteristics. Because Trusted Firewall filters inter-VLAN communications, it checks another characteristic: the VLAN tag.



A packet is inspected by using the following seven attributes:

- ▶ Source IP address
- ▶ Source port number
- ▶ Destination IP address
- ▶ Destination port number
- ▶ IP protocol: UDP, TCP, ICMP, and ICMPV6
- ▶ VLAN ID of source IP address
- ▶ VLAN ID of destination IP address

The next section describes how Trusted Firewall manages the denied IP packets.

## 8.1.2 Deny and permit

When Trusted Firewall receives an IP packet that does not qualify for VLAN crossing, it denies the packet.

The packet is denied from going through Trusted Firewall directly to the destination VLAN in the frame. Therefore, the packet returns to its default route. The default route is through the external network with its associated firewalls or Intrusion Prevention Systems/Intrusion Detection Systems (IPS/IDS) appliances.

Trusted Firewall sends the denied packet back to its original Shared Ethernet Adapter (SEA). This SEA then sends the packet by the physical adapter to the external network. The packet is denied the use of the cross-VLAN capability of Trusted Firewall.

The Deny decision, which is made by Trusted Firewall, means that the IP packet is ineligible based on the filtering rules table to route cross-VLAN.

Therefore, the goal of Trusted Firewall Version 1.1.2 is to propose the following paths:

- ▶ A short path within the frame through the Virtual I/O Servers to IP packets that do not need to be inspected (Permitted Packets)
- ▶ A default path (the expected routing to the external network) to IP packets that still require an examination and inspection (Denied Packets)

**Consideration:** When Trusted Firewall authorizes packets to go back into the SEA devices to bridge the VLANs, it opens the shortest path for the permitted packets. On the other side, the denied packets are sent back to the SEA devices to be exposed to external firewalls and IPS/IDS appliances. The denied packets include an inspection requirement.

For a system administrator, denying this behavior results in the following benefits:

- ▶ Trusted Firewall saves network bandwidth and physical network cards bandwidth only. It proposes the shortest path to cross-VLAN secured traffic.
- ▶ Because Trusted Firewall does not alter the IP traffic in any way, you do not need extensive logging of the Trusted Firewall activity. In Version 1.1.2, Trusted Firewall does not provide any logging feature.
- ▶ The filtering rules definition is simpler and easier. The system administrator focuses on only the cross-VLAN traffic. The systems administrator wants to grant the shortest route, which is the secured traffic between VLANs.
- ▶ The Trusted Firewall is safe to implement and nondisruptive to the network traffic. The deny behavior does not alter the IP traffic, only the routing paths.

The scope of Trusted Firewall is to grant a short path within the frame to secured cross-VLAN traffic (Permitted Traffic). Unsecured cross-VLAN traffic is denied and exposed to the external firewall and IPS appliances (Denied Traffic).

Packet filtering concepts are described next.

### 8.1.3 Packet filtering rules

IP packet filtering is the core protection mechanism of a firewall. *Packet filters* are sets of rules that limit IP packet flow into or out of two VLANs within the same frame.

With Trusted Firewall, the IP packets that enter the Virtual I/O Servers (outbound) and the IP packets that exit the Virtual I/O Servers (inbound-to-external) are not filtered.

**Important:** The Trusted Firewall filtering scope is limited to only the IP flow between VLANs that are defined in the same server frame. The *outbound* traffic and the *inbound-to-external* traffic of the server frame are not filtered.

A filtering rule for Trusted Firewall applies to both directions of the IP flow (flow from IP source to IP destination and from IP destination to IP source). However, it is advised that you create filtering rules for each IP flow direction.

Filter rules can control many aspects of communications. In Trusted Firewall Version 1.1.2, the filtering scope is limited to source and destination addresses, source and destination VLANs, port numbers, and protocol.

### 8.1.4 Security policies

As a firewall administrator, you must define security policies that determine which packets Trusted Firewall permits between the VLANs of your frame. You must define the cross-VLAN secured traffic (Permitted Traffic) and the cross-VLAN traffic that requires network inspection and filtering (Denied Traffic).

By default, Trusted Firewall denies all packets and sends them back to the SEAs to be externally routed. This network behavior is expected and it is achieved by the Trusted Firewall default rule (see Table 8-1).

Table 8-1 Trusted Firewall default rule

Action	VLAN ID source	VLAN ID destination	From IP	Source Port	To IP	Destination port	Protocol type
Deny	Any	Any	Any	Any	Any	Any	Any

**Important:** By default, Trusted Firewall denies all packets to cross VLANs within the frame. All VLAN traffic is exposed to the external network to be inspected.

From the external network, the firewall administrator tasks follow this path:

1. Define only the cross-VLAN secured traffic: Permit first.
2. Apply the default Denied Traffic rule to the rest of the traffic.

In terms of system administration (rules creation and rules maintenance), this approach works if the frame handles little secured traffic. The resulting rules table is small and easily maintained, controlled, and audited. Also, these goals are key in an efficient security framework.

However, if the frame contains only VLANs that belong to the same level of security where most cross-VLAN traffic is secured, the system administrator task is more difficult. The rules table is highly populated, difficult to maintain, and prone to maintenance errors. With this approach, it is difficult to achieve efficiency in terms of security.

For more secured traffic, the firewall administrator must use the following approach:

1. Define only the non-secured traffic (the packets to be inspected): Deny first.
2. Define generic cross-VLAN authorizations without details.

Table 8-2, Table 8-3 on page 278, Table 8-4 on page 278, and Table 8-5 on page 279 are a simple planning tool for administrators to gather data about the VLAN connections to allow through the Trusted Firewall.

Table 8-2 is a filter rules table that is designed as Permit First. It first authorizes the detailed secured traffic and denies all remaining traffic from the frame in generic form with the default rule.

Table 8-2 Filter rules table that is designed as Permit First

Action	VLAN ID source	VLAN ID destination	From IP	Source port	To IP	Destination port	Protocol type
Permit	VLAN X	VLAN Y	IP address source	Port number	IP address destination	Port number	Protocol
Permit	VLAN X	VLAN Y	IP address source	Port number	IP address destination	Port number	Protocol
Permit	VLAN Z	VLAN U					Any
Permit	VLAN A	VLAN B	IP address source	Port number	IP destination source	Port number	Protocol
Permit							
Deny	Any	Any	Any	Any	Any	Any	Any

The Permit First table features a special singularity. The following example is based on the Permit First table and shown in Table 8-3:

- ▶ VLAN Z - VLAN U can exchange their packets freely (*Permit First*).
- ▶ Exception: Two LPARs (Z-LIFE and U-BOAT) cannot talk together without packet inspection.

In this example, the deny rule between Z-LIFE and U-BOAT must be entered before the generic permit authorization of VLAN Z - VLAN U, as listed in Table 8-3.

Table 8-3 Singularities in a Permit First table

Action	VLAN ID source	VLAN ID destination	From IP	Source port	To IP	Destination port	Protocol type
Permit							Protocol
<b>Deny</b>	VLAN Z	VLAN U	IP LIFE		IP BOAT		<b>Any</b>
Permit	VLAN Z	VLAN U					Any
Permit							
<b>Deny</b>	Any	Any	Any	Any	Any	Any	Any

Table 8-4 proposes which rule first denies the detailed non-secured traffic (to be sent externally) and authorizes all remaining traffic from the frame in a generic form. Trusted Firewall does not propose a generic rule to authorize all the traffic. Therefore, a generic authorization rule VLAN-per-VLAN connection must be defined. In Table 8-4, VLAN C - VLAN D and VLAN E - VLAN F can freely exchange their IP packets within the frame without external inspection or further control.

Table 8-4 Filter rules table that is designed as Deny First

Action	VLAN ID source	VLAN ID destination	From IP	Source port	To IP	Destination port	Protocol type
Deny	VLAN X	VLAN Y	IP address source	Port number	IP address destination	Port number	Protocol
Deny	VLAN X	VLAN Y	IP address source	Port number	IP address destination	Port number	Protocol
Deny	VLAN Z	VLAN U					Any
Deny	VLAN A	VLAN B	IP address source	Port number	IP destination source	Port number	Protocol
<b>Permit</b>	VLAN C	VLAN D	Any	Any	Any	Any	<b>Any</b>
<b>Permit</b>	VLAN E	VLAN F	Any	Any	Any	Any	<b>Any</b>

The Deny First table also features a special singularity. The following example is based on Table 8-4 on page 278 and Table 8-5:

- ▶ VLAN Z - VLAN U are denied the ability to communicate directly (Deny First).
- ▶ Exception: Two LPARs (Z-LIFE and U-BOAT) can talk together without packet inspection.

In this example, the permit rule between Z-LIFE and U-BOAT must be entered before the generic deny authorization of VLAN Z - VLAN U. Table 8-5 lists the example in this table extract.

*Table 8-5 Singularity in a Deny First table*

Action	VLAN ID source	VLAN ID destination	From IP	Source port	To IP	Destination port	Protocol type
Deny							
<b>Permit</b>	VLAN Z	VLAN U	IP LIFE		IP BOAT		<b>Any</b>
Deny	VLAN Z	VLAN U					Any
Permit	VLAN C	VLAN D					<b>Any</b>
Permit	VLAN E	VLAN F					<b>Any</b>

In both approaches (Permit First and Deny First), the order of the rules for singularities is important. Filtering rules tables for the firewall often are ordered by matching scan, and Trusted Firewall is no exception. If a packet is validated by one filter rule, the associated rule action (Deny or Permit) is taken immediately, and the rules scan stops. If no filter rule can validate the packet, the default action is to deny the packet.

**Consideration:** The filter rules table must be ordered in respect to the expected scan order. Rules are processed in order from the top to the bottom of the rules file. By default, Trusted Firewall uses the first filter rule that matches the packet that it is evaluating.

The study of the VLANs of the frame and the security level that is required in their connections can help you to choose between the Permit First methodology or the Deny First methodology to create the filter rules table. It is suggested to prioritize the approach that results in fewer rules. Fewer rules mean less work to administer and maintain and it is easier to monitor and control.

When these tables are completed, the security policies of the network traffic within the frame are established. Then, the firewall administrator can use the Trusted Firewall rules management commands to activate these policies in the Virtual I/O Server.

For more information about these commands, see 8.5.2, “Configuring the filter rules” on page 299.

How Trusted Firewall is implemented within the Virtual I/O Server is described next.

## 8.2 Detailed implementation

The Shared Ethernet Adapter (SEA) is a VLAN Trunk that ensures the tagging of the untagged traffic and the packet distribution based on the packet VLAN ID tag. It acts at OSI Layer 2, as shown in Figure 8-1.

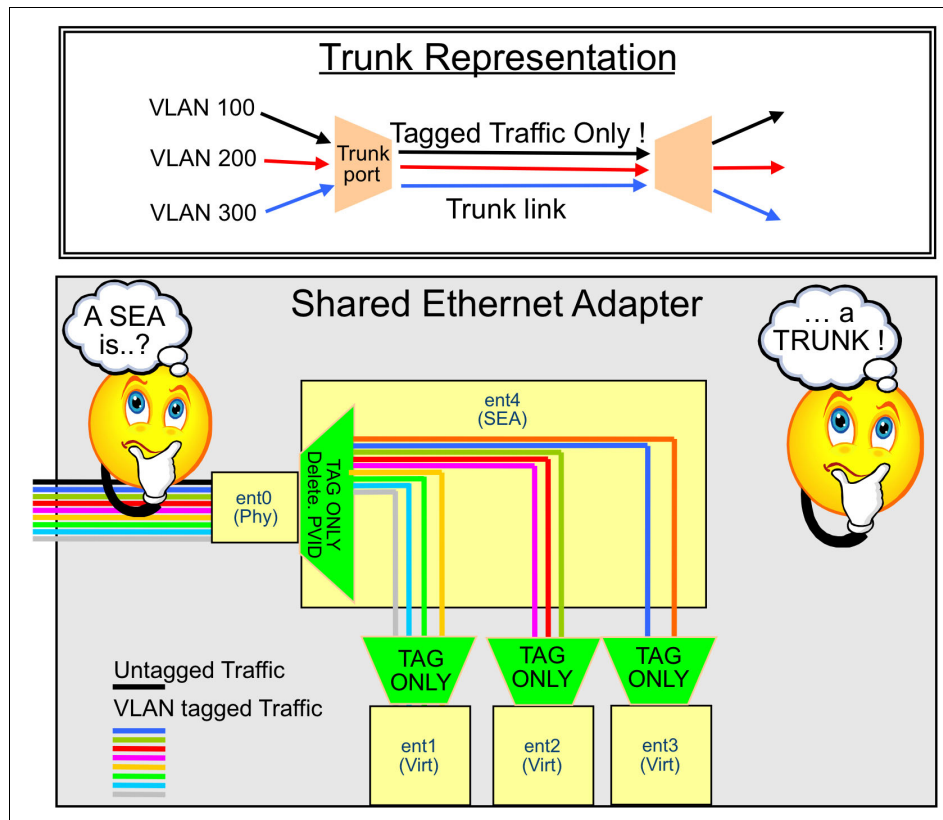


Figure 8-1 Representation of SEA as a trunk

Because SEA devices act only at OSI Layer 2 and to implement Trusted Firewall to act as a cross-VLAN router at OSI Layer 3, a new component is needed. The Secure Virtual Machine (SVM) is added to the Virtual I/O Server.

The SVM is implemented as a driver kernel extension. It minimizes the effect on the SEA device driver by performing VLAN crossing operations on network packets in the SVM kernel extension. The configuration and setup of Trusted Firewall is performed on the Virtual I/O Server command line.

SVM enables LPARs on the same system, but on different VLANs to communicate with each other by way of the SEA devices.

The SVM kernel extension consists of the following inter-virtual LAN routing functions:

- Layer 3 routing

VLANs represent different logical networks. Therefore, a Layer 3 router is required to connect the traffic between the VLANs.

- Network filtering rules

Network filtering rules are required to permit or deny inter-VLAN network traffic. Network filtering rules can be configured by using the Virtual I/O Server command-line interface.

Figure 8-2 shows the SVM implementation with an inbound network flow. The outbound traffic that enters the frame is never filtered because it is not initiated by an LPAR that is in the frame. Therefore, this traffic is never represented.

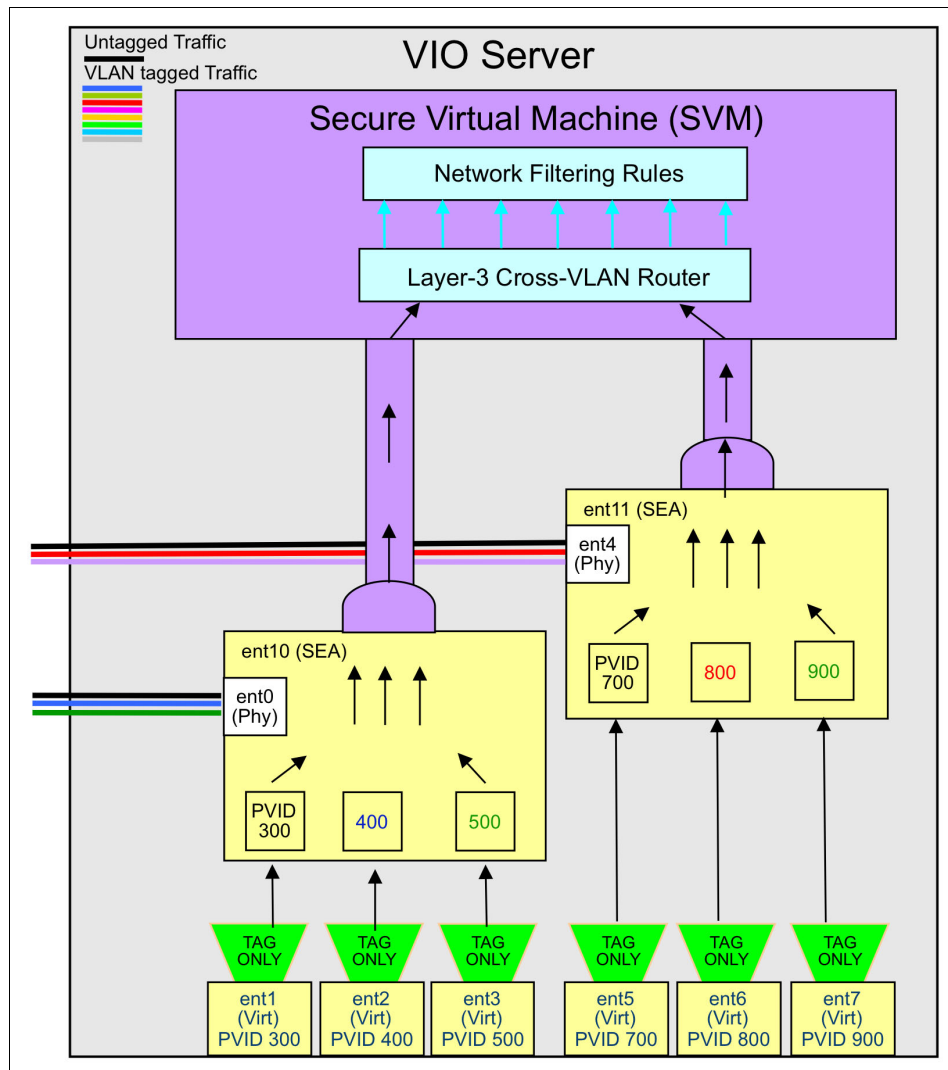


Figure 8-2 SVM Implementation within the Virtual I/O Server

Trusted Firewall ensures the inter-VLAN routing between LPARs in the same frame. Therefore, the packet filtering rules apply to only the IP addresses that belong to the frame.

The following types of traffic are excluded from packet filtering, as shown in Figure 8-3:

- External network traffic that enters through the physical adapters of the Virtual I/O Server

We filter only the cross-VLAN traffic in the same frame. The outbound traffic that comes into the frame is never filtered because it is not started by an LPAR that is in the frame.

- Intra-VLAN traffic between two LPARs

The intra-VLAN traffic occurs between two LPARs that belong to the same VLAN and same virtual switch. The IBM POWER Hypervisor™ forwards the packet at its Layer 2 Virtual Switch level. Packets are delivered based on their MAC address and VLAN tag at the level of the virtual switch.

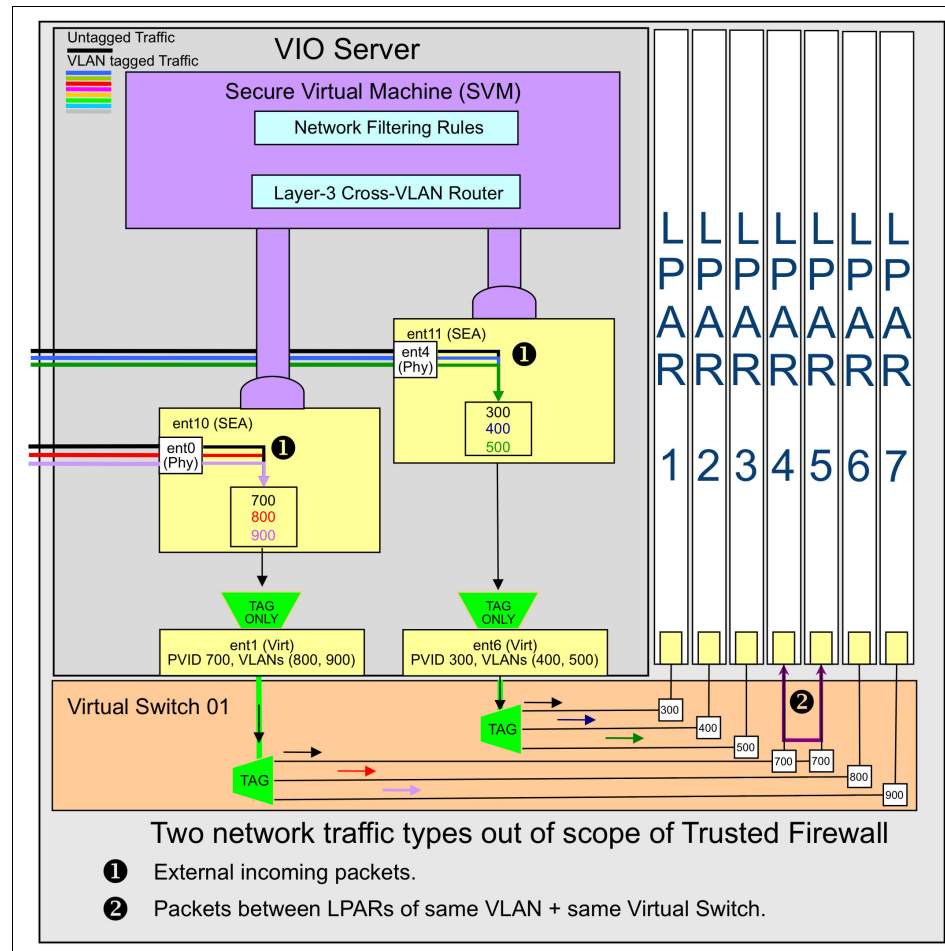


Figure 8-3 Excluded network traffic of the Trusted Firewall Filtering



Because of the Trusted Firewall implementation, the following configurations are ineligible for the Trusted Firewall packet filtering:

- ▶ Redundant SEA Load Sharing

Trunk adapters that are split between Virtual I/O Servers cannot be configured for Trusted Firewall Filtering (see Figure 8-4).

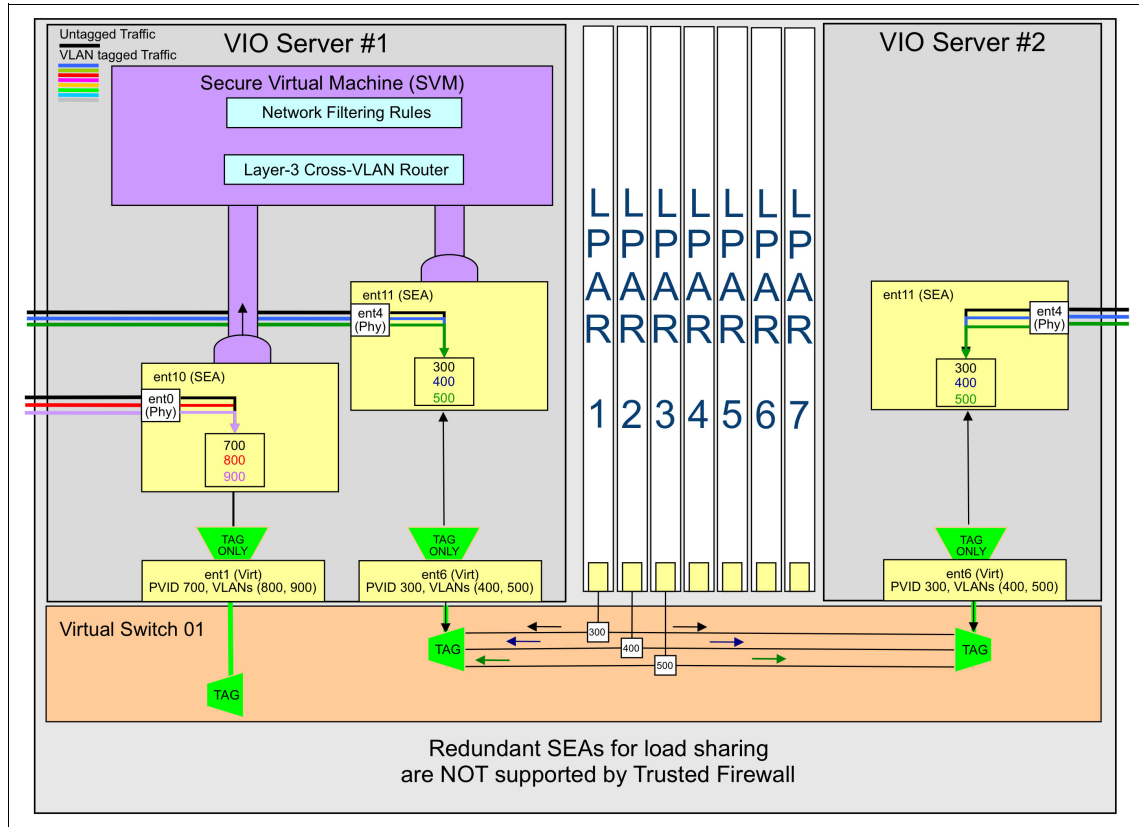
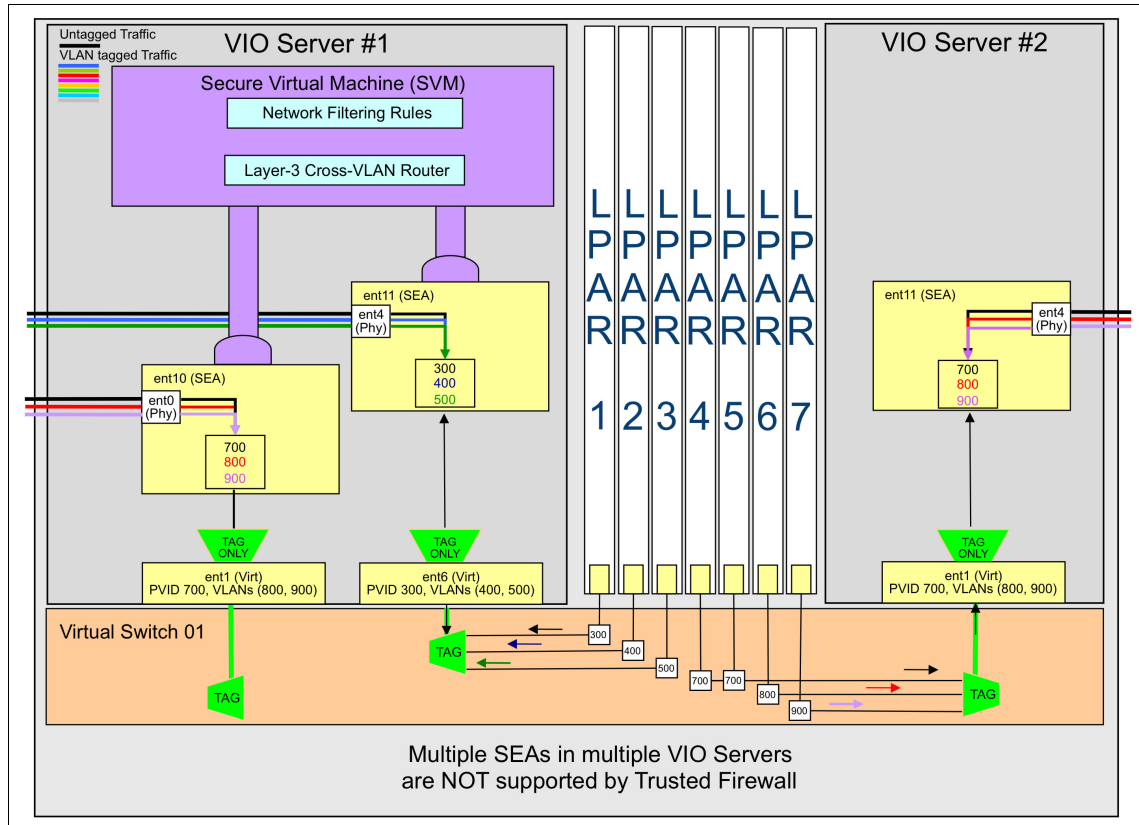


Figure 8-4 Redundant SEAs for load sharing are not supported

- Multiple SEAs in multiple Virtual I/O Servers

Trusted Firewall does not bridge multiple Virtual I/O Servers (see Figure 8-5).



Inbound network traffic takes the following forms (as shown in Figure 8-6 on page 285):

- LPAR network traffic that goes to the external network.

Trusted Firewall denies the packets if the destination IP address does not belong to the frame. The deny is to reject the packet to the physical network card of the SEA. Therefore, the traffic remains on its default path to the external network.

- LPAR network traffic that goes to another internal VLAN. It is inspected:

- The packet is validated by a filtering rule: it is permitted.

Trusted Firewall resends the packet to the destination SEA. The packet is not exposed to the external network. It has the benefit of the shortest path. That way, permitted traffic can bridge different SEAs in the same Virtual I/O Server.

- The packet is not validated by any filtering rule. It is denied.

Trusted Firewall resends the packet to the source SEA to be exposed to the external network. The deny is to reject the packet to the physical network card of the source SEA. Therefore, the denied traffic remains on its default path to the external network.

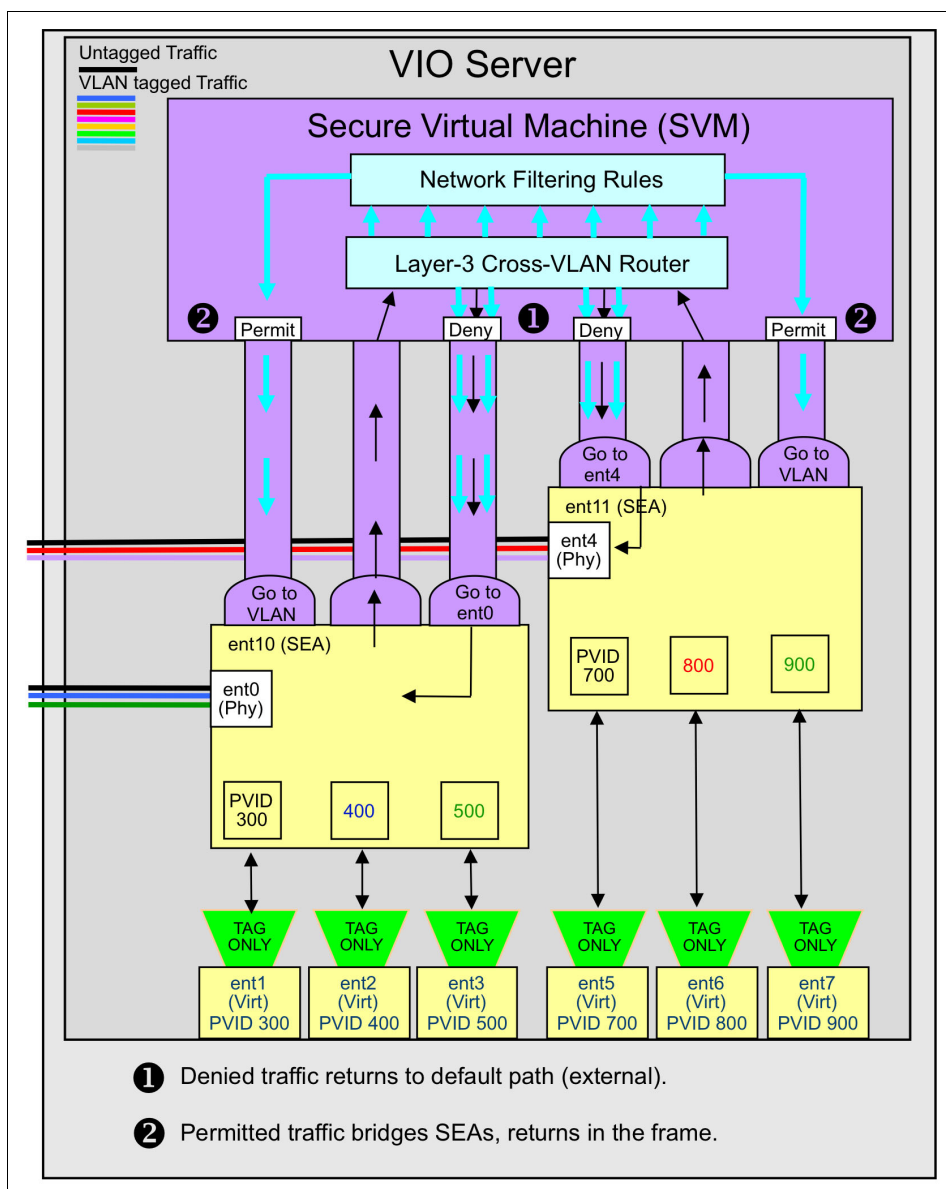


Figure 8-6 Description of filtered traffic by SVM

Trusted Firewall filters the traffic that reaches the trunks of the SEA devices. Therefore, the following supported SEA configurations are available:

- The SEAs with trunk adapters on the same Power hypervisor virtual switch

This configuration is supported because each SEA receives network traffic with different VLAN IDs (see Figure 8-7).

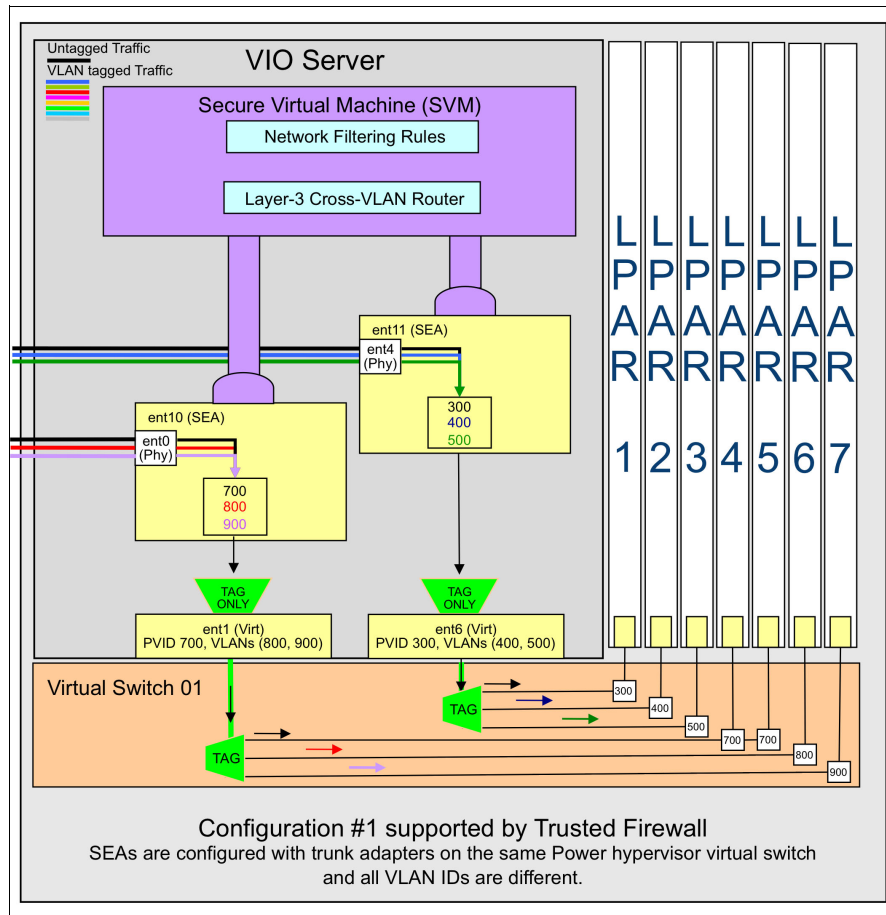


Figure 8-7 SEAs on the same virtual switch

- The SEAs with trunk adapters on different Power hypervisor virtual switches

Each trunk adapter is on a different VLAN ID. In this configuration, each SEA still receives network traffic by using different VLAN IDs as in the previous example when SEAs are on the same virtual switch.

The same VLAN IDs are reused on the virtual switches. In this case, the traffic for both SEAs has the same VLAN IDs, as shown in Figure 8-8.

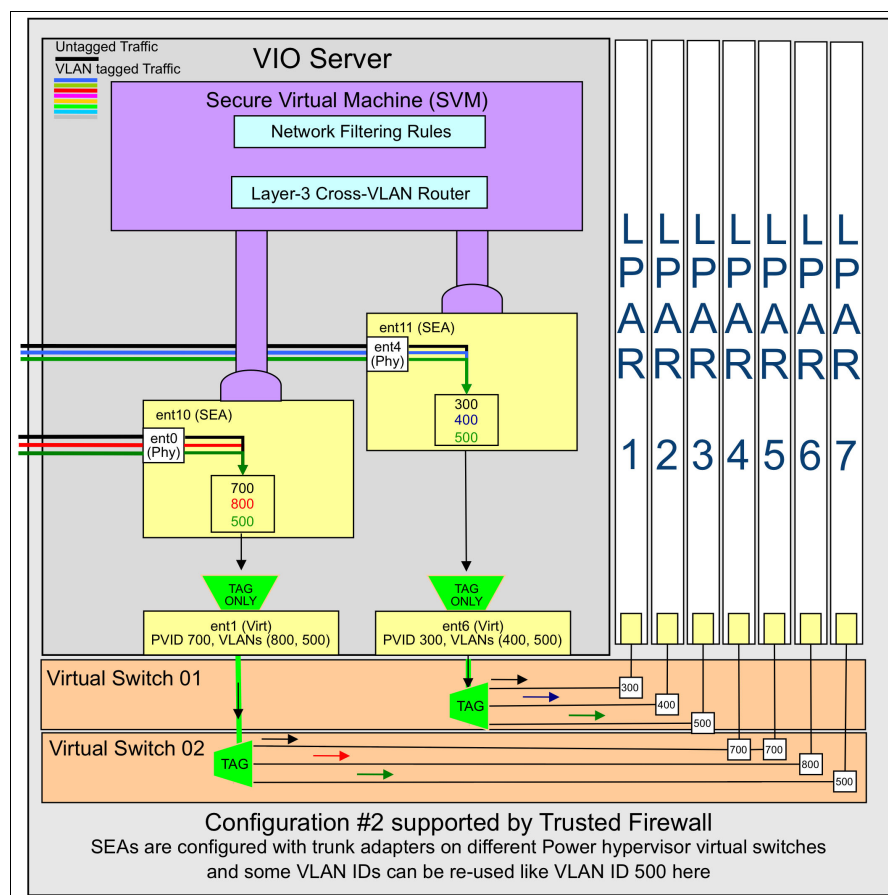


Figure 8-8 SEAs on different virtual switches

The following configurations are also supported:

- Trusted Firewall and SEA failover

Trusted Firewall must be configured in both Virtual I/O Servers. They must be enabled and configured with the filtered rules.

When the traffic fails over to the second Virtual I/O Server, Trusted Firewall immediately populates its mapping table with the new incoming traffic in the second Virtual I/O Server.

- Trusted Firewall and IBM PowerHA

Trusted Firewall denies the heartbeat packets because their destinations are outside the frame. Therefore, the heartbeat traffic remains on its default path to the external network.

- Trusted Firewall and Live Partition Mobility (LPM)

Partition mobility is supported by Trusted Firewall. Trusted Firewall refreshes its address entries automatically. However, Trusted Firewall must be enabled and configured with the filtering rules on the destination frame.

## 8.3 Deployment considerations

In this section, we describe the prerequisites to install Trusted Firewall.

PowerSC versions before version 1.1.1.0 did not include the required files to install Trusted Firewall. Therefore, ensure that you have the PowerSC installation CD or ISO image for PowerSC Standard Edition Version 1.1.1.0 or later.

PowerSC requires Virtual I/O Server Version 2.2.1.4, or later. Complete the following steps:

1. Log in to your Virtual I/O Server where you want to install Trusted Firewall.
2. Ensure that you are running Virtual I/O Server Version 2.2.1.4 or later by using the **ioslevel** command under the restricted shell:  

```
$ ioslevel
2.2.1.4
```
3. Install any required patches for your Virtual I/O Server and PowerSC Trusted Firewall. It is an opportunity to review the last HIPER/Critical Patches that were released by IBM. The patches are available IBM Support's [Fix Central web page](#).

Consider the following points:

- For the Virtual I/O Server software, select as Product Group:  
**Virtualization software** and **PowerVM Virtual I/O Server**
  - For the PowerSC Offering, select as Product Group:  
**Other software** and **PowerSC Standard Edition**<sup>1</sup>
4. Verify that you have superuser capabilities (PAdmin role) for the Virtual I/O Server where you are installing Trusted Firewall. Run the **lsuser** command as shown in the following example:  

```
$ whoami
padmin
$ lsuser
padmin roles=PAdmin default_roles=PAdmin account_locked=false expires=0
histexpire=0 histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8
minage=0 minalpha=0 mindiff=0 minlen=0 minother=0 pwdwarntime=0 registry=files
SYSTEM=compat
$
```

## 8.4 Installation

In the following sections, we describe a step-by-step Trusted Firewall installation and a step-by-step installation verification.

### 8.4.1 Trusted Firewall installation

In this procedure, we are installing Trusted Firewall with an ISO image.

With Electronic Software Delivery (ESD), you can download IBM i and AIX products when your software order is processed. You do not have to wait for your software to be delivered to you.

---

<sup>1</sup> Fix Central is being configured with this area at the time of this writing.

You receive electronic images that are bit-for-bit copies of the software order that you can download to an IBM i, AIX, or PC. You can burn the images to optical media or install the electronic images directly onto your system. For a list of products ESD, see this [web page](#).

ESD is available for PowerSC Standard Edition under the reference “5765-PSE: IBM PowerSC Std Ed V1.1”.

To order electronic delivery, the software order must include the following feature codes:

- ▶ United States, EMEA, LA, and AP: feature code 3450
- ▶ Canada: feature code 3470
- ▶ Japan: feature code 3471

For more information about registering to use electronic delivery, [this web page](#).

In this example, we use PowerSC internal images before the product release. Complete the following steps:

1. To install the fileset, you must escape the Virtual I/O Server restricted shell by using `$oem_setup_env`.
2. Download the ISO image of PowerSC Standard Edition and PowerSC Express Edition, as shown in the following example:

```
# cd /tmp/ISOIMAGES
# ls
cd.1231A_EXP_PSC.cksum  cd.1231A_EXP_PSC.iso    cd.1231A_STD_PSC.cksum
cd.1231A_STD_PSC.iso
```

3. To install from the ISO image of PowerSC Standard Edition, use the `loopmount` command, which is introduced in AIX 6.1 TL4. The `loopmount` command mounts the ISO image as though it is a CD (burning a CD is unnecessary):

```
# loopmount -i cd.1231A_STD_PSC.iso -o "-V cdrfs -o ro" -m /mnt
# cd /mnt/installp/ppc; pwd
/mnt/installp/ppc
```

4. Use SMIT to perform a regular installation by using the # **smitty installp** command (see Figure 8-9).

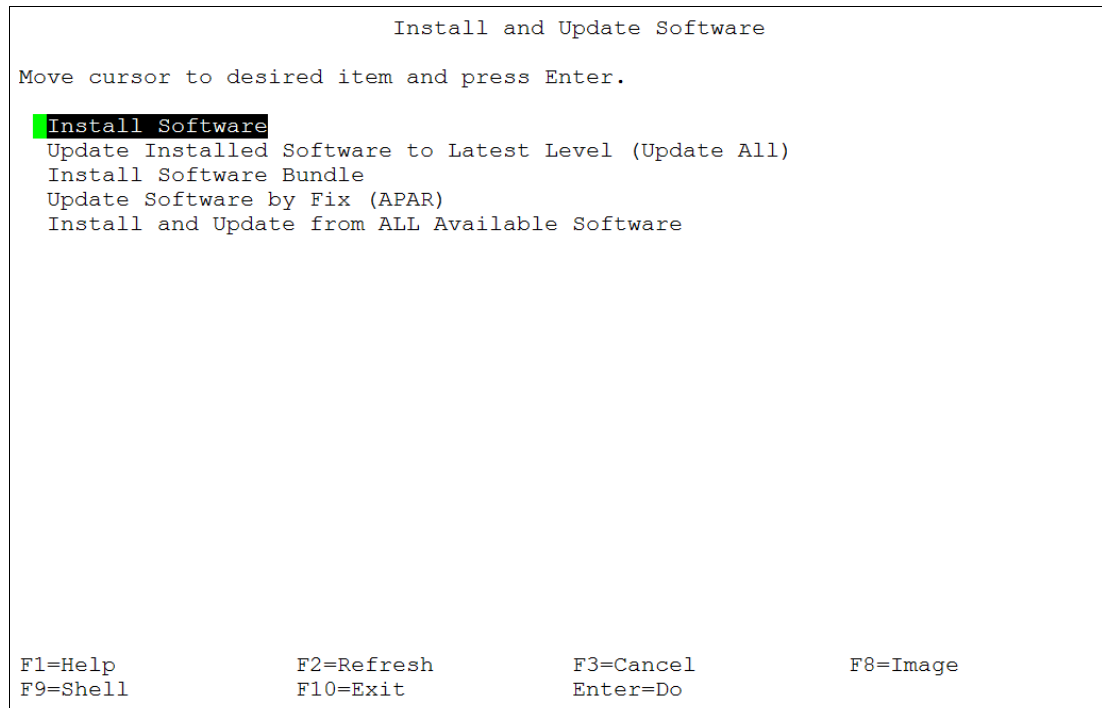


Figure 8-9 Smit pane for the smitty installp command

5. Select **Install Software**. In the INPUT field (see Figure 8-10), enter the directory of the licensed program products (LPP) files: /mnt/installp/ppc.

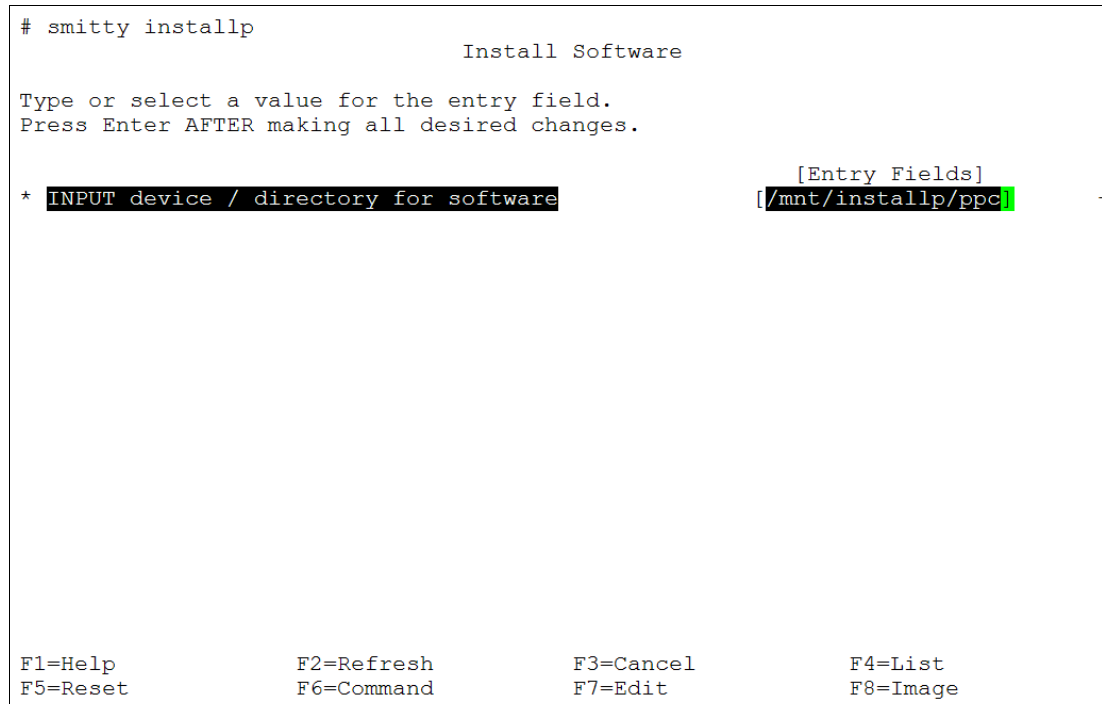


Figure 8-10 Location of LPP products in an ISO image



- ```

                                Install Software
-----
Type or select-----
Press Enter A|                                SOFTWARE to install
|
| Move cursor to desired item and press F7. Use arrow keys to scroll.
* INPUT devic| ONE OR MORE items can be selected.
* SOFTWARE to| Press Enter AFTER making all selections.
PREVIEW onl|
COMMIT soft| [MORE...5]
SAVE replac| + 1.1.2.0 ICE Express Security Extension
AUTOMATICAL|
EXTEND file| powerscExp.license ALL
OVERWRITE s| + 6.1.6.15 PowerSC Express Edition
VERIFY inst|
Include cor| powerscExp.rtc ALL
DETAILED ou| + 1.1.2.0 Real-Time Compliance
Process mul|
ACCEPT new | powerscStd.license ALL
Preview new| + 6.1.8.0 PowerSC Standard Edition
|
WPAR Manage| powerscStd.svm ALL
Perform| + 1.1.2.0 Secure Virtual Machine
Perform|
Det| powerscStd.tnc_pm ALL
Remount| + 1.1.2.0 Trusted Network Connect for Patch Management
Alterna|
| powerscStd.vlog ALL
| + 1.1.2.0 Virtual Log Device Software
| [MORE...3]
|
| F1=Help F2=Refresh F3=Cancel
| F7=Select F8=Image F10=Exit
| Enter=Do /=Find n=Find Next
F1=Help
F5=Reset
F9=Shell
-----

```

The last installation pane (see Figure 8-12) shows a successful installation for the powerscStd.svm.rte fileset at the 1.1.2.0 level.

```

COMMAND STATUS

Command: OK          stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

[MORE...78]
: accessauths

bosboot: Boot image is 51228 512 byte blocks.
0503-292 This update will not fully take effect until after a
        system reboot.

* * * A T T E N T I O N * * *
System boot image has been updated. You should reboot the
system as soon as possible to properly integrate the changes
and to avoid disruption of current functionality.

installp: bosboot process completed.
+-----+
+-----Summaries:-----+
+-----+

Installation Summary
-----
Name                               Level      Part      Event      Result
-----
powerscStd.svm.rte                 1.1.2.0    USR        APPLY      SUCCESS
powerscStd.svm.rte                 1.1.2.0    ROOT       APPLY      SUCCESS
[BOTTOM]

F1=Help          F2=Refresh          F3=Cancel          F6=Command
F8=Image         F9=Shell            F10=Exit           /=Find
n=Find Next

```

Chapter 8. Trusted Firewall **291**

The message states that you must restart because the modification of bosboot; however, a restart is not required.

## 8.4.2 Verifying the Trusted Firewall installation

You can check that Trusted Firewall is installed by using one of the following methods:

- Use the **ls1pp** command under the Virtual I/O Server restricted shell:

```
# ls1pp -L | grep powerscStd*
powerscStd.svm.rte 1.1.2.0    C    F    Secure Virtual Machine
```

- Verify that the Trusted Firewall commands are added to the Virtual I/O Server:

- The **mksvm** command is in the `/usr/sbin` directory.
- The **genvfilt**, **mkvfilt**, **lsvfilt**, **rmvfilt**, **chvfilt**, and **v1antfw** commands are in the `/usr/ios/utills` directory.

- The RBAC authorization `vios.security.svm` on the Virtual I/O Server grants the privilege to run these commands. Check that `vios.security.svm` exists in your system by using the **lsauth** command:

```
$ lsauth vios.security.svm
vios.security.svm id=11566 dfltmsg=System SVM Authorization
msgcat=viosauths.cat msgset=11 msgnum=2
```

## 8.5 Working with Trusted Firewall

In 8.2, “Detailed implementation” on page 280, we described the following characteristics of Trusted Firewall:

- Is a Layer 3 router

This router is implemented as a kernel extension or driver, which is called the Secure Virtual Machine (SVM).

- Runs the Firewall Packet Filtering rules

These Packet Filtering rules must be created in tables. Then, these tables are pushed to the SVM driver to become operational.

In this section, we describe how to configure SVM and how to manage the packet filtering rules.

### 8.5.1 Configuring the Secure Virtual Machine

In this section, we describe how to initialize the SVM driver and manage the SVM address mapping table.

#### Initializing the SVM driver

As a Virtual I/O Server kernel extension, SVM requires configuration by a super user with the PAdmin role under the non-restricted shell. To start SVM, use the **mksvm** command. This command needs to be run only once.

To load the SVM kernel extension in the Virtual I/O Server, complete the following steps:

1. Load SVM as a kernel extension by using the **mksvm** command as the padmin role:

```
$ mksvm
$
```

2. Verify that a device is created by using the **lsdev** command. In the output, look for the svm entry:

```
$ lsdev -virtual
```

**Note:** After the **mksvm** command is run *once only*, the SVM driver is loaded automatically by cfgmgr stage during each restart. The **mksvm** command can be run under the Virtual I/O Server restricted shell.

3. Couple the SVM driver with the SEA interfaces to start the packet routing.

## Managing the SVM firewall

With only one unique command, **vlanfw**, you can manage the SVM and the Trusted Firewall and ensure cross-VLAN communication. This **vlanfw** command can be used in the Virtual I/O Server restricted shell.

By using the **vlanfw** command options, you can start, stop, and query the status of SVM. These options are listed in Table 8-6.

Table 8-6 The **vlanfw** command options to configure the SVM

| vlanfw command option | Description                                      |
|-----------------------|--------------------------------------------------|
| vlanfw -s             | Starts SVM (the Trusted Firewall)                |
| vlanfw -t             | Stops SVM (the Trusted Firewall)                 |
| vlanfw -q             | Queries the status of SVM (the Trusted Firewall) |

The command includes the following execution examples:

- Start the Trusted Firewall (SVM) by using the **vlanfw -s** command:

```
$ vlanfw -s
$
```

We are under the Virtual I/O Server restricted shell execution environment.

- Query the Trusted Firewall (SVM) status by using the **vlanfw -q** command:

```
$ vlanfw -q
vlanfw: TFW=True capability=4
$
```

**Note:** When the SVM is active, the Trusted Firewall field value is True. When the SVM is used as the Trusted Firewall, its capability value is 4.

- Stop the Trusted Firewall (SVM) by using the **vlanfw -t** command:

```
$ vlanfw -t
$
```

**Note:** When the SVM is stopped, the Trusted Firewall field value is False and its capability value is 0.

## Managing the SVM address mapping table

When SVM Firewall starts, one of its main tasks is to register itself to all SEA devices that are available in the Virtual I/O Server. This registration (see Figure 8-13) allows the SVM driver to receive and send packets from all the SEA devices to ensure the cross-VLAN Layer 3 routing.

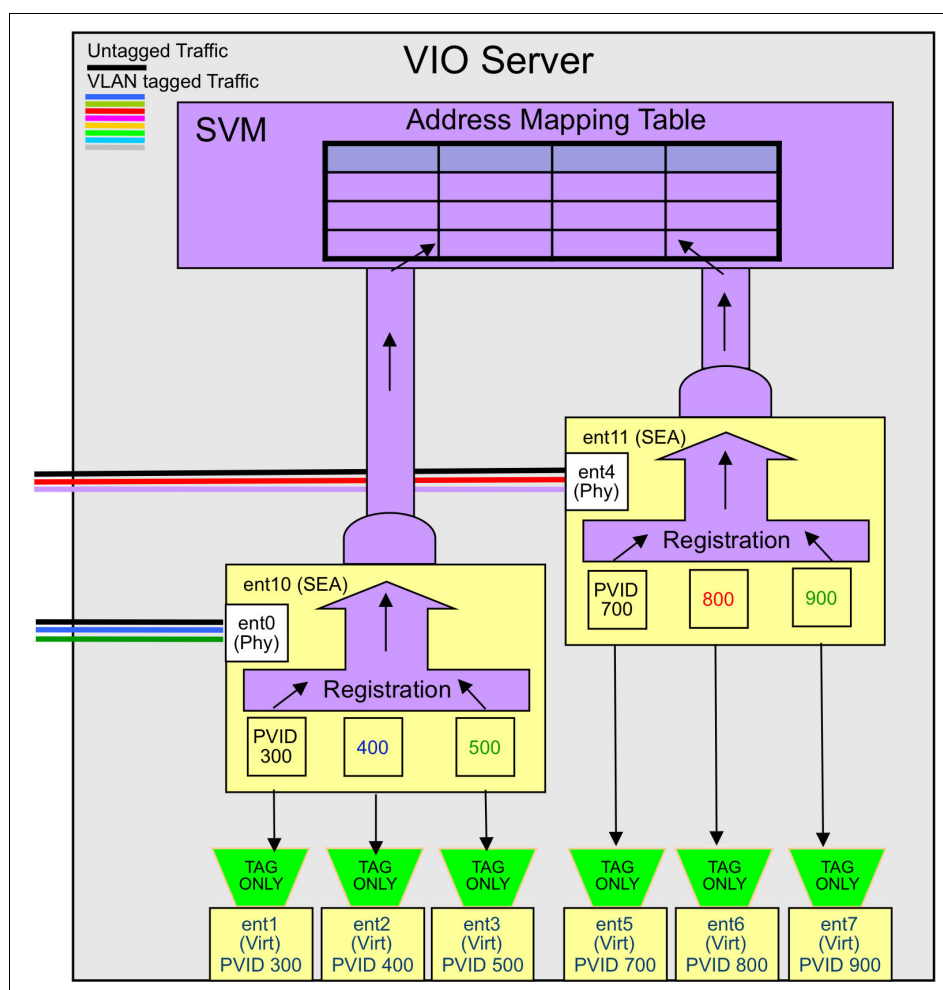


Figure 8-13 SVM registration to the SEA devices

We use the command options that are listed in Table 8-7 to display the entries and flush the content of the SVM Firewall address mapping table.

Table 8-7 Commands to manage the SVM address mapping table

| vlanfw command option | Description                         |
|-----------------------|-------------------------------------|
| vlanfw -d             | Display all the IP mappings         |
| vlanfw -f             | Remove or flush all the IP mappings |

This SVM registration with all SEA devices causes a configuration constraint when one SEA device must be removed. SVM must first be unregistered from all SEA devices before any SEA device is removed. For more information about removing the SVM, see 8.5.3, “Removing Trusted Firewall” on page 311.

**Important:** Removing any SEA device before you remove the SVM driver can result in system failure.

To ensure this Layer 3 routing, SVM must associate each IP address (OSI Layer 3) with its associated MAC address and VLAN ID (OSI Layer 2). When SVM Firewall starts, it populates an internal dynamic table with the information that it collected from all the SEA devices. Table 8-8 lists an example of the SVM address mapping table.

Table 8-8 SVM dynamic IP - MAC address mapping table

| VLAN ID | Trunk ID or SEA (PVID) | IP address    | MAC address      |
|---------|------------------------|---------------|------------------|
| 1       | 1                      | 172.16.20.121 | 66:1d:ba:7:38:2  |
| 1       | 1                      | 172.16.20.122 | 66:1d:b2:c9:1c:2 |
| 1       | 1                      | 172.16.20.123 | 66:1d:bd:de:51:2 |
| 1       | 1                      | 172.16.20.124 | 66:1d:bf:53:dc:2 |

SVM Firewall keeps this table in its internal memory, even when SVM Firewall is stopped.

Check it by completing the following steps:

1. Check that SVM Firewall is active by using the **vlantfw -q** command:

```
$ vlantfw -q
vlantfw: TFW=True capability=4
```

2. Query the IP-MAC mapping table by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

3. Stop SVM Firewall by using the **vlantfw -t** command:

```
$ vlantfw -t
$
```

4. Check that SVM Firewall is stopped by using the **vlantfw -q** command:

```
$ vlantfw -q
vlantfw: TFW=False capability=0
$
```

5. Query the SVM IP-MAC mapping table by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
$
```

**Important:** Although SVM Firewall is stopped, it maintained this IP-MAC address mapping table in its own driver memory.

To verify when this table is created and when this table is populated, complete the following steps:

1. Remove the SVM device driver by using the **rmdev -dev svm** command:

```
$ rmdev -dev svm
svm deleted
```

2. Query the status of SVM Firewall by using the **vlantfw -q** command:

```
$ vlantfw -q
vlantfw: failed to open device /dev/svm
$
```

When the SVM driver is not installed, the device driver does not exist.

3. Install the SVM device driver by using the **mksvm** command:

```
$ mksvm
$
```

4. Query the status of SVM Firewall by using the **vlantfw -q** command:

```
$ vlantfw -q
vlantfw: TFW=False capability=0
$
```

5. Query the IP-MAC mapping table by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 0
$
```

The IP-MAC Mapping table is created but it is still not populated.

6. Start the SVM Firewall by using the **vlantfw -s** command:

```
$ vlantfw -s
$
```

7. Query the IP-MAC mapping table by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
$
```

**Consideration:** The SVM address mapping table is empty when it is created when the SVM driver is installed (by using **mksvm** command) or is populated when the SVM Firewall is started (by using **vlantfw -s** command).

However, this SVM address mapping table must be refreshed when LPARs have no more network activity. This lack of activity can be because of LPAR inactivity, LPAR removal, or LPAR migration. Therefore, this table is *dynamic*, and the content can be flushed.

Verify how this dynamic table works by completing the following steps. Four LPARs are running on the server, as shown in Figure 8-14.

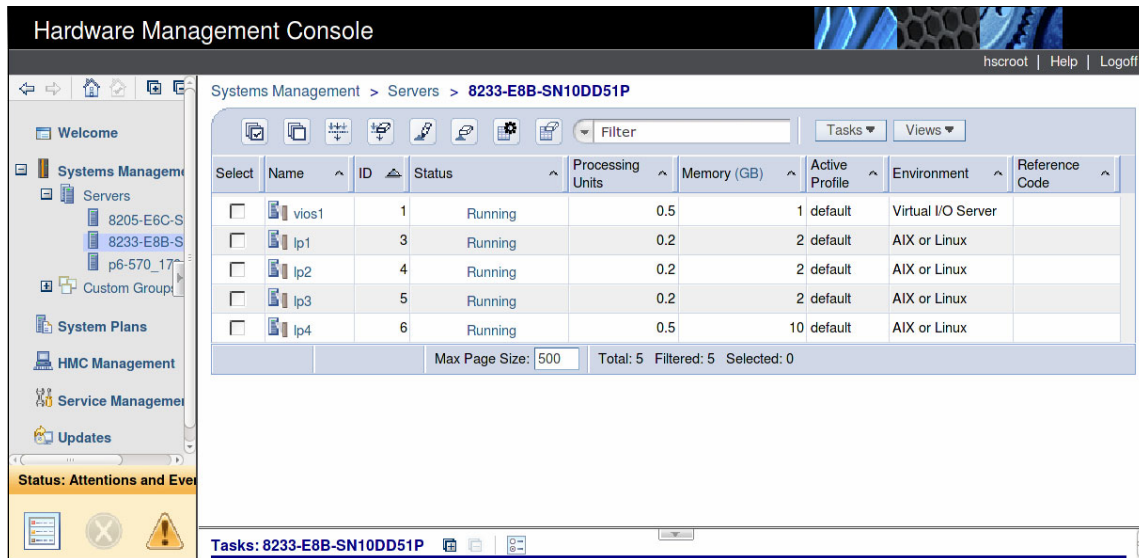


Figure 8-14 Virtual I/O Server with Trusted Firewall and four running LPARs

1. Query the SVM address mapping table:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

2. Stop three LPARs. Shut down lp1, lp2, and lp3, as shown in Figure 8-15.

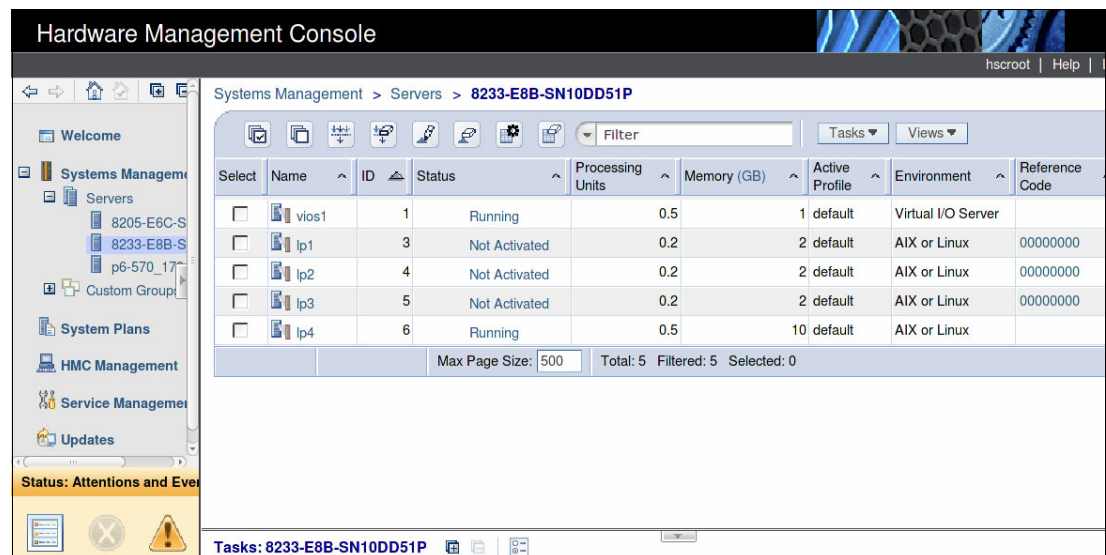


Figure 8-15 Virtual I/O Server with Trusted Firewall and three stopped LPARs

3. Query the SVM address mapping table again:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 1
0: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

The SVM Firewall updated its address mapping table. The first row shows the IP signature of the lp4 LPAR.

4. Restart lp1, lp2, and lp3 LPARs. Query the SVM mapping table again:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

The lp1, lp2, and lp3 LPARs are back automatically in the SVM Firewall address mapping table after they are restarted. The SVM Firewall address mapping is refreshed and maintained automatically.

SVM performs a regular lookup of the mapping table entries. To flush the SVM table content, use the **vlantfw -f** command. Complete the following steps:

1. Display the table content by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

2. Flush the table content by using the **vlantfw -f** command:

```
$ vlantfw -f
$
```

3. Check that the table content is empty by using the **vlantfw -d** command:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 0
```

4. Check the table again; the content is back:

```
$ vlantfw -d
vlantfw: /dev/svm dump dynamic learning IP and MAC: count: 4
0: vid: 1 pvidflag: 1 addr: 172.16.20.121 mac: 66:1d:ba:7:38:2
1: vid: 1 pvidflag: 1 addr: 172.16.20.122 mac: 66:1d:b2:c9:1c:2
2: vid: 1 pvidflag: 1 addr: 172.16.20.123 mac: 66:1d:bd:de:51:2
3: vid: 1 pvidflag: 1 addr: 172.16.20.124 mac: 66:1d:bf:53:dc:2
```

The SVM Firewall updated its address mapping table in a few seconds.

**Important:** The address mapping table is automatically refreshed and maintained by SVM, based on the traffic that passes through the SEA's devices. The stale entries are managed automatically by SVM.

The SVM VLAN-IP-MAC mapping table is, by itself, useful information for any Virtual I/O Server system administrator. Permanent activation of the SVM Firewall is not required. The definition of the packet filtering rules is not required.



## 8.5.2 Configuring the filter rules

Trusted Firewall ensures the cross-VLAN routing between LPARs in the same frame. Therefore, the packet filtering rules apply to only the IP addresses that belong to the SVM address mapping table (for more information, see “Managing the SVM address mapping table” on page 294).

The following types of traffic are excluded from filtering:

- ▶ The external network traffic that enters through the physical adapters of the Virtual I/O Server. We are filtering only the cross-VLAN traffic in the same frame. Therefore, the external packets that are received by the SEA physical adapters are never sent to SVM.
- ▶ The intra-VLAN traffic between two LPARs.

The intra-VLAN traffic occurs between two LPARs that belong to the same VLAN and the same virtual switch. The POWER Hypervisor forwards the packet at its Layer 2 virtual switch level. Therefore, packets are delivered based on their MAC addresses and VLAN tags at the level of the virtual switch.

There is an easy way to remember which type of traffic SVM routes: *SVM receives the packets that are received by the SEA trunk devices*. Consider the following points:

- ▶ The intra-VLAN does not reach the SEA trunks; SVM cannot filter this traffic.
- ▶ The external traffic does not arrive through the SEA trunks; SVM does not filter it.

**Important:** Trusted Firewall does *not* route the following types of traffic:

- ▶ The external network traffic that is received by the physical adapters of the Virtual I/O Server
- ▶ The intra-VLAN traffic between two LPARs (the same VLAN and the same vSwitch)

The network filtering is applied to only the IP packets that are eligible by SVM (the destination IP of the packet is in the frame); therefore, it is in the Address Mapping Table.

It is advised that you limit filter rule creation to only the IP addresses and VLANs, which are displayed in the Address Mapping Table (**vlantfw -d** command). These two tables with their precedence order are shown in Figure 8-16 on page 300.

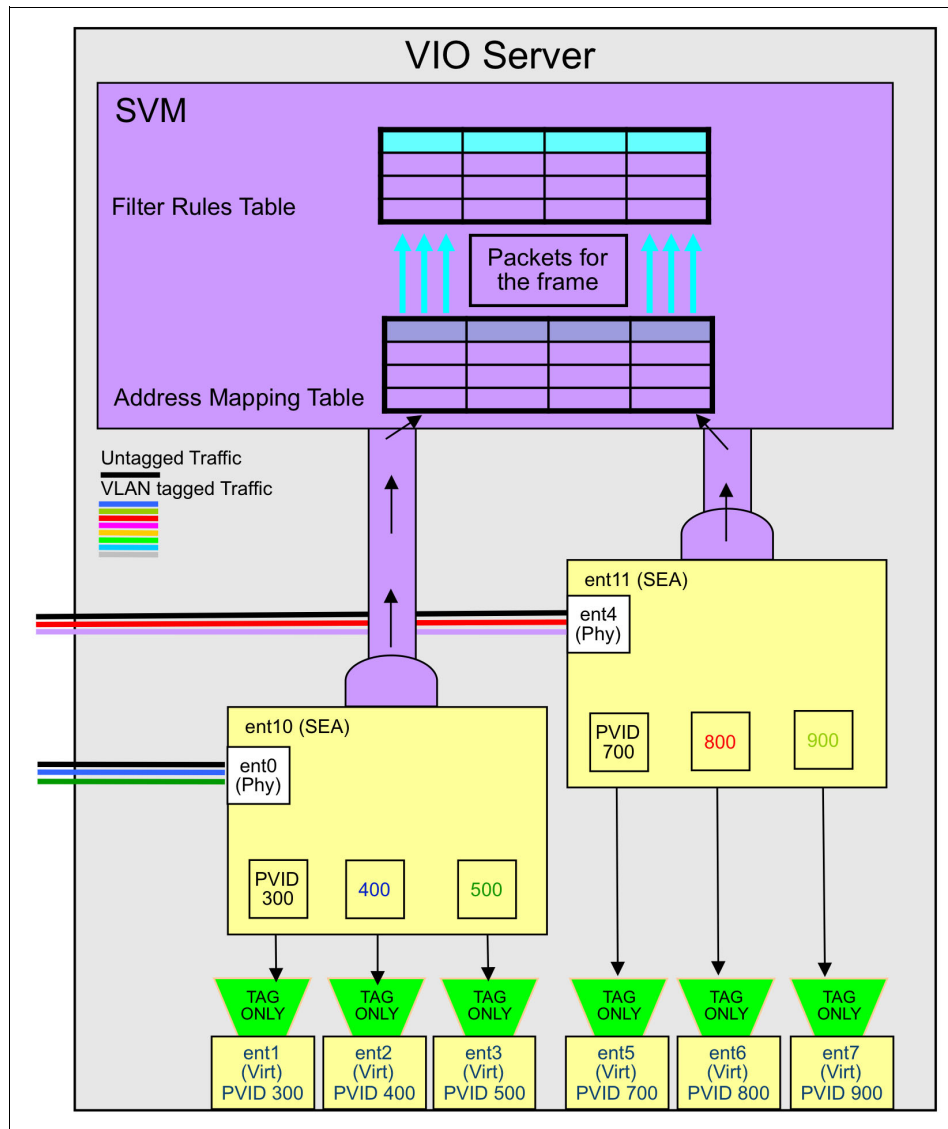


Figure 8-16 Address Mapping Table and Filter Rules Table

To create and maintain the filtering rules, two filtering tables are available, as shown in Figure 8-17 on page 301:

- The active filtering rule table, which is loaded into SVM, is used for Trusted Firewall operations. Actions are limited in this table because it is in use to match arriving IP Packets. Consider the following points:
  - You can deactivate a filter rule immediately by using the rule removal.
  - You can list the table content, which is the active filtering rules.
  - You can change the filtering rule interactively.
  - You can disable the table by using its content flush. This command flushes the repository simultaneously. It is suggested that you have a script to re-create the rules in the repository.

- The inactive filtering rule table is used as a repository that you can maintain. Consider the following points:
  - You can create a rule or modify it in the repository.
  - You can list the repository content.
  - You can remove a rule.
  - You can deactivate the repository by using its content flush. This command flushes the active table simultaneously. It is advised that you have a script to re-create the rules in the repository.

The repository and the active rules table content must remain identical and reflect each other. For this reason, the rules management commands are applied first to the active rules table, then to the repository (see Figure 8-17). Earlier versions of Trusted Firewall v1.1.2 required that you activate the rules table and the repository to update both sides. This requirement disappeared in Trusted Firewall Version 1.1.2.

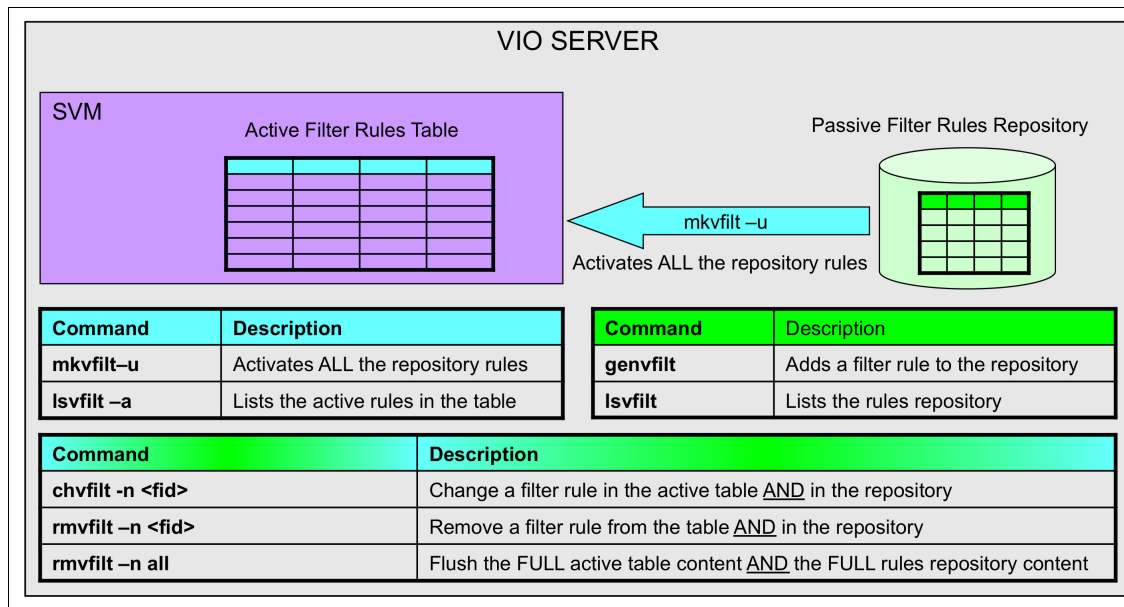


Figure 8-17 Filter rule management commands

## Managing the filter rules repository

IP Filtering rules are by definition not easy to work with. You must create and read them by using Filtering rules management commands to detail the IP packet characteristics.

Therefore, to become familiar with these command options, we use a table that is based on the model that we introduced in 8.1.4, “Security policies” on page 276.

In Table 8-9 on page 302, each option flag is associated with the generic security policy column. Then, the possible values for these parameters are in the second line. The third line shows whether these parameters are required (Req.) or optional (Opt.).

Table 8-9 Options of Filter Rules Definition commands

| IPv.   | Action | VLAN src | VLAN dst | From IP | Src port | Src oper.             | To IP | Dst port | Dst oper.             | Protocol type                      |
|--------|--------|----------|----------|---------|----------|-----------------------|-------|----------|-----------------------|------------------------------------|
| -v     | -a     | -z       | -Z       | -s      | -p       | -o                    | -d    | -P       | -O                    | -c                                 |
| 4<br>6 | P<br>D | value    | value    | value   | value    | lt<br>gt<br>eq<br>any | value | value    | lt<br>gt<br>eq<br>any | udp<br>tcp<br>icmp<br>icmp6<br>any |
| Req.   | Req.   | Req.     | Req.     | Opt.    | Opt.     | Opt.                  | Opt.  | Opt.     | Opt.                  | Opt.                               |

Trusted Firewall maintains the IPv4 Filter rules and the IPv6 Filter rules in the same repository. You can administer both IP protocols by using the same commands.

**Important:** The IBM documentation about the **genvfilt** command and **chvfilt** command provides **icmpv6** as a parameter value for the ICMPv6 protocol. This value is a typographical error. The correct ICMPv6 parameter value is **icmp6**.

For clarification, consider the following example:

```
$ genvfilt -v6 -aP -z404 -Z504 -cicmpv6
Invalid protocol "icmpv6".
$ genvfilt -v6 -aP -z404 -Z504 -cicmp6
Filter index is 1
Filter rule 1 has been added successfully.
$
```

For more information about the **genvfilt** and **chvfilt** commands, see [IBM Knowledge Center](#).

The first steps to managing the rule repository are based on simple firewall policies that are defined in Table 8-10. You can define only the minimum required parameters. They define mainly a generic connection between one VLAN to another VLAN.

Table 8-10 Simple firewall policy table

| IPv. | Action | VLAN src | VLAN dst | From IP | Src port | Src oper. | To IP | Dst port | Dst oper. | Protocol type |
|------|--------|----------|----------|---------|----------|-----------|-------|----------|-----------|---------------|
| -v   | -a     | -z       | -Z       | -s      | -p       | -o        | -d    | -P       | -O        | -c            |
| 4    | P      | 304      | 404      |         |          |           |       |          |           |               |
| 4    | P      | 404      | 504      |         |          |           |       |          |           |               |
| 6    | D      | 306      | 406      |         |          |           |       |          |           |               |
| 6    | D      | 406      | 506      |         |          |           |       |          |           |               |

In our simple firewall policies, we want VLAN 304 and VLAN 404 (IPv4) to communicate freely through Trusted Firewall. The same firewall policies must be configured for VLAN 404 and VLAN 504 (IPv4). However, any communication between VLAN 306 and VLAN 406 (IPv6) or between VLAN 406 to VLAN 506 (IPv6) must be exposed to the external network for inspection. Because the rest of the traffic is not described by a permit rule, it is exposed to the external network (deny state, by default).

The commands to manage the filter rule repository are listed in Table 8-11.

Table 8-11 Commands to manage the filter rule repository

| Command                       | Description                                                          |
|-------------------------------|----------------------------------------------------------------------|
| <b>genvfilt</b> <parameters>  | Adds a filter rule to the repository                                 |
| <b>lsvfilt</b>                | Lists the rules in the repository                                    |
| <b>chvfilt -n &lt;fid&gt;</b> | Change a rule in the repository <i>and</i> in the active rules table |
| <b>rmvfilt -n &lt;fid&gt;</b> | Remove a rule in the repository <i>and</i> in the active rules table |
| <b>rmvfilt -n all</b>         | Flush the rules repository <i>and</i> the rules table content        |

### Creating and listing a filter rule

Now, we run some of these commands by completing the following steps:

1. List the default repository content by using the **lsvfilt** command:

```
$ lsvfilt
Beginning of filter rules.
number of filters in ODM is 1
Filter ID:0
Filter Action:1
Source VLANID:-1
Destination VLANID:-1
Source Address:0.0.0.0
Destination Address:0.0.0.0
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:0
```

2. List the active rule table in SVM by using the **lsvfilt -a** command:

```
$ lsvfilt -a
Number of active filter rules in kernel = 0
```

3. Create four filtering rules in the repository by using the **genvfilt** command:

```
$ genvfilt -v4 -aP -z304 -Z404
Filter index is 1
Filter rule 1 has been added successfully.
$ genvfilt -v4 -aP -z404 -Z504
Filter index is 2
Filter rule 2 has been added successfully.
$ genvfilt -v6 -aD -z306 -Z406
Filter index is 3
Filter rule 3 has been added successfully.
$ genvfilt -v6 -aD -z406 -Z506
Filter index is 4
Filter rule 4 has been added successfully.
$
```

4. Verify and check the creation of the four filtering rules by using the **lsvfilt** command:

```
$ lsvfilt | tee -a /tmp/lsvfilt.out
$ wc -l /tmp/lsvfilt.out
68 /tmp/lsvfilt.out
```

In Trusted Firewall Version 1.1.2, the output of the **lsvfilt** command is large. Therefore, it is advised that you redirect the output to a file through the help of **tee -a** command because we are working under the Virtual I/O Server restricted shell.

5. Review filter rule ID 2 and filter rule ID 4 output:

**Filter ID:2**

Filter Action:1  
Source VLANID:404  
Destination VLANID:504  
Source Address:0.0.0.0  
Destination Address:0.0.0.0  
Source Port Operation:any  
Source Port:0  
Destination Port Operation:any  
Destination Port:0  
Protocol:0

**Filter ID:4**

Filter Action:2  
Source VLANID:406  
Destination VLANID:506  
Source Address::  
Destination Address::  
Source Port Operation:any  
Source Port:0  
Destination Port Operation:any  
Destination Port:0  
Protocol:0

The output of the **lsvfilt** command needs some assistance for you to read it. The **lsvfilt** output format reflects the parameter values as the IP values. Therefore, it is much easier to read this output when you are familiar with **tcpdump** readings.

Therefore, for a broader audience, Table 8-12 lists the **lsvfilt** output values that are associated with the **genvfilt** parameter values that you entered. This table can help you read these output files.

*Table 8-12 lsvfilt and genvfilt parameter values*

| Parameter             | genvfilt command         | lsvfilt output value |
|-----------------------|--------------------------|----------------------|
| Filter Action: Permit | -aP                      | 1                    |
| Filter Action: Deny   | -aD                      | 2                    |
| IPv4 Address: Any     | Do not set the parameter | 0.0.0.0              |
| IPv6 Address: Any     | Do not set the parameter | :::                  |
| Protocol: any         | Do not set the parameter | 0                    |
| Protocol: tcp         | -ctcp                    | 6                    |
| Protocol: udp         | -cudp                    | 17                   |
| Protocol: icmp        | -cicmp                   | 1                    |
| Protocol: icmpv6      | -cicmp6                  | 58                   |
| Port Operation: any   | Do not set the parameter | any                  |
| Port Operation: lt    | -olt -p or -olt -P       | lt                   |

| Parameter          | genvfilt command   | lsvfilt output value |
|--------------------|--------------------|----------------------|
| Port Operation: gt | -ogt -p or -0gt -P | gt                   |
| Port Operation: eq | -oeq -p or -0eq -P | eq                   |

### Changing a rule

Use the **chvfilt** command to change a rule. The parameters of the **chvfilt** command are identical to the parameters of the **genvfilt** command. The only required parameter is the rule number in the repository to be modified, followed by one or more parameters that you want to change.

To keep the active rules table and the repository consistent, *an earlier version* of the **chvfilt** command is required to activate the rule table. Therefore, during the time you worked on the rules repository, the **chvfilt** command activated the repository and the rules that you wanted to modify. In the most recent version, the **chvfilt** command does not require that the repository is active.

To determine whether you are using the old version, the **chvfilt** command produces the following error:

```
$ chvfilt -n 4 -aP
ioctl(QUERY_FILTER) failed no filter rule err =2
Cannot Change the filter rule.
```

Upgrade to the latest version of the **chvfilt** command. You also can use the following workaround:

1. Stop Trusted Firewall. All traffic is not interrupted but routed to the external network. Trusted Firewall is not disruptive by design:

```
$ vlantfw -t
$ vlantfw -q
vlantfw: TFW=False capability=0
```

2. Activate your repository within SVM safely because no other traffic arrives at the SVM. To activate the repository, use the **mkvfilt -u** command:

```
$ mkvfilt -u
$
```

3. Check whether the repository is active by using the **lsvfilt -a** command:

```
$ lsvfilt -a | grep kernel
Number of active filter rules in kernel = 4
$
```

Now, the repository rules are activated.

**Important:** You can activate the Filtering Rules when Trusted Firewall is stopped. The two layers of Trusted Firewall (SVM Address Mapping Table and Filtering Rules Table) are independent, which makes maintenance easier.

4. Verify the rule 4 content by using the **lsvfilt -a | grep -p "Filter ID:4"** command:

```
$ lsvfilt -a | grep -p "Filter ID:4"
Number of active filter rules in kernel = 4
Filter ID:4
Filter Action:2
Source VLANID:406
```

```

Destination VLANID:506
Source Address:::
Destination Address:::
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:0
$

```

5. Change rule 4. Modify two parameters by using the **chvfilt -n** command:

```

$ chvfilt -n4 -aP -cicmp6
Filter rule 4 has been changed successfully.
$

```

6. Verify the new rule 4 by using the **lsvfilt -a | grep -p "Filter ID:4"** command:

```

$ lsvfilt -a | grep -p "Filter ID:4"
Number of active filter rules in kernel = 4
Filter ID:4
Filter Action:1
Source VLANID:406
Destination VLANID:506
Source Address:::
Destination Address:::
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:58

```

### ***Removing a rule***

The command to change a rule is the **rmvfilt** command. The only required parameter is the rule number in the repository or in the active rules table.

Because the two tables are kept consistent, the rule has the same number. Complete the following steps:

1. Verify active rule 2 by using the **lsvfilt -a** command:

```

$ lsvfilt -a | grep -p "Filter ID:2"
Filter ID:2
Filter Action:1
Source VLANID:404
Destination VLANID:504
Source Address:0.0.0.0
Destination Address:0.0.0.0
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:0

```

2. Verify rule 2 in the repository by using the **lsvfilt** command:

```

$ lsvfilt | grep -p "Filter ID:2"
Filter ID:2
Filter Action:1
Source VLANID:404

```



```
Destination VLANID:504
Source Address:0.0.0.0
Destination Address:0.0.0.0
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:0
```

3. Remove active rule 2 by using the `rmvfilt -n` command:

```
$ rmvfilt -n 2
$
```

4. Check that active rule 2 is removed by using the `lsvfilt -a` command:

```
$ lsvfilt -a | grep -p "Filter ID:2"
$
```

We can see that slot rule 2 is now empty. The rules are not renumbered.

5. Review the active rule table by using the `lsvfilt -a` command output:

```
$ lsvfilt -a
Number of active filter rules in kernel = 3
Filter ID:4
Filter Action:1
Source VLANID:406
Destination VLANID:506
Source Address:::
Destination Address:::
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:58
```

```
Filter ID:3
Filter Action:2
Source VLANID:306
Destination VLANID:406
Source Address:::
Destination Address:::
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
Protocol:0
```

```
Filter ID:1
Filter Action:1
Source VLANID:304
Destination VLANID:404
Source Address:0.0.0.0
Destination Address:0.0.0.0
Source Port Operation:any
Source Port:0
Destination Port Operation:any
Destination Port:0
```

```
Protocol:0  
$
```

6. Review the rules repository by using the `lsvfilt` command output:

```
$ lsvfilt -a  
Number of active filter rules in kernel = 3  
Filter ID:4  
Filter Action:1  
Source VLANID:406  
Destination VLANID:506  
Source Address::  
Destination Address::  
Source Port Operation:any  
Source Port:0  
Destination Port Operation:any  
Destination Port:0  
Protocol:58
```

```
Filter ID:3  
Filter Action:2  
Source VLANID:306  
Destination VLANID:406  
Source Address::  
Destination Address::  
Source Port Operation:any  
Source Port:0  
Destination Port Operation:any  
Destination Port:0  
Protocol:0
```

```
Filter ID:1  
Filter Action:1  
Source VLANID:304  
Destination VLANID:404  
Source Address:0.0.0.0  
Destination Address:0.0.0.0  
Source Port Operation:any  
Source Port:0  
Destination Port Operation:any  
Destination Port:0  
Protocol:0  
$
```

**Important:** The `rmvfilt -n` command updates the active rules table *and* the rules repository simultaneously in Trusted Firewall Version 1.1.2.

However, in earlier versions of `rmvfilt -n`, the removal is effective in the active rules table only, which forces flushing all repository content to remove one rule.

Table 8-13 lists the commands to manage the filter rule repository.

Table 8-13 Commands to manage the filter rule repository

| Command                                  | Description                                                      |
|------------------------------------------|------------------------------------------------------------------|
| <code>genvfilt &lt;parameters&gt;</code> | Adds a filter rule to the repository                             |
| <code>lsvfilt</code>                     | Lists the rules in the repository                                |
| <code>chvfilt -n &lt;fid&gt;</code>      | Change a rule in repository <i>and</i> in the active rules table |
| <code>rmvfilt -n &lt;fid&gt;</code>      | Remove a rule in repository <i>and</i> in the active rules table |
| <code>rmvfilt -n all</code>              | Flush the rules repository <i>and</i> the rules table content    |

## Managing the active filter rules table

Table 8-14 lists the commands to activate the repository rules, list the active rules' content, and remove one or all rules from active rules and the repository.

Table 8-14 Commands to manage the active filter rules table of Trusted Firewall

| Command                             | Description                                                      |
|-------------------------------------|------------------------------------------------------------------|
| <code>mkvfilt -u</code>             | Activates all filter rules                                       |
| <code>lsvfilt -a</code>             | Lists the loaded active filter rules and their status            |
| <code>rmvfilt -n &lt;fid&gt;</code> | Remove a rule in repository <i>and</i> in the active rules table |
| <code>rmvfilt -n all</code>         | Flush the rules table content <i>and</i> the rules repository    |

For more information about these commands, see the examples that are described in "Managing the filter rules repository" on page 301.

## Magic numbers: Ports and codes

To create more detailed filtering rules, you *must* be familiar with TCP ports, UDP ports, and ICMP codes. For more information about these values, see Appendix A, "Trusted Firewall addendum" on page 315. This list is maintained by the Internet Assigned Numbers Authority (IANA<sup>2</sup>).

### TCP and UDP ports

The ports for TCP and UDP are in `/etc/services` in AIX and Linux.

Port numbers are assigned based on the following ranges:

- ▶ System ports or *well-known ports* (0 - 1023)  
System Ports are assigned by the Internet Engineering Task Force (IETF) process per RFC 6335.
- ▶ User ports or *registered ports* (1024 - 49151)  
User ports are assigned by IANA.
- ▶ Dynamic and *private and ephemeral ports* (49152 - 65535)  
Dynamic ports are not assigned.

For more information about TCP/UDP ports, see the [IANA website](#).

<sup>2</sup> The IANA is responsible for the global coordination of the Domain Name Server (DNS) Root, IP addressing, and other Internet Protocol resources.

The following ports require special attention:

- ▶ Server connections and utilities:
  - Telnet (TCP-UDP port 23)
  - SSH (TCP port 22)
  - FTP (TCP port 20 - 21)
  - FTPS (TCP-UDP port 989 - 990)
  - Telnet over TLS/SSL (TCP-UDP 992)
  - NFS (TCP port 2049)
  - DLPAR Operations/LPM (TCP - UDP port 657)
  - SNMP (TCP port 162 and UDP port 162)
  - SSL (TCP port 446, TCP port 448, and TCP port 449)
  - NIM (TCP port 1058 and TCP port 1059)
  - nimsh (TCP port 3901 and TCP port 3902)
  - NTP (UDP port 123)
  - rlogin (TCP port 513 and TCP port 1023)
  - rlogin (TCP port 513)
  - rsh (TCP port 514)
  - BOOTP (UDP port 67 - 68)
  - TFTP (TCP port 69 and 32768 - 65535)
  - mountd (TCP port 32768 - 65535)
  - portmapper (TCP port 111)
  - LDAP (TCP-UDP port 389)
  - syslog (UDP port 514)
- ▶ Web servers:
  - HTTP (TCP port 80)
  - HTTPS (TCP Port 81 and TCP Port 443)
- ▶ DNS server
  - DNS (TCP-UDP port 53)
- ▶ Email server:
  - POP2 (TCP port 109)
  - POP3 (TCP port 110)
  - IMAP (TCP port 143)
  - SMTP (TCP port 25)

### ***ICMP code***

The Internet Control Message Protocol (ICMP) is defined in RFC792, as part of the Internet Protocol Suite. ICMP has many messages that are identified by a type field. Many of these type fields include a code field. The code field provides more specific information about the message type.

For more information about ICMP types and codes, see [this IANA web page](#).

Both tables are reproduced in Appendix A, “Trusted Firewall addendum” on page 315.

**Note:** The `genvfilt` and `chvfilt` commands require the ICMP *type* value.

### ***ICMPv6 code***

The Internet Control Message Protocol version 6 (ICMPv6) is defined in RFC4443, as part of the Internet Protocol version 6 (IPv6).

ICMP has many messages that are identified by a type field. Many of these type fields include a code field. The code field value depends on the message type and provides another level of message granularity.

ICMPv6 messages are classified into the following categories:

- ▶ *Error messages* are represented in Table A-2 on page 318.
- ▶ *Information messages* are represented in Table A-3 on page 318.

For more information about ICMPv6 types and codes, see [this IANA web page](#).

Both tables are reproduced in Appendix A, “Trusted Firewall addendum” on page 315.

**Note:** The `genvfilt` and `chvfilt` commands require the ICMPv6 *type* value.

### 8.5.3 Removing Trusted Firewall

Complete the following steps to uninstall Trusted Firewall:

1. Query the status of Trusted Firewall by using the `vlantfw -q` command:

```
$ vlantfw -q
vlantfw: TFW=True capability=4
$
```

2. Stop the Trusted Firewall by using the `vlantfw -t` command:

```
$ vlantfw -t
$
```

3. Check the status of Trusted Firewall by using the `vlantfw -q` command:

```
$ vlantfw -q
vlantfw: TFW=False capability=0
$
```

4. Unconfigure the SVM by using the `rmdev` command (Virtual I/O Server restricted shell):

```
$ rmdev -dev svm
svm deleted
$
```

5. Uninstall the Trusted Firewall fileset by using the `installp` command (Virtual I/O Server unrestricted shell):

```
$ oem_setup_env
# installp -u powerscStd.svm.rte
```

.....

Installation Summary

| Name               | Level   | Part | Event     | Result  |
|--------------------|---------|------|-----------|---------|
| powerscStd.svm.rte | 1.1.2.0 | ROOT | DEINSTALL | SUCCESS |
| powerscStd.svm.rte | 1.1.2.0 | USR  | DEINSTALL | SUCCESS |

## 8.6 Troubleshooting Trusted Firewall

In this chapter, we described the installation commands followed by verification and checking procedures. If you encounter any problems with Trusted Firewall, they are often installation-related problems. Check all of the installation verification steps that are described in 8.4, “Installation” on page 288.

If you experience execution trouble, you are facing the following types of issues:

- Performance issues

The active filtering table is sequentially scanned:

- Optimize the placement of the most used rules without modifying the global semantic of the table (the first rule match stops the scan).
- Reduce the size of the table to only the necessary rules (be concise in your rule definition).

- The routed paths are not the expected paths

If this error occurs, validate the firewall routing by using the **tcpdump** command. Check the content of the active rules table and not the repository. Verify whether one rule takes precedence during the packet matching operation.

You rebooted the Virtual I/O Server. The Trusted Firewall SVM is not started, and the Trusted Firewall rules repository is not active. This result might be normal because the Trusted Firewall installation process does not modify the reboot sequence of the Virtual I/O Server.

Complete the following steps to integrate Trusted Firewall into your boot sequence:

1. Escape the restricted shell and go to the /etc directory:

```
$ oem_setup_env
#
```

2. Create the following rc.trustedfw launch script in the /etc directory with the **vi** editor:

```
# pwd
/etc
# ls rc.trusted*
rc.trustedboot rc.trustedfw
```

Escape the **vi** editor with the sequence **:w!** and **:q** commands. The script is shown in Figure 8-18 on page 313.

```
#!/usr/bin/ksh
#
# start Trusted Firewall SVM
echo "Starting Trusted Firewall SVM"
/usr/ios/utis/vlantfw -s
if [ $? -ne 0 ]
then
    echo "Failed to launch Trusted Firewall SVM."
    exit 1
fi
# Activate the Filtering rules of Trusted Firewall
echo "Starting Trusted Firewall SVM"
/usr/ios/utis/mkvfilt -u
if [ $? -ne 0 ]
then
    echo "Failed to activate Filtering rules."
    exit 1
fi
exit 0
```

Figure 8-18 rc.trustedfw script in the /etc directory

3. Integrate the rc.trusted script in /etc/inittab by using the **mkitab** command:

```
# mkitab "trustedfw:2:wait:/etc/rc.trustedfw > /dev/console 2>&1"
# cat inittab
.....
cons:0123456789:respawn:/usr/sbin/getty /dev/console
climgrcim:23456789:once:/usr/ios/sbin/climgr cimserver start > /dev/null 2>&1
trustedfw:2:wait:/etc/rc.trustedfw > /dev/console 2>&1
```

## 8.7 Conclusion

In this chapter, we presented the PowerSC Trusted Firewall design, components, and installation with several supported configurations. We showed you how to define the filtering rules for your frames and how to keep your rules simple and manageable.

The use of a simple approach is sound and secure. A simple approach is a key success factor for any security appliance.







# A

## Trusted Firewall addendum

This appendix provides the tables of Internet Control Message Protocol (ICMP) and ICMPv6 types and codes.

You must specify only the type field of these tables to **genvfilt** and **chvfilt** commands. The code field is not used by Trusted Firewall.

This appendix contains the following topics:

- ▶ “ICMP codes” on page 316
- ▶ “ICMPv6 codes” on page 318

## ICMP codes

For more information about the entire list of ICMP types and codes, see this Internet Assigned Numbers Authority (IANA) [web page](#).

Table A-1 lists the ICMP types and codes.

Table A-1 The ICMP types and codes

| Type                        | Code | Description                               |
|-----------------------------|------|-------------------------------------------|
| 0 - Echo Reply              | 0    | Echo reply (used to ping) 1 and 2         |
| 1 and 2                     |      | <i>Reserved</i>                           |
| 3 - Destination Unreachable | 0    | Destination network unreachable           |
|                             | 1    | Destination host unreachable              |
|                             | 2    | Destination protocol unreachable          |
|                             | 3    | Destination port unreachable              |
|                             | 4    | Fragmentation required                    |
|                             | 5    | Source route failed                       |
|                             | 6    | Destination network unknown               |
|                             | 7    | Destination host unknown                  |
|                             | 8    | Source host isolated                      |
|                             | 9    | Network administratively prohibited       |
|                             | 10   | Host administratively prohibited          |
|                             | 11   | Network unreachable for TOS               |
|                             | 12   | Host unreachable for TOS                  |
|                             | 13   | Communication administratively prohibited |
|                             | 14   | Host Precedence Violation                 |
| 4 - Source quench           | 0    | Source quench (congestion control)        |
| 5 - Redirect Message        | 0    | Redirect Datagram for the Network         |
|                             | 1    | Redirect Datagram for the Host            |
|                             | 2    | Redirect Datagram for the TOS & network   |
|                             | 3    | Redirect Datagram for the TOS & host      |
| 6                           |      | Alternate Host Address                    |
| 7                           |      | <i>Reserved</i>                           |
| 8 - Echo Request            | 0    | Echo request (used to ping)               |
| 9 - Router Advertisement    | 0    | Normal Router Advertisement               |
|                             | 16   | Does not route common traffic             |
| 10 - Router Selection       | 0    | Router discovery/selection/solicitation   |

| Type                                 | Code | Description                           |
|--------------------------------------|------|---------------------------------------|
| 11 - Time Exceeded                   | 0    | Time to live (TTL) expired in transit |
|                                      | 1    | Fragment reassembly time exceeded     |
| 12 - Parameter Problem Bad IP header | 0    | Pointer indicates the error           |
|                                      | 1    | Missing a required option             |
|                                      | 2    | Bad length exposed                    |
| 13 - Timestamp                       | 0    | Timestamp                             |
| 14 - Timestamp Reply                 | 0    | Timestamp reply                       |
| 15 - Information Request             | 0    | Information request                   |
| 16 - Information Reply               | 0    | Information reply                     |
| 17 - Address Mask Request            | 0    | Address Mask Request                  |
| 18 - Address Mask Reply              | 0    | Address Mask Reply                    |
| 19                                   |      | <i>Reserved for Security</i>          |
| 20 - 29                              |      | Reserved for robustness experiment    |
| 30 - Traceroute                      | 0    | Information Request                   |
| 31 - Datagram Conversion Error       |      |                                       |
| 32 - Mobile Host Redirect            |      |                                       |
| 33 - Where-are-you (IPv6)            |      |                                       |
| 34 - Here-I-am (IPv6)                |      |                                       |
| 35 - Mobile Registration Request     |      |                                       |
| 36 - Mobile Registration Reply       |      |                                       |
| 37 - Domain Name Request             |      |                                       |
| 38 - Domain Name Reply               |      |                                       |
| 39 - Skip                            |      |                                       |
| 40 - Security Failure                |      |                                       |
| 41                                   |      | ICMP for experimental mobility        |
| 42 - 255                             | 0    | <i>Reserved</i>                       |

## ICMPv6 codes

For more information about the entire list of ICMPv6 types and codes, see this IANA [web page](#).

Table A-2 lists the ICMPv6 error messages.

Table A-2 ICMPv6 error messages

| Type                        | Code | Description                                 |
|-----------------------------|------|---------------------------------------------|
| 0 - Reserved                |      | <i>Reserved</i>                             |
| 1 - Destination unreachable | 0    | No route to destination                     |
|                             | 1    | Destination communication prohibited        |
|                             | 2    | Beyond scope of source address              |
|                             | 3    | Address unreachable                         |
|                             | 4    | Port unreachable                            |
|                             | 5    | Source address failed ingress/egress policy |
|                             | 6    | Reject route to destination                 |
|                             | 7    | Error in Source Routing Header              |
| 2 - Packet Too Big          | 0    |                                             |
| 3 - Time Exceeded           | 0    | Hop limit exceeded in transit               |
|                             | 1    | Fragment reassembly time exceeded           |
| 4 - Parameter Problem       | 0    | Erroneous header field encountered          |
|                             | 1    | Unrecognized Next Header type encountered   |
|                             | 2    | Unrecognized IPv6 option encountered        |
| 100 - 101                   |      | Private Experimentation                     |
| 102 - 126                   |      | <i>Unassigned</i>                           |
| 127                         |      | <i>Reserved for ICMP v6 Error messages</i>  |

Table A-3 lists the ICMPv6 informational messages.

Table A-3 ICMPv6 Informational messages

| Type                             | Code | Description |
|----------------------------------|------|-------------|
| 128 - Echo request               | 0    |             |
| 129 - Echo Reply                 | 0    |             |
| 130 - Multicast Listener Query   | 0    |             |
| 131 - Multicast Listener Report  | 0    |             |
| 132 - Multicast Listener Done    | 0    |             |
| 133 - Router Solicitation        | 0    |             |
| 134 - Router Advertisement (NDP) | 0    |             |

| Type                                                   | Code | Description                                                                                                             |
|--------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------|
| 135 - Neighbor Solicitation (NDP)                      | 0    |                                                                                                                         |
| 136 - Neighbor Advertisement (NDP)                     | 0    |                                                                                                                         |
| 137 - Redirect Message (NDP)                           | 0    |                                                                                                                         |
| 138 - Router Renumbering                               | 0    | Router Renumbering Command                                                                                              |
|                                                        | 1    | Router Renumbering Result                                                                                               |
|                                                        | 255  | Sequence Number Reset                                                                                                   |
| 139 - ICMP Node Information Query                      | 0    | The Data field contains an IPv6 address that is the subject of this query.                                              |
|                                                        | 1    | The Data field contains a name that is the subject of this query, or is empty, as in the case of a No Operation (NOOP). |
|                                                        | 2    | The Data field contains an IPv4 address that is the subject of this query.                                              |
| 140 - ICMP Node Information Response                   | 0    | A successful reply. The Reply Data field might or might not be empty.                                                   |
|                                                        | 1    | The Responder refuses to supply the answer. The Reply Data field is empty.                                              |
|                                                        | 2    | The Qtype of the Query is unknown to the Responder. The Reply Data field is empty.                                      |
| 141 - Inverse Neighbor Discovery Solicitation Message  | 0    |                                                                                                                         |
| 142 - Inverse Neighbor Discovery Advertisement Message | 0    |                                                                                                                         |
| 143 - Multicast Listener Discovery (MLDv2) reports     |      |                                                                                                                         |
| 144 - Home Agent Address Discovery Request Message     | 0    |                                                                                                                         |
| 145 - Home Agent Address Discovery Reply Message       | 0    |                                                                                                                         |
| 146 - Mobile Prefix Solicitation                       |      |                                                                                                                         |
| 147 - Mobile Prefix Advertisement                      |      |                                                                                                                         |
| 148 - Certification Path Solicitation                  |      |                                                                                                                         |
| 149 - Certification Path Advertisement                 |      |                                                                                                                         |
| 151 - Multicast Router Advertisement                   |      |                                                                                                                         |
| 152 - Multicast Router Solicitation                    |      |                                                                                                                         |
| 153 - Multicast Router Termination                     |      |                                                                                                                         |
| 155 - RPL Control Message                              |      |                                                                                                                         |
| 200 - Private Experimentation                          |      |                                                                                                                         |
| 201 - Private Experimentation                          |      |                                                                                                                         |

| Type           | Code | Description                                 |
|----------------|------|---------------------------------------------|
| 255 - Reserved |      | Expansion of ICMPv6 informational messages. |

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

The IBM Redbooks publication *Using the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, SG24-8100, provides more information about the topic in this document. Note that this publication might be available in softcopy only.

You can search for, view, download or order this document and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Online resources

The following websites are also relevant as further information sources:

- ▶ IBM Fix Central  
<http://www.ibm.com/support/fixcentral>
- ▶ IBM Knowledge Center: Configuring Virtual I/O Server system security hardening  
<https://ibm.co/318nA2q>
- ▶ IBM Knowledge Center: Configuring Virtual I/O Server firewall settings  
<https://ibm.co/2Z0UzUu>

## Help from IBM

IBM Support and downloads:

[ibm.com/support](http://ibm.com/support)

IBM Global Services:

[ibm.com/services](http://ibm.com/services)





**Redbooks**

## **Simplify Management of Security and Compliance with IBM**

SG24-8082-01  
ISBN 0738457973



(0.5" spine)  
0.475" <-> 0.873"  
250 <-> 459 pages







SG24-8082-01

ISBN 0738457973

Printed in U.S.A.

Get connected

