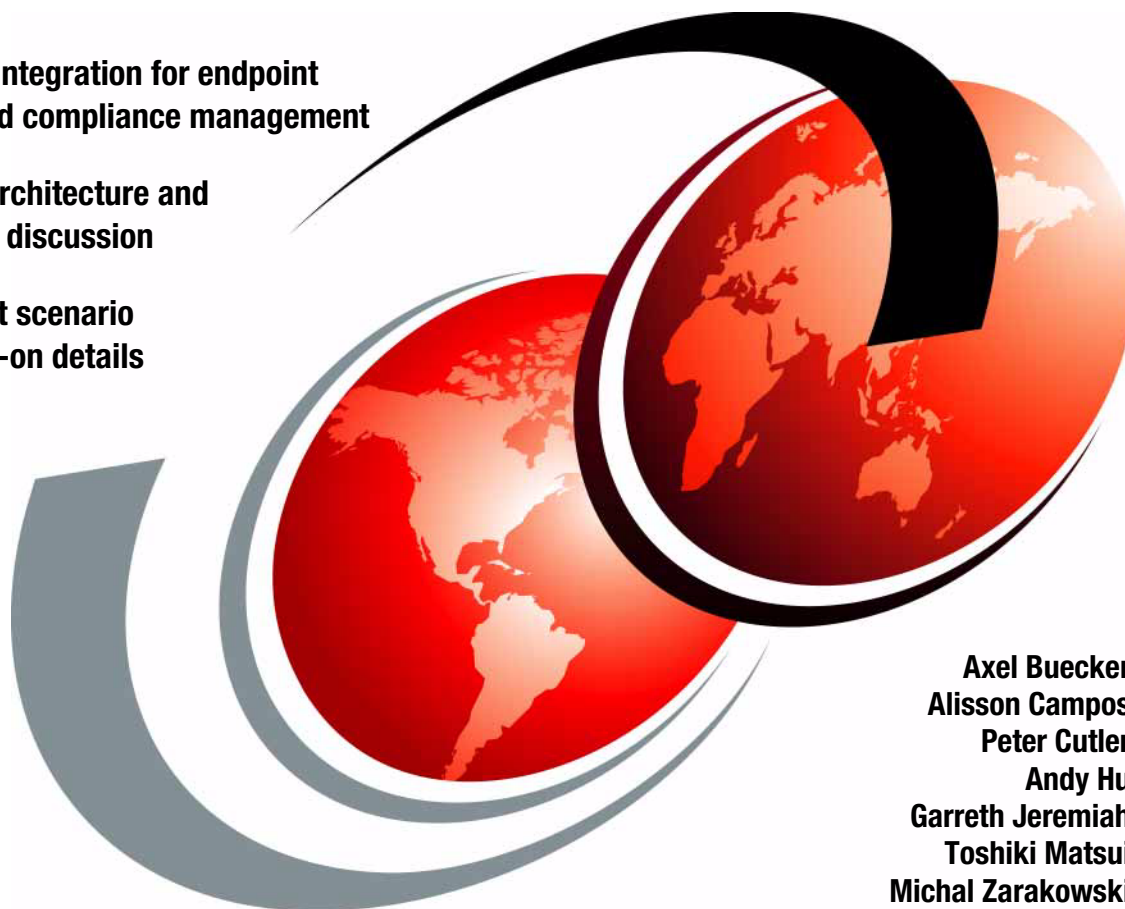


Endpoint Security and Compliance Management Design Guide Using IBM Tivoli Endpoint Manager

Enterprise integration for endpoint
security and compliance management

Complete architecture and
component discussion

Deployment scenario
with hands-on details



Axel Buecker
Alisson Campos
Peter Cutler
Andy Hu
Garreth Jeremiah
Toshiki Matsui
Michal Zarakowski



International Technical Support Organization

**Endpoint Security and Compliance Management
Design Guide Using IBM Tivoli Endpoint Manager**

August 2012

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

First Edition (August 2012)

This edition applies to Version 8, Release 2, of IBM Tivoli Endpoint Manager for Security and Compliance.

© Copyright International Business Machines Corporation 2012. 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

Notices	ix
Trademarks	x
Preface	xi
The team who wrote this book	xii
Now you can become a published author, too!	xiv
Comments welcome	xiv
Stay connected to IBM Redbooks	xv
Part 1. Architecture and design	1
Chapter 1. Business context for endpoint security and compliance management	3
1.1 Drivers that influence security	3
1.1.1 Business drivers that influence security	4
1.1.2 IT drivers that influence security	6
1.2 Introducing the IBM Security Framework	9
1.2.1 Security Governance, Risk Management, and Compliance model ..	10
1.2.2 Network, Server, and Endpoint Domain	11
1.3 IBM Security Blueprint	12
1.4 Endpoint security and compliance management	16
1.5 Conclusion	20
Chapter 2. Introducing the IBM Tivoli Endpoint Manager solution	21
2.1 Overview	22
2.2 IBM Tivoli Endpoint Manager	23
2.2.1 IBM Security Blueprint	24
2.2.2 Platform	27
2.2.3 High-level concept	31
2.2.4 Managed environment	32
2.2.5 Key terms	36
2.3 Tivoli Endpoint Manager for Security and Compliance	43
2.3.1 Security functions	44
2.3.2 Patch management	47
2.3.3 Security configuration management	52
2.3.4 Security Compliance Analytics	56
2.4 Conclusion	62
Chapter 3. IBM Tivoli Endpoint Manager component structure	63

3.1	Logical component overview	64
3.1.1	Fixlet Server	65
3.1.2	Tivoli Endpoint Manager Server	65
3.1.3	Database	69
3.1.4	Console	71
3.1.5	Relay	74
3.1.6	Agent	77
3.1.7	Web Reports Server	82
3.1.8	Analytics	84
3.1.9	Fixlet message	86
3.1.10	Users	86
3.2	Software component breakdown	89
3.2.1	Tivoli Endpoint Manager platform	89
3.2.2	Tivoli Endpoint Manager Server	91
3.2.3	Tivoli Endpoint Manager Web Reports	95
3.2.4	Tivoli Endpoint Manager Console	95
3.2.5	Tivoli Endpoint Manager Relay	97
3.2.6	Tivoli Endpoint Manager Agent	99
3.2.7	Tivoli Endpoint Manager Analytics	102
3.2.8	Fixlet message structure	105
3.3	Network communications and usage	106
3.3.1	Intercomponent traffic	108
3.4	Physical component placement	122
3.5	Conclusion	124
Chapter 4. IT endpoint security and compliance solution design		125
4.1	Design consideration	126
4.1.1	Functional considerations	127
4.1.2	Non-functional considerations	133
4.2	Tivoli Endpoint Manager solution design	142
4.2.1	Deployment planning	143
4.2.2	Deployment design	148
4.2.3	Operational maintenance	154
4.3	Patch Management solution design	156
4.3.1	Before you patch	157
4.3.2	Deploying a patch	159
4.3.3	Patch report	163
4.4	Security configuration management solution design	165
4.4.1	Security configuration management Fixlet Sites	166
4.4.2	Security configuration management Fixlet design	169
4.4.3	Customizing Fixlet Sites	174
4.4.4	Copy wizard	176
4.4.5	Subscribing endpoints to sites	178

4.4.6 Analysis activation	178
4.5 Security and compliance analytics solution design	178
4.5.1 Extract, transform, load process	179
4.5.2 System and hardware guidelines for Analytics	181
4.6 Conclusion	183
Part 2. Customer environment	185
Chapter 5. Overview of scenario, requirements, and approach	187
5.1 Organization profile	188
5.1.1 Current IT infrastructure	188
5.1.2 Security issues within the current infrastructure	193
5.2 Business vision	194
5.3 Business requirements	195
5.3.1 IBM Security Framework mapping to business requirements	195
5.4 Functional requirements	198
5.4.1 IBM Security Blueprint mapping to functional requirements	198
5.5 Design approach	203
5.6 Implementation approach	205
5.7 Conclusion	209
Chapter 6. Phase I: Tivoli Endpoint Manager platform design and implementation	211
6.1 Design	213
6.1.1 Business requirements	213
6.1.2 Functional requirements	213
6.2 Implementation	216
6.2.1 Network considerations	216
6.2.2 Tivoli Endpoint Manager Server	216
6.2.3 Tivoli Endpoint Manager Relay	220
6.2.4 Tivoli Endpoint Manager Agents	225
6.2.5 Asset discovery	228
6.3 Maintenance	230
6.3.1 Tivoli Endpoint Manager health check	230
6.3.2 Performance tuning	232
6.3.3 Maintenance plan	236
6.4 Conclusion	237
Chapter 7. Phase II: Patch Management design and implementation	239
7.1 Design	240
7.1.1 Introduction	240
7.1.2 Defining business requirements	241
7.1.3 Defining functional requirements	242
7.1.4 Patching rating and policy design	242

7.1.5	Defining the patch management process	247
7.1.6	Designing a patch management process	249
7.1.7	Patch management design conclusion	256
7.2	Implementation	256
7.2.1	Introduction	257
7.2.2	Console operation	257
7.2.3	Windows server patching	268
7.2.4	Workstation patching	275
7.2.5	Implementation conclusion	283
7.3	Maintenance	283
7.3.1	Baseline updates	284
7.3.2	Precaching	286
7.3.3	Corrupted patches	286
7.3.4	Minimizing endpoint reboots	287
7.3.5	Locking endpoints	289
7.3.6	Patching overview dashboard	289
7.3.7	Maintenance conclusion	290
7.4	Conclusion	290
Chapter 8. Phase III: Security policy configuration design and implementation		293
8.1	Design	294
8.1.1	Current endpoint security policies	295
8.1.2	Tivoli Endpoint Manager security policy customization	298
8.1.3	Designing a new policy model	301
8.2	Implementation	305
8.2.1	Create Custom Site for policies	306
8.2.2	Customizing SCM Fixlets	314
8.2.3	Compliance evaluation	320
8.2.4	Fixlet remediation	324
8.2.5	Practice	336
8.3	Project scope change	337
8.3.1	Monitoring anti-virus health	337
8.4	SCM administration and maintenance	345
8.5	Real-time reports	348
8.5.1	Tivoli Endpoint Manager real-time reports requirements	348
8.5.2	Tivoli Endpoint Manager real-time reports basic configuration	349
8.6	Conclusion	355
Chapter 9. Phase IV: Security Compliance Analytics reporting		357
9.1	Design	358
9.1.1	Hardware design	358
9.1.2	Computer grouping	359

9.1.3 Users and roles	360
9.1.4 Security configuration management content	360
9.2 Implementation	361
9.2.1 Installing the Security Compliance Analytics solution	361
9.2.2 Implementing users, computer groups, and permissions	363
9.3 Usage	366
9.3.1 Report execution	367
9.3.2 Exception management	378
9.4 Conclusion	382
Appendix A. Service offerings	383
Guiding principles for Tivoli Endpoint Manager	383
Scale and expertise	384
Speed	384
Solutions, process mapping, and integrations	384
Service offerings overview	386
Platform deployment services	388
Solutions services	390
Advanced services offerings	391
Conclusion	392
Appendix B. IBM deploys Tivoli Endpoint Manager internally	393
Continuous compliance with internal security policies	394
A pilot program builds the business case	394
A 78% decrease in endpoint security issues	395
Millions in savings	396
Advanced investigations for sophisticated security challenges	396
Related publications	399
IBM Redbooks	399
Online resources	399
Help from IBM	399

Notices

This information was developed for products and services offered in the U.S.A.

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 give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 supply in any way it believes appropriate without incurring any obligation to you.

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.

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	Global Technology Services®	Redpaper™
AppScan®	Guardium®	Redbooks (logo)  ®
Bigfix®	IBM®	Service Request Manager®
Cognos®	InfoSphere®	SiteProtector™
Command Center®	PowerPC®	System x®
Common Platform®	Proventia®	Tivoli®
developerWorks®	Rational®	
Fixlet®	Redbooks®	

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation 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.

Bigfix, Fixlet, and B device are trademarks or registered trademarks of BigFix, Inc., an IBM Company.

Command Center, and OpenPages device are trademarks or registered trademarks of OpenPages, Inc., an IBM Company.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Preface

Organizations today are more widely distributed than ever before, which can make systems management tasks, such as distributing software, patches, and security policies, extremely challenging.

The IBM® Tivoli® Endpoint Manager platform is architected for today's highly diverse, distributed, and complex IT environments. It provides real-time visibility and control through a single infrastructure, single agent, and single console for systems lifecycle management, endpoint protection, and security configuration and vulnerability management. This platform enables organizations to securely manage their global IT infrastructures faster and more accurately, resulting in improved governance, control, visibility, and business agility. Plus, it gives organizations the ability to handle tomorrow's unforeseen challenges.

In this IBM Redbooks® publication, we provide IT security professionals with a better understanding around the challenging topic of endpoint management in the IT security domain. We focus on IBM Tivoli Endpoint Manager for Security and Compliance and describe the product architecture and provide a hands-on design guide for deploying the solution.

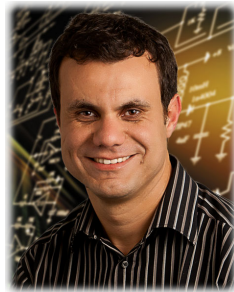
This book is a valuable resource for security professionals and architects who want to understand and implement a centralized endpoint management infrastructure and endpoint protection to better handle security and compliance challenges.

The team who wrote this book

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



Axel Buecker is a Certified Consulting Software IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of Software Security Architecture and Network Computing Technologies. He holds a degree in computer science from the University of Bremen, Germany. He has 25 years of experience in various areas related to Workstation and Systems Management, Network Computing, and e-business Solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.



Alisson Campos is a Certified Senior Software Specialist at IBM Brazil with a specialization in IBM Security solutions. He is also the Technical Leader for the Tivoli Security Software Group team in Brazil. He has 14 years of experience in designing and implementing large and complex Information Security and Service Management Solutions in several client organizations and industries, based on client business needs and technical requirements. He is a Master Certified IT Specialist by the Open Group and he holds numerous Information Security, Network, Operating System, and Services Management Certifications. He holds a degree as a Systems Analyst from University Mackenzie and a Post-Graduation in Network Solutions from FASP.



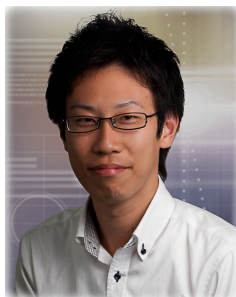
Peter Cutler is a Client Technical Professional for the IBM Software Group Security team in the United Kingdom, demonstrating, implementing, and consulting on Security technologies. Peter has an enthusiastic focus on the IBM Tivoli Security portfolio, specifically Endpoint Protection and Intrusion Prevention, while also consulting on Identity and Access management solutions. Peter has worked for several years in the Semiconductor manufacturing industry as a design engineer for wafer handling robotics and power semiconductors. Peter holds a Bachelor of Engineering (BEng) degree in Electro Mechanical Engineering with Business Management from the University of Surrey, United Kingdom.



Andy Hu is an IBM Security Solutions Software Engineer at the IBM Software lab in Taiwan. He has worked on IBM endpoint and management security products, such as IBM Proventia® Server Protection for Windows, Proventia Desktop, and Site Protector. Before joining IBM, he was a senior test engineer at TrendMicro specializing in endpoint security products. He helped to develop the first Trend Micro endpoint security product to fully integrate with the Bigfix® Enterprise Suite (BES) - Core Protection Module for Windows and later, Core Protection Module for Mac. He has two years of experience in developing, operating, and managing BES before IBM acquired BigFix and later renamed it to IBM Tivoli Endpoint Manager. Andy holds a Bachelors degree in Computer Science from the University of Auckland, New Zealand, and a Masters degree in Business Information Systems.



Garreth Jeremiah is a Senior Security Specialist with IBM Global Services. Garreth focuses on Endpoint Protection, Perimeter Protection, and Information Security technologies, such as DLP, Firewall, Intrusion Prevention, and Encryption systems. Garreth has over 16 years of experience in the security field. He has over 14 years of experience in Solution Architecture Design and Implementation in EMEA, AP, and North America across diverse industries. He began his post-military career with IBM in EMEA performing penetration testing, ethical hacking, and support to IBM Network Services, globally. Garreth has several patents in the area of Information Security and is currently working with the IBM Global CISO focusing on endpoint security and compliance. He is the Global Architect for the IBM CIO internal Tivoli Endpoint Manager deployment providing compliance enforcement and patching for workstations across the enterprise.



Toshiki Matsui is an IT Specialist for IBM Systems Engineering (ISE) in IBM Japan. He has two years of experience in the IT Service Management field and specializes in Tivoli Automation products, such as Tivoli Endpoint Manager for Lifecycle Management. He works on consultation and technical support as a delivery support team member and has considerable experience in providing technical guides and product workshops to delivery engineers. Before supporting Tivoli Endpoint Manager, he worked with Tivoli Process Automation Engine products, such as Tivoli Service Automation Manager.



Michal Zarakowski is a Development Leader for the Security and Compliance Management solution for IBM Global Technology Services® based on the IBM Tivoli Endpoint Manager platform. Working for more than five years in the security and compliance area, Michal was involved in the development of IBM Tivoli Security Compliance Manager and multiple client interactions related to the support and maintenance for the product. He gained experience in the area of compliance verification and vulnerability detection using both platforms. Michal manages IBM client and IBM Business Partner meetings in the Center for Business Innovation in Krakow, presenting the IBM Tivoli Endpoint Manager for Security and Compliance. He holds a Masters degree in Computer Science from the AGH University of Science and Technology, Krakow, Poland.

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

Aaron Bauer, Jacob Campbell, James Evans, Jonathan Fan, Stephen Hull, Lloyd Jobe, Benjamin Kus, Wei Lee, Dexter Liu, Chris Loer, Michael Luu, Naveed Makhani, Anna Min, Daniel Montgomery, Michael Ottum, Noah Salzman, Jeff Spitulnik, John Talbert, May Yang
IBM

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

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

- ▶ Send your comments in an email to:

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 IBM Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent IBM Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Part 1

Architecture and design

In part 1, we describe the overall business context for IT endpoint security and compliance management and explain the general business requirements for an endpoint security and compliance management solution. We then describe a framework for providing functionality throughout an organization. In addition, we introduce the high-level components and new concepts for the design of an endpoint security and compliance management solution that uses IBM Tivoli Endpoint Manager.

Additionally, we provide an understanding of the high-level product architecture of IBM Tivoli Endpoint Manager for Security and Compliance.



Business context for endpoint security and compliance management

In this chapter, we describe the overall *business context* for IT endpoint security and compliance management. We examine the drivers that influence *why* and *how* IT endpoint security and compliance management must be conducted in a certain business context. We also describe the business requirements for an IT endpoint security and compliance management solution. This chapter is organized in the following manner:

- ▶ “Drivers that influence security” on page 3
- ▶ “Introducing the IBM Security Framework” on page 9
- ▶ “IBM Security Blueprint” on page 12
- ▶ “Endpoint security and compliance management” on page 16

1.1 Drivers that influence security

Most projects are driven by both business and IT drivers, although we can probably agree that business drivers are almost always the initiating factor.

We look at these influencing factors:

- ▶ **Business drivers:** Business drivers measure value, risk, and economic costs that influence their approach to IT security. Value drivers determine the worth of the assets of the system to the business and the worth of the business itself. Risk drivers involve compliance, corporate structure, corporate image, and the risk tolerance of the company. Economic drivers determine the productivity impact, competitive advantage, and system cost.
- ▶ **IT drivers:** IT drivers represent operational constraints in the general IT environment. For example, the complexity of a system, including its environment, that is exposed to internal and external threats presents risks that the organization must address.

Business drivers also represent issues and consequences of significance to the stakeholders of the managed business system. This set of drivers might vary from industry to industry, from organization to organization in the same industry, and even between different business applications in an organization.

IT drivers represent technical considerations that affect the trustworthiness of the IT environment and likely the managed business systems as a whole. IT drivers are universal and must be considered within the context of the business drivers in all efforts. The combination of business and IT drivers represents the key initiatives for security management.

1.1.1 Business drivers that influence security

Business drivers represent a relationship between the IT organization and the rest of the business. They refer to business values that must be supported by the IT security infrastructure.

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 appropriate response or function with no errors. *Reliable* means that the same result occurs all the time. Any IT system must consistently provide stakeholders with the expected results.

Security events and incidents might affect 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.

Service-level agreements

This driver applies to circumstances where security threats and threat agents can affect the ability of an organization to conduct business. Service-level agreements (SLAs) incorporate acceptable conditions of operation within an organization. SLAs might vary from business system to business system or application to application. Availability of systems, data, and processes is a condition commonly referenced within SLAs.

IT asset value

From the business perspective, the IT asset value directly relates to the value of the business transactions that it supports. These assets might be tangible or intangible. For an e-retailer, these assets are tangible assets. For a financial services company, the asset might be the client information or other data used in transactions of the system.

Protection of the business asset value or brand image

This driver captures the desire of the firm to protect its image. The loss of goodwill from a security incident or attack has a direct consequence to the business. Therefore, the security measures are likely to be proportional to the consequence. When the desire to avoid negative publicity increases upon encountering a security breach, the stipulation for this driver becomes stronger.

Legal and regulatory compliance

Legal and regulatory compliance refers to the externally imposed conditions on the transactions in the business system and the company, including the rules and policies imposed by regulatory and government agencies. Civil, criminal liability, or regulatory penalty from a security incident or attack have a negative impact on the business. Therefore, the amount of regulation and steps to ensure compliance must be factored in this driver, which includes privacy issues, the ability to prove the transaction initiator, and proving compliance.

An implemented log management system can tell who did what, where, and when. Log management, therefore, is part of an IT security compliance management system. For the retention period of the logs, it is ensured that the necessary information is available and can be analyzed or interpreted to a level that can help management to better investigate security incidents or comply with external regulation or laws. Compliance is a key business driver today, and log management must be a part of every IT security compliance management solution. But, it can also be implemented alone as an initial step toward a larger IT security compliance initiative.

Many international standards and regulatory controls require logging to be enabled and implemented. Also, these logs must be analyzed periodically and

stored for a specific period of time, depending on the particular standard or regulatory control.

Contractual obligation

Security measures for an IT system are likely to be proportional to the consequences encountered when the business encounters contractual liability from a security attack. Depending on the structure and terms of the contract, the consequence might lead to financial loss or liability. For example, when security incidents are encountered, the business might be unable to fulfill its contractual obligations of providing goods or services.

Financial loss and liability

Direct or indirect financial loss is a consequence to the business as a result of a security incident. Direct loss might include theft of an asset, theft of a service, or fraud. Indirect loss might include loss based on civil or criminal court ruling, loss of good will, or re-prioritized budget allocation. This driver identifies the fact that security measures for an IT system are likely to be in proportion to these consequences.

Critical infrastructure

This driver applies where security threats or threat agents can have a major impact on services or resources that are common to, or shared among, a community of businesses, the population at large, or both. Examples include telecommunications, electrical power, transportation systems, and computing. The loss of critical infrastructure by its provider might have a ripple effect, causing secondary losses and driving security decisions for affected businesses and resources. An important part of risk analysis is identifying critical infrastructure.

Safety and survival

This driver applies where security threats and threat agents can have a major impact on aspects of human life, government function, and socio-economic systems. Examples of processes to be considered for their safety and survival impact include the continuity of a critical infrastructure, medical system, life support, or other high-impact or time-dependent process.

1.1.2 IT drivers that influence security

IT drivers make up the second group of key security initiatives. These universal drivers must be considered in every modern IT solution in a manner commensurate with the risks and consequences of a related failure or incident.

Internal threats and threat agents

Security-related failures and incidents are caused by threats or threat agents found within the physical and logical boundaries of the organization or enterprise that operates and controls the IT system. These threats and threat agents might be associated with technology or people.

An example of an internal threat is a poorly designed system that does not have the appropriate controls. An example of an internal threat agent is a person who uses an ability to access the IT system or influence business or management processes to carry out a malicious activity.

External threats and threat agents

Security-related failures and incidents are caused by threats or threat agents found outside the physical and logical boundaries of the organization or enterprise that operates and controls the IT system. These threats and threat agents are also associated with technology or people. They seek to either penetrate the logical or physical boundary to become internal threats or threat agents, or to influence business or management processes from outside the logical or physical boundary.

Examples of external threats are single points of failure for one or more business or management processes that are outside the enterprise boundary, such as a power system grid or a network connection, or a computer virus or worm that penetrates the physical or logical network boundary. An example of an external threat agent is a malicious hacker, or someone who gained the ability to act as an insider, by using personal electronic credentials or identifying information.

IT service management commitments

This driver identifies the fact that failure to manage the operation of the IT system might result in security exposures to the business. This driver can be divided into two categories: IT service delivery and IT service support.

- ▶ **Service delivery commitments**

The failure of the IT system to meet its metrics for managing itself can be viewed as a security exposure to both business or management processes.

An example of security exposure for service delivery is when IT operations processes cannot respond to critical events in a timely manner. Another example is when IT resilience processes cannot recover from a denial of service attack in a timely manner, resulting in a loss of capacity or response time for business processes.

► **Service support commitments**

The failure of the business or IT management system to meet its service level agreements (SLAs) can be viewed as a security exposure to business or management processes.

An example of security exposure for service support is a situation in which the customer relationship processes do not add, modify, or remove users from access control lists in a timely manner.

IT environment complexity

The complexity of the IT environment might contribute to the security or insecurity of the IT system. The IT environment reflects the infrastructure on which the business system is placed.

For example, any IT environment that is connected to the intranet or extranet is exposed to internal or external threats or threat agents and requires specific security responses. A stand-alone facility for our system represents the lowest complexity. A hosting facility with other systems and other firms represents a more complex environment. An environment with a larger number of systems, varied network access paths, or a complex architecture, is a complex IT environment.

Business environment complexity

Because most businesses rely on IT, most business environments are an interconnected set of businesses, each with its own complex IT environment, business processes, and IT management processes. This complexity might contribute to the security or insecurity of the IT system.

Audit and traceability

This driver identifies the need for the IT system to support an audit of information contained within the system, whether it is associated with management data or business data.

IT vulnerabilities: Configuration

Configuration vulnerabilities are potentially present in every IT system, providing an opening to a potential attack based on the system and how it is designed and set up.

IT vulnerabilities: Flaws

Software flaws potentially exist in every IT system. These flaws represent vulnerabilities that were not detected and are not evident in the design documents. Therefore, they are an unexpected deviation from what was

designed. An example is a defect in an operating system or application that is discovered after implementation.

IT vulnerabilities: Exploits

The basic design of software in any IT system might be exploited by threats or threat agents as a part of an attack on the IT system, the business, or the management processes, which might include the use of a function within a system in a way to compromise the system or underlying data. Although certain people might define an exploit as both the flaw and the method, we treat them separately because an exploit might involve the use of normal functions as designed in an unusual manner to attack the system. The exploits can also be viewed as the openings or avenues that an attacker can use.

Now it is time for us to introduce the IBM Security Framework, which focuses on the *what*, not the *how*. It can help you translate your requirements into coarse-grained business solutions, not into specific IT components or IT services.

1.2 Introducing the IBM Security Framework

Business leaders are expected to manage risk in their areas of responsibility in the same way that CFOs manage risks in their domains. Security risks and the potential impact on IT need to be communicated to executive peers in business terms. Additionally, they need to align IT security controls with their business processes, monitor and quantify IT risk in business terms, and dynamically drive business-level insight at the executive level. Finally, business leaders need to manage risk and orchestrate security operations in a way that enforces compliance and optimizes business results.

As an organization secures its business processes, a business-driven approach needs to become the guiding influence for ensuring that all the different security domains work together in a holistic and synergistic manner, in alignment with the over-arching business objectives. Otherwise, the risk stance of the organization becomes vulnerable due to misalignment of priorities between IT and the business strategy. Using a standards-based approach to map business drivers to IT security domains is often difficult and is often an afterthought.

IBM created a comprehensive IT security framework (Figure 1-1 on page 10) that can help ensure that every necessary IT security domain is properly addressed when using a holistic approach to business-driven security.

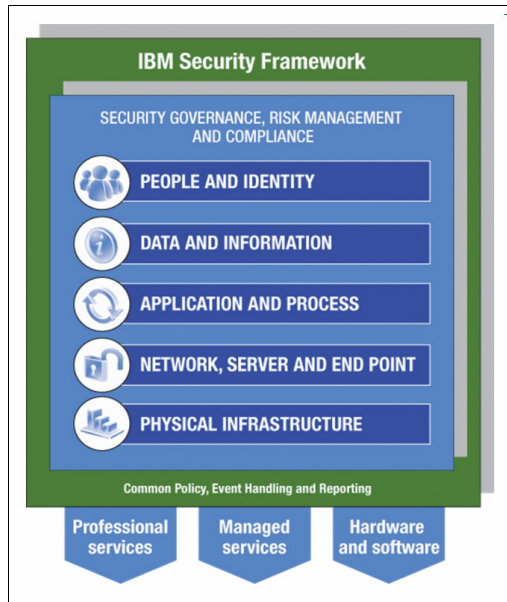


Figure 1-1 The IBM Security Framework

IBM provides the full breadth and depth of solutions and services that can enable organizations to take this business-driven, secure-by-design approach to security in alignment with the IBM Security Framework. Comprehensive professional services, managed services, and hardware and software offerings are available from IBM to support your efforts in addressing the different security domains covered by the IBM Security Framework.

1.2.1 Security Governance, Risk Management, and Compliance model

Every organization needs to define and communicate the principles and policies that guide the business strategy and business operation. In addition, every organization must evaluate its business and operational risks. It must develop an enterprise security plan to serve as a benchmark for the execution and validation of the security management activities that are appropriate for the organization.

These principles and policies, the enterprise security plan, and the surrounding quality improvement process represent the enterprise Security Governance, Risk Management, and Compliance model. Specifically, the five security domains have the following requirements and compliance criteria:

- ▶ People and Identity
This domain covers how to ensure that the correct people have access to the correct assets at the correct time.
- ▶ Data and Information
This domain covers how to protect critical data in transit or at rest across the organization.
- ▶ Application and Process
This domain covers how to ensure application and business services security.
- ▶ Network, Server, and Endpoint (IT infrastructure)
This domain covers how to stay ahead of emerging threats across IT system components.
- ▶ Physical Infrastructure
This domain covers how to use the capability for digital controls to secure events, on people or things, in the physical space.

We now look at the Network, Server, and Endpoint domain. We focus on this domain because it is the driving factor for implementing an endpoint security and compliance management solution. For more information about the other IBM Security Framework domains, see the IBM Redpaper™ publication *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

1.2.2 Network, Server, and Endpoint Domain

Organizations need to *preemptively* and *proactively monitor* the operation of the business and the IT infrastructure for *threats* and *vulnerabilities* to avoid or reduce breaches.

The Security Governance, Risk Management, and Compliance model can provide guidance on the business implications of technology-based risks. In practice, the definition, deployment, and management of technology-based threats, and the technical aspects of incident response, can be delegated to operational management and staff, or outsourced to a service provider.

Security monitoring and management of the network, server, and endpoints of the organization are critical to staying ahead of emerging threats that can

adversely affect system components and the people and business processes that they support. The need to identify and protect the infrastructure against emerging threats dramatically increased with the rise in organized and financially motivated network infiltrations. Although no technology is perfect, the focus and intensity of security, monitoring, and management can be affected by the type of network, server, and endpoints deployed in the IT infrastructure and how those components are built, integrated, tested, and maintained.

Endpoints need to be kept secure to effectively manage risk. In far-reaching environments, the number of endpoints is growing at unprecedented rates. These endpoints are commonly used on unsecured networks as they cross physical boundaries from the workplace to the home.

Figure 1-2 shows a summary and additional topics to be addressed within the Network, Server, and Endpoint domain.

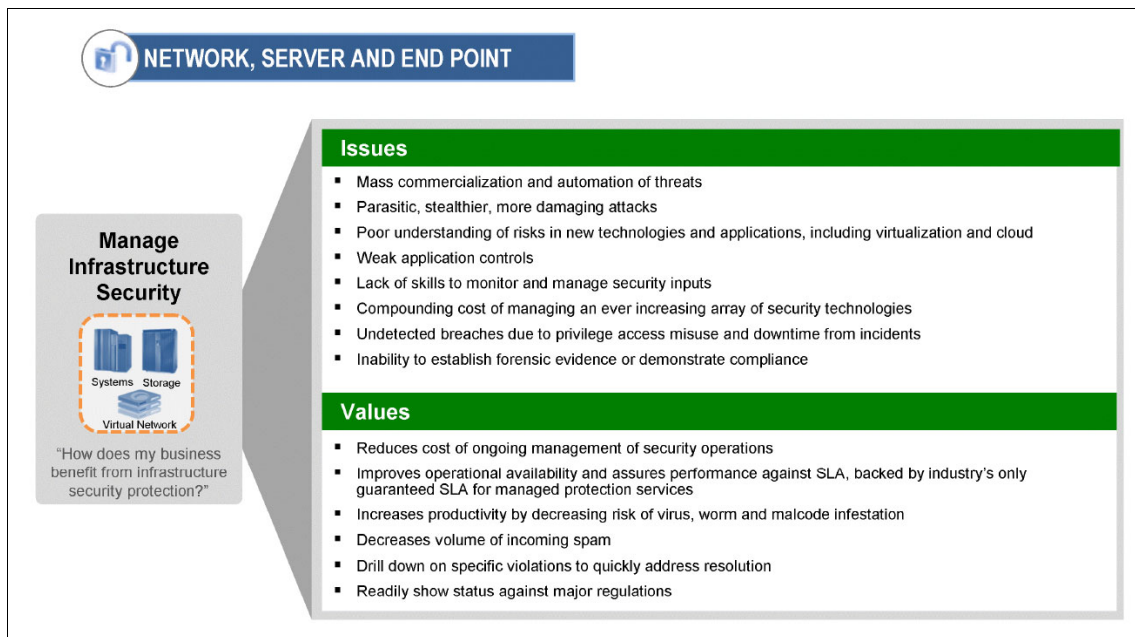


Figure 1-2 The Network, Server, and Endpoint domain of the IBM Security Framework

1.3 IBM Security Blueprint

The IBM Security Framework divides the area of business-oriented IT security into five domains. The next step is to break down these domains into further detail to work toward a common set of core security capabilities 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-independent and solution-independent 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, focusing 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 these areas, but does not replace them. Using the IBM Security Blueprint in this way can provide a solid approach to considering the security capabilities in a particular architecture or solution.

IBM uses a high-level, service-oriented 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. To better position and understand the IBM Security Blueprint, see Figure 1-3.

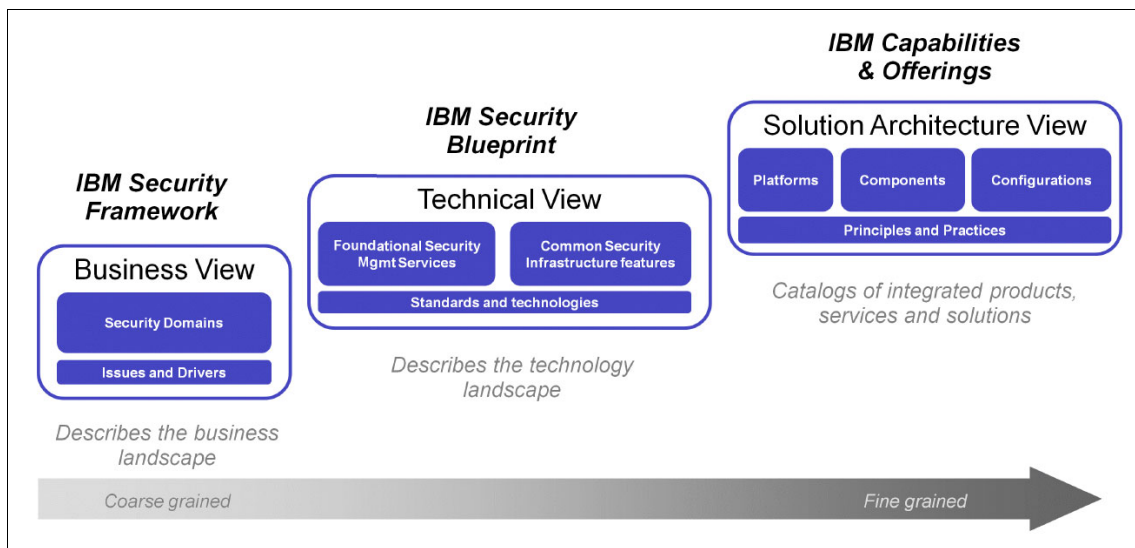


Figure 1-3 IBM Security Blueprint positioning

The left portion of Figure 1-3 on page 13 represents the IBM Security Framework, which describes and defines the security domains from a business perspective. It was covered in 1.2, “Introducing the IBM Security Framework” on page 9.

The middle portion in Figure 1-3 on page 13 represents the IBM Security Blueprint, which describes the IT security management and IT security infrastructure capabilities needed in an organization. As discussed earlier, the IBM Security Blueprint is product and vendor independent.

The right portion of Figure 1-3 on page 13 represents the solution architecture views, which describe specific deployment guidance particular to an IT environment. Solution architecture views provide details about specific products, solutions, and their interactions.

Figure 1-4 on page 15 highlights the components and subcomponents of the IBM Security Blueprint that must be examined for every solution in the Network, Server, and Endpoint 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 associated with the particular business security domain, in this case, Network, Server, and Endpoint.

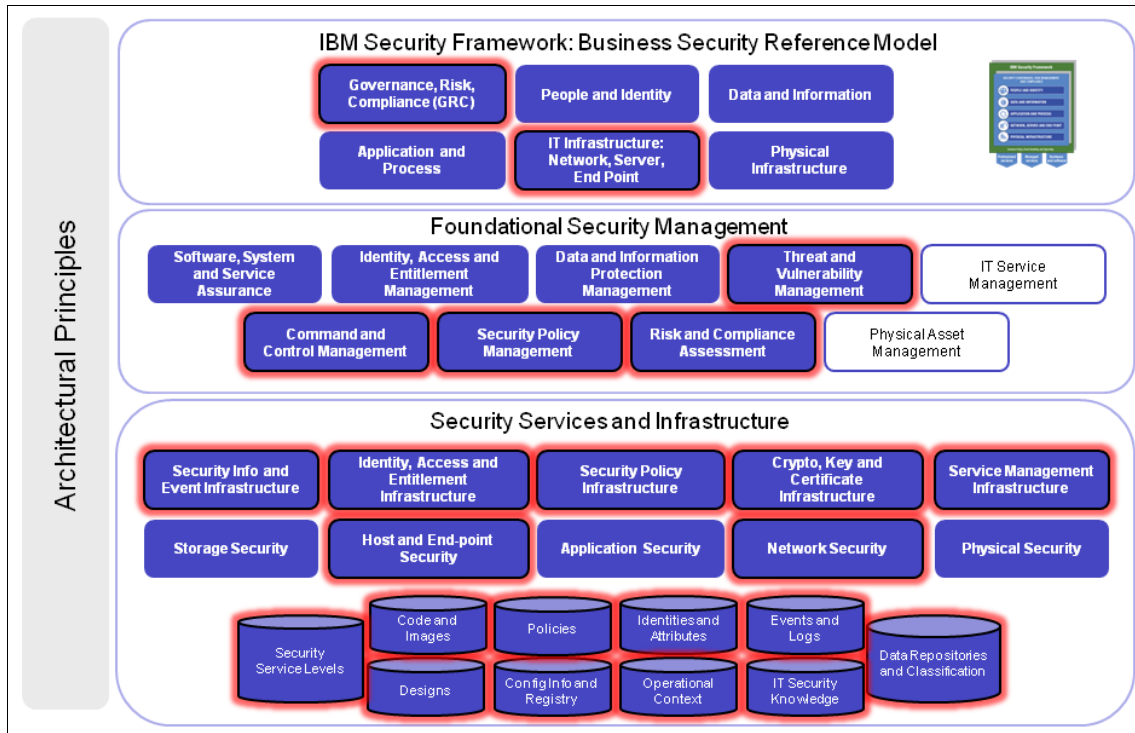


Figure 1-4 IBM Security Blueprint components for the Network, Server, and Endpoint solution pattern

We can see in Figure 1-4 that almost all infrastructure components can be required for a Network, Server, and Endpoint security solution apart from Storage Security, Application Security, and Physical Security. The reason why those components are not included is that they are mostly covered by other domains of the IBM Security Framework.

If you want to learn more about the Foundational Security Management and the Security Services and Infrastructure subcomponents, see *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

In the next section, we examine the endpoint security and compliance management.

1.4 Endpoint security and compliance management

Organizations can have few or as many as several hundreds of thousands of endpoints that must be tightly controlled to effectively manage risk. In far-reaching environments, the numbers and varieties of servers, desktops, notebooks, mobile IT devices, and specialized equipment, such as point-of-sale devices, ATMs, and self-service kiosks, which are known collectively as *endpoints*, are growing at unprecedented rates. With rapidly increasing numbers of remote workers and roaming devices, there is no well-defined perimeter anymore. The perimeter, by necessity, must be the endpoint itself.

The pains caused by security and compliance issues, however, are not only in the attacks, but also in the way that organizations protect themselves. Protection can be costly, complex, and time-consuming, stretching IT staff thin and driving costs even higher. After security is in place, many organizations must prove compliance with internal policies, security standards, and government regulations. In addition to the pain involved in achieving initial compliance, *compliance drift* is another key concern. After compliance levels are attained, organizations must ensure that the compliance levels are continuously maintained.

Controlling costs is high on the priorities of IT leaders, affecting IT teams that are being asked to do more with less. Organizations require a tool that is simple and scalable. The tool must automate management capabilities so that costs and complexity are controlled, while still being able to meet compliance mandates.

The Tivoli Endpoint Manager Agent constantly monitors endpoint compliance, communicating endpoint status and providing real-time visibility through a single, centralized console. And by using a continually updated policy database of thousands of IBM Fixlet® messages, and providing the ability for clients to create their own Fixlets, the Tivoli Endpoint Manager Server always contains current endpoint compliance, configuration, and change status, enabling real-time reporting. Reporting through a centralized console provides real-time visibility into the configuration and compliance status in various easy-to-understand reports.

Managing compliance can be seen as operating in accordance with expectations. These expectations are formalized from mission statements and requirements that are derived from external laws and regulations, such as the following examples:

- ▶ Payment Card Industry Data Security Standard (PCI DSS)
- ▶ ISO 27001/27002
- ▶ Sarbanes-Oxley
- ▶ Basel II
- ▶ Food and Drug Administration (FDA)
- ▶ NERC-CIP
- ▶ Health Insurance Portability and Accountability Act (HIPPA)
- ▶ Federal Information Security Management Act (FISMA)
- ▶ Gramm-Leach-Bliley Act (GLBA)

Addressing these challenges requires a flexible, cross-platform approach that provides the visibility of all of the IT endpoints of the organization and also the control to manage the configuration. Continually enforcing the configuration of endpoints helps to reach a compliant state and therefore satisfies the stakeholders in an organization.

Audit reports help document the level of compliance to any internal policy, external regulation, or applicable law. The mandate to produce these reports can be a time-consuming process. Considering the management of endpoints starts with managing the configuration of these endpoints. Enforcing the configuration for a particular endpoint requires an intelligent agent to be deployed that constantly evaluates the state of these settings. The unified Agent deployed with Tivoli Endpoint Manager can identify current patch and configuration levels, comparing them against defined policies. It can then apply operating system and application updates, regardless of the endpoint location, connection type, or status, and continuously enforce policy compliance, even if endpoints are not connected to the network at all times.

Compliance versus control: If you were audited (or if you audited someone else), you probably know that there is a difference between being in compliance and being in control:

- ▶ When you are in compliance, all 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 end up being 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 endpoints has to be a holistic approach that starts with gaining visibility into the endpoints within the infrastructure. The saying that you cannot manage what you cannot see is as true in the realm of security as it is anywhere else. To properly remediate vulnerabilities and enforce configurations, you must first know which endpoints are at risk. Many failed audits result from poor visibility into endpoint vulnerabilities due to endpoint configuration drift, or the inability to rapidly deploy (and confirm) the application of patches and updates.

Delivering a range of capabilities through a single intelligent agent at the endpoint and a light but scalable platform can provide key security capabilities:

- ▶ Security standards support
Using the Tivoli Endpoint Manager platform and the single intelligent Agent can shorten patching times and can provide feedback for assurance that the actions taken are applied to endpoints even on low-bandwidth, globally distributed networks, regardless of whether the endpoint is in or outside of the organization firewall.
- ▶ Security configuration management
The capability to deliver meaningful information about the health and security of endpoints regardless of location, operating system, applications installed, or connection type.

- ▶ Vulnerability management
Assesses endpoints against Open Vulnerability and Assessment Language¹ (OVAL)-based vulnerability definitions and reports on non-compliance in real time to support the elimination of known vulnerabilities across endpoints.
- ▶ Patch management
Using the Tivoli Endpoint Manager platform and the single intelligent Agent can shorten patching times and provide feedback for assurance that the actions taken apply to endpoints even on low bandwidth, globally distributed networks regardless of whether the endpoint is inside or outside of the firewall of the organization.
- ▶ Client manager for endpoint protection
Provides a single point of control for managing third-party antivirus and firewall products from vendors, enabling organizations to enhance the scalability, speed, and thoroughness of protection solutions.

A unified solution that incorporates this range of capabilities with a single console can help organizations move to a unified management approach, enhancing visibility and control. It can help bridge the gap between the establishment of security strategy and policy, execution of that strategy, real-time operational endpoint management, and security and compliance reporting, for example.

IBM Tivoli Endpoint Manager offers functionality that operates from the same console, Management Server, and endpoint Agent. This approach can help to consolidate tools, reduce the number of endpoint agents, and lower your management costs.

Endpoint security and compliance management are important to managing IT security. 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 continuum. 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, their product development, and their daily operations. It is factored into the initial design, not added afterward. This methodology allows an organization to securely and safely adopt new forms of technology that run on new endpoint devices, such as cloud computing or virtualization used on netbooks and mobiles. Business models, such as teleworking and outsourcing can be used more safely for cost benefit, innovation, and a shorter time to market.

¹ For more information about OVAL, go to this website: <http://oval.mitre.org/>

1.5 Conclusion

In this first chapter, we examined the business and technology drivers that influence security in organizations. We then examined the approach and tools available for considering a holistic approach to securing the IT operations of an organization. The IBM Security Framework and the 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 taking a holistic approach to business-driven security. We also discussed the business context for specifically endpoint security and compliance management.

This IBM Redbooks publication proceeds with a technical discussion around endpoint security and configuration to meet compliance. The second part of this book introduces a scenario for an organization that is looking for a solution to endpoint security and compliance management. Chapter 5, “Overview of scenario, requirements, and approach” on page 187 discusses the business and functional requirements of the solution. It uses the IBM Security Blueprint as an example of how an organization can comprehensively achieve core capabilities to secure its operations as it implements a holistic approach to security.



Introducing the IBM Tivoli Endpoint Manager solution

In this chapter, we introduce the IBM Tivoli Endpoint Manager solution, which offers various features to help administrators be in charge of their endpoint environments. While introducing the platform and presenting methodologies and processes used by Tivoli Endpoint Manager, we focus on the security and compliance capabilities of the product.

2.1 Overview

There is an unprecedented increase in the volume, virulence, and velocity of new security threats plaguing every computing asset. Organizations recognize that traditional systems management tools do not provide the ability to effectively manage change to successfully defend against these new attacks. Traditional tools cannot defend against attacks across large, geographically distributed environments with a growing number of mobile computers that operate outside of their secured network perimeter. Revealed vulnerabilities are being exploited by the attackers within hours of releasing patches, well before most organizations can respond.

Administrators must rely on latent data collected by a slow scanning process or an intermittently scheduled resource-intensive agent, which is limited to only computers that are online at that particular time. Administrators must create custom packages or scripts to deploy and install the change. Typically, they have little to no automated feedback on the progress of the deployment. Only after rescanning or waiting for the agents to report on their predetermined schedule, can administrators begin to calculate the success of the change. Often this information comes days after the change was initiated.

Even worse, administrators have no ability to manage computers that are not directly connected into the organizational network, for example, using virtual private networks (VPNs). Organizations must overcome these limitations and meet the increasingly strict internal service level agreements (SLAs) and regulatory compliance requirements. Organizations demand a systems management platform that can scale to perform in (almost) real time across the entire network. Organizations need to provide centralized reporting and delegated administration even for computers not directly connected to the organizational network at all times.

The security and compliance area is not limited to a definition of policies and systems that are trying to evaluate whether the systems meet desired standards. We must consider the following concerns:

- ▶ What is the desired solution availability?
How fast can the solution provide a remediation for recently discovered vulnerabilities? How quickly can the fix be distributed among the components of the managed environment? Do we get instant feedback about current state of the fix deployment? Can we stage the fix propagation to allow controlled distribution?
- ▶ What are the policy enforcement capabilities?
Does the solution allow for continuous evaluation of the managed systems?
Can it fix the application after the noncompliance is detected without

administrator intervention? Is it able to fix the application without connection to a central server (for example, a disconnected VPN link)?

- ▶ What is the impact on the managed infrastructure?

What additional resources are required to set up the solution? Does the solution provide any means to control network utilization? What are the minimum hardware requirements?

- ▶ What are the customization capabilities?

Is it possible to adapt solution capabilities to our needs without significant effort? Can you customize the integration with the solution reporting capabilities?

- ▶ What is the cost of the solution?

How large of an environment can be managed from a single control center? Is it possible to manage heterogeneous environments using a unified interface? Is it possible to manage physically distributed locations? What is the required size of the administrative team to manage the solution?

These questions are typically part of a long list of issues of which we must be aware when managing the organizational infrastructure. There are many topics associated with the security and compliance field. This chapter takes a step-by-step approach, from a high-level description to a more detailed view, into an endpoint management system. We focus on the security and compliance aspect of the presented solution.

2.2 IBM Tivoli Endpoint Manager

IBM Tivoli Endpoint Manager is designed to address the increasingly complex problem of keeping critical endpoint systems updated, compliant, and free of security issues. By using patented *Fixlet technology*, Tivoli Endpoint Manager can identify vulnerable and non-compliant systems throughout the organization. Then, Tivoli Endpoint Manager can apply remediation measures from a central management console.

By using the Fixlet technology, organizations can perform these functions:

- ▶ Analyze vulnerabilities
- ▶ Automatically remediate networked endpoints
- ▶ Establish and enforce configuration policies across the organization
- ▶ Distribute and update software packages
- ▶ View, modify, and audit the properties of endpoints

Tivoli Endpoint Manager allows organizations to analyze the status of configurations, vulnerabilities, and inventories, and to enforce policies automatically in near real time. Administrators can create or customize Fixlet solutions and Tasks to suit their specific requirements.

Organizations can keep their networked endpoints correctly configured, updated, and patched from the central console. They can track the progress of each endpoint system as updates or configuration policies are applied, enabling a clear view of the endpoint compliance level across the organization. In addition, Tivoli Endpoint Manager is able to examine managed endpoints by specific attributes, group those endpoints into action deployments, deploy policies, or manage endpoint assets. All results are logged for auditing while a web-based reporting application can chart overall activity.

The Tivoli Endpoint Manager platform is architected for highly diverse, distributed, and complex IT environments. It provides near real-time visibility and control:

- ▶ Single infrastructure
- ▶ Single agent
- ▶ Single console for systems lifecycle management
- ▶ Endpoint protection
- ▶ Security and compliance configuration management
- ▶ Vulnerability management

This platform enables organizations to securely manage their global IT infrastructures faster and more accurately, resulting in improved governance, control, visibility, and business agility. Plus, it gives organizations the ability to handle unforeseen challenges.

2.2.1 IBM Security Blueprint

To understand how to map the security capabilities of Tivoli Endpoint Manager to the IBM Security Blueprint, see Figure 2-1 on page 26. This diagram shows the functional components of the Governance, Risk, and Compliance solution pattern with the Network, Server, and Endpoint domain. The highlighted elements indicate those functional components that can be fulfilled, or implemented, using Tivoli Endpoint Manager. This functional highlighting also applies for the infrastructure service components.

Besides the darker highlighted elements, Figure 2-1 on page 26 also shows medium-highlighted elements. Although Tivoli Endpoint Manager can address these components to a lesser degree, the area of coverage is not a core function of the product and is limited.

The desired functionality of a solution can be determined by using the Governance, Risk, and Compliance solution pattern with the Network, Server, and Endpoint domain. Use the mapping in Figure 2-1 on page 26 as a quick reference of the functional security management aspects of Tivoli Endpoint Manager. This reference can help determine which functions of a solution are adequately covered by using Tivoli Endpoint Manager.

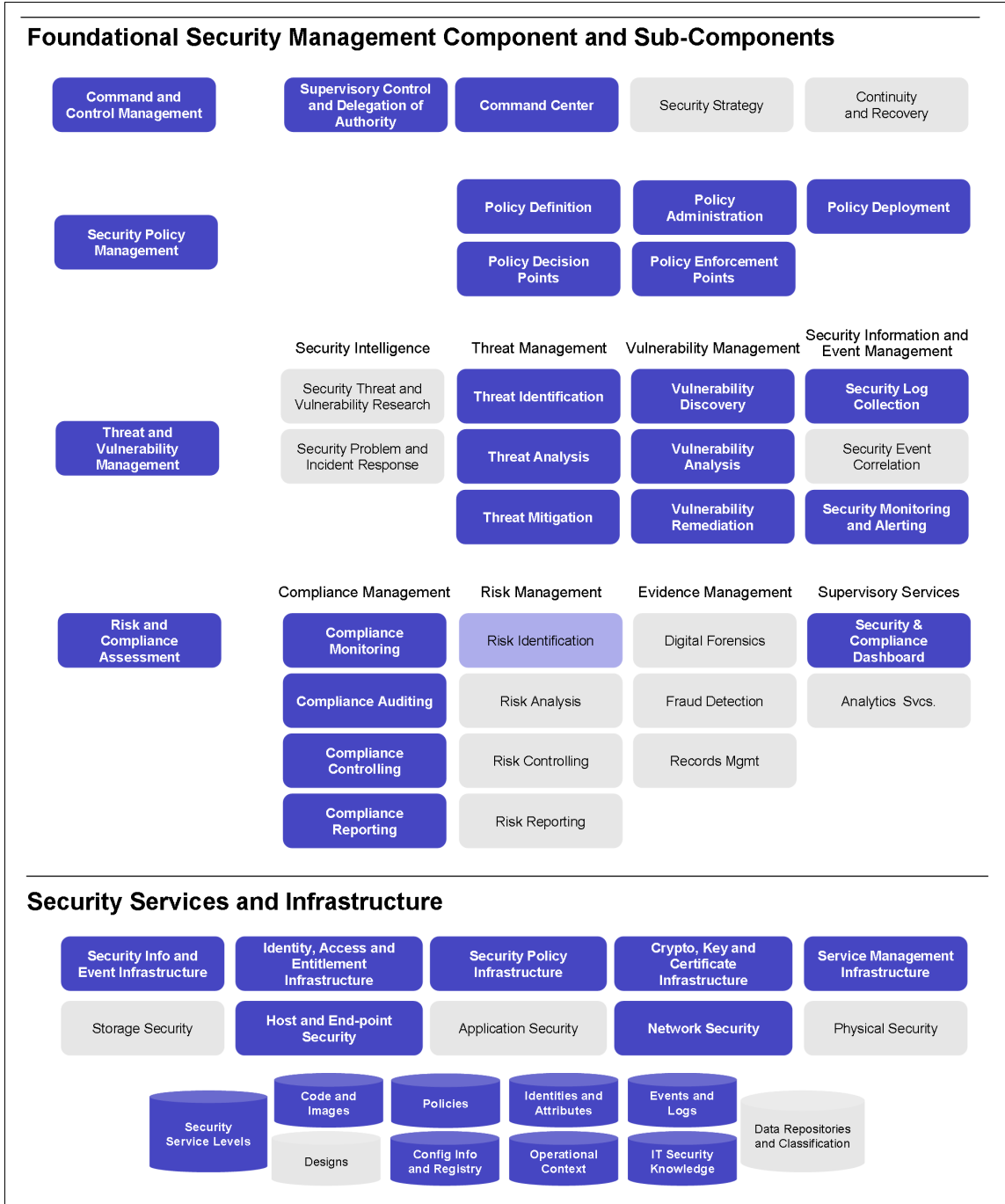


Figure 2-1 Mapping of Tivoli Endpoint Manager to the IBM Security Blueprint

Tivoli Endpoint Manager is a multilayered technology platform that acts as the central nervous system of your global IT endpoint infrastructure. As a dynamic, content-driven messaging and management system, the technology distributes the work of managing IT infrastructures out to the managed devices. The Tivoli Endpoint Manager platform is able to operate in near real time, delivering the scalability and performance that large organizations demand.

Acquisition information: As of February 2011, BigFix is officially integrated into IBM. All BigFix products are integrated into the Tivoli Software portfolio and rebranded as IBM Tivoli Endpoint Manager.

IBM Tivoli Endpoint Manager is derived from the *BigFix* Unified Management Platform (<http://www.bigfix.com>). Because IBM recently acquired this technology, this book sometimes refers to the platform as both Tivoli Endpoint Manager and BigFix, for example, in URLs to additional resources.

2.2.2 Platform

The Tivoli Endpoint Manager solution consists of five packages as part of the overall offering:

- ▶ Tivoli Endpoint Manager for Lifecycle Management
- ▶ Tivoli Endpoint Manager for Security and Compliance
- ▶ Tivoli Endpoint Manager for Patch Management
- ▶ Tivoli Endpoint Manager for Power Management
- ▶ Tivoli Endpoint Manager for Core Protection

These packages are depicted in Figure 2-2.



Figure 2-2 Tivoli Endpoint Manager packages

We describe these packages in detail in the following sections.

Tivoli Endpoint Manager for Security and Compliance

Tivoli Endpoint Manager for Security and Compliance addresses the security challenges of distributed environments. Endpoint management and security are included in a single solution that supports security among distributed endpoints. This solution reduces costs and the complexity of management while increasing business agility, speed to remediation, and accuracy. This offering provides these functions:

- ▶ Automates time-consuming device configuration and change management tasks.
- ▶ Effectively manages the compliance lifecycle with an ongoing, closed-loop process.
- ▶ Gains greater visibility into network resources in dynamic and complex environments.
- ▶ Provides accurate, precise, and up-to-the minute visibility into and continuous enforcement of security configurations and patches.
- ▶ Centralizes the management of functions that provide advanced antivirus and firewall protection.
- ▶ Employs a unified management infrastructure to coordinate among IT, security, desktop, and server operations.

Tivoli Endpoint Manager for Security and Compliance reaches endpoints regardless of location, connection type, or status. Tivoli Endpoint Manager for Security and Compliance provides comprehensive management for all major operating systems, third-party applications, and policy-based patches.

Tivoli Endpoint Manager for Lifecycle Management

Tivoli Endpoint Manager for Lifecycle Management addresses the management challenges of distributed environments with real-time visibility and advanced functionality. Tivoli Endpoint Manager for Lifecycle Management helps users see and update systems and remediate issues with continuous management. With Tivoli Endpoint Manager for Lifecycle Management, IT staff can maintain service levels and focus on critical issues, and perform these functions:

- ▶ Manages hundreds of thousands of endpoints regardless of location, connection type, or status.
- ▶ Manages heterogeneous platforms, such as Microsoft Windows, UNIX, Linux, and Mac OS operating systems, running on physical or virtual machines.
- ▶ Employs an agent-based approach that delivers up-to-date visibility and automatically remediates issues.

- ▶ Reduces management complexity and cost, increases accuracy, and boosts productivity.
- ▶ Simplifies and streamlines help desk calls and problem resolution with remote desktop control.
- ▶ Shortens update cycles, improves the success rates for provisioning, and reduces IT labor requirements.

Tivoli Endpoint Manager for Patch Management

Tivoli Endpoint Manager for Patch Management provides unified, real-time visibility and enforcement to deploy and manage patches to all endpoints. Tivoli Endpoint Manager for Patch Management uses a single console that supports comprehensive patch management capabilities among distributed endpoints. Tivoli Endpoint Manager for Patch Management reduces business risk, costs, complexity, and time while enhancing security. This offering provides these functions:

- ▶ Automatically manages patches for multiple operating systems and applications across hundreds of thousands of endpoints regardless of location, connection type, or status.
- ▶ Reduces security risks by cutting remediation cycles from weeks to days or hours.
- ▶ Provides greater visibility into patch compliance with flexible, real-time monitoring, and reporting.
- ▶ Provides up-to-date visibility and control from a single management console.
- ▶ Efficiently deploys patches, even over low-bandwidth or globally distributed networks.
- ▶ Automatically remediates problems related to previously applied patches.
- ▶ Patches endpoints on or off the network, including roaming devices using Internet connections.

Tivoli Endpoint Manager for Power Management

Tivoli Endpoint Manager for Power Management addresses the power management challenges of distributed environments. Tivoli Endpoint Manager for Power Management is a policy-driven power management solution that supports comprehensive power control among distributed endpoints. Tivoli Endpoint Manager for Power Management reduces energy usage and costs while avoiding disruptions in systems management.

This offering provides the following functions:

- ▶ Controls energy costs with a centralized, scalable, policy-driven power management system for all endpoints that run Microsoft Windows and Mac operating systems.
- ▶ Manages power settings for hundreds of thousands of endpoints regardless of location, connection type, or status, from a single console.
- ▶ Empowers users with an opt-in approach that allows them to select the appropriate power profile.
- ▶ Applies easily integrated capabilities to deal with common power management issues, such as placing a continuously running personal computer into hibernation during off-hours to save energy.
- ▶ Creates *what if* energy usage scenarios to encourage conservation initiatives.
- ▶ Displays energy consumption as measures of power or carbon dioxide to help encourage organizational efforts to *go green*.
- ▶ Provides real-time visibility into current power usage and costs.

IBM Tivoli Endpoint Manager for Core Protection

IBM Tivoli Endpoint Manager for Core Protection manages both technology and business risk. It protects physical and virtual endpoints from damage caused by malware and other vulnerabilities. Tivoli Endpoint Manager for Core Protection reduces the business disruptions that can result from attacks on endpoints. This offering provides these functions:

- ▶ Protects physical and virtual endpoints from damage caused by viruses, Trojan horses, worms, spyware, rootkits, web threats, and their new variants.
- ▶ Cross-references threat information with a large, cloud-based database created by Trend Micro and continuously updated through the Trend Micro Smart Protection Network.
- ▶ Delivers increased protection through policy enforcement to ensure that antivirus services are always installed, running and up-to-date.
- ▶ Automatically cleans endpoints of malware, including processes and registry entries that are hidden or locked.
- ▶ Protects return on virtual desktop infrastructure (VDI) investments with its virtualization awareness by preventing resource contention to enable higher density of guests to hosts in virtualized environments.
- ▶ Provides security for both fixed, network-connected endpoints and roaming, Internet-connected endpoints.

- ▶ Centralizes the management of functions that provide advanced antivirus and firewall protection.
- ▶ Employs a unified management infrastructure to coordinate among IT, security, desktop, and server operations.
- ▶ Supports the following operating systems: Windows family and Mac OS X.

2.2.3 High-level concept

The Tivoli Endpoint Manager platform is built around the following three key concepts. These concepts work in concert to enable real-time visibility from a single, central point of control:

- ▶ Single management Agent
- ▶ Single management Console and Server
- ▶ Single policy-based model

Single management Agent

The lightweight, intelligent Agent can be deployed on every desktop, mobile computer, mobile device, and server that you want to manage. It has the following characteristics:

- ▶ Multipurpose client that offers the ability to consolidate and replace existing point-product software.
- ▶ Only 2 - 4 MB of required endpoint system memory.
- ▶ Real-time and continuous policy processing, remediation, validation, and reporting.
- ▶ Policies remain enforced even when remote devices roam from the organizational network.
- ▶ Support for dynamic queries and management actions.
- ▶ Policy-based and dynamic bandwidth throttling to work over Very Small Aperture Terminal (VSAT), Multi-Protocol Label Switching (MPLS), and other bandwidth-constrained networks.
- ▶ Broad platform support, including virtualized operating systems, such as VMware ESX Server and Microsoft Hyper-V. For details, see “Tivoli Endpoint Manager Agent” on page 35.

Single management Console and Server

The Console and Server work together to orchestrate a high level of visibility and control and have the following characteristics:

- ▶ A basic Server model can manage up to 250,000 devices.

- ▶ Built-in reporting and analysis tools.
- ▶ Support for automatic multiserver synchronization and nonstop services, even during a disruptive event.
- ▶ Integrated security infrastructure controls Agent actions and ensures administrator accountability.
- ▶ Capability to set configuration standards and baselines from defined groups of managed agents.
- ▶ Standard SQL and SOAP interfaces for integration with other database applications and systems.

Single policy-based model

The policy language, referred to as the *Fixlet Relevance language* (see “Relevance” on page 40), is a published command language. With it, organizations, business partners, and developers can create custom policies and services for managed assets.

As a single method for interrogating and managing endpoints independently of platform or domain, the policy language can be used to solve common problems experienced by most organizations. Examples include deployment of patches, configuration management, antivirus management, or dynamic queries and remediation to manage unforeseen and unstructured problems. Without Tivoli Endpoint Manager, these problems either cannot be solved or must be solved manually. Solving them manually can take a long time and require excessive amounts of resources. The policy language has these characteristics:

- ▶ Cloud-based service delivery of policy content for on-demand functionality
- ▶ New solutions provisioned without additional hardware, infrastructure, or network changes
- ▶ Open architecture for easy policy customization and development

2.2.4 Managed environment

Figure 2-3 on page 33 depicts a sample environment that can potentially be managed with Tivoli Endpoint Manager. In this section, we introduce the platform elements and describe how to use them for endpoint management. For a more technical description of the components, see 3.1, “Logical component overview” on page 64.

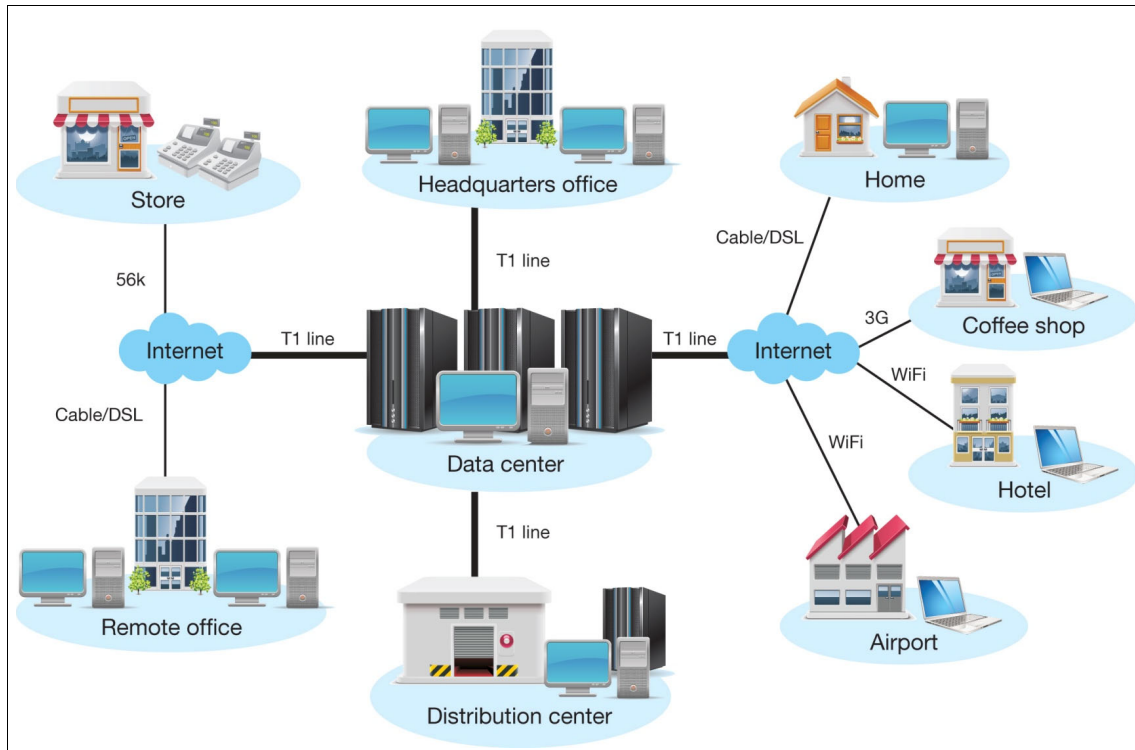


Figure 2-3 Sample environment managed with Tivoli Endpoint Manager

The Tivoli Endpoint Manager platform contains these major components:

- ▶ Tivoli Endpoint Manager Server
- ▶ Tivoli Endpoint Manager Console
- ▶ Tivoli Endpoint Manager Agent
- ▶ Tivoli Endpoint Manager Relay

These four components are initially described in the following sections. Chapter 3, “IBM Tivoli Endpoint Manager component structure” on page 63 provides more detailed technical information.

Do not get confused: The terms *Agent* and *Client* are often used interchangeably when discussing the platform. In this IBM Redbooks publication, we refer to the platform as the Agent.

Tivoli Endpoint Manager Server

The Tivoli Endpoint Manager Server manages policy content, delivered in messages called Fixlets (see “Fixlets and Tasks” on page 37) and updated

continuously through the Tivoli Endpoint Manager Content Delivery cloud-based service. The Server enables the operator to maintain real-time visibility and control over all devices in the environment, including instantaneous discovery of devices not yet managed. The Agent performs most of the analysis, processing, and enforcement instead of the Server. Therefore, one Tivoli Endpoint Manager Server can support up to 250,000 endpoints, enabling organizations to benefit from their security and systems management investment. In most deployments, there is usually only one Server. But, Tivoli Endpoint Manager can also work in high-availability scenarios that combine multiple servers for safety and service continuity.

Tivoli Endpoint Manager Console

The Tivoli Endpoint Manager Console (Figure 2-4) integrates all components to provide a system-wide view of your networked computers, along with their vulnerabilities and suggested remedies. As an authorized user, with the Tivoli Endpoint Manager Console, you can distribute a fix to those computers that need it, with no impact on the rest of the network.

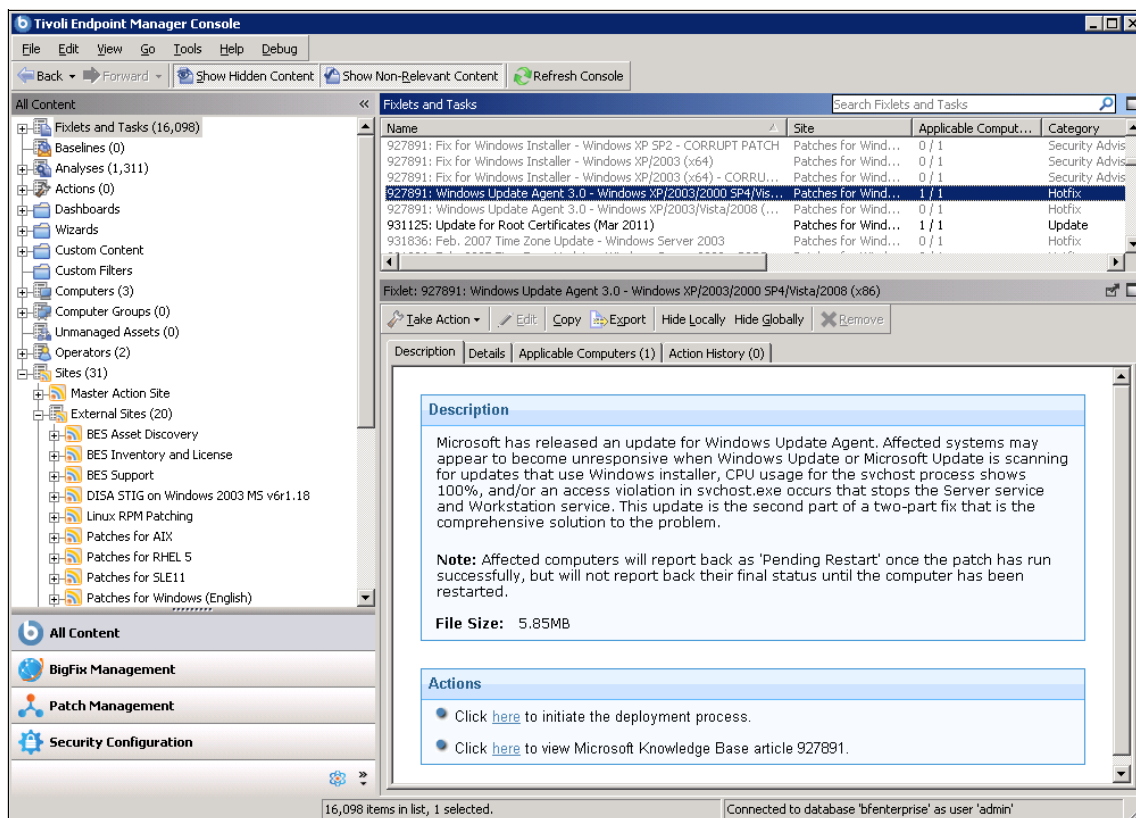


Figure 2-4 Tivoli Endpoint Manager Console

The Tivoli Endpoint Manager Console is an application that can be run on any Windows computer with network access to the Tivoli Endpoint Manager Server. Because the Tivoli Endpoint Manager Console is the management center of operations, we provide more detailed information about it in 3.1.4, “Console” on page 71.

Tivoli Endpoint Manager Agent

The Tivoli Endpoint Manager Agent must be deployed on every device that you want to manage by using the Tivoli Endpoint Manager platform.

The Agent accesses a collection of Fixlet messages (see “Fixlets and Tasks” on page 37) that seek out security holes, vulnerabilities, and deviations from the desired operating environment. If a vulnerability is found, the Agent can then implement corrective actions received from the Console. In most cases, the Agent operates silently, without any direct intervention from the user. However, if you need to solicit user response, you can provide prompts with the solution.

The Agent is capable of assessing the state of the endpoint against the policy. It can bring the endpoint back into compliance with the policy, without any instruction from the management Server. This capability is true of the security and systems management applications of the entire platform, from security configuration management to software distribution to power management. Only a single Agent is necessary.

The Tivoli Endpoint Manager Agent is designed to work continuously in an endless loop, as shown in Figure 2-5. For more information, see 3.1.6, “Agent” on page 77. An extended description of this loop, using a patch management example, is presented in 2.3.2, “Patch management” on page 47.

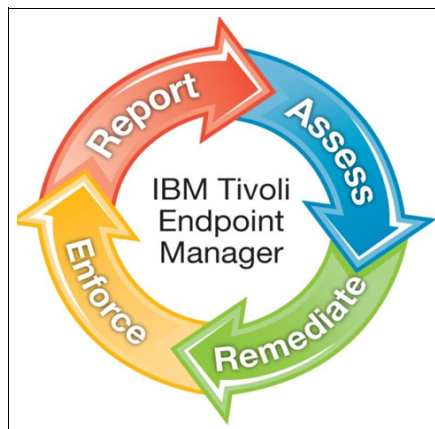


Figure 2-5 Tivoli Endpoint Manager Agent evaluation loop

Broad support: The Tivoli Endpoint Manager Agent can be deployed on a wide range of endpoints, including Mac OS, Windows, Windows Mobile, VMware ESX Server, Linux, and UNIX operating systems. For a more detailed list of supported operating systems, go to this website:

<http://support.bigfix.com/bes/misc/supportpolicy.html>

Tivoli Endpoint Manager Relay

Relays are optional network components that can improve the performance of your platform deployment. A Relay simultaneously mitigates two bottlenecks:

- ▶ It can relieve the load on the servers.

The Server has many duties, including distributing patches and other files. A Relay can be set up to ease this burden so that the Server does not need to distribute the same files to every Agent. Instead, the file is sent one time to the Relay, which in turn distributes it to the Agents. The impact on the Server is reduced, on average, by the ratio of Relays to Agents.

- ▶ It can reduce congestion on low-bandwidth connections.

If you have a Server that communicates with a dozen computers in a remote office over a slow VPN, designate one of those computers as a Relay. Then, instead of sending patches over the VPN to every Agent independently, the Server only sends a single copy to the Relay. That Relay, in turn, distributes the file to the other computers in the remote office over the LAN. This effectively removes the VPN bottleneck for remote groups on your network.

A key benefit of deploying Relays is that they can be deployed on shared hardware, such as file, print, or domain servers, or other computers, such as kiosks, that are operational all the time. This way, organizations can scale with minimal hardware requirements. Any Microsoft Windows, Solaris 10, Red Hat Enterprise Linux 4 or 5, or IBM AIX® 5.3 or 6.1 computer with an installed Agent can be designated dynamically to be a Relay.¹

2.2.5 Key terms

In this section, we describe the key terms that appear throughout this book. These terms are elements of the Tivoli Endpoint Manager solution. Your familiarity with them is important while we introduce the platform.

¹ For more information about operating systems support for Relays, go to this website:
<http://support.bigfix.com/bes/misc/supportpolicy.html>

Fixlets and Tasks

Because Fixlets and Tasks perform similar functions, they are grouped together into a single menu item in the console:

- ▶ Fixlets

A Fixlet is a piece of code within the Tivoli Endpoint Manager solution that first identifies a *problem situation*, such as a missing operating system patch.

The Fixlet instructs the Agent on the endpoint to query the Server for this missing patch and to download it to the endpoint. Fixlets are grouped into *sites* that denote a collection of Fixlets that apply to a certain category of issues to be managed, such as patch management. At this point, we begin to see messages in the console that alert the operator to the current process and show progress toward completion. These messages are called *Fixlet messages*.

- ▶ Fixlet messages

Fixlet messages are the core of the Tivoli Endpoint Manager functionality. Using *Relevance statements*, they can target specific computers, remediating only those client computers with issues and never affecting the computers that do not have issues. Fixlets come with an *action script* that can resolve the issue with a single mouse click. Typically, when the action completes, the Fixlet detects that the issue is no longer applicable to that computer.

While Fixlet actions propagate through your network, you can track their progress with the console, Web Reports, and the Visualization Tool. After every computer in your network is remediated, the Fixlet message disappears from the list. If the issue reappears, the Fixlet again shows up in the list, ready to address the issue again.

Fixlet messages contain a text description of the issue and can offer several actions, including links to more information. Often, a Fixlet message has a default action, allowing you to simply click from the Fixlet list to deploy it.

Fixlet messages can be grouped into *Baselines*, allowing even higher levels of automation. If you create a Baseline of Fixlet messages that contain default actions, you can turn the tedious chore of maintaining a common operating environment into a single-click operation.

At any time, you can open a Fixlet message to inspect the Relevance Expressions that target the Agents and the action script that remediates the issue. This inspection capability provides a high degree of confidence in the applicability and efficacy of the remedial action. You can also see which computers on your network are affected by each Fixlet message. You can view a history of the actions taken on an Agent-by-Agent basis.

► **Tasks**

Tasks are similar to Fixlet messages, but are designed for ongoing tasks, and as a consequence, they are more persistent. Tasks come with one or more action scripts to help you to adjust settings or run maintenance tasks.

Tasks contain a description of the issue and might have a default action, allowing you to click from the Task list to deploy it. Tasks and Fixlet messages can be grouped into Baselines, allowing even higher levels of automation.

At any time, you can open a Task to inspect the Relevance Expressions that qualify the Agents and the action script that addresses the Task.

Baselines

Baselines are collections of Fixlet messages and Tasks. They provide a powerful way to deploy a group of actions across an entire network with a single command.

Baselines provide a way to maintain a common operating environment. They ensure that all users in any specific domain have the same software, patches, and drivers. Baselines are easy to set up by selecting the Fixlet messages, Tasks, and other Baselines that you want to be a part of the group. To limit the scope of a Baseline, use a Relevance Expression to target any subset of your network. Use IP addresses, computer names, operating systems, and other qualifiers.

For example, create a Baseline named “All critical hotfixes”, and populate it with all the current critical hotfixes available in the Fixlet list. Or, you might create a Baseline named “Finance department Baseline”. This Baseline is designed to keep that particular group of computers updated with the latest financial programs, financial tables, updates, and patches.

Analysis

Analysis allows an operator to view and summarize various properties of client computers across a network. There are several predesigned analyses supplied by IBM that examine various aspects of your networked computers. These analyses examine hardware, applications, and Server, Relay, and Agent relationships.

Studying these predesigned analyses can be instructive when you want to create analyses or customize existing analyses. Custom analysis can help you monitor aspects of your network that are interesting or vital to the operation of your organization.

The Retrieved Properties that form the foundation of each analysis are created with Relevance Expressions. For example, to ensure that you fully deploy the

most recent Agent software, you might use an expression, such as “version of Agent”. This simple expression is evaluated on every computer where the analysis is targeted. You can see explicitly which version of the Agent is running on each computer. Or, you can view a summary of how many machines are running each version.

Analyses are targeted with another Relevance statement. Generally, you want to narrow the scope with a Relevance statement. If you specify the name of the operating system as a lowercase term that starts with “win”, you limit the analysis to Windows computers only.

Actions

Actions represent the core functionality of the system. Fixlet messages, Tasks, and Baselines depend on Actions to execute their remediation mission.

Actions are typically scripts that can customize a specific solution for each Agent, using the power of Relevance Expressions. Although the Relevance language itself cannot alter an Agent, it can be used to direct Actions in a way that parallels the original trigger. For example, a Fixlet might use the Relevance language to inspect a file in the system folder. By using a similar Relevance clause, the action can then target that same file without knowing explicitly where that folder is. The Action author (and issuer) can concentrate on the issue without knowing where the folder is.

You can inspect an Action script before you execute it by looking at the Details tab of Fixlet messages and Tasks. You can also write your own custom Action scripts.

Computer groups

With the Tivoli Endpoint Manager Console, you can group your computers so that you can target them appropriately. You might want to group your development computers, for example, to ensure that you do not interfere with certain earlier software projects. There are several ways to group computers, but the two most common techniques are *manual grouping* and *automatic grouping*.

Manual groups are static. Automatic groups can change dynamically, depending on the current values of the inclusion properties.

Fixlet Sites

Upon installation, the platform automatically subscribes itself to the Fixlet Sites that you select during installation. Each Fixlet Site contains a collection of Fixlet messages that perform certain tasks. A sample list of Fixlet sites is presented in Figure 2-6 on page 40.

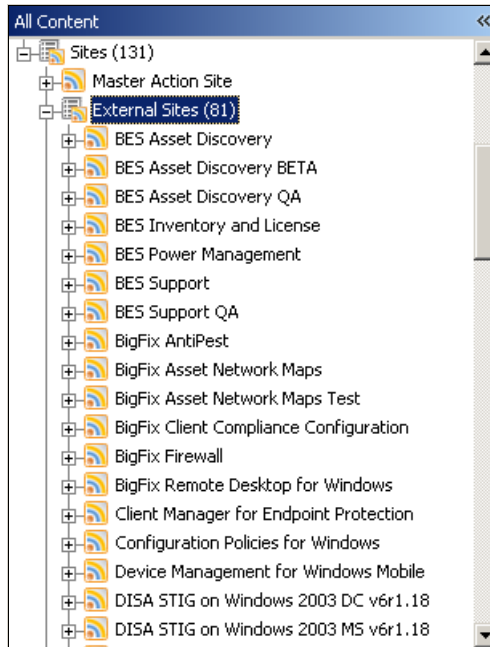


Figure 2-6 Fixlet Sites visible in the Console

Relevance

The Relevance language introduced in “Single policy-based model” on page 32 was developed to quickly inspect various aspects of a computer. This human-readable language is at the core of Tivoli Endpoint Manager. It allows Fixlet authors to target Actions at only those computers that need the fix, and no others. Thus, you can be confident that only broken machines are being fixed and that the rest are never bothered.

The Relevance language can query an exhaustive set of computer properties quickly. Most Console operators rely on third parties to write Fixlet messages. Their exposure to the Relevance language is not critical to operating the console. However, you can customize the Console with short lines of code from the Relevance language (called Relevance Expressions). These lines of code can provide an unprecedented amount of control over the client computers in the network.

You can use Relevance Expressions to create retrieved properties, which you can then use to organize and filter the Agents in the network. You can experiment and debug your Relevance Expressions using the Relevance Debugger, which is automatically installed with the Console. The program can also format your expression for easier reading. There are literally thousands of useful Retrieved Properties, and there are far too many to list here.

Relevance language: For tutorials, examples, and additional documentation about the Relevance language, go to this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Fixlet%20Authoring>

Dashboards and wizards

Dashboards are an internal part of the Tivoli Endpoint Manager Console. The Dashboard aggregates information to present it to the user in a usable way. In contrast to Fixlets and Analysis, where raw data and the compliance state are presented, a Dashboard can initially evaluate and aggregate collected information.

Dashboards are provided as a part of the Fixlet Site; thus, their functionality is limited to that particular area typically. For example, a Dashboard that is part of a patch site might present status for computers and applied patches or percentage compliance data regarding patch severity. A sample patch management Dashboard is depicted in Figure 2-7 on page 42.

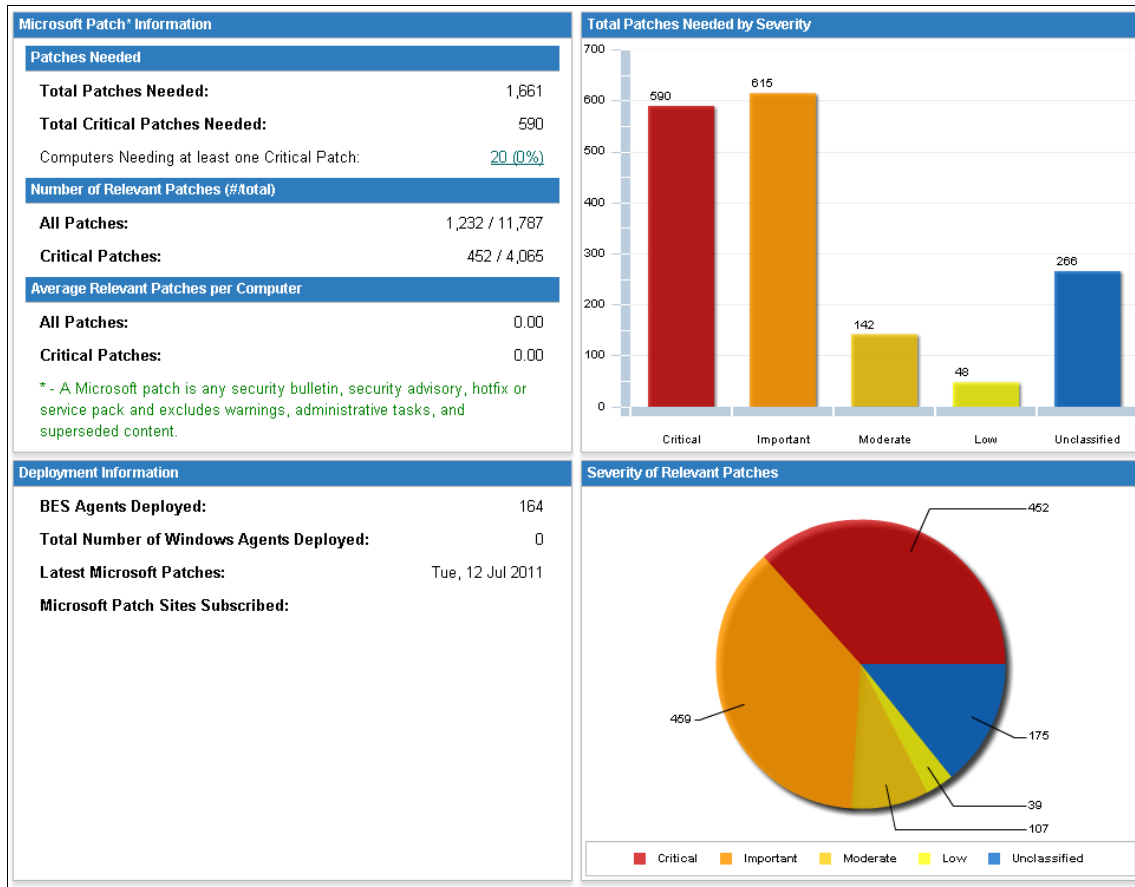


Figure 2-7 Tivoli Endpoint Manager Console: Patch management overview dashboard

Wizards are similar to Dashboards, but the major difference is that wizards facilitate administrative actions. A Dashboard is a report-oriented tool. A Wizard can help the user perform actions, such as the automated creation and deployment of Fixlets and Tasks or environment maintenance activities.

Web Reports

The Web Reports program can monitor, print, or archive the status of the local database. It also can read the databases of other servers and aggregate the data to offer a top-level view of the overall organization with multiple database servers. Aggregation servers allow you to view information from multiple networks that contain hundreds of thousands of computers.

With the Web Reports program, you can see an overview of your relevant Fixlet messages and your remediation efforts. You can see charts that summarize the

number of administered computers in your network and the vulnerability status. In addition, you can see overall statistics and a list of the most common issues that are detected. You can click these common relevant Fixlet messages to see them in greater detail. In addition, you can see at a glance how remediation and policy enforcement efforts progress.

Next, we look at the IBM Tivoli Endpoint Manager for Security and Compliance.

2.3 Tivoli Endpoint Manager for Security and Compliance

Tivoli Endpoint Manager for Security and Compliance addresses security challenges that are associated with complex, distributed corporate endpoint environments. It helps ensure continuous protection and compliance by providing endpoint security and compliance management in a single integrated solution. This solution can close gaps in security exposures. This solution can bridge gaps that exist between business functions, such as functions that establish and execute strategy and policy, manage devices in real time, and generate security compliance reports.

Tivoli Endpoint Manager for Security and Compliance offers the following capabilities:

- ▶ Accurate up-to-the minute visibility into and continuous enforcement of security configurations and patches.
- ▶ Centralized management for third-party anti-malware and firewall protection.
- ▶ Support for preferred practices that meet the US Federal Desktop Configuration Control (FDCC) regulations and Defense Information Systems Agency Security Technical Implementation Guides (DISA STIGs).
- ▶ National Institute of Standards and Technology (NIST) certified for both assessment and remediation using the Security Content Automation Protocol (SCAP). Tivoli Endpoint Manager for Security and Compliance is the first product to be certified for these specific functions.
- ▶ Securely transmitted endpoint instructions as demonstrated through NIAP CCEVS EAL3 and FIPS 104-2, Level 2 certifications.
- ▶ Support for the Open Vulnerability and Assessment Language (OVAL) standard to promote open and publicly available security content.
- ▶ Ability to receive and act on vulnerability and security risk alerts published by the SANS Institute.

- ▶ Trending and analysis of security configuration changes available through advanced reporting.

All products in the Tivoli Endpoint Manager family offer the following capabilities:

- ▶ Discover endpoints that organizations might not be aware were in their environment.
- ▶ Provide a single console for management, configuration, discovery, and security functions capable of simplifying operations.
- ▶ Use virtually any hardware or software property to target specific actions to an exact type of endpoint configuration or user type.
- ▶ Employ a unified management infrastructure to coordinate among IT, security, desktop, and server operations.
- ▶ Reach endpoints regardless of location, connection type, or status with comprehensive management for all major operating systems, third-party applications, and policy-based patches.

Tivoli Endpoint Manager for Security and Compliance enables automated, targeted processes that provide control, visibility, and speed to effect change and report on compliance. Remediation cycles are short and fast with malware and virus issues addressed with rapid patch management capabilities.

2.3.1 Security functions

Tivoli Endpoint Manager for Security and Compliance offers a wide array of functions. It offers the ability to add or remove targeted functions as and when needed without adding additional infrastructure and incurring additional implementation costs. Tivoli Endpoint Manager for Security and Compliance offers the following core functions:

- ▶ Patch management
- ▶ Security configuration management
- ▶ Security Compliance Analytics
- ▶ Vulnerability management
- ▶ Asset discovery
- ▶ Multivendor endpoint protection management
- ▶ Network self-quarantine

In the following sections, we briefly describe the security functions for Tivoli Endpoint Manager for Security and Compliance. Later, we put more focus on three of those core components:

- ▶ “Patch management” on page 47
- ▶ “Security configuration management” on page 52
- ▶ “Security Compliance Analytics” on page 56

Patch management

The patch management capability includes a comprehensive list of capabilities for delivering patches to many operating platforms and software vendors:

- ▶ Microsoft Windows
- ▶ UNIX
- ▶ Linux
- ▶ Mac OS
- ▶ Adobe
- ▶ Mozilla
- ▶ Apple

This delivery capability allows Tivoli Endpoint Manager for Security and Compliance to effectively deliver patches to distributed endpoints, regardless of their location, connection type, or status. A single management server can support up to 250,000 endpoints. Patch deployment times are shortened over low-bandwidth and globally distributed networks without sacrificing endpoint functionality.

Real-time reporting provides information about which patches are deployed, when they are deployed, and who deployed them. Automatic confirmation that patches are applied for a complete closed-loop solution to the patching process is also provided.

Security configuration management

The Tivoli Endpoint Manager for Security and Compliance security configuration features provide a comprehensive library of technical controls. These controls assist in achieving security compliance by detecting and enforcing security configurations. These technical controls are validated through the National Institute of Standards and Technology (NIST).

Security Compliance Analytics

Tivoli Endpoint Manager for Security and Compliance Analytics (SCA) is a web-based application for security and risk assessment. The system archives security compliance check results to identify configuration issues and report levels of compliance toward security configuration goals.

Vulnerability management

The vulnerability management feature enables organizations to discover, assess, and remediate vulnerabilities before endpoints are affected. This feature assesses systems against standardized Open Vulnerability and Assessment Language (OVAL²) vulnerability definitions and reports non-compliant policies in

² To find more information about OVAL, go to this website: <http://oval.mitre.org/index.html>

real time. The result is enhanced visibility and full integration at every step in the entire *discover-assess-remediate-report* workflow.

By using automated or manual actions, security administrators are able to identify and eliminate known vulnerabilities on endpoints across the entire organization. By using a single tool that discovers and remediates vulnerabilities, security administrators can increase speed and accuracy and shorten remediation cycles for patch deployment, software updates, and vulnerability fixes. Security administrators can also extend security management to mobile Agents on or off the network. They can set alarms to identify rogue assets and take steps to locate them for remediation or removal.

Asset discovery

With Tivoli Endpoint Manager for Security and Compliance, asset discovery is no longer a *snapshot exercise*. It creates dynamic situational awareness about changing conditions in the infrastructure. The ability to scan the entire network frequently delivers pervasive visibility and control to help ensure that organizations quickly identify all IP-addressable devices. These devices include virtual machines, network devices, and peripherals, such as printers, scanners, routers, and switches, in addition to computer endpoints with minimal network impact. This function helps maintain visibility into all organization endpoints, including mobile and notebook computers that roam beyond the organization network.

Multivendor endpoint protection management

This feature gives administrators a single point of control for managing third-party endpoint security clients from the following vendors:

- ▶ Computer Associates
- ▶ McAfee
- ▶ Sophos
- ▶ Symantec
- ▶ Trend Micro

With this centralized management capability, organizations can enhance the scalability, speed, and reliability of protection solutions. The feature monitors system health to ensure that endpoint security clients are always running and that virus signatures are updated. In addition to providing a unified view of disparate technologies, it facilitates migrating endpoints from one solution to another with “one-click” software removal and reinstallation. Closed-loop verification ensures that updates and other changes complete, including Internet-enabled verification for endpoints disconnected from the network.

Network self-quarantine

Tivoli Endpoint Manager for Security and Compliance automatically assesses endpoints against required compliance configurations. If an endpoint is out of compliance, the solution can configure the endpoint so that it is placed in network quarantine until it complies. The Tivoli Endpoint Manager Server has management access to the endpoint, but all other access is disabled.

Now, we look into those capabilities, starting with patch management.

2.3.2 Patch management

Patch management is challenging because of the massive complexity involved. Despite the risks, organizations are reluctant to patch because of the required time and labor, plus the potential of disrupting business operations. In an organization with a heterogeneous hardware and software environment, managing the multitude of patches and issuing them in a timely manner can overextend IT staff and budgets.

What is needed is a rapidly deployable, cost-effective, and policy-based patch management solution. Tivoli Endpoint Manager for Patch Management can provide the solution:

- ▶ Works for all endpoints in organizations of all sizes, including the largest.
- ▶ Supports multiple vendors, operating systems, applications, and platforms³.
- ▶ Works over low-speed connections and supports devices that roam outside of the organizational network.
- ▶ Minimizes the demand on IT staff.
- ▶ Operates in near real time, deploying patches organization-wide in hours.
- ▶ Presents the endpoint status continuously during the patching process.

A preferred practice approach of patching involves a closed-loop process with six steps: *research*, *assess*, *remediate*, *confirm*, *enforce*, and *report* (as shown in Figure 2-8 on page 48). Historically, many of these steps were implemented through separate, non-integrated technologies, making it virtually impossible to create a closed-loop, real-time patch management process. Tivoli Endpoint Manager provides all of these steps as part of a unified, fully integrated process. This process helps to enhance security and save money, time, and resources.

³ For more information about patch Fixlets for supported products, go to this website:
<http://support.bigfix.com/resources.html#Patch>

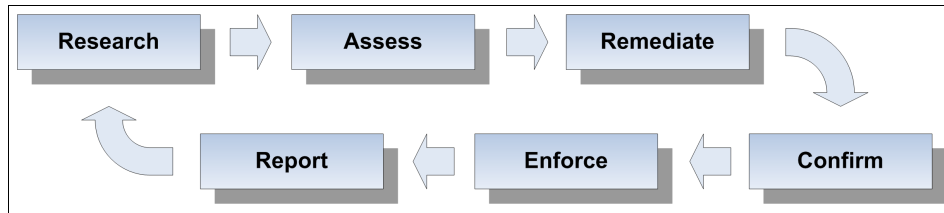


Figure 2-8 Tivoli Endpoint Manager Agent processing flow

Research

The first step in the patch management process involves discovering which patches are available. This discovery includes researching patch availability through vendor email messages, application pop-up notifications, websites, blogs, and various other sources. This process must be repeated for hundreds of patches, across various operating system or application vendors.

One alternative is relying on default vendor auto-updates. This approach can lead to mistakes that can have negative consequences. Automating the acceptance of patches without testing them can put organizations at a huge risk. There is no control over timing or reporting, and relying on users to apply updates is risky and unreliable.

A better approach is for the patch management vendor to provide a consolidated stream of the most common patches. The organization needs to evaluate each load of patches only as they arrive and test them for compatibility with the organizational environment. This approach allows for controlled deployment of only selected patch packages. However, if this approach is not automated, it requires significant time and resources that organizations might not have.

IBM acquires, tests, packages, and distributes patches from operating system and common third-party application vendors directly to Tivoli Endpoint Manager Servers. When a supported vendor releases a new patch, IBM receives the patch, conducts preliminary analysis and creates patch Fixlets. The patch Fixlets wrap the update with policy information, such as patch dependencies, applicable systems, and severity level. Published Fixlets are then automatically downloaded by Tivoli Endpoint Manager Servers. By using this solution for patching, you can use the IBM provided Fixlets and organization-specific patches through a wizard-driven interface. This process works for virtually any update, including internal application patches.

Assess

For each identified patch, the IT organization must determine the applicability and criticality of the update, identifying which endpoints need patching across the organization. For security updates, this critical data translates directly into risk,

because business risk increases with the number of unpatched endpoints. There are tools that can help acquire this data, but many tools require days or weeks to collect and collate this information by scanning every endpoint in the network. Because many roaming endpoints are rarely connected to the network, this process can potentially take days to complete. We want all this information to be immediately available to system administrators at the time of patch release, because many patches are time critical. The process of risk assessment and patch prioritization must take place as quickly as possible.

Tivoli Endpoint Manager uses a single intelligent software Agent installed on all managed endpoints to continuously monitor and report the endpoint state to a Server. Patch Fixlets, which were created and applied in the previous step, are delivered to the Tivoli Endpoint Manager Agent. The Agent compares endpoint compliance against defined patch existence conditions. This information is especially critical during emergency patch scenarios when a vendor releases a highly critical, out-of-band patch. Organizations must rapidly quantify the overall magnitude and risk from the related exploits. The assessment process is executed in the infinitive loop to assure continuous monitoring.

Remediate

A patch is assessed and a determination is made to distribute it across the organization. Then, it must be packaged and tested to ensure that it does not conflict with other patches and third-party software installed on the target endpoints. Patch prerequisites and dependencies, such as minimum service pack levels, must also be determined. Usually, you apply and test the update on a select number of endpoints before a general release. A process that can take days or weeks to complete using manual tools.

After testing indicates that the patch is probably safe for organization-wide deployment, it is applied to affected endpoints. It is applied typically in batches, further extending the patch window. Long remediation times are primarily due to the inability to rely on patch quality. Secondly, long remediation times are due to unreliable distribution mechanisms. Both reasons result in low first-pass patch rates. Most organizations are therefore forced to proceed slowly in case a patch causes an unforeseen problem. Organizations must also ensure that network links are not overwhelmed by the patch distribution process. As a result, remediation is often difficult to accomplish quickly and effectively on an organizational scale.

Another major problem is that many patch management tools only work on Microsoft Windows due to dependencies on tools, such as Windows Server Update Services. Many of these tools do not work until endpoints are connected to a high-speed corporate network, leaving roaming computers and other mobile endpoints out of the update cycle for long periods.

The automated update processes usually require that you enter the following controls:

- ▶ Patch installation time windows
- ▶ Whether a user must be present
- ▶ Reboot options
- ▶ The method of distribution (including bandwidth and CPU throttles)
- ▶ System type
- ▶ User notification options

When new patch Fixlets must be evaluated by a Tivoli Endpoint Manager Agent, an administrator can immediately determine the scope of the update. While the Agent performs the continuous evaluation, a current list of noncompliant endpoints is always available. Because patch Fixlets include distribution instructions, there is no need to define them again. Administrators can then spend time to determine this information:

- ▶ When the patch must be distributed
- ▶ What notification to display to users (if any)
- ▶ Whether to allow users to delay a patch implementation and for how long
- ▶ Whether to force or delay reboots

Mostly within minutes, the Tivoli Endpoint Manager Agent receives the new patch execution order. If that patch is applicable, it downloads and applies the patch, reporting back success or failure within a short amount of time. During the patching process, the Tivoli Endpoint Manager Relay structure is heavily used to reduce network load and improve first-pass success rates.

It is an important advantage of the Tivoli Endpoint Manager solution that the Fixlets are automatically distributed to an Agent. The Tivoli Endpoint Manager method is that no action is carried out on a system without administrator confirmation. The platform architecture assures that a Fixlet can only read information using base operating system I/O. If changes are to be applied on a target system, for example, a patch package needs to be installed, an administrator confirmation is required. A security password and a digital signature are required.

The solution introduces various access levels, ensuring that only authorized administrators can create and distribute policies. Tivoli Endpoint Manager stores audit information that tracks who ordered which policies to be applied to which endpoints. Tivoli Endpoint Manager does not require specific operating system expertise for the operators that initiate the remediation process.

Confirm

After patches are scheduled to be applied, successful installation must be confirmed so that IT operations knows when the patch cycle is complete, and to

support compliance reporting requirements. This data must be communicated back to a central reporting system as fast as possible, ideally in real time.

After a patch is deployed, the Tivoli Endpoint Manager Agent automatically and continuously reassesses the endpoint status to confirm successful installation. The Tivoli Endpoint Manager Agent immediately updates the management Server in near real time or at the earliest opportunity for roaming devices. This step is critical in supporting compliance requirements, which require a proof of patch installation. With this solution, an operator can observe the patch deployment process in near real time, while Tivoli Endpoint Manager reports the actual stage of the process in the Console.

Enforce

After the application of the patch, it is important to ensure that the update continues to take effect. Users sometimes intentionally or accidentally uninstall patches, and new applications or patches might corrupt existing updates. If a patch is removed contrary to security policy, it must be reinstalled immediately. If a patch creates a major problem after application, organizations must also be able to issue a rapid mass rollback. The Tivoli Endpoint Manager Agent continuously enforces patch policy compliance, ensuring that endpoints remain updated. If a patch is uninstalled for any reason, the policy can specify that the Agent must automatically reapply it to the endpoint as needed. If there are problems with a patch, Tivoli Endpoint Manager operators can quickly and easily issue a rollback to endpoints either in total or to a select few.

Report

Reporting is an important component of the patch management process. Compliance policies might require detailed, up-to-date dashboards and reports that indicate the risk position of the organization and the patch management status. The Tivoli Endpoint Manager integrated web reporting capabilities allow operators to view dashboards and reports. These dashboards and reports can indicate the patches that were deployed, when they were deployed, who deployed them, and to which endpoints. The dashboards show patch management progress in near real time.

Tivoli Endpoint Manager for Patch Management solution

Tivoli Endpoint Manager for Patch Management offers wide coverage, speed, automation and cost-effectiveness, providing comprehensive operating system and third-party application patches. Policy-based controls provide administrators with fine-grained, automated patch management capabilities, and comprehensive reports. Policy compliance is continuously assessed and enforced.

Another key aspect of the architecture is support for endpoints that are on and off the corporate network. Roaming devices, such as mobile computers, for example, can receive patches through any Internet connection, such as Wi-Fi or even dial-up. The patch management process is virtually transparent to the user.

Organizations might need to establish, document, and prove compliance with patch management processes to comply with regulations, service level agreements (SLAs), and policies. The ability of Tivoli Endpoint Manager to enforce policies and quickly report on compliance can help improve the audit readiness of an organization.

2.3.3 Security configuration management

Controlling the security status of managed environments is not limited to applying the latest security patches. Security requirements relate to various regulations, which are either internal to the organization or defined by government or other organizations. External policies, prepared by different organizations, define rules that must be obeyed to meet compliance levels. The Tivoli Endpoint Manager platform offers a capability to support compliance verification against various sets of rules, which is referred to as *security configuration management*.

Tivoli Endpoint Manager provides a set of policy libraries. These libraries support continuous enforcement of configuration Baselines, including reporting, remediation, and confirmation of remediation of noncompliant endpoints in near real time. This feature can deliver meaningful information about the health and security of endpoints. This feature provides information regardless of location, operating system, connection (including wired computers or intermittently connected mobile computers), or applications installed. It can help consolidate and unify the compliance lifecycle, reducing endpoint configuration and remediation times.

The provided functionality in the security configuration management area is similar to the functionality of patch management. The power of the Tivoli Endpoint Manager platform is commonly used for multiple purposes.

Security Content Automation Protocol (SCAP)

The ability to automate technical configurations on devices across organizational infrastructures is always a challenge. Organizations, such as the NIST, National Security Agency (NSA), the Center for Internet Security (CIS), and the DISA attempt to provide guidance through documentation, standards, and guidelines. The real problem is to provide a means to translate regulations into rules and make it work on the unified platform. As a part of security configuration

management, SCAP is a method for automating the conversion of definition into a set of assessment rules.

SCAP⁴ consists of a set of standards that enable automated vulnerability management, measurement, and compliance evaluation. Specifically, SCAP addresses the following objectives:

- ▶ Enumerate software and security-related configuration issues.
- ▶ Measure systems to determine the presence of vulnerabilities.
- ▶ Evaluate the results of the compliance by defining measurement techniques.

SCAP consists of the following standards:

- ▶ Common Vulnerabilities and Exposures (CVE)

The SCAP CVE standard is a dictionary of publicly known information security vulnerabilities that enable data exchanges between security products. This standard provides a Baseline index point for evaluating the coverage of tools and services. Tivoli Endpoint Manager supports CVE for several versions of the product. Any security patch or vulnerability that has an associated CVE ID and is available as either a SCAP data stream or is available through other Tivoli Endpoint Manager developed processes displays the relevant CVE ID within the Tivoli Endpoint Manager Console. For more information, see this website:

<http://cve.mitre.org/>

- ▶ Common Configuration Enumeration (CCE)

The SCAP CCE standard provides unique identifiers to system configuration issues to facilitate fast and accurate correlation of configuration data across multiple information sources and tools. For example, CCE IDs can associate checks in configuration assessment tools with statements in configuration preferred practice documents. Tivoli Endpoint Manager supports CCE and displays the CCE ID for each misconfiguration for which there is a CCE ID within the Tivoli Endpoint Manager Console. In the case where a misconfiguration is associated with multiple CCE IDs, all IDs are cross-referenced and displayed. For more information, see this website:

<http://cce.mitre.org/>

- ▶ Common Platform Enumeration (CPE)

The SCAP CPE standard is a structured naming scheme for information technology systems, platforms, and packages. CPE is based on the generic syntax for Uniform Resource Identifiers (URI). CPE includes a formal name format, a language for describing complex platforms, a method for checking names against a system, and a description format for binding text and tests to a name. Tivoli Endpoint Manager uses CPE to ensure that configuration

⁴ For more information about SCAP, go to this website: <http://scap.nist.gov/>

settings are assessed on the correct system. Regardless of the operating system, the CPE ID can identify a platform and ensure that an assessment is performed. For more information, go to this website:

<http://cpe.mitre.org/>

▶ Common Vulnerability Scoring System (CVSS)

The SCAP CVSS standard provides an open framework for communicating the characteristics of IT vulnerabilities. Its quantitative model ensures repeatable, accurate measurement while displaying vulnerability characteristics used to generate the scores. Tivoli Endpoint Manager assesses and reports on vulnerabilities and quantifies the impact for multiple computing platforms. For more information, see this website:

<http://www.first.org/cvss>

▶ Extensible Configuration Checklist Description Format (XCCDF)

The SCAP XCCDF standard is a specification language for writing security checklists, benchmarks, and related documents. An XCCDF document represents a structured collection of security configuration rules for several sets of target systems. The specification also defines a data model and format for storing results of checklist compliance testing.

SCAP data streams use the XCCDF format to translate underlying configuration checks that are defined in Tivoli Endpoint Manager Fixlets. When created, these SCAP-based configuration Fixlets allow administrators to assess their computing assets against the SCAP-defined configuration rules in real time and on a global scale. For more information, see this website:

<http://scap.nist.gov/specifications/xccdf/>

▶ Open Vulnerability and Assessment Language (OVAL)

The SCAP OVAL standard is an international, information security community standard that promotes security content and standardizes the transfer of this information across an entire spectrum of security tools and services. The OVAL language is a collection of XML schema for representing system information, expressing specific machine states, and reporting the results of an assessment. For more information, go to this website:

<http://oval.mitre.org/>

Security configuration management checklists

The checklists provided as part of Tivoli Endpoint Manager for Security and Compliance enable organizations to gain visibility into the security configurations of their systems. Checklists are provided for Windows, UNIX, and Linux systems and are based on industry best standards. Organizations can use these

checklists as a starting point or customize them through parameterization to meet their specific security standards.

The following list presents the checklists that are available at the time of writing this document:

- ▶ Federal Desktop Core Configuration (FDCC):
 - Windows XP
 - Windows XP Firewall
 - Windows Vista
 - Windows Vista Firewall
 - Internet Explorer 7
- ▶ United States Government Configuration Baseline (USGCB):
 - Windows 7
 - Windows 7 Energy
 - Windows 7 Firewall
 - Internet Explorer 8
- ▶ Defense Information Systems Agency - Security Technical Implementation Guide (DISA STIG):
 - Windows XP
 - Windows Vista
 - Windows 7
 - Windows 2003 Domain Controller/Member Server
 - Windows 2008 Domain Controller/Member Server
 - AIX 5.1, 5.2, and 5.3
 - Red Hat Enterprise Linux 3, 4, and 5
 - Hewlett-Packard UNIX (HP-UX) 11.00, 11.11, and 11.23
 - Solaris 8, 9, and 10

Each checklist is offered as a Fixlet Site. The full solution contains Fixlets for compliance state evaluation, analysis for detailed data review, and reporting. Administrators can get the information about workstation noncompliance and start the remediation action to fix the detected issue. We presented the processing flow in Figure 2-8 on page 48. Tivoli Endpoint Manager Agent performs the same activities related to the security configuration of examined endpoints. The Agent delivers meaningful information about the health and security posture of endpoints regardless of location, operating system, applications installed, or connection type.

Checklist availability: The list of checklists provided by IBM might change according to demands and needs.

2.3.4 Security Compliance Analytics

Tivoli Endpoint Manager for Security and Compliance Analytics (SCA) is a web-based application for security and risk assessment. The system archives security compliance check results to identify configuration issues and report levels of compliance toward security configuration goals.

SCA is a component of Tivoli Endpoint Manager for Security and Compliance. It includes libraries of technical controls and tools based on industry best practices and standards for endpoint and server security configuration. The technical controls enable continuous, automated detection and remediation of security configuration issues. Report views and tools for managing the security configuration management checks are provided by SCA.

Tivoli Endpoint Manager for Security and Compliance Analytics generates the following reports, which can be filtered, sorted, grouped, exported, printed, emailed, and customized using any set of Tivoli Endpoint Manager properties:

- ▶ Overviews of Compliance Status and History
- ▶ Checklists of Compliance Status and History
- ▶ Checks of Compliance Status, Values, and History
- ▶ Computers and their Compliance Status, Values, and History
- ▶ Computer Groups and their Compliance Status and History
- ▶ Exceptions and their Management, Status, and History

By using Tivoli Endpoint Manager for Security and Compliance Analytics, you can navigate and explore security configuration check results. Each computer in your deployment evaluates the appropriate security configuration management checks that you activated using the Tivoli Endpoint Manager Console. Each computer reports a *pass*, *fail*, or *not applicable* for each check. Each endpoint also reports computer properties and analysis values, such as security configuration management check measured values, that are active in your Tivoli Endpoint Manager deployment.

The security configuration management check results are aggregated by the Tivoli Endpoint Manager for Security and Compliance Analytics server. The check results are augmented with computer properties and analysis values to provide compliance overviews and drill-down lists into the results. Your navigation paths through these overviews and detailed report views vary depending on the desired compliance reports.

Security and Compliance Analytics reporting

Tivoli Endpoint Manager for Security and Compliance Analytics reports display graphical and tabular views of aspects of your deployment compliance status.

Four major report types are available in the system. Each type displays a different, configurable view of the current and historical compliance status of the deployment. All users with accounts on the system can see all report types. The data visible to each user depends on the computers to which they are granted visibility.

Tivoli Endpoint Manager for Security and Compliance Analytics provides the following set of reports:

- ▶ Overview reports

This type of report is designed to present general information about selected objects. It can provide high-level information about object relationships, for example, on how many computers a particular check is being evaluated. It provides the graphical representation of the compliance state changes. The following types of reports are available:

- Deployment Overview: This report shows deployment information (such as quantity of computers and quantity of checks). It shows the overall, historical aggregate compliance for all checks on all computers visible to the logged-in user. This report is presented in Figure 2-9 on page 58.

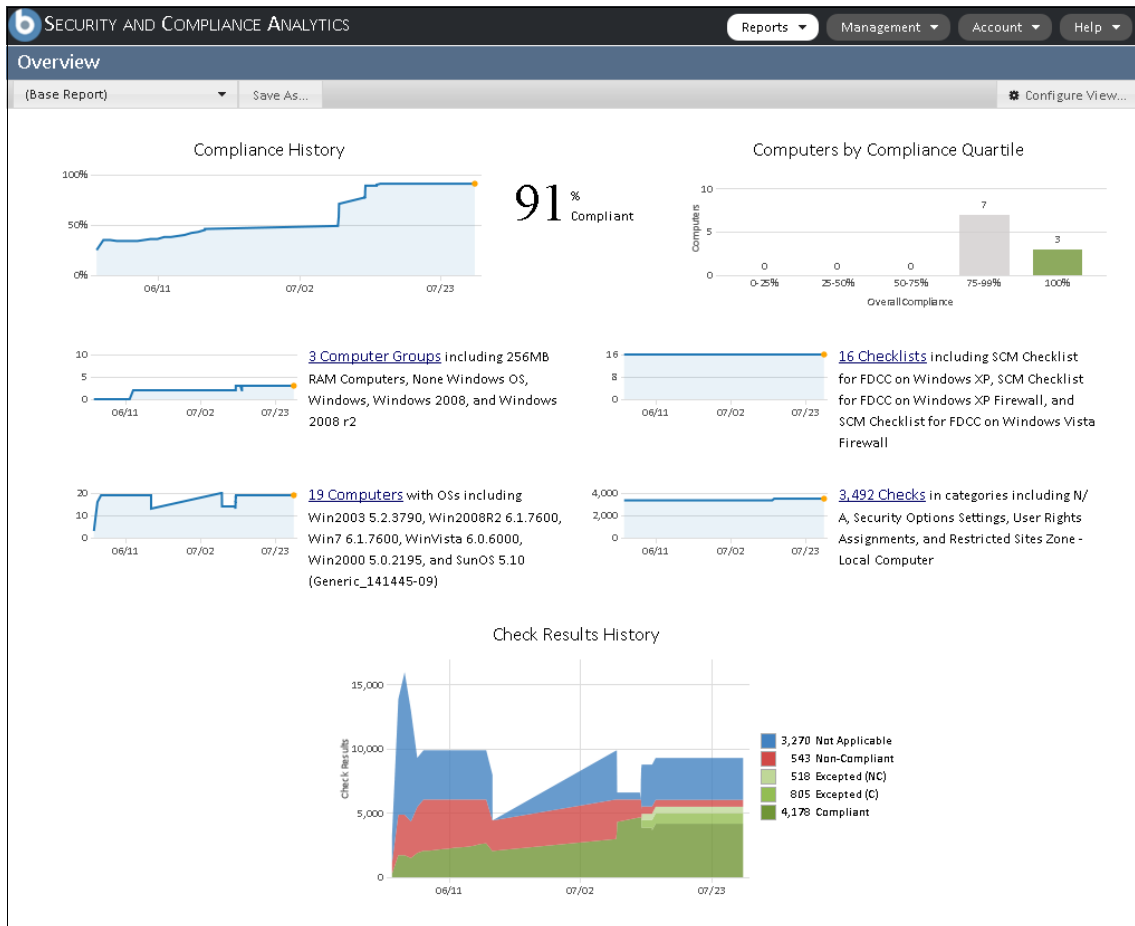


Figure 2-9 Security and Compliance Analytics: Deployment Overview report

- Checklist Overview: This report shows information about a single checklist (such as quantity of checks in the checklist). It shows the overall, historical aggregate compliance for the checklist as applied to all computers visible to the logged-in user.
- Check Overview: This report shows information about a single check (such as check source and check description). It shows the overall, historical aggregate compliance of the check as evaluated by all computers visible to the logged-in user. A sample Check Overview report is depicted in Figure 2-10 on page 59.

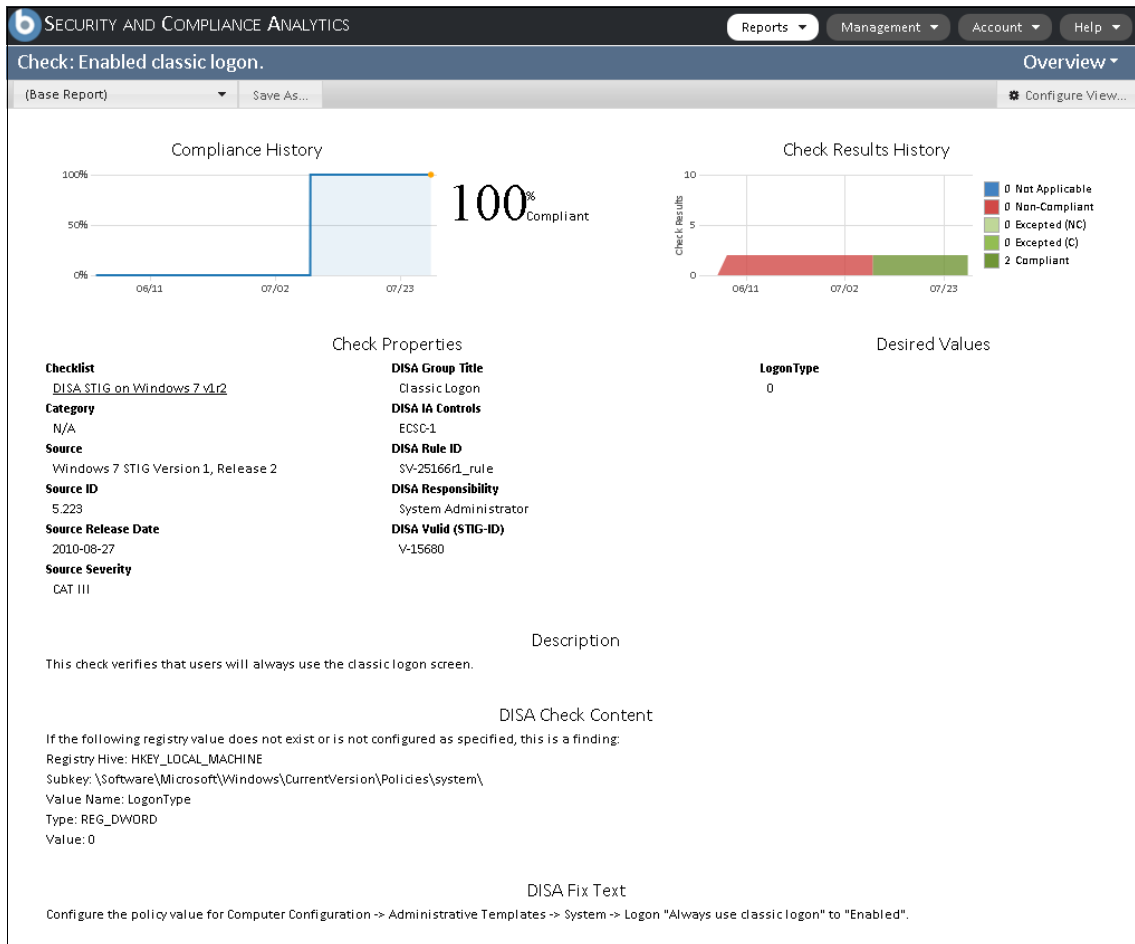


Figure 2-10 Security and Compliance Analytics: Check Overview report

- Computer Group Overview: This report shows information about a computer group (such as number of children/subgroups and number of member computers). It shows the overall, historical aggregate compliance of the group.
 - Computer Overview: This report shows information about a single computer (such as number of checks evaluated on the computer). It shows the overall, historical aggregate compliance of all checks evaluated by the computer.
 - ▶ List reports
- List reports provide more detailed views of the inspected aspects. These reports organize data into a table that allows a high level of customization for

an administrator. Each overview report described (other than the Deployment Overview) has a corresponding list report. In addition, there are two special reports available: Check Results and Exception Results. The following list presents all types of list reports:

- Checklists: This report shows the list of checklists in the deployment along with attributes of each checklist. It shows the overall, historical aggregate compliance results of all checks on all visible computers for each checklist. A sample report is shown in Figure 2-11.

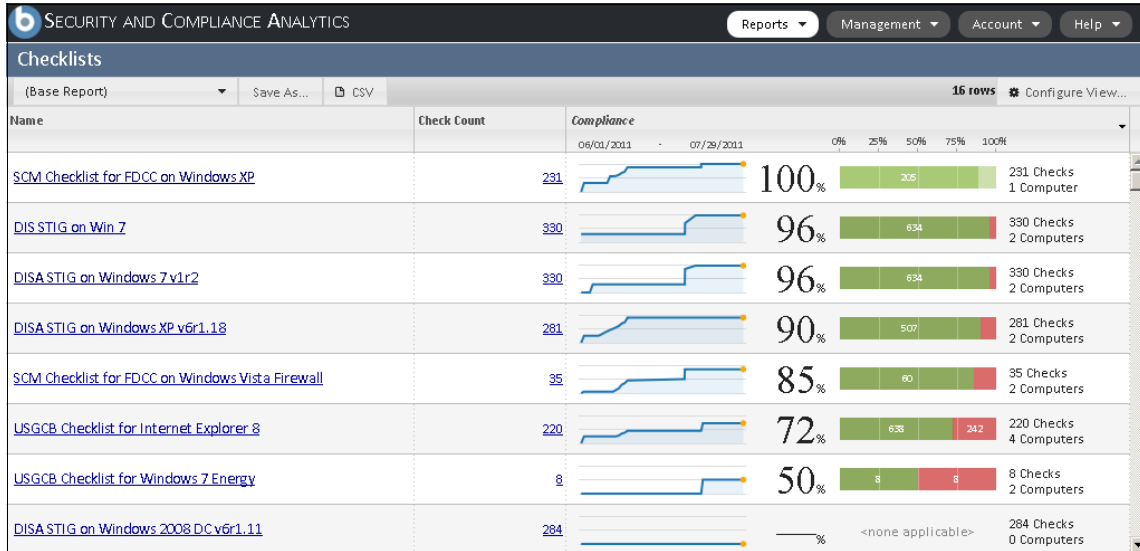


Figure 2-11 Security and Compliance Analytics: Checklists report

- Checks: This report shows the list of checks in the specific scope along with attributes of each check. It shows the overall, historical aggregate compliance results (that is, the aggregate of all the visible computer pass or fail scores) of each check.
- Computer Groups: This report shows the list of all computer groups in the specific scope visible to the logged-in user along with the attributes of each group. It shows the overall, historical aggregate compliance results of all checks on all computers in each group.
- Computers: This report shows the list of all computers in the specific scope visible to the logged-in user along with attributes of each computer. It shows the overall, historical aggregate compliance results of all checks evaluated on the computer.
- Check Results: This report shows the list of all checks and all computers in the specific scope visible to the user along with attributes of each

computer and each check. It shows the historical compliance result (pass, fail, excepted, or not applicable) for each check on each computer. This report allows an administrator to inspect each check result collected by a Tivoli Endpoint Manager Agent. Beside the typical compliance state, it offers additional functionality to review the actual data that was gathered from the endpoint by activated analysis. Based on that information, an administrator can review why the computer was marked as noncompliant. A sample Check Results reports is presented in Figure 2-12.

Checklist	Check Name	Desired Values	Compliance	Measured Values
(Base Report)	Save As... CSV		06/01/2011 - 07/26/2011	330 rows
DISA STIG on Windows 7 v1r2	Auditing records are configured as required...	Sensitive Privilege Use: SUC	Compliant	Sensitive Privilege Use: True, Sensitive Privilege Use: True,
DISA STIG on Windows 7 v1r2	Configure the default autorun behavior to pr...	NoAutorun: 0	Non-Compliant	NoAutorun: 1, NoAutorun: 1
DISA STIG on Windows 7 v1r2	Prevent Windows Update for device driver s...	SearchOrderConfig: 0	Compliant	SearchOrderConfig: 0, SearchOrderConfig: 0
DISA STIG on Windows 7 v1r2	The system is not configured to recommend...	LDAPClientIntegrity: 1	Compliant	LDAPClientIntegrity: 1, LDAPClientIntegrity: 1
DISA STIG on Windows 7 v1r2	Minimum password age does not meet mini...	MinimumPasswordAge: 1	Compliant	MinimumPasswordAge: 1, MinimumPasswordAge: 1
DISA STIG on Windows 7 v1r2	User Account Control - Non UAC compliant a...	EnableVirtualization: 1	Compliant	EnableVirtualization: 1, EnableVirtualization: 1
DISA STIG on Windows 7 v1r2	User rights and advanced user rights settin...	SeImpersonatePrivilege: *S	Compliant	SeImpersonatePrivilege: s-1, SeImpersonatePrivilege: s-1
DISA STIG on Windows 7 v1r2	Prevent Microsoft Support Diagnostic Tool f...	DisableQueryRemoteServer	Non-Compliant	DisableQueryRemoteServer: 1, DisableQueryRemoteServer: 1

Figure 2-12 Security and Compliance Analytics: Check Results report

- Exception Results: This report shows the list and status of exceptions in the specific scope applied to each computer visible to the logged-in user, along with attributes of each check, each computer, and each exception. Because exceptions are inevitable in any managed environment, this report provides a detailed view of all defined rules including the time frame within which the particular exception is effective. Figure 2-13 on page 62 shows a sample Exception Results report.

Checklist	Check Name	Computer Name	Expiration Date	Reason	State
SCM Checklist for FDCC on Windows XP	Security Patches Up-To-Date	VSXPS232-01	Never	Just because	Excepted (NC)
SCM Checklist for FDCC on Windows XP	Domain member: Disable machine account password...	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	Retention of Events in Security Log	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	Network access: Named Pipes that can be accessed an...	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	mshta.exe Permissions	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	Impersonate a Client After Authentication	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	Network DDE Share Database Manager (DSDM) Service...	VSXPS232-01	Never	Just because	Excepted (C)
SCM Checklist for FDCC on Windows XP	WebClient Service	VSXPS232-01	Never	Just because	Excepted (C)

Figure 2-13 Security and Compliance Analytics: Exception Results report

2.4 Conclusion

In this chapter, we introduced the Tivoli Endpoint Manager platform and explained how it can provide an organization with visibility, control, and accuracy over its distributed endpoints. We explained how Tivoli Endpoint Manager can safeguard the endpoint devices of an organization. It enforces security compliance by using security configuration management and patch distribution for operating systems and applications. We also explained that Tivoli Endpoint Manager can provide both near real-time and historical compliance data for every endpoint by using the Console, Web Reports, or Security and Compliance Analytics.



IBM Tivoli Endpoint Manager component structure

In this chapter, we introduce the major components of the Tivoli Endpoint Manager solution by presenting an initial component overview. We then decompose the elements that make up the Tivoli Endpoint Manager system. This information can help you better understand sizing, deployment options, and other architectural considerations.

Where relevant, we identify key differences between Version 8.2 of the Tivoli Endpoint Manager system and the previous editions.

3.1 Logical component overview

In 2.2.4, “Managed environment” on page 32, we introduced the key components of the Tivoli Endpoint Manager platform. In Figure 3-1, we depict these components placed in a logical context; in this section, we take a closer look at each of these components.

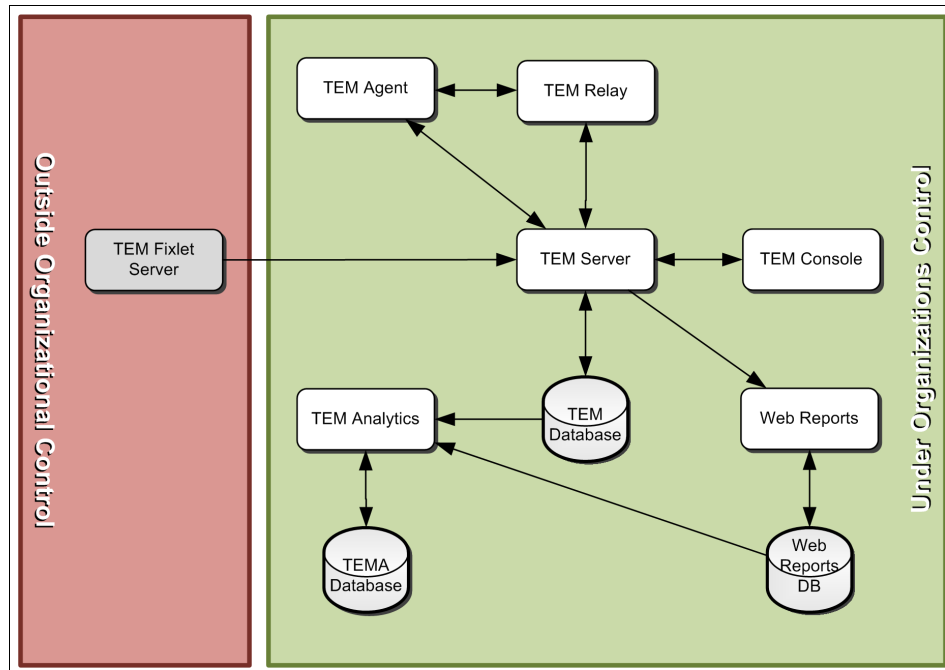


Figure 3-1 High-level components in a logical context

Most of the components in Figure 3-1 are within the control and responsibility of the organization that deploys the overall solution, except for the Tivoli Endpoint Manager Fixlet Servers. The Fixlet Servers exist outside of the organizational control and are hosted by IBM.

Arrows in the diagram indicate data flow, not network traffic initiation. Network flow is examined further in 3.3, “Network communications and usage” on page 106. Figure 3-1 shows that data flows from the Fixlet Servers (external to organization) into the Tivoli Endpoint Manager Server. This diagram does not mean that the Fixlet Servers initiate the connection; it merely depicts the way that the data flows.

A basic Tivoli Endpoint Manager solution contains a Tivoli Endpoint Manager Server, Tivoli Endpoint Manager database, Tivoli Endpoint Manager Console,

and one or several Tivoli Endpoint Manager Agents. Tivoli Endpoint Manager Relays are suggested in most deployments. Other components are optional, depending on what features of the system the organization plans to use.

Next, we explain each component in turn and its role in the overall system.

3.1.1 Fixlet Server

We first look at the Fixlet Servers and how they interact with the Tivoli Endpoint Manager Server, shown in Figure 3-2.

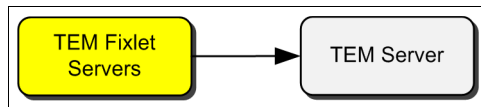


Figure 3-2 Fixlet Server component in context

In 2.2.5, “Key terms” on page 36, we briefly introduced Fixlet messages and Sites. Sites are a logical collection of Fixlets that are related in some way. For example, they all relate to patching Microsoft Windows endpoints. Or, they relate to measuring and managing compliance to a particular standard or operating system. These Sites are normally hosted on the IBM Tivoli Endpoint Manager Fixlet Servers as a *cloud service* and are therefore external to the organization that uses Tivoli Endpoint Manager. A Fixlet Server is not a component that the organization must build or maintain.

By hosting the sites externally, each organization can subscribe to a content feed from these servers and receive updates to their content dynamically, when that content changes. This approach removes the manual steps required to ensure that the organization has the latest patches, compliance controls, or any other update to a subscribed site.

Obtaining updates in this way does not actively change your environment, it merely provides the new content. Use of this new content to change the organization endpoints must still be performed by an approved administrator. Furthermore, any existing actions are equally not affected by this new content.

3.1.2 Tivoli Endpoint Manager Server

Next, we look at the Tivoli Endpoint Manager Server. We look at how it interacts with the other surrounding Tivoli Endpoint Manager components, shown in Figure 3-3 on page 66.

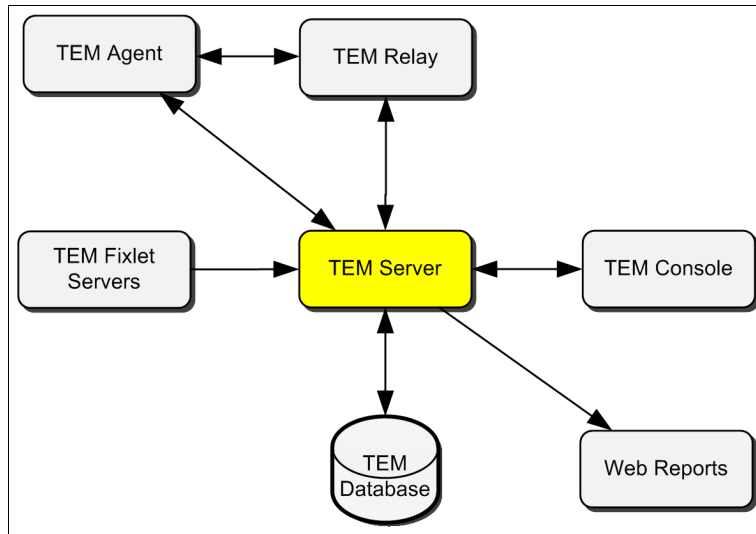


Figure 3-3 Server component in context

The Tivoli Endpoint Manager Server provides the central server functionality for administering the Tivoli Endpoint Manager infrastructure. It forms the core of the Tivoli Endpoint Manager system, orchestrating the flow of information. It distributes content to Relays and Agents, and it brokers the data returned for insertion into the Tivoli Endpoint Manager database. The Tivoli Endpoint Manager Server is the only device responsible for obtaining information from the external Fixlet Servers. It is often the only device in the entire Tivoli Endpoint Manager system that needs access to the Internet. Even access to the Internet can be avoided with air-gap¹ configurations, in which the Tivoli Endpoint Manager Server exists in an environment logically isolated from the Internet. The Tivoli Endpoint Manager Server brokers most access to the Tivoli Endpoint Manager database, reducing the exposure of the database to the network.

The Tivoli Endpoint Manager Server is an application that requires little resource. Most resources are required by the Tivoli Endpoint Manager database, which is often installed on the same system with the Tivoli Endpoint Manager Server. The Tivoli Endpoint Manager Server is administered by authorized users or *operators*, connecting to it by using the Tivoli Endpoint Manager Console. It is the responsibility of the Tivoli Endpoint Manager Server to authenticate operators.

¹ In an isolated or “air-gap” configuration, external data must be manually brought into the system because Tivoli Endpoint Manager has no Internet access. IBM provides tools to facilitate this process.

Important: When deploying the Tivoli Endpoint Manager database on the same system as the Tivoli Endpoint Manager Server, ensure that the database is configured not to use all available memory. Leave at least 4 GB, preferably 8 GB, free for the Tivoli Endpoint Manager Server and other OS operations. This amount helps to avoid overuse of the virtual memory system and paging. Overuse can slow down the Tivoli Endpoint Manager operations.

The Tivoli Endpoint Manager Server receives data from Agents and Relays in the form of a *report*, which it stores in the Tivoli Endpoint Manager database. This data is made available through the Web Reports Server by periodic synchronization. Equally, the Console provides a view of the reported data to the Console operators.

When the Tivoli Endpoint Manager Server is installed, a private/public key pair is generated. The Tivoli Endpoint Manager Server private key must be treated with the highest level of security. It is central to the security of the overall Tivoli Endpoint Manager system and must be protected against unauthorized access. The public key is digitally signed by IBM Tivoli and inserted into a file called the *masthead*. The masthead, containing the digital certificate from the Server, exists on all Tivoli Endpoint Manager Relays and Tivoli Endpoint Manager Agents. Two important features are enabled:

► Content validation

All Fixlets (including Actions) are digitally signed by the Tivoli Endpoint Manager Server. This digital signature is verified by the endpoint before acting upon any Fixlet or Action. If the validation fails, the content is ignored. This technique ensures that no one can masquerade as the Tivoli Endpoint Manager Server, or cause any action to be taken, unless they have the Tivoli Endpoint Manager Server key material. Even if a malicious party obtained access to the Tivoli Endpoint Manager database, that person cannot cause Tivoli Endpoint Manager to act. It is imperative to protect the Tivoli Endpoint Manager Server private key.

► Message level encryption

Message level encryption (MLE) allows a Tivoli Endpoint Manager Agent to return information to the Tivoli Endpoint Manager Server as encrypted data (encrypted reports). This encryption exists at the data level and not the transport level. An encryption public key is added to the Agent masthead file². The Agent generates a “session key” (Advanced Encryption Standard (AES) 256) that it uses to encrypt the data. This “session key” is then encrypted with the “encryption public key”, resulting in an “encrypted session key”. The

² All Tivoli Endpoint Manager Servers have a public/private key pair, but the additional encryption public/private keypair is only needed when MLE is enabled. The encryption public/private key pair is generated by using the BES Administration tool.

“encrypted session key” is added to the report to be sent to the Server. Effectively, the data is only decryptable by an entity with the “encryption private key” that corresponds to the “encryption public key” from the masthead file. In most circumstances, this entity is the Tivoli Endpoint Manager Server, but it is also possible to provide the “encryption private key” to Tivoli Endpoint Manager Relays. The use of this mechanism is depicted in Figure 3-4.

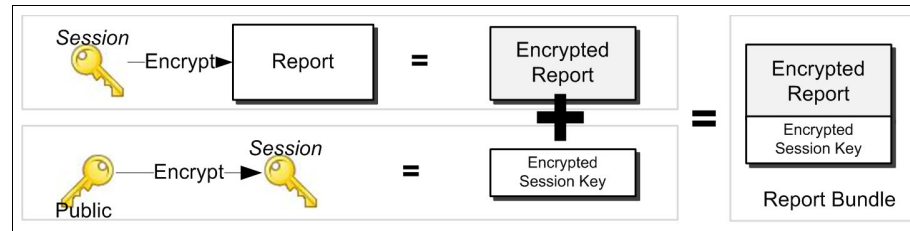


Figure 3-4 Message level encryption key use

Encryption versus signing: MLE does not encrypt data sent to the Agents, only data that comes from the Agents. Data sent to the Agents is digitally signed. Data from the Agent is not. With these functions, data from the Server is digitally signed. Data to the Server is encrypted.

Think of Tivoli Endpoint Manager as a *Secure Question and Answer System*. The Tivoli Endpoint Manager Server distributes content in Fixlet messages. For more information, see 3.2.8, “Fixlet message structure” on page 105. Think of the content of the Fixlet messages as the *questions* that we want answered by the endpoints. Often, these questions are answered with true or false (yes or no), such as “Do you need Patch MS11-056?” But, you can also have more interesting questions, such as “How much free disk space does the computer have?” One major benefit of the Tivoli Endpoint Manager system is the ability to change an environment, to *fix* the issues that are identified. Tivoli Endpoint Manager calls these instructions *Actions*, another type of Fixlet message.

We described five major types of *Fixlet message* (also known as *Fixlets*, which is a trademarked term) in the previous chapter:

- ▶ Fixlet: Identifies a problematic situation with knowledge of how to fix it.
- ▶ Task: Like a Fixlet, but it does not require a *problem*.
- ▶ Baseline: A Meta Fixlet (or Fixlet of Fixlets). A container for other Fixlets and Tasks, deployable as a single entity.
- ▶ Analysis: More detailed questions provide more insight into a situation. Typically, it is more than yes or no answers.

- ▶ Action: The approved, signed execution of a Fixlet, Task, or Baseline.

The Tivoli Endpoint Manager Server can run on almost any current version of Microsoft Windows. For production deployments, it is advisable that the Tivoli Endpoint Manager Server is deployed on either of these versions:

- ▶ Microsoft Windows 2008
- ▶ Microsoft Windows 2008 R2 (suggested)

The Tivoli Endpoint Manager Server greatly depends on the database software, which must be one of the following products:

- ▶ Microsoft SQL Server 2005
- ▶ Microsoft SQL Server 2008
- ▶ Microsoft SQL Server 2008 R2 (suggested)

The choice between standard and enterprise editions of both OS and SQL Server depends on the size of the deployment. For more information, see the IBM support website:

<http://support.bigfix.com/bes/install/serverreq.html>

There are no other software requirements for the Tivoli Endpoint Manager Server.

Previous versions: If you use a version of Tivoli Endpoint Manager before Version 8.2, the Console, Server, or database interactions differ slightly:

- ▶ The Server is not involved in authenticating Console users.
- ▶ Content is not signed by the Tivoli Endpoint Manager Server, but by a Console user that uses its own key issued by the Tivoli Endpoint Manager Server.
- ▶ The Console wrote directly to the database and then notified the Server that content changed.

3.1.3 Database

Next, we take a closer look at the Tivoli Endpoint Manager database and how it interacts with the Tivoli Endpoint Manager Server and Analytics components, shown in Figure 3-5 on page 70.

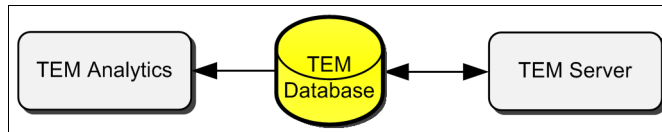


Figure 3-5 Database component in context

The Tivoli Endpoint Manager database stores the digitally signed Fixlet messages, results returned from endpoints, and other configuration information. To facilitate multiple Agents that report the results of Fixlets and Analysis to the Server, the database is designed for write optimization (instead of being designed to read large amounts of data quickly). The database mostly writes small amounts of data. The configuration must deliver good performance from the database, which is discussed in the IBM developerWorks® wiki³. The Tivoli Endpoint Manager database is named “BFEnterprise”.

Various technologies are used for the database storage. *Storage latency* is identified as a large factor in the storage technology selection for a deployment, as well as *I/O operations per second (IOPS)* and *throughput*.

Solid-state PCIe performance: A RAID 10 configuration generally is an excellent choice. The IBM High IOPS PCIe Solid State Storage technology is used in many large deployments where database performance is critical. The following solid-state PCIe performance information is from <http://www.ibm.com/systems/storage/disk/ssd/>:

- ▶ Latency of 50 μ s, 1/100th of a reference mechanical hard disk (15K RPM)
- ▶ 100,000 IOPS versus 420 IOPS for a reference mechanical hard disk
- ▶ On-chip redundancy lessens the need for RAID
- ▶ Predictable mean time between failures (MTBF) with no mechanical parts
- ▶ High IOPS and throughput
- ▶ Low-power utilization, reducing heat in the data center to help ensure longevity

The database schema is open and available, allowing the organization and other users to access the data with tools external to the Tivoli Endpoint Manager System. Other application programming interfaces (APIs⁴) are also available that might be preferable. A SOAP-based API to the Server is available instead of communicating with the database and exposing it on the network.

³ <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Server%20Disk%20Performance>

⁴ <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Customizations>

The Server brokers most connections to the database, except when the database is deliberately exposed to allow external entities to query it.

It is common to see the Server and the database on the same physical system as the Tivoli Endpoint Manager Server. This design reduces the communication path between the Tivoli Endpoint Manager Server and the Tivoli Endpoint Manager database and has other benefits. However, you must be careful to ensure that the system has enough resources to prevent it from exhausting physical memory. Be careful to ensure that I/O bottlenecks are avoided. They can occur when you use the same disks and file systems for both the Tivoli Endpoint Manager Server and the Tivoli Endpoint Manager database. Counter this issue by using different disks for the database to avoid possible contention.

The Tivoli Endpoint Manager database and the Tivoli Endpoint Manager Server components can be on physically separate devices. However, with the amount of data exchanged between the Tivoli Endpoint Manager Server application and the Tivoli Endpoint Manager database, performance is optimized when they coexist on the same physical system. Both configurations are supported: the database and Server physically separated or collocated on the same system. Chapter 4, “IT endpoint security and compliance solution design” on page 125 identifies configuration considerations.

The role of the database is to store the current state of the Tivoli Endpoint Manager environment. It does not store historical information or capture previous states of the endpoints. It represents the state of the organization endpoints at the current point in time. For historical representations of data, see 3.1.8, “Analytics” on page 84.

Previous versions: If you use a version of Tivoli Endpoint Manager before Version 8.2, the Console/Server/database interactions differ slightly:

- ▶ The Tivoli Endpoint Manager database is more exposed in versions before Version 8.2. Consoles are required to connect directly to the database for authentication and to change the Tivoli Endpoint Manager system.
- ▶ Users are authenticated by their database passwords.
- ▶ Content is signed with the Console user key.
- ▶ The Console wrote directly to the database and notified the Server that content changed.

3.1.4 Console

The Tivoli Endpoint Manager Console allows an operator to interact with the Tivoli Endpoint Manager Server, shown in Figure 3-6 on page 72.

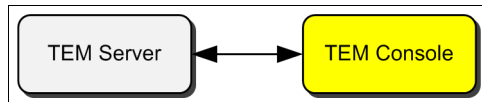


Figure 3-6 Console component in context

The Console is a Microsoft Windows application that is used to administer and configure the Tivoli Endpoint Manager environment. It provides its operators with a single tool to view the security status of all endpoints that report to the system, such as their *security patch* status or *compliance* to corporate policy. It presents the Tivoli Endpoint Manager operator with reports, dashboards, and an overall view of the system.

Operators can be restricted to prevent them from seeing certain content and specific endpoints (including groups of machines). For more information about the various users of Tivoli Endpoint Manager, see 3.1.10, “Users” on page 86. The Console communicates with the Tivoli Endpoint Manager Server over a Secure Sockets Layer (SSL)-encrypted HTTP connection (HTTPS), ensuring that the interaction between the two components remains secure.

Authenticating Console operators is performed at the Server, which in turn might authenticate the user locally or by using a directory service. When a user changes content, the content is sent over the HTTPS connection to the Server where it is digitally signed by the Server. The content is entered into the database before the Tivoli Endpoint Manager Server notifies Agents of the changed content.

An in-memory cache is used by the Tivoli Endpoint Manager Console for data that is received from the Server. By maintaining a sequence number for the received data, the Console can request only the data that is updated after the current sequence number. This method reduces network utilization for Console operations. The frequency of this update must be tuned to the size of the deployment of the organization. In small deployments, the default refresh of 15 seconds is good. In a large deployment, 15 seconds might not be long enough to retrieve a single update before the next update is requested. Increasing the refresh frequency can make the Console more responsive and further reduce network utilization.

Even by reducing the Console refresh rate, there is still a large amount of data that is exchanged between the Console and the Server. This volume is compounded by having multiple Consoles, which occurs in every deployment that we see. Large amounts of data transferred across a network, particularly where Remote Access, WAN, or High Latency links are involved, can adversely affect the network performance.

To combat this issue, we often see Console operators use a remote desktop technology to log in to a Server that is both physically and logically close to the Tivoli Endpoint Manager Server. They typically use the same network switch. From this “Remote Console Server”, multiple Consoles can be run by multiple operators. By using a “Remote Console Server”⁵, the operator no longer receives large amounts of data across the network to a Console that runs on the workstation of that operator. Instead, the large amounts of data move between the Server and the Console close to the “Remote Console Server”. Only the presentation layer is transferred across the network to the remote desktop application of the operator.

The Tivoli Endpoint Manager Console requires Microsoft Windows XP or later. The content of the reports, dashboards, wizards, and other elements are delivered by using various web technologies through an embedded browser (Internet Explorer) in the Console application. Each of these browsers can require additional products, such as Adobe Flash.

Remote desktop: We commonly see Citrix, Microsoft Terminal Services/Remote Desktop, and Virtual Network Computing (VNC) used in this capacity. Each product has advantages. The choice is typically based on the product that is commonly used by an organization. The server is typically dedicated to running Tivoli Endpoint Manager Consoles. The Console is a 32-bit application. The in-memory caching can potentially use as much as 4 GB per console (for large installations).

⁵ Tivoli Endpoint Manager Console server is not an official part of the Tivoli Endpoint Manager System. It depicts a server that is running multiple Consoles that reside physically and logically close to the Tivoli Endpoint Manager Server.

Previous versions: If you use a version of Tivoli Endpoint Manager before Version 8.2, the Console, Server, and database interactions differ slightly:

- ▶ Users each have a unique private/public key pair.
- ▶ Users are authenticated by their database passwords.
- ▶ Content is signed with the Console user key.
- ▶ The Console writes directly to the database and then notifies the Server that content is changed.
- ▶ The Remote Desktop Server must be close to the Tivoli Endpoint Manager database, not the Tivoli Endpoint Manager Server.
- ▶ Creating users with a user interface (UI) is now performed by using the Console and not in the BESAdmin tool.
- ▶ When logging in, you no longer select an Open Database Connectivity (ODBC) connection. Instead, you provide the URL/IP address of the Tivoli Endpoint Manager Server to which you want to connect.

3.1.5 Relay

Next, we take a closer look at the Tivoli Endpoint Manager Relay and how it interacts with the Tivoli Endpoint Manager Server and Agent, shown in Figure 3-7.

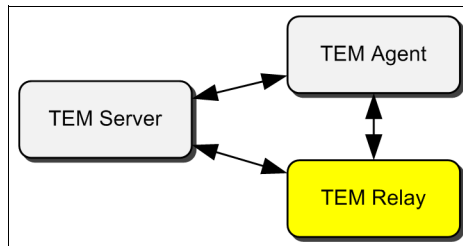


Figure 3-7 Relay component in context

The Tivoli Endpoint Manager Relay component plays a key role in obtaining the high levels of scalability that the Tivoli Endpoint Manager system can reach. The Relay offloads downloading updates and receiving reports from endpoints away from the Server, and maintains a low resource footprint. The footprint is so low that the Relay is designed to operate on an endpoint, sharing its available resources. It is intended that, for most circumstances, an organization does not dedicate hardware and OS resources to a Relay.

Relays have parents and children. Parents can either be Relays or the Tivoli Endpoint Manager Server. Children can either be Relays or Agents. In this way, the Relays can form an n -Tier hierarchy to move information between the Tivoli Endpoint Manager Server and Agents.

We often see Relays installed on file and print servers, domain controllers, and department servers, all of which do not primarily act as a Relay. Similar to other components in the Tivoli Endpoint Manager system, the Relay is a self-contained web-based application (it does not need a separate web server). The Relay workload is demand-driven (performing tasks requested by the Agents or Server). Even though we describe the use of “shared servers” as Relays, you can also run Relays on desktop computers. The Relay requires few resources.

Candidacy: Relays can be installed on various machines. It is important to identify devices with a *static IP address* (or infrequently changing IP address). The devices must be consistently online and available within the Tivoli Endpoint Manager environment. Notebooks and mobile devices are poor candidates for Tivoli Endpoint Manager Relays.

IBM suggests a ratio of one Relay for each 1,000 Agents in a shared environment. We advise that you use a lower ratio if your Relays are *not* servers, for example, desktops or workstations.

Agent registration

Agents can register through the Relay, which the Relay forwards on to the Server possibly through other Relays. The Relay keeps a record of the Agent IP address so that the Relay knows how to contact the Agent in the future.

Registering with Tivoli Endpoint Manager Relays: In reality, the Agent is not registering with the Relay; it is registering with the Server. The Relay is merely brokering the exchange. As a result, the Server must be online, although it does not need to be directly accessible by the Agent.

Message propagation

The Server must inform Agents of new content. Instead of sending thousands of *notifications* to Agents, the Server sends them to each of the Relays, which in turn propagate that message to other Relays. Finally, a Relay sends a notification to the Agents that registered through it. This message propagation helps to reduce the workload on the Server. It also reduces the network traffic when Relays are sufficiently distributed in relation to the Agents that they serve. The notification concept is downstream, from Server to Agent, but message propagation has a role to play upstream also. When sending data back to the

Server (reports), the Relay can bundle multiple messages together. Compression of these bundles further reduces the load to the network.

File Caching

When an Agent needs to download files, it does not go to the Internet or to the Server. Instead, it asks a Relay for the file. The Relay maintains a file cache and first looks in the cache for the requested file. If the file is present, it can serve it to the Agent. If the Relay does not find the file in its cache, it asks a parent Relay, or, eventually, the Server. They serve the file from their cache if they have it, causing the requesting Relay to also cache it. Only the Server needs to obtain the file from outside the system if it does not have the file already cached. The cache size on each Tivoli Endpoint Manager Relay is configurable, removing the oldest-requested files first.

Cache currency: If no Agent requests a file through a particular Relay, the Relay does not have it and does not need it. Content exists where it is used only, and it is not needlessly transferred across the network. If an Agent in the future requires the content, it is obtained by the Relay and cached for future requests.

Other capabilities of the Relay include *bandwidth management* and *throttling*. Throttling allows an organization to control how much bandwidth to use from Relay to Relay and between Relay and Agents. Relays also participate in an *automatic Relay selection*, where Agents attempt to register with the closest⁶ Relay. We describe this process from the view of the Agent in 3.1.6, “Agent” on page 77.

Relays are assigned to *affiliation groups*. Agents are configured to attempt registration with specific groups that are likely to be close to the Agent. A Relay can be affiliated with multiple groups, or none. The Agent can also *seek* Relays in multiple groups.

A Relay can be used in a special configuration where it can decrypt messages from the Agents. In 3.2.1, “Tivoli Endpoint Manager platform” on page 89, we describe Message Level Encryption (MLE). In MLE, the Agent can encrypt its reports so that only the Server can decrypt them. The process of decryption requires additional resource over nonencrypted reports. In certain configurations, you might want to offload this work to a few Relays. Each Relay receives the report from an Agent or a subordinate Relay. Normally, the Relay passes this report on to its parent, but in this case, it can decrypt the data if it is encrypted. The decrypted messages are then passed to the parent of the Relay (typically the Server).

⁶ The closest Relay is determined by network hops, which might not be a suitable method for all environments.

Top-level Relays: A top-level Relay is a designation sometimes for the Relays whose parent is the Server. These top-level Relays initiate the hierarchy and effectively keep load away from the Server. Typically, although not necessarily, the top-level Relays decrypt if the decryption is not performed on the Server. Offloading the decryption to Relays requires that the *encryption private key* is present at the decrypting Relay, further exposing the key. Additional care must be taken to protect this key and the machines that physically store it.

For an endpoint to be a Relay, it must also have an Agent installed. The Relay code can run on many operating systems. No other software is required, only the Agent and the Relay. The minimum requirements are sufficient for small Relay:Agent ratios. The greater workload of higher ratios requires more resources.

3.1.6 Agent

Next, we take a closer look at the Tivoli Endpoint Manager Agent and its interaction with the Tivoli Endpoint Manager Server and Relay, shown in Figure 3-8.

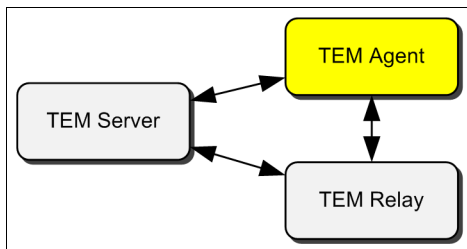


Figure 3-8 Agent component in context

The Agent, sometimes also called the Client, is the component on each endpoint managed by the system. It is a native executable and requires little resource. For a default installation, the Agent is throttled to use only 2% of the endpoint processor resource by being awake for approximately 20 ms out of every second. Depending on the platform, the installer is between 5 - 15 MB.

Resource utilization: A 2% resource utilization is achieved by performing useful work for 10 ms and sleeping for 480 ms (yes, this total does not equal 500 ms). This utilization configuration is a configurable item called the *idle mode*, and it is used most of the time. There are times when alternative modes are needed. An example is *normal mode*, where the CPU resource throttling is ignored. Normal mode is used for actions that must occur without sleep, such as downloading new content.

The Agent communicates with a parent, which might be a Relay or a Server, to receive Fixlet messages. The Fixlet message consists of various items:

- ▶ Questions that need to be answered, such as “How much memory does the Endpoint have?” or “Does the Endpoint need patch MS11-022?”
- ▶ Actions that the Agent must perform on the endpoint, such as “install patch MS11-022” or “set the minimum password length to 8 characters”

Before any question is answered and before any action is taken, the Agent verifies the digital signature of the content. The content must be signed with the Tivoli Endpoint Manager Server private key that corresponds to its public key. And, it must be distributed to the Agent *masthead file*, which also specifies the URL to the Server.

The ability of the Agent to provide answers to questions is implemented in the *Relevance Language*. The Relevance Language is a *loop-less* language. It invokes *inspectors* on the endpoints to allow the inspection and retrieval of various properties of the endpoint. For example, there might be an inspector to query information about a *folder*. This query might return a folder *object* whose properties can then be queried, such as in this example:

```
exists file “autoexec.bat” of folder “c:\”
```

Inspectors cover many aspects of the operating system and beyond, such as OS hardware, file system, registry, installation, and processes.

The questions and answers are separate from the execution, in this case, *Relevance* is separate from *Action*. This separation ensures that the Relevance clauses operate passively and do not change the system. Changes to a system cannot be made until an administrator approves an Action to be taken. This approval is often in response to the result of an endpoint answer to a Relevance query. For example, a Fixlet might exist to determine whether an endpoint needs the MS11-056 patch. The endpoint processes the Relevance to that Fixlet and reports its evaluation. Even if it needs the patch, the endpoint does not act at this time, because there is no approved Action. A Console operator can now see that there are endpoints that need the MS11-056 patch and can elect to deploy (approve an Action) it to specific endpoints. It is also possible to approve Actions

so that if it becomes relevant in the future, the action is taken immediately because it is already approved.

When a report (for example, an evaluation result) is sent back to the Server, it is typically a *difference* report. Only items that changed since the prior report are reported. Only if explicitly requested does the endpoint need to send a *full report*. The full report might be sent deliberately, at the discretion of the operator, because of the age of the report or if an intermediate report is missing.

An Agent gathers new content from its parent and inserts it into an *evaluation loop*. The evaluation loop is the main process for the Agent. The Agent proceeds from Fixlet to Fixlet, assessing each Fixlet, acting, and reporting back where necessary. After the Agent reaches the last Fixlet, it starts again, all within the resource constraints (operating at around 2% CPU utilization, by default). In this manner, Tivoli Endpoint Manager provides a continuous assessment capability (and enforcement, too, if Actions are already deployed).

One of the most useful features of the Fixlet approach is *closed loop verification*. This Fixlet provides the operator with assurance that an Action succeeded or failed. The operator does not deploy an Action and assume that it succeeds. When patching an endpoint, for example, many factors can impede success. Instead of assuming success, the Agent reevaluates the original Relevance statement after the completion of an Action to determine whether it successfully fixed a problem. Before a patching Action, the Relevance returned *true*, indicating that the endpoint needed the patch. The same Relevance evaluation returns *false* after the patching Action to indicate that the patch is no longer required. If the Relevance evaluates to *true*, the patching did not work. This closed loop verification creates a closed loop from problem determination through fix to validated resolution.

We provide more detail about the various Agent processes in the next section. We list the various Agent processes and short descriptions:

- ▶ Agent registration

This process identifies the Agent to the Tivoli Endpoint Manager system, ensuring that the selected Relay knows how to reach the Agent to provide future notifications. This process also *licenses* the Agent.

- ▶ Content gathering

New content must be obtained periodically from the Server, usually through a Relay. This process is triggered by a notification from a parent Relay or on a schedule. Notifications are sent when content is changed.

- ▶ Relevance evaluation

This process is part of the evaluation loop mentioned earlier where Fixlets are evaluated.

- ▶ Action evaluation

This process is part of the evaluation loop mentioned earlier where Actions are executed.

- ▶ Results reporting

This process is part of the evaluation loop mentioned earlier where results of the Fixlet evaluation and Action execution are reported to the parent Relay.

- ▶ Relay auto selection

This process allows an Agent to select an optimal Relay.

The parent Relay of the Agent can either be selected statically (manual Relay selection) or dynamically (automatic Relay selection). If the currently selected Relay of an Agent becomes unavailable, the Agent can use another Relay. The Agent then automatically uses another Relay, providing resiliency in the Relay hierarchy.

In manual Relay selection, an operator defines a Relay for the Agent to attempt to use as its primary parent. The operator can also specify a secondary Relay. This secondary Relay is used if the Tivoli Endpoint Manager Agent fails to communicate with the primary Tivoli Endpoint Manager Relay. Furthermore, the operator can configure a list of Relays, known as *tertiary Relays*, to use if both primary and secondary Relays are unreachable.

In automatic Relay selection, Tivoli Endpoint Manager Agents can be configured to search among a list of Relays to discover the closest Relay. Automatic Relay selection follows these steps:

1. The list of Relays is examined to determine whether any Relays are in the current subnet. If they are, registration is attempted.
2. If no Relay is associated with the Agent, successive rounds of Internet Control Message Protocol (ICMP) echo requests (pings) are sent to the remaining Relays. On each round, the *time to live* (TTL) value is increased (starting at 1).
3. If an ICMP echo response is received, registration is attempted with the responder.
4. If registration fails, the process continues.
5. If multiple devices are identified, one device is randomly selected.
6. In all cases, the Tivoli Endpoint Manager Server is used as the Relay of last resort.

The automatic Relay selection mechanism can be useful to optimize the Relay selection process to allow Tivoli Endpoint Manager Agents to discover the closest Relay. For environments with many Relays, the automatic Relay selection

mechanism can result in many ICMP echo requests. There are several configurations and features available to deal with this situation, including *relay affiliation*. With Relay affiliation, the Agent does not need to send ICMP echo requests to every Relay in the organization. Instead, each Relay advertises membership to one or more *affiliation groups*. Agents are made aware of these groups and the Relays within each group. Each Agent is also given a *seek list*, which is a list of affiliation groups to search for a Relay. The Agent searches for Relays only from its seek list and not for every Relay in the organization.

Automatic Relay selection challenges: Automatic Relay selection is a beneficial tool, but there are implications. Other controls are often required when using automatic Relay selection. Read the advice on the IBM website carefully. Typically, we see controls that restrict the maximum number of hops to ping. The default is 255, but an organization must strive to keep this number under 10.

We must also control *when* to use automatic Relay selection, which can be performed by a Fixlet. The Tivoli Endpoint Manager Agent might be placed into an isolated network environment that has no access to Relays or the Tivoli Endpoint Manager Server. If endpoints are configured with a maximum TTL of 10 and seek 50 Relays, but none are available, this process generates 500 pings per Agent. If this process happens to multiple Agents, the resultant ping generation can adversely affect the network. Also, consider how *fast* the pings need to occur and if, in certain situations, manual Relay selection is preferable.

When disconnecting from the corporate environment, Agents might not be able to connect to Tivoli Endpoint Manager Relays. We often see organizations make Tivoli Endpoint Manager Relays available in a DMZ. This approach allows Agents that are disconnected from the corporate environment to still communicate with Tivoli Endpoint Manager Relays.

In networks, such as Multiple Protocol Label Switching (MPLS), which obscure the network hop count, detecting the *nearest* Relay by hop count might not result in an optimal selection. Distant Relays *might* appear to be only a few hops away. Alternatives must be sought in these situations.

There are typically no external software requirements for the Tivoli Endpoint Manager Agent except for the Agent on the Windows platform. The Agent on the Windows platform requires that at least Internet Explorer 5 is installed for certain features of the Agent-side UI. The Tivoli Endpoint Manager Agent can function on a wide variety of operating systems, including Microsoft Windows, Apple Mac OS, Linux, UNIX, and mobile operating systems. The Tivoli Endpoint Manager Agent can function across Intel x86, x64, Sparc, IBM PowerPC®, Itanium,

embedded systems, and mobile platforms. The list is maintained, and the current version is at this website:

<http://www.ibm.com/support/docview.wss?uid=swg21570458>

The Tivoli Endpoint Manager Agent can also present a local UI on certain systems, mostly Windows and Mac OS. This local UI can be used to represent the current state of the system and to provide *optional* Fixlet Actions to users, called *offers*. With these offers, the users can select from, for example, a catalog of software that they might want to install on their machines.

3.1.7 Web Reports Server

Next, we take a closer look at the Tivoli Endpoint Manager Web Reports Server and how it interacts with the Tivoli Endpoint Manager Server and Web Reports database, shown in Figure 3-9.

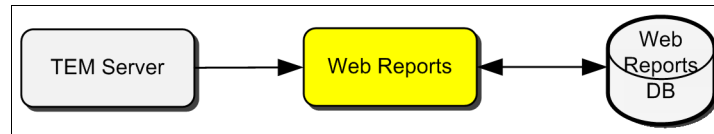


Figure 3-9 Web Reports Server component in context

Tivoli Endpoint Manager Web Reports Server is a reporting component for the Tivoli Endpoint Manager system. It provides a view over the data that is stored in the Tivoli Endpoint Manager database. The database stores information about the current state of endpoints (needed patches and compliance state) only and not historical information or prior settings. As a result, the Tivoli Endpoint Manager Web Reports Server also represents the current state⁷ of a Tivoli Endpoint Manager environment only, not a historical picture. The reports can be delivered in HTML and JavaScript, and are viewable in any standard web browser. As a web server, the Tivoli Endpoint Manager Web Reports Server can use HTTP (default) or HTTPS (suggested) to transport data to the browser. Reports can be generated on demand, scheduled, emailed, and exported to formats, such as comma-separated value (CSV), PDF, and spreadsheets.

⁷ There are minor, limited exceptions where Tivoli Endpoint Manager captures historical information. Report Archiving can be used to provide periodic *snapshots*, but the Analytics system is intended to be the solution for historical information and analysis.

Both the Web Reports Server and the Console can view the same data, but the two servers differ:

- ▶ Read only

The Console operators are intended to be able to change the system. The Console provides an operational view of the environment of the organization and the ability to respond to that environment. Not all users need the ability to change the system and can be considered *consumers of reports*. The Web Reports Server, a read-only mechanism, can provide this function.

- ▶ Aggregated view

Unlike the Console, which is intended to be operated against a single Tivoli Endpoint Manager instance, the Web Reports Server can aggregate information from multiple installations. It presents a “*single pane of glass*” view similar to the view that single installations offer.

Credentials for users of the Web Reports Server are not the same credentials that are used for a Tivoli Endpoint Manager Console user. A Web Reports Server user can still be restricted by content and endpoint visibility. Additional filtering is also available. However, filtering cannot override the base restrictions to provide visibility over an object that the user does not have permission to view.

The Web Reports Server communicates with the Tivoli Endpoint Manager Server to obtain new data. Retrieved data is stored by Tivoli Endpoint Manager Web Reports Server in memory. Storing this data in memory allows for fast and efficient data access. It can also use large amounts of memory, depending on the deployment size and other factors. Do not confuse the in-memory data of the Web Reports Server with the Web Reports Server database. The Web Reports Server database holds authentication credentials for the Web Reports Server. It is not used for persistent storage of the data that comes from the Server.

The in-memory data is divided into multiple stores that are specific to the type of data in each store, which is then managed by the session. The session is responsible for periodic updates to these stores and any dependencies between them. The stores allow for fast access and querying when *exploring* data in the Web Reports Server. You do not need to constantly interact with the Tivoli Endpoint Manager Server or Tivoli Endpoint Manager database. The stores are only updated with data that changed since its last refresh to prevent unnecessary work. Even with this data strategy, the amount of data transferred between the Server and the Web Reports Server can be large.

The Web Reports Server also provides a Web Services interface by using SOAP. The SOAP API allows external parties to access the data without the need to directly communicate with the Tivoli Endpoint Manager database. Furthermore, the SOAP API uses the language common to the Tivoli Endpoint Manager components (Relevance Language) to provide consistent access methods to

Tivoli Endpoint Manager data across all components. For more information about the APIs to interact with the system, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Customizations>

Web Reports Server database

The Web Reports Server database exists to manage Web Reports Server-specific data, such as user logins, user settings, stored reports, and saved filters. It does not act as a data warehouse for query results or hold any data extracted from the main database. Those functions are performed by the Tivoli Endpoint Manager Web Reports Server in-memory data. The Web Reports Server database is often on the same server as the main Tivoli Endpoint Manager database, because the Web Reports Server database is small. The performance implications of access to the Web Reports Server database are negligible, allowing collocation.

The Tivoli Endpoint Manager Web Reports database has the same software requirements as the main Tivoli Endpoint Manager database. Use one of the following products:

- ▶ Microsoft SQL Server 2005
- ▶ Microsoft SQL Server 2008
- ▶ Microsoft SQL Server 2008 R2

3.1.8 Analytics

Next, we look at the Tivoli Endpoint Manager Analytics component. We look at how it interacts with other Tivoli Endpoint Manager components, shown in Figure 3-10.

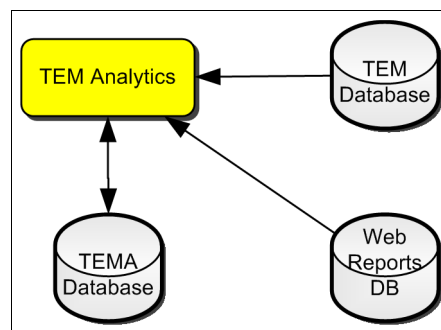


Figure 3-10 Analytics component in context

Tivoli Endpoint Manager Analytics, previously known as *Decision Support System* (DSS), is a web-based platform to deliver historical reporting. It provides analytics features to solutions delivered by the Tivoli Endpoint Manager System. Tivoli Endpoint Manager Analytics serves as a data warehouse for trend reporting of Tivoli Endpoint Manager data. The Tivoli Endpoint Manager Analytics system implements a plug-in/modular approach to deliver its functionality.

Security and Compliance Analytics (SCA) is the first plug-in/module released that uses the underlying Tivoli Endpoint Manager Analytics platform. SCA provides organizations with the ability to report on historical analysis and aggregate reporting metrics. It presently, however, does not aggregate data from multiple installations like the Web Reports Server.

Solutions that use the Analytics capabilities perform an *extract, transform, and load* (ETL) function brokered by the Analytics application. This process *extracts* data from the main database⁸ and *transforms* it into a form that can be better used for querying and data warehousing. Finally, the process inserts or *loads* the data into the Analytics database.

This process allows the historical information to be maintained over time, in a form that is conducive to reading. This process does not affect the main database, which is designed for fast writing.

Tivoli Endpoint Manager Analytics users can be defined and authenticated exclusively to the Analytics component or can be linked to users in the Web Reports Server for authentication. As with all other user interactions in the Tivoli Endpoint Manager System, Tivoli Endpoint Manager Analytics users can be restricted to certain systems and data.

Web Reports Server authentication: Analytics users can be linked with Web Reports Server users, but not if the Web Reports Server user is defined by using Active Directory authentication. Analytics users can be linked with locally defined Web Reports Server users only. When authenticating a user linked to the Web Reports Server, the Analytics application verifies the credentials of the user directly against the Web Reports Server database.

Analytics and Web Reports Servers provide different functionality. There is an overlap in their capabilities and a common goal to facilitate reporting in various formats. Because Analytics is an evolution of the Tivoli Endpoint Manager reporting capabilities, it currently exists separately from the Web Reports Server.

⁸ In Tivoli Endpoint Manager Version 8.2, this component communicates directly with the Tivoli Endpoint Manager database and is not brokered by the Tivoli Endpoint Manager Server.

The Analytics system is often deployed on the same system as its database, similar to the Tivoli Endpoint Manager Server. Tivoli Endpoint Manager Analytics uses Java technology and requires the Java Development Kit (JDK) Version 6u18 or higher.

Because the Tivoli Endpoint Manager Analytics system represents a data mining and historical data warehouse, its database size differs from the required size for the major Tivoli Endpoint Manager database. Its actual size depends on the amount of data and the retention periods required for that data.

Although Analytics can be implemented on the same physical device as the main Tivoli Endpoint Manager Server, this configuration is uncommon. We suggest this configuration for small deployments only, because of the difference in read/write optimizations of the databases, among other factors.

We described all the major components and their interactions. Next, we briefly describe two more aspects of the Tivoli Endpoint Manager system.

3.1.9 Fixlet message

Fixlet messages are not strictly a component but are obviously a part of the system. They are introduced in 2.2.5, “Key terms” on page 36 and are given a more detailed treatment in 3.2.8, “Fixlet message structure” on page 105. A Fixlet message is the construct that is exchanged in the Tivoli Endpoint Manager system. It contains the logic to enable the Agent to answer questions and take Actions.

Terminology: A *Fixlet* is a type of *Fixlet message*. An *Action* is also a type of *Fixlet message*, as is *Task* and *Baseline*, among others. Sometimes, we see the term *Fixlet* used for both a *Fixlet* and the more generic *Fixlet message*.

3.1.10 Users

Many Tivoli Endpoint Manager components have users and operators, and therefore require user authentication. All users and operators in the system can be authenticated locally by their individual components. Certain components can authenticate users to an enterprise directory. Other components can authenticate users against other components.

Today, this situation produces a mixture of authentication techniques. In all systems, it is possible to restrict the visibility of a user in terms of *content* (for example, restricting a user to patching sites only, but not compliance sites). It is also possible to restrict the visibility of a user in terms of *endpoint visibility*,

defining which computers they can see (for example, can see Marketing group only, but not Sales group). First, we describe the various users, and then we provide a comparison of users in the various Tivoli Endpoint Manager components:

▶ Site administrator

The site administrator is responsible for managing the site-level key. The site administrator is responsible for setting global options. These global options include the minimum refresh interval of the console, a high availability or Distributed Server Architecture (DSA) configuration, and masthead management.

▶ Console master operator

The master operators are the highest privileged Console operators and are typically unrestricted in their visibility within the system. They can perform all actions that a non-master operator can perform and also these functions:

- Assign rights to non-master operators
- Manage properties that are retrieved from endpoints
- Define the Agent heartbeat interval
- Create Custom Sites
- Manage site subscription
- Audit Console actions

▶ Console non-master operator

Non-master operators are typically the day-to-day users of the system, operating the areas of the Console content that they can see. Users are typically restricted in both content and computer visibility, which is not necessary, but remains a good practice. Within the restrictions placed on them by the Console master operators, they can perform these tasks:

- Deploy Actions
- Create custom content (if granted the permission by the site administrator)
- Manage computer settings
- Own Custom Sites, if permitted by the master operator

▶ Web Reports Server user

A Web Reports Server user does not have any rights to access the Console. The users can be restricted based on roles and filters in the Web Reports Server and also on the Console rights of an existing Console operator⁹. Various levels of Web Reports Server users exist that permit or deny the ability to create reports, manage filters, and administer other Web Reports Server users. The Web Reports Server users cannot change any content in

⁹ The Web Reports Server user has the same restrictions as the Console operator. Both accounts have separate authentication credentials. The credentials of the Console operator are not used.

the Tivoli Endpoint Manager Server or database. Consider them, in this regard, as *read-only*¹⁰.

► Analytics user

As with the Web Reports Server users, Analytics users cannot change content in the main Tivoli Endpoint Manager database. They cannot change content that is extracted from it. Various roles are available to restrict the capabilities of the users in the Analytics system and filters are available to restrict their visibility. Specific Analytics users also have accounts in the Web Reports Server. It is possible to link a user in the Analytics system with a Web Report System user so that the Analytics user is authenticated against the Web Report System database.

Table 3-1 summarizes the various user types and their authentication capabilities.

Table 3-1 User types and authentication

User type	Where	Authorization			
		Local	AD	LDAP	Other
Console master operator	Console	Yes	Yes	Yes	No
Console non-master operator	Console	Yes	Yes	Yes	No
Web Reports Server user	Web Reports	Yes	Yes	No	No
Tivoli Endpoint Manager Analytics user	Analytics	Yes	No	No	Yes - Web Reports

In this section, we examined the logical components of the Tivoli Endpoint Manager system. We described the context for each component and where data flows in the system. In the next section, we look at the software and services that make up these components and show the interactions between them.

¹⁰ There also exists a “read-only” role for Web Reports Server users, but this role reflects that they cannot change anything in the Web Reports Server.

3.2 Software component breakdown

In this section, we examine the software components that make up each of the platform tools. You need to understand their function so that you can better understand how the platform works.

3.2.1 Tivoli Endpoint Manager platform

Understanding the platform is key to understanding how Tivoli Endpoint Manager can deliver unmatched scalability. Understanding the platform offers the visibility and feedback needed in a secure environment. The following list includes each the platform entity and its subcomponents:

- ▶ Tivoli Endpoint Manager Server:

- Root Server:

- Gather component
 - Post results
 - Propagate
 - Agent registration
 - Data server plug-in

- BES Gather

- GatherDB

- FillDB

- ▶ Web Reports

- ▶ Console

- ▶ Tivoli Endpoint Manager Relay

- Root Server

- ▶ Tivoli Endpoint Manager Agent:

- Fixlets

- Inspectors

- Download manager

- Gather component

- ▶ Tivoli Endpoint Manager Analytics

Figure 3-11 on page 90 shows a logical representation of these software components and the interrelationships between them.

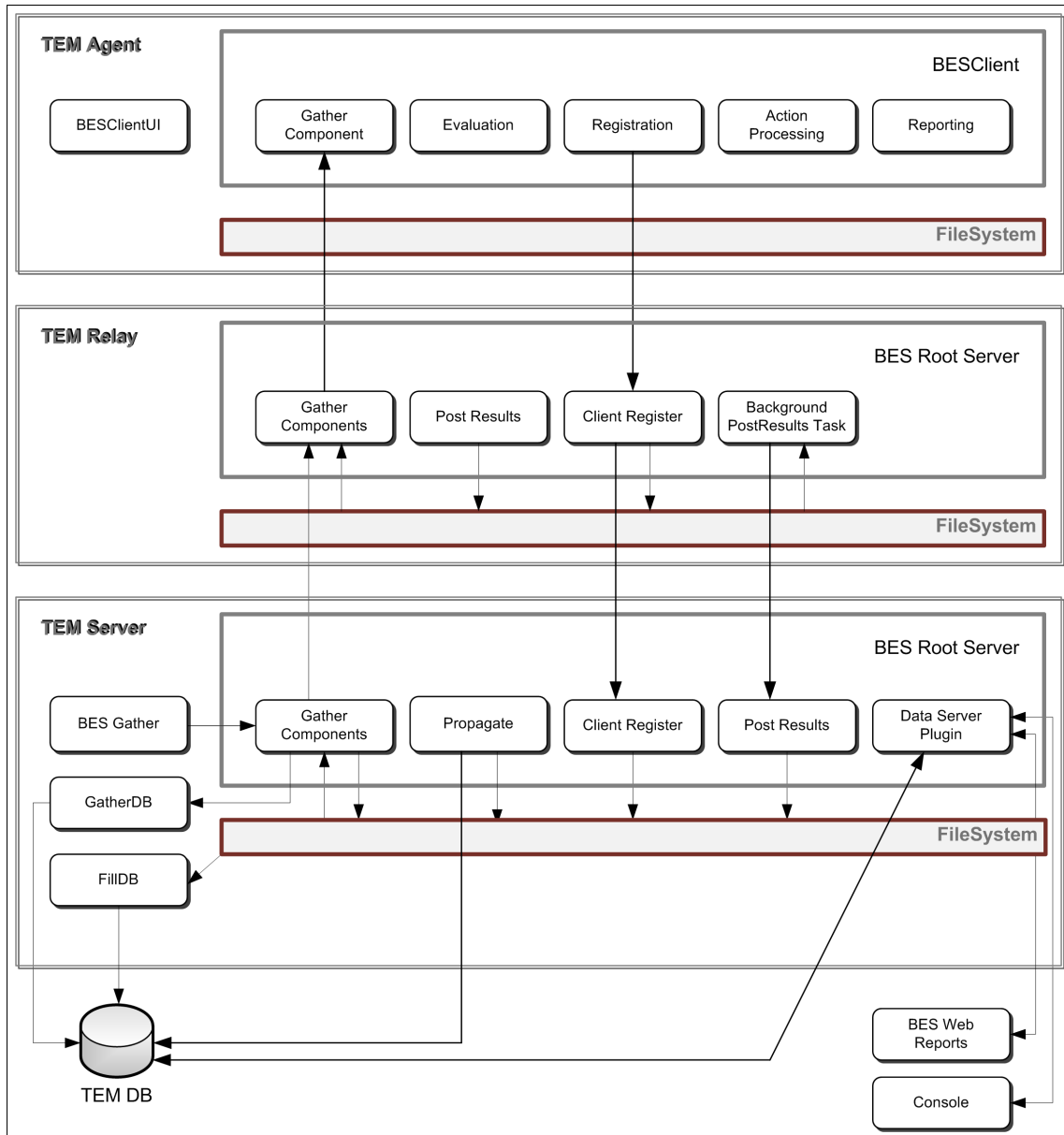


Figure 3-11 Tivoli Endpoint Manager software components

3.2.2 Tivoli Endpoint Manager Server

The Tivoli Endpoint Manager Server is the core of the platform. It is responsible for the coordination of data that interacts with the database. This data ultimately feeds information back to the user for visibility of the endpoints of the organization. Figure 3-12 displays the software components that make up the Tivoli Endpoint Manager Server.

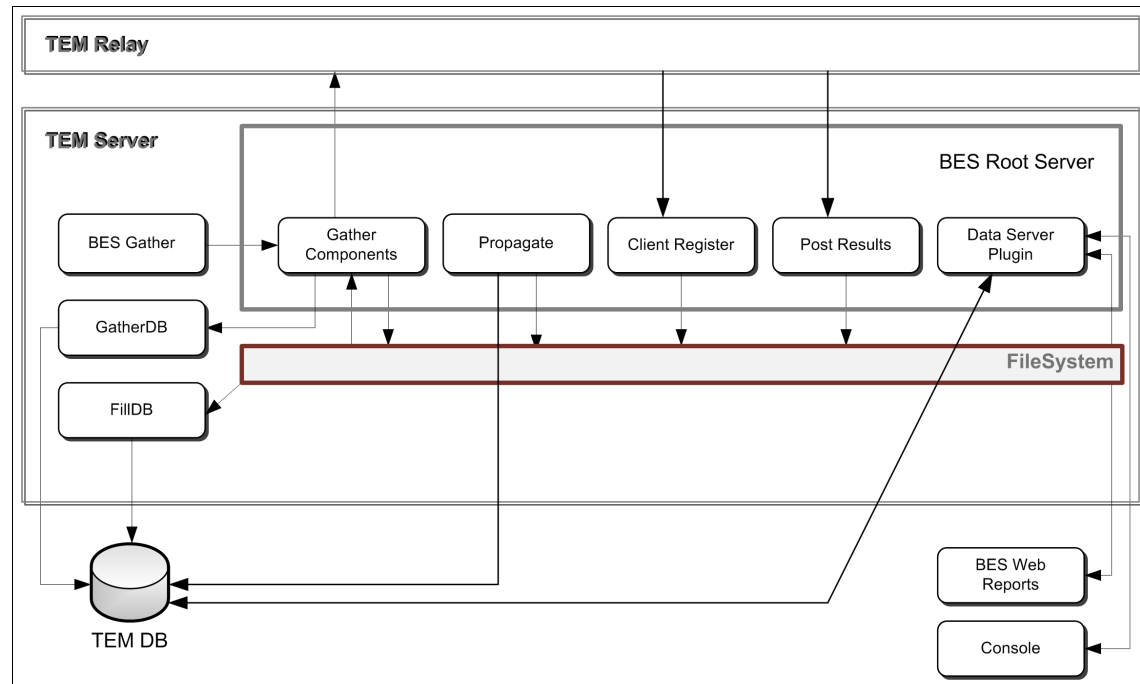


Figure 3-12 Tivoli Endpoint Manager Server software components

Root server

The Root server exists as a web server on the Tivoli Endpoint Manager Server and runs as a service. This web server exposes several subcomponents with different responsibilities. Excluding certain components, the Root server in this discussion is the same component that runs on the Relays. This aspect of the deployment is an important feature. It demonstrates a system design that is common among the Server and Relay and also resilient to faults, efficient, and easy to troubleshoot. The common Root server components ensure that Fixlets are received at the endpoint and that the results of these Fixlets show in the Console to provide the operator visibility of the infrastructure.

The Root server is responsible for all interactions with the Agents, Relays, Console, and Web Reports Server. A web server hosts standard Fixlets that are

subscribed to and downloaded from the IBM content servers and custom Fixlet Sites. These interactions occur over HTTP.

The Root server also brokers the connection to the database (*bfenterprise*) and handles requests from the Web Reports Server and Console. There is no longer a direct connection from the Web Reports Server and the Console to the database. The Root server can now enforce permissions at a granularity that is finer than at the pure database level. You no longer need to create a user account at the database level for every Console operator, reducing the workload on database administrators in the user creation process. Architecturally, the database can now have a firewall placed in-line between it and the Root server. This firewall results in less exposure to servers on the network, because only the Tivoli Endpoint Manager Server needs to maintain full access. For the users of the system, there are fewer configuration steps necessary for deployment.

Previous versions: If you use a version of Tivoli Endpoint Manager before Version 8.2, the Console, Server, and Web Reports Server all maintain ODBC connections to the database.

The Root server handles Wake on LAN (WOL) packets and calculates the most efficient method of reaching an endpoint to wake it. This calculation is achieved by knowing the relationship of the Relays to the endpoints across the environment.

Agent reports, which are handled by the Root server, flow from the endpoints to the Server. Because these reports are upstream data, they can also be encrypted by using Message Level Encryption. If this feature is used, the Root server decrypts the data.

Gather component

This component of the Root server is responsible for gathering the Fixlets and Action sites from its parent service, BESGather. This process of gathering is optimized and the same for the Tivoli Endpoint Manager Relay and Agent. The child Relays or Agents establish HTTP connections to the Gather component of the Server to collect a *fullsite* if the Relay or Agent did not gather before. In this instance, a fullsite is gathered, which is a compressed file with all of the site information. If the Relay or Agent already has a version of the site, it notifies the Server or Relay. It then receives the *diffsite*, which is a compressed file that contains only the differences between the sites. If the Agent version of the full site is outdated, it gathers the fullsite.

BESGatherMirror: The Gather service is sometimes called *BESGatherMirror*.

Post results

This component is used when an Agent decides that a Fixlet is relevant. The Agent reports to the Server through the Relay by using an HTTP POST operation. The registered ID of the computer and the relevant Fixlet are identified. This information is passed onto the database using the FillDB service and the information is then available to view in the Console. Other state changes are also reported through the Post results component. This information is passed to the database using the FillDB service and the appropriate information becomes viewable in the Console of the operator. You can also use this mechanism for posting binary files back up to the Tivoli Endpoint Manager Server, which is useful for data and Analysis.

Propagate

Propagating Actions is the method by which instructions are sent to the endpoints. First, an operator defines an Action, then the Console instructs the Data Server Plug-in (using HTTP) to insert this request into the database. The Propagate component monitors the database for changes, checks the authenticity of the content, and finally propagates the Action to the endpoints.

Agent registration

The Agent registration component is responsible for enrolling new endpoints so that Tivoli Endpoint Manager can manage them. First, the Agent installer is run on the endpoint either manually or through use of the deployment tools. Then, the masthead file instructs the endpoint how to reach the Server for registration. Each endpoint that joins a deployment is given a unique identification number so that it can be identified independently of the IP address. Information is also noted about how the endpoint communicates with the Server on an ongoing basis. For instance, if the IP address of an Agent changes, due to the common use of a Dynamic Host Configuration Protocol (DHCP) service, the Agent automatically registers the new IP address with the Agent registration component. This component populates the file `clientregistrationslist.txt`. This data flows throughout the deployment to inform the Server of all new endpoints that are registered.

If you run a Distributed Server Architecture environment, the two Servers maintain a subset of unique IDs for this issue. New machines that report into separate instances of the Server continue to receive a unique ID, avoiding the potential conflict in this scenario.

Data Server Plug-in

The Data Server Plug-in acts as the communications broker between the Console, Web Reports Server, and the database, handling secure HTTP requests. There are several benefits to this approach, including increased performance and ease of use for the administrator.

BES Gather

This service runs on the Tivoli Endpoint Manager Server and handles all Internet requests for new Fixlet content. The Gather component of the Root server communicates with this service and collects new Fixlet content from the Internet for its child Relays and Agents. If the Tivoli Endpoint Manager Server does not have access to the Internet, it is common to establish a limited user account with access to the Internet through proxy settings entered into Microsoft Internet Explorer.

The Gather service monitors changes in the Fixlet Sites to which the Server is subscribed. It downloads updates to the Server, and makes them available for the GatherDB component to populate them into the database. Then, the new content is visible for the Console operators.

BES GatherDB

The GatherDB component places Fixlets into the database so that they can be available to the operators. The results can be seen when the status message in the Console displays Gather Fixlet Site <Sitename>. The operation of gathering new sites by the BES Gather component is executed one time each hour, and all messages are logged in the GatherDB.log file.

BES FillDB

The FillDB or *Fill DataBase* service is responsible for populating the database with the Relevance results that are evaluated at the endpoints. The results of Fixlets that are processed at the endpoints can then be reviewed at the Console. The files that are sent from the Agents to the Server are first stored in a temporary folder on the file system. They are then processed by the FillDB service for data entry into the database. Under idle operation, this folder is empty. Under normal operating conditions, the BES FillDB removes files from this store and places them into the database in chunks. Any messages or errors from this process are logged in the FillDB.log file.

Important: Consider the infrastructure that supports the Tivoli Endpoint Manager deployment. Tivoli Endpoint Manager issues many write commands, which it receives from all its Agents, to insert data into the database. Deployments that use a storage area network (SAN) often slow the overall operation of Tivoli Endpoint Manager. In this environment, you see a directory with many files, which act as buffers, that contain entries to write to the database.

The Distributed Server Architecture (DSA) is an optional feature of a Tivoli Endpoint Manager deployment. It allows two instances of the Tivoli Endpoint Manager Server to be synchronized for high availability. FillDB handles the

replication of the databases (BFEnterprise) and the differences between the two instances of the Tivoli Endpoint Manager Server.

3.2.3 Tivoli Endpoint Manager Web Reports

The Web Reports feature of Tivoli Endpoint Manager is a web-based read-only view of the database, BFEnterprise. The same database is used by the Tivoli Endpoint Manager Server. The displayed reports, available from a browser, are the results of Fixlet messages that are evaluated on the endpoints. The default configuration for Web Reports is to communicate on port 80 unless the Microsoft Internet Information Service is running. In that case, Web Reports communicates on port 52312 to avoid any potential conflict.

Generating reports from the results of Fixlet messages requires communication with the database, which is brokered by the Data Server Plug-in. This component exists as part of the Root server. Web Reports submits secure HTTP requests to the Data Server Plug-in to receive the results that it needs.

Web Reports keeps a cache of information from the database to display reports to the user. The advantage of this approach is that reports can be generated quickly, which means less waiting for the user and no performance impact to the database. Web Reports uses the same caching method that the Console implements. Web Reports can be run and installed on a separate server if required.

Previous versions: As of Version 8.2, the Data Server Plug-in orchestrates communication to the database from the Web Reports Server as a secure HTTP request. Then, an ODBC connection is used to access the database.

3.2.4 Tivoli Endpoint Manager Console

The Console is the visual entry point for operators to see the results of Fixlets that are evaluated at the endpoint. The Console function can be split into several domains:

- ▶ Visualizing information
- ▶ Acting on information
- ▶ Managing users
- ▶ Hosting applications
- ▶ Categorizing relevant content split into appropriate domains
- ▶ Displaying wizards
- ▶ Displaying dashboards

Previous versions: Versions before 8.2 used the Administration tool for user management. Now, you manage users in the Console.

We install the Console by running the executable BESConsole.exe on a Windows system and connecting to the Tivoli Endpoint Manager Server. After the Console has been installed and started, the user is required to authenticate against either a Lightweight Directory Access Protocol (LDAP) or an Active Directory.

On initial start-up, the Console performs several HTTP requests to the Data Server Plug-in. In turn, the Data Server Plug-in queries the database for data that relates to Fixlets, Tasks, Actions, Analysis, and computers. After these initial steps complete, the information is stored in memory for rapid retrieval. If the Console is shut down, the information currently in memory is compressed and stored on disk. The next time that the Console starts, the Console loads the cache and requests to query the database for updates. The cache is periodically refreshed at a rate that is set in the Console preferences. During this time, the Console receives all updated information and alters the display to reflect these changes.

Endpoint results that are stored in the database are stored with a sequence number. Each time that the Console attempts a refresh, it requests the update since the last sequence number.

Outside of these requests to the database that are brokered by the Data Server Plug-in, the Console performs HTTP requests for the following reasons:

- ▶ Propagating new Actions by requesting interaction to the Propagate component of the Root server
- ▶ Prefetching files from the BESGather service
- ▶ Requesting a refresh of Agent information

Dashboards are a useful feature of the Console. They are content domain specific and display the latest results of Fixlets that are evaluated at the endpoints. You can use an API to display the contents of the Dashboards in a portal. For example, the health information of computers that provide services can be displayed on the intranet portal page of an organization to display service levels for a range of users.

The main Console window uses an embedded version of Internet Explorer. JavaScript calls are used to display the relevant information.

3.2.5 Tivoli Endpoint Manager Relay

Relays are a crucial part of a Tivoli Endpoint Manager deployment. They help achieve the unmatched scalability that Tivoli Endpoint Manager offers in terms of managed endpoints, resources, and geography.

Similarities: It is no coincidence that the Relay shares many of the same software components as the Tivoli Endpoint Manager Server. This use of common software modules helps facilitate the underlying *question and answer* concept that is implemented with Fixlets and the return of results. Components used in this way simplify the system, reduce load on the Tivoli Endpoint Manager Server, create a repeatable process that is easy to troubleshoot. In fact, a Tivoli Endpoint Manager deployment is informally known as a *secure question and answer environment*.

Relays operate efficiently. They are entirely demand-driven, which is demonstrated in the suggested system resources required for the endpoint. These resources consist of two software components for their operation. One resource is the same as the Root Server that is also on the Tivoli Endpoint Manager Server. The following components are required for the endpoint:

- ▶ Root Server
- ▶ BESGather

Figure 3-13 is a logical representation of these software components. It shows the relationship between them and the Tivoli Endpoint Manager Server and Tivoli Endpoint Manager Agent.

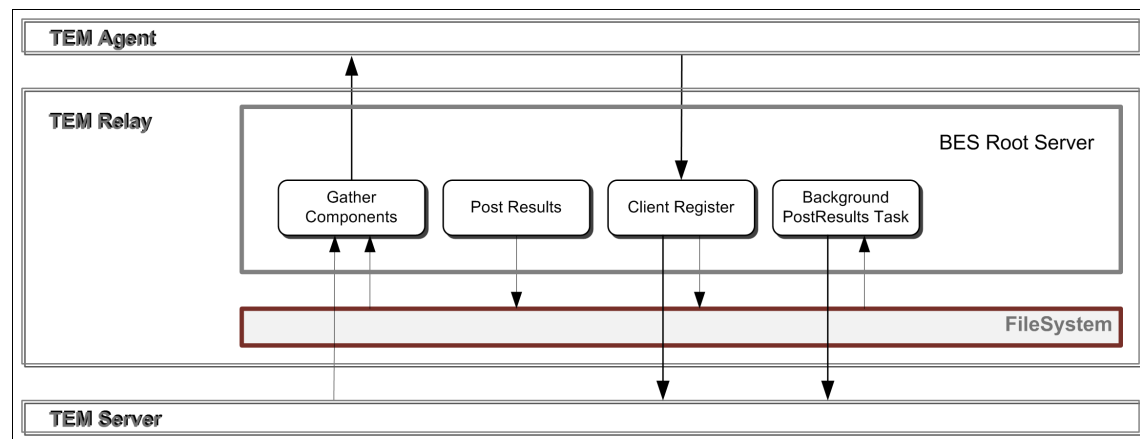


Figure 3-13 Tivoli Endpoint Manager Relay software components

In the following sections, we examine each component in more detail. We explain how they contribute to the overall operation of the Tivoli Endpoint Manager Server.

Root Server

This component is almost identical to the Root Server on the Tivoli Endpoint Manager Server with the exclusion of the Data Server Plug-in and Propagate function for propagating actions from the Console to the endpoints.

Relay Gather Components

Known also as BESGather, this component is responsible for gathering the sites and Fixlets relevant for the endpoints for which it is acting as a Relay. If an endpoint is instructed to gather new content, the message contains information about the gather site. If the Relay does not have an immediate copy of the site, it uses its parent to gather what it needs. This parent can be the Tivoli Endpoint Manager Server or the next Relay on the most efficient route back to the Tivoli Endpoint Manager Server.

Relay Post Results

Think of the posting of results as the answers to Fixlets (questions) that are gathered by the endpoint. This software component works together with the file space allocated at the Relay to buffer results from Agents or child Relays before these results are transferred to the Server.

The same process at the Tivoli Endpoint Manager Server also writes the incoming results to the file space, but in this instance the FillDB feature posts these results directly to the database.

Relay Background Post Results Tasks

This background process works with the same file system space, also known as the *port results buffer*. It batches, compresses, and sends the results from endpoints to the parent Relay or the ultimate parent, the Tivoli Endpoint Manager Server.

Relay Client Register

The registration process is the same as on the Tivoli Endpoint Manager Server. Each Relay forwards registration connections to the Tivoli Endpoint Manager Server. Each Relay notes the IP address of the endpoint and registers a unique ID for the endpoint. This registration information is written to a text file on the Relay and passed as updated information to its parent.

BESGather

This component runs as a service on the system that hosts the Relay. It is identical to the BESGather on the Root Server at the Tivoli Endpoint Manager Server. The BESGather component gathers new Fixlet content from a parent only when instructed by the endpoint or child Relay (demand-driven method of Relays).

3.2.6 Tivoli Endpoint Manager Agent

Tivoli Endpoint Manager requires the installation of a single software agent on each desktop, mobile computer, mobile device, and server that you want the Tivoli Endpoint Manager Server to manage. The role of the agent role is to continually assess Fixlet messages and provide answers to questions if the question asked is relevant.

In the context of the platform, the Tivoli Endpoint Manager Agent is a single lightweight software package installed at each endpoint. It communicates with the Server, eventually through the Relay, to provide the Tivoli Endpoint Manager Server with feedback that the actions on the endpoint are complete. Collectively, these components make up the Tivoli Endpoint Manager platform.

The Tivoli Endpoint Manager Agent can assess the state of the endpoint against a policy and return the endpoint configuration to the wanted state. This configuration can enable the endpoint to comply with external and internal regulations. These actions occur without instructions from the Tivoli Endpoint Manager Server. The operators at the Console are responsible for shaping policies and deciding which actions to apply across the infrastructure.

The Agent consists of two major components:

- ▶ ClientUI
- ▶ BESClient

Figure 3-14 on page 100 is a logical representation of the software components and shows the relationship between them and the Tivoli Endpoint Manager Relay.

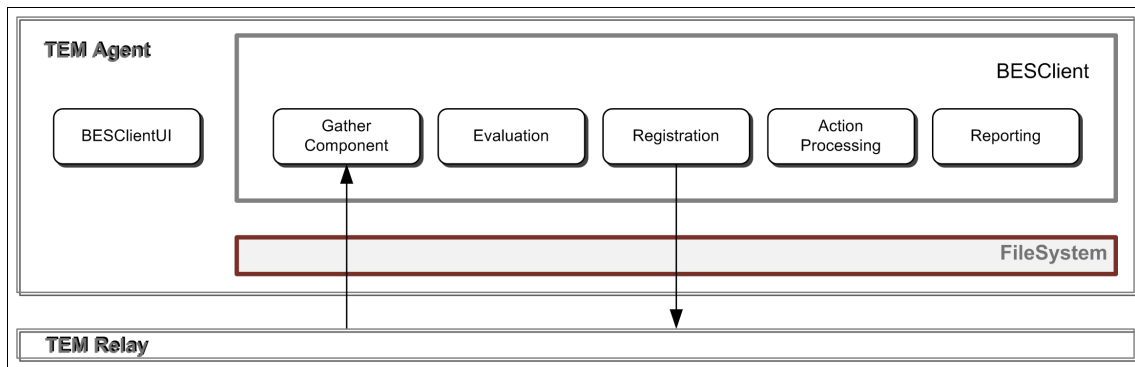


Figure 3-14 Tivoli Endpoint Manager Agent software components

BESClient

The BESClient hosts the major functional components of the Agent.

Registration

After the installation of the Agent completes, the first Agent task is to register with the Tivoli Endpoint Manager Server. The Agent knows how to contact the Tivoli Endpoint Manager Server because the information is in the related masthead file that accompanies the executable.

Microsoft Windows information: If you install the Agent on a Microsoft Windows operating system, the installer places the masthead file into the registry at this location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\BigFix\EnterpriseClient\GlobalOptions
```

The Agent sends and receives information by communicating with the registration component of the Root Server installed on any parent Relays on the path to the Tivoli Endpoint Manager Server. The Agent notifies the Tivoli Endpoint Manager Server about the IP address that it is assigned along with receiving a unique computer ID and a license number. Duplicate ID detection algorithms avoid conflicting IDs.

Gather Component

The Gather Component is responsible for collecting Fixlets and action sites in readiness to receive Actions from the Tivoli Endpoint Manager Server operators. The action site contains configuration information for the Agent that includes the Fixlet Sites to which to subscribe, the latest version of the masthead file, and a list of applicable Relays. Agents gather a site when they receive a UDP Gather message from the parent Relay or once every day as a default. After this UDP

message is received, the Agent establishes an HTTP connection to the parent Relay of the Agent and connects to the Gather Component. The Gather Component is the exposed component of the web server. The Gather Component is also responsible for authenticating the content and actions that it pulls from its parent Relays.

Evaluation

The Tivoli Endpoint Manager Agent is designed to process the Fixlets that it receives and to be agnostic to the content. Fixlet technology is at the core of the Tivoli Endpoint Manager platform, and it is made up of the language known as Relevance. Relevance is a declarative language, it is not procedural. It is human readable, extensible, and object-oriented. The use of *inspectors* allows the language to be extensible across multiple platforms.

The Tivoli Endpoint Manager Agent operates in a continuous loop known as the *evaluation loop*. During that time, the Agent processes a list of all the relevant Fixlet messages. It examines each site along with the applicable actions and reports the results back to a parent Relay or directly to the Tivoli Endpoint Manager Server. When the Agent finishes evaluating all Fixlet messages, it starts all over again.

Duration: The time taken for an Agent to finish its evaluation loop depends on many factors. Those factors include the number of Fixlets and the number of actions that it is required to evaluate.

The evaluation loop can be described in the following linear fashion:

1. Auto-select the Relay (if the Auto-select feature is used)
2. Register
3. Gather new Fixlets and Actions
4. Evaluate all Actions
5. Evaluate all Fixlets
6. Post the results
7. End and start over again

Action Processing

This part of the BESClient is responsible for orchestrating how Actions are evaluated and implemented on the Agent. For instance, if a patch requires a restart during the process, this component ensures that the Agent starts at the correct place to continue processing the Action after the restart. This component might also request the evaluation of specific configurable settings on the system before the installation continues. Then, this component feeds this evaluation back to the Tivoli Endpoint Manager Server to ensure a closed loop visibility of the state of the Action. When an Action is propagated and received by the Agent, it

has the potential for granular controls that govern when and how to process the Action. All of these actions are managed by the Action Processing component.

Reporting

Reports that contain the results of Fixlets that are evaluated on the Agent are posted to the Tivoli Endpoint Manager Server. The Agent also tracks changes to configuration items and resamples Fixlets to check whether anything becomes relevant in recurring loops.

BESClientUI

The BESClientUI implements the user interface components on the Agent.

3.2.7 Tivoli Endpoint Manager Analytics

Tivoli Endpoint Manager Analytics can be considered *monolithic* when comparing it to other components. Figure 3-15 contains a graphical representation of the Analytics components.

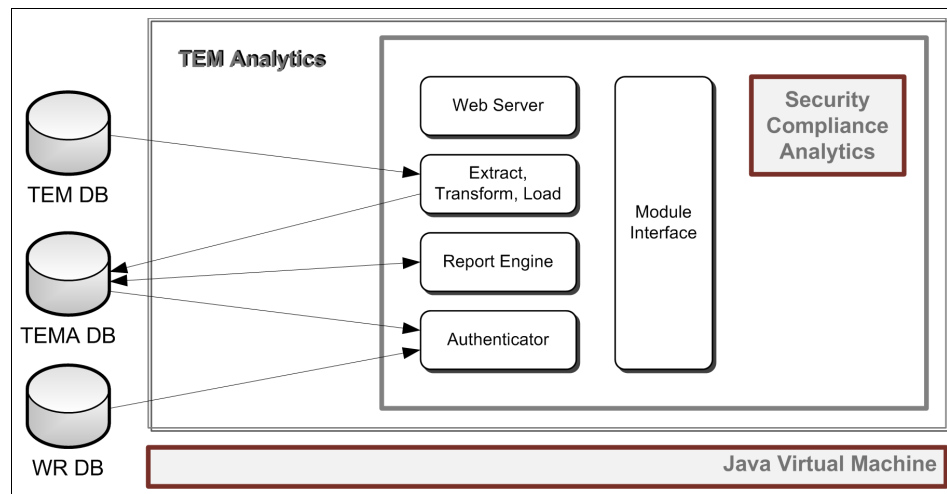


Figure 3-15 Tivoli Endpoint Manager Analytics software components

There are three database systems depicted in the diagram:

- ▶ Tivoli Endpoint Manager database (TEM DB)
- ▶ Web Reports Server database (WR DB)
- ▶ Tivoli Endpoint Manager Analytics database (TEMA DB)

Java virtual machine

The Tivoli Endpoint Manager Analytics platform uses Java technology as a base environment. Due to technical requirements, you must use the Java Development Kit (JDK) in versions newer than 6u18 to successfully use Tivoli Endpoint Manager Analytics. The JDK is the prerequisite for the Tivoli Endpoint Manager Analytics installation. It is the responsibility of the administrator to set up the Java virtual machine (JVM) environment before installing Tivoli Endpoint Manager Analytics.

Installing JVM: Although the JDK installation is not a part of the solution, Tivoli Endpoint Manager offers functionality that allows an administrator to perform this task directly from the Console. A tool called *Windows Software Distribution Wizard*, available at the BES Support Fixlet Site, can help automate the process. For more information, see this website:

<http://support.bigfix.com/cgi-bin/kbdirect.pl?id=1669>

Tivoli Endpoint Manager Analytics framework

From the operating system view, this component acts as a single Windows service called *Tivoli Endpoint Manager Analytics*. That single process is responsible for all Tivoli Endpoint Manager Analytics functionality.

Web server

Because Tivoli Endpoint Manager Analytics is delivered as a Java solution, it includes an HTTP server to provide a web user interface. The administrator must use this web user interface to configure the component and interact with the application.

Authenticator

As a web application, Tivoli Endpoint Manager Analytics offers controlled access to its resources. Only authenticated and authorized users can execute and view reports. The authentication subcomponent is responsible to verify the identity and grant the appropriate level of access. Users can be either defined directly in Tivoli Endpoint Manager Analytics or can be referenced to the Tivoli Endpoint Manager Web Reports Server, which is a one-way reference. The Tivoli Endpoint Manager Web Reports Server is unable to use users defined in the Tivoli Endpoint Manager Analytics database.

Secure login: The default installation of Tivoli Endpoint Manager Analytics does not provide Secure Socket Layer (SSL) security connections. For SSL configuration details, see this website:

<http://support.bigfix.com/cgi-bin/kbdirect.pl?id=1783>

Extract, transform, and load processing

The extract, transform, and load (ETL) process is responsible for transferring data between the Tivoli Endpoint Manager database and the Tivoli Endpoint Manager Analytics database. Those two databases have multiple differences. The first database is optimized for fast writing, because a single Tivoli Endpoint Manager Server can handle up to 250,000 endpoints. There is no need to read data quickly because reading data quickly is the responsibility of reporting systems, such as Tivoli Endpoint Manager Web Reports. The Tivoli Endpoint Manager Analytics database was designed as a data warehouse solution. Thus, it has a much more complicated structure, which makes it easier to process large amounts of data in a short time.

The ETL process is responsible for searching the Tivoli Endpoint Manager database for all artifacts that are required to be imported into the Tivoli Endpoint Manager Analytics database. Environment information, such as computers, groups, users, data sources, and correlated security checklist contents, such as the results from Fixlets, Analysis, and Exceptions are included. That part of processing is called *extract*.

After the information is read, ETL converts the data into the Tivoli Endpoint Manager Analytics database formats. That step is called *transform*.

The final action performed is the *load* process in which converted information is stored in the database for reporting. The ETL process is executed in a predefined schedule. For more information about the ETL configuration, see Chapter 4, “IT endpoint security and compliance solution design” on page 125.

Reporting engine with historical data processing

After the data is available, you use the main reporting engine to create reports and manage the data. By following user permissions, the main reporting engine can display any of the reports in 2.3.4, “Security Compliance Analytics” on page 56. The main reporting engine can also be tailored by all defined computer groups, scopes, exceptions, and other report-specific conditions.

Module interface

The Tivoli Endpoint Manager Analytics component is designed as a universal platform. The plug-in interface enables future subcomponents to use the solution capabilities. Tivoli Endpoint Manager for Security and Compliance Analytics is the first module that is able to use the Tivoli Endpoint Manager Analytics platform. Because this module is the first to set up, the installation package contains both the framework and the Security and Compliance Analytics module.

Installation: You can initiate the Tivoli Endpoint Manager Analytics installation process from Tivoli Endpoint Manager Console. A Tivoli Endpoint Manager Agent *must* be available on the Tivoli Endpoint Manager Analytics system. In the security configuration domain, a dashboard called Security and Compliance Analytics lists all workstations with compatible operating systems. From that dashboard, you can run the installation task for Tivoli Endpoint Manager Analytics.

3.2.8 Fixlet message structure

Each Fixlet has a specific structure that defines how the Fixlet is processed when gathered to the endpoint. Figure 3-16 shows the structure of a Fixlet. We describe the terms next.



Figure 3-16 Structure of a Fixlet

Each of the components in a Fixlet are shown in the Console to the operator:

- Properties

Displayed in the Console are various properties of the Fixlet, such as its unique ID, Common Vulnerability and Exposures (CVE) reference, file size, and platform applicability. This information is a high-level view of the Fixlet.

- ▶ Relevance

Relevance clauses are written into this part in sections to make the Relevance easier to read. Statements are often separated, because each statement must return the value of true for the next action to run.

- ▶ Action

Based on the endpoint returning true for all Relevance clauses that it evaluates, the Action script is then run, if needed, to implement the remediation. An Action script is another proprietary language that is used to execute the required Actions at the endpoint. The goal for the Action script is to ensure that the next time that the Relevance evaluates the clauses, they all return false.

- ▶ Comments

Operators can insert comments for Fixlets.

Process: After the Action finishes processing, the Relevance clauses are evaluated again. If they all return false, the action is successful and the Fixlet completes what it intended to remediate.

Exporting a Fixlet and examining its content reveals additional metadata that other components of the Tivoli Endpoint Manager system use.

3.3 Network communications and usage

Next, we identify the network flows between the various identified components, and entities external to the components, such as users, identified in 3.1, “Logical component overview” on page 64. Later sections help to provide suggestions about security zones. This section enables architects and solution designers to understand the implications of placing specific components in various network segments. Figure 3-17 on page 107 illustrates the components and their network interactions, and the limitations of the components managed by the organization.

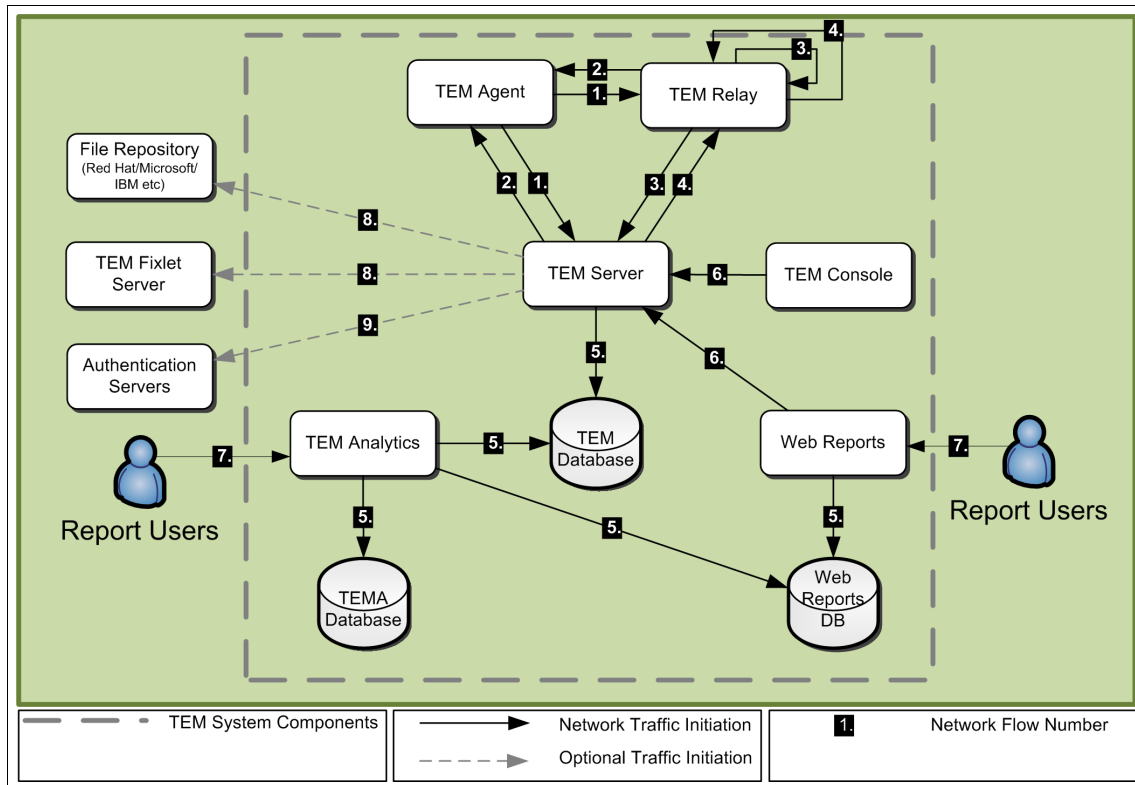


Figure 3-17 Component network flows overview

The diagram shows the major network flows between components in the Tivoli Endpoint Manager system. Each network flow is marked with a flow number that is described in Table 3-2 on page 108. Table 3-2 on page 108 provides the port and protocol information. We explain each flow and the volume and frequency of the traffic that uses that port. We represent this traffic in terms of “High”, “Medium”, and “Low”. The definition of each term changes with the size of the organization and is subjective but they serve as reference points relative to each other. For additional material, see the IBM Wiki pages:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli%20Endpoint%20Manager/page/Network%20Traffic%20Guide>

This section helps you to place a component in a particular network zone and understand the expected traffic between it and other components.

3.3.1 Intercomponent traffic

The Tivoli Endpoint Manager platform requires few ports. Interactions among the Agent, Relay, and Server connect by using a single port, 52311, and Internet Control Message Protocol (ICMP). Database communications all use a single port, TCP 1433. Only minor changes are required for the infrastructure of the organization to accommodate Tivoli Endpoint Manager. Fewer changes to the security configuration of the network enable organizations to maintain a secure stance. Table 3-2 provides a breakdown of each numbered flow identified in Figure 3-17 on page 107 and identifies the major uses of that flow.

Table 3-2 Traffic flows

Number	From	To	Protocol/port	Description
1	Agent	Relay Server	TCP:52311	Gather Download Post Report Register Primarily, Agent to Relay, so that Agent to Server can be avoided
	Agent	Relay Server	ICMP Echo Request	Relay Discovery Relay Distance Primarily, Agent to Relay, so that Agent to Server can be avoided
2	Relay Server	Agent	UDP:52311	Notification
3	Relay	Relay Server	TCP:52311	Gather Download Post Report Register
4	Relay Server	Relay	TCP:52311	Notification
5	Server Analytics	TEM DB	TCP:1433	SQL
	Analytics	WRS DB TEMA DB	TCP:1433	SQL
	TEM WRS	WRS DB	TCP:1433	SQL

Number	From	To	Protocol/port	Description
6	Console TEM WRS	Server	TCP:52311	Login
	Console TEM WRS	Server	TCP:52383	SSL Login TEM Query
7	Report Users	TEMA TEM WRS	TCP:80 TCP:443	Web browser
8* (external to the Tivoli Endpoint Manager system)	Server	Repositories	TCP:80 TCP:443 TCP:21	Downloads
9* (external to the Tivoli Endpoint Manager system)	Server	Directories	Various	Authentication

Table 3-2 on page 108 and Figure 3-17 on page 107 can be used together to identify the Tivoli Endpoint Manager traffic that occurs at a certain point in the organizational network. Flows 8 and 9 are external to the Tivoli Endpoint Manager system. Next, we look at the individual flows to describe them in more detail.

Traffic flow 1: Agent to parent Relay/Server

Traffic flow 1 (Figure 3-18) is the connection that is initiated from the Agent to the Relay or Server.

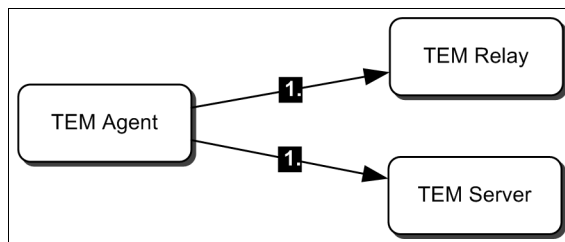


Figure 3-18 Traffic flow 1 in context

Scenarios described in the next sections show that you can and, in certain cases, might want to prevent the Agents from communicating directly with the Server by forcing them to use Relays exclusively. In these cases, there is no flow between the Server and the Agent except through a Relay.

The Agent uses ICMP Echo Requests (ping) for the following tasks:

► Determine Relay distance

An Agent identifies its distance to a Relay, which is determined by using ICMP. This task is only performed for manual Relay selection, because automatic Relay selection already determined the Relay distance.

An ICMP packet varies in size by platform. If, on average, 60 bytes (Ethernet + IP + ICMP) is a valid assumption, the amount of traffic required to determine distance to a Relay is 60 bytes x time to live (TTL) hops to the Relay. This amount is considered low:

- Volume: Low
- Frequency: Low

► Automatic Relay discovery

To discover responding Relays, automatic Relay discovery uses an ICMP Echo Request packet with incrementing TTLs to find the closest Relays. By incrementing the TTL successively, responses to the current TTL must be closer than those responses that are not received. The TTL for this round of packets causes the ping to not reach the Relay. This search is a breadth-first search in that all Relays within the current affiliation group that the Agent is seeking are tested at TTL1, then TTL2, and so on. A depth-first search does not provide the Agent with the optimally close Relay.

For automatic Relay discovery, the number of packets sent is a function of the distance to the Relays and the number of Relays for which we are searching. After we successfully register through a particular Relay, no more ICMP packets are required. So for n Relays at distance d , the number of packets sent is $n \times d$. The number of bytes is equally $n \times d \times 60$. Automatic Relay selection is triggered when the Agent IP Address changes as well as at a schedule interval:

- Volume: Low
- Frequency: Low-Medium

Warning: The frequency can increase with Agents that change networks often. The frequency can increase with large affiliation groups and maximum TTL. The worst case scenario results in this formula for the number of packets: $n \times \text{MAX TTL}$. The worst case is realized when no Relay responds. So, for an affiliation group of 50 Relays and MAX TTL of 10, the worst case scenario is to exhibit 500 ICMP packets for each endpoint. This scenario is probably not too bad for a few endpoints, but it might be a problem for many endpoints. This scenario is a clear reason for controlling Maximum TTL and the affiliation group sizes.

The Agent uses TCP port 52311 to connect to the Server or Relay to perform the following tasks:

► Gather

Gather is used to obtain new content, either in response to the User Datagram Protocol (UDP) notification described in flow 2 or at scheduled intervals, based on a polling mechanism. In this exchange, the Agent obtains the latest version of the content. By identifying the current content version of the Agent and the current content version of the Server or Relay, the Agent can receive only the differences in site content rather than a complete download.

Gather occurs once a day or in response to notifications from a parent. Only compressed differences are transferred, but the frequency can be high if the operators change much. As a result, these operations are spread throughout the period when the operators perform their tasks, not all at one time. The actual size varies. Experience shows that the size is 1 - 3 KB for each gather. Although for certain organizations and for custom content, this size can potentially be larger:

- Volume: Low
- Frequency: Medium-High

► Download

When performing validly signed Actions, the Agent might encounter one of several instructions that require it to download software. Unless otherwise directed, these downloads always occur from the Relay or Server, independently of the URL used. For example, even if the url for a patch is <http://www.microsoft.com/download/somepatch.exe>, somepatch.exe is requested from the Relay or Server. This request is an HTTP GET request.

Download traffic tends to be low frequency, because downloads occur when an Action includes a download command. Download commands are common in patch content, but less so in compliance content. The frequency might be high for a particular day, for example, “Patch Tuesday” or the days after that

day. However, when examined over a longer period, we expect to see fewer downloads. The volume, however, can be large and relates to the size of the patch that is downloaded. There is little overhead. Patches can range from a few KBs to tens of MBs. Patching Microsoft Service packs can take hundreds of MBs. If the endpoint does not have content to download, the volume and frequency are 0:

- Volume: High
- Frequency: Low-Medium

► Post Reports

At periodic intervals, the Agent sends results to the Server through the Relay (or directly if the Server is the parent of the Agent). This post is an HTTP Post.

Post Reports contain the results of an endpoint evaluation of Fixlets and Actions. These results are the differences from the last report. In this manner, if there are no changes in content, there is no report. Periodically, the endpoint still posts a report as a “Heartbeat” at a configurable interval (5 minutes, by default). The size of a post can vary but experience shows posts between 1 - 3 KB on average. The frequency depends on the rate of change of the data that is inspected on the endpoint. Questions, such as “free space of drive c:”, can change frequently. Questions, such as “minimum password length” change infrequently:

- Volume: Low
- Frequency: Medium-High

► Registration

An HTTP GET is sent to the Relay. The Relay forwards the request to the Server. If the Server is unavailable, the registration cannot be serviced.

Registration consists of an HTTP GET request and an associated response. The request is around 1 KB and a successful response is around 700 KB. The registration process runs automatically based on “triggers”, such as changes to the network interface and IP address. Additionally, a periodic registration is performed at configurable intervals, defaulting to six hours. If the Server rejects the Agent, registration is to a different Relay:

- Volume: Low
- Frequency: Low

SSL: It is possible to configure an Agent and a Relay to communicate by using HTTPS. Avoid this configuration if possible, because it requires additional resources. Consider that content is digitally signed, and reports can be encrypted with MLE. This combination commonly meets the security requirements without enabling HTTPS on the Relays, although not in all cases.

Traffic flow 2: Parent Relay/Server to Agent

This traffic flow (Figure 3-19) identifies traffic that is sent from a Relay or Server to the Agent.

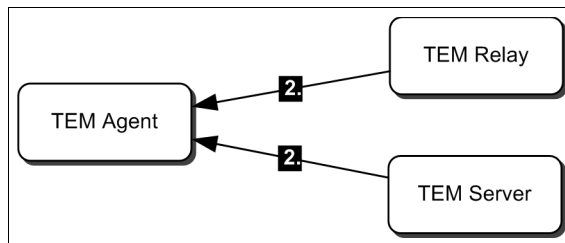


Figure 3-19 Traffic flow 2 in context

As with flow 1, it is possible to prevent the Server to Agent flow by forcing the Agents to communicate exclusively with Relays. This approach allows the Tivoli Endpoint Manager Server resources to be fully used for its other tasks while Relays deal with the Agent communications.

The Relay or Server uses UDP port 52311 for the following tasks:

- Notify the Agent of content updates

When the content is changed on the Server, the Agents are notified to ensure that every system has the latest copy. Typically, the Agent initiates flow 1. No UDP response is given by the Agent to this UDP notification. If an Agent does not receive the notification (blocked by a firewall or the Agent is offline), the content is refreshed by its own polling mechanism. And, the Agent performs a check for new content, if it is restarted, for example.

Additional settings are available to configure the timing for the polling mechanism from the six hour default. Typically, we see a Relay that sends one UDP message for each Agent. However, the Relay can send up to three UDP messages if the Agent does not initiate the gather that is identified in flow 1.

The size of a UDP packet “on the wire” can potentially vary slightly depending on the options in the lower-level IP Protocol and the media frame, for example, Ethernet. At the UDP layer, the packet contains 21 bytes of data. The IP packet is then approximately 49 bytes, resulting in approximately 63 bytes on the wire.

The frequency of the notification is purely a function of the number and frequency of changes to the Tivoli Endpoint Manager content. Typically, content does not change frequently, but changes really depend on the uses of Tivoli Endpoint Manager. Patch content typically changes once or twice a month. Compliance content might change once a quarter.

Changes also include when an operator takes an Action, which also can vary.

- Volume: Low
- Frequency: Low-Medium, although high if many operators change content frequently

Traffic flow 3: Subordinate Relay to parent Relay/Server

Flow 3 (Figure 3-20) identifies traffic that is initiated by the Relay to its parent, which can be the Tivoli Endpoint Manager Server or another Relay.

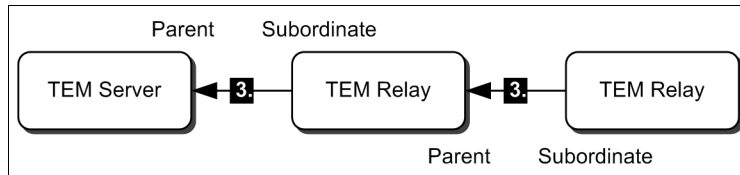


Figure 3-20 Traffic flow 3 in context

The initiation of this flow is demand-driven. An Agent that requests a download that is not present in the Relay cache causes the Relay to request that file from its parent. Many Agents can request a file from the Relay, but the Relay does not generate a request to its parent for each of these requests. The request to the parent occurs one time. This way, a Relay is used to reduce the amount of traffic effectively over a particular network link.

The Relay uses TCP port 52311 for the following tasks:

► Gather

In response to being notified by the parent that there is new content, a subordinate Relay connects to the Server to download the compressed differences in site content. The notification is documented in Flow 4.

The gather volume/size is about the same as the gather volume/size for an Agent. The same is true for the frequency, but there is only one gather executed from the Relay to the parent. There is one for each Agent to the Relay (Flow 1), so all the caveats apply from that section:

- Volume: Low
- Frequency: Medium-High

► Download

When requested by an Agent, the Relay attempts to deliver content from its cache. If the cache does not contain the requested file, the Relay requests the file from its parent, which is this flow. The mechanism is the same as with the Agent, identified in flow 1, which is an HTTP GET request.

As with Agents, download traffic tends to be low in frequency, because downloads occur only when an action includes a download command. For a Relay, the frequency is even less, because the Relay downloads a file only if it

is not already in the cache. If 1,000 Agents try to download a file from the Relay, the Relay only needs to download this file one time. The size of an individual download is the same as the size for the Agent:

- Volume: High
- Frequency: Low

► Post Reports

A Relay stores Agent reports on the Relay file system, allowing them to accumulate. When either a timer expires (defaults to 3 seconds) or the report file count limit is exceeded (defaults to 500 reports), the reports are compressed and sent to the parent of the Relay.

Post Reports are aggregated on the Relay file system. The size therefore depends on the size of reports from the Agents, but the size also depends on the frequency of the Agent reports. Furthermore, the size is affected by the number of Agents that uses the Relay and where the Relay is in the Relay hierarchy. Lower Relays in the hierarchy have less volume but potentially higher frequency than those Relays near the top of the hierarchy that transfer larger aggregations with potentially less frequency:

- Volume: Medium
- Frequency: Medium

► Registration

An HTTP GET is sent to the Relay by the Agent for registration and the Relay forwards the request to the Server. If the Server is unavailable, the registration cannot be serviced.

Registration consists of an HTTP GET Request from an Agent and an associated response from the Relay. The Relay, however, must forward the request to its parent. The request is around 500 bytes and a successful response is around 700 KB (there are situations where these sizes can be larger). These requests are not aggregated in any way by the Relay, so each Relay forwards each request. For this reason, the frequency for a Relay to Relay or Relay to Server “forwarded” registration is higher:

- Volume: Low
- Frequency: Medium-High

► Connectivity Test

Periodically, a Relay tests connectivity to the parent with an HTTP GET.

The connectivity test occurs periodically, every 10 minutes if successful, but at 1-minute intervals if the test failed, by default. The test is a small TCP-based network request and response. Typically, in a failed test, the TCP handshake fails to establish so there is little overhead in this situation:

- Volume: Low

- Frequency: Low

Network flow 4: Parent Relay/Server to subordinate Relay

Flow 4 is the reverse of flow 3. Now, we describe a parent that initiates traffic to a child or subordinate Relay (Figure 3-21).

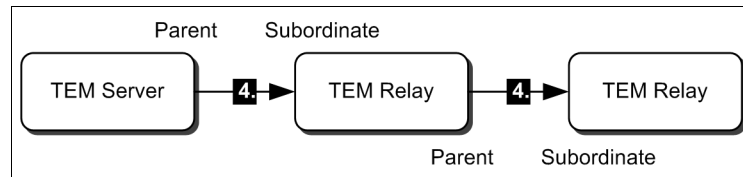


Figure 3-21 Network flow 4 in context

It is probably no surprise that the notifications from the parent Relay or Server to a subordinate Relay are also performed over TCP port 52311. Therefore, all Relay-Relay and Server-Relay communication is TCP only, no UDP. UDP is only used in the communication from a Relay to an Agent. Unlike a notification to an Agent, which goes unanswered, the TCP notification receives a response from the subordinate Relay. Notifications can be sent from a parent to request the following information:

- ▶ Notification of new content

This notification causes the Relay to gather the new content and then notify subordinates or Agents to gather the new content.

The notification of a new content package to a Relay is larger than the notification sent by the Relay to an Agent, but it is still small. The overall number of notifications to a subordinate Relay is fewer than the number of notifications that is sent to an Agent. A notification and associated response is around 500 - 600 bytes, plus the TCP overhead. The new content notification is expected to be low-medium in frequency as identified in the explanation of flow 2:

- Volume: Low
- Frequency: Low-Medium

- ▶ Notification of Agent requests

If the Server needs the Agent to send a full report, or to reissue a report that the Server missed (received an out-of sequence report), a notification to resend can be sent.

Agent requests, too, are expected to be low frequency; however, these requests can be seen in higher frequencies under certain circumstances:

- Volume: Low
- Frequency: Low

Network flow 5: Components to database

This flow (Figure 3-22) shows components that query databases, such as the main Tivoli Endpoint Manager database, the Analytics database, or the Web Reports Server database.

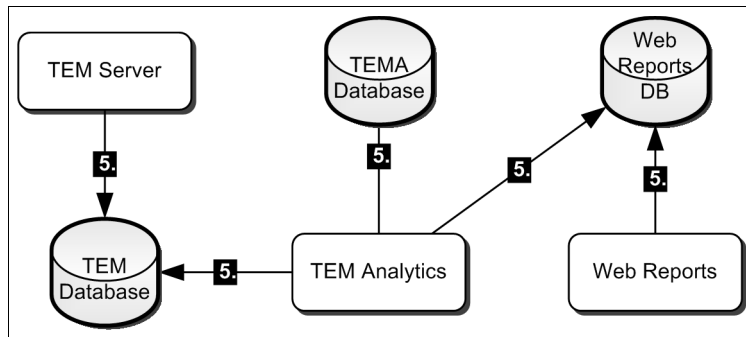


Figure 3-22 Network flow 5 in context

If an organization wants to perform custom queries against these databases, this flow also applies. When the database and the system querying the database are on the same server, these flows are often not seen. They are still operating over TCP/IP, and they use the loopback interface of the system. The following communication flows can occur:

- ▶ Tivoli Endpoint Manager Server to the main database

The Tivoli Endpoint Manager Server to Tivoli Endpoint Manager database flow occurs most frequently. The data that is retrieved typically consists of the changes since the last retrieval. Even so, the volume of data can be large. This data can be data for the Console, inserted data received from Relays and Agents, data gathered from external Fixlet Servers, or queries from the Web Reports Server:

- Volume: High
- Frequency: High

- ▶ Web Reports Server to the Web Report database

The Web Reports Server to Web Report database flow is used to authenticate locally defined Web Reports Server users, record their filters and settings, and provide general persistence to the Web Reports Server. It is not used to store queried data from the Tivoli Endpoint Manager Server, which is performed in memory:

- Volume: Low
- Frequency: Low

- ▶ Analytics system to the Analytics database

The Analytics system to Analytics database flow allows for authentication, persistence, and data warehousing to support the reporting and analysis functions. Data is periodically extracted from the main Tivoli Endpoint Manager database and loaded (after transformation) into the Analytics database, which occurs on a defined schedule. The communication is demand-driven by a user of the Analytics system that interacts with the Analytics reporting interface. The Analytics database is often on the same system as the Analytics application:

- Volume: High
- Frequency: Medium-High

► Analytics to Web Reports Server database

The Analytics to the Web Reports Server database flow is used to support the authentication of Analytics users that are already defined in the Web Reports Server database. The query is small and infrequent:

- Volume: Low
- Frequency: Low

► Analytics to the main Tivoli Endpoint Manager database

The Analytics to the main database flow allows the Analytics system to obtain data about endpoints that it analyzes and retains for historical reference and trending. The frequency of this flow is based on a schedule and configurable. Typically, the flow is performed once a day, but it can vary. The volume is considered similar to the volume between the Analytics server and the Analytics database:

- Volume: High
- Frequency: Low

In each case, the TCP 1433 port is used, unless the database that is queried is not the “Default Instance”.

Traffic visibility: In a situation where the database and application querying the database are on the same system, the volume and frequency is inconsequential to the network traffic. However, if the components are on separate systems, it is important to analyze the network traffic.

Encryption: By default, the transactions to the database systems are unencrypted.

Network flow 6: Web Reports Server to Tivoli Endpoint Manager Server

Flow 6 (Figure 3-23) obtains data from the Tivoli Endpoint Manager Server. The Console and Web Reports Server perform similar tasks.

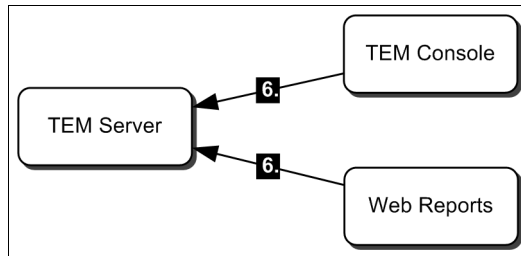


Figure 3-23 Network flow 6 in context

The Web Reports Server must obtain data from the Tivoli Endpoint Manager Server. This communication occurs periodically, and it occurs over an SSL-encrypted HTTPS connection. The process requires the exchange of a security token to identify and authenticate the Web Reports Server to the Tivoli Endpoint Manager Server. This token is generated on the initial use or definition of the Tivoli Endpoint Manager Server as a data source in the Web Reports Server interface.

An initial request is sent to the Tivoli Endpoint Manager Server on TCP port 52311 to the login URL, to which the Tivoli Endpoint Manager Server responds with a redirection to the configured SSL port. This port is TCP 52383 by default, although you can change the port. The Tivoli Endpoint Manager Server brokers the queries to the Tivoli Endpoint Manager database on behalf of the Web Reports Server. The Tivoli Endpoint Manager Server updates the Web Reports Server in-memory stores with changes since the last refresh. The volume of data exchanged therefore depends on the changes in state and the time between synchronizations. The frequency is on schedule, which by default is 15 seconds. This default might be too fast for many systems, and it can be reduced to a more appropriate value. Under most circumstances, we suggest that you increase this interval:

► Console or Web Reports Server to Tivoli Endpoint Manager Server

This flow uses port TCP 52311 for the initial request to provide a redirect to the SSL port. The volume of this flow is small and occurs when the Web Reports Server synchronizes content or a Web Reports user initiates a new connection:

- Volume: Low
- Frequency: Medium-High (Low for users, medium for Web Reports Server)

► Console or Web Reports Server to Tivoli Endpoint Manager Server

This flow uses port TCP 52389 redirected from the initial request. The volume of this flow is considered to be high. The data transferred synchronizes the Console or Web Reports Server in-memory data stores and contains all data changes since the last refresh. The flow occurs on a scheduled basis for the Web Reports Server. The Console maintains an in-memory data store for each Console operator, and each data store is updated separately. For many simultaneously active operators, this design potentially increases the perceived frequency. Web Reports Server users query against the single in-memory data store maintained by the Web Reports Server, so no additional traffic exists as a result of adding more Web Reports Server users:

- Volume: High
- Frequency: High, although it can be lowered by changing the Web Reports Server refresh interval and the console refresh interval

Network flow 7: Report Users to Reporting systems

Report consumers that use web browsers are the focus of this flow (Figure 3-24).

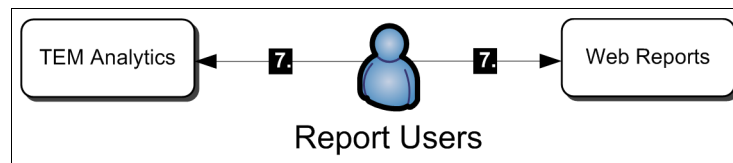


Figure 3-24 Network flow 7 in context

The two reporting-only systems are the Analytics and Web Reports Server systems. Each system can be accessed through a web browser over an unencrypted HTTP TCP port 80 or HTTPS TCP port 443. The ports are configurable. The information transferred over these connections is presentation-layer information and is no larger than the information of visiting a website on the public Internet. Frequency is 100% driven by the report users.

With many users, the frequency and thus the volume increases:

- Volume: Low
- Frequency: Low

We described all the flows that are core to the Tivoli Endpoint Manager system. The remaining flows represent optional flows, and they are represented in Figure 3-25 on page 121.

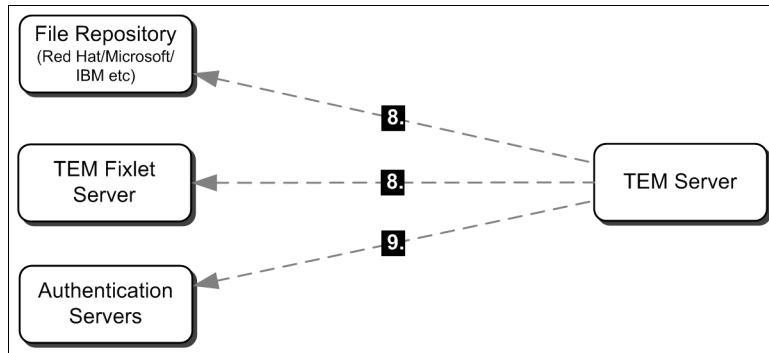


Figure 3-25 Optional flows in context

Network flow 8: Downloading from repositories

Network flow 8 is outside of the realm of the core system, but it is included for clarity. In most situations, the Tivoli Endpoint Manager Server is the only system that requires Internet access. *Air Gapped* configurations are possible where the Server has no Internet access, typically through physical isolation, although sometimes only through logical isolation. Updates must be obtained separately and manually imported into an Air Gapped environment.

When Agents request a file to be downloaded, the request eventually arrives at the Server, through Relays. If the file is not already downloaded and cached, or is missing from the cache, the Server downloads the file from the service identified by the URL of the request. For Windows patches, for example, a download from a Microsoft support site occurs. The file repository can be on the organizational intranet or external networks, such as the Internet. Typically HTTP, HTTPS, or FTP is used to download the file, although Tivoli Endpoint Manager also allows for extension of this capability with “download plug-ins” that can use other protocols and implement other authentication capabilities. Additionally, the Server synchronizes and updates Fixlet message content and External Sites from an authoritative server. This server is called the Tivoli Endpoint Manager Fixlet Server, an Internet-based service provided to all organizations that use Tivoli Endpoint Manager.

Network flow 9: Directory authentication

This flow identifies that certain components can use directory services for authentication versus only authenticating users locally. The Tivoli Endpoint Manager Server can authenticate users and authorized groups against an enterprise directory. Enterprise directory examples are LDAP and Active Directory. They use protocols appropriate to those technologies, such as LDAP over SSL, LDAP, Kerberos, and Windows NT LAN Manager (NTLM). The Web

Reports Server also can authenticate against an enterprise directory, although only Active Directory is supported at this time.

We conclude the overview of the network traffic flows in relation to the components in the Tivoli Endpoint Manager system. Additional flows might be required for integration with components external to the Tivoli Endpoint Manager system. An example of an external component is IBM Tivoli Change and Configuration Management Database (CCMDB). Next, we provide a brief overview of the locations of the concrete implementations of those components and present an example logical network diagram. The diagram is an example only. For more details about preferred practices and component placement, see Chapter 4, “IT endpoint security and compliance solution design” on page 125.

3.4 Physical component placement

By moving from a component view of a technology to a realized logical view, an organization can start considering where those systems and databases must be placed. Organizations can also consider *multiplicity*. Component views relate to a singular component, such as a Relay or an Agent. The realized logical perspective starts to consider multiples.

The determination of placement is affected by many factors. Examples include the existing policy about zoning, security requirements of the systems, access requirements, and access frequency.

Figure 3-26 on page 123 represents a simple example of the realization of the identified components to illustrate their placement in an actual scenario. It is not intended to be a definitive guide for where or how to place systems, merely an example.

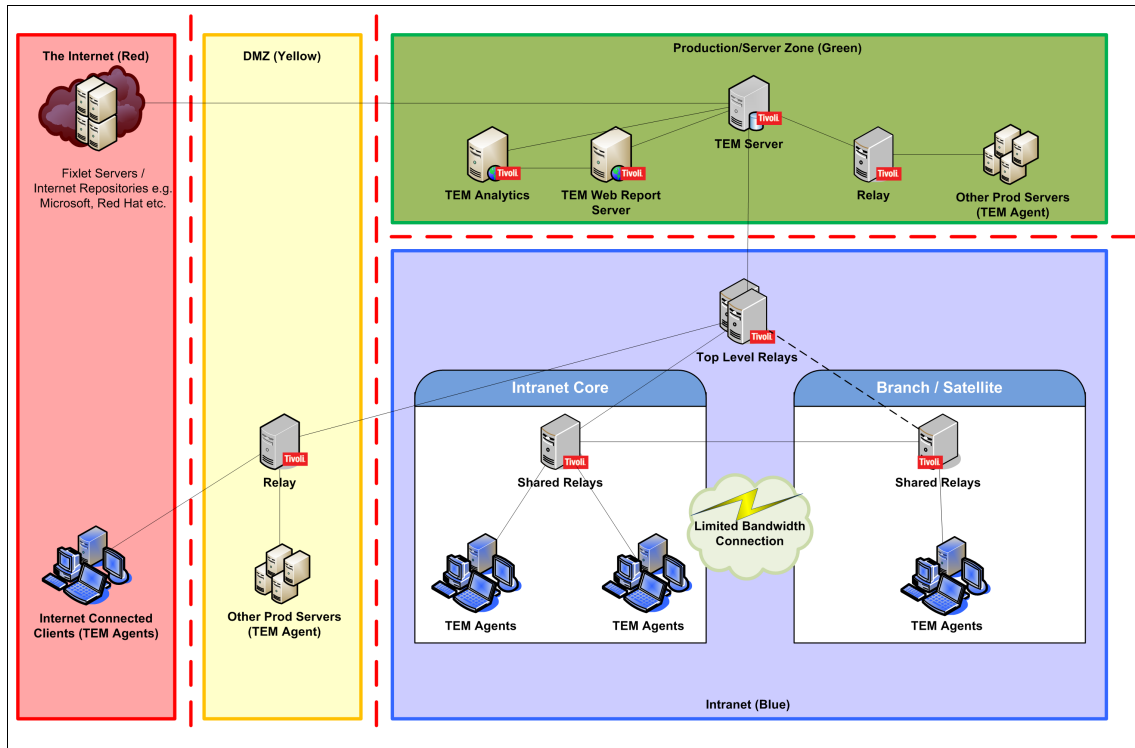


Figure 3-26 Example placement of realized systems based on the components

In Figure 3-26, we identify the following components:

- ▶ A Tivoli Endpoint Manager Server in a protected *Production* or *Green Zone*
- ▶ Production Zone servers that communicate through a Green Zone Relay
- ▶ *Yellow Zone* Relays for Internet-connected endpoints and devices in a DMZ
- ▶ Intranet Agents communicating through *shared Relays*
- ▶ Intranet-shared Relays existing on both sides of a low-bandwidth link to reduce traffic across the link and manage the bandwidth consumption
- ▶ *Top-level Relays* responsible for all Agents and subordinate Relays, reducing the number of connections into the Tivoli Endpoint Manager Server and Green Zone
- ▶ Low number of traffic flows crossing security zones

Placing the Tivoli Endpoint Manager Server in the Production Zone provides a layer of isolation from the intranet. Access to and from the system can be easily monitored, managed, and measured. Agents can be prevented from accessing

the Server directly, or this access can be permitted on a selective basis. Preventing the Server from having to deal with direct Agent communication allows it more time to handle its core responsibilities of arbitrating the data reads and writes to the database. Other servers can exist in the same security zone, or another zone with the same classification, as the Server. To prevent the data from needing to travel outside of this classification zone only to be returned to it, certain Relays are deployed inside the zone. Agents communicate through the Relay infrastructure to the top-level Relays.

These top-level Relays are responsible for shielding the Server and for handling any decryption of MLE-encrypted reports. In certain cases, the top-level Relays can communicate directly with Agents, but typically Agents communicate with other Relays in the infrastructure that are closer to them. These Relays all reside on existing devices, such as file and print servers, domain name servers (DNS), and DHCP servers. The organization does not need to procure additional hardware to act as Relays. Where low-bandwidth connections exist, such as in smaller satellite or branch offices, devices in the site can be designated as Relays for that site. This approach can better use the available bandwidth and act as a convenient control point to manage the bandwidth and throttle it, if necessary.

The DMZ (Yellow) Zone devices can use Relays in their own environment rather than accessing Relays in another zone directly. The Yellow Zone Relays also provide Relay services to the remote access users (connected to the organization through a virtual private network (VPN)). The DMZ does *not* provide Relay services to Internet-connected devices that are not connected to the organization through a VPN. These configurations result in fewer *holes* occurring in firewalls to cross security zones. These configurations provide appropriate security protection, inspection, and monitoring capabilities based on the needs and risks that exist in the environment of an organization.

3.5 Conclusion

In this chapter, we present a component overview of the Tivoli Endpoint Manager system to provide insight into the details of Tivoli Endpoint Manager. You must be able to place these components in the context of an IT organization and understand the data movement of data and network traffic flows. Many component combinations exist that you can use in your environment to satisfy security requirements.

In the next chapter, we look at the decisions made in the previous chapters and design considerations and preferred practices to help you formulate what is needed to plan a Tivoli Endpoint Manager implementation.



IT endpoint security and compliance solution design

In this chapter, we describe important aspects associated with designing a security and compliance solution with Tivoli Endpoint Manager. We describe the implementation plan by considering the geographical and network setup, maintaining and monitoring the system, and performance tuning.

We show how to plan the deployment by assessing the scale of the distributed endpoints. We also focus on the design solution for endpoint security that uses *patch management*, compliance that uses *security configuration management*, and *security and compliance analytics*. We conclude this chapter with the correct optimization and maintenance steps to avoid performance degradation.

4.1 Design consideration

In this section, we examine the considerations, issues, and advice for each aspect of a Tivoli Endpoint Manager deployment as defined in Table 4-1.

We do not go into the details of the Tivoli Endpoint Manager implementation process here; instead, we cover these topics briefly as a part of the overall picture.

Table 4-1 Overview of design considerations

Common aspects	Common considerations
4.1.1 Functional considerations	
Tivoli Endpoint Manager Server	<ul style="list-style-type: none"> ▶ Deployment size ▶ Web Reports ▶ Console and terminal access ▶ Scalability
Tivoli Endpoint Manager Relay	<ul style="list-style-type: none"> ▶ Performance and scalability ▶ Tivoli Endpoint Manager Relay/Agent location relationship ▶ Selection
Tivoli Endpoint Manager database	<ul style="list-style-type: none"> ▶ Performance and latency ▶ Size ▶ Backup, restore, and disaster recovery ▶ Authentication ▶ Infrastructure
4.1.2 Nonfunctional considerations	
High Availability (HA) and scalability	<ul style="list-style-type: none"> ▶ Severity and impact ▶ Performance and workload ▶ Database management
Network and security zones	<ul style="list-style-type: none"> ▶ Connection speed ▶ Port ▶ Network segment and firewall setting ▶ Air-gapped network specifics
Monitoring	<ul style="list-style-type: none"> ▶ Network accessibility ▶ Services and processes ▶ Database ▶ Log files

4.1.1 Functional considerations

We describe the functional aspects and focus on the design of the hierarchy between Tivoli Endpoint Manager Server, Relays, and Agents.

Tivoli Endpoint Manager Server

We begin with the Tivoli Endpoint Manager Server considerations, issues, and advice.

Considerations

- ▶ Deployment size

Organizations need to assess how many Tivoli Endpoint Manager Relays and Agents are going to be in their environments to determine hardware sizing for the Server.

- ▶ Web Reports

Organizations need to assess if components, such as Tivoli Endpoint Manager Web Reports and Tivoli Endpoint Manager for Security and Compliance Analytics, are required. Organizations need to assess whether these components can be deployed on the same system with the Tivoli Endpoint Manager Server.

- ▶ Console and terminal access

Organizations need to know where and how their operators access the Console. There are advantages for each of the methods related to performance. The connection speed between a remote system and the Tivoli Endpoint Manager Server needs to be considered.

- ▶ Scalability

Organizations typically plan to increase the number of Tivoli Endpoint Manager Agents in the future.

Issues and advice

- ▶ Deployment size

You need to assess the number of Tivoli Endpoint Manager Agents in the deployment. In most cases, this process is not complicated. For more information, see 4.2, “Tivoli Endpoint Manager solution design” on page 142. However, the number of Tivoli Endpoint Manager Relays can vary widely. We further examine the details in “Tivoli Endpoint Manager Relay” on page 150.

- ▶ Web Reports

A common consideration is whether Tivoli Endpoint Manager Web Reports and Tivoli Endpoint Manager for Security and Compliance Analytics are installed on the same system as the Tivoli Endpoint Manager Server. Our

suggestion is to put Tivoli Endpoint Manager Web Reports on the same Server as the Tivoli Endpoint Manager Server unless you operate in a large deployment. A large deployment is more than 100,000 Tivoli Endpoint Manager Agents. The Web Reports Server is installed automatically with the Tivoli Endpoint Manager Server. It caches all necessary information in memory to create its reports. For more information, see 3.2.3, “Tivoli Endpoint Manager Web Reports” on page 95.

The Tivoli Endpoint Manager for Security and Compliance Analytics Server uses a local database for its analytic investigations and is deployed on its own system. For more information, see 3.2.7, “Tivoli Endpoint Manager Analytics” on page 102.

- ▶ Console and terminal access

The Console is typically run on workstations or from a Terminal or Citrix server. The “BigFix Console System Requirements” report provides specifications for both configurations at the following location:

<http://support.bigfix.com/bes/install/consolereq.html>

If you prefer the Terminal or Citrix server access to maintain acceptable server performance, a 10 Mbps+ connection is required. Consoles that use high-latency or low-speed connections respond poorly, because large amounts of information are transferred between Tivoli Endpoint Manager Server and the Console. Thus, this type of deployment is affected by unacceptable latency problems. We suggest that you position the Terminal or Citrix server physically close to Tivoli Endpoint Manager Server.

- ▶ Scalability

Scalability is not a concern, because scalability is a benefit of Tivoli Endpoint Manager. By properly planning the layout for Tivoli Endpoint Manager Relays, the overall solution can be scaled to support up to 250,000 Agents per Server.

Tivoli Endpoint Manager Relay

Next, we describe the Tivoli Endpoint Manager Relay considerations, issues, and advice.

Considerations

- ▶ Performance and scalability

The number of Tivoli Endpoint Manager Relays needs to be estimated by considering the number and distribution of Agents.

- ▶ Tivoli Endpoint Manager Relay/Agents location relationship

Where Tivoli Endpoint Manager Agents exist influences where to implement Tivoli Endpoint Manager Relays.

- ▶ Selection

Design how Tivoli Endpoint Manager Agents select their Tivoli Endpoint Manager Relays. Manual and automatic Relay selection methods are available.

Issues and advice

► Performance and scalability

Relays can improve the performance of your installation. Relays lighten both upstream and downstream burdens on the Server. Instead of communicating directly with a Server, Agents can be instructed to communicate with designated Relays. This approach considerably reduces both server load and client/server network traffic.

Relays are a requirement for any network with slow links or more than a few thousand Agents. Even with only a few hundred Agents, Relays are advised. Relays download faster by distributing the load to several computers rather than being constricted by the physical bandwidth of the Server.

An overall Tivoli Endpoint Manager system is powerful. It is easy to deploy an Action that causes hundreds of thousands of Agents to download large files all at once. A Microsoft Windows service pack alone can span more than 200 MB. It is not uncommon to distribute software packages that are gigabytes in size. Without Relays, network pipes as fast as T1 (or faster) lines can be overwhelmed by many Agents requesting large, simultaneous file downloads.

Establishing the appropriate Relay structure is one of the most important aspects of deploying Tivoli Endpoint Manager to a large network. When Relays are fully deployed, an Action with a large download can be sent out to tens of thousands of computers with minimal WAN usage.

To ease deployment burdens and reduce the total cost of ownership, the Relays run on shared servers¹, such as file or print servers, domain controllers, storage management subsystem (SMS) servers, and anti-virus distribution servers. A typical installation has less than 1% of its Relays run on dedicated computers.

Generally, the Relay uses minimal resources and does not have a noticeable impact on the performance of the computer that runs it. When you use top-level Relays, we advise that you have one top-level Relay per 500 - 1,000 child Relays and directly reporting Agents.

For more information about Relay health, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Relay%20Health>

► Tivoli Endpoint Manager Relay/Agents location relationship

¹ A shared server is considered a server-quality computer that is always turned on.

The most common issue is how the overall hierarchy of the Tivoli Endpoint Manager system is designed. The preferred practice is not to assign more than 1,000 Agents for each Relay for the best performance. This number is not a hard limit. The number of Tivoli Endpoint Manager Relays can vary. We list typical examples in Table 4-2.

Table 4-2 Tivoli Endpoint Manager Relays deployment examples

Examples	Number of Tivoli Endpoint Manager Agents	Number of Tivoli Endpoint Manager Relays
<i>Recommendation is 1,000 Tivoli Endpoint Manager Agents for each Relay.</i>		
Small organization A	N/A	None
Large organization B	200,000	100
Larger organization C	200,000	15,000

Larger organization C deployed many Tivoli Endpoint Manager Relays, because it has many distributed stores. The geography of organizations can determine a large part of how Tivoli Endpoint Manager Relays are deployed. Typically, you need to position at least one Relay in every physical location. The number of Relays for each location depends on the number of Tivoli Endpoint Manager Agents in that location. For more information about Relay deployment, see the following article:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/BigFix%20Relays>

Sometimes, top-level Relays are required in large environments to maintain the appropriate performance characteristics.

► Selection

An important feature to consider when deploying Relays is the *automatic Relay selection*. In large deployments where the Agents are configured incorrectly, the Internet Control Message Protocol (ICMP) traffic sent from the Agents during automatic Relay selection can cause network problems and high router loads. You can constrain the number of ICMP packets by adjusting a configuration setting in the Agents to reduce the risks involved. For more information about the “Autoselection Failsafe Controls”, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Autoselection%20Failsafe%20Controls>

A Relay does not require dedicated system resources solely to be a Tivoli Endpoint Manager Relay. Ideally Relays are placed to coexist on file or print servers.

Tivoli Endpoint Manager database

Finally, we describe the Tivoli Endpoint Manager database considerations, issues, and advice.

Considerations

- ▶ Performance and latency

Organizations are concerned how to best configure the database to achieve optimum performance.

- ▶ Size

Organizations need to consider the volume of data to collect and retain after going into production.

- ▶ Backup, restore, and disaster recovery

Organizations need to outline detailed operations for backup, restore, and disaster recovery that include schedules, data sources, target location, and media.

- ▶ Authentication

Microsoft SQL Server can use two kinds of authentication.

- ▶ Infrastructure

Most organizations have an established database infrastructure and want to use it as much as possible.

Issues and advice

- ▶ Performance and latency

Tivoli Endpoint Manager performs many small database writes, which can introduce additional latency when not implemented by using fast and low latency I/O subsystems. An excellent solution is to use directly attached RAID 10 disks to ensure the best performance. For more details, see 3.1.3, “Database” on page 69.

- ▶ Size

Every organization is concerned about how much storage it needs to provide within the IT environment. The Tivoli Endpoint Manager database requires little space compared to many other application systems. A typical Tivoli Endpoint Manager database, even in a large environment, is rarely larger than 50 GB. See the “Disk Space Requirements” section of the Tivoli Endpoint Manager “Server Requirements” document:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Server%20Requirements>

► Backup, restore, and disaster recovery

Every organization is concerned about backup, restore, and disaster recovery.

For the Tivoli Endpoint Manager deployment, you need to back up data for archived Web Reports, support files for custom Web Reports, private encryption keys (if you use Message Level Encryption), and custom packages that were uploaded to the system for distribution to Agents. Some information about the IP addresses of computers, Actions, and Fixlets is necessary but automatically rebuilt by the Tivoli Endpoint Manager Server.

For a recovery procedure, you need to prepare an identical version of Microsoft SQL Server. You need to ensure that the new Tivoli Endpoint Manager Server can be reached on the network that uses the same URL as specified in the masthead file.

For more details, see the “Disaster Recovery Overview” document:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Disaster%20Recovery%20Overview>

► Authentication

Microsoft SQL Server authentication and Windows NT Authentication mechanisms are available for the Microsoft SQL database users. You do not need to select a method, because Tivoli Endpoint Manager uses SQL Server Authentication as a default. To use the Windows NT Authentication mechanism instead, see “How can I change my database authentication type from SQL to NT for my Tivoli Endpoint Manager Server?”:

<http://support.bigfix.com/cgi-bin/kbdirect.pl?id=1806>

► Infrastructure

Tivoli Endpoint Manager is designed to use an SQL database that is run locally on the main Tivoli Endpoint Manager Server system. For reasons of cost, performance, maintenance, and control, some organizations want the SQL database on a separate computer.

There are advantages and disadvantages to creating a remote database connection. The advantages include cost savings on SQL licensing, use of existing hardware, access to high performance hardware, consolidation of SQL databases, and compliance with existing company policies.

The disadvantages of a remote database connection include added complexity to the installation of the Tivoli Endpoint Manager Server, possible performance loss, and additional support. The use of a remote database

installation introduces complexity that can affect the performance and stability of a Tivoli Endpoint Manager Server installation. Ensure that there is a high-speed connection from the Server to the database computer (1 GBps minimum). Ensure that the Tivoli Endpoint Manager database does not compete with another application for database resources. Ensure that the Tivoli Endpoint Manager databases are put onto high-performance disks and that the SQL Server computer has sufficient memory. In addition, there are many manual configuration steps required to make the remote connection and creating them adds support time to setting up the Tivoli Endpoint Manager Server. Finally, upgrades are likely to reset most of the remote database configuration settings, and you need to redo them on upgrades to Tivoli Endpoint Manager. For more information about the use of a remote database system, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#wiki/Tivoli%20Endpoint%20Manager/page/Remote%20Database%20Guide>

4.1.2 Non-functional considerations

In this section, we describe the aspects of the Tivoli Endpoint Manager system design that do not directly relate to the functional requirements. We present considerations when designing the non-functional, or operational, aspects.

High availability (HA) and scalability

We begin with the considerations, issues, and advice about HA design.

Considerations

► Severity and impact

The organization needs to evaluate its risk posture toward the availability of the endpoint management solution where they might need two or more Tivoli Endpoint Manager Servers.

► Performance and workload

If the organization deployed many Agents, it might need two or more Tivoli Endpoint Manager Servers for performance and workload reasons.

► Database management

To design an HA architecture, the organization must deploy at least two databases.

Issues and advice

► Severity and impact

The impact of a Tivoli Endpoint Manager system that is not operational is that functionality, such as patching, analysis, software distribution, and reporting, is not available. Although important, for most organizations, a Tivoli Endpoint Manager system that is not operational is not considered a critical situation, and many times an HA solution is not required.

Additional servers can help to create a redundant system that is hardened to outages. The same redundant system can address the performance issue and distribute the workload between servers. Understand how the redundant system helps you so you can create the most efficient deployment for your particular network. Consider these important elements of multiple server installations:

- Servers communicate on a regular schedule to replicate their data.
- When each server is ready to replicate from the other servers in the deployment, it calculates the shortest path to every other server in the deployment. Primary links are assigned a length of 1, secondary links 100, and tertiary links 10,000. Links, which resulted in a connection failure the last time they were used, are considered to be non-connected.
- When an outage or other problem causes a network split, it is possible for a custom Fixlet or a retrieved property to be modified independently on both sides of the split. When the network is reconnected, precedence goes to the version on the server with the lowest Server ID.
- If multiple copies of Web Reports are installed, they operate independently. Each Web Reports Server can connect to the Server that is most convenient, because they all contain equivalent views of the database.
- By default, server 0 (zero) is the master server. With the Tivoli Endpoint Manager Administration Tool, you can perform certain administrative tasks (such as creating and deleting users) only when connected to the master server.
- If you want to switch the master to another server, you can set it.

Figure 4-1 on page 135 depicts a typical *Distributed Server Architecture*² (DSA) setup with two servers. Each Server is behind a firewall, possibly in a separate office, although it is easy to set up multiple servers in a single office, as well. It is important that the Servers have high-speed connections to replicate the Tivoli Endpoint Manager data. Generally, LAN speeds of 10 - 100 MBps are required. The Tivoli Endpoint Manager Servers communicate over Open Database Connectivity (ODBC) and HTTP protocols. This DSA

² Sometimes DSA is also referred to as Disaster Server Architecture

configuration provides automatic failover and failback services and minimizes data loss.

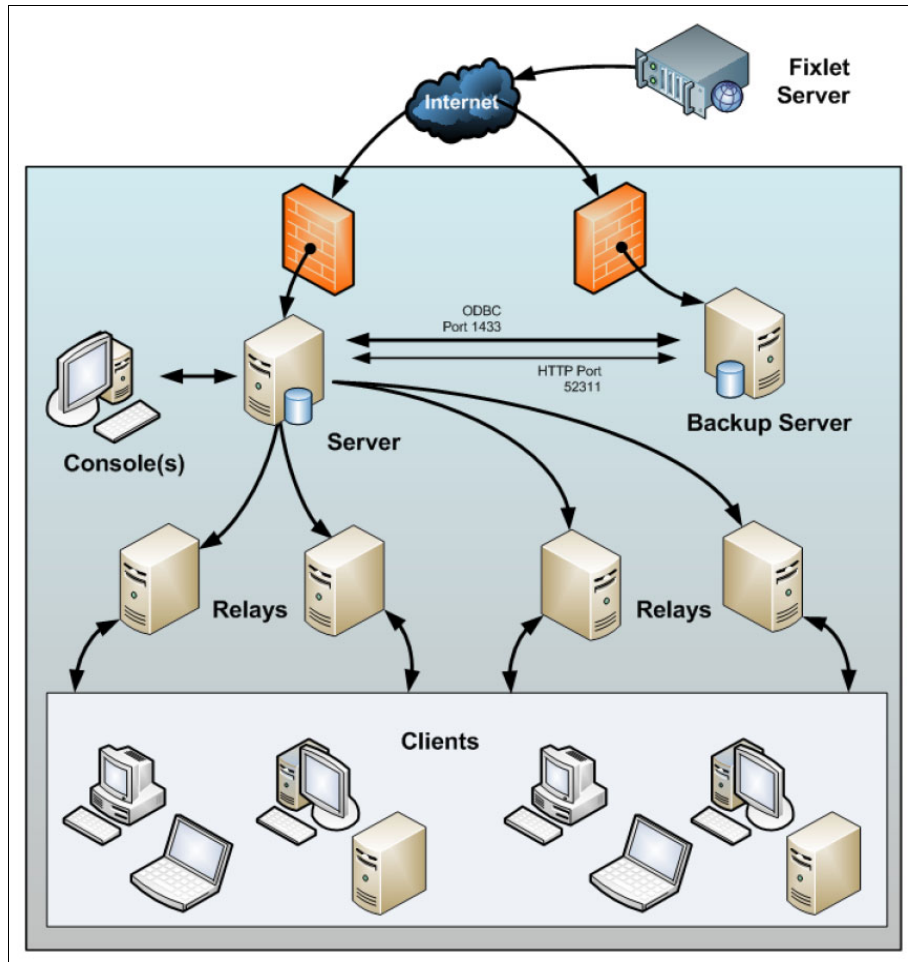


Figure 4-1 DSA architecture overview

If a Server fails, whether due to disaster or planned maintenance, the DSA deployment reconfigures itself (hot failover) as the orphaned Relays find a new Server connection. When the disabled server comes back online, its data automatically is merged with the data on the healthy server (Figure 4-2 on page 136).

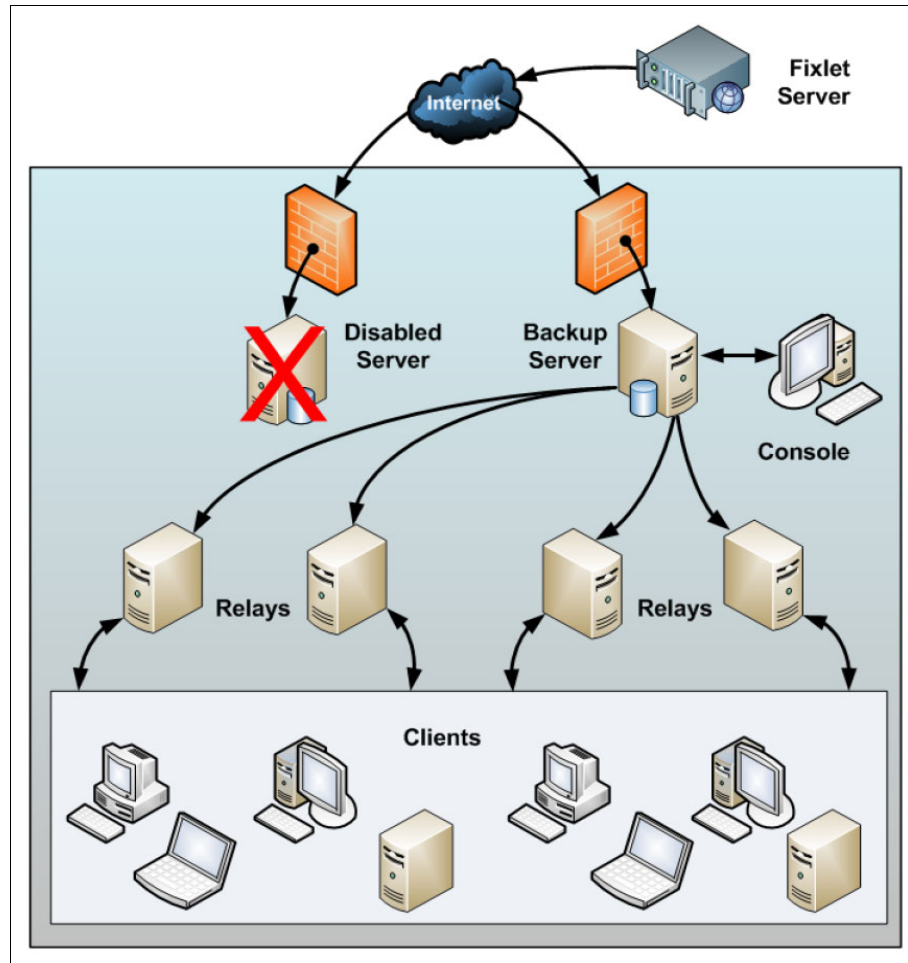


Figure 4-2 DSA failover

For more information, see the “Disaster Server Architecture” document:

[https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Disaster%20Server%20Architecture%20\(DSA\)](https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Disaster%20Server%20Architecture%20(DSA))

► Performance and workload

The guideline for the Agent/Server ratio is that you need one Tivoli Endpoint Manager Server for every 250,000 Agents. For more detailed information about “Server Requirements”, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Server%20Requirements>

If you must deploy multiple Tivoli Endpoint Manager Servers, see the details about the Distributed Server Architecture explained in “Considerations” on page 133.

- ▶ Database management

We describe all concerns about database management in “Tivoli Endpoint Manager database” on page 131.

Network and security zones

Next, we describe the considerations and issues that relate to network and security zones.

Considerations

- ▶ Connection speed

The connection speed between the Tivoli Endpoint Manager Server, Relay, Agent, and other components is an important consideration.

- ▶ Port

For more complex deployments where an organization tightly manages port usage, you must thoroughly understand the ports that are required by the Tivoli Endpoint Manager solution.

- ▶ Network segments and firewall setting

Organizations deploy segregated network segments and use firewalls and other technology to control the information flow over those segments. You need to examine how Tivoli Endpoint Manager can be deployed in this environment.

- ▶ Air-gapped network specifics

The Tivoli Endpoint Manager Servers can connect to the Internet on port 80 to connect to the IBM Fixlet Server. The Tivoli Endpoint Manager Server can be set up to use a proxy, which is a common configuration. Alternatively, an *air-gap* can be used to physically separate the Tivoli Endpoint Manager Server from the Internet Fixlet Server.

Issues and advice

- ▶ Connection speed

File downloads, for example, Microsoft Service Packs, consume the bulk of the bandwidth in a typical Installation. You can control this bandwidth by *throttling*, which limits the number of bytes per second. You can specify the bandwidth throttling on either the Server, the Agent, or both (in which case the lower of the two values is used).

Bandwidth throttling can be important whenever you have bandwidth issues, as in the following situations:

- A remote office with a thin channel
- Remote dial-in users or users on a slow connection
- A shared channel with higher-priority applications
- A WAN or LAN that is already saturated or has stringent load requirements

Bandwidth throttling settings (and other Relay, Server, and Client settings) can be set by using the Tasks option on the Support site.

You might, for example, throttle an Agent to 2 KBps if the Agent is more than three hops from a Relay. However, the optimal data rates can vary, depending on the current hierarchy and the network environment.

A better technique is to use *dynamic bandwidth throttling*, which monitors and analyzes overall network capacity. Normal throttling specifies a maximum data rate. Dynamic throttling adds a *busy time* percentage. This busy time percentage is the fraction of the bandwidth that you want to allocate when the network is busy. For example, you can specify that downloads must not use more than 10% of the available bandwidth whenever Tivoli Endpoint Manager detects existing network traffic. Dynamic throttling also provides for a minimum data rate in case the busy percentage is too low to be practical.

When you enable dynamic throttling for a link, Tivoli Endpoint Manager monitors and analyzes the existing data throughput to establish an appropriate data rate. If there is no competing traffic, the throughput is set to the maximum rate. In existing traffic, it throttles the data rate to the specified percentage or the minimum rate, whichever is higher.

For more details about throttling, see the document “Bandwidth Throttling”:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Bandwidth%20Throttling>

► Port

The Tivoli Endpoint Manager Console and Server communicate by using ODBC, which operates on port 1433 by default. If you need to change this port, you must work with your database operator to ensure an organization-compliant configuration.

By default, the Server uses port 52311 to communicate with the Agents, but any port number can be chosen. Avoid the reserved ports from 1 - 1024 because of potential conflicts and the difficulty of managing network traffic.

Your choice of the Server port number is factored into the generation of the masthead, which specifies URLs for the action, registration, reporting, and mirror servers. As a consequence, you must finalize your port number before the installation.

- ▶ Network segment and firewall setting

You can encounter serious problems if you do not work closely with your firewall operators to ensure that adequate ports are available for communication in the overall Tivoli Endpoint Manager deployment. It is important that firewall operators and network engineers understand the types of traffic that the Tivoli Endpoint Manager Servers, Relays, Agents, and Consoles can generate. For more information, see the “Network Traffic Guide”:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Network%20Traffic%20Guide>

Through specially configured Relays, Agents that connect to the Internet (at home, airports, or coffee shops) only can be managed as though they are within the corporate network, with no VPN connection. These Relays are deployed in the DMZ. For more information, see “Internet Relays”:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Internet%20Relays>

- ▶ Air-gapped network specifics

Some organizations do not allow Tivoli Endpoint Manager Servers to access the Internet-based IBM Fixlet Server directly. In this case, the air-gap network configuration is the solution. A tool, BESAirgapTool, helps you create a request file for license and Fixlet content updates. For more details about how to use this tool, see “Installing in an Air-Gapped Network”:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Installing%20in%20an%20Air-Gapped%20Network>

Monitoring

Many organizations consider Tivoli Endpoint Manager components part of their critical IT infrastructure. They want to monitor the components by using third-party monitoring tools to verify the correct functionality. We describe the considerations and issues that relate to monitoring.

Considerations

- ▶ Network accessibility

Network accessibility is likely the most important aspect for the Tivoli Endpoint Manager deployment. If the network connections fail between components, Tivoli Endpoint Manager cannot send or receive any Fixlet content, reports, or requests. Many organizations are concerned with the network connections to monitor to avoid these problems.

- ▶ Services and processes

The Tivoli Endpoint Manager environment requires certain services and processes to be available on the Server, Relay, database, and Agent. Organizations need to monitor these certain services and processes. They must plan to handle and recover from any outages.

- ▶ Amount of data collected

Organizations must monitor and properly plan where Tivoli Endpoint Manager log files and other data is stored and how much the amount of data increases over time.

Issues and advice

- ▶ Network accessibility

To avoid any problems caused by unavailable communication to endpoints, it is necessary to monitor the Tivoli Endpoint Manager Server periodically to ensure that it is up and accessible by all top Tivoli Endpoint Manager Relays. Without this top-down communication channel, the correct content cannot be distributed. Another critical connection to monitor is between the Tivoli Endpoint Manager Server and the IBM Fixlet server. The IBM Fixlet server provides baseline content to the Tivoli Endpoint Manager Server, which provides the content to all participating systems in the hierarchy.

- ▶ Services and processes

If services or processes (instead of a complete system failure) fail on a Tivoli Endpoint Manager component, the hierarchical information flow to and from Agents can be interrupted. Operators cannot access the latest information on the Console. To react immediately to this type of failure, you need to monitor services and processes that contain either BES or BigFix on all your Tivoli Endpoint Manager Server and Relay components.

- ▶ Amount of data collected

The typical amount of data collected from any log files is not large, but information in those data tables must be monitored to avoid any problems. A suggestion for closer monitoring is the BufferDir directory, which temporarily stores reports from Tivoli Endpoint Manager Agents before they are put into the database. If this directory fills up, it indicates that information is not properly stored in the database immediately. By default, the BufferDir directory is at C:\Program Files\BigFix Enterprise\BES Server\FillDBData\BufferDir\.

The two types of systems to monitor closely in the Tivoli Endpoint Manager environment are the Server and the Relay. We provide a quick list of items to consider for the Server and the Relay monitoring:

- ▶ Tivoli Endpoint Manager Server:
 - Root Server: Handles all incoming connections to the Server.
 - FillDB: Puts information from the Agents into the database.
 - GatherDB: Puts new Fixlet information into the database.
 - Gather: Contacts the Internet to download files and to download new Fixlet messages.
 - Agent (optional): The Agent checks for known issues on the Server.

Agent on the Server: Without an Agent on the Server, the Server cannot process Server upgrade Fixlet messages.

- Web Reports (optional): Many times, the Web Reports Server runs on the same system as the other Server components.
- The BufferDir is considered *full* if it has 3 MB of files or if it has more than 10,000 files (by default). It is a good idea to monitor the BufferDir folder and issue an alert if the folder has more than 2.5 MB of files or has more than 9,000 files. Be careful to not monitor this folder too often, because it might cause performance problems. You can check once every 10 minutes, but do not check every 10 seconds. The BufferDir is one of the most important monitoring activities. If the BufferDir fills up, it indicates that information is not getting to the Server quickly, and it can be a severe problem.
- Ensure that the MSSQLServer service is running.
- Ensure that the SQL Server Agent is running.
- Any additional standard SQL Server checks are useful.
- Each Fixlet message site to which your Server subscribes has a GatherURL (the GatherURL is stored in the masthead file for each site). For example, the *Patches for Windows (Enterprise Security)* site has a GatherURL of <http://sync.bigfix.com/cgi-bin/bfgather/bessecurity>. If you enter the URL into a browser and retrieve the data at that location, you receive information about the site. Within this returned data, approximately 13 lines from the top, the line `Version: XXX` indicates the current version of the provided site.
- Each Fixlet message site is *mirrored* on the Server. The mirrored GatherURL gives the same information as the GatherURL of the Fixlet servers. For example, to access the mirrored GatherURL, use

`http://yourservername:52311/cgi-bin/bfenterprise/besgathermirror.exe?url=http://sync.bigfix.com/cgi-bin/bfgather/bessecurity.`

- By default, the Server looks for new Fixlet message sites every 60 minutes from the main Fixlet servers, so there is a potential lag of 60 minutes when the two URLs do not match.
- Network accessibility: Ping the Server periodically to ensure that it is up and accessible from the network. It must be reachable by all top-level Relays.
- ▶ Tivoli Endpoint Manager Relay:
 - BigFix Client: The BigFix Client is important to the normal operations of the BigFix Relay.
 - The BufferDir is *full* if it has 3 MB of files or if it has more than 10,000 files (by default). It is a good idea to monitor the BufferDir folder and issue an alert if the folder has more than 2.5 MB of files or more than 9,000 files. Be careful to not monitor this folder too often, because it might cause performance problems. It is alright to check once every 10 minutes, but do not check every 10 seconds.
 - The Relay mirrors data in the same way as the main Server.
 - In almost all cases, Relays must have the same information as the Server within a few seconds or minutes of the Server update.
 - You can check to see whether the Relay mirrors the same information as the Server by checking the URL and comparing that information mirrored by the main Server:
`http://yourrelayname:52311/cgi-bin/bfenterprise/besgathermirror.exe?url=http://sync.bigfix.com/cgi-bin/bfgather/bessecurity`
 - Check to ensure that the *actionsite* and *opsites* are mirrored properly.
 - Network accessibility: Ping each Relay periodically to ensure that it is up and accessible from the network. It must be reachable by all of the Agents that select this Relay.
 - Ensure that the Relay is getting up-to-date Fixlet information: A Relay gathers new Fixlet messages from the main Servers whenever the new Fixlet message site versions are available.

4.2 Tivoli Endpoint Manager solution design

In this section, we highlight the necessary considerations before you implement the Tivoli Endpoint Manager.

4.2.1 Deployment planning

We look at the planning steps to take before deploying Tivoli Endpoint Manager in your organization. This information is broken down into the deployment scope, how to fit Tivoli Endpoint Manager into your network, and the required equipment required, and high availability and disaster recovery.

Deployment scope

Before deploying a Tivoli Endpoint Manager solution, an organization needs to estimate how many endpoints to deploy in each location. The Tivoli Endpoint Manager Agent must be deployed to every endpoint that needs to be protected. Do not forget to include the computers that are not Windows computers, home users, mobile users, and leave room for growth. The Tivoli Endpoint Manager Agent is compatible with most versions of Microsoft Windows, Mac OS, Linux, and AIX.

We approximate the scale for Tivoli Endpoint Manager deployments:

Small	< 5,000 Agents
Medium	< 25,000 Agents
Large	< 100,000 Agents
Very Large	> 100,000 Agents

Network requirements

Tivoli Endpoint Manager works well with network security devices, such as firewalls and Intrusion Prevention Systems (IPS). For Tivoli Endpoint Manager to function properly, the network of the organization must meet the following conditions:

- ▶ TCP port 52311 (customizable) open on all networks.
- ▶ Allow UDP packets on all networks for Server and Relay notifications.
- ▶ Network must allow ICMP if automatic Relay selection is needed.
- ▶ Avoid stripping packet headers in the network.
- ▶ Proxy server must ideally allow non-standard user-agent.
- ▶ Tivoli Endpoint Manager Server must have access to the Internet.

In Figure 4-3 on page 144, we illustrate how Tivoli Endpoint Manager fits within the broader spectrum of a secure network implementation of an organization. For a more detailed explanation of the concept of network security zones, see *IBM Security Solutions Architecture for Network, Server and Endpoint*, SG24-7581.

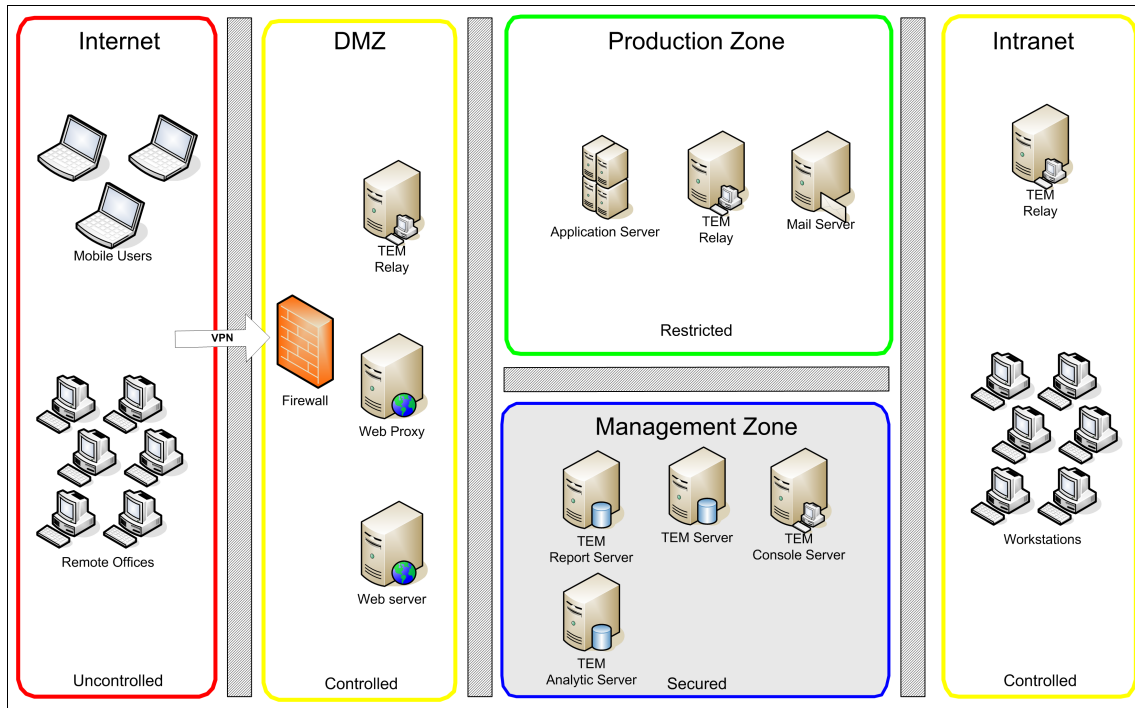


Figure 4-3 Example of Tivoli Endpoint Manager deployment in network security zone view

Endpoint distribution

Tivoli Endpoint Manager can support large organizations with multiple distributed sites in different locations around the world. Tivoli Endpoint Manager can control, manage, and deploy solutions to endpoints in remote sites and roaming endpoints through Relays. For technical details about Tivoli Endpoint Manager Relays, see 3.1.5, “Relay” on page 74.

Figure 4-4 on page 145 illustrates an example of a typical environment of an organization and the Tivoli Endpoint Manager deployment.

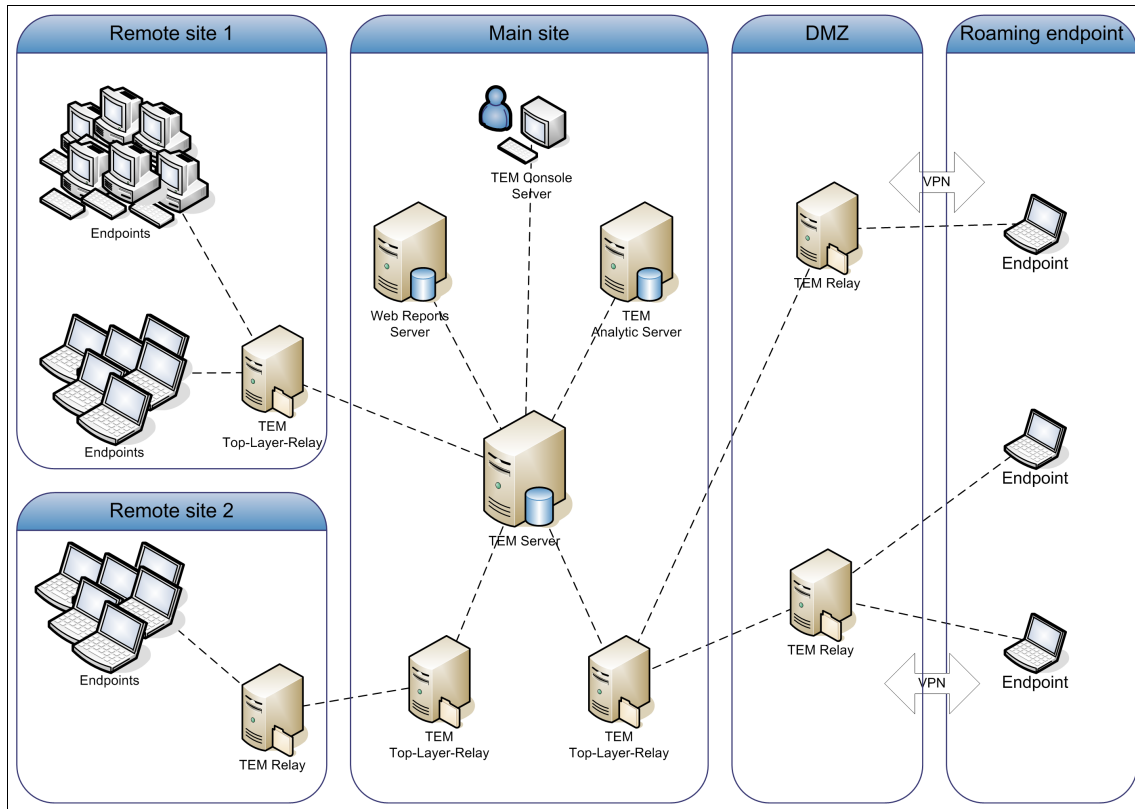


Figure 4-4 Tivoli Endpoint Manager deployment example

Figure 4-4 shows one way that Tivoli Endpoint Manager can be implemented in an organization. Remember that there are many ways to implement Tivoli Endpoint Manager. The requirement is that the endpoint that needs to be protected is connected to a Tivoli Endpoint Manager Relay or Tivoli Endpoint Manager Server³.

The Tivoli Endpoint Manager Server must have Internet connectivity with port 80 open to connect to the Tivoli Endpoint Manager Fixlet server. A dedicated port⁴ must be opened between the Tivoli Endpoint Manager Server, Tivoli Endpoint Manager Relay, Tivoli Endpoint Manager Console, and the endpoint in the network.

³ Direct connect to Server is not advised. It is a preferred practice to connect endpoints to a top-level Relay even if the endpoint is in the same network segment as Tivoli Endpoint Manager Server.

⁴ The port is customizable. The default is 52311.

High availability

A well-designed Tivoli Endpoint Manager deployment must be robust against a network or hardware outage of any component under the Server level due to the self-initiated Relay select process. If a Relay or a network route to a Relay fails, the Tivoli Endpoint Manager Agent can identify another Relay. See the illustrations in Figure 4-5 and Figure 4-6 on page 147.

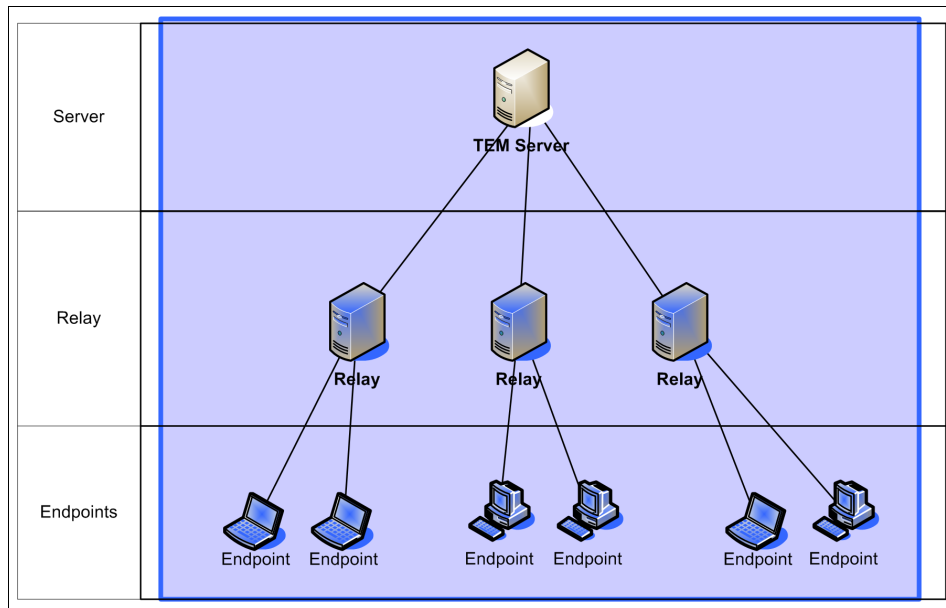


Figure 4-5 Normal runtime circumstance

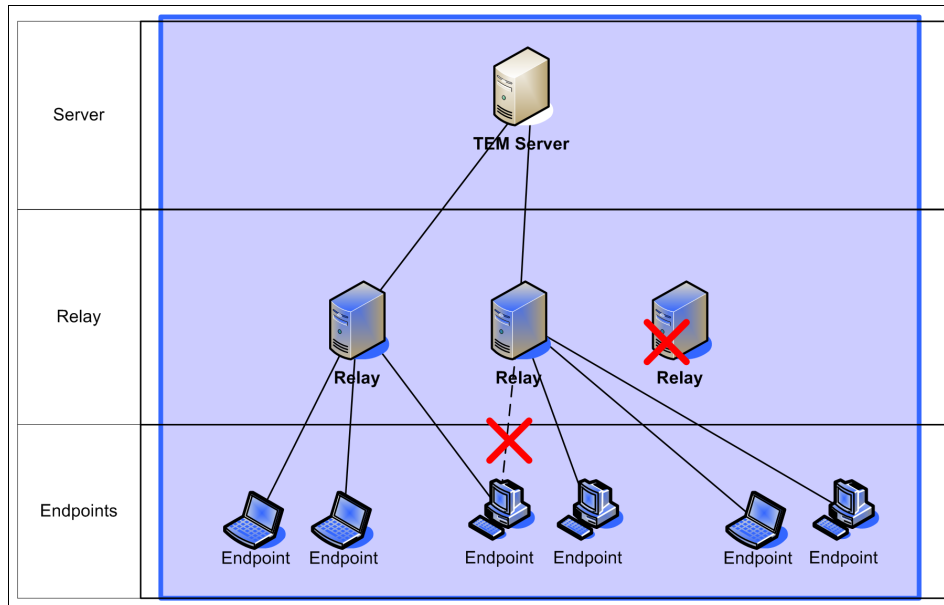


Figure 4-6 Failure circumstance

When designing a Tivoli Endpoint Manager system, consider the fault tolerance in a Relay failure case. A good design incorporates enough Relays to handle the extra load when a failure occurs. For more information, see “Tivoli Endpoint Manager Relay” on page 150.

The high-availability solution that Tivoli Endpoint Manager provides for the Server is called *Distributed Server Architecture (DSA)*. However, in most solution designs, DSA is not considered a requirement for the implementation. With an appropriate backup mechanism in place, a restore or a rebuild of a Tivoli Endpoint Manager Server takes a short time, maybe a few hours. After the Tivoli Endpoint Manager Server and Tivoli Endpoint Manager database are restored, the Tivoli Endpoint Manager infrastructure continues to function.

The Tivoli Endpoint Manager Agent and Relay cache any data when connection is lost to the Server, and gradually push the data back to the Tivoli Endpoint Manager Server after the connection is restored. During the Server downtime, configuration enforcements and any scheduled tasks already received by an Agent continue to function. The frequency of compliance changes and patching applications is done on a daily, if not longer, period. From an endpoint security and compliance solution view, the impact of a Tivoli Endpoint Manager Server failure is considered minor for most organizations. We describe implementation of DSA in “Implementing a Distributed Server Architecture” on page 150.

4.2.2 Deployment design

In this section, we examine the deployment of Tivoli Endpoint Manager components to the appropriate hosts. We focus on the components that are required to provide endpoint security and compliance.

Detailed installation steps are not covered in this book. See the *IBM Tivoli Endpoint Manager Administrator Guide* at the following location:

http://publib.boulder.ibm.com/infocenter/tivihelp/v26r1/topic/com.ibm.tem.doc_8.2/Platform/Adm/c_introduction.html

Tivoli Endpoint Manager Server

The Tivoli Endpoint Manager Server performs intensive file I/O operations on temporary data⁵. It is considered a preferred practice to use a local storage system with good read and write performance. You can choose to connect to a remote database. However, to maximize performance, install Tivoli Endpoint Manager database locally with Microsoft SQL 2008 R2 preinstalled. If no database is available, Tivoli Endpoint Manager also provides the option to install Microsoft SQL 2005 Express for you.

Microsoft SQL 2005 Express: Use Microsoft SQL 2005 Express for testing deployment only. Use a full version of Microsoft SQL for production deployment.

An encryption key is generated during the installation. The key must be kept safe. The key must not be stored on any components of Tivoli Endpoint Manager, such as the Server, Relay, Console Server, or database. If the key is obtained by an unauthorized user, unauthorized personnel can control the entire Tivoli Endpoint Manager system. It is not possible to recover this key from IBM if it is lost.

There must be at least one master operator created during the installation for Tivoli Endpoint Manager to function properly. More users can be added after the Tivoli Endpoint Manager Console is installed. For more information about users and privileges, see 3.1.10, “Users” on page 86.

⁵ Temporary file locations are:
BES Server\FillDBData\BufferDir
BES Server\sitearchive
BES Server\wwwrootbes\bfmirror\bfsites
BES Server\wwwrootbes\bfsites

Master operator: Because Tivoli Endpoint Manager is a powerful management system, any user with master operator access can cause serious harm to the information system of the whole organization. It is imperative that only trained and trusted personnel in the organization are granted this access right to Tivoli Endpoint Manager. You must revoke access for users who no longer need to access the system.

The storage performance of the Server directly affects the overall performance of the Tivoli Endpoint Manager environment. If improperly configured, it can cause severe performance problems. For the best performance, use solid-state storage technology with RAID 10. For more details about solid-state storage, see 3.1.3, “Database” on page 69. For details of suggested RAID array settings, see this website:

<http://support.bigfix.com/bes/misc/raidconfig.html>

Tivoli Endpoint Manager Console

A Tivoli Endpoint Manager Console is necessary to manage the Tivoli Endpoint Manager Server. To install the Console, your computer must meet the following minimum requirements:

- ▶ Hardware: Intel Pentium III class processor with 512 MB RAM. Larger deployments require more capable computers.
- ▶ Software: Microsoft Windows XP, Windows 2003 Vista, Windows 2008, Windows 7, or Windows 2008 R2 with Internet Explorer Version 7.0 or later.

The Tivoli Endpoint Manager Console can be installed on any moderately powerful computer. However, as the number of computers that you are managing with the Console grows, you might need a more powerful computer. The latest Console suggestions are at the Tivoli Endpoint Manager support site:

<http://support.bigfix.com/bes/install/consolereq.html>

The Tivoli Endpoint Manager Console also requires a high-bandwidth, fast connection to the Server due to the large amount of data that needs to be transferred to the Console. IBM suggests that remote operators use a remote control connection, such as a Citrix server or Terminal Services computer to a Tivoli Endpoint Manager Console. The Tivoli Endpoint Manager Console must be physically close to Tivoli Endpoint Manager Server. Do not have a console that runs on a system that is far from the Tivoli Endpoint Manager Server. For more information about the Tivoli Endpoint Manager Console, see 3.1.4, “Console” on page 71.

Tivoli Endpoint Manager Relay

The Tivoli Endpoint Manager Relay can be installed through a Task, Install Tivoli Endpoint Manager Relay. Relays must be installed on systems that run one of the following products:

- ▶ Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2
- ▶ Red Hat Enterprise Linux 4, 5, and 6
- ▶ Solaris 10
- ▶ AIX 5.3 and 6.1 computers

We suggest that you install a Relay on a system with a static IP address and little downtime. A Relay must have a two-way TCP connection to its parent Relay or Tivoli Endpoint Manager Server.

IBM advises that you have at least one Relay in a network segment with limited bandwidth to Relays in other network segments. Add more Relays to share the load if there are more than 1,000 Agents connected to a single Relay. For the best results, distribute Relays evenly across your network. Also, avoid chaining too many Relays together. An increase of Relay hop counts increases the delay in the time that it takes endpoints to receive data from the Tivoli Endpoint Manager Server. Try to design a Tivoli Endpoint Manager implementation with most endpoints connected to the Tivoli Endpoint Manager Server within three Relay hop counts. This design offers better overall response time and performance.

Tivoli Endpoint Manager Agent

There are many ways to install Tivoli Endpoint Manager Agent to an endpoint, including network shares, the Active Directory, a login script, or a manual installation through an installation package. An operator can choose the deployment method best suited for the environment.

Implementing a Distributed Server Architecture

DSA enables an organization to install multiple Tivoli Endpoint Manager Servers that replicate information between each other. In a failure, a backup Server can automatically take over as a fully functional Server. When the failed Server is restored, it automatically receives updated information from the backup Server. See Figure 4-7 on page 151, and Figure 4-8 on page 151.

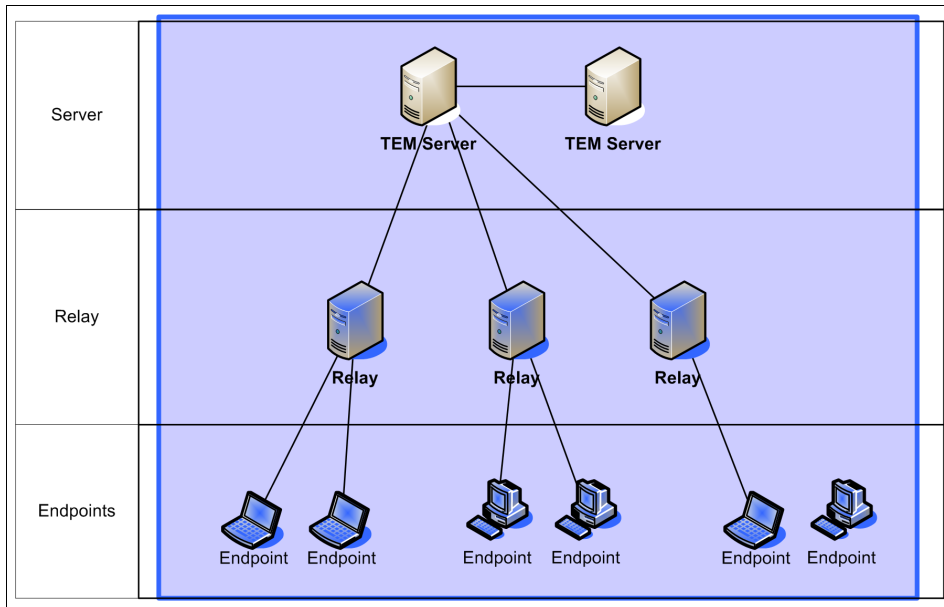


Figure 4-7 Normal circumstance

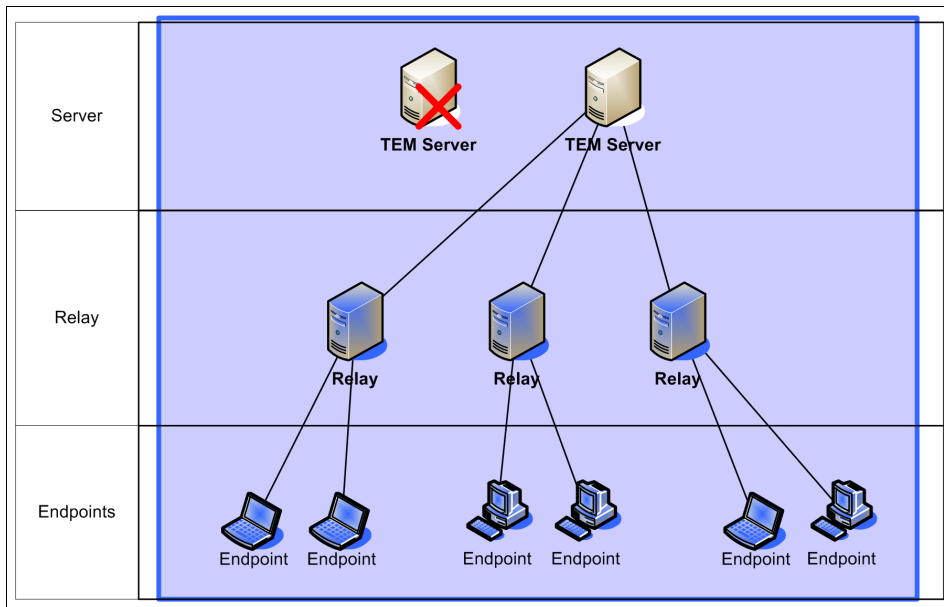


Figure 4-8 Failure circumstance

Requirements

- ▶ All DSA Servers must use the same authentication method.
- ▶ DSA Servers must have similar performance. We suggest that you use the same hardware for all DSA Servers, if possible.
- ▶ DSA Servers must have the same version of Microsoft SQL Server installed.
- ▶ The larger the Tivoli Endpoint Manager deployment, the more you need a fast, high-bandwidth connection between DSA Servers. For a large deployment, less than 1 ms latency is required. Custom engineering is needed for a large deployment to achieve the optimal connections between DSA Servers.

Asset discovery

One of the questions asked after a deployment is, “*Did we cover all the endpoints in the organization?*”

Any endpoints that were missed during the initial deployment phase might become a security vulnerability. IBM suggests that you use *asset discovery* to discover unmanaged assets.

With asset discovery, you can deploy *Scan Points*, which are Nmap scanners, to specified Tivoli Endpoint Manager Agents in your network by using Fixlet messages and Tasks. See Figure 4-9 on page 153. You can then use Fixlets and Tasks to periodically run scans. The scan results are delivered automatically to the Tivoli Endpoint Manager Server, which imports the data into the database.

The scan information can then be viewed in the Tivoli Endpoint Manager Console in the Unmanaged Asset tab. Operators can then choose either to deploy a Tivoli Endpoint Manager Agent to manage those assets, or take other security measures. Operators decide the period and frequency of the scan. They evaluate their network environment before deploying an asset discovery procedure to effectively discover unmanaged assets without overloading their network.

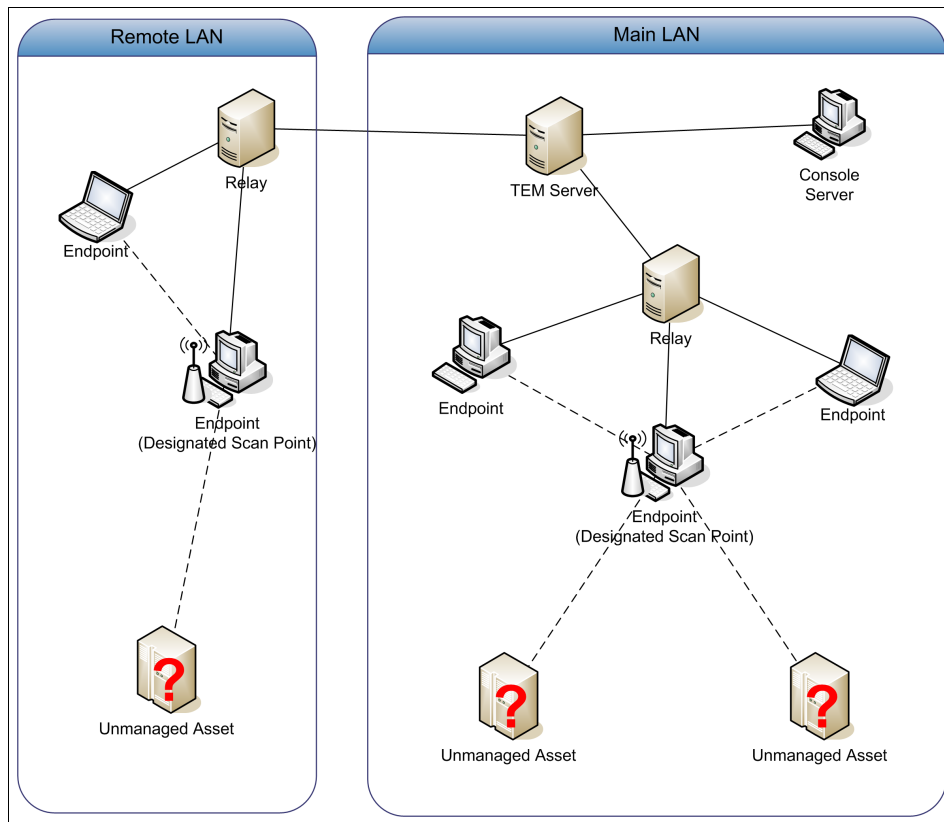


Figure 4-9 Asset discovery schematic

Potential scanning issues

You can encounter the following scanning issues:

- ▶ Network scans can potentially trigger Intrusion Detection Systems.
- ▶ Network scans can potentially cause old network devices, such as old printer network devices, to fail if scanned.
- ▶ Network scans can potentially cause personal firewalls to advise the user that a computer is scanning the local computer.
- ▶ NMAP is sometimes flagged by virus scanners as a potentially harmful tool.
- ▶ Check to ensure that your virus scanner is not set to block NMAP from running.
- ▶ If you set NMAP to scan a large network, it takes a long time and can use significant bandwidth during the scan. The default scan is the local Class C

network, which is a fast LAN. Avoid scanning large networks across the WAN with this tool.

For more information about asset management, see this website:

<http://support.bigfix.com/bes/sites/assetdiscovery.html>

4.2.3 Operational maintenance

In this section, we describe the operational aspects of the Tivoli Endpoint Manager deployment. We focus on monitoring and ensuring that Tivoli Endpoint Manager operates properly. We also mention configurations for optimizing Tivoli Endpoint Manager performance, especially for a large deployment.

Health check and monitoring

After the initial Tivoli Endpoint Manager deployment, it is advised to examine active Tivoli Endpoint Manager Support Fixlets for any potential issues. Tivoli Endpoint Manager Support Fixlets can pick up issues, such as a service not running, whether the Tivoli Endpoint Manager Server cannot connect to the Internet, whether disk space is low, and whether the system time is wrong. Most of the issues can be fixed by deploying these Fixlets to the appropriate system. Operators must also update their Tivoli Endpoint Manager components regularly. Component updates are available through Tivoli Endpoint Manager Support Fixlets. For most updates, operators apply an update Fixlet to the necessary components.

Configuring for better performance

Most of the Tivoli Endpoint Manager default configurations are set to deliver a good balance between performance and responsiveness. However, in a large deployment (greater than 15,000 Agents), a change to those default parameters can help you avoid performance problems. In this section, we describe the configuration options for each major component.

Tivoli Endpoint Manager Server

- ▶ Server heart beat interval

By default, the Tivoli Endpoint Manager Agents checks into the Tivoli Endpoint Manager Server on a regular interval that is known as a *heartbeat*. In medium to large Tivoli Endpoint Manager deployments, processing the heartbeats can consume significant Tivoli Endpoint Manager Server resources. To ensure optimal performance, the heartbeat must be increased from the default 15 minutes to 1 hour or 2 - 6 hours for larger Tivoli Endpoint Manager deployments. The heartbeat can be changed under the **File** → **Preferences** menu in the Tivoli Endpoint Manager Console.

- ▶ Configure RAID cache

Certain onboard RAID controllers are configured with the hard disk cache set to 100% read and 0% write performance. This configuration causes poor performance because hard disk drive writes are slow. The optimal configuration is 50% read and 50% write performance. For instructions to change this configuration, see your RAID controller manual.

- ▶ Move the Tivoli Endpoint Manager Server installation folder to a different drive from the SQL Server database. This instruction applies if the Tivoli Endpoint Manager Server and database are installed on the same system.

The Tivoli Endpoint Manager Server writes large amounts of files to disk while it also writes large amounts of data to the database. Place the Tivoli Endpoint Manager Server program folder on a separate physical disk drive from the SQL Server database. This change causes the hard disk drive writes to occur in parallel, which can lead to performance improvement.

Tivoli Endpoint Manager Server database

- ▶ Set the SQL server transaction log type to Simple

By default, Microsoft SQL Server uses Full transaction logging to allow for optimal data recovery. To get better performance from Tivoli Endpoint Manager, change the transaction type to Simple. See the Microsoft SQL Server documentation for instructions to perform this change.

- ▶ Schedule database maintenance

Ensure that the Reindexing Tasks are run every six hours. Look at the “Last time completed” column in the Enterprise Manager under **Management** → **SQL Server Agent** → **Jobs**. The job name is “BFEnterprise Results Table Reindexing Job”.

Tivoli Endpoint Manager Relay

The settings for the Relay also apply to the Tivoli Endpoint Manager Server.

- ▶ Disable Anti-Virus scans on the Tivoli Endpoint Manager Server and Relay folder

The Server and Relay normal operations involve creating and processing many temporary files. This activity is essential for the good performance of the Tivoli Endpoint Manager deployment, but performance can be slowed down dramatically if a virus scanner scans each file. To address this issue, configure your virus scanner on the Server computer to exclude the Server folder and all subfolders (default is C:\program files\bigfix enterprise\bes server). Configure your virus scanner on the Relay computer to exclude the Relay folder and all subfolders (default is C:\program files\bigfix enterprise\bes relay). For more information about setting this exclusion rule, see your virus scanner instructions.

- ▶ Disable file indexing

The Server and Relay normal operations involve creating and processing many temporary files. This activity can be slowed down dramatically if Windows file indexing is turned on or if the drive is set to use file compression.

Tivoli Endpoint Manager Console

- ▶ Lower the Tivoli Endpoint Manager Console refresh interval

The Console retrieves and caches dashboard information from the Server database, which can stress the bandwidth of the connection if the amount of data is large. Raise the Console refresh period from the default of 15 seconds to 30 seconds, 60 seconds, or 120 seconds for large deployments with many simultaneous Console users. The refresh rate can be changed under the **File** → **Preferences** menu in the Console. This setting is for each Console.

- ▶ Use Tivoli Endpoint Manager management rights

The amount of information transferred from the Server database grows as the number of Agents grows and can become large for medium to large deployments. The correct use of the management rights feature to restrict the number of Agents managed by each Console operator can reduce the traffic between the Server and Console.

- ▶ Avoid excessive open actions

It is a preferred practice to give each action a reasonable expiration date unless it is an enforcement action. Tivoli Endpoint Manager can normally handle hundreds of open actions. However, if thousands of actions are opened, consider either closing or condensing some of the actions.

- ▶ Delete closed and expired actions

The Console lists every action taken, including every result from every Agent. As the number of actions grows larger, the Console uses more memory to load the actions. You can delete old actions. When you delete old actions, you mark them as deleted in the database. After they are marked as deleted, the Console no longer loads them, saving memory and load time. The actions continue to be in the database after they are deleted, but they are not accessible to the Console or Web Reports.

4.3 Patch Management solution design

In this section, we describe factors to consider and approaches to use when adopting Tivoli Endpoint Manager Patch Management as a solution for patching. We look at factors to consider before the implementation, and functions that Tivoli Endpoint Manager Patch Management provides to resolve these problems.

4.3.1 Before you patch

Patching devices in an organization can be a risky business. Tivoli Endpoint Manager Patch Management provides functions and tools to enable organizations to patch devices in an efficient way while minimizing the impact to their daily business operations. Due to the diversity of requirements for different organizations, operators must investigate how to best use Tivoli Endpoint Manager for maximum patching effectiveness.

Identifying what to patch

Tivoli Endpoint Manager provides patching solutions for Microsoft Windows (English and non-English), Windows applications, Mac OS X, Red Hat Enterprise Linux, SUSE Linux, AIX, and Solaris⁶.

However, due to organization policy or other considerations, you might not want to patch all devices. Organizations are advised to have a security policy defined about patching that indicates one or both of the following rating systems:

- ▶ Ratings of patches

This method might be accepting the manufacturer rating or a provision for an internal rating of each patch.

- ▶ Ratings of systems

A system is often categorized based on location (that is, interacts with the Internet, notebook, or within a secured data center), operating system, or function (that is, application server, email server, or database server).

Based on the ratings, you might be required to patch systems within a specified time frame according to the security policy of your organization. For example, if a patch is rated as *critical*, you might be required to patch a system that accesses the Internet within 48 hours, a notebook within five days, and any data center system within 30 days. A patch with a different rating has a time frame for each category of the devices. These values are examples only and you must refer to your security policy for your requirements.

Managing patching processes

Devices can be listed under levels of risk and importance to indicate that they need to be treated differently according to the security policy of the organization.

⁶ Microsoft products: <http://support.bigfix.com/bes/misc/supportedproducts.html>

Non-English Windows: <http://support.bigfix.com/bes/misc/nonenglish.html>

Windows applications: <http://support.bigfix.com/bes/sites/updateswindowsapps.html>

AIX: <http://support.bigfix.com/bes/sites/aixpatches.html>

Mac: <http://support.bigfix.com/bes/sites/macosex.html>

Red Hat: <http://support.bigfix.com/bes/sites/rhelpatches.html>

Solaris: Spark 7, 8, 9, and 10; x86 10. Does not include vintage support and unbundled patches.

For example, a business-critical server might need to receive critical patches within 48 hours, and patching can occur outside of business hours only. Or, a workstation might need to receive critical patches within three days, and patching can occur at any time of the day. Tivoli Endpoint Manager provides manual and automatic grouping functions to effectively categorize devices for different patching operations.

Patching processes can get complicated in large organizations. Most organizations use internal approval processes to ensure that a patch is safe before they deploy it to any of their production-critical systems.

Patch testing: Although IBM tests all the patches before the patches are released as patch Fixlets, the test cannot cover all diverse platforms and software configurations. The patch works on most systems, but it can fail or create problems with installed applications on certain other systems. For this reason, you must always test every patch against a representative test environment before deploying a patch against any production systems.

Figure 4-10 illustrates a typical process for a Tivoli Endpoint Manager Patch Management solution.

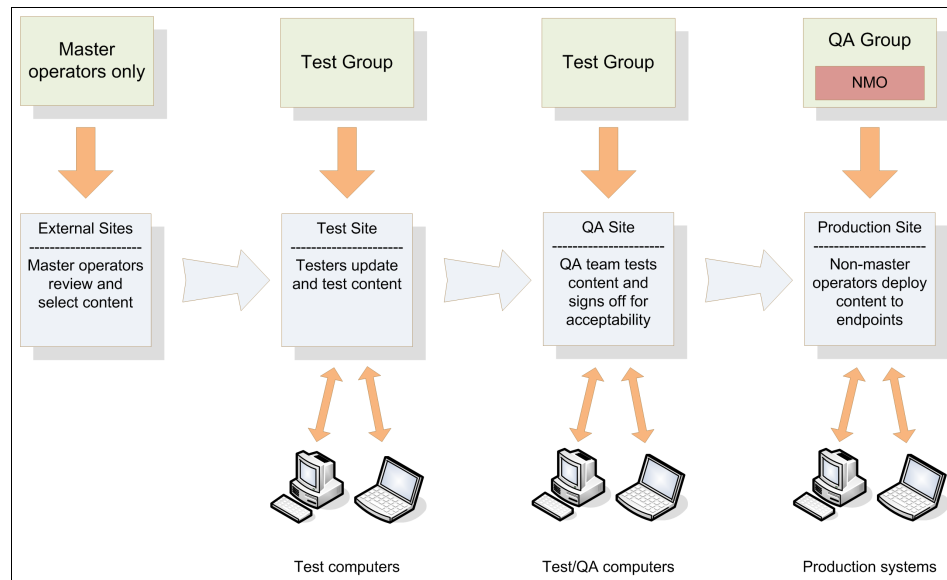


Figure 4-10 Patch deployment process

Operators are divided in groups according to their responsibilities. Computers are divided into groups according to their role in patching:

1. Master operators are responsible for determining pertinent content. They review all content and custom copy any pertinent content to the custom test site.
2. Testers test the content that the master operators provide. They might modify the Relevance and Action Scripts to meet internal requirements.
3. The Quality Assurance (QA) team reviews what the testers submit. This test is a secondary test to ensure that the content works as expected. The QA team might use the same systems as the Test group or a subset of the production systems.
4. Finally, after the QA team signs off on the content, the content is custom copied to the Production site. The non-master operators have read access only to this content and can only deploy the patches. If issues or errors are discovered, the non-master operators must have the Test group review and update the content.

4.3.2 Deploying a patch

You need to consider several details when you plan to deploy a patch:

- ▶ Applying a patch
- ▶ Determining when to patch
- ▶ Use of baseline
- ▶ Patching a server
- ▶ Rolling back a patch
- ▶ Pre-caching patch
- ▶ Patching Linux and UNIX
- ▶ Patching in an air-gapped network

Applying a patch

When applying a patch, always read the Fixlet description. It includes the following types of information:

- ▶ Details about the patch that might affect the application of the patch.
- ▶ The system might not report that the problem is *fixed* until the system is rebooted, which can skew reporting results.
- ▶ The patch might change a file association. For example, updating Adobe Acrobat Reader might change the file association of a system with Adobe Acrobat for PDF files from Acrobat to Reader.
- ▶ Tivoli Endpoint Manager might receive reports that application of a particular patch can result in abnormal behavior and must be applied after thorough testing only.

- ▶ Issues to be aware of or any potential problem that might be associated with the patch.

An operator must assess the risk level and decide whether to apply the patch to certain devices.

Determining when to patch

Deploying patches can take significant network resources. An operator must consider the following information:

- ▶ When does the patch need to be deployed?
- ▶ How long does the patch deployment window need to be?
- ▶ Will deployment of this patch affect any of the daily operations?

Tivoli Endpoint Manager provides a flexible scheduling capability that enables an operator to deploy patches whenever the operator wants.

Operators can also determine how long the patch deployment window needs to be. Mobile devices need a longer deployment window, because they do not connect to the network as often as workstations.

Confining patching to specific hours

It is possible to use an automatic grouping technique to confine patching to specific hours. This method enables the devices to start patching even they miss the scheduled patching time:

1. To use Relevance to control *automatic group membership*, set the following Agent settings on the desired systems:

```
PatchWindowStart = 21:00 Establishes when the system joins the group  
PatchWindowEnd = 23:59 Establishes when the system leaves the group
```

2. Use the following Relevance to define the *automatic group relevance*:

```
((current date as string & " " & value of setting "PatchWindowEnd"  
of Agent) as local time) > now) AND (((current date as string & " "  
& value of setting "PatchWindowStart" of Agent) as local time) <  
now)
```

3. Target the patch to the automatic group previously defined.

Wake-on-LAN (WoL)

Wake-on-LAN must be implemented to wake a system before the time required for patching. In this example, the patch window each day is between 9:00 pm and 11:59:59 pm. A WoL packet must be sent before 9:00 pm to ensure that the Agent is active and that the Agent can join the appropriate group and begin the patching process. If no patches exist, the system is free to go back to sleep after reaching the defined interval.

Use of baseline

A baseline can be used to stack up multiple patch Fixlets. The baseline can be edited so that the tested patch Fixlet can be moved and included in baselines on the production site. A baseline is ideal for ensuring that newly built systems, or systems that do not update regularly, have up-to-date patches installed. The Agent setting can be used to control applicability of the baseline. The operator can also add a Fixlet to reboot at the end of deploying the baseline.

Operators need to be aware of the following details when using a baseline:

- ▶ Never create a baseline as a Master Operator. A Master Operator baseline can multiply system overhead.
- ▶ Use a Custom Site to provide baselines.
- ▶ Do not include a corrupted patch Fixlet⁷.
- ▶ Components in a baseline are executed sequentially, so ensure that they are in the correct order.
- ▶ A baseline can hold a maximum of 250 components.
- ▶ Do not add a previous monthly baseline to the current baseline. Review all previous patches to ensure applicability, order of installation, and relevance.

Synchronizing the baseline

A synchronization of the baseline is needed whenever a component in the baseline changes, for example, an update by the Fixlet Server. For instructions to synchronize a baseline, see the following support page:

<http://support.bigfix.com/cgi-bin/kbdirect.pl?id=401>

Rebooting a system

Typically, a Windows system can accommodate multiple patches between reboots. However, if a file is in use when a patch is applied, the system is vulnerable until reboot. Although the file on storage is patched, the file actively used by the operating system in memory is not patched. An operator must consider whether a patch reboot is needed or implement a periodic reboot plan. For example, reboot workstations at 7 pm every weekday. It is suggested to add a reboot step at the end of a patch baseline.

Patching a server

Patching a server requires more consideration than patching an endpoint. The patch evaluation process and result can differ compared to a workstation. A larger organization might have a *change control policy* in place where a *change*

⁷ Corrupt patches indicate that at least one file that was supposed to be updated by the original patch on the computer is deprecated to a previous version. Applying the Fixlet can resolve this problem. Corrupt patch is only available for Windows systems.

approval is needed before any patching occurs. Operators can organize their systems into sites or groups for patching so that it is easier for the patching process to comply with the security policy. To learn about the design for patching in our fictional organization, see Chapter 7, “Phase II: Patch Management design and implementation” on page 239.

Rolling back a patch

Windows patch management has a *rollback wizard* so that you can roll back a patch. Linux patch management (Red Hat and SUSE) offers the *RPM uninstall Fixlet*, which allows the user to uninstall an RPM that is not a dependency of other installed RPMs. *Treat the rollback as a last resort*. Users must always test and run a pilot patching process before patching to the production environment.

Pre-caching patch

The pre-caching wizard can download large files, such as service packs or application patches, to the Tivoli Endpoint Manager Server cache folder and Relay cache folder to improve download efficiency. The Fixlet that is generated by the pre-caching wizard downloads the patch files into the cache. It does not deploy the patch to the endpoints.

Manual recaching

Cache files are stored in the Server or Relay cache folder. Users can download the patch file from any source and copy it to the Server or Relay manually. To enable Tivoli Endpoint Manager to recognize those files, the file name needs to be the SHA1 hash code or the file.

Patching Linux and UNIX

Both the locations and the protocols of many of the Linux and UNIX sites that are used to provide content changed over time. Certain Linux vendors require a login or accept a user license agreement. This change makes mimicking the on-demand download difficult to incorporate.

Tivoli Endpoint Manager Patch Management provides a download plug-in to cope with this problem. By using the plug-in, you can mimic Windows patching in a way where most of the files are downloaded only when required⁸. For instructions to install the plug-in, see the appropriate Linux or UNIX patch guide.

Custom patch Fixlet

Custom Fixlets enable users to customize their patch RPM package. Users can perform these tasks:

⁸ Certain files might still need manual caching, which requires that users manually download the patch from the vendor and cache it on the Tivoli Endpoint Manager Server. You might receive a “Manual Caching Required” notification.

- ▶ Choose a targeted operation system and version.
- ▶ Manually add RPM.
- ▶ Determine whether to use older or newer RPM when forced to upgrade.
- ▶ Custom switch for RPM, for example, -ivh.
- ▶ “Whitelist” preferred RPMs.
- ▶ Blacklist RPMs to avoid.
- ▶ Modify RPM preference lists.

Patching in an air-gapped network

You can patch in an air-gapped network by using the Tivoli Endpoint Manager Airgap Tool. For instructions to use the Airgap Tool, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Installing%20in%20an%20Air-Gapped%20Network>

4.3.3 Patch report

After deploying a patch, you can access a report that covers the overall progress, and the success and failure rate of the deployment. An operator can get this information as a result of the action, as shown in Figure 4-11 on page 164.

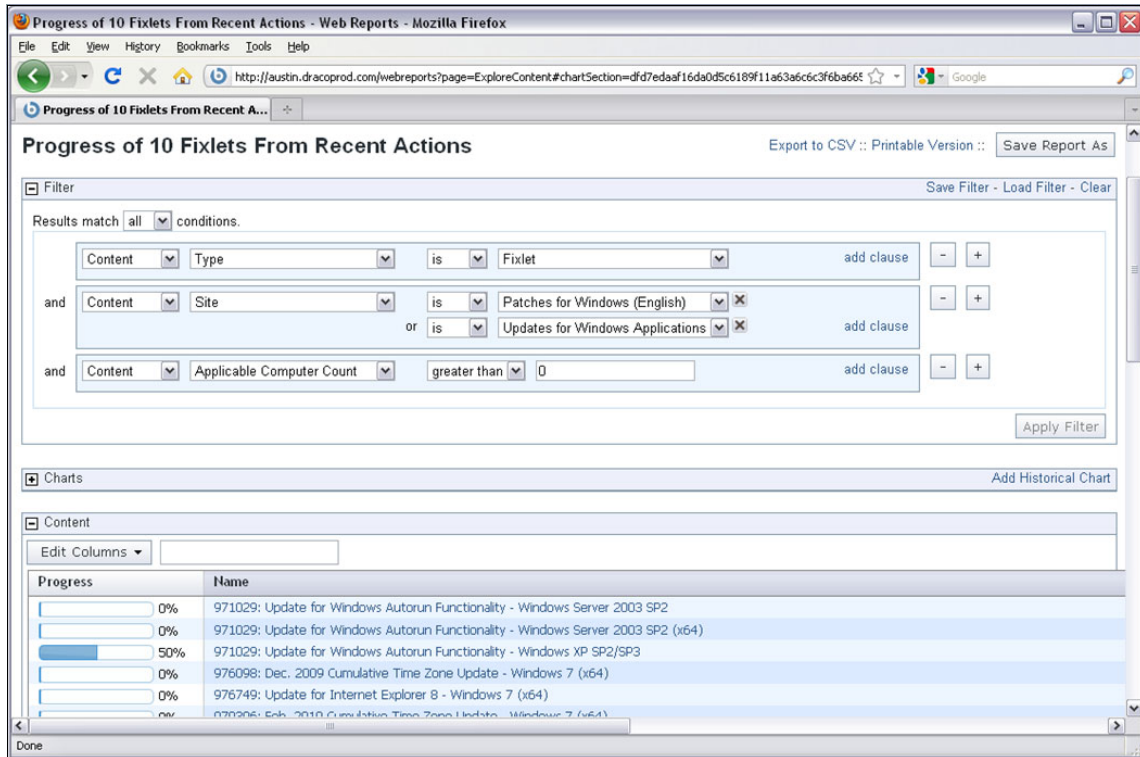


Figure 4-11 Patch progress report

The challenge arises when you work with ongoing patching, such as persistent patch baseline, because there is no *end* against which to measure. You can solve this situation by using two actions. Set the first action to run for an initial period, as documented in the *corporate security policy* documents. Then, set a second action to run on the day after the initial period.

Patches for Windows

The Windows patching site provides a comprehensive graphical report that displays a summary of patch information in your deployment through tables, graphs, and pie charts. Specifically, the Microsoft Patch Information overview report displays Microsoft patch information, deployment information, a Total Patches Needed by Severity graph, and a Severity of Relevant Patches pie chart. The overview report provides a quick summary of your Windows remediation, including the number of existing patches, broken down by severity and relevance. It also includes information for each computer, such as the average number of patches and critical patches (Figure 4-12 on page 165).

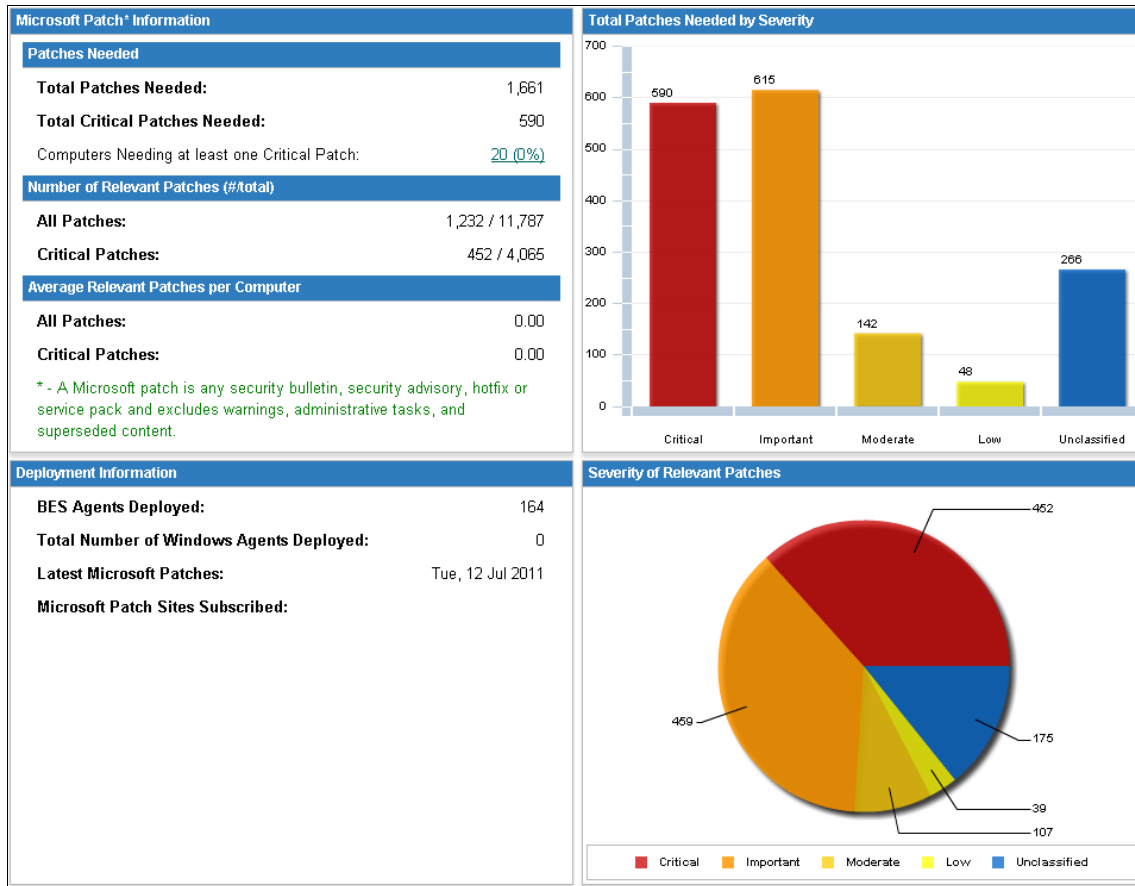


Figure 4-12 Patches for Windows overview dashboard

4.4 Security configuration management solution design

Security configuration management (SCM) is a major subset of the discipline of security and compliance management. In this section, we introduce a guide that follows the preferred practices for the SCM tasks of an organization.

One of the Tivoli Endpoint Manager offerings is called IBM Tivoli Endpoint Manager for Security and Compliance. This offering includes several subpackages. One subpackage is a set of patch management Fixlet Sites for various platforms (2.3.2, “Patch management” on page 47). Another subpackage is the Tivoli Endpoint Manager Security and Compliance Analytics reporting solution (2.3.4, “Security Compliance Analytics” on page 56). Additionally, the

product provides a comprehensive library of technical controls that can help you achieve security compliance by detecting and enforcing security configurations.

Tivoli Endpoint Manager provides predefined preferred practices that meet US *Federal Desktop Configuration Control (FDCC)* regulations and *Defense Information Systems Agency Security Technical Implementation Guides (DISA STIGs)*.

The checklists are ready to be used in an established Tivoli Endpoint Manager environment immediately after installation. The greatest advantage of these checklists is that they can be customized to the specific needs of your organization.

4.4.1 Security configuration management Fixlet Sites

In 3.1.1, “Fixlet Server” on page 65, we introduced the Tivoli Endpoint Manager Fixlet Server. This cloud-based service, managed by IBM, is used as a main source for Tivoli Endpoint Manager content. All Tivoli Endpoint Manager Servers must be able to connect to the Tivoli Endpoint Manager Fixlet Server to download the latest data. This model allows IBM to deliver updates and new features directly to an environment, as needed.

Isolated networks: Tivoli Endpoint Manager assumes connectivity between a Tivoli Endpoint Manager Server and a Tivoli Endpoint Manager Fixlet Server on the Internet. However, IBM knows that sometimes connectivity is not possible. Certain organizational regulations or technical obstacles might prevent this Internet access requirement. Tivoli Endpoint Manager offers a solution for providing Fixlets and all needed files for isolated, *air-gapped* networks. For more information, see the following IBM developerWorks:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Installing%20in%20an%20Air-Gapped%20Network>

Figure 4-13 on page 167 shows a high-level view of the Tivoli Endpoint Manager Fixlet Server content. The content is organized into *Fixlet Sites* that Tivoli Endpoint Manager Servers can download for the operators. The available Fixlet Sites are displayed in the Tivoli Endpoint Manager Console. An operator can review the site content by expanding the appropriate node in the content tree. Fixlet Sites are containers that hold smaller artifacts, such as Fixlets, Tasks, Analysis, or other items (dashboards or wizards).

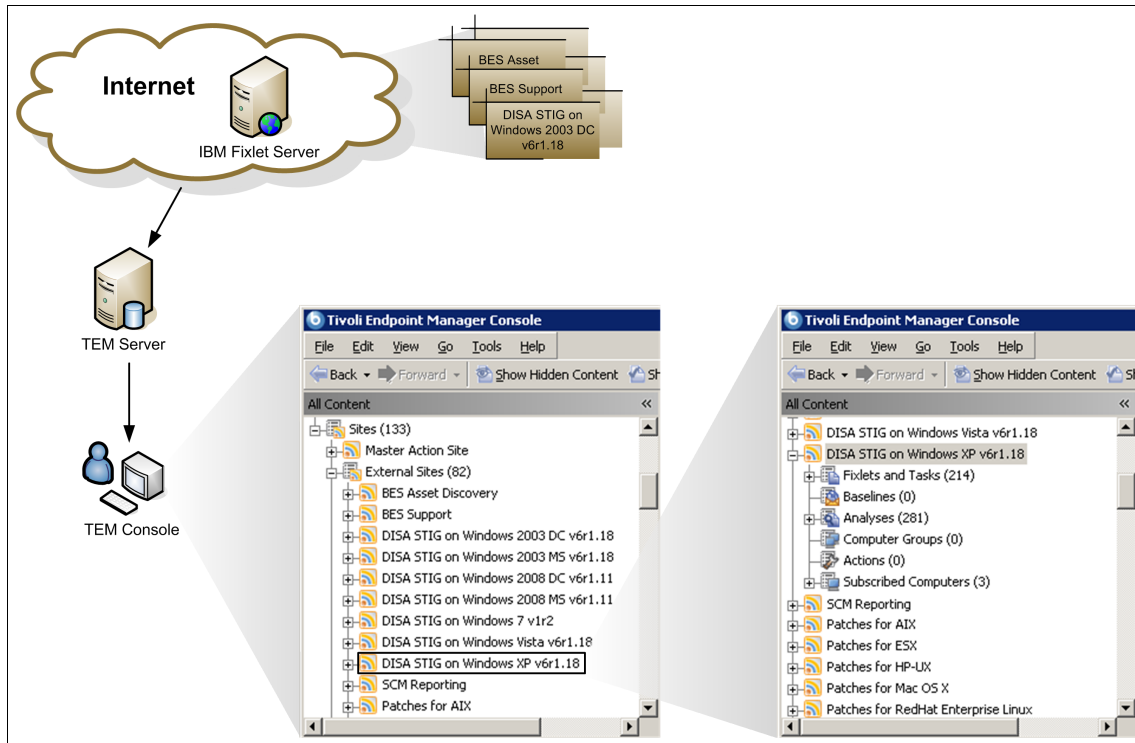


Figure 4-13 Fixlet Sites that are visible in the Tivoli Endpoint Manager Console

The lower-right side in Figure 4-13 shows a Tivoli Endpoint Manager Console snippet. We selected a sample Fixlet Site that represents the checklist for DISA STIG on Windows XP v6r1.18. We can see what is included in this checklist site. Figure 4-14 depicts a diagram that represents the Fixlet Site.

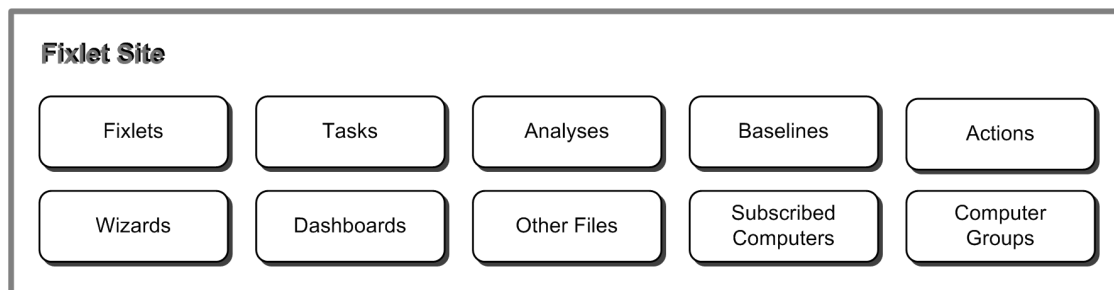


Figure 4-14 Fixlet Site subcomponents

The *Fixlet Site* can contain the following artifacts:

- ▶ **Fixlets:** Responsible for evaluating a single aspect of the operating system, looking for a noncompliance according to a predefined policy. For example, there might be a Fixlet that checks whether the maximum password age setting is at least 90 days.
- ▶ **Tasks:** Responsible for repetitive actions or maintenance activities. For example, there might be a Task that updates Tivoli Endpoint Manager Agent to the latest available version.
- ▶ **Analysis:** Responsible for gathering actual data from the operating system. For example, there might be an Analysis that retrieves the value of the system maximum password age setting.
- ▶ **Wizards:** Responsible for providing a user interface for automating actions. For example, there might be a wizard that automates the copy process of the Fixlets between Fixlet Sites.
- ▶ **Dashboards:** Responsible for providing a user interface to display certain aspects of the system. For example, an operator can use a dashboard that displays the Tivoli Endpoint Manager environment information. This information includes the number of Tivoli Endpoint Manager Agents that are running and Tivoli Endpoint Manager Relays.
- ▶ **Other files:** Responsible for providing any other files that might be useful from the Fixlet Site perspective. For example, there are files that can provide additional evaluation and generate results that are later used by an operator for Analysis.
- ▶ **Baselines:** Represent groups of Fixlets that allow for a high level of automation. For example, a set of patch Fixlets can be grouped into a baseline.
- ▶ **Actions:** Represent a collection of Tasks or Fixlets, which can be executed in case of remediation. An Action represents the activity that happens in the environment. For example, a particular Action can be designed for patch installation.
- ▶ **Subscribed computers:** Represent a list of computers that evaluate the content from the site.
- ▶ **Computer groups:** Represent a list of computer groups defined for this site that evaluate the content of the site.

All of these types of objects belong to the Fixlet Site. However, the last four types are called *user-authored artifacts*. They are created and managed by Tivoli Endpoint Manager operators.

4.4.2 Security configuration management Fixlet design

For security configuration management (SCM), Tivoli Endpoint Manager provides a special type of Fixlet and Analysis. With the built-in functionality, you can logically link the Fixlet and Analysis. This link creates a pair that can provide complete compliance information about the verified aspect. We evaluate the example of the maximum password age setting. Figure 4-15 depicts a graphical representation of the SCM Fixlet and Analysis relationship.

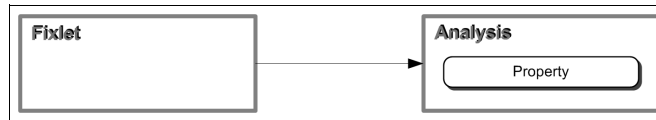


Figure 4-15 Fixlet and analysis relationship

Why do we need a link between those objects? To answer that question, we need to evaluate each object separately to see what information it can provide:

- ▶ A Fixlet evaluates the system setting against a specific policy value. For this example, the Fixlet checks whether the maximum password age value is equal to or less than 90 days. If there is a violation of the policy, the Fixlet becomes relevant, which means that noncompliance is detected. The Tivoli Endpoint Manager operator receives a warning in the Console. It is not possible to deliver the value that exceeds the operating system setting, whether it is 61 or 200 days. Both values are considered out of policy, but the system with the higher deviation in value might need to be fixed first. The Fixlet defines a link to the Analysis to retrieve the actual setting.
- ▶ An Analysis is responsible for the retrieval of the actual setting from the operating system. For the maximum password age, the Analysis returns a number to the Tivoli Endpoint Manager Server. The Analysis does not evaluate. Without the Fixlet to evaluate, it is impossible to know whether the actual setting represents a noncompliance at all.

Both objects linked together provide the full information to determine the compliance. This collaboration is not required, for example, in a Patch Management scenario. In that scenario, a Fixlet is sufficient to discover whether the patch is missing or installed. An Analysis for a single patch is not needed.

From the SCM perspective, Analysis and Fixlets that are provided within Fixlet Sites must always exist together. This coexistence is important for reporting. In 2.3.4, “Security Compliance Analytics” on page 56, we described a type of report that presents the compliance state and the actual values collected by Analysis from the operating system. That report is called *Check Results*.

Fixlet parameters

Typically, a Fixlet provides information about a single, atomic aspect on an inspected system. If there is a need to customize, for example, to provide a parameter to the Fixlet evaluation, use a *Task*. Tasks are responsible for setting properties on a managed endpoint. Those properties can later be read by the Fixlet and interpreted by a Relevance statement, influencing the final Fixlet state.

The challenge with this approach is that it requires that you maintain the Task object and execution. This maintenance can require additional effort for an operator and can create an additional load on the Tivoli Endpoint Manager Server and Tivoli Endpoint Manager Agent.

Operators mostly prefer to have a simpler solution that includes homogeneous parameter values across all computers. This approach matches the use cases described by most organizations better. It can simplify Analysis, and it allows for a more precise representation of what is measured by a specific check. This content-based solution can be based on embedding code within the Fixlet description code. This code can modify the Relevance of the Fixlet to match the parameter values that you want. This code can be coupled with an HTML form for the user to enter the information that you want. In addition to enforcing value homogeneity, this solution can provide these benefits:

- ▶ Simplify the control of Relevance, because it no longer needs to check the conditional site setting.
- ▶ Eliminate the need for a separate parameterization Task.
- ▶ Provide a better user interface than the built-in action setting prompt.
- ▶ Permit validation of user-provided values.

Figure 4-16 on page 171 shows the description of a sample Fixlet with a focus on the Fixlet parameter form, which allows a user to define values directly in the Fixlet object. After a user clicks **Save**, the Fixlet is automatically updated and redistributed among all computers that already have this Fixlet assigned.

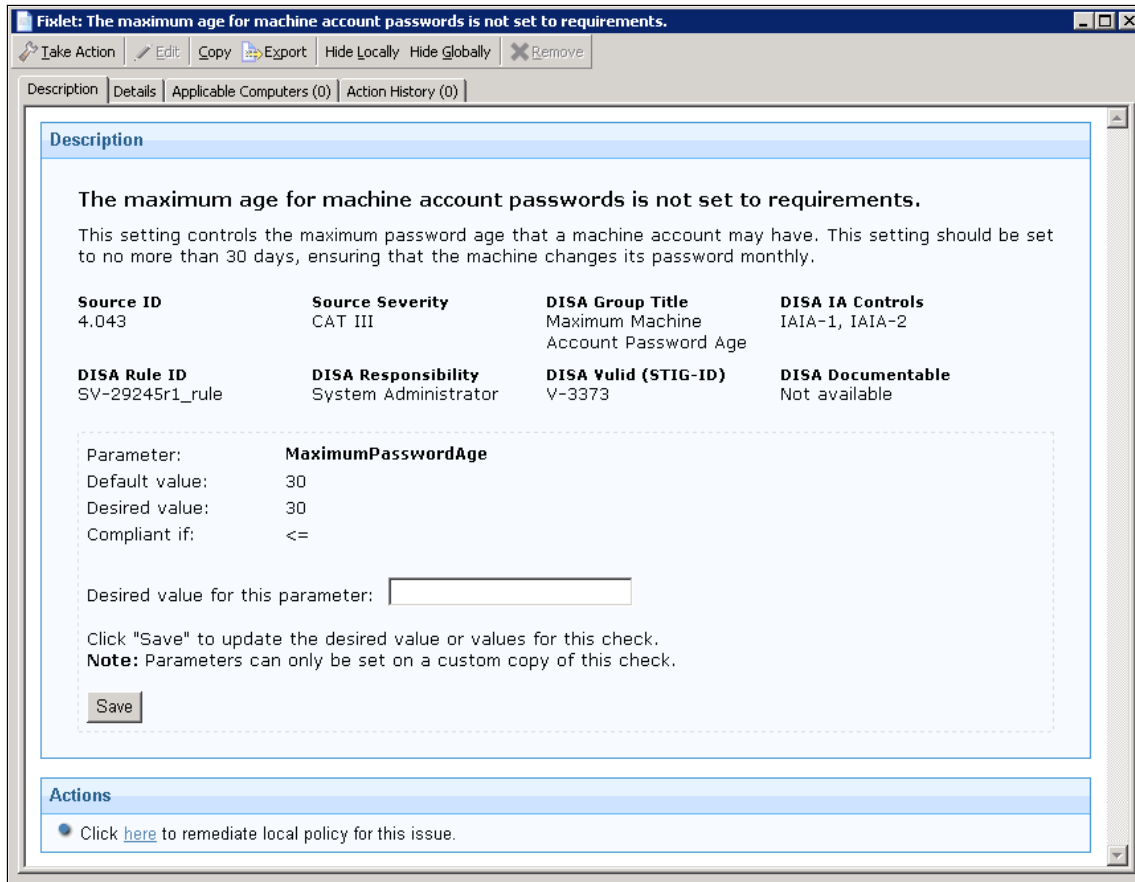


Figure 4-16 Fixlet description with parameter form

Unlike the current parameterization mechanism, users cannot parameterize controls until the custom copy is created. Parameterization within the external site is not possible.

Technically, the link between the Analysis and the Fixlet (introduced in 4.4.2, “Security configuration management Fixlet design” on page 169) is at the Fixlet parameter level. The link is at the Fixlet parameter level for the following reasons:

- ▶ The Fixlet defines a parameter that a user can configure in the description. For example, the maximum password age is set to 90 days.
- ▶ The Fixlet checks the compliance verification against the value of the parameter that is given by the user. For example, the maximum password age extracted from the operating system is compared against the parameterized value of 90 days.

- ▶ The Fixlet can remediate noncompliance by changing the system settings to a value defined in the Fixlet parameter. For example, the remediation action sets the maximum password age to the value of 90 days.
- ▶ A correlated Analysis provides an actual value of the system setting. For example, an Analysis returns the actual maximum password age setting. In our example, the value is 100 days (noncompliance). Later, after remediation, the value is 90 days.

Fixlet applicability

In the previous section, we explained the link between the Fixlet and Analysis. That solution introduced the capability to provide a more detailed compliance message to the Tivoli Endpoint Manager operator. The real challenge in the Fixlet solution is that this approach can provide only two kinds of information.

A Fixlet can either be relevant or not relevant. That limitation makes it impossible to provide an answer to the question: Why is the Fixlet not relevant? Is the Fixlet not relevant because the system is compliant, or because there is no reason to check that aspect on that particular operating system? We look at a simple example that happens frequently.

Sometimes, systems are subscribed to the wrong Fixlet Site by mistake. If we subscribe a Linux system to the Windows specific site, the Linux Tivoli Endpoint Manager Agent reports all those Fixlets as compliant (not relevant). Although this information is a true statement, in fact, those Fixlets are not applicable to that platform. From the reporting perspective, if there are 100 Windows machines and 100 Linux machines subscribed to the site, and all Windows machines are not compliant, the overall compliance score is 50% without applicability checking. With the applicability functionality, all Linux machines are treated as nonapplicable, thus the noncompliance level is reported correctly at 100%.

To obtain data on Fixlet applicability, you need to create a link between each check Fixlet and *applicability* Fixlet that is relevant if the check is applicable and not relevant otherwise.

Many Fixlets typically share applicability relevance based on operating system or application presence. Therefore, only a limited number of additional sentinel Fixlets are required to be shared by many check Fixlets. These Fixlets are likely hosted on a separate shared site, to which all computers are subscribed. This approach has the advantage of keeping applicability evaluation at the endpoints, and this approach can be used by the reporting systems.

Figure 4-17 on page 173 depicts a graphical representation between Fixlet, Analysis, and the Applicability Fixlet relationship.

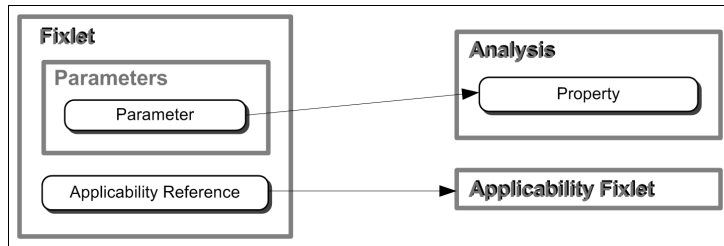


Figure 4-17 SCM Fixlet, Analysis, and Applicability Fixlet relationship

From a technical point of view, each Applicability Fixlet defines a special hidden identification field. This field is not visible in the Tivoli Endpoint Manager Console to the operator. The field value can hold any text, but that value must be unique within the entire Tivoli Endpoint Manager environment. When a Fixlet references an Applicability Fixlet, it uses that identifier. There are several Applicability Fixlets available within Tivoli Endpoint Manager. They target various operating systems. To avoid duplication of Applicability Fixlets in multiple Fixlet Sites, a single Fixlet Site is used. This Fixlet Site is called *SCM Reporting*. In addition to providing the Applicability Fixlets, it also provides additional tools for security configuration management.

Various locations: Although the SCM Reporting site contains all the Applicability Fixlets that can be used by Fixlets, the Custom Sites can contain Applicability Fixlets, too. The operators must pay close attention when they create Custom Sites.

If a Fixlet wants to use an Applicability Fixlet, another hidden field must be defined. This field is called the *applicability reference field*, and its value must contain the same text as the Applicability Fixlet. Table 4-3 on page 174 summarizes how the Applicability Fixlet and the check Fixlet results can be interpreted for the overall compliance state.

Table 4-3 Fixlet compliance state with applicability checking

		Check Fixlet		
		No results ^a	Not relevant	Relevant
Applicability Fixlet	No results	Not applicable	Not applicable	Not applicable
	Not relevant	Not applicable	Not applicable	Not applicable
	Relevant	Compliant	Compliant	Not compliant

a. A *no results* state occurs while the Fixlet is assigned to the system, but the data from the Tivoli Endpoint Manager Agent is not received yet by the Tivoli Endpoint Manager Server.

At the time of writing this book, the only reporting tool that can interpret the results of Applicability Fixlets is the Tivoli Endpoint Manager Security and Compliance Analytics reporting system. We introduce it in 2.3.4, “Security Compliance Analytics” on page 56.

4.4.3 Customizing Fixlet Sites

After the Fixlet Site is available to the operator, the process of adopting the checklist to the organizational requirements can begin. In 4.4.2, “Security configuration management Fixlet design” on page 169, we introduced security configuration management Fixlet design. With this approach, the Fixlets can set evaluation parameters by directly editing the form inside the description. That functionality makes the management process easier, but it adds a specific requirement. All changes can be made only if the Fixlet exists in the Custom Site. After this requirement is met, it is possible to edit the Fixlets and distribute the updated versions to all subscribed endpoints.

To allow Fixlet Site customization, a custom copy of the site or of individual Fixlets must be created. Use one of the following guidelines:

- ▶ Use the checklist in a form provided by IBM.
- ▶ Use the checklist in a form provided by IBM but with changes in individual Fixlets, so that the compliance policy requirements of the organization can be met.
- ▶ Use parts of various checklists provided by IBM with changes in individual Fixlets. Checklists must target the same operating system.

Depending on the internal policies of the organization, the process of creating custom Fixlet Sites can differ. In general, the process needs to follow one of the guidelines. We evaluate each guideline in more detail.

Whole Fixlet Site adoption

Tivoli Endpoint Manager for Security and Compliance is delivered as a complete solution. Among various features, it includes a set of predefined checklists created according to a set of federal standards. Those checklists were briefly introduced in “Security configuration management checklists” on page 54. If an organization is willing to follow the federal guidance without any customization, the obvious solution is to use the Fixlet Site authored by IBM.

Although this approach is acceptable, we encourage you to create a Custom Site and move the complete Fixlet Site content to that Custom Site. The Tivoli Endpoint Manager delivery model assumes Internet connectivity and automatic downloads of the updated site content directly from the Tivoli Endpoint Manager Fixlet Server. Therefore, Fixlets can be updated without operator intention. Keeping all Fixlets in the Custom Site can prevent uncontrolled updates. It also allows the process to be executed at a time that is more convenient for the organization.

Selective Fixlet Site adoption

The adoption of a complete Fixlet Site, although simple, is used in a minority of actual client cases. An organization defines its own policies, which, even based on federal standards, are limited or updated to match particular functional and business requirements. Then, you choose only those elements of the Fixlet Site that match the needs of your organization.

An adoption of only portions of the site results in your closely reviewing the functionality of each Fixlet, so that matching items only are moved to a previously created Custom Site. Because this approach is linked to the SCM solution, a Fixlet can be linked to the Analysis, thus it is a requirement to copy the Analysis components, as well. In addition, operators can change the default Fixlet parameters to better align with the security requirements of the organization.

Fixlet Site content compilation

Tivoli Endpoint Manager supports implementations for various security standards, including DISA STIG and the FDCC. Even though many standards are similar, there are obvious differences. It might be worthwhile to select subsets for Fixlets and Analysis from each of the sites and prepare a single Custom Site that can better match the needs for the organization.

This case is similar to the Selective Fixlet Site adoption option, but in this case, the Fixlet sources come from multiple Fixlet Sites. The operator must be careful to copy the Fixlet and the correlated Analysis to ensure that the compliance state can be properly reported later in the Tivoli Endpoint Manager Security and Compliance Analytics tool.

Each solution presents a major challenge in creating many copies of Fixlets and Analysis components. The single copy action might be effective for a few objects. However, when managing several hundreds of Fixlets, you need a special tool, the *copy wizard*.

4.4.4 Copy wizard

The creation of a Fixlet Site copy can be a time-consuming activity. It requires copying the Fixlets and also the Analysis objects that are defined in the Fixlet Site. If an organization is willing to adopt an entire Fixlet Site in the form that IBM provides, the process is simple. The challenge arises while there is a need to selectively copy a subset of the site, or combine a final solution from various sites.

The SCM Fixlet and Analysis components, even enhanced with the logical links between them, are compatible with an earlier version. Those objects can exist independently and there are no Tivoli Endpoint Manager platform restrictions that might prevent splitting them. To ease the pain of the customization process with SCM Fixlets and Analysis, IBM provides a wizard that enables operators to complete the task with a few mouse clicks.

The wizard is called the *Create Custom Checklist Wizard*, and it is provided as part of the SCM Reporting Fixlet Site. Figure 4-18 on page 177 shows the active wizard in the Tivoli Endpoint Manager Console.

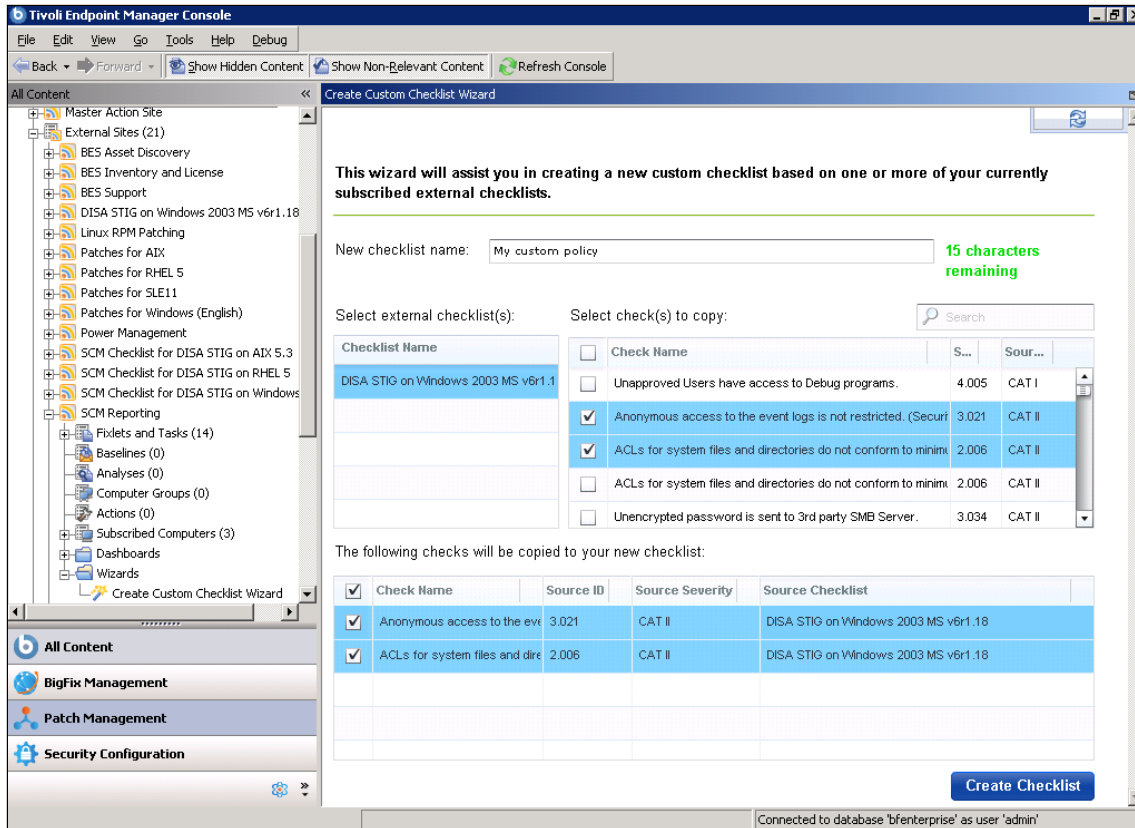


Figure 4-18 Create Custom Checklist Wizard in the Tivoli Endpoint Manager Console

To create the Custom Site, the operator needs to provide the new Custom Site name and select all sites and Fixlets to be part of the new site. After the operator clicks Create Checklist, the wizard creates the site and copies all the Fixlet and Analysis components. For additional information about the Create Custom Checklist Wizard, see “Cloning the SCM checklist from the Custom Checklist Wizard” on page 308.

Continuous development: The Create Custom Checklist Wizard tool is constantly enhanced. The user interface might change for better usability or be modified to contain more functionality. Even if the interface differs, the underlying concept of helping the operator to create Custom Sites remains the same.

4.4.5 Subscribing endpoints to sites

After the new site is created and updated to meet the requirements of the organization, Tivoli Endpoint Manager Agents must be subscribed to it. An operator can use Relevance statements to subscribe machines or create a computer group that can be subscribed to the Fixlet Site.

Considering the special Applicability Fixlet introduced in “Fixlet applicability” on page 172 and the functionality it provides, you might think that it does not matter which endpoints are subscribed to the site. Although nothing breaks and the compliance reporting is accurate (due to the *not applicable* machines), it is still advised that you subscribe those systems that must evaluate the specific content only. A Tivoli Endpoint Manager Agent can compute compliance results as soon as possible, assuring a high level of workstation security.

4.4.6 Analysis activation

The SCM solution introduced Fixlets and the correlated Analysis. Providing better environment compliance is useful, but it also can affect the Tivoli Endpoint Manager Server infrastructure.

Because Fixlets can return only one value out of two (true or false), they are designed to work with high performance results. The Analysis components, introduced inside the checklists, can return actual data that is retrieved from the operating system. As a result, the amount of data to be transferred back to the server and then stored in the database can be higher. To keep the Tivoli Endpoint Manager environment performance at peak performance levels, implement the following strategy.

It is suggested to start the evaluation process for the endpoint compliance posture *without* using any of the Analysis components in the activated state. At this time, the operator must focus on noncompliant elements only. And if appropriate, activate the Analysis components for those noncompliant endpoints to gather more data. After the issues are remediated, the Analysis components must be deactivated again.

4.5 Security and compliance analytics solution design

The Tivoli Endpoint Manager Analytics platform introduced in 3.1.8, “Analytics” on page 84 and 3.2.7, “Tivoli Endpoint Manager Analytics” on page 102 offers, at the current state, the functionality of reporting on security and compliance results. The following sections describe how to design a Tivoli Endpoint Manager

Analytics solution in conjunction with the previously established Tivoli Endpoint Manager environment.

4.5.1 Extract, transform, load process

The Tivoli Endpoint Manager Analytics platform imports data from the main database. Before storing event information for data warehousing, the *extract, transform, and load* (ETL) processing must occur, which can be complex. Various conditions can cause the process to be fast or slow. Technically, an ETL process is responsible for two types of activities:

- ▶ Category 1 ETL/Historical
 - This activity handles the historical reporting data set. This activity has the most impact on the Analytics database size over time.
- ▶ Category 2 ETL/Point-in-time
 - This activity handles a point-in-time data set. This activity typically has little impact on the overall database size, but it can affect the ETL runtime duration.

According to these statements, you can define equations to better understand the actions that occur for each type of ETL process. See Table 4-4 for details.

Table 4-4 Scale equations for ETL process

ETL type	Equations of actions
Historical	<ul style="list-style-type: none"> ▶ (number of computers) x (number of checks of subscribed SCM sites that changed) x (number of ETL processes executed) ▶ (number of computer groups defined in SCA) x (number of ETL processes executed) ▶ (number of checklists defined) x (number of computer groups defined in SCA) x (number of ETL processes executed) ▶ (number of checks defined in subscribed SCM sites) x (number of computer groups defined in SCA) x (number of ETL processes executed) ▶ (number of computers) x (number of ETL processes executed) ▶ (number of checklists defined) x (number of computers) x (number of ETL processes executed)
Non-historical or point-in-time	<ul style="list-style-type: none"> ▶ (number of computer properties defined in SCA) x (number of measured values analysis) x (number of changes detected per subscribed system)

To minimize the associated impact of data over time and optimize the ETL process duration, consider the following principles:

- ▶ Reduce the number of checklists used with the Tivoli Endpoint Manager system. It does not matter whether computers are subscribed to the Fixlet Site. The existence of the site is sufficient to influence the ETL process.
- ▶ Reduce the number of computer groups within Tivoli Endpoint Manager Analytics.
- ▶ Reduce the number of checks defined in Tivoli Endpoint Manager.
- ▶ Reduce the number of site subscriptions of checklists to machines that are intended to be evaluated.

The ETL process can be scheduled during off-peak hours. Although it must not disrupt the Tivoli Endpoint Manager Server function, there is always a minimal impact (database reads). The initial ETL can take considerable time. The actual duration depends on many variables. After the initial import, the ETL captures incremental changes based on the scheduling of the ETL. The duration of the actual ETL varies based on the considerations and the rate of change across the data retrieved.

There are scheduling variables to consider. The less that ETLs are run, the longer the duration to run the ETL. Conversely, running ETLs more often likely decreases the duration required to run an ETL, but likely also uses more space because more trend data is recorded. The correlation of duration and frequency versus size and storage requirements is not necessarily predictable in a single equation and varies based on the Tivoli Endpoint Manager environment and business practices or conditions. It is suggested that after you run the initial ETL process, attempt to run it every day over the same constant of data sets (number of checklists, number of checks, and number of computers). Measure the increment to the database size daily, as well as the duration of the import process. You can view this information on the import window in the management section of the web interface.

To obtain more statistical data, consider the following test. Perform daily ETLs over a period of one week, then lower the ETL frequency to two days and record the same metrics over two weeks. Lower the ETL frequency to three days and record the same metrics over three weeks. Gathering the same statistics (increment in the database size and duration of the ETL process) over variations in ETL frequency while maintaining the same (or close to) data set of checks, checklists, and computers can help define the rate of change over time. This metric can then be used to determine the projected size of the database based on various ETL frequencies. After this data is captured and recorded, an organization can further understand the balance of storage requirements versus ETL frequency and better calculate the duration of the ETL process.

Additional ETL process considerations

Certain function (after it is defined) cannot be used or referenced until after another ETL import occurs. Certain function is not available while an import is underway, as well. The following items are affected:

- ▶ New or redefined computer groups cannot be referenced until after the next import.
- ▶ New or redefined computer properties cannot be referenced until after the next import.
- ▶ New or redefined exceptions cannot be referenced until after the next import.
- ▶ Exceptions cannot be defined when an import is in progress.
- ▶ Computer groups cannot be defined when an import is in progress.
- ▶ Computer properties cannot be defined when an import is in progress.
- ▶ If the computer scope of a user is defined to a computer group that is based on a computer property, and that underlying computer property must be changed, another import is required to reflect the change to the computer scope of the user that uses the computer property.

4.5.2 System and hardware guidelines for Analytics

While considering the software and hardware setup for a Tivoli Endpoint Manager Analytics solution (Table 4-5), it is crucial to understand that the size of the managed environment only partially influences the final Analytics setup.

Table 4-5 *Tivoli Endpoint Manager Analytics system and hardware guidelines*

Requirement	Solution	Notes
Processor	One core (> 2 GHz) per concurrent user	N/A
Disk space	10 - 20 MB for each single managed system for each year	This requirement depends highly on the number of checks defined in the subscribed SCM Fixlet Sites and the number of ETL processes executed during the assumed time frame. Follow the Microsoft best practices for the SQL Server configuration ^a , trying to reach the write latency at about 2 - 3 ms and read response time at about 20 ms. These results depend highly on the configuration of the storage system.

Requirement	Solution	Notes
Memory	Memory for operating system plus memory for Tivoli Endpoint Manager Analytics platform (about 1 - 2 GB) plus 200 KB for each managed computer	An SQL Server database can use significant memory during the import process.
Network	One or 10 Gb network interface card (NIC)	N/A
Operating system	64-bit edition of Microsoft Windows 2003, 2008, or 2008 R2	N/A
Database	Microsoft SQL Server 2005 or 2008	N/A

a. SQL Server Best Practices article:

<http://msdn.microsoft.com/en-us/library/cc966412.aspx>

Tivoli Endpoint Manager Analytics is designed as a data warehouse solution. Thus, it is crucial to ensure that the I/O system can adequately handle data reads and writes.

SQL transactions

All transactions at import use snapshot isolation. From that point, SQL Server uses a prepared batch statement, and this statement is fed into a database trigger. The trigger performs two functions:

- ▶ To read the applicable row to see whether an update is required
- ▶ To insert or update as needed

At the time of import, the *tempdb*⁹ database might be heavily used, and from a read and write perspective, this process can be considered 50% for reading and 50% for writing. Additionally, during the end of the import process, where aggregates are computed and stored (actual warehousing occurs), there is a large read impact (approximately 90%) and less of a write impact (approximately 10%). During normal activities to run reports, a large read impact is seen and nearly no write operations. To properly balance the Analytics database, optimize it roughly to 50% for reading and 50% for writing with a slight preference for reading based on application usage.

⁹ The tempdb system database is used to hold all temporary tables and temporary stored procedures. The tempdb is a global resource; the temporary tables and stored procedures for all users connected to the system are stored there.

Consider and apply the following specifications, depending on the size of the managed environment:

- ▶ Dedicated partition for the SQL Server transaction log
- ▶ Dedicated partition for the SQL Server database file
- ▶ Dedicated partition for the SQL Server *tempdb* database
- ▶ Organization of each partition in the RAID matrix
- ▶ High-speed hard disk drives

Storage area network and RAID configurations

A storage area network (SAN) is a dedicated network that provides specialized storage. Devices attached to the network create a container for data, but it is impossible to access each device separately. A SAN is managed through its own system; therefore, you can create logical devices for particular purposes. As a result, you can define logical units that are identified by a *logical unit number* (LUN). LUN is sometimes used as the term for a particular device. With a configured storage area network, you might assign a LUN to a physical disk so that the I/O processing on that device can be used for one purpose, such as for an SQL Server database. SQL Server performance depends heavily on read and write access to physical disks, so RAID fundamentals are as important as other areas when deploying on a SAN.

SQL is an I/O-expensive application. The highest priority for performance must be the SQL Server disk configurations. Microsoft recommendations and best practices suggest that SQL Server log files must be separated from data files at a physical disk level. In a SAN environment, allocate the database log files to LUNs with dedicated physical disks (not shared with other LUNs), including those disks that are used for other database log, data, or index files. A write response time of 2 - 3 ms and a read response time of under 20 ms are the goals for this configuration per Microsoft best practices. Any physical disks that are shared between servers or applications must be isolated. Different servers that run different applications with different I/O characteristics can cause problems.

Within large database servers that use SANs, in the best case scenarios, *tempdb*, logs, and database file subsystems must be isolated from physical disk contention. Because large database engines typically use *tempdb* and transaction logs heavily, separate SQL *tempdb* and transaction log files to their own RAID subsystem, preferably RAID 10.

4.6 Conclusion

In this chapter, we describe the solution design and implementation that starts with advanced planning considerations and actual implementation planning. We

then continued with the actual implementation. We looked at preferred practices for maintaining high performance when Tivoli Endpoint Manager operates in a large-scale deployment. We also explained how to further fine-tune the performance of Tivoli Endpoint Manager, especially in a large-scale deployment. We explained Patch Management, SCM, and Security Compliance Analytics in detail in terms of design considerations and preferred practices.

This chapter concludes the first part of this book. Part two of this book explains how our solution can be applied to an actual client situation.



Part 2

Customer environment

In this part of the book, we describe a scenario about a fictional financial institution. We explain the implementation of endpoint security and compliance management with IBM Tivoli Endpoint Manager for Security and Compliance.



Overview of scenario, requirements, and approach

In this chapter, we introduce a business scenario related to a fictional *financial accounting company*. This organization uses the IBM Security Framework and IBM Security Blueprint to implement an endpoint security and compliance management solution with Tivoli Endpoint Manager.

We cover the following topics:

- ▶ “Organization profile” on page 188
- ▶ “Business vision” on page 194
- ▶ “Business requirements” on page 195
- ▶ “Functional requirements” on page 198
- ▶ “Design approach” on page 203
- ▶ “Implementation approach” on page 205

Important: All names and references for organization and other business institutions used in this chapter are fictional. Any match with a real organization or institution is coincidental.

5.1 Organization profile

The financial accounting company is a leading financial services organization with headquarters in London. It operates on the continents of Europe, Asia, and the Americas. The financial accounting company offers private banking and insurance products.

The financial accounting company started as a privately held organization and was acquired by a large universal bank in England. This bank made an initial public offering for the financial accounting company six months ago on the New York Stock Exchange.

The financial accounting company is expanding its worldwide operations quickly, opening many new branch offices in the last two years. The company currently employs approximately 80,000 employees and 40,000 contractors. The company provides insurance products to more than 800,000 households and performs wealth management for more than 135,000 private investors.

Information security sample: In the following sections, we describe organization information that is relevant to an endpoint security and compliance management solution. It is not intended to provide a complete description of the organization, and the subsequent sections do not cover all necessary activities related to information security in detail.

5.1.1 Current IT infrastructure

The financial accounting company uses mirrored and resilient data centers in London for the European and worldwide operations. These centers also host the mainframe system and a mirrored storage area network (SAN) for 250 TB of data. The organization hosts a small local data center in each main office on each continent where it operates. The data centers are in these cities:

- ▶ Newark to support North American operations
- ▶ Sao Paulo to support Latin American operations
- ▶ Taipei to support Asian operations
- ▶ Zurich, because of the regulatory restrictions in exporting banking customer data outside of Switzerland

The financial accounting company has four satellite offices to support strategic sales and geographical expansion:

- ▶ Toronto, Canada
- ▶ Tokyo, Japan

- ▶ Austin, United States
- ▶ Krakow, Poland

Figure 5-1 shows the current geographical distribution.

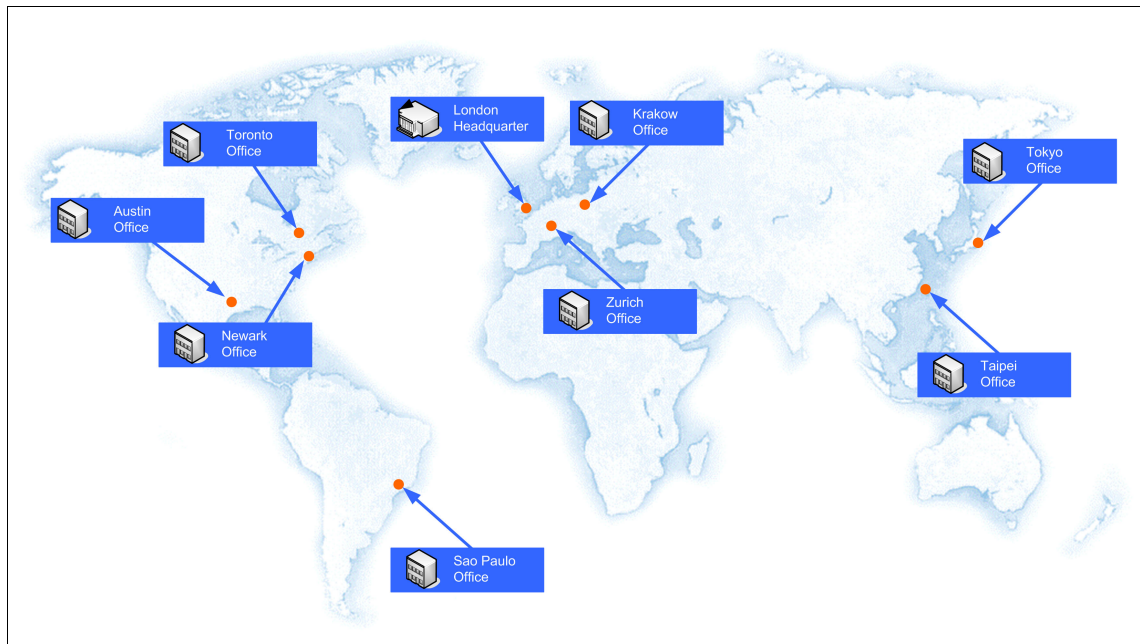


Figure 5-1 Geographical distribution of the financial accounting company

The financial accounting company uses an IT environment with common elements for financial services institutions. The company uses mainframes to perform its most critical core business banking data processing, and a distributed environment to provide IT services to customers, employees, contractors, and business partners.

The distributed environment consists of these components:

- ▶ Workstations and notebooks, which use Microsoft Windows XP and Windows 7, that are deployed in all branches
- ▶ File and print servers that run on Windows Server 2003 and 2008, which are deployed in data centers and satellite offices as departmental servers
- ▶ Web servers and application servers that run on Linux Red Hat Enterprise Server and that are deployed in data centers
- ▶ Anti-virus Relay that runs on Linux Red Hat Enterprise Server and is deployed in each location
- ▶ Database servers that run on IBM AIX and that are deployed in data centers

The financial accounting company estimates that it deployed approximately 120,000 endpoints, including servers, workstations, and notebooks.

Table 5-1 shows how these endpoints are distributed among the locations.

Working remotely: We consider small offices and home offices, which are in the same country, part of the main offices or satellite office locations. Small offices and home offices are called *remote offices*.

Table 5-1 Endpoint distribution

Endpoint location	Platform	Amount
London, UK	Windows XP and 7	25,000
	Windows Server 2003 and 2008	80
	Linux Red Hat Enterprise Server 5	140
	IBM AIX 6.1	14
Krakow, Poland	Windows XP and 7	4,000
	Windows Server 2003	12
	Linux Red Hat Enterprise Server 5	8
Zurich, Switzerland	Windows XP and 7	500
	Windows Server 2003	7
	Linux Red Hat Enterprise Server 5	4
	IBM AIX 6.1	2
Newark, US	Windows XP and 7	18,000
	Windows Server 2003	10
	Linux Red Hat Enterprise Server 5	20
Austin, US	Windows XP and 7	5,000
	Windows Server 2003	20
	Linux Red Hat Enterprise Server 5	10
Toronto, Canada	Windows XP and 7	10,000
	Windows Server 2003	18
	Linux Red Hat Enterprise Server 5	25

Endpoint location	Platform	Amount
Sao Paulo	Windows XP and 7	8,000
	Windows Server 2003	12
	Linux Red Hat Enterprise Server 5	10
Tokyo, Japan	Windows XP and 7	13,000
	Windows Server 2003	60
	Linux Red Hat Enterprise Server 5	50
Taipei	Windows XP and 7	28,000
	Windows Server 2003	100
	Linux Red Hat Enterprise Server 5	120

Each office has its own IT team, which is responsible for locally defining the endpoint policies, configuration, and patch management processes without any centralized coordination and administration. Employees that work in remote offices must manage and maintain their own machines.

A corporate security policy and configuration management process is in place for the servers, but it is implemented and maintained locally by each IT team.

Most business applications are web-based and are accessed by employees and contractors. The most critical web application is called *Quant Unlimited Access Network for Traders* (QUANT). It manages the liquidity, investments, and trading operations of the organization. The supported web browsers are Microsoft Internet Explorer and Mozilla Firefox.

The financial accounting company uses an existing client/server application that is based on Java, called *Network for Traders System* (NTS). The traders and branch office employees use this application to generate new proposals or new customer contracts. NTS generates the documents in PDF format, which are opened by Adobe Acrobat Reader so that the employees can confirm the information and create a printed contract for the customers to sign.

More employees and contractors use virtual private network (VPN) connections to access the network of the organization, because they either work in remote offices or spend time on customer sites in business meetings.

The way that the information assets are used by the organization can be considered critical and highly confidential. These assets largely determine the success of the organization in the competitive financial market. The nature of the

services that the financial accounting company offers requires that the information assets are always available and accessible by all corporate traders. Their locations do not matter. Due to concerns that Internet attacks against financial institutions might be used to destabilize nations, the financial accounting company established its own Security Operations Center in London. The center operates on a 24-hour, seven day schedule.

Some of the IT security components and the network zone layout are shown in Figure 5-2.

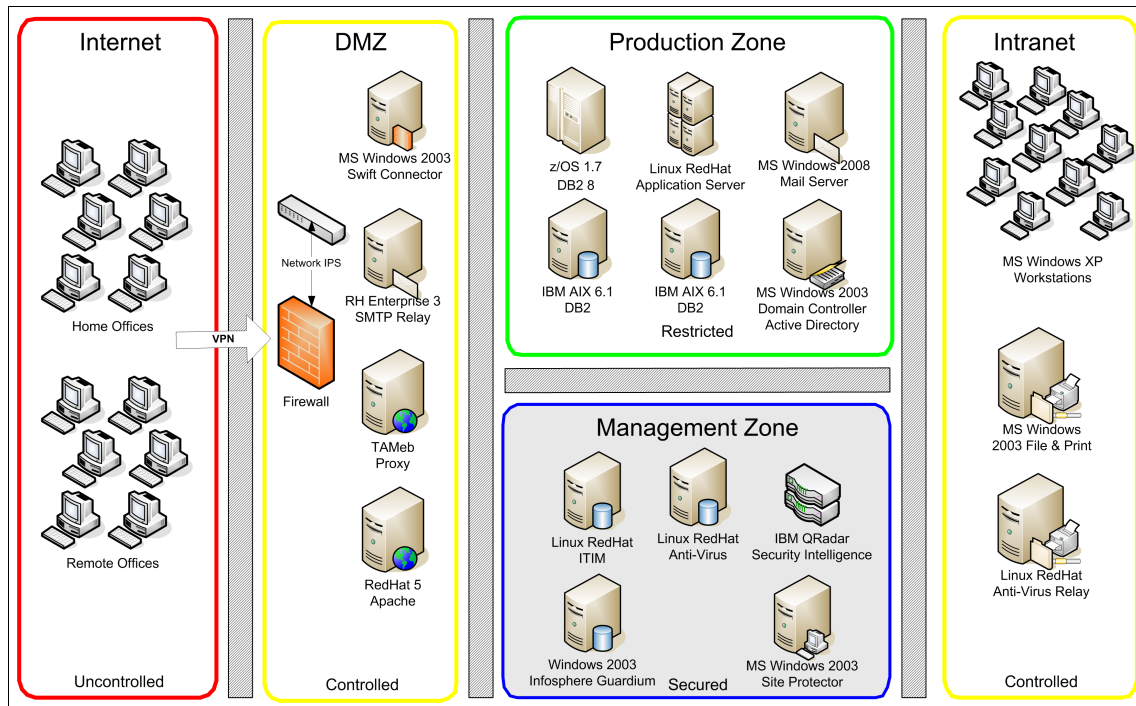


Figure 5-2 Current IT architecture: Network zones

We describe the components:

- ▶ A centralized identity management solution based on IBM Tivoli Identity Manager (ITIM). That system manages the full identity lifecycle for employees, contractors, and business partners. Tivoli Identity Manager workflows are used to implement business and user provisioning automation processes. These workflows mitigate security risks of excessive access rights on IT systems, segregation of duties, and other security controls.
- ▶ Centralized web access control management is implemented based on the IBM Tivoli Access Manager for e-business solution (TAMEb). Because most of

the financial accounting company business applications are web-based, this solution implements an extra application security layer. This layer provides centralized authentication, authorization, and single sign-on (SSO) for those applications. The financial accounting company also uses IBM Rational® AppScan® for web application vulnerabilities testing.

- ▶ The IBM InfoSphere® Guardium® distributed real-time database monitoring system is used to monitor all database activities in real time, including privileged user access. This solution reports *separation of duty* issues of the native database logging mechanism. Guardium also provides capabilities, such as blocking, workflow management, and vulnerability assessments for the databases.
- ▶ For network threats, the financial accounting company uses the IBM Security Network intrusion prevention system (IPS) devices to protect the network perimeter and all systems behind the Internet DMZ firewall. This solution uses a centralized console to correlate events, configure and apply new security rules, and prevent threats and hackers from gaining access to sensitive and confidential data.
- ▶ Finally, a centralized compliance analytics and log management solution is implemented based on IBM Security QRadar components. This solution collects logs from the most critical servers, network, security, and application infrastructure. QRadar analyzes millions of security-related events and uses advanced analytics to pinpoint about 20 daily incidents. QRadar also provides regulatory standards-based compliance reports that are based on the corporate security policies.

5.1.2 Security issues within the current infrastructure

We can identify several security issues in the current IT environment.

The financial accounting company invested in IT security solutions for years. Looking at a holistic IT security approach that follows the IBM Security Framework, those solutions help the company mitigate many security risks and threats in several of the security domains. However, there are areas that are inadequately or not covered by a security solution.

Staying focused: We show relevant issues related to an endpoint security and compliance management solution. There are other security issues, but we do not address them in this book.

We identified the following security issues in this company:

- ▶ The lack of corporate security policies, standards, and tools to support *endpoint security management* to gain control of and visibility into the current IT environment.
- ▶ Due to current business process rules, the users must have administrative rights on their own machines. These rights allow them to execute and install any software, which can introduce increased risks and exposures, such as malware, trojans, and similar threats.
- ▶ Increased risks and exposures related to patch levels for operating system and application software, including security-related solutions, such as anti-virus software and signature patterns. This shortcoming surfaced because the financial accounting company does not have control and centralized visibility of its IT endpoint environment.
- ▶ At least two critical risk factors because each branch or satellite office uses a unique IT team, which is responsible for defining and enforcing local security policies:
 - No central corporate security configuration and compliance management solution. A corporate solution is a critical requirement to control security, stay in control of security, and adhere to corporate and regulatory compliance guidelines.
 - No IT environment control, because the financial accounting company does not have a centralized visibility and enforcement solution for securing the endpoint environment. This lack of control exponentially increases the risks with each acquired system.

5.2 Business vision

In this section, we examine the growth plan that the financial accounting company created for the next three to five years. The company wants to engage in the following areas:

- ▶ Continue the geographical expansion strategy to other countries in the European Union and Asia. The financial accounting company plans to open strategic satellite offices worldwide and hire approximately 15,000 employees and contractors.
- ▶ Open a subsidiary to process credit cards worldwide in the next two years. This subsidiary starts in the UK. The company plans to hire 1,000 new employees.
- ▶ Buy or merge with other financial institutions to improve gross profit and decrease operational costs.

- ▶ Implement an effective governance, risk, and compliance strategy to improve the stock value of the company.
- ▶ Increase gross profit by reducing costs in key operational areas, such as IT and call centers. The financial accounting company plans to increase gross profit without decreasing the quality of customer service and increasing operational and security risks.

5.3 Business requirements

Based on this business strategy vision, the financial accounting company wants to achieve the following short-term business goals, related to the IT environment:

- ▶ Improve worldwide IT services quality and availability. Implement a new process and tools to improve the automation procedures and the visibility of the IT environment.
- ▶ Mitigate the exposure of customer-sensitive and confidential information by increasing the security controls to protect the reputation of the financial accounting company.
- ▶ Implement auditing processes to manage internal and industry regulation controls, such as Basel, Sarbanes-Oxley (SOX), and Payment Card Industry - Data Security Standard (PCI-DSS). The first project phase must address the internal controls and auditing requirements, to improve the corporate visibility of the asset and IT environment. In the second phase, the company implements the industry regulations based on priorities and needs.
- ▶ Implement processes and IT tools for better resilience to support the geographical expansion strategy. The financial accounting company must work on the expansion in parallel with the day-to-day operations. The company must have the processes and tools to support this business requirement.

5.3.1 IBM Security Framework mapping to business requirements

Using the IBM Security Framework definitions for business-driven security, the business requirements discussed in 5.3, “Business requirements” on page 195, and the current organizational infrastructure discussed in 5.1.1, “Current IT infrastructure” on page 188, we engage in a discussion with the financial accounting company to better address the needs and requirements. This discussion helps us deliver more value when evaluating the functional requirements, by using the underlying IBM Security Blueprint.

We look at each of the IBM Security Framework security domains:

▶ People and Identity

The financial accounting company uses a mature identity and access management process and tools that help maintain low costs and mitigate risks related to this domain. The implementation uses IBM Tivoli Identity Manager and IBM Tivoli Access Manager software to manage the employee and contractor identity and access lifecycle and enforce access to the business applications.

▶ Data and Information

The financial accounting company uses a granular information asset classification scheme paired with a least privilege principle. Access to the database servers is monitored in real time consistently. The access is enforced, including privileged users, without causing any of the performance impact and separation of duties issues of native database logging. The access is enforced by using IBM InfoSphere Guardium Database Monitoring and Protection. The solution is integrated with the IBM Security QRadar security analytics solution in the Security Operation Center.

▶ Application and Processes

The financial accounting company follows a rigorous release management process with a granular promotion-to-production path that specifies security testing criteria. The company uses IBM Rational AppScan software for testing during the early development stages through to applications that run in the production environment. This approach helps with practicing security during the application development phase, and also helps discover any application vulnerabilities. The processes of the financial accounting company achieve a high level of automation and embrace security controls, such as the separation of duties and creation of auditable records.

▶ Network, Server, and Endpoint

The financial accounting company implemented a threat management system worldwide, based on IBM Security SiteProtector™ and IBM Security Network IPS. This solution implements an extra security layer for the financial accounting company network architecture and supports the business requirement approach of the company.

For network and server management and vulnerability scanning of the entire IT infrastructure, the company uses an IBM Managed Services solution. Today, there is no solution to address the endpoint management.

We identified a critical gap related to endpoint security and compliance management for this environment. The financial accounting company has a clear strategy to geographic expansion. Currently, the company uses approximately 120,000 endpoints. There are more than 100,000 potential

entry points for fraud and threats, because of the lack of visibility, control, and automation for these infrastructure components.

We discovered another important gap that is related to endpoint protection, to mitigate other security risks and threats on endpoints. The financial accounting company has no confidence in the current anti-virus solution or the workstation protection, such as personal firewall and host IPS.

The company decided first to focus on the security configuration and compliance management gap and to postpone the endpoint protection solution for a second project phase.

- ▶ Physical Security

Physical security controls are also part of the financial accounting company security program. Physical access controls to facilities and systems are in all locations.

- ▶ Governance, Risk, and Compliance

The financial accounting company enforces compliance by managing a security controls framework and strict audit and security awareness program. From a security monitoring view, the company runs a Security Information and Event Management solution with compliance reporting modules for SOX, PCI-DSS, and other specially developed custom reports. This solution helps address important regulations for the financial accounting company operations. The financial accounting company designed and implemented a security policy framework and supporting processes for security governance. The company proactively identifies and eliminates security threats that enable attacks against systems, applications, and devices by using IBM Rational AppScan and IBM InfoSphere Guardium Database Monitoring and Protection. The company integrates these products with the IBM Security QRadar technology for compliance reporting.

It is a priority for the financial accounting company to implement an enterprise-wide Governance, Risk, and Compliance solution (eGRC). From the information security view, the Security Governance, Risk, and Compliance strategy must be aligned with the organizational eGRC. The financial accounting company already deployed several security tools to cover almost all security domains to provide input to the GRC solution. With an endpoint security and compliance management solution acquisition, the company can close a critical gap and map another important IT security solution to support the GRC strategy.

We based our conclusions on the engagement session discussed in 5.3.1, “IBM Security Framework mapping to business requirements” on page 195, the gaps identified in this session, and the strategy and prioritization defined by the financial accounting company. We conclude that the required approach is based on an endpoint security and compliance solution. This solution is covered by both

the Network, Server, and Endpoint, and the Governance, Risk, and Compliance domains of the IBM Security Framework.

We describe the next steps to better understand the functional requirements and map them to the IBM Security Blueprint. Then, we describe the implementation approach at a high level.

5.4 Functional requirements

The financial accounting company needs improvements in the endpoint security, governance, risk, and compliance areas. To address these areas, IBM and the financial accounting company developed a set of high-level functional requirements. The following requirements are needed for the successful implementation of an endpoint security and compliance solution:

- ▶ Review the existing set of information security policies, update them with the new requirements, and transform them into a corporate security organizational policy.
- ▶ Implement corporate endpoint security controls based on federal standards best practices and internal security policies.
- ▶ Implement effective patch management mechanisms to address the security and corrective software requirements for current operating system environments (for workstations, notebooks, and servers) and client-side applications.
- ▶ Create centralized real-time and historical compliance reports for auditing information and endpoint security governance.
- ▶ Establish consolidated policy enforcement for security configuration throughout the enterprise to ensure that all systems use the appropriate policies. To better manage compliance in accordance to data privacy laws and industry regulations, these policies must be rapidly deployed. They need to provide a view of the overall endpoint coverage throughout the environment.
- ▶ Implement and manage all these requirements by using a centralized architecture.

5.4.1 IBM Security Blueprint mapping to functional requirements

Although we now understand the functional requirements for the additional security measures that the financial accounting company needs to implement, we still must determine the specific solutions that can potentially fulfill the functional requirements. By using the IBM Security Blueprint, we can map the

functional requirements into specific architectural artifacts by using the IBM Security Blueprint, identifying the appropriate solutions to implement. Figure 5-3 shows the mapping of the functional requirements to the IBM Security Blueprint.

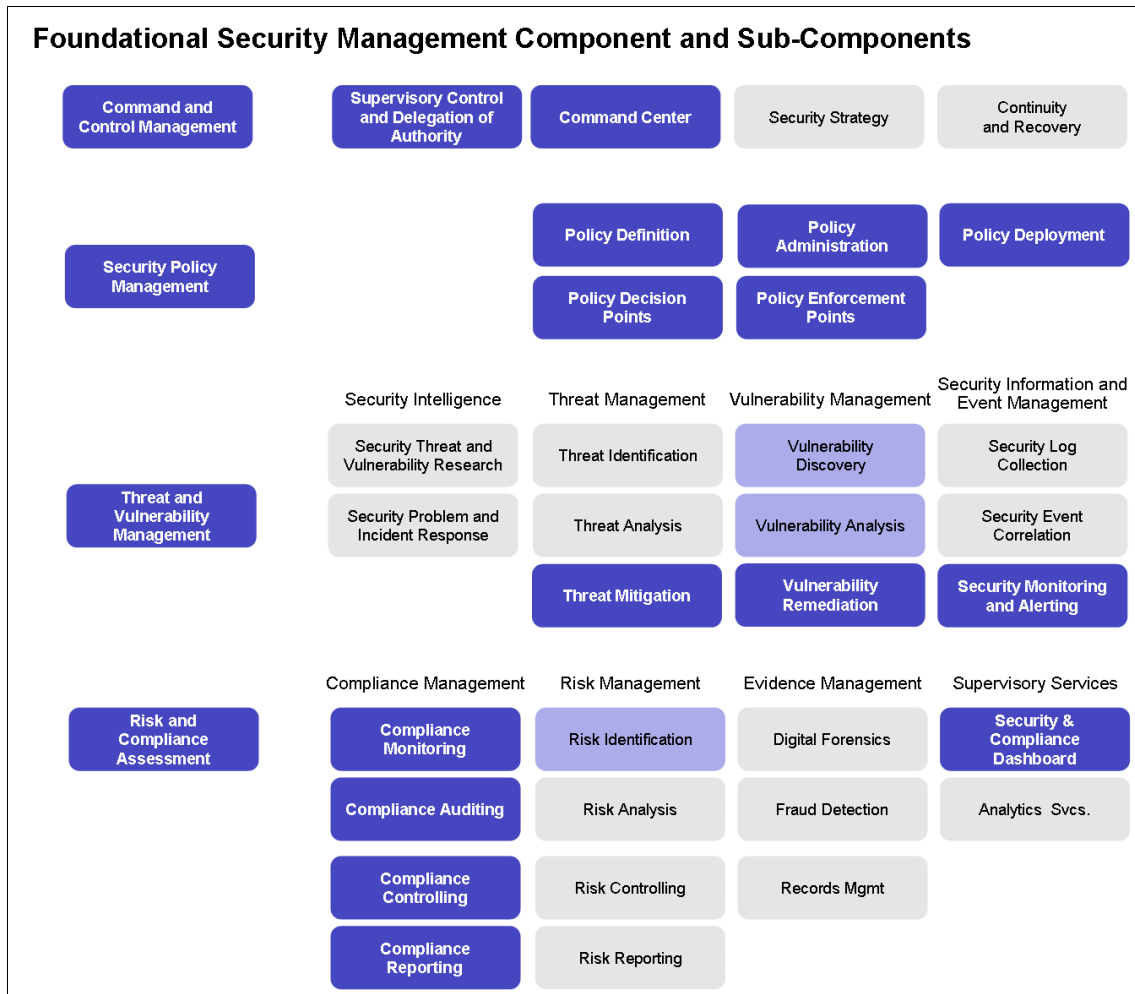


Figure 5-3 IBM Security Blueprint: Foundational components for functional requirements

We look at each of the required IBM Security Blueprint subcomponents:

- ▶ Command and Control Management:
 - Security Control and Delegation of Authority

Based on roles definition and governance policies, this subcomponent provides a segregated and access-controlled solution to manage the overall IT environment, security policies, and report views.

- IBM Command Center®

The Command Center provides a robust platform to support a centralized endpoint management solution for the entire organization.
- ▶ Security Policy Management:
 - Policy Definition

Based on the current business requirements, the financial accounting company must implement several security controls as soon as possible. The company can use a mature and market-independent checklist as a guideline.
 - Policy Administration

The Policy Administration subcomponent addresses the centralized endpoint lifecycle management of security administration policies. Management includes creating, modifying, deleting, and other maintenance tasks for policies.

Implement financial accounting company-specific security controls, based on internal policies, to complete the security checklist requirements.
 - Policy Deployment

This subcomponent provides a centralized architecture and a secure channel for security policy distribution and deployment to the endpoints. The Policy Deployment is responsible for validating the deployment status, and if the deployment fails, the Policy Deployment must redeploy the policies.
 - Policy Decision Points

A Policy Decision Point analyzes the endpoint security posture and generates alerts about noncompliant configurations, based on the security policies already defined and applied to the IT environment.
 - Policy Enforcement Points

Policy Enforcement Points enforce internal and external IT endpoint security controls.

The standard security configuration, which reflects the organizational security policy, must be enforced to all endpoints at all times. If changes occur on endpoints, either accidentally or maliciously, the changes must be identified. The standard security configuration must be reapplied.
- ▶ Threat and Vulnerability Management:
 - Threat Mitigation

Threat Mitigation represents the implementation of real-time security reports for alerting about security risks based on a noncompliant security

configuration, either with internal security policies or external security regulations.

Based on the integration with security alert bulletins sent by operating system and third-party application vendors, an automated patch installation process is triggered to correct the issues that caused the threat to the IT environment.

- Vulnerability Discovery

Vulnerability Discovery helps to identify the endpoints with out-of-date patches, based on the published security bulletin information from operating system and third-party application vendors.

Based on internal security policies or federal security configuration best practices, such as the FDCC or DISA STIGs, the endpoints are assessed for noncompliant security configuration.

Vulnerability Discovery is shaded in light blue (Figure 5-3 on page 199), because the financial accounting company uses an IBM Managed Service offering for its network-based vulnerability scanning. The Vulnerability Discovery for its endpoint management acts in a restricted way, which is also true for Vulnerability Analysis.

- Vulnerability Analysis

The Vulnerability Analysis, in this case, is only responsible to report the vulnerability ratings, based on the federal agencies best practices analysis, such as FDCC and DISA STIGs.

- Vulnerability Remediation

Vulnerability Remediation mitigates security risk and threat exposures and improves IT services quality by implementing a patch management solution.

- Security Monitoring and Alerting

Security Monitoring and Alerting implements historical and real-time reports to consistently monitor the endpoint security behavior to validate deviations of standard operations in a specific period.

- ▶ Risk and Compliance Assessment:

- Compliance Monitoring

Compliance Monitoring constantly checks security configuration reports to analyze and identify the actual security threats and configuration management issues and deviations. It can provide real-time compliance reports, based on security and compliance endpoint policy.

- Compliance Auditing

Compliance Auditing provides a historical database to retain the audit data about the endpoint-specific compliance posture and overall organization endpoint compliance statistics.

- Compliance Controlling

Compliance Controlling provides different compliance views and compliance policy rules that are based on business requirements, such as country, type of system, and organization department.

- Compliance Reporting

Compliance Reporting provides compliance reports and audit information, based on organizational security controls and federal standards best practices, including real-time and historical reports.

- Risk Identification

Risk Identification refers to the ability to discover, recognize, and verify the existence of specific risks. It also encompasses the structuring of risk by mapping it into clearly defined classification schemes that can be specific to the industry or even to the risk taxonomy of an individual organization.

The Risk Identification subcomponent for the endpoint management solution provides additional information for the holistic Risk Identification approach, which is why it is also colored in light blue (Figure 5-3 on page 199).

- Security and Compliance Dashboard

A Security and Compliance Dashboard helps identify and report in real time the threats, compliance posture, and vulnerabilities discovered on the managed endpoints, based on patch information, security configuration, and policies.

Further reading: If you want to learn more about general functionality of the IBM Security Blueprint, see the IBM Redpaper *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528.

Although business and functional requirements are the main parts of the security design solution, we also must consider the non-functional requirements and constraints. These non-functional requirements and constraints might include objectives that are necessary to meet general business requirements or practical constraints about constructing security architectures. The architectural team performed an analysis of non-functional requirements, and identified the following key non-functional requirements and constraints:

- ▶ Provide a high-availability solution that can be replicated to the backup data center in London.

- ▶ Implement endpoint self-monitoring and self-enforcement to guarantee the solution is installed in as many endpoints as possible.
- ▶ Ensure that the management platform supports heterogeneous environments. The solution must support at least Microsoft Windows 2003 and 2008 Server, Windows XP, Windows 7, Linux Red Hat Enterprise Server ES 5, and IBM AIX 6.1.

The following sections show how to further use the IBM Security Framework and IBM Security Blueprint in both the design and implementation of new security solutions.

5.5 Design approach

Now that we determined the Foundational Security Management components and subcomponents of the IBM Security Blueprint for our solution, we can map the technical requirements into the Security Services and Infrastructure components of the IBM Security Blueprint. The exercise helps us determine which security solutions best satisfy all our requirements: business, functional, non-functional, or technical.

Figure 5-4 shows how the mapping was done for the financial accounting company by using the functional requirements and existing architecture.

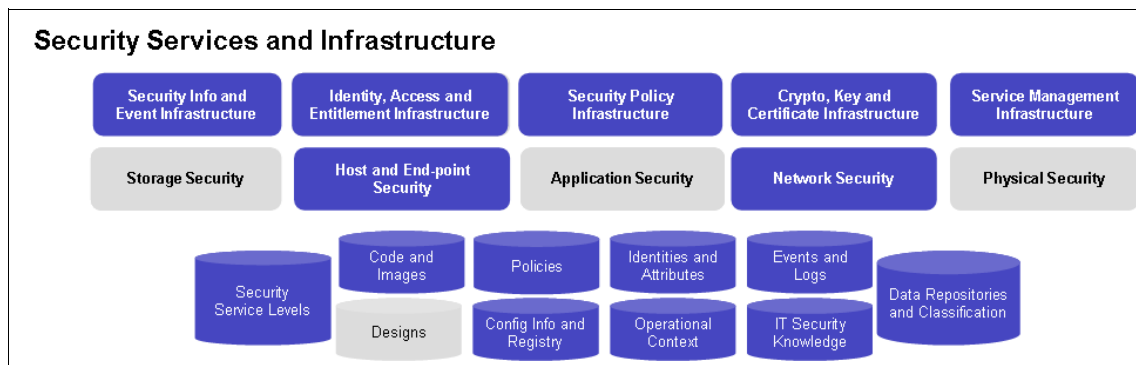


Figure 5-4 IBM Security Blueprint: Security Services and Infrastructure

As part of the design, we can produce a detailed implementation plan for our deployment that involves the following steps:

- ▶ Prioritize the requirements and define the *quick wins*.
- ▶ Map the requirements to IBM product features to define a better project plan and project expectations.

- ▶ Define the tasks involved in using those features to satisfy the requirements and estimate the effort required for each task.
- ▶ Define the project phases.

Therefore, we now focus on the technical components of the IBM Security Blueprint and how they can be mapped into technical and operational requirements.

Based on the mapping, we know that the solution must provide the following Security Services and Infrastructure components:

- ▶ The solution must provide security communications among all components. The messages, or information, exchanged among components must be encrypted.
- ▶ The solution must provide a single agent for security patching, policy enforcement, compliance reporting, and vulnerability remediation.
- ▶ The solution must contact the operating system and third-party application vendors, and, based on the IT environment analysis, must download the required patches from the vendor site.
- ▶ The solution needs to provide one or more mechanisms to automatically install an agent based on a predefined endpoint list.
- ▶ The solution must provide a console view to check and validate which machines are not being managed.
- ▶ The solution must provide centralized administration in London. All administrative action, security policy customization, patch management, and reports are defined and managed in London.
- ▶ The solution needs to provide access to report information that can be classified and segregated. For example, a Windows administrator can see only security information about Windows machines, or a manager for Poland can access Polish server information only.
- ▶ The solution needs to implement a patch management solution that supports the business requirements and security policies:
 - Multiple language support
 - Patches for Windows
 - Patches for Linux Red Hat Enterprise Server
 - Patches for AIX
 - Custom application patches developed by the organization
 - Client-side application patches, including but not limited to Mozilla Firefox, Adobe Reader, and Sun Java Runtime Environment (Sun JRE)

- Disconnected patching, because the financial accounting company sales force and some remote offices work disconnected from network part of the time
- Patch management approval process
- ▶ The solution must provide auditing and compliance reports that can be accessed by using an Internet browser, without needing to install any additional components on the manager, auditor, or security officer systems. The reports must be accessible in any part of the network.
- ▶ The solution must provide historical and real-time reports about security configuration, patch management status, and endpoint compliance posture.
- ▶ The solution must integrate with external security checklists, such as FDCC and DISA STIGs, to manage and control endpoint security configuration.

At this time, the financial accounting company decided to postpone the following requirements to the second project phase:

- ▶ Security configuration and patch management for the AIX platform.
- ▶ Patch management for the Linux platform.
- ▶ Patch management for the Sun JRE application.

5.6 Implementation approach

We now have a clear understanding about all the company requirements and how they map to the IBM Security Framework and IBM Security Blueprint. We can apply the knowledge to select the appropriate solutions to satisfy all those requirements.

Based on the design approach discussed in 5.5, “Design approach” on page 203 and by scrutinizing the previous chapters in this book, which discussed endpoint security and compliance management solutions, we can create the solution that best satisfies our requirements:

- ▶ IBM Tivoli Endpoint Manager for Security and Compliance is designed to solve the increasingly complex problem of keeping critical endpoint systems updated, compliant, and free of security vulnerabilities. To address the financial accounting company requirements, we defined the following approach:
 - Platform implementation

In Chapter 6, “Phase I: Tivoli Endpoint Manager platform design and implementation” on page 211, we cover the basic platform implementation approach and design, as shown in Figure 5-5 on page 207.

- Patch management

In Chapter 7, “Phase II: Patch Management design and implementation” on page 239, we describe how patch management can address the vulnerability and security risk requirements.

- Security configuration

In Chapter 8, “Phase III: Security policy configuration design and implementation” on page 293, we write about how security configuration can address the requirements related to security policies.

- Compliance reports

In Chapter 9, “Phase IV: Security Compliance Analytics reporting” on page 357, we explore how Tivoli Endpoint Manager Analytics can help address the compliance report requirements.

By using the IBM Tivoli Endpoint Manager for Security and Compliance as our solution for the financial accounting company business requirements, we designed a new logical diagram for this solution. The diagram is depicted in Figure 5-5 on page 207.

Endpoint view: To keep the figure clear, we defined a logical diagram to show the architectural solution only from an endpoint security perspective. We hid the other security solutions from this view.

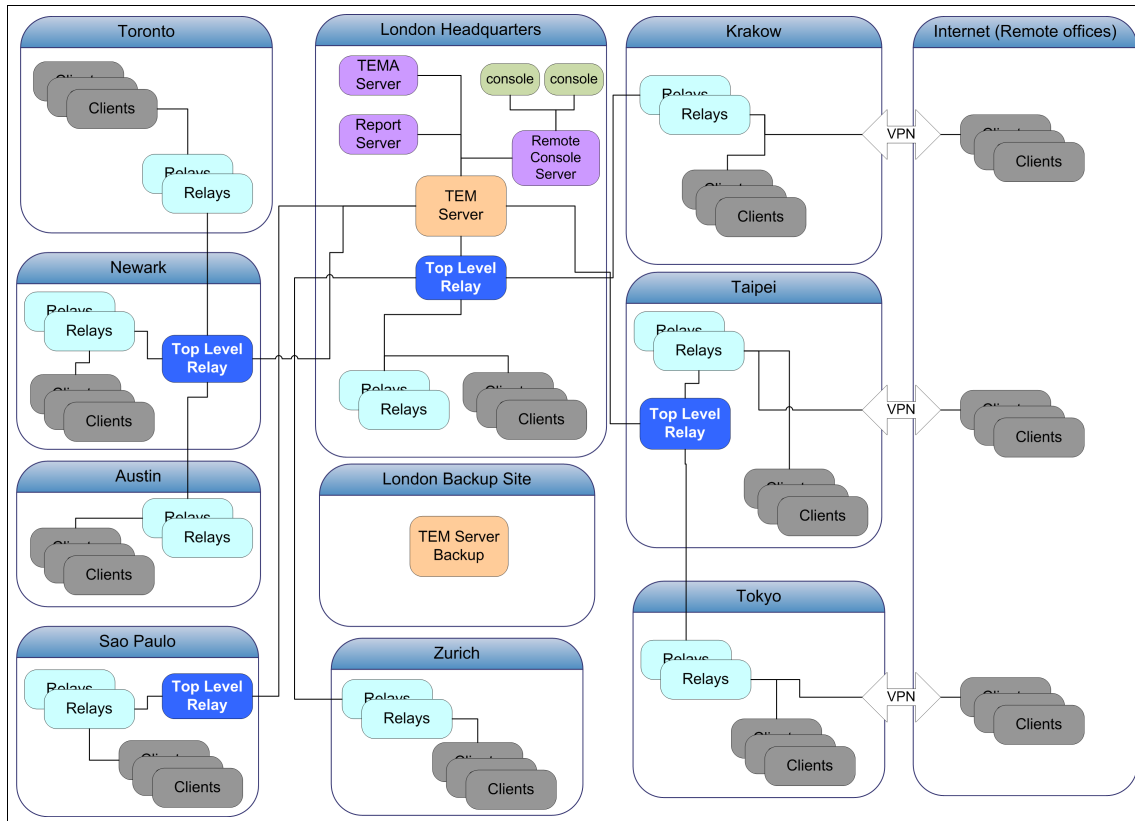


Figure 5-5 Logical architecture overview for the new proposed endpoint security management solution

By using the logical components, we can now create an updated solution that uses a network zones diagram, shown in Figure 5-6 on page 208.

Location: We present the London Headquarter physical diagram only, which is where the main components for the Tivoli Endpoint Manager platform are located. However, to manage the financial accounting company worldwide endpoint solution, it is necessary to define a physical diagram in each location. The physical diagram in each location must follow the same architecture pattern in a simpler design with local network architecture restrictions and fewer components.

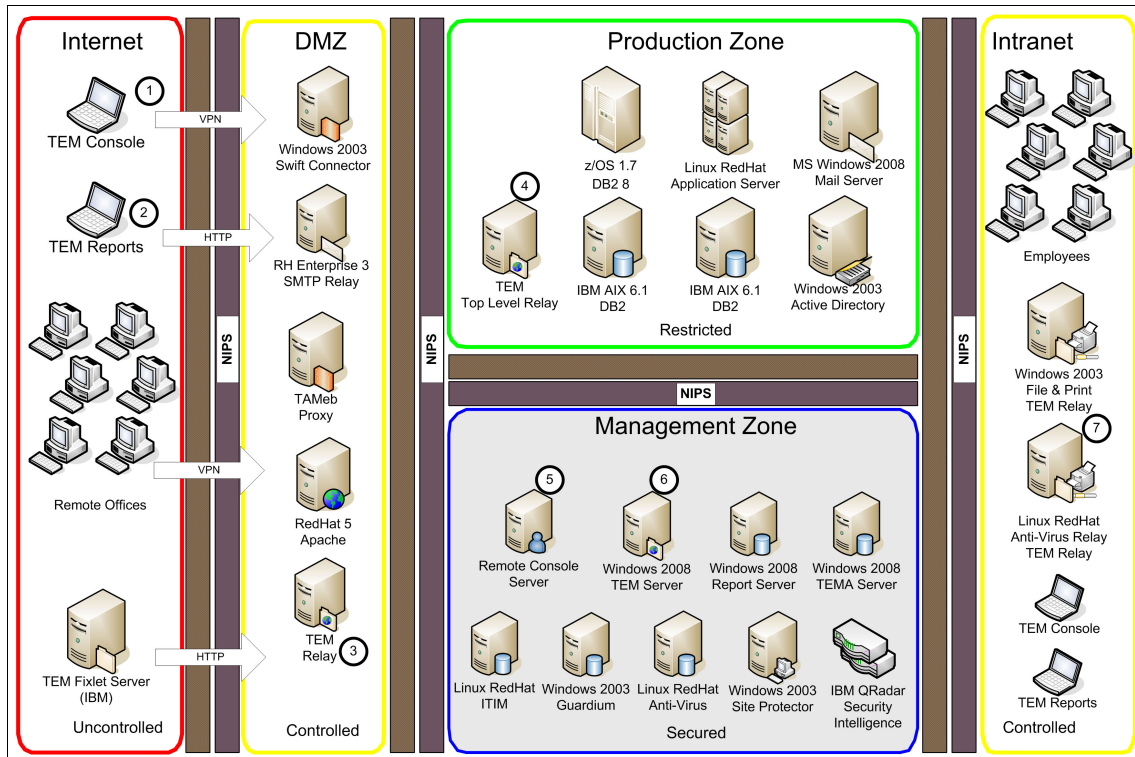


Figure 5-6 Physical architecture for the new proposed solution that uses a network zone

Consider the following points about the new proposed endpoint management architecture:

- ▶ The Tivoli Endpoint Manager Console (TEM Console) must use a VPN connection when accessed by an operator in a remote location.
- ▶ Tivoli Endpoint Manager Reports (TEM Reports) can be accessed through an Internet browser that uses an Internet (HTTP) connection.
- ▶ The endpoints used by employees or contractors in remote offices connect to the endpoint management solution with a Tivoli Endpoint Manager Relay (TEM Relay) in a DMZ network zone.
- ▶ The Tivoli Endpoint Manager Top Level Relay (TEM Top Level Relay) is placed in the Production Zone.
- ▶ The internal or external operators, auditors, or managers, who need access to the Tivoli Endpoint Manager Console, must use a connection that uses a remote console server solution.

- ▶ All other Tivoli Endpoint Manager Server components are in the Management Zone.
- ▶ Based on the current server loads, several of the loads are used as Tivoli Endpoint Manager Relays (TEM Relays) for the Agents installed on internal workstations, notebooks, and servers.

For further details about the architectural decisions and other preferred practices for designing a Tivoli Endpoint Manager architecture, look at Chapter 4, “IT endpoint security and compliance solution design” on page 125.

5.7 Conclusion

In this chapter, we described the design approach that the financial accounting company used to design the security configuration and compliance management solution that uses IBM Tivoli Endpoint Manager for Security and Compliance. We also showed how the IBM Security Framework and the IBM Security Blueprint can provide a structure to derive the IT functional and technical requirements from the business vision, goals, and requirements.

First, we introduced the financial accounting company. We discussed the organization profile, current IT infrastructure, and issues with endpoint security.

Next, we described the business vision, business requirements, and associated functional requirements. After refining these requirements to a more detailed technical level, we described the design approach that the financial accounting company used for the solution. We followed the IBM Security Framework and the Endpoint Security Management Solution Pattern of the IBM Security Blueprint. When applied to the unique IT environment of the financial accounting company, this process of analysis and design helped the financial accounting company define an implementation plan.



Phase I: Tivoli Endpoint Manager platform design and implementation

In this chapter, we describe the financial accounting company design and implementation for Tivoli Endpoint Manager to provide an endpoint security and compliance management solution to meet all of the regulatory requirements. We divide the task into the following sections:

- ▶ “Design” on page 213
- ▶ “Implementation” on page 216
- ▶ “Maintenance” on page 230

The financial accounting company plans to list with the US Stock Exchange six months from today. The company wants to meet the auditing and reporting compliance needs within that period. The company plans to use IBM Tivoli Endpoint Manager for Security and Compliance and IBM Tivoli Endpoint Manager for Patch Management as the basis for the endpoint security and compliance management solution.

Tivoli Endpoint Manager provides centralized control for a collection of heterogeneous endpoints that can operate on a global scale. It provides visibility in real time for every endpoint in the organization, and it is scalable and adaptable for future expansions of the IT systems.

To properly use IBM Tivoli Endpoint Manager for Security and Compliance and IBM Tivoli Endpoint Manager for Patch Management to address regulatory requirements, all components of the solution must be implemented correctly. In the next six months, the financial accounting company IT team is focused to design and implement Tivoli Endpoint Manager.

6.1 Design

In this section, we describe how the financial accounting company plans to implement Tivoli Endpoint Manager according to its business requirements. We explain how the company creates a deployment method that is best suited for the business and technical structure of its organization.

6.1.1 Business requirements

The financial accounting company needs to fulfill the business requirements outlined in 5.3, “Business requirements” on page 195. The company is looking for a solution to centrally manage the distributed endpoints and provide patching and security policy compliance management capability. The company needs to cover the entire set of IT systems, including all platforms, such as Microsoft Windows, Linux, AIX, and Solaris UNIX. After all endpoint systems are registered, the company can begin to implement the patching and security compliance solutions.

Because the financial accounting company is a global organization, the system-wide solution must support global operations while providing central control. The system must comply with both corporate policy and local IT policy. It also must support the future expansion of existing sites and the addition of sites. All the business needs must be met with the current or fewer IT personnel resources.

6.1.2 Functional requirements

The financial accounting company first needs to deploy the Tivoli Endpoint Manager platform components to the IT organization to meet the software requirement. The company plans to use Tivoli Endpoint Manager for Patch Management for its patching solution and Tivoli Endpoint Manager for Security and Compliance for its security configuration compliance solution.

The IT team listed the following functional requirements for this implementation:

- ▶ Capability to support and manage 120,000 endpoints
- ▶ No disruption of business operation
- ▶ No significant degradation of endpoint performance
- ▶ No requirements for disabling any security devices, such as firewalls
- ▶ Capability to support patch management solution
- ▶ Capability to support endpoint security compliance solution
- ▶ Scalable for future expansion
- ▶ No requirements for additional IT resources, preferably less

- ▶ Capability to provide high endpoint visibility
- ▶ Capability to provide complete endpoint control
- ▶ Capability to implement project within six months
- ▶ Capability for an operational disaster recovery time of less than 36 hours

A single Tivoli Endpoint Manager infrastructure can scale up to 250,000 - 300,000 endpoints, which can cover the needs of the financial accounting company. The installation of an Agent usually takes a few minutes without requiring a reboot. After careful evaluation, the IT team is sure that Tivoli Endpoint Manager can meet all the requirements listed.

Teams and responsibilities

In the current situation, each local site uses its own IT team and follows local security policies. The financial accounting company uses the following IT teams:

- ▶ Newark, managing North America sites
- ▶ Sao Paulo, managing Latin America
- ▶ Taipei, managing Asia Pacific
- ▶ London, managing Europe, the Middle East, and Africa (EMEA) region

The Tivoli Endpoint Manager Console provides the capability for the teams to separate the management responsibilities for endpoints either by geographical area or function. For the current implementation task, each regional IT team is responsible for its Relay and Agent deployment. Only a few senior members in the London-based IT team are granted *master operator* access. The master operators are responsible for managing all other operators, creating custom endpoint groups, and performing global configuration and deployment tasks. Regional endpoints are grouped automatically according to their IP addresses, as shown in Figure 6-1 on page 215.

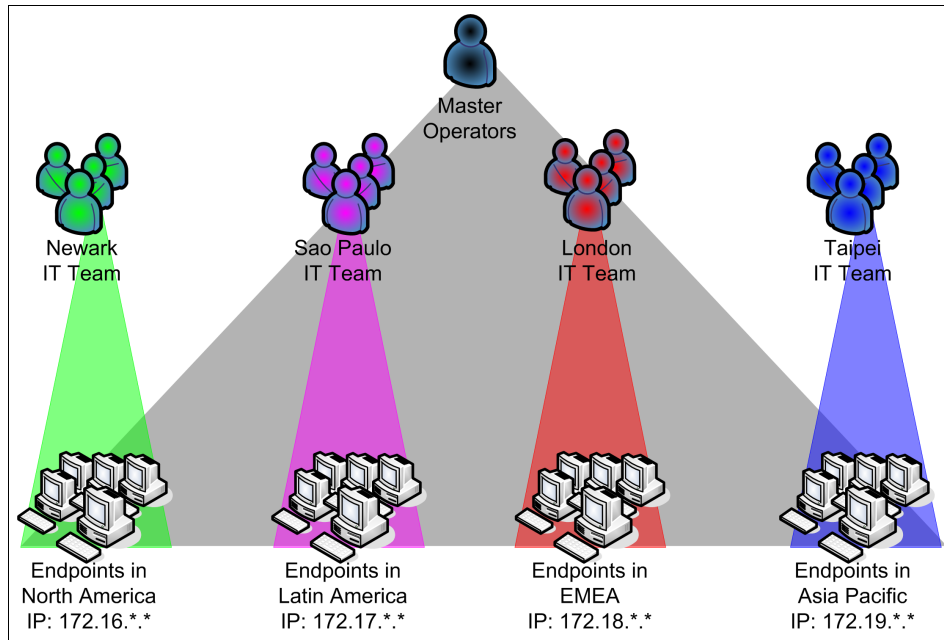


Figure 6-1 Operator responsibility regional breakdown

The use of IT teams with identical functions in different locations can create management overhead. It can also complicate the entire security policy structure and make it more difficult to manage as the financial accounting company grows. The financial accounting company wants to investigate its current IT team structures and tasks to reduce costs and increase efficiency. The financial accounting company intends to unify the regional IT teams into a global IT team with duties separated in functions. A group of members in the team is specialized in patch management and another group of members is focused on security compliance. To achieve this goal, the financial accounting company relies on the flexibility in user role management that Tivoli Endpoint Manager provides.

Tivoli Endpoint Manager Console can support this goal by modifying operator privileges based on groups and Fixlet messages. The master operator can manage those privileges for other operators, limiting their access to only content and tasks that are required to handle the endpoints for which those operators are responsible. For more information about operations and responsibilities, see 3.1.10, “Users” on page 86.

User role: Tivoli Endpoint Manager supports *operator roles*. Users can define roles, such as patch operator or configuration operator. User management by function is much easier.

6.2 Implementation

The IT team deploys Tivoli Endpoint Manager into the organizational IT infrastructure, beginning with the Tivoli Endpoint Manager Server, and continuing with the Relays. After the Server and Relay structure is deployed and operational, the team deploys the Agents to the endpoints.

6.2.1 Network considerations

The IT deployment team identified, documented, and informed the IT networking department about the necessary network traffic for the Tivoli Endpoint Manager components. A summary of all traffic is in 3.3.1, “Intercomponent traffic” on page 108.

The IT networking team configures the firewall ports that are needed by Tivoli Endpoint Manager to allow traffic between the components. In this setup, it is not necessary to shut down or disable any of the security and network devices.

6.2.2 Tivoli Endpoint Manager Server

Tivoli Endpoint Manager Server is the core of the entire Tivoli Endpoint Management system. It is suggested to deploy the Tivoli Endpoint Manager Server on a network infrastructure with large bandwidth and low delay. The London main site is the immediate candidate, because the site network is recently upgraded. It has the best wide area network (WAN) connection quality compared to any of the other sites. Also, London has the largest IT team. This team worked in the financial accounting company the longest and has extensive knowledge about the organizational IT infrastructure. This knowledge helps during the Tivoli Endpoint Management implementation project.

The site also needs to have enough available IT resources that can potentially be used as Relays to accommodate network traffic from remote sites worldwide. The London site deployed over 200 servers that can be configured as Relays. This number can support all the endpoints, combining the potential roaming mobile devices in the EMEA region.

The Tivoli Endpoint Manager Console operates best if it is connected to the Server locally. It also is reasonable to deploy the Tivoli Endpoint Manager Server at the site with most of the operators. For these reasons, the IT management chooses London as the site to host Tivoli Endpoint Manager Server.

Network zones

Because Tivoli Endpoint Manager Server almost fully controls all the managed endpoints connected to it, it is imperative to place it in the network zone with the highest security measures. Figure 6-2 depicts how to implement Tivoli Endpoint Manager Server in terms of the network security zone.

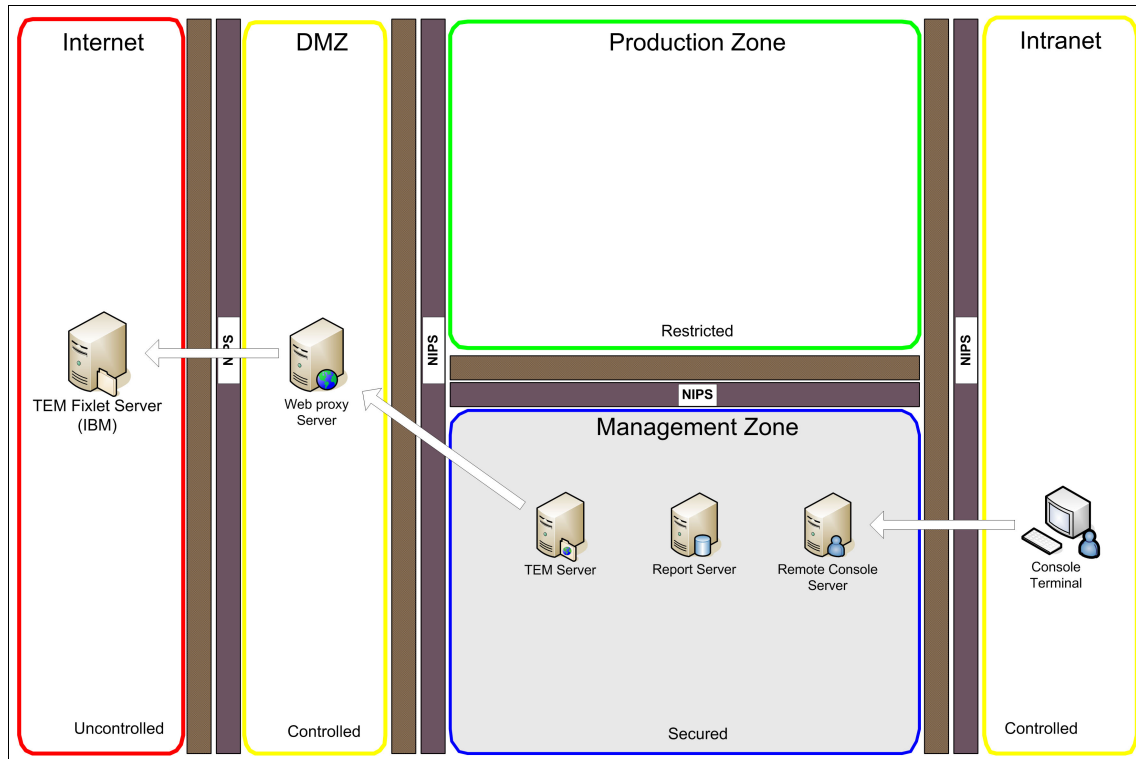


Figure 6-2 Security zone illustration

Operation-critical servers, including Tivoli Endpoint Manager Server, Tivoli Endpoint Manager Report Server, and Tivoli Endpoint Manager Remote Console Server, are placed in the Management Zone. The Management Zone has maximum security that is provided by the firewall and network intrusion prevention systems (IPS). To receive updated content from an IBM Fixlet Server, the Tivoli Endpoint Manager Server connects to the Internet by using a web proxy server. By doing so, it is isolated and shielded from any direct connection to the Internet.

Disaster recovery

The IT team plans to implement two backup deployments, backing up OS, application, and database information to facilitate a fast recovery in a disaster.

The first backup deployment is placed at the London site, on the same network with the main Tivoli Endpoint Manager Server. The company can recover quickly by collocating the backup deployment with the production system and by using the same high-speed LAN connection. The second backup deployment is in the data center in Newark; this backup ensures independence from any geographical hazard. The Tivoli Endpoint Manager Server and Report Server are backed up on a daily base in the London data center. The Tivoli Endpoint Manager Server and Report Server are backed up on a weekly basis to the Newark data center, as depicted in Figure 6-3.

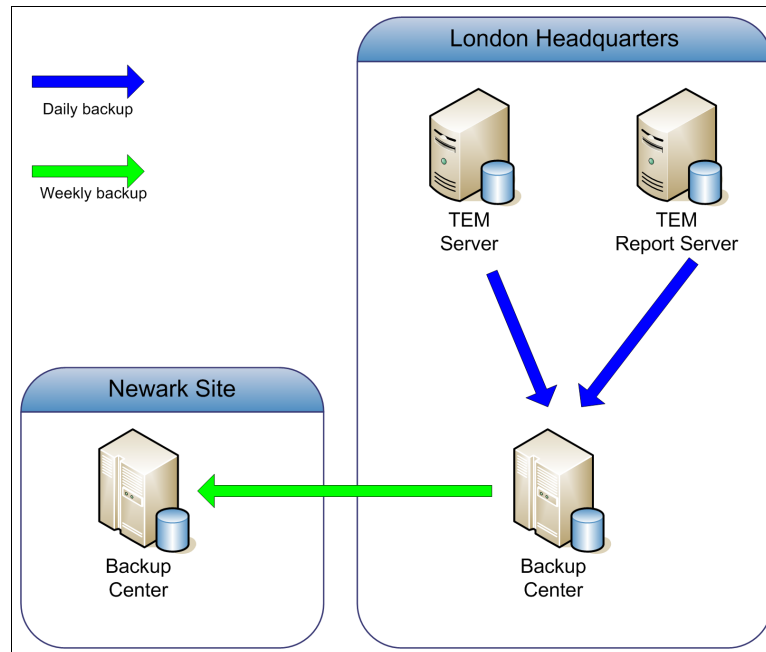


Figure 6-3 Tivoli Endpoint Manager backup

The IT team decided not to invest in a Distributed Server Architecture at this time. The backup and recovery process requirements of regaining production status within 36 hours after a disaster can be achieved by using the deployment architecture that we described. The services that Tivoli Endpoint Manager provides cover deployment, patching, and policy-based solutions. These services require an established connection between Tivoli Endpoint Manager Server and an Agent or Relay only when new content or updates of policies need to be distributed. These operations usually do not require immediate action. Therefore, for many Tivoli Endpoint Manager deployments, several hours of downtime are acceptable.

Hardware

The IT team wants to deploy a Tivoli Endpoint Manager Server to manage the initial 120,000 endpoints, plus offer future expansion capability for 16,000 new employees and endpoints. Based on the hardware proposal from IBM¹, the company purchased an IBM System x with two 8-core processors that run at 2.3 GHz and 64 GB of RAM.

A typical performance bottleneck for a Tivoli Endpoint Manager deployment is storage I/O. Considering future expansion, the financial accounting company decides to invest in two RAID controllers, each of which runs two array slots, as shown in Figure 6-4. The internal controller and its disks are used for the operating system (Microsoft Windows 2008 R2), Tivoli Endpoint Manager application, and the SQL application (Microsoft SQL 2008 R2). The external controller is used for database and transaction logs. In addition to this high-performance setup, the company also uses solid-state drives (SSD) for the arrays to maximize I/O performance.

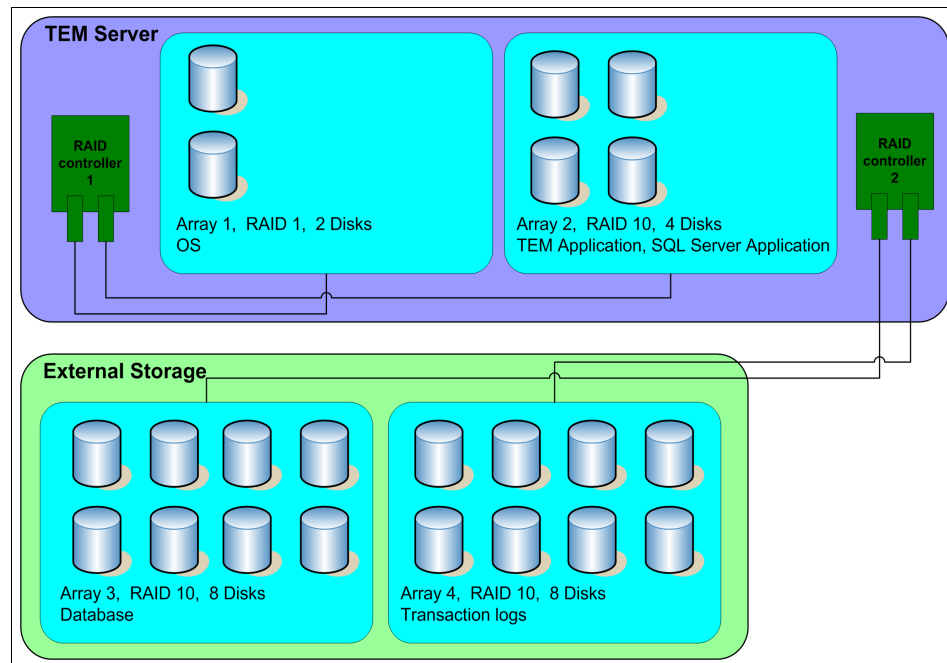


Figure 6-4 RAID array setup for Tivoli Endpoint Manager

¹ For more information about IBM Tivoli Endpoint Manager hardware requirements, see this website: <http://support.bigfix.com/bes/install/serverreq.html>

A gigabit network connection is implemented between the Tivoli Endpoint Manager Server, Report Server, and Console Server to provide high speed transport for large data transmissions between the servers.

Virtualization: The Tivoli Endpoint Manager Server, Relay, and Agent components are fully compatible with VMware, Hyper-V, KVM, and other hypervisors that run the supported OS. However, IBM does not suggest that you host Tivoli Endpoint Manager Server in a virtual machine. Tivoli Endpoint Manager Server relies heavily on I/O performance. A virtualized deployment causes increased I/O overhead that can drastically reduce performance.

For more information about virtualization, see this website:

<http://support.bigfix.com/cgi-bin/kbdirect.pl?id=366>

Deployment

The actual deployment of the Tivoli Endpoint Manager Server does not differ much from other ordinary servers. The IT team tests the Server for reliability and fine-tunes the system for optimized I/O speed. Then, the IT team installs the OS, database, and Tivoli Endpoint Manager Server package.

6.2.3 Tivoli Endpoint Manager Relay

A good Relay implementation strategy is critical to the entire Tivoli Endpoint Manager implementation. A good Relay placement must effectively share the load of the Tivoli Endpoint Manager Server.

Topology plan

According to Table 6-1, the IT team determined the number of Relays to implement for each site. The general rule is to have at least one Relay for each 1,000 endpoints.

Table 6-1 Relay plan

Location site	Number of endpoints	Number of Relays
London	25,234	32
Kracow	4,004	6
Zurich	508	2
Newark	18,030	23
Austin	5,030	7

Location site	Number of endpoints	Number of Relays
Toronto	10,043	12
Sao Paulo	8,022	10
Tokyo	13,110	16
Taipei	28,220	35

The number of Relays might seem more than initially needed. The extra servers are backup for any eventual hardware or connection failure. The number also includes a separate DMZ Relay to support mobile users that connect through a virtual private network (VPN). Because the financial accounting company already has servers that are good Relay candidates, the company purchases new hardware for its Top-Layer-Relays only. The rest of the Relays are shared Relays, made up by installing the Relay application to existing servers. The logical design diagram of the Tivoli Endpoint Manager implementation is depicted in Figure 5-5 on page 207.

Desktop Relay: Relays can also be deployed on desktop computers if the organization has insufficient candidate servers for its Relays. Generally, avoid this configuration, because the Relay adds load to the system, which might bother a user.

The Relay infrastructure deployment is top-down. First, a Tivoli Endpoint Manager Agent is installed on the four Top-Layer-Relays, then the Relay installer is pushed out from the Tivoli Endpoint Manager Server. After all Top-Layer-Relays are successfully deployed, the lower-tier Relays are included in the deployment cycle, and eventually all Relays are deployed successfully.

The IT team reviews its network environment and calculates how many Relays it needs for each site. The company needs at least one Relay for approximately 1,000 endpoints and enough backup Relays per subnet. For more than 50 Relays, the company needs another Top-Layer-Relay to handle the load. Any connection must always traverse the Top-Layer-Relay. No endpoints connect directly to the Tivoli Endpoint Manager Server. See Figure 6-5 on page 222 for the Tivoli Endpoint Manager tier view.

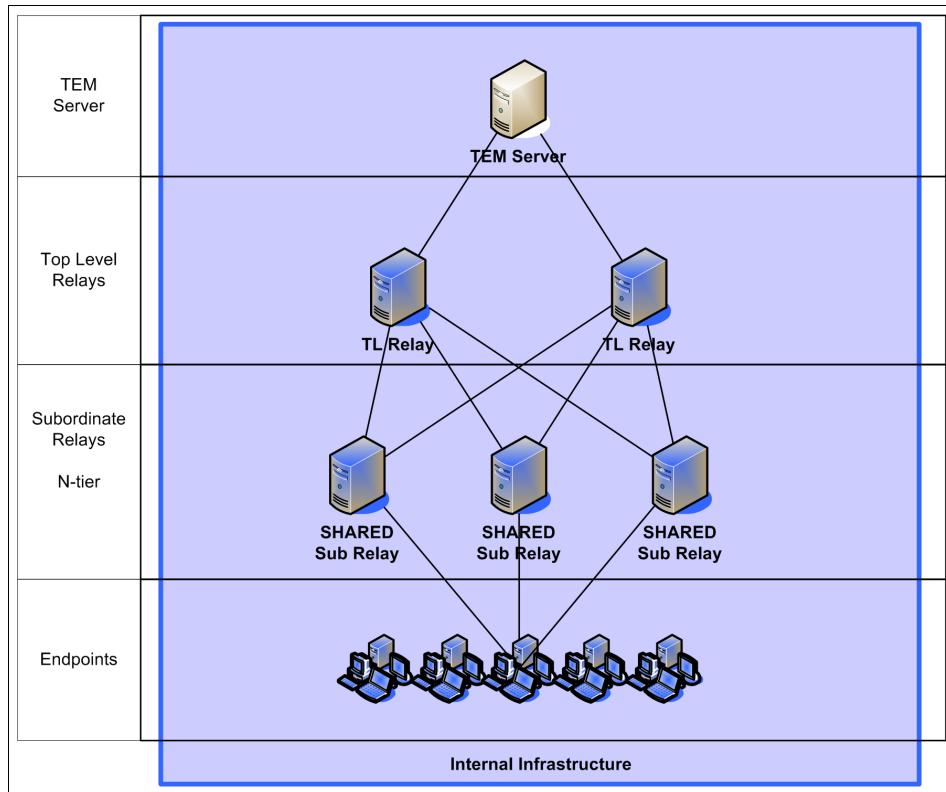


Figure 6-5 Tivoli Endpoint Manager tier view

Extra Relays can be added at any time to serve the branch offices or mobile users. A Relay can help aggregate network traffic, making it more efficient and easier to manage, as depicted in Figure 6-6 on page 223.

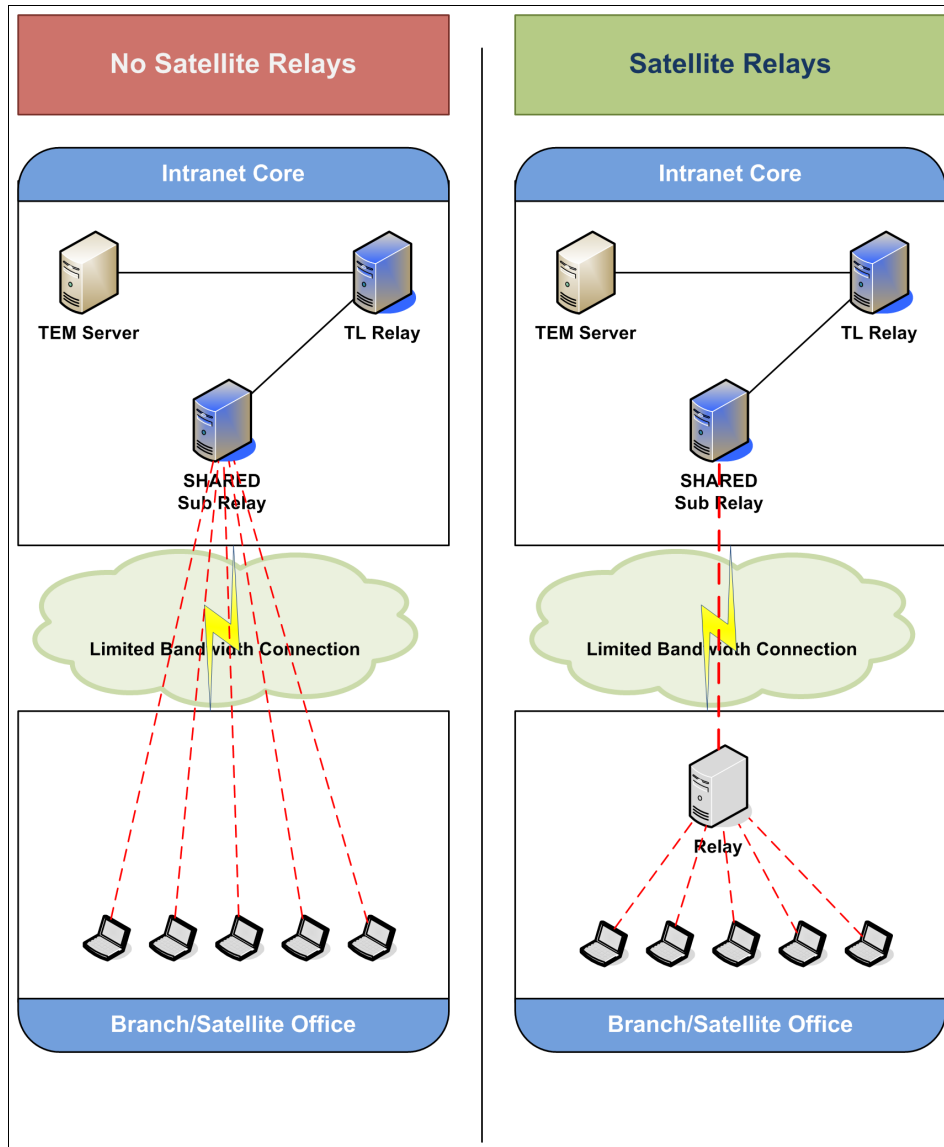


Figure 6-6 Branch office Relay deployment example

Hardware

Tivoli Endpoint Manager Relays are designed to coexist with existing server services. File or print servers, domain controllers, systems management server, and application servers are good candidates to become a Relay. Other servers with a static IP address, little downtime, and enough storage space to host

deployment cache files are good candidates to become a Relay. The financial accounting company wants to deploy Relay services to their existing servers to coexist with the services that already run on the server. This entire implementation is much more cost-effective compared to acquiring new servers.

A Top-Layer-Relay, however, usually creates more load due to the aggregate traffic of its lower-tier Relays. Consider a more powerful server to run a dedicated Relay service to handle the extra load. The IT team decided to purchase four dedicated IBM System x servers for the Top-Layer-Relay servers.

Virtual Machine Relay: Although the virtual machine approach might not be ideal for hosting your Tivoli Endpoint Manager Server, it can be a good candidate for dedicated Relays. Unlike Tivoli Endpoint Manager Server, the I/O bottleneck is not a direct impact to the scale and performance of the Tivoli Endpoint Manager infrastructure. Therefore, the use of Virtual Machine Relays can be an effective and economic solution for the Relay design if the organization already uses a virtualization infrastructure.

Deployment

After the Relay candidates are selected, an Agent is installed on the candidate server. After the Agent is running on the server, the Tivoli Endpoint Manager Relay can be installed by deploying the *Installs the Tivoli Endpoint Manager Relay* Fixlet, as depicted in Figure 6-7 on page 225. The regional IT teams are responsible for all their Relay deployments and the maintenance in their region.

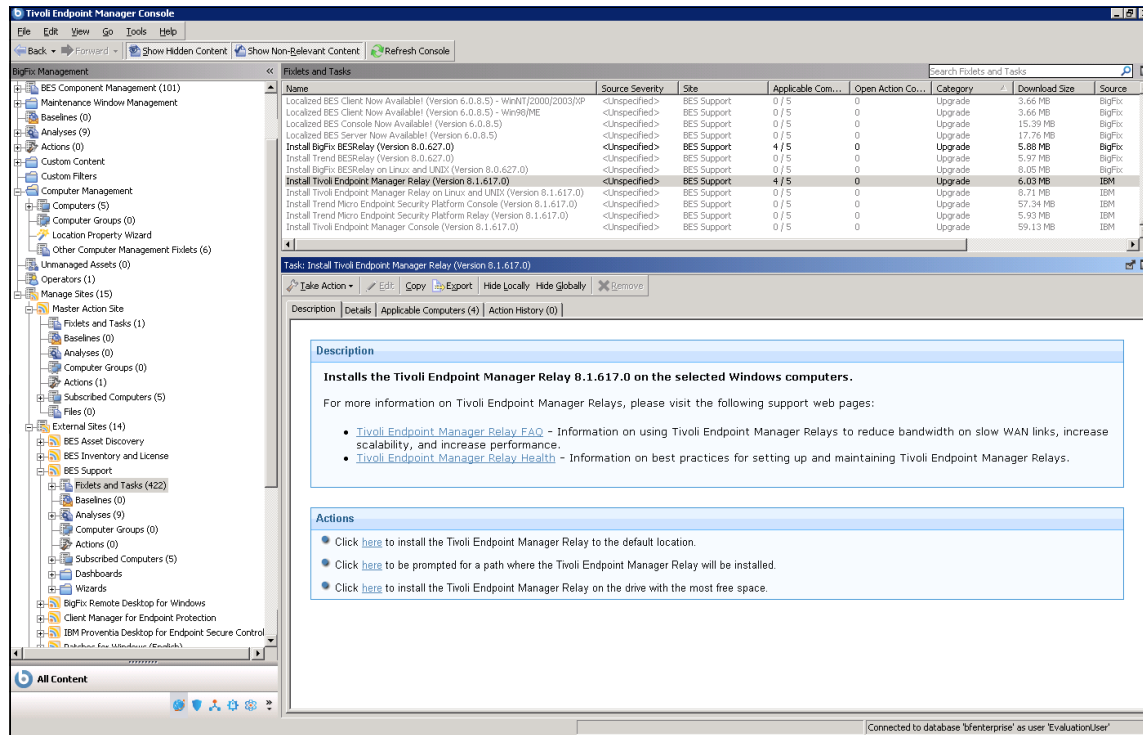


Figure 6-7 Installs the Tivoli Endpoint Manager Relay Fixlet

6.2.4 Tivoli Endpoint Manager Agents

There are several techniques for installing the Agent, including the Client Deploy Tool, login scripts, nonIBM utilities, and a manual installation. The financial accounting company allows its regional IT teams to choose the best way to deploy Tivoli Endpoint Manager Agents according to their local IT infrastructure and policy.

Relay selection configuration

The implementation team knows that using automatic Relay selection can greatly reduce the deployment and maintenance effort to achieve good load balance between Relays. However, the team must consider that if mass Relay or network failure occurs simultaneously, large amounts of *Relay search request packets* can be generated from the Agents, potentially disabling the network. Although that possibility is low in a well-implemented Tivoli Endpoint Manager system, the team wants to take extra measures to eliminate even the slightest possibility. The team decided to adopt a mixed approach that helps them prevent Relay search

request packets from flooding the entire network. This approach offers the benefit of the smart load balancing of automatic Relay selection.

This implementation uses *manual Relay selection* for all Relays, including Top-Layer-Relays. Each site is assigned a few country Relays that are responsible to aggregate the traffic of the country site to the Top-Layer-Relays. The number of country Relays for each site is proportional to the quantities of the endpoints in that site, which is depicted in Figure 6-8. The subordinate Relays attempt to connect to Tivoli Endpoint Manager Server only if it is present in their Relay selection list.

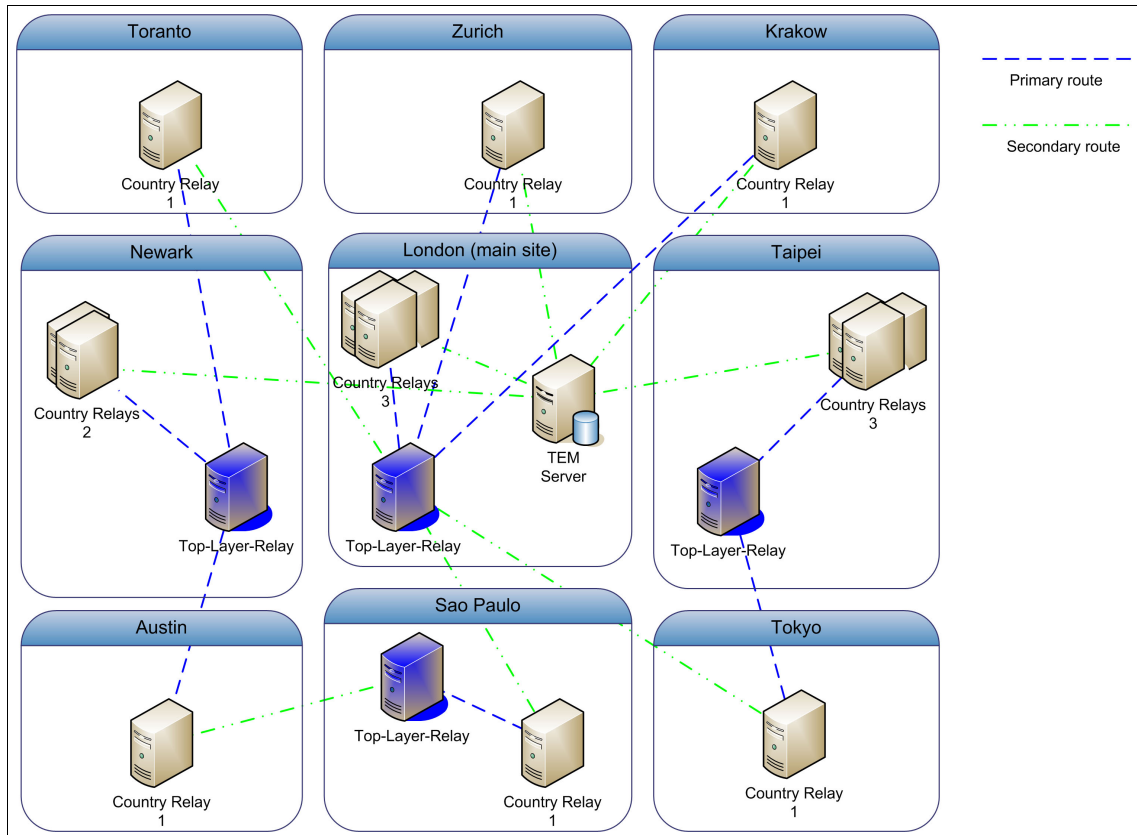


Figure 6-8 Top-Layer-Relay and Country Relay selection illustration

The implementation plan is based on the following considerations:

- ▶ Network route failure is not considered. An alternative network route is expected to be supplied by their routers when the route fails.
- ▶ The Top-Layer-Relay has no secondary Relay choice, because if the connection to Tivoli Endpoint Manager Server fails, it is likely that the Tivoli Endpoint Manager Server is down. A secondary Relay that points to other Relays does not help in that situation.
- ▶ The sites with many Relays (Taipei and Newark) define their second Relay choice as the Tivoli Endpoint Manager Server to avoid overloading the Top-Layer-Relay of other sites.
- ▶ Rather than a long list of backup Relay server routes, the financial accounting company chooses to keep the Relay route simple for easy troubleshooting. All the country Relays have one primary and one secondary Relay choice.

Lower-tier Relays (Relays that are not Top-Layer-Relays or country Relays) are set so that their country Relays are the primary Relay, and the Top-Layer-Relay is the secondary or tertiary Relay.

Endpoints that do not run Relay services use the automatic Relay selection with Relay affiliation. Regional affiliation groups are broken down into North America (NA), Latin America (LA), Europe, the Middle East, and Africa (EMEA), and Asia Pacific (AP). The endpoints and Relays of each region have their own regional affiliation group, and connections are made in their own region only. No endpoints are allowed to connect to the Tivoli Endpoint Manager Server directly under any circumstances.

Relays in a DMZ of a major site belong to an additional affiliation group, called DMZ, which enables mobile endpoints to connect to them. Endpoints with mobile capabilities, such as notebooks, are assigned to both their own regional group and the DMZ affiliation. When the endpoint roams and cannot connect to the regional Relay, it automatically tries to connect to the DMZ Relays (Figure 6-9 on page 228).

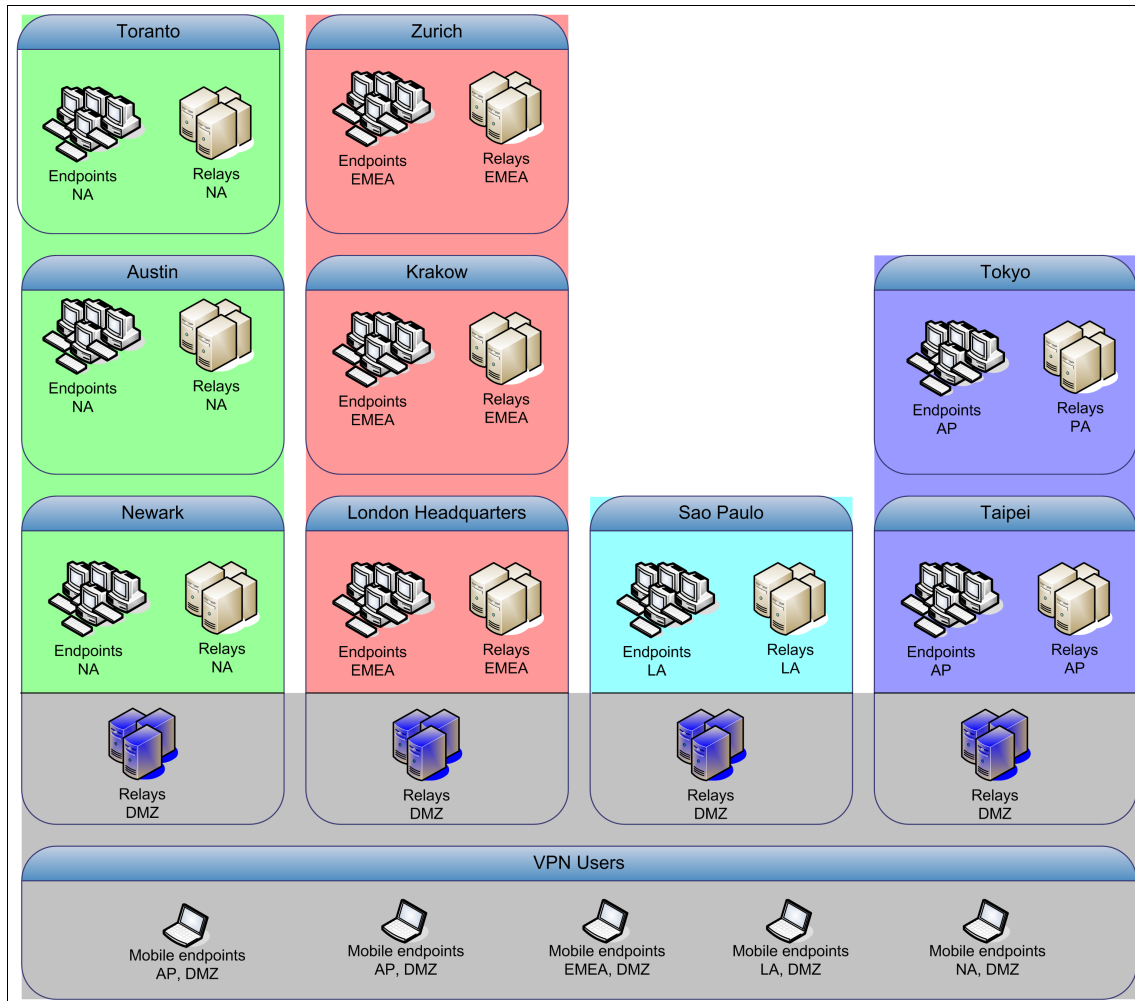


Figure 6-9 Regional Relay affiliation setting for the financial accounting company

6.2.5 Asset discovery

To comply with the policy, the financial accounting company needs to ensure that all endpoints are managed by Tivoli Endpoint Manager so that the company can receive Patch Management and Security Compliance services. Although the IT team thinks it can track all endpoints in the organization, the team wants to eliminate the possibility of any unmanaged or unknown endpoints in the network. Tivoli Endpoint Manager *Asset Discovery* is designed to locate any unmanaged endpoints.

After the team successfully deployed Tivoli Endpoint Manager, each regional IT team starts to select the *Scan Point candidates* in the region of responsibility. Scan Points can be deployed to any Windows system that runs Windows XP, Windows 2003 Server or later, or Red Hat Enterprise Linux release 5. Avoid a single Scan Point scanning a large subnet, because it can take a long time to finish. Choose a system that is constantly on during the scanning period.

Each regional IT team plans to execute its own asset discovery procedures to best suit its network environment and security policy. Figure 6-10 is an example of the asset discovery procedures in a small section of the London office. By using the Tivoli Endpoint Manager Console, navigate to **Sites** → **External Sites** → **BES Asset Discovery** → **Fixlets and Tasks**. Select **Designate Nmap Scan Point** and follow the instructions.

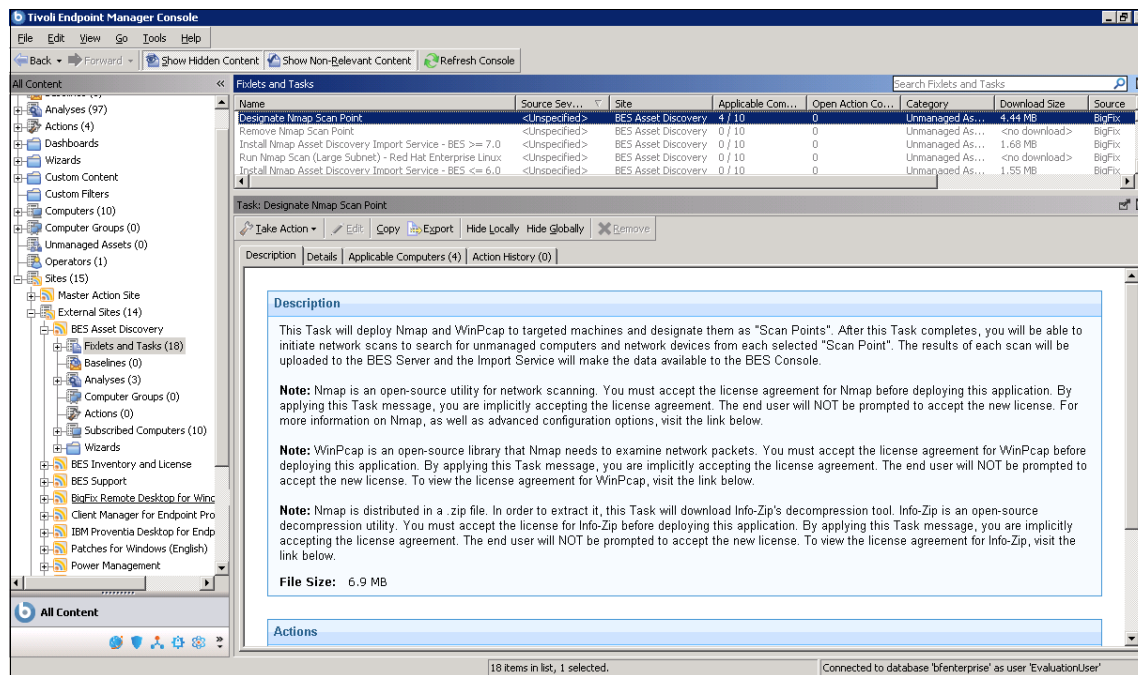


Figure 6-10 Asset discovery

The London IT team sets the discovery search process to run for one week. The team informs the network security team that during that period, the team expects to receive security events from their network IPS devices. The asset discovery service uses an NMAP scanner to search for endpoints. Because these events have no malicious intent, they are handled through an exception process in the network security group. After the discovery period is finished, the active search

procedures from all Scan Points are deactivated, and the Scan Points are removed from all designated endpoints.

6.3 Maintenance

After Tivoli Endpoint Manager is successfully deployed, the IT team focuses on keeping Tivoli Endpoint Manager running as smoothly as possible. For the scale of the deployment, the daily data throughput can be large. For Tivoli Endpoint Manager to continue running efficiently, the IT team needs a maintenance plan for the Tivoli Endpoint Manager database and the Tivoli Endpoint Manager Server. This information is covered in this section.

6.3.1 Tivoli Endpoint Manager health check

Tivoli Endpoint Manager provides a set of tools to monitor and diagnose the health of the entire Tivoli Endpoint Manager system. The IT team uses these tools on a regular basis to ensure that the system performs as designed.

Tivoli Endpoint Manager Server diagnostic tool

To ensure that the Tivoli Endpoint Manager Server runs properly, the IT team activates the Tivoli Endpoint Manager Server diagnostic tool (Figure 6-11 on page 231). With this tool, you can verify that the server passes all the checks to verify the correct operation of services, connections, and databases. A failed test is displayed with a red cross. A fully functional Tivoli Endpoint Manager Server is expected to pass all tests.

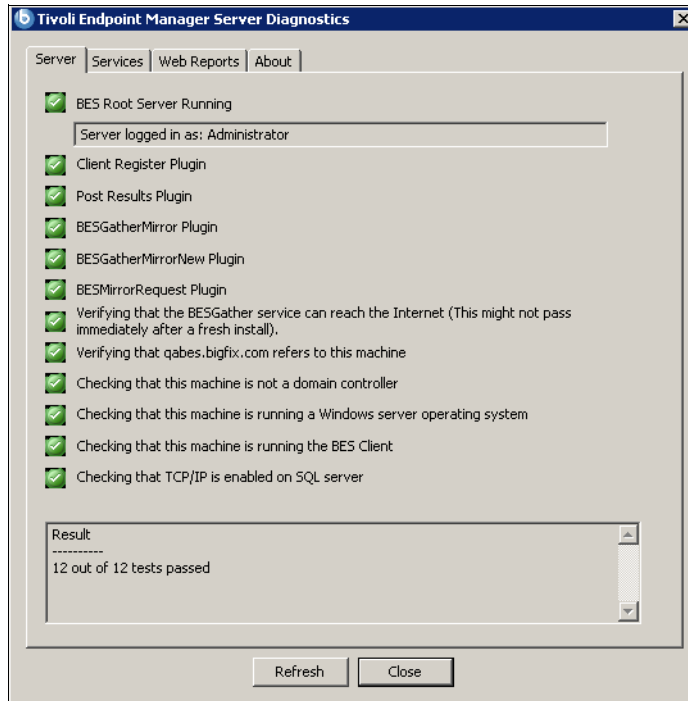


Figure 6-11 Diagnostic tool

BES Support site

After the Tivoli Endpoint Manager systems are deployed, they are automatically subscribed to the *BES Support site*. The site provides Fixlets for troubleshooting, support, performance tuning, and Tivoli Endpoint Manager component upgrades. Operators must monitor the BES Support site regularly for active Fixlets or Tasks that address any potential problems, such as stopped Tivoli Endpoint Manager client services, low disk space, and component updates.

The BES Support site also provides configuration settings for the Server, Relay, and Agent, such as to throttle download traffic, enable automatic Relay selection, and set client CPU usage (Figure 6-12 on page 232).

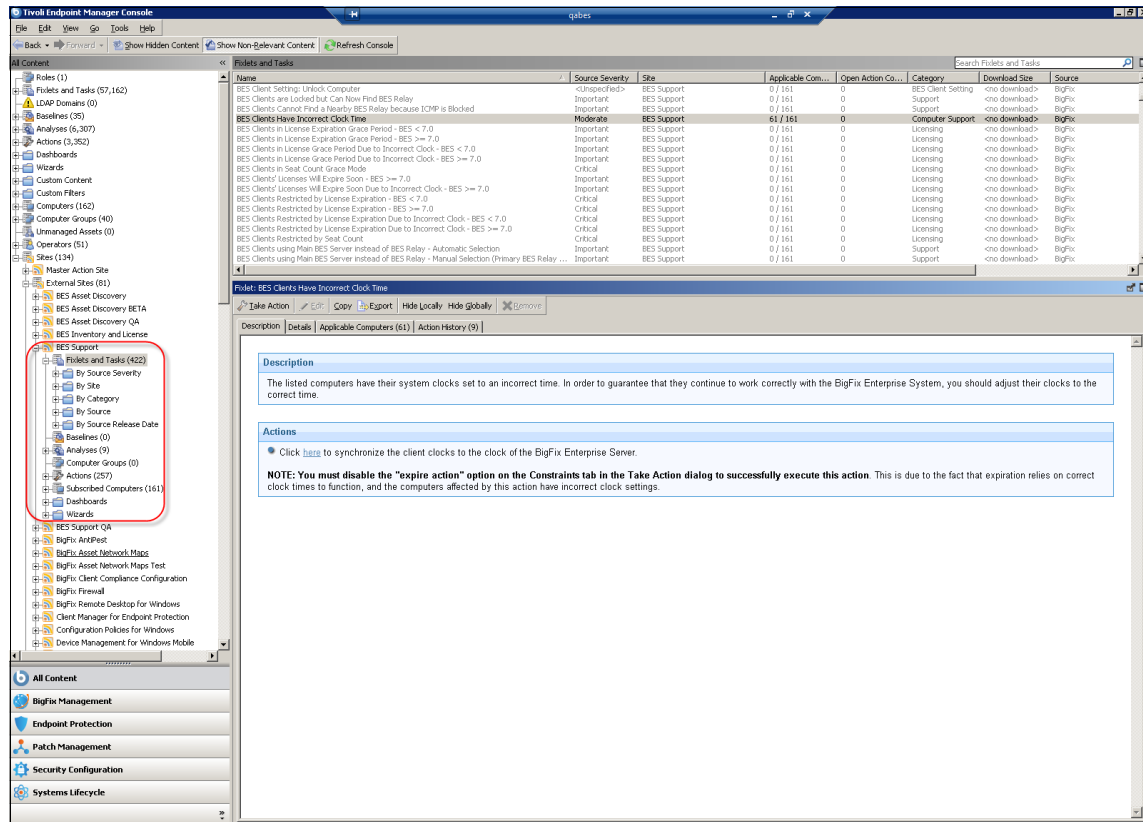


Figure 6-12 BES Support site

6.3.2 Performance tuning

For large Tivoli Endpoint Manager deployments, such as the financial accounting company deployment, performance is important. Without the correct configuration and settings, performance can degrade quickly as the organization grows.

Tivoli Endpoint Manager is designed to operate in large enterprise environments. It provides configuration parameters that can be set on endpoints to modify Agent, Relay, and Server behavior to increase efficiency. For the suggested settings in large deployments, see “Configuring for better performance” on page 154.

In addition to tuning parameters for individual components, we need to look at the network environment and how to tune the required Tivoli Endpoint Manager bandwidth.

Bandwidth throttling

Most of the operational sites of the financial accounting company deployed a modern network facility that provides sufficient bandwidth. The site in Taipei, however, experiences slow Internet connection due to rapid growth. Employees are saturating their current network bandwidth. The situation intensifies when a major patch release is available, for example, a set of large operating system patches released by Microsoft on a particular day of the week.

The IT team in Taipei decided to use bandwidth throttling to reduce the patching traffic for the present, before upgrading the networking infrastructure (Figure 6-13 on page 234). By using the Tivoli Endpoint Manager Console, navigate to **Sites** → **External Sites** → **BES Support** → **Fixlets and Tasks**. Select **BES Relay Setting: Download Throttling Task**. Next, click the first Actions item to set a specific limit for the download traffic in bytes/seconds. This configuration is called *manual throttling*.

As an option, you can use *dynamic throttling*. By using dynamic throttling, the operator can define the allowed percentage of bandwidth when other downloading is in place. If Tivoli Endpoint Manager detects no other downloads, the Tivoli Endpoint Manager downloads uses all the available bandwidth.

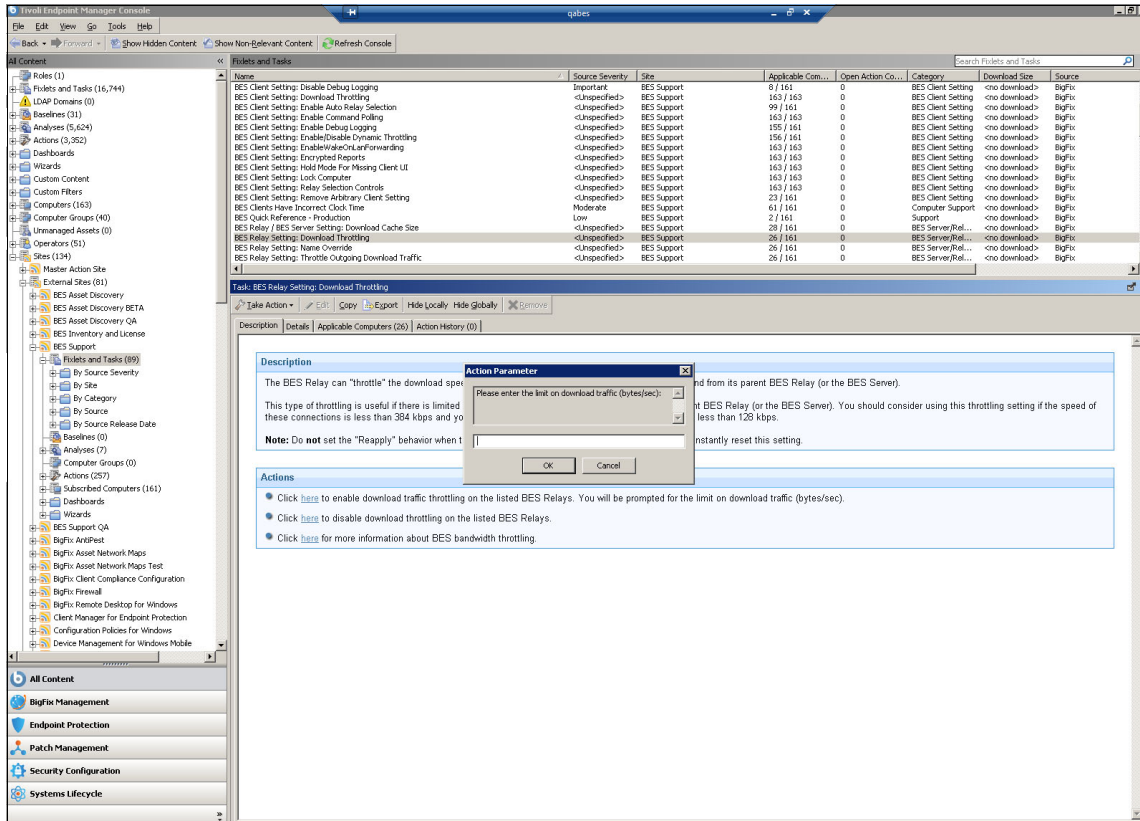


Figure 6-13 Bandwidth throttling

By using a Tivoli Endpoint Manager solution, operating system and application patch downloads from the software vendor websites through the Internet are handled by the Tivoli Endpoint Manager Server. This approach is more efficient than every endpoint handling individual downloads, as depicted in Figure 6-14 on page 235. This approach effectively controls and reduces network traffic during patching.

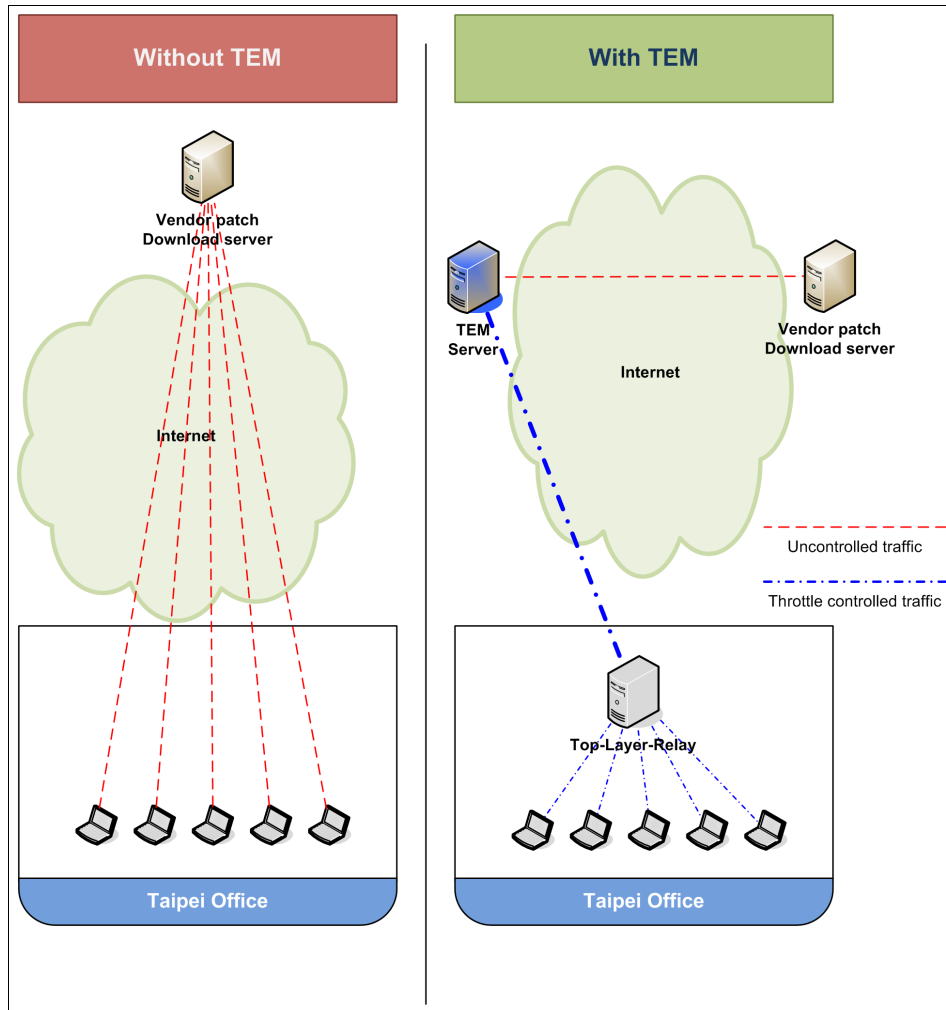


Figure 6-14 Download traffic comparison

The Taipei IT team defines the following scheduled Task, BES Client Setting: Download Throttling. The team limits Tivoli Endpoint Manager download bandwidth from 8 am and then disables the setting at 6 pm. This setting minimizes the effect of patching traffic during the site operating hours.

6.3.3 Maintenance plan

Both the Tivoli Endpoint Manager database and Tivoli Endpoint Manager Server need periodic maintenance to ensure that they run at peak performance. Daily and weekly backups are also required as part of the disaster recovery plan.

Database maintenance

The required maintenance tasks for the Tivoli Endpoint Manager database do not differ much from standard maintenance tasks for any other database server. For Tivoli Endpoint Manager Server to run at peak performance, up-to-date database indexing is important. Operators can shrink the database as part of the regular maintenance tasks, but an index update is required after shrinking the database.

Tivoli Endpoint Manager Server maintenance

After extensive use of Tivoli Endpoint Manager, the system can experience degraded performance due to various causes, for example, unreported endpoints or abuse of Custom Sites and groups.

The overall policy grants only the IT team in the UK the privilege to create Custom Sites and computer groups, so that Custom Sites and computer groups can be controlled and managed centrally. Dedicated maintenance engineers are responsible to check for open Actions, and stop these Actions when these Actions are no longer needed. The maintenance engineers also run the BigFix Computer Remover on a nightly basis. The maintenance engineers run the BigFix Audit Trail Cleaner biweekly to clean up data in the Tivoli Endpoint Manager database that is no longer needed. The following steps show a typical sequential maintenance workflow:

1. Run the BigFix Computer Remover and BigFix Audit Trail Cleaner.
2. Shrink the database.
3. Update the database index.

Tivoli Endpoint Manager utilities: Tivoli Endpoint Manager utilities are available for public download at the Tivoli Endpoint Manager wiki website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/TEM%20Utilities>

Users must understand that the tools can cause serious impact on your Tivoli Endpoint Manager system if misused. Follow the instructions carefully and always back up your Tivoli Endpoint Manager database.

6.4 Conclusion

In this chapter, we describe the design approach that the financial accounting company uses to design the Tivoli Endpoint Manager implementation plan. We outline the business requirements and associated functional requirements. We also explained how the financial accounting company deployed Tivoli Endpoint Manager and how the company overcame difficulties during the deployment stage. In the last part of the chapter, we describe the maintenance processes and practices of the financial accounting company. We introduce the use of support tools to assist the maintenance processes.

This chapter explains how the financial accounting company deployed the Tivoli Endpoint Manager platform into its IT landscape. In Chapter 7, “Phase II: Patch Management design and implementation” on page 239, we describe to implement a Patch Management solution.



Phase II: Patch Management design and implementation

In this chapter, we describe the approach of the financial accounting company with the IBM design team to define the corporate Patching Policy for its endpoints. We examine the available options for implementing a process for patching, differentiating between servers and workstations. This chapter then describes how the financial accounting company implements the patching process by using Tivoli Endpoint Manager. We describe why the patching process chosen was appropriate for the financial accounting company. We also cover the required maintenance. We divide the discussion into the following sections:

- ▶ “Design” on page 240
- ▶ “Implementation” on page 256
- ▶ “Maintenance” on page 283

7.1 Design

This section introduces the situation that the financial accounting company faces in 5.1, “Organization profile” on page 188 and the strategic direction of the business. We also describe the current security Patching Policy and how it influences the deployment of Tivoli Endpoint Manager. We define the business and technical requirements of the organization. We conclude with patching processes to help to enforce the corporate security patching policy.

Perspective: Tivoli Endpoint Manager as a platform for patching endpoints offers a range of options for using the functionality that it offers to deliver the requirements of the organization. The methods displayed in this scenario are not necessarily the only or best approaches for all organizations. The methods that we chose are selected to demonstrate the flexibility of the solution and to highlight more potential uses.

For more information about preferred practices, see 4.2, “Tivoli Endpoint Manager solution design” on page 142.

For more detail about the overall business requirements and how they map to functional and technical requirements, see 5.4, “Functional requirements” on page 198.

Scenario: This section describes a scenario for defining policies. We do not intend to implement and document an actual security strategy with all its requirements, and details in this book. We create a high-level endpoint security patch policy definition with a range of requirements to use as fictional high-level policies.

7.1.1 Introduction

The financial accounting company estimates a current endpoint total of 120,000. These endpoints are distributed globally across all continents with the major servers in the data center in London, United Kingdom. Servers are also in each location to support local business functions. The organization also supports a large mobile workforce. This workforce uses its own Internet connection in a home office environment and a virtual private network (VPN) to connect to the corporate network.

The organization wants to further expand its operations with new locations across Europe and Asia. The infrastructures of these sites differ with varying speeds to connect to the corporate network. This expansion adds 15,000 staff to the overall headcount. The new subsidiary for credit card processing adds 1,000 staff. These expansion plans require a solution that is scalable enough to handle many additional endpoints that run various operating systems.

The following list outlines how the financial accounting company currently patches the endpoints:

- ▶ Servers

The financial accounting company currently defines the Patching Policy for its servers at a local level. No deadlines exist for patching applications or operating systems, therefore there is compliance baseline to which to adhere. The responsibility for patching the servers is with the system administrators, who are already overloaded. The current method for patching servers is for the individual administrators to request tests of patches at a local level and then send those patches for quality assurance. Patches ready for production require staff to travel to the data center location to use a console and media, either CD or USB device, to transfer and apply the patch to the server. This lengthy process can take days and involve extra expense for staffing, travel, and living expenses.

- ▶ Desktops and notebooks

The responsibility for patching these types of assets is delegated to the users of those systems. The only loosely enforced mandate is that the local built-in operating system update functionality is enabled. This way, all patches are downloaded as soon as they are available from a vendor. The help desk of the organization experiences many support requests when each workstation and notebook downloads the same version of the vendor-released patch. This approach affects the performance of services for the business to operate because bandwidth is saturated. In some instances, personnel who travel on business use a General Packet Radio Service (GPRS) connection to the Internet, which is a low-bandwidth connection.

7.1.2 Defining business requirements

A corporate security policy for the servers exists and is implemented locally in each operating country. These systems serve organization-critical an application named *Quant Unlimited Access Network for Traders (QUANT)*.

The financial accounting company has many endpoints that can access the corporate network where sensitive data exists. In many cases, there is no centralized management of the endpoints that access the network from outside of the organizational firewall. These endpoints are distributed globally and can

run on many types of platforms. The operating systems that run on these endpoints can be AIX, Linux, Windows, or Macintosh.

We identified the following security exposures that the financial accounting company needs to address for patch management:

- ▶ Create a consistent corporate security policy for patching.
- ▶ Create a consistent corporate security compliance plan for patch management.
- ▶ Define a patching process for servers and other endpoints.
- ▶ Enforce patching for endpoints to maintain a consistently secure operating stance.
- ▶ Implement the patch management of endpoints outside of the corporate firewall.
- ▶ Create visibility of all of the endpoints.
- ▶ Control the endpoints for the ability to patch.

7.1.3 Defining functional requirements

The financial accounting company specified the following high-level functions that the patch management solution must provide:

- ▶ The ability to review the existing set of security policies and update them with new requirements that translate the corporate security policy.
- ▶ The ability to patch the heterogeneous environment at the financial accounting company and across the range of operating systems supported for workstations, servers, and notebooks.
- ▶ The ability to patch third-party applications.
- ▶ A centralized management infrastructure for the patch solution and the flexibility for the local teams to enforce their security policies while allowing the team to address the compliance requirements of the corporation.

7.1.4 Patching rating and policy design

In this section, we look at the *Patching Policy* design based on the *Vendor Severity Classifications* and *Patching Class Rating*. The financial accounting company is concerned with how it defines whether an endpoint needs to receive an available patch. The company has several IT security teams that manage many security and patching policies for their local clients. We look at how the IT security teams can best determine the important policies.

Context: The Vendor Severity Classifications refer to the criticality that is reported by vendors. The Tivoli Endpoint Manager Console shows this classification as “Source Severity”.

Patching Class Rating means to evaluate the Vendor Severity Classifications and to create your own specific classes.

The Patching Policy is the most suitable Patching Class Rating and a grace period that can be selected by an authorized person or team.

Vendor Severity Classifications

We see severity classifications for operating systems, applications, and middleware that are determined by each vendor when a new patch is released. First, you need to acknowledge the types of classifications that are defined for your patches. We explain in detail about how the financial accounting company evaluates these classifications in “Patch Class Rating” on page 245.

What and how many patches and classifications an organization needs depends on the situation. The financial accounting company needs to evaluate more closely the classifications for Microsoft Windows, Red Hat Enterprise Linux, IBM AIX, Adobe Acrobat Reader, Mozilla Firefox, and Oracle Java. We look at the types of severity classifications that these vendors release:

- ▶ Microsoft Windows

The financial accounting company uses Microsoft Windows XP and 7 for its clients, and Microsoft Windows 2003 and 2008 for its servers. Microsoft provides four severity classifications that are Critical, Important, Moderate, and Low. For more information, see the Microsoft Security Bulletin:

<http://www.microsoft.com/technet/security/current.aspx>

- ▶ Red Hat Enterprise Linux

The financial accounting company uses Red Hat Enterprise Linux Version 5 for some of its servers. Security, Bug Fix, and Enhancement Advisory patches are available. Red Hat releases severity classifications for Security Advisories that are Critical, Important, Moderate, and Low impact. For more information, see the Red Hat web page:

<https://access.redhat.com/security/updates/classification/>

- ▶ AIX

The financial accounting company uses IBM AIX for its database servers. On the Tivoli Endpoint Manager Console, we see four kinds of source severities for AIX patches that are Technology Level/Service Pack, Security Advisories, High Impact/Highly Pervasive Fixes, and PTFs in Error, which are

classifications of AIX patches. For more information about AIX, see the IBM Fix Central site:

<http://www.ibm.com/support/fixcentral/>

► Adobe Acrobat Reader

Adobe Acrobat Reader is required to work with many documents at the financial accounting company. Adobe usually releases two kinds of patches: a security-related patch and a patch that is released quarterly. Their severity classifications are Critical, Important, Moderate, and Low. For more information about Adobe, see this website:

<http://www.adobe.com/support/security/#readerwin>

► Mozilla Firefox

Most applications of the financial accounting company are web-based. Mozilla Firefox is defined as the standard for employees worldwide. Firefox patches are released with severity classifications, such as Critical, High, Moderate, and Low. For more information, see the Mozilla website:

<http://www.mozilla.org/security/known-vulnerabilities/>

Table 7-1 provides a summary of the vendor classifications.

Table 7-1 Vendor Severity Classifications

Vendor and OS/application	Vendor Severity Classification
Microsoft Windows	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate ▶ Low
Red Hat Enterprise Linux (for Security Advisories)	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate ▶ Low
IBM AIX	<ul style="list-style-type: none"> ▶ Technology Level/Service Pack ▶ Security Advisories ▶ High Impact/Highly Pervasive Fixes ▶ PTFs in Error
Adobe Reader	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate ▶ Low

Vendor and OS/application	Vendor Severity Classification
Mozilla Firefox	<ul style="list-style-type: none"> ▶ Critical ▶ High ▶ Moderate ▶ Low
Oracle Java	<ul style="list-style-type: none"> ▶ Critical ▶ High ▶ Moderate ▶ Low

Tivoli Endpoint Manager can manage many patches for operating systems and applications through a single console. For information about patching, see 4.3.1, “Before you patch” on page 157.

Patch Class Rating

Next, the financial accounting company needs to estimate the severity and impact to its business if it does not apply patches to the endpoints. The financial accounting company plans to determine the impact by using the severity classifications that are provided by vendors. The financial accounting company uses Patch Class Ratings to create specific patching classes based on the Vendor Severity Classifications. In this phase, the financial accounting company solely determines patching classes. The company does not describe in depth how to manage patching grace periods and local policy differences from each of the IT security teams.

As mentioned in “Vendor Severity Classifications” on page 243, the vendors release their patch information and use specific classifications, which must be assigned to the financial accounting company Patch Class Rating. Table 7-2 shows the Patch Class Rating for all applicable systems. These settings are common among all local IT security teams.

Table 7-2 Patch Class Rating

Vendor and OS/application	Class “Urgent”	Class “Essential”	Class “Advisable”
Microsoft Windows	<ul style="list-style-type: none"> ▶ Critical ▶ Important 	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate 	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate ▶ Low
Red Hat Enterprise Linux	<ul style="list-style-type: none"> ▶ Critical ▶ Important 	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate 	<ul style="list-style-type: none"> ▶ Critical ▶ Important ▶ Moderate ▶ Low

Vendor and OS/application	Class “Urgent”	Class “Essential”	Class “Advisable”
IBM AIX	<ul style="list-style-type: none"> ▶ Technology Level/Service Pack ▶ Security Advisories 	<ul style="list-style-type: none"> ▶ Technology Level/Service Pack ▶ Security Advisories ▶ High Impact/Highly Pervasive Fixes 	<ul style="list-style-type: none"> ▶ Technology Level/Service Pack ▶ Security Advisories ▶ High Impact/Highly Pervasive Fixes ▶ PTFs in Error
Adobe Acrobat Reader	<ul style="list-style-type: none"> ▶ Critical 	<ul style="list-style-type: none"> ▶ Critical 	<ul style="list-style-type: none"> ▶ Critical ▶ Important
Mozilla Firefox	<ul style="list-style-type: none"> ▶ Critical 	<ul style="list-style-type: none"> ▶ Critical 	<ul style="list-style-type: none"> ▶ Critical ▶ High
Oracle Java	<ul style="list-style-type: none"> ▶ Critical Patch Updates 	<ul style="list-style-type: none"> ▶ Critical Patch Updates 	<ul style="list-style-type: none"> ▶ Critical Patch Updates ▶ Security Alerts

Determining the patching policy

Finally, the financial accounting company needs to determine due dates for patching, which means how many days the company allows for the completion of the deployment of patches to all managed clients. These *grace periods* must not depend on the number of clients. Grace periods need to be common among all teams for appropriate IT compliance management.

The financial accounting company does not allow individual IT security teams to determine their own grace periods. The individual IT security teams collaborate with the international teams and define the most suitable policy to which each team can agree. The result of this discussion is depicted in Table 7-3.

Table 7-3 Patching grace periods (business days)

Vendor and OS/application	Class “Urgent”	Class “Essential”	Class “Advisable”
Microsoft Windows	3	10	20
Red Hat Enterprise Linux	3	10	20
IBM AIX	3	10	20
Adobe Acrobat Reader	5	10	25
Mozilla Firefox	5	10	25

Vendor and OS/application	Class “Urgent”	Class “Essential”	Class “Advisable”
Oracle Java	5	10	25

Determining the *Patching Policy* consists of choosing the most suitable Patch Class Ratings and patching grace periods. The financial accounting company allows its local IT teams to choose the most suitable Patch Class Ratings and patching grace periods, because the teams understand their local IT environment well and know what patches are needed. Table 7-4 depicts an example that shows the Patch Class Ratings and grace periods for Microsoft monthly security updates.

Table 7-4 Microsoft monthly security updates for Windows in 2011 August

Local IT security team	Patch class rating	Patching grace period
London	Urgent	3
Krakow	Advisable	20
Zurich	Essential	10
Newark	Urgent	3
Austin	Essential	10
Toronto	Essential	10
Sao Paulo	Essential	10
Tokyo	Urgent	3
Taipei	Urgent	3

Conclusion for patching rating and policy design

Many organizations operate on a worldwide scale. These organizations must comply with various requirements for patching, such as the financial accounting company. Our suggestion is that an organization authorizes an appropriate person or team, that understands and acknowledges what security matters are relevant to the IT environment, to choose suitable patches that are required to be deployed.

7.1.5 Defining the patch management process

The prevalence of security exploits that come as malicious code that targets known vulnerabilities in operating systems and vendor software exposes

organizations to an increased amount of risk. Organizations must react faster than they did previously. IT teams must be flexible and responsive in managing the IT assets of their organization. There are important ingredients to consider for a patch process:

- ▶ Visibility to quickly detect where the exposures exist
- ▶ Agility where speed is of the essence when security is involved
- ▶ Efficiency to perform all patching tasks without affecting IT services
- ▶ Continuity to keep all systems patched at all times
- ▶ Usability to easily and consistently implement all these actions

Initially, the expense of remediating a system affected by a vulnerability was a driver for patching machines. This viewpoint recently expanded to include more subtle forms of intrusion that can expose customer data and affect the reputation of an organization among its customer base. There are several high-profile examples. Regulatory compliance is another factor that continues to mandate that systems remain updated and properly patched. As the number of endpoints increases and the employees of organizations work in more effective and mobile ways that change the risk posture of an organization, patching can quickly escalate to a top and fundamental security priority.

In 7.1.4, “Patching rating and policy design” on page 242, we examined fundamental parts in formulating a patch management policy. We examined how an organization receives and defines whether the patches from vendors are important in a business context. Essentially, is this available patch relevant to my company? By using Tivoli Endpoint Manager for your patch management solution, you can provide all patch releases from vendors and the relevant patches that can be applied for the operating systems and software within the environment. Tivoli Endpoint Manager has visibility of all deployed assets that are connected to the network. The financial accounting company decided to take further control of its servers and select from this relevant content which patches the company wants to apply.

After consultation, the financial accounting company decided to use the security severity that was assigned to each patch by the vendors. This decision enables the financial accounting company to standardize the definition of a vendor security severity rating and translate that definition into the potential organizational impact to the business.

The financial accounting company uses local security policies defined at each operating site. This approach allows for autonomy at a local level. By classifying the asset risk in each operating country, a standardized rating is formulated based on the potential impact to the organization if its operation is affected.

By taking the vendor patch security severity and combining it with the risk to which each asset is exposed, the financial accounting company can define a

baseline for compliance. For example, if a patch is rated as Critical, you might be required to patch a system that interacts with the Internet within 48 hours, a notebook within five days, and any data center system within 30 days. Patching systems within these time frames can be stated as a *compliance mandate*.

The critical business functions at the financial accounting company are processed on servers at the main data center in London. There are a few local business functions served up in other locations. Patching critical systems, whatever the method used, involves an element of risk. Patching a system is essentially modifying parts of the software, which can potentially cause conflicts. There are various aspects of a system that interact differently with a shared collection of files. *Patch management* is the process implemented to mitigate these risks.

Patch management: The impact to the business, if a service is interrupted, must be carefully balanced with the security requirements and compliance mandate to secure each server. The potential risk of not patching servers poses a greater danger to many organizations, but both of these risks must be mitigated as much as possible. How an organization handles each of these situations depends on the tolerance for risk of the organization in general.

7.1.6 Designing a patch management process

After consulting with the financial accounting company, it is clear that the company tolerance for risk is low. But, the company also requires a patch management process to ensure business continuity. A common approach to handling new patches is to test the patch in environments that replicate the system in production to confirm the success or failure of the patch. The level of testing ideally includes the complexity of the environment to which the patch is applied and the criticality of the asset that is patched. At latter stages, organizations might want to phase the approach to the deployment of the tested patch to groups within the production environment.

The financial accounting company decided on standardizing a testing process for each patch that entered into the production server environment. The mechanics of these tests are also standardized and performed at a local level in the country where the service is provided.

By using Tivoli Endpoint Manager, the financial accounting company can implement a new process for patching servers that run in the production environment. Tivoli Endpoint Manager is the centralized platform for enforcing endpoint security patch content. The core of this deployment is the Tivoli Endpoint Manager Server. The Server is accessed by operators responsible for each stage of the patch management process. IBM worked with the financial

accounting company to define the following features of the patch management process:

- ▶ Specific patches are selected from a set of relevant content made available by Tivoli Endpoint Manager.
- ▶ A centralized test team is established to handle the overall patch management process (possibly a virtual global team).
- ▶ A local IT team is established to handle patches so that compliance can be monitored at a country level.
- ▶ The separation of duties is defined between the team that approves the patches and the central test team.
- ▶ A patch cycle is established to be executed one time each month.

The diagram in Figure 7-1 shows the patch management process for servers.

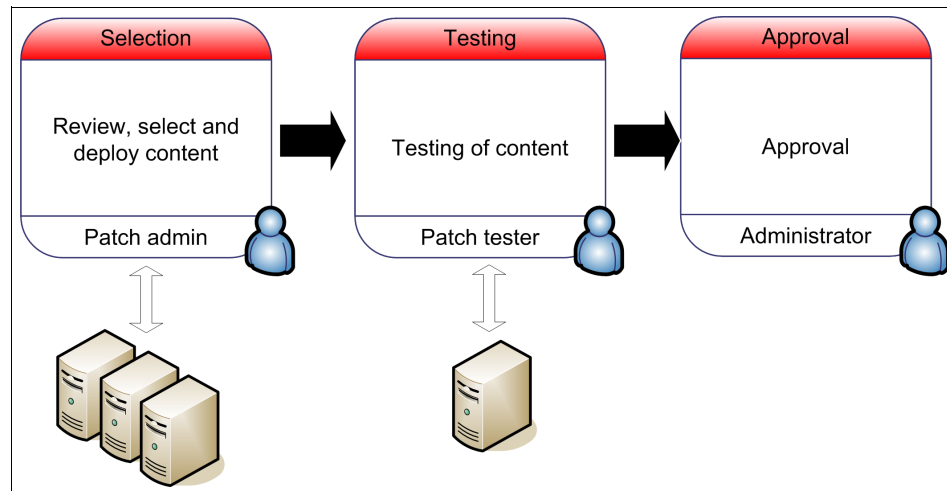


Figure 7-1 The patching process

Tivoli Endpoint Manager provides several tools for operators of the Console to deploy patches to machines. We list each tool with a brief explanation. For a more detailed description of each tool, see 3.1, “Logical component overview” on page 64.

- ▶ Site and Custom Site

A *Site* provides collections of Fixlet messages that are created internally by you, by IBM, or by other vendors. You subscribe to a Site and agree on a schedule for downloading the latest batch of Fixlet messages. The financial accounting company can use a *Custom Site* to hold patches that are approved before they are deployed.

► Baselines

A *Baseline* acts as a container for Fixlet messages and can be deployed to a single system or a group of systems. A Baseline can be an effective way to deploy Actions across the entire network with a single command. For example, you might create a Baseline named “All critical hotfixes” and populate it with all the current critical hotfixes available in the Fixlet list. Or you might create a Baseline named “Finance department baseline” to keep that particular group of computers updated with the latest financial programs, financial tables, updates, and patches.

► Fixlets and Tasks

Fixlets and *Tasks* differ in how they are resolved. A Fixlet is triggered by a Relevance clause that detects a vulnerability. Then, an Action is started to remediate the vulnerability. The Fixlet automatically loses relevance and is thus no longer applicable on that specific Tivoli Endpoint Manager Agent. A Task includes one or more Action scripts that help you adjust settings or run maintenance tasks. The Task generally stays relevant after its Action script is run.

► Groups

Grouping your Tivoli Endpoint Manager Agents can simplify the maintenance of large networks. There are many ways to group computers, from a simple manual selection to more flexible automatic grouping. A simple grouping technique is to manually select members of a group from the listing in the Computers List Panel and add them to a group that you define. Membership is constant. Dynamic groups offer the capability to define membership based on the values of specific computer properties. You can, for example, group computers by IP address ranges, operating systems, applications, and thousands of other criteria by using Relevance expressions. Groups created this way have the benefit of automatic enrollment and expulsion. So, a computer that is repurposed to a different task or department automatically switches groups without operator intervention.

Change management enables an effective activity trail of changes to the IT systems, which is especially important when those changes are critical to the business. The financial accounting company already implemented a change management system. This change management system is currently used to create tickets for changes that are then approved by the receiver of the ticket. Using Tivoli Endpoint Manager with the change management system involves keeping the system external to the process as it is currently done at the financial accounting company. This approach means less overhead for the organization and a method that is familiar to all IT operators. Applying the patch starts when the patch is approved by the system administrator.

We can use the tools available in Tivoli Endpoint Manager to get a closer relationship between the two systems by using a Task to initiate the deployment of a patch. By prompting the approval operator to insert the number of the change management ticket, the approval can trigger the deployment of a patch. Involving the change control tool in the process helps ensure that the systems administrators of the servers are aware of the patch that is being deployed. The systems administrators can maintain a separation of duties between the operators that request the application of a patch and the administrators that approve the patch for the server for which they are responsible.

Choosing a design

All the tools mentioned act as a flexible way to assemble and implement a patching process. There are many ways to define a process, but each company must determine the best fit for it. In 5.4, “Functional requirements” on page 198, we defined the features that the financial accounting company wants in a process.

The financial accounting company decided to split the process for server and workstation patching due to the criticality of the services that run on these machines. So, we now look at these categories in more detail.

Server patching

The following steps show how to use tools within Tivoli Endpoint Manager to implement the patch management process for servers:

1. Patches are selected by operators from content that is made available within the Tivoli Endpoint Manager Console. Selected patches are cloned into a container for the chosen content, which is a *Custom Site* that is named descriptively, XYZ Windows Patches, for example.
2. Patches that appear in the Custom Site are tested by the central or virtual test team on systems that replicate the servers in the production environment.
3. The patch selector builds a Baseline to contain the relevant and tested Fixlet messages for the patch cycle for this period, December, for example. This Baseline contains Relevance to evaluate a computer setting value that contains the ticket number of the monthly change ticket. This evaluation happens on all clients within the deployed group.
4. System administrators receive the change ticket that mandates the requirement for the deployment of this patch and requests their approval by a certain date.
5. The system administrators are required to run a Task to approve the patches on their specific machines. This Task is available for that particular user only to perform, ensuring an appropriate separation of duties. The Task contains an Action script that changes the property value on the chosen machines to 1,

rather than 0. After the Task is propagated and the Action is taken to approve the patches, the Relevance for the patch Baseline evaluates to true and patching begins.

The diagram in Figure 7-2 displays the deployment process, including the tools within Tivoli Endpoint Manager that are used and the change management system that is included within the process.

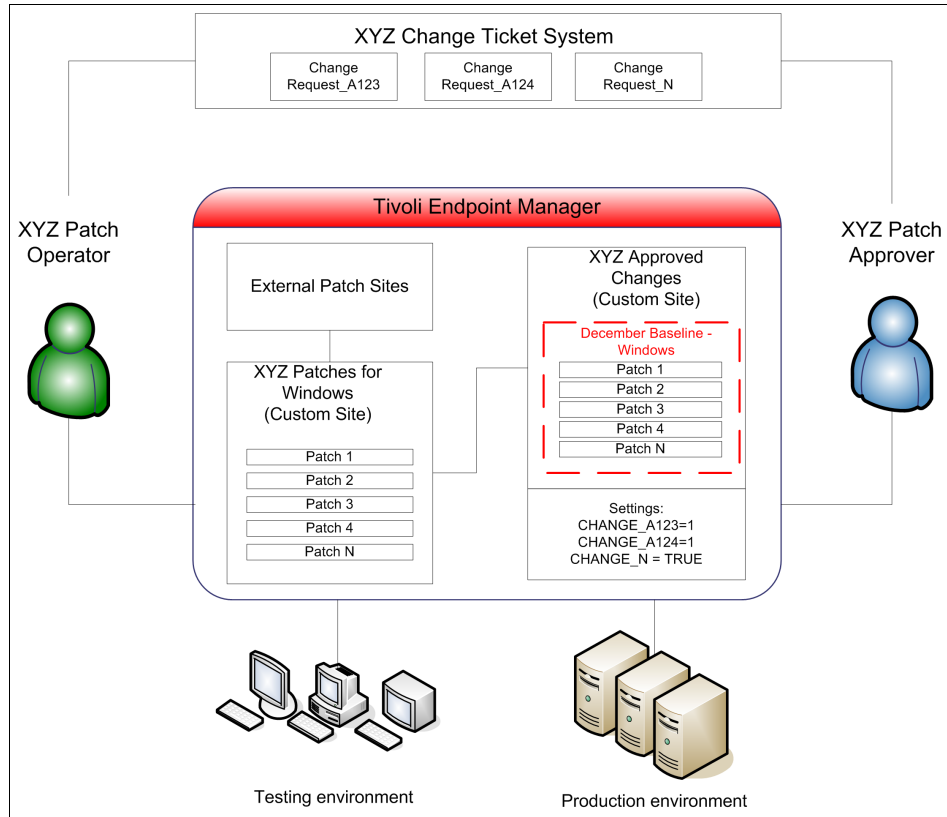


Figure 7-2 Server patching process

Workstation patching

The importance of the IT assets means that the action of patching operating systems and applications on the servers in particular is required to be a separate process. The financial accounting company understands the importance of their IT assets, in particular their production environment servers. This means that patching operating systems and applications on those servers is required to be a separately managed process. The financial accounting company classifies endpoints outside the production environment to be a *workstation*.

The following characteristics define the patch management process of the workstations:

- ▶ No real visibility of endpoints that are connected to the corporate network at any specific time.
- ▶ Users manage and are responsible for updates.
- ▶ Operating systems use the automatic update feature if updating.
- ▶ There is an unmanageable load on the help desk when Microsoft releases its patches on a monthly basis.
- ▶ Load on the network is not acceptable.
- ▶ There is no control of the endpoints outside of the corporate network.

IBM worked with the financial accounting company to propose the following use of Tivoli Endpoint Manager for solving the preceding issues. These issues affect the delivery of services offered to customers. These issues must be solved efficiently and quickly to update all of the workstation endpoints.

For the financial accounting company to define a corporate workstation patching policy, we must consider the method used in 7.1.4, “Patching rating and policy design” on page 242 to place all workstations into a risk class of “Advisable”. The implication is that the time for the organization to patch its workstations is a lower priority than the time for the organization to patch its servers. The financial accounting company also requested that, because of the existence of a dedicated support team, operating system and application patches are not required. However, the financial accounting company requested that a process is created for the deployment of the patches. This process can help solve the business problems of the network load that affects crucial services and the load on the help desk after repeatable vendor release dates. Another defining feature of the workstation patch management process is the approval step, which is moved from a specific member of the IT team to the user of the endpoint.

Figure 7-3 on page 255 shows the process that the financial accounting company chose.

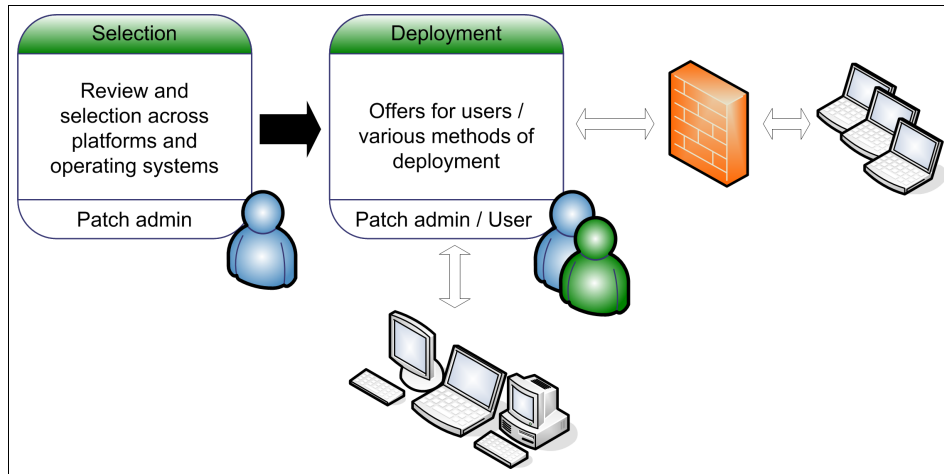


Figure 7-3 Workstation patching process

The Tivoli Endpoint Manager platform can immediately solve one of the issues that the financial accounting company faces. The current network load is experienced because each endpoint uses the native operating system update tool. An individual copy of each patch is downloaded to each endpoint. Even if only half of the workstations downloaded a patch of 10 MB, it still equates to a substantial load on the network.

The Tivoli Endpoint Manager platform uses *Relays* to reduce this load on the network. The Tivoli Endpoint Manager platform efficiently delivers the required patch to the next Relay, or groups of workstations in the local geography of the Relay. The Action script can also be defined for workstations to remotely download patches from the vendor source, which further distributes the load.

The following functional requirements are implemented for the workstation patch management solution:

- ▶ Ability to patch cross-platform operating systems.
- ▶ Capability to patch third-party applications, such as Java, Mozilla Firefox, and Adobe Acrobat Reader.
- ▶ The user has the ability with a self-service feature to approve the user's own patches. The user controls the required reboots.
- ▶ Capability to patch by using multiple languages.
- ▶ Ability to set maximum timescales for required patches if offered to the user.
- ▶ Ability to mandate the immediate installation of a patch if it is a high security severity.

- ▶ Capability to stagger the deployment of patches.

7.1.7 Patch management design conclusion

The financial accounting company determined its corporate Patching Policy to satisfy the business requirements. The functional requirements were determined, and the decision was made to differentiate server and workstation patching. The flexibility of the Tivoli Endpoint Manager platform allows the organization to maintain its current change ticket system while incorporating both tools into the new process. The organization also maintained a central point of control for endpoint management. The organization can enforce the Patching Policy at a local level, and it can report on a compliance stance in detail.

The financial accounting company chose to implement this server patch management process for the following reasons:

- ▶ It allows a separation of duties between the patching team and the patching approvers that maintains overall responsibility for the servers.
- ▶ Additional testing can be performed, if necessary.
- ▶ Changes require a change ticket, which requires approval. The change ticket includes an audit trail of who, what, and when the Action was taken.
- ▶ Local teams can choose to perform additional testing, if required.
- ▶ Compliance to the Patching Policy can be reported at a local country level.

In the following section, we describe the implementation of this Tivoli Endpoint Manager patch management solution in detail.

7.2 Implementation

This section documents how the financial accounting company implements the specific parts of the chosen Tivoli Endpoint Management patch management solution. We start with a discussion of how to perform basic operations by using the single management Console and how to organize the Console for multiple operator use. We then examine how we can use the common tools with the flexible Relevance language to implement the server and workstation patching processes. Where possible, we include useful implementation guidelines based on experience. We want to help you to more effectively think about the ways to use Tivoli Endpoint Manager to gain more visibility and control of your IT environment.

7.2.1 Introduction

So far, we defined a corporate patch policy for the financial accounting company and defined the patching process for servers and workstations. Collectively, the term workstation refers to endpoints that are not in use to serve a business service from the production environment. This workstation environment consists of a heterogeneous multiple platform, multiple operating system environment with a policy risk definition that is reflected in the patching policy.

We show how to use Tivoli Endpoint Manager to implement patching to address the functional requirements of the organization. We describe the following areas for implementing Tivoli Endpoint Manager:

- ▶ Console operation:
 - Creating operators
 - Creating patch groups
 - Creating Custom Sites
- ▶ Server patching
- ▶ Workstation patching:
 - Cross-platform patching
 - Patching third-party applications, such as Firefox and Adobe Acrobat
 - Microsoft Windows
 - Options
 - Timescales
 - Offline patching
 - Reboots

7.2.2 Console operation

Tivoli Endpoint Manager provides a central Console for managing all endpoints. From the Console, you can orchestrate changes and see the information that is sent back from the Agents. The financial accounting company decided that a small team of operators, which is the patching team, can use the Console to select and deploy patches to the endpoints.

Creating operators

The financial accounting company uses operators that authenticate locally on the Tivoli Endpoint Manager Server. We create two users for our two patch

management processes for the Windows Server operating system, XYZ_patch_windows and XYZ_server_approver.

In the Tivoli Endpoint Manager Console, we create these users by selecting **Tools** → **Create Operator**. In the dialog box, we enter the User name, enter the selected password, and click **OK**.

We specify the type of operator, We click the **Details** tab. For Master Operator, we select **No**, as shown in Figure 7-4.

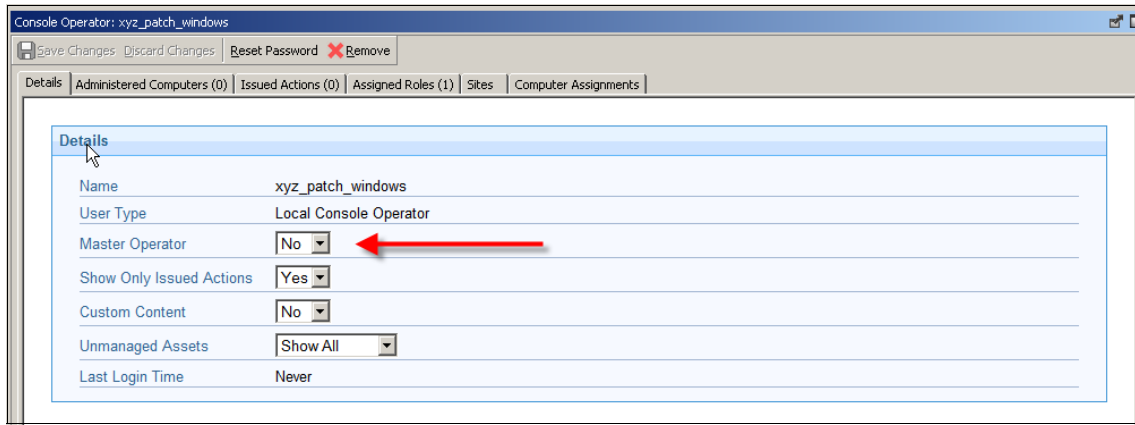


Figure 7-4 Type of operator

We assign the site content to the users. After selecting the user, we click the **Sites** tab and click **Assign Site**. The following site content must be assigned to the two users with the corresponding permissions. The XYZ_patch_windows operator is responsible for selecting patches for the Microsoft Windows operating system. This operator owns the Custom Site that is created as a container for the selected patches.

However, read-only access must be granted for the Change-Approval-Patch Site to maintain a separation of duties. The xyz_server_approver operator needs read access to all patch sites, because approvers can span all operating system types. Table 7-5 on page 259 outlines the users, their permissions, and the required site subscriptions for the financial accounting company.

Table 7-5 Users and permissions

Operator	Site subscriptions	Permissions
xyz_patch_windows	<ul style="list-style-type: none"> ▶ Patches for Windows ▶ Windows Server Patches (Custom Site) ▶ Change-Approval-Patch (Custom Site) 	<ul style="list-style-type: none"> ▶ Read Only ▶ Owner ▶ Read Only
xyz_server_approver	<ul style="list-style-type: none"> ▶ Patches for Windows ▶ Patches for AIX ▶ Patches for RHEL 5 ▶ Change-Approval-Patch (Custom Site) 	<ul style="list-style-type: none"> ▶ Read Only ▶ Read Only ▶ Read Only ▶ Read Only

Figure 7-5 shows the sites to which operator XYZ_patch_windows is subscribed.

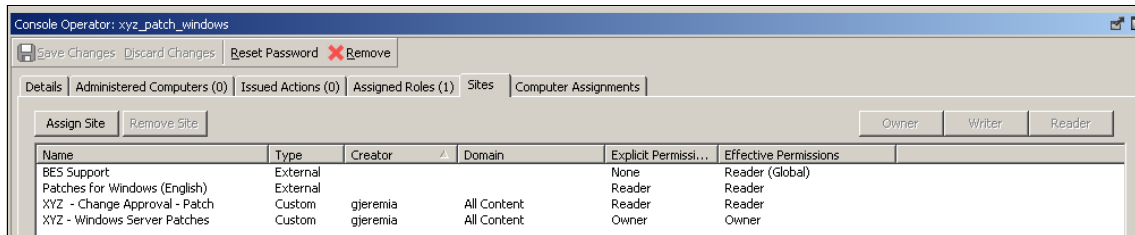


Figure 7-5 Site subscriptions for operator XYZ_patch_windows

Information: Tivoli Endpoint Manager can use the central directory of users, which includes Microsoft Active Directory, and other generic Lightweight Directory Access Protocol (LDAP) servers. This feature is new to Tivoli Endpoint Manager Version 8.2. To use this directory option for user authentication, use the Console and click **LDAP Domains** → **Add LDAP Domain**.

Multiple security levels: You can further enhance security by using a single sign-on (SSO) solution. For instance, you can use a unique radio frequency identification (RFID) access badge, for example, to authenticate an operator in addition to the LDAP directory of the organization.

Creating roles

We assign these new operators to a role within Tivoli Endpoint Manager. Roles can be used as an easy way to automatically assign sites to either stand-alone Console operators or operators that authenticate against an organizational LDAP directory.

Update: Roles are a new feature in Tivoli Endpoint Manager Version 8.2.

We create two new roles: one role for the Patch Admin (PatchAdmin) and one role for the Patch Approver (PatchApprover). Figure 7-2 on page 253 shows these roles. In the Console, we click **Tools** → **Create Role**. We are prompted to enter the name of each role that we create. We enter the name for each role. We enter a description of the role in the Description field and assign the appropriate users to each role. Figure 7-6 displays the details of the PatchAdmin role.

The screenshot displays the Tivoli Endpoint Manager Console interface. On the left, a navigation pane shows various content categories like Roles, Filelets and Tasks, LDAP Domains, Baselines, Analyses, Actions, Dashboards, Wizards, Custom Content, Custom Filters, Computers, Computer Groups, Unmanaged Assets, Operators, and Sites. The main area shows a table of roles with columns for Name, Master Operator, Unmanaged Assets, Custom Content, Sites, Computers, and Operators. The PatchAdmin role is highlighted with a red box. Below the table, a detailed view for the PatchAdmin role is shown, with fields for Name, Description, Master Operator, Custom Content, and Unmanaged Assets. Red arrows point to the Name and Description fields.

Name	Master Operator	Unmanaged As...	Custom Content	Sites	Computers	Operators
TEM Infra Admins	No	Show All	Yes	24	0	0
PatchAdmin	No	Show None	Yes	2	0	2
PatchApprover	No	Show None	No	4	0	1

Role: PatchAdmin

Details

Name: PatchAdmin

Description: The Patch Admin role is responsible for selecting patches from the external site. This content (patches in this case) will then be allocated for testing on the XYZ Financial accounting servers before production.

Master Operator: No

Custom Content: Yes

Unmanaged Assets: Show None

Figure 7-6 Creating a role by using Tivoli Endpoint Manager

Creating patch groups

Computer groups are containers of Tivoli Endpoint Manager Agents. They can help simplify maintenance in large networks. We create groups for the financial accounting company for several reasons:

- ▶ To create logical containers of target servers based on the classification of risk for the organization. For example, we can target all Urgent class servers for patching first before we patch the Advisable group.
- ▶ To stagger the deployment of patches to many workstation endpoints, for example, patching a group of employees that volunteered to be the first systems to be patched.
- ▶ To help us to define which group of servers to use for testing and which group of servers to use for production.

You can group computers in many ways, from simple manual selection to more flexible automatic grouping. A simple grouping technique is to manually select members of a group from the listing in the Computers List Panel, and add them to a group that you define. Automatic groups can automatically place computers into the group based on computer property values.

It is important to remember that the financial accounting company server patching process targets the property of the endpoint. When a Baseline is created, we target the property of the endpoint, commonly the operating system. Therefore, we do not require many groups to split up how we target machines. The financial accounting company uses groups to define the separate risk classes for its different servers. A single major group is used for workstations to help logically separate them. The financial accounting company uses these groups:

- ▶ Servers: Production Urgent Class
- ▶ Servers: Production Essential Class
- ▶ Servers: Production Advisable Class
- ▶ Servers: Test
- ▶ Workstations

We first create our server group and define this group as manual. We use the Console and select **Tools** → **Create New Manual Computer Group**. Figure 7-7 on page 262 shows the new computer groups that we created.

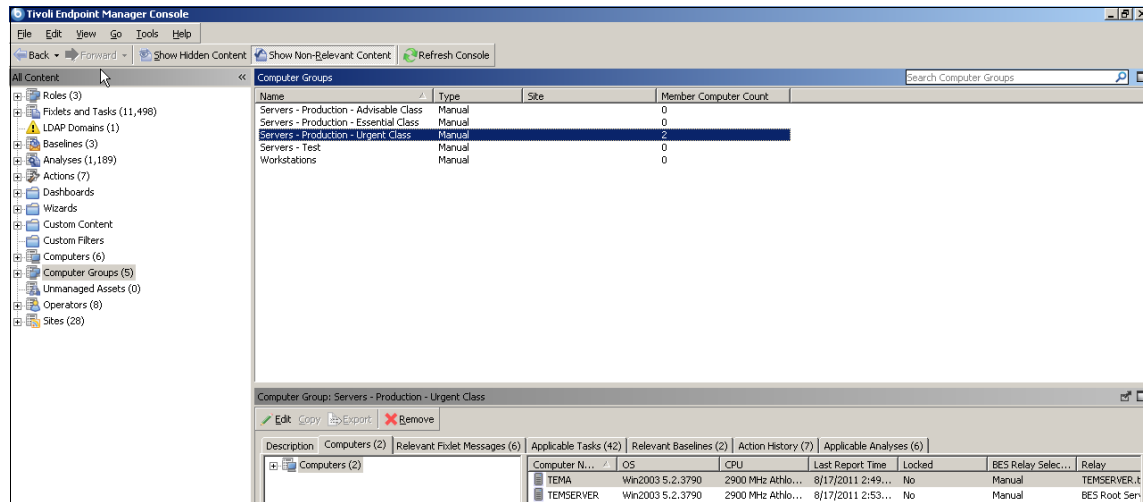


Figure 7-7 The groups are used to logically separate servers and workstations

Creating Custom Sites

Based on the financial accounting company patching process for server and workstation endpoints, a patch administrator is responsible for selecting the patches for testing. This selection is largely based on the risk that the patch represents to the security of the endpoints. We previously subscribed the operator XYZ_patch_admin to a Site known as Patches for Windows. This Site contains all of the Fixlet messages to patch Microsoft Windows systems. Because we test these patches before deployment, we must create several Custom Sites to act as containers for the relevant Fixlet messages for the testers. We then create another Custom Site for the XYZ_patch_approver to deploy within the Baseline.

We do not need many Sites for this process; in fact, we do not have to use any Sites at all. However, logically separating the patches after they are tested is a helpful way to track operations. We use the Console. We click **Tools** → **Create Custom Site** and enter the name of the site. We insert a description for this Site. We also need to subscribe some machines to the content within this Site. Because this Site is targeted for servers that run the Microsoft Windows operating system, we define a computer property that specifies whether a computer system is subscribed to this site.

We select the **Computer Subscriptions** tab and select **Computers which match the condition below** → **OS** → **Contains** → **Win2**. This action subscribes all operating systems from Windows Server 2003 and Windows Server 2008. We can customize our system property in more detail here if we want. Figure 7-8 on page 263 shows this action.

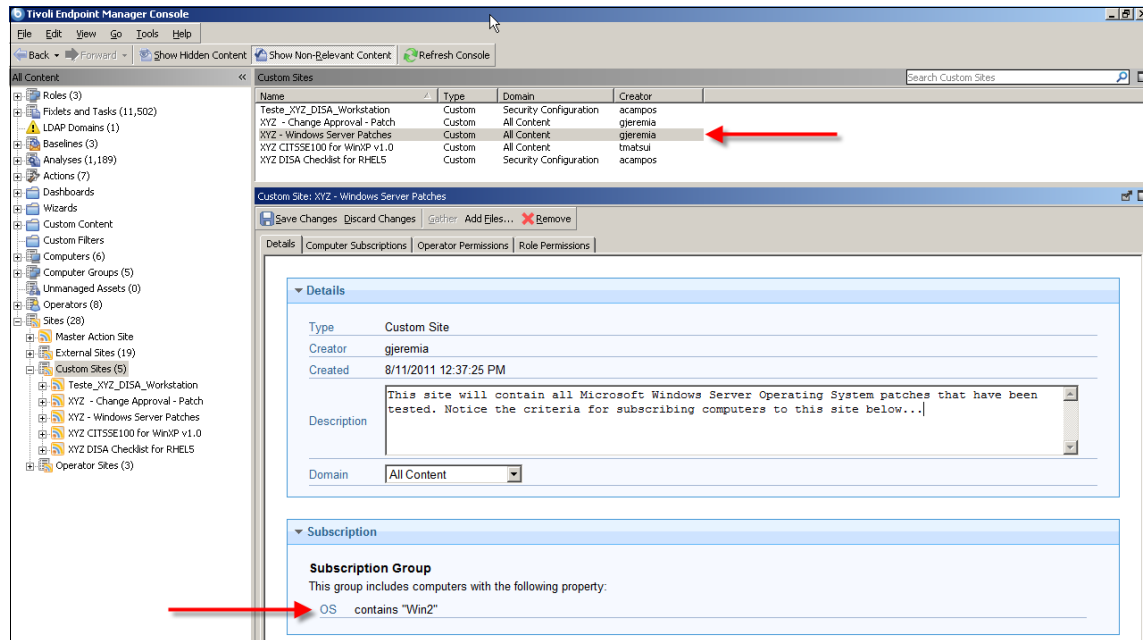


Figure 7-8 After creating the Custom Site

Figure 7-9 shows how we define the computers that are subscribed to this Custom Site.

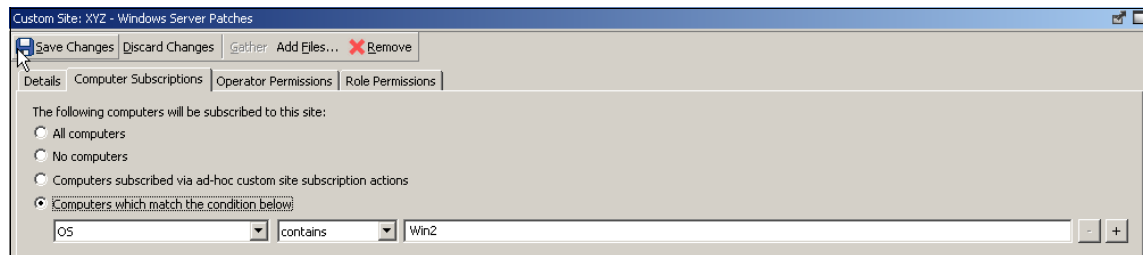


Figure 7-9 Which computers are subscribed

Important: At this stage, when creating a Custom Site, all of the tools are available to us for managing content within this Site. Baselines, Tasks, Fixlets, Analysis, Groups, Actions, and more are all available, as shown in Figure 7-10 on page 264.

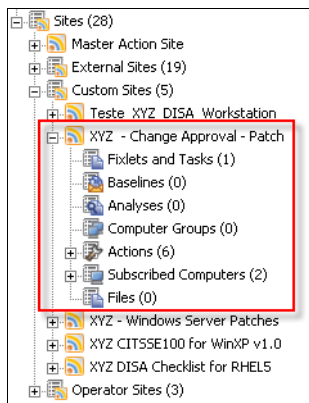


Figure 7-10 Custom Fixlet Site

We also must set the operator permissions to enforce the separation of duties between the patch administrator and the server approver. Figure 7-11 shows the permissions for the XYZ_patch_windows operator that owns the Site. The XYZ_patch_windows operator can modify the content, but the patch approver has no permissions.

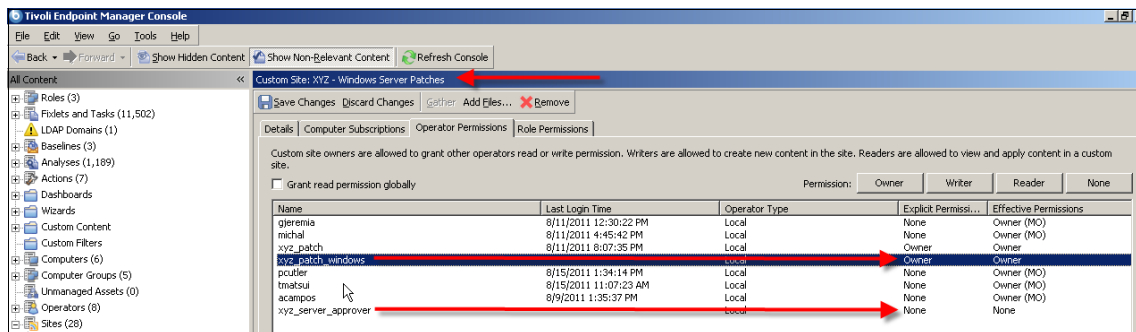


Figure 7-11 Examining the operator permissions for the XYZ_patch_windows server patches Custom Site

We can see that the correct permissions are set by logging in to the Console as the xyz_patch_approval operator. Figure 7-12 on page 265 shows the approval Task for server patch approvers that work with the Custom Site.

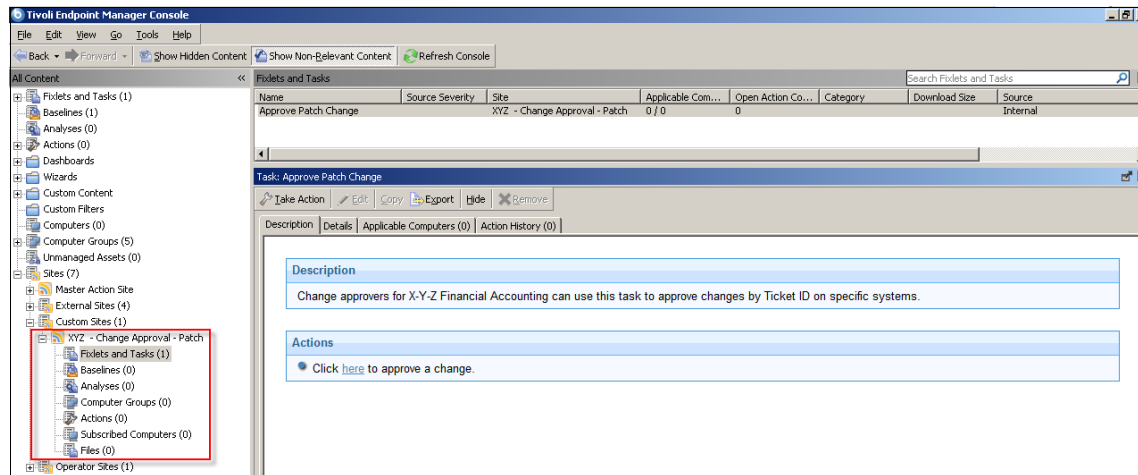


Figure 7-12 Correct console permissions for enforcing the separation of duties

Creating a custom approval Task

In “Choosing a design” on page 252, we describe the process that the financial accounting company uses for patching its servers by using Tivoli Endpoint Manager. The method involves a separation of duties between the patch administrator, responsible for patch selection, and the server approver, responsible for taking an action to approve the patch. This process incorporates the current change ticketing system, allowing a full audit trail of activity for the production server environment. We now examine the implementation.

In the first step, we create a custom Task for the approval process. The server approver receives a patching ticket ID from the patch administrator. The approver then takes an Action to approve this patch if the approver wants to deploy the patch to the clients. We describe these procedures in more detail in 7.2.3, “Windows server patching” on page 268.

This Task creates a registry key with a ticket ID value of “1” when the approver takes an Action from this Task, selecting the ticket ID as a parameter. This Action is important technically, because patching a Baseline evaluates the relevance by this registry key for each Agent. If the Baseline receives a value of “1” from this registry key, Relevance becomes true. This Agent already approved the Action and now accepts this patch.

We click the **All Content** domain on the lower-left side of the Tivoli Endpoint Manager Console. In the All Content content tree, we select **Sites** → **Custom Sites** → **XYZ - Change Approval - Patch** → **Fixlets and Tasks**, which we create in 7.2.3, “Windows server patching” on page 268. We choose **Create New Fixlet** on the right-click menu in the **Fixlets and Tasks** field. We change the

Fixlet name to ApprovePatchChange and describe the purpose of this Fixlet (Figure 7-13).

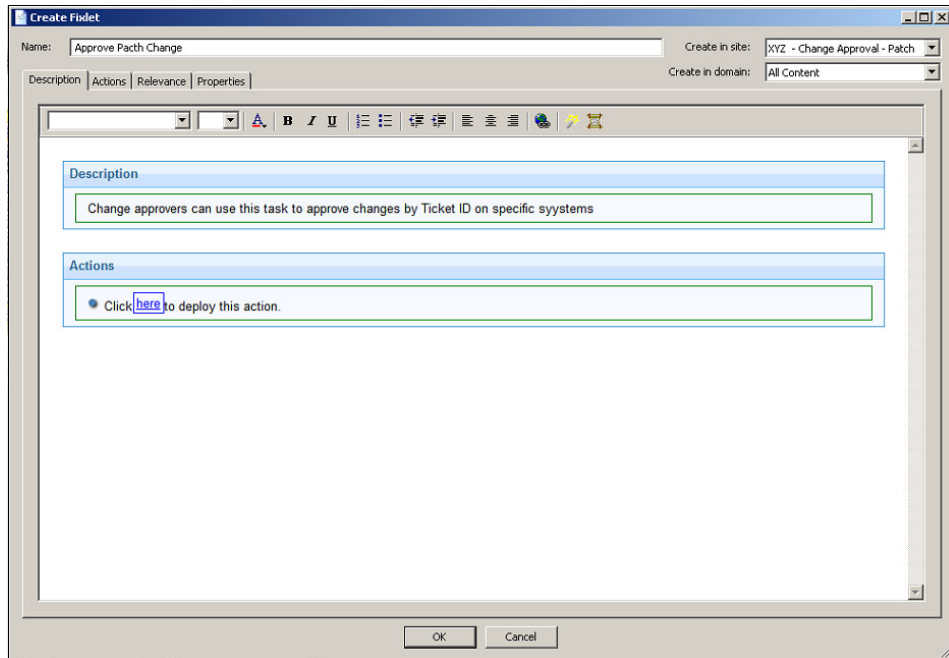


Figure 7-13 Create an approval Task

Next, we click the **Actions** tab. We add an Action Script that creates a “CHANGE_PATCH_TICKETNUM” registry key from the ticket ID as an Action parameter and sets the value to “1”, as shown in Figure 7-14.

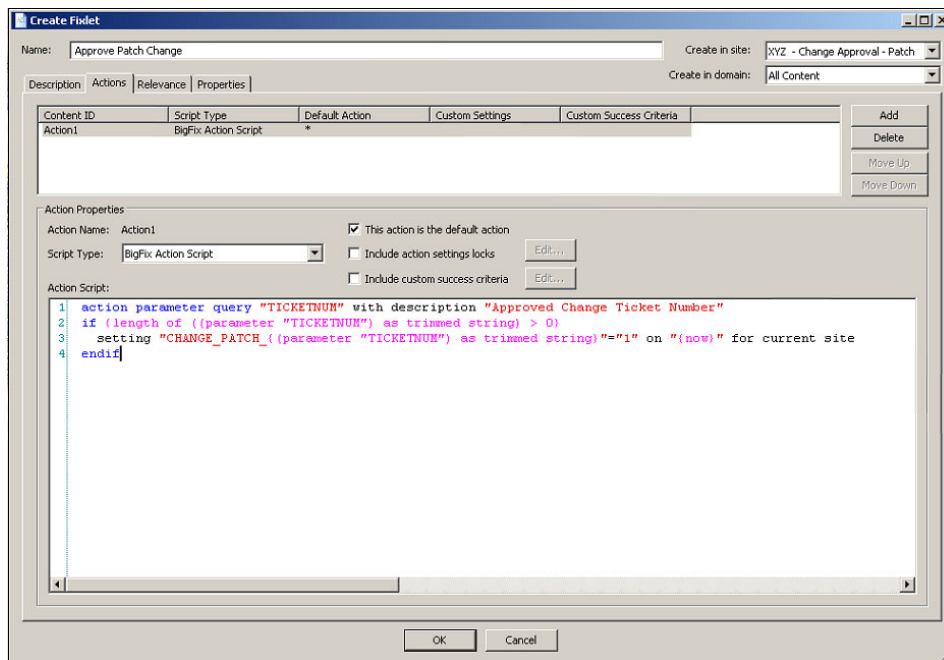


Figure 7-14 Example for Action Script that creates a key with a ticket ID

We click the **Relevance** tab. We choose **Computers which match all of the relevance clauses below** with **1. true** and select **OK** to complete.

Creating a computer property

It is useful for operators to create an additional computer property that enables them to acknowledge approved endpoints. On the Tivoli Endpoint Manager Console, we define the endpoint information that must be displayed as a property by the Relevance Action Script. When an approver operator takes an approval Action, the deployed endpoints receive the registry key related to the ticket ID, as mentioned in “Creating a custom approval Task” on page 265. Thus, the Relevance must be written to find endpoints that have the value “1” in this registry key. We look at the detailed procedure now.

We open the **All Content** domain on the lower-left side of the window. We click the **Computers** field in the content tree. On the computer properties items above the list of endpoints, we right-click and see Column Picker. Then, we use the

Manage properties window, select **Add New**, and enter the Name and the Relevance.

If you want to manage properties with directories, you can select the appropriate directories with ::, such as XYZ::Asset::ApprovedPatches, which goes to XYZ\Asset\ApprovedPatches.

We see an example of Relevance checking the registry keys on endpoints in Figure 7-15. To submit, we click **OK**.

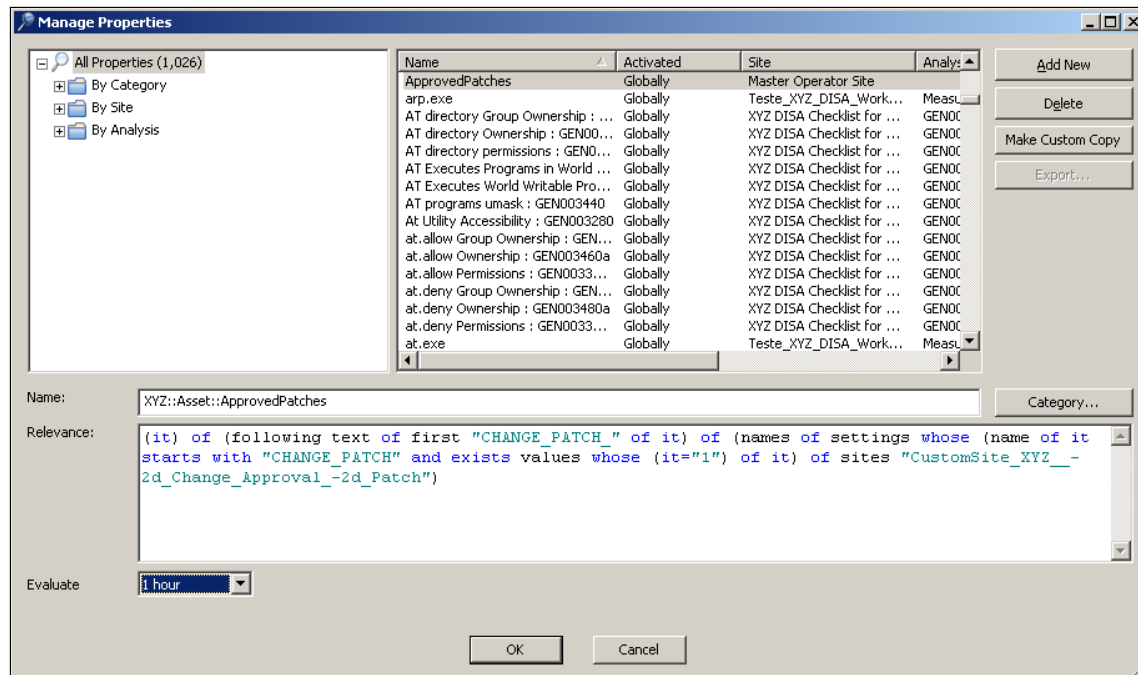


Figure 7-15 Relevance for computer property

7.2.3 Windows server patching

In this section, we look at how to implement server and workstation patching with Tivoli Endpoint Manager. We investigate how to use custom patching sites, patching operators, approval Fixlets, and how to patch server and workstation groups that we created in preparation for patching. In this approach, a selector operator provides information about the available patches for a particular ticket, and an approver operator approves that ticket. After the approval step, the selector operator first creates a Baseline, which consists of individual patches, and then uses an Action to deploy the Baseline.

Constraints

In the constraints, you can configure the Baseline to be deployed according to the change control guidelines for your Windows Server machines. Configuring specific controls allows the operator to ensure that any disruption to production is minimized. For example, specific days and conditions for the deployment, such as midnights and weekends, can be set to trigger the deployment of a patch.

Behavior

Baselines can be set for a number of retries if, for whatever reason, the Baseline does not successfully deploy. Due to the constantly changing operating system environments, it can be useful to reapply Baselines if the content becomes relevant again at any stage. Reapplying Actions ensures that patches automatically maintain their status, helping the organization to maintain its secure server posture.

Use the Users tab in the Console to define whether a user is required to be present and whether we show the user that the Action is occurring. You can also refine this behavior further by requesting that Actions are taken only when a specific user logs on. The financial accounting company wants to provide an option to the systems operators at each monthly Baseline for low severity patches. Use the Messages tab to inform the users of the Action taking place and to provide them with additional options for deployment.

By using the newly created Baseline within the Custom Site, the operators can now deploy this action group to Windows Server machines by selecting the Baseline and clicking **Click here to deploy this action group**. The operator is presented with a selection of relevant computers to which to deploy the Baseline, and the preset profile created earlier can be used to deploy.

Reboots during patching

For patches that require a reboot, the Agent contains a software component known as the *action manager* to cache all activity that must be continued during a patch. It is important to remember that if the system operators can decide when to reboot their servers, the patch that is being applied might not reply to the Tivoli Endpoint Manager Server as *patched* until the endpoint is rebooted. For this reason, the financial accounting company decided to force a reboot of all servers immediately after patching during the night. Figure 7-16 on page 270 depicts how the Post-Action tab needs to be configured for this forced reboot to happen.

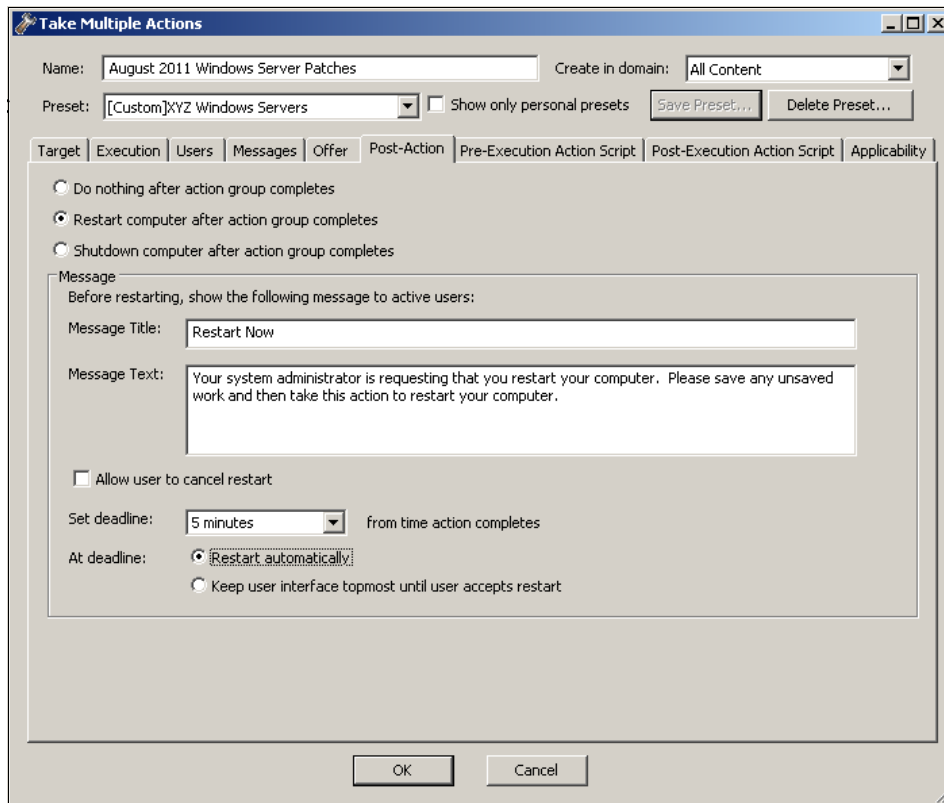


Figure 7-16 Reboot servers immediately during the night

Procedure

We now look at the steps that we need to configure for the server patching:

1. We log on to the Tivoli Endpoint Manager Console as the selector operator. We create a patching ticket with an ID and inform the approver operator. This procedure is not performed by using Tivoli Endpoint Manager.
2. Next, we clone the deployed patch Fixlets to the Custom Site XYZ - Windows Server Patches. We open the **Patch Management** domain on the lower-left side of the Tivoli Endpoint Manager Console. We navigate to **Fixlets and Tasks** and click **OS Vendors** → **Microsoft Windows** in the content tree. We choose all appropriate Fixlets and select **Create Custom Copy** on the right-click menu.
3. In the Copy Content dialog, the Custom Site is listed in the Create in site list box, as shown in Figure 7-17 on page 271. We select **XYZ - Windows Server Patches** and select **OK**.

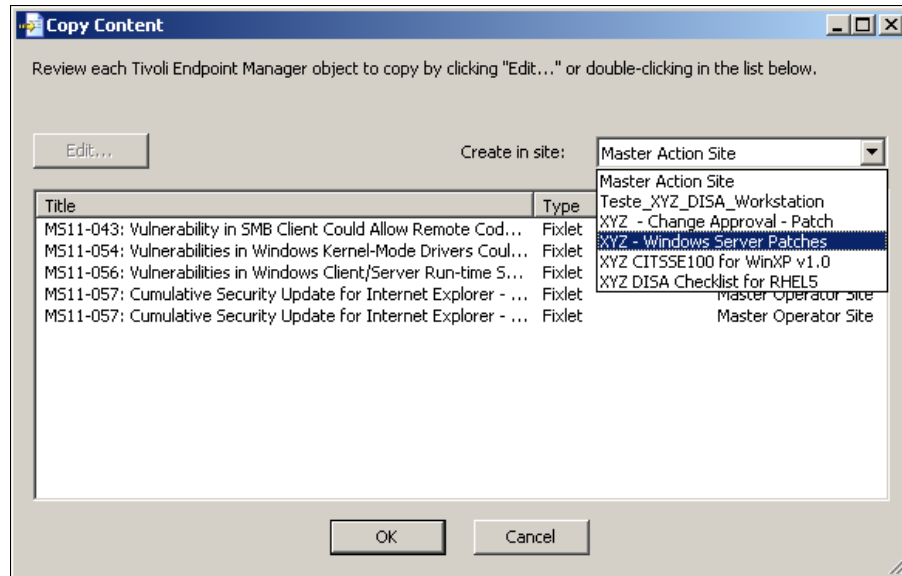


Figure 7-17 Clone patch Fixlet to Custom Site

- To complete the cloning, we open the **All Content** domain. We click **Custom Sites** → **Fixlets and Tasks** → **XYZ - Windows Server Patches**. We build a Baseline that consists of patches cloned in step 2 and insert the ticket ID in the Relevance as "Required Approva1".

5. We choose **Create New Baseline** on the right-click menu with the appropriate Fixlets in the **Fixlet and Tasks** field. We select **XYZ - Windows Server Patches** for the Create in site list box. We provide a Name, for example, August 2011 Windows Server Patches, and a Description. We navigate to the **Relevance** tab and choose **Computers which match the condition below** with **XYZ::Asset::ApprovedPatches equals 1111** (*ticket ID number*), as shown in Figure 7-18. We click **OK** to complete this step.

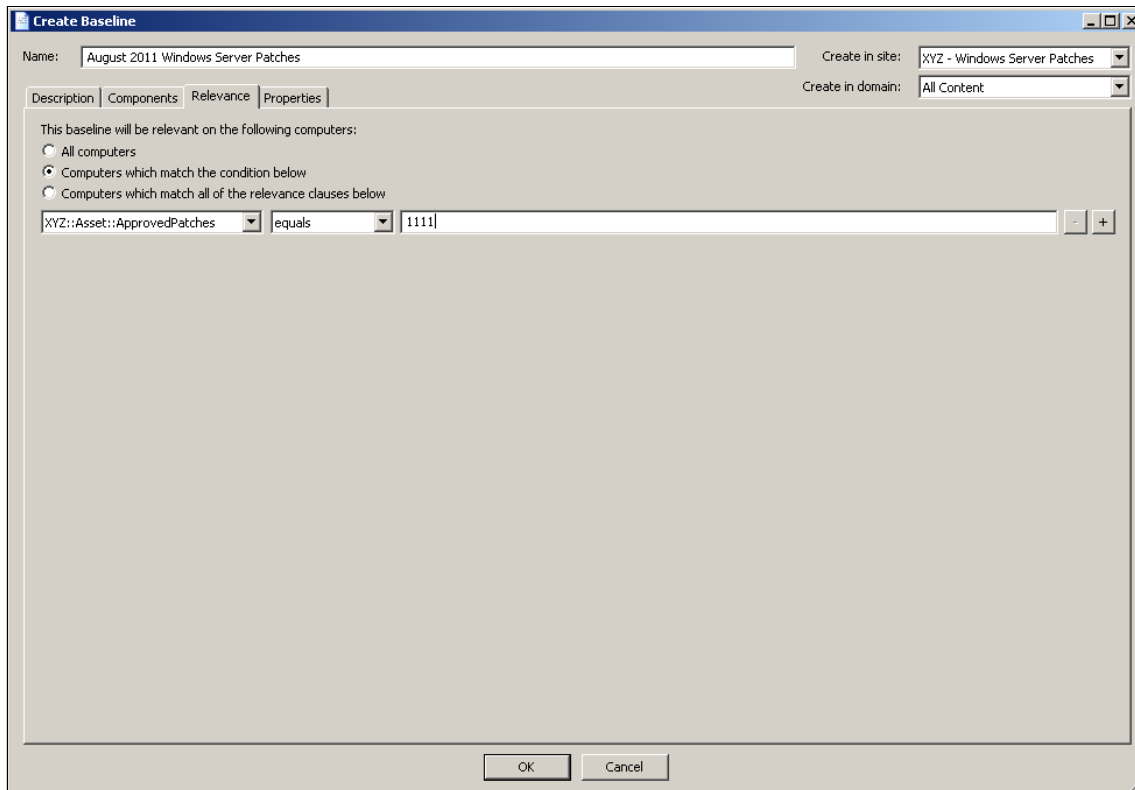


Figure 7-18 Insert ticket ID Relevance into Baseline

6. We log on to the Tivoli Endpoint Manager Console as the approver operator. This approver operator received a patching ticket ID in the change ticket system. The approver operator can now approve one or more appropriate computer groups for this ticket.
7. We deploy an Action for the Approve Patch Change Task to identify the approved change ticket number. We navigate to the **All Content** domain and open **Custom Sites** → **Fixlets and Tasks** → **XYZ - Change Approval - Patch**.

8. We click **Take Action** for the Approve Patch Change Task, and we set the ticket ID as an **Action Parameter**, as shown in Figure 7-19.

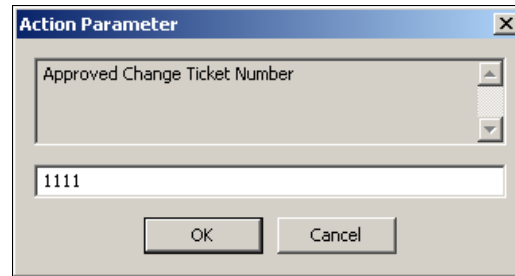


Figure 7-19 An Action Parameter for the approval Task

9. We choose our target computer or groups (**Essential Class Windows 2003 servers** in our example) and select **OK**, as shown in Figure 7-20.

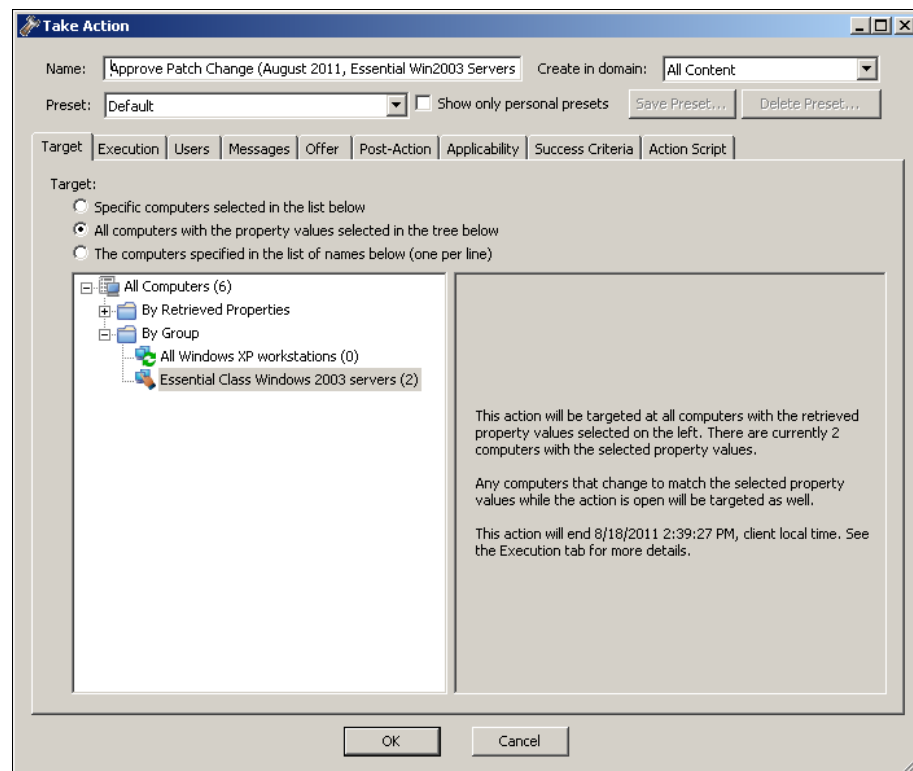


Figure 7-20 Take an approval Action to target computers

10. We log on to the Tivoli Endpoint Manager Console as the selector operator. We take an Action of the Baseline to deploy the patches that target *All computers* evaluated as relevant by the Action Script.
11. We open the **All Content** domain and select the **Baselines** field of the **XYZ - Windows Server Patches** in the **Custom Sites** content tree. We click **Take Action** for the *August 2011 Windows Server Patches Baseline*. This action presents the Take Multiple Actions dialog shown in Figure 7-21. We select **All Computers with the property values selected in the tree below**. The Relevance Action Script inserted in the Baseline now selects the appropriate deployment targets, as shown in Figure 7-21.

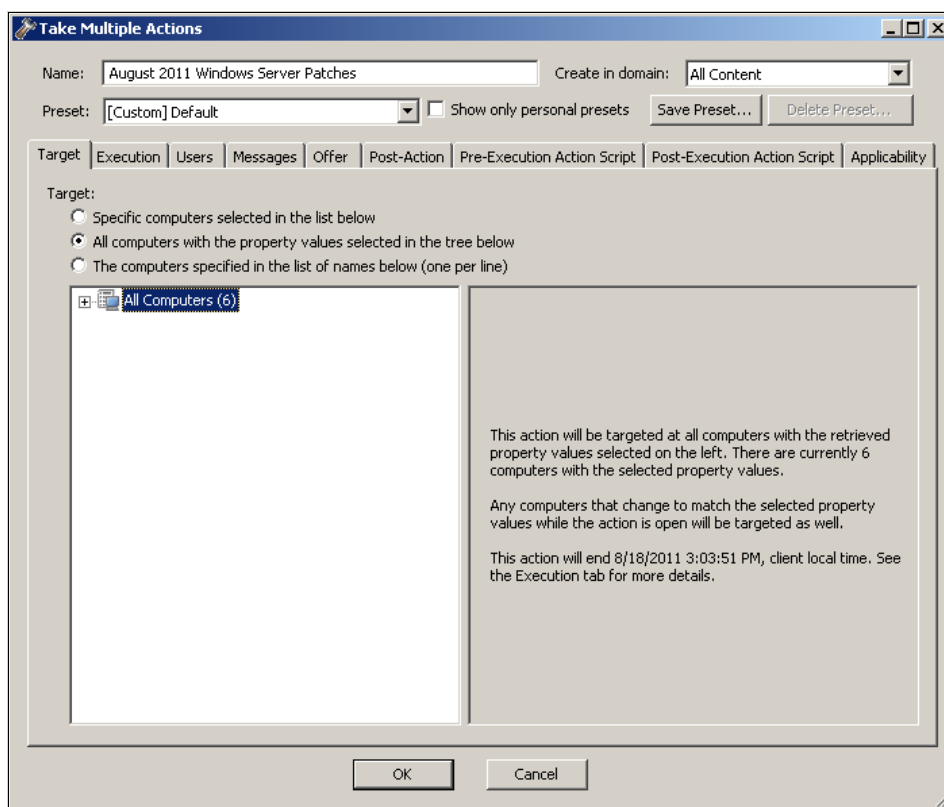


Figure 7-21 Select target computers

12. We choose an appropriate **Preset** for server patching (we create and choose a preset) or navigate through the other tabs to define options for this Action. We click **OK** to begin the deployment.

7.2.4 Workstation patching

In “Choosing a design” on page 252, we explained the patching process. For the first stage of this project, the financial accounting company wants to gain visibility and control of its endpoints that run Microsoft Windows. The Microsoft Windows operating system is the most prevalent operating system that is deployed among the employees. However, the organization supports a heterogeneous endpoint environment that includes systems that run the Apple OSX operating system, Ubuntu, Red Hat, and Centos Linux. The workstation patching process is implemented on a monthly cycle. The responsibility for the workstation patching belongs to the patch administrator. Patch approval for informational patches and feature enhancements remains with the user for a limited time to give employees some autonomy. For this phase of the project, the financial accounting company focuses on patching the Windows operating system and some of the third-party applications.

Operating system patching

Tivoli Endpoint Manager displays vendor-released patch content that is combined with an Action script in Fixlet messages. All of this information is displayed in the Console in the appropriate Site. The workstation patching process defined that the operator `XYZ_Patch_windows` selects the content from the external `Patches for Windows` Site and creates a Baseline within a new Custom Site labeled `XYZ - Windows Workstation Patch`.

Figure 7-22 on page 276 displays the content within the `Patches for Windows` Site. Each Fixlet message shown in the Fixlets and Tasks window has a corresponding lower window that displays a description of the actions to be taken and often links to vendor release information. The intention is to inform the operator as much as possible, enabling the operator to make informed deployment decisions.

The screenshot displays the Tivoli Endpoint Manager Console interface. On the left, a navigation tree shows various management areas, with 'Patches and Tasks (1,286)' selected. The main area shows a table of patches with columns for Name, Source Severity, Site, Applicable Com..., Open Action Co..., and Category. A red arrow points to the 'Fixlet Message' column header. Below the table, a detailed view for patch MS11-044 is shown, with a red arrow pointing to the 'Description' tab. The description text reads: 'Microsoft has released a security update that resolves a publicly disclosed vulnerability in Microsoft .NET Framework. The vulnerability could allow remote code execution on a client system if a user views a specially crafted Web page using a Web browser that can run XML Browser Applications (XBAPs). Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights. The vulnerability could also allow remote code execution on a server system running IIS, if that server allows processing ASP.NET pages and an attacker succeeds in uploading a specially crafted ASP.NET page to that server and then executes the page, as could be the case in a Web hosting scenario. This vulnerability could also be used by Windows .NET applications to bypass Code Access Security (CAS) restrictions. After downloading and installing this update, affected computers will no longer be susceptible to this vulnerability. Note: Affected computers may report back as 'Pending Restart' once the update has run successfully, but will not report back their final status until the computer has been restarted. Important Note: There are known issues associated with the installation of this update. See the Known Issues section of the security bulletin for more information. Note: Microsoft has announced that this update may be included in a future service pack or update rollout. Note: This security update is also referenced under KB2518863. File Size: 6.02 MB. Actions: Click here to initiate the deployment process. Click here to view Microsoft Security Bulletin MS11-044.'

Figure 7-22 Windows patch content within the Patches for Windows site

A Baseline is created each month to deploy the patches. To create a Baseline, the financial accounting company operators need to highlight the patches that they want to insert. Then, they right-click an individual patch and select **Add to new Baseline**. Then, the operator can insert a description for the new Baseline. It is important to create this Baseline in the correct Site and content domain. The Create in site selection list must be set to **XYZ - Patches for Windows Workstations** so that the Baseline can be deployed from that site. Figure 7-23 shows this initial window.

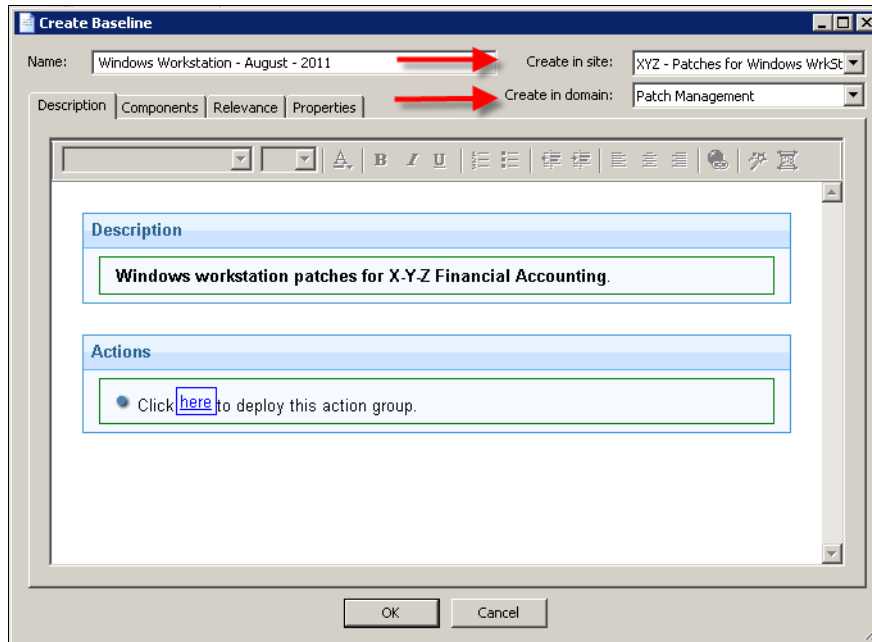


Figure 7-23 Initial Baseline creation window

The Components tab displays the groups of Fixlet messages that are included in the Baseline to be deployed to the targets. Additional Fixlets always can be added to the same Baseline at a later stage, if needed. On this window, the operator can select the default Actions for each Fixlet. It is important that **Use custom action settings for this baseline** is selected, as depicted in Figure 7-24. When finished, click **OK**.

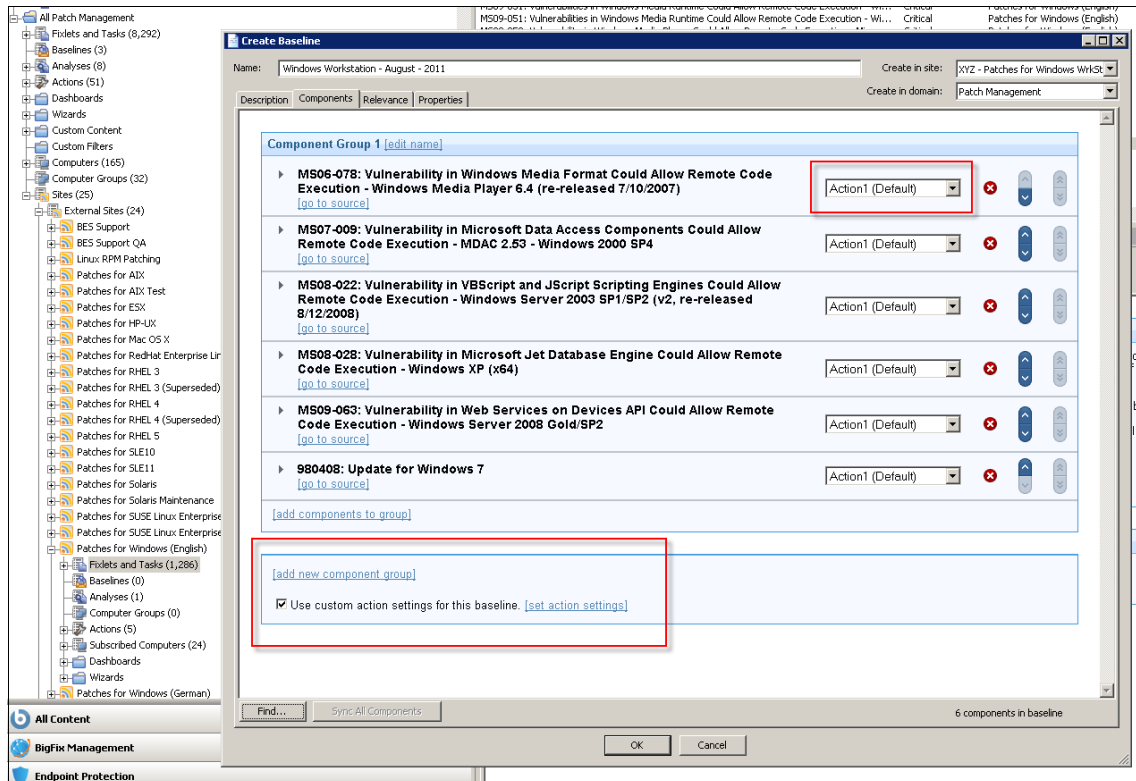


Figure 7-24 Baseline Action settings

Constraints

The Baseline can be set to deploy either according to change control windows or at a convenient time for the users. Selecting an **Ends On** date and time allows an operator to set a patch deployment. This option ensures that the patch deployment either finishes or returns false, before the employees return to work to perform their daily business functions. Therefore, the patch deployment does not affect their workload. Specific days and conditions can also be set to trigger the deployment of a patch. Select **Starts On** → **5pm Friday** and **Ends On** → **9am Monday** to run only on **Fri, Sun, Mon**.

Behavior

Baselines can be set for a number of retries if, for whatever reason, the Baseline does not successfully deploy. Retries are important due to the constantly changing operating system environments. This capability can be useful to reapply Baselines if the content becomes relevant at any stage. For instance, a user manually installs software that requires a component of a previous Windows version or patch level. Conflicts can occur and the system might become vulnerable again after the third-party software redeploys an outdated component. Reapplying Actions can ensure that patches automatically maintain their status, which means that the organization can maintain its secure workstation posture. To achieve this task, check **Reapply this action** → **while relevant, waiting 30 minutes between reapplications**. Also, choose to stagger the action start time over 5-minute intervals to reduce the load on the financial accounting company network. All those Action settings are depicted in Figure 7-25.

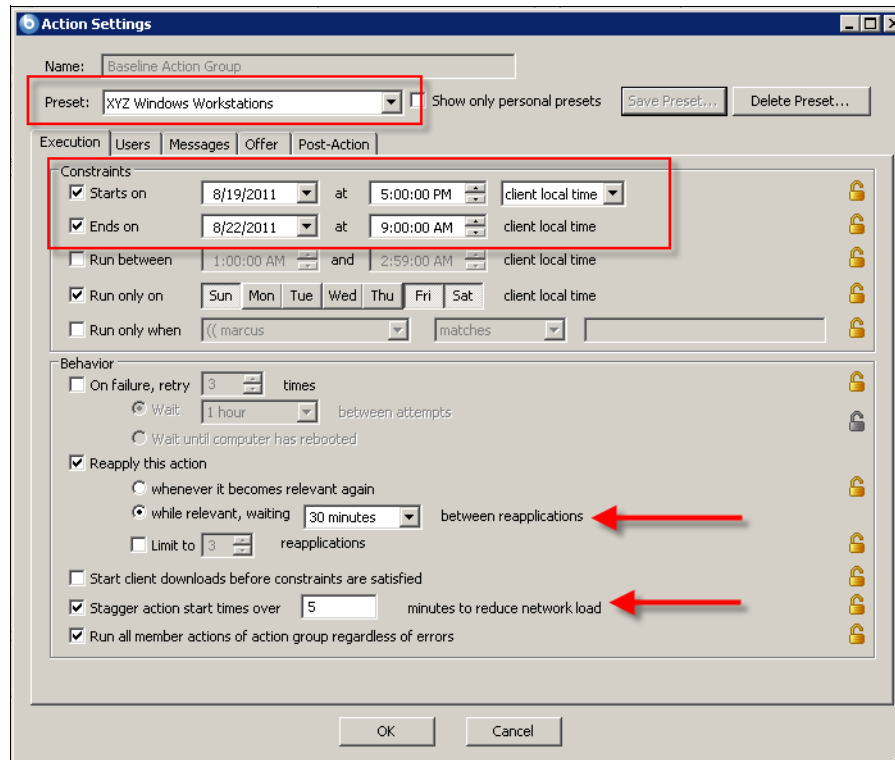


Figure 7-25 Baseline Action Settings

With the Users tab, you can define whether a user is required to be present and whether you show the user the Action that is taking place. You can also refine this information further by requesting that actions take place only when a specific user logs on. The financial accounting company wants to provide options to the users at each monthly Baseline, which is typically for low severity patches.

With the Messages tab, shown in Figure 7-26, you can inform the users of the Action that is taking place and give them options for deployment. It might not always be convenient to patch a system at the current time.

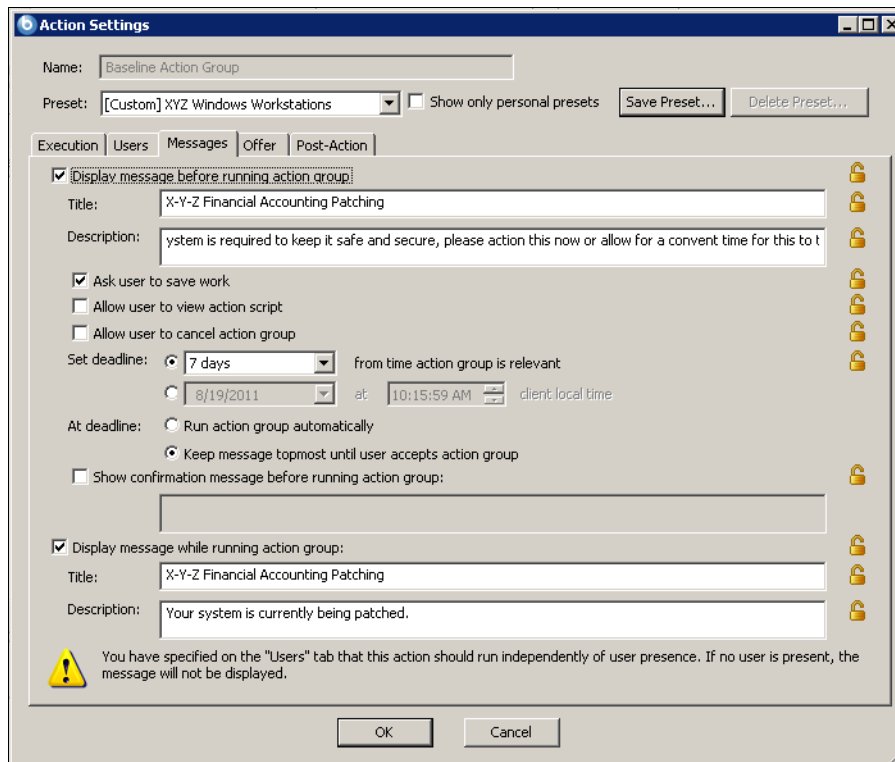


Figure 7-26 Defining the control that a user has in the patch deployment

Users can also be offered additional Actions for software updates, which is achieved by using the Offer tab (Figure 7-27 on page 281).

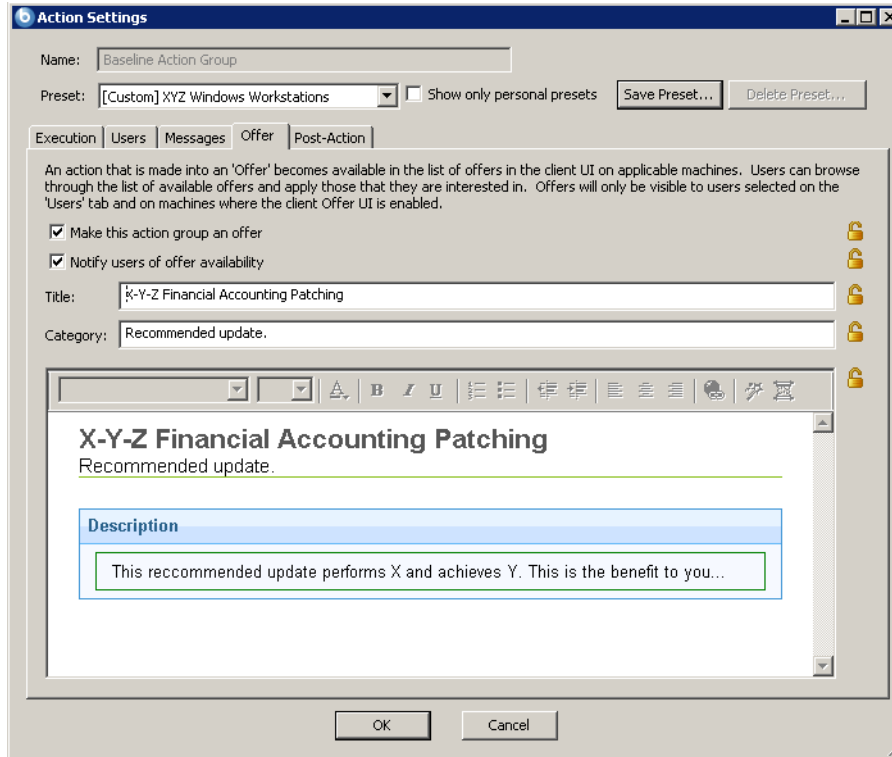


Figure 7-27 Allowing options for users when deploying patches

Using the newly created Baseline within the Custom Site, operators can now deploy this patch to workstation machines within the organization. This task is completed by selecting the Baseline and clicking **Click here to deploy this action group**.

Patching third-party applications

The financial accounting company wants to include its Mozilla Firefox browser, Adobe Acrobat Reader, and the Java Runtime Environment in its patching process with Tivoli Endpoint Manager. These products are updated frequently. Also, these products are required to perform critical business functions every day. The third-party application patching at this stage exists for Windows operating systems, because the financial accounting company wanted to roll out patch control for this operating system in the first phase.

Patching these applications can be performed in the same way that patches are applied to operating systems. If you navigate to the Console, you can see the content for the third-party applications by selecting **Patch Domain** → **Application Vendors** → **Adobe** or **Oracle** or **Mozilla**.

Baselines can be used to deploy updates to groups of machines. Deployment presets can also be used as a quick way to configure the Action deployment. Figure 7-28 displays the patching domain with the list of supported third-party application vendors that contain Fixlet content.

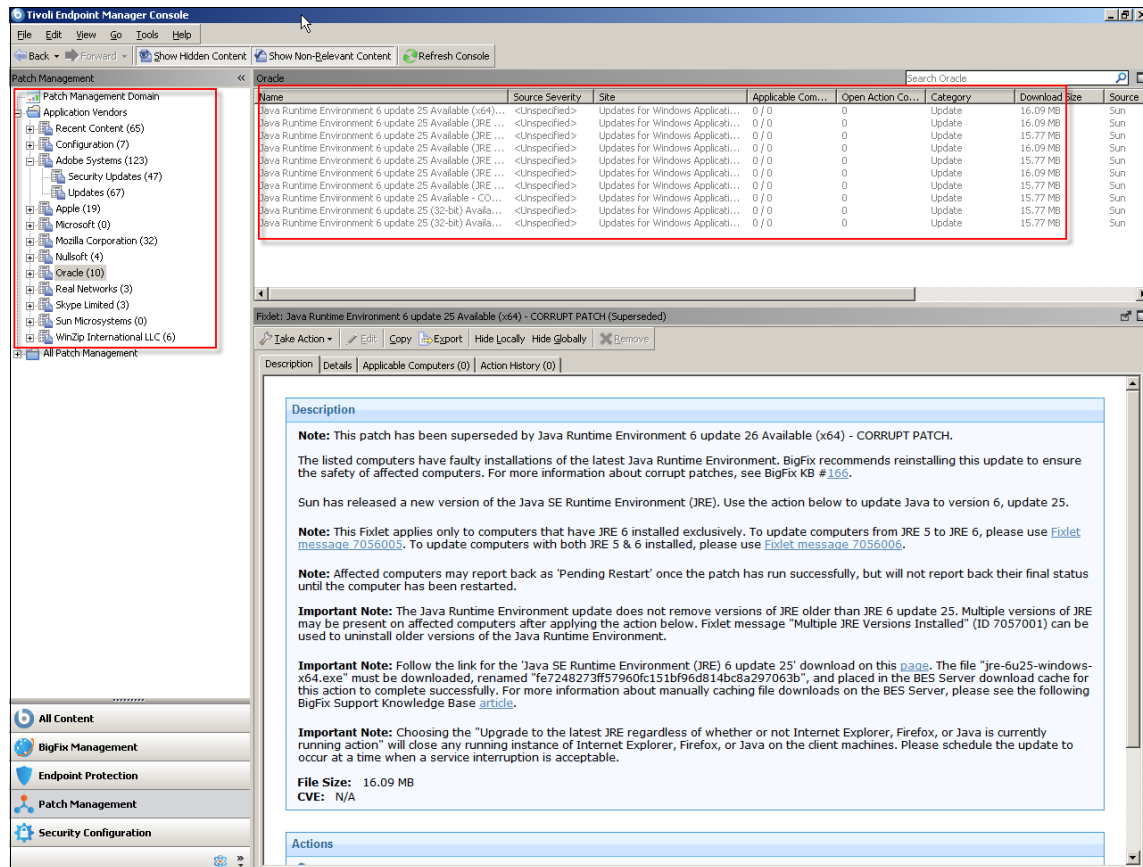


Figure 7-28 Third-party application patching in the console

Patching from outside the firewall

Tivoli Endpoint Manager is able to patch all endpoints that remain outside of the corporate firewall temporarily or consistently. In “Network and security zones” on page 137, we describe the network topology and detailed aspects about how this solution works. By using the Tivoli Endpoint Manager platform with a single intelligent Agent at the endpoint, patching can be executed on endpoints that are connected to a network anywhere in the world. Placing a Relay in the financial accounting company DMZ is the method used for the Tivoli Endpoint Manager Server to receive communications from Agents that reside on remote endpoints. Patching and general endpoint control can also be performed by using small

bandwidths because of the light communications between the Agent and the Tivoli Endpoint Manager Server.

When endpoints are offline, they can be switched on or messages can be cached at their nearest Relay so that when the endpoints reconnect to the network, the patching automatically begins.

Reboots during patching

For patches that sometimes require a reboot, the Agent contains a software component known as the *action manager* to cache all activity that must continue during a patch. It is important to remember that if the users can choose when to reboot their system, the patch that is being applied might not reply to the Tivoli Endpoint Manager Server as “patched” until the endpoint is rebooted. The financial accounting company might decide to force a reboot of the endpoints if the endpoints are on for more than a week.

7.2.5 Implementation conclusion

During the implementation of the financial accounting company Tivoli Endpoint Manager patch management solution, we incorporated a new process for patching. The new process uses the current change ticket system of the organization and the separation of duties by using a new approval process. We also described how the financial accounting company can use the Tivoli Endpoint Manager platform and patching content to patch workstations and servers, solving the business and functional requirements.

7.3 Maintenance

The intention of this section is to document the considerations of the financial accounting company about maintaining a Tivoli Endpoint Manager patch management solution. We describe the following features of Tivoli Endpoint Manager:

- ▶ Using Baseline updates
- ▶ Precaching
- ▶ Handling corrupt patches
- ▶ Minimize endpoint reboots
- ▶ Locking endpoints
- ▶ Patching overview dashboard

It is important to remember the business requirements of the patch management solution for the financial accounting company. Tivoli Endpoint Manager provides a central view of all the endpoints by using a single console, single server

solution, that is accessible by the individual operating countries. This centralized tool works even with the hierarchical nature of the IT department. Tivoli Endpoint Manager aims to automate many of the tasks involved with managing the large number of endpoints. This automation ultimately reduces the load on the department. Therefore, resources can be allocated to serve the business more efficiently.

7.3.1 Baseline updates

A Baseline is used as the deployment container for Fixlet Messages for each monthly patch cycle. The financial accounting company corporate patching process defines that the patch administrator operator clones patches from the external content site into the Custom Site before approval by the system administrator. The content that is provided by Tivoli Endpoint Manager in the form of Fixlet Messages contains an Action script designed for deployment that uses the Tivoli Endpoint Manager platform. The vendor-released patches packaged up into Fixlet Messages can undergo changes to modify the way that they are deployed to systems. These changes are to the Relevance language that is used for deployment. As a maintenance task for the financial accounting company, the Baselines created for the monthly patch cycle need *synchronizing* to ensure that any constantly enforced patches are updated on the endpoint.

To synchronize a Baseline, locate it in the Baselines tab within the Console and follow these steps:

1. Highlight the Baseline, right-click, and click **Edit**.
2. In the Edit Baseline dialog, go to the **Components** tab.
3. Baseline components whose source is modified display a Source differs message.
4. Click the [**sync with source**] link to synchronize the Baseline component with its modified source.

Figure 7-29 on page 285 displays the contents of a Baseline and the option to visit the source of the Fixlet message. In this case, there are no modifications to the source Fixlet, so we are offered the option to see the Fixlet source only.

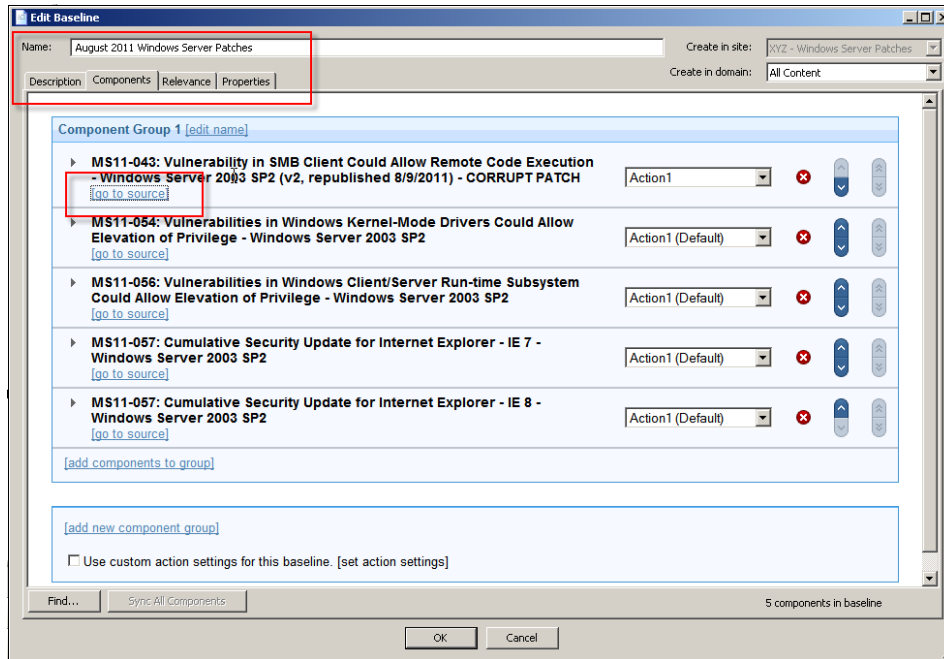


Figure 7-29 Baseline contents with up-to-date source Fixlets

Tip: To pick up the new Relevance, ensure that you deploy a new Action for the Baseline.

Baselines are also a good way of rolling up large numbers of updates into what is known as a *Gold* image for new workstations and server endpoints that join the network. This update normally occurs one time each year with new service packs, patches, and application patches and updates. The applicability of Baselines can be controlled by targeting a client setting. A Task must always be included within the Gold Baseline to change the client setting after deployment. This tip is important, because this step ensures that an entire Baseline does not become applicable on the system again, or the entire Action, large by its nature, might be propagated again.

Tip: Microsoft maintains historical information about all patches at the following address:

<http://www.microsoft.com/technet/security/bulletin/summary.aspx>

We list a few other Baseline-related maintenance tasks:

- ▶ If reusing a Baseline, ensure that it is synchronized unless specific testing was performed on an exact patch.
- ▶ Ensure that the order of installation is correct.
- ▶ Check the Relevance. Ensure that the patch is still relevant to the Baseline. Patches are also sometimes removed by vendors.

7.3.2 Precaching

The Tivoli Endpoint Manager Server and Relays can cache content within the file space allocated to the Server or Relay. This caching is useful when discussing patching because certain updates can contain large files to transfer over the network. Transferring large files is a particular concern of the financial accounting company. Caching content on Relays allows for the downloading of large files for future use, typically completed during times of low network load, which is an efficient way of downloading content.

7.3.3 Corrupted patches

In certain situations, a Windows computer that is already patched can have at least one file deprecated to a previous version, which means that the computer can be vulnerable even after patching. A *corrupt patch* situation can happen when a system is interrupted during patching (switched off, process ended unexpectedly, or a “blue screen”). The patch updated the registry but failed to update the files, or the patched file reverted or was tampered with after a patch deployment.

The financial accounting company patch team is aware of the *corrupted patch* issue. Before the team adopted Tivoli Endpoint Manager for Patch Management, they were unable to identify corrupt patches. By relying on traditional Windows automatic updates, it is impossible to detect corrupt patches in a large operational environment. This situation is an issue for the financial accounting company IT security team, because it gave a false impression of being safe from known exploits after all the critical patches were applied. In fact, a computer might still be vulnerable to attack, because the files in the system still contain those vulnerabilities, leaving the system vulnerable when those files are executed and loaded into the memory. The IT team did not have an effective way to detect any intended or unintended corruption after the computer was patched.

IBM releases new patch Fixlet Messages and the corresponding *corrupt patch* Fixlet (if there is a corrupt patch for it, shown in Figure 7-30 on page 287). So, operators can fix the corrupt patch computers immediately after patch

deployment. When the financial accounting company patch team deployed a patch that has a corrupt patch Fixlet associated with it, they monitor and fix the corrupted patch on an ongoing basis. The corrupt patch Fixlet is deployed to remediate the corrupted patch when the corrupted patch is detected.

The deployment of corrupted patch is separate from the general patch deployment process. For one reason, the corrupted patch is expected to happen only in a minority of the Windows systems patched. Another reason is that the IT team wants to track the systems where a corrupted patch occurs. A manual investigation is conducted by the IT team if a computer has a corrupt patch problem on a regular base.

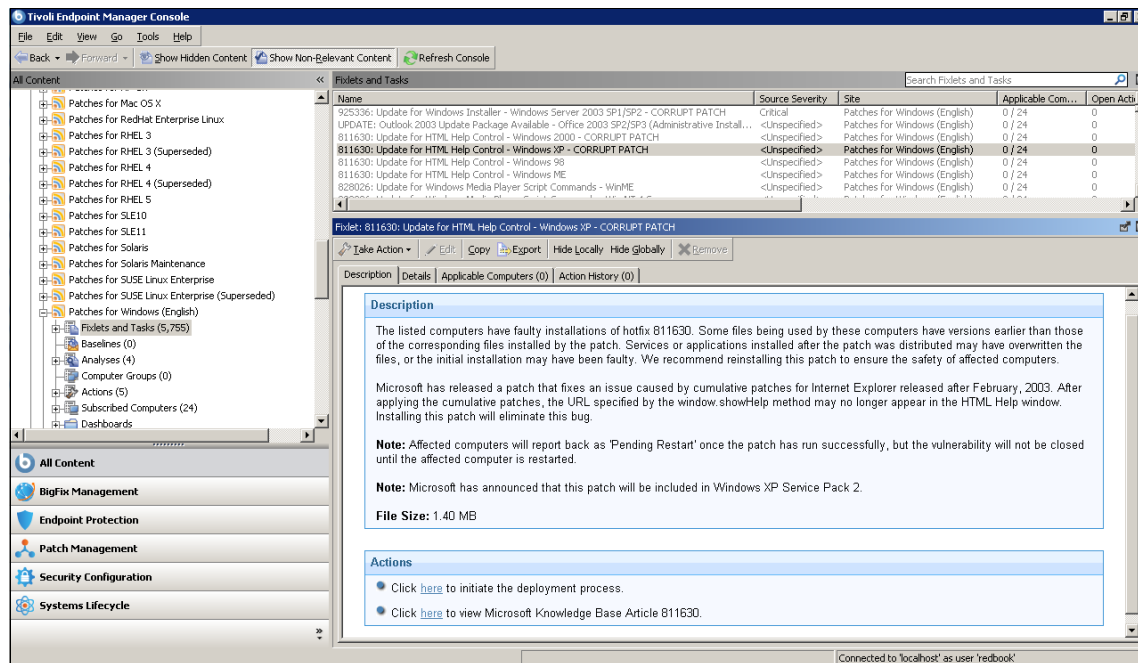


Figure 7-30 The corrupt patch Fixlet Message

7.3.4 Minimizing endpoint reboots

Organizations are sometimes required to reboot their endpoints for patching. If the reboot does not complete, the vulnerability is still reported as present. The Agent contains the action manager, which is a software component to cache all activity that must continue during a patch.

The financial accounting company needs to consider how to minimize the number of endpoint reboots. Patching during the night (1:00am - 4:00am) for servers can help avoid peak load times, shown in Figure 7-31 on page 288. The

company separates the reboot from the patching activity for workstations and allows users to choose the option to decide when to reboot due to the inconvenience of a forced reboot. However, to fix the vulnerabilities on their endpoints, the financial accounting company can force their machines to reboot one time a week. This practice is commonly known as *therapeutic reboots*.

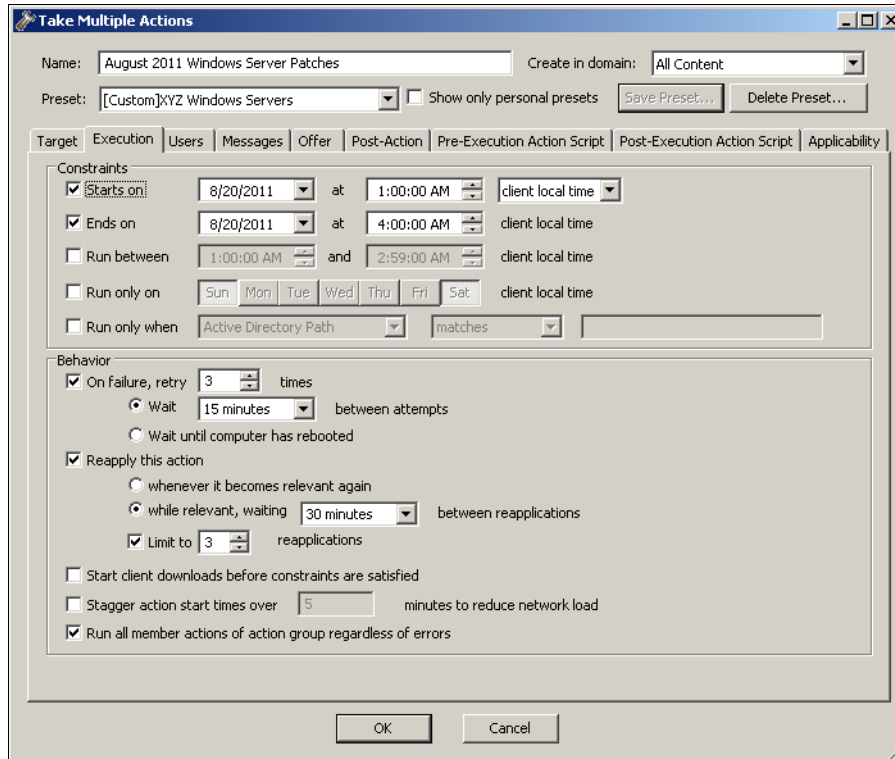


Figure 7-31 Server patching during the night

The financial accounting company separates the endpoint reboot from the patching by using these procedures:

- ▶ Performing a regular therapeutic reboot at specific times each week.
- ▶ Rebooting following patching activities each night.

Automatic Group to confine patching hours: Organizations can confine patching hours by using an *Automatic Computer Group*, which defines Relevance to get endpoints automatically to join and leave. We see the items to check in this Relevance:

- ▶ Comparing the endpoint setting for starting the Action to the current time.
- ▶ Comparing the endpoint setting for ending the Action to the current time.
- ▶ If both values are true, the endpoint joins the Automatic Group and the patch might be deployed.
- ▶ When both values are false, the endpoint disconnects from the Group and the patch Action is not relevant.

7.3.5 Locking endpoints

Certain organizations do *not* want to deploy patches at a specific time. For instance, retail organizations often have a freeze from before Thanksgiving until after the end of the year. This season is the busiest season of the year; thus, the retail organizations need to keep their businesses and systems available. Locking an endpoint permits previously defined endpoints to be exempt from any Actions although the endpoints might report relevant Fixlet Messages. It is important to remember that the IBM Support Site is exempt from locking, by default.

7.3.6 Patching overview dashboard

Tivoli Endpoint Manager can display the current patch status at a glance within the Console; this feature is available for Microsoft Windows Operating systems. Here, we can see a summary of the patch information, deployment information, and the total number of necessary patches categorized into the vendor severity rating. This dashboard provides a quick summary of the Windows operating system remediation, including the number of existing patches. Figure 7-32 on page 290 shows this dashboard.

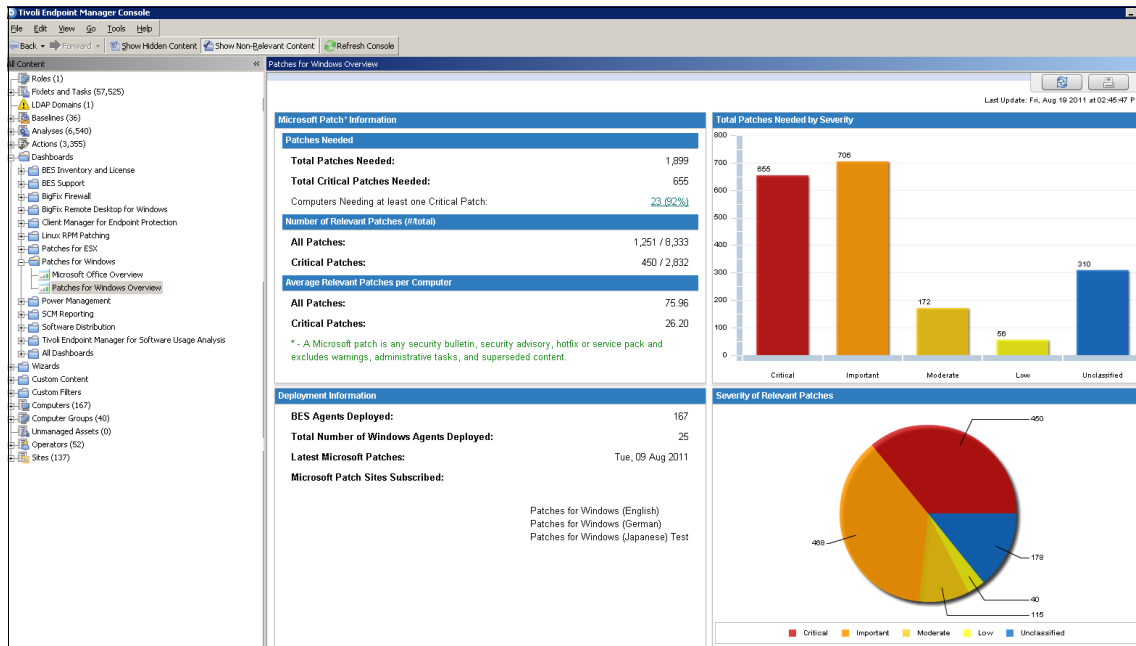


Figure 7-32 Overview dashboard for Windows patches

7.3.7 Maintenance conclusion

This section focused on the tasks that operators need to perform at various intervals to maintain the Tivoli Endpoint Manager solution. To maintain a Tivoli Endpoint Manager patching solution, you first must use the feedback provided by Fixlet messages and the dashboard reports. You can use the Web Reports tool to understand the required patches throughout the organization. By designing a patching process to be as simple as possible, you can reduce the maintenance requirements later.

7.4 Conclusion

In this chapter, we described how the financial accounting company designed its patching solution based on Tivoli Endpoint Manager and how it deployed patches to endpoints. The company designed its patching policies based on the severity from operating system or application vendors. This approach allowed the company to deploy patches by using the appropriate policies for each country. The company integrated its approval procedures with Tivoli Endpoint Manager and uses Custom Sites, operators, Tasks, and other flexible Tivoli Endpoint

Manager constructs. In the last section, we shared tips for maintenance that covered how to manage Baselines, precaching, and endpoint reboots.

In the following chapter, we describe how the Security and Compliance Management solution is implemented for the financial accounting company.



Phase III: Security policy configuration design and implementation

In this chapter, we describe the approach that the financial accounting company is taking to design the Tivoli Endpoint Manager for Security and Compliance solution. The company is using the module for *security configuration management* (SCM) that meets all of its business and functional requirements, related to the endpoint management realm. We divide the discussion into the following sections:

- ▶ “Design” on page 294
- ▶ “Implementation” on page 305
- ▶ “Project scope change” on page 337
- ▶ “SCM administration and maintenance” on page 345
- ▶ “Real-time reports” on page 348

8.1 Design

The financial accounting company wants to be in charge of security configuration management as shown in detail in Chapter 5, “Overview of scenario, requirements, and approach” on page 187. To address those requirements, we design a security configuration and compliance solution for the financial accounting company. The best starting point is a complete list of all the security policies and rules that are currently being used in the organization. Based on that start, we can map those policies with best practices and standards and validate the gaps and improvements for the financial accounting company security policies and controls.

The financial accounting company and IBM Security consultant team are working together to collect the current endpoint security policies, and update them with the new control requirements. This update is based on the functional and technical requirements discussed with Tivoli Endpoint Manager specialists and IBM security architects.

There are two methodologies to design the final solution for a new security configuration policy. They can be used independently or together, depending on the requirements:

- ▶ A *top/down* strategy assumes that the organization has a mature and well-documented security policy in place. Then, each of the policy requirements is mapped to the technical solution. In the current case, a single policy requirement can be implemented by either a single Fixlet (check) or group of Fixlets (checklist), depending on the complexity of the requirement. The advantage of this approach is that a well-defined scope is available initially, because the organizational policies are predefined. A disadvantage can be that the organization does not plan to enhance the security policy by validating the external security checklists that are available through Tivoli Endpoint Manager.
- ▶ A *bottom/up* strategy assumes that an entire set of security policies and controls must to be deployed in the environment for a later review. Each control must be evaluated and compared to the requirements of the organization. It must be determined whether the control creates value within the security and compliance area of the business. If the check is appropriate, it becomes a part of the organizational security policy and must be documented. The advantage of this approach is the chance to increase the security level of the environment. New policies might check various aspects of the environment that previously were not considered by the security policy. The drawback of the approach is the amount of work to be done, while reviewing security configuration management checklists.

- ▶ There is also a possible hybrid solution. An organization with a defined policy might be able to map its current security requirement into the technical solution. That *top/down* approach allows the IT team to verify whether the Tivoli Endpoint Manager platform is able to detect and handle all aspects defined in the policy. Then, if the organization is planning to enhance its security policies, the *bottom/up* approach must be used. The implementation of federal standards, such as US Federal Desktop Configuration Control (FDCC) regulations and Defense Information Systems Agency Security Technical Implementation Guides (DISA STIGs), can be reviewed. They can help validate that the technical solutions that apply to organizational needs must be adopted.

The financial accounting company follows the hybrid approach. In the following sections, we present the process of building the final versions of the security policies for the Tivoli Endpoint Manager platform.

For more detail about the business requirements and how they are mapped to functional requirements and technical requirements, see 5.4, “Functional requirements” on page 198.

Planned coverage: We do not intend to implement and document a real-world security policy, with all its requirements, in detail, in this book. We create a high-level endpoint security policy definition with a range of requirements to use as a fictional high-level policy in our security configuration scenario.

8.1.1 Current endpoint security policies

Before we start to implement a security configuration management solution in the organization, we need to clearly understand the current corporate security policy for endpoints. We need to divide all the requirements into groups, for example, depending on the endpoint type, such as server, notebook, workstation, or mobile device. We might list rules for each group that the organization requires to be verified. Each rule can then be implemented with Tivoli Endpoint Manager platform. This approach fits into the *up/down* implementation option.

The financial accounting company security policy for endpoints is called *Corporate IT Security Standard Endpoint 100 (CITSSE100)*. The security policy defines rules for each of the groups of machines.

The following lists of tables present an excerpt from the policy listing rules for each of the machine groups:

- ▶ CITSSE100 for workstations and notebooks (Table 8-1).

Table 8-1 CITSSE100 for workstations

CITSSE100 ID	Requirement	Definition
CIT001	Screen lock with password protection	Set a password-protected screen lock for each active account that is automatically activated by a period of inactivity, and when the workstation resumes from standby or hibernate. The inactivity time interval must be no more than 30 minutes.
CIT002	Windows Firewall protection enabled	Windows Firewall must be active on every client machine.
CIT003	Administrator automatic logon disabled	When enabled, an automatic logon occurs when a machine reboot occurs, which potentially grants full access to any unauthorized individual who starts the machine.
CIT004	Password required on resume from sleep	A password must be supplied when a notebook resumes from sleep mode (when the notebook is running on battery or plug-in mode).

- ▶ CITSSE100 for Windows Server 2003 and 2008 (Table 8-2).

Table 8-2 CITSSE100 for Windows Server systems

CITSSE100 ID	Requirement	Definition
CIT101	Anonymous share must be disabled	Given access to shared resources to anyone without a valid authentication on a server, domain, or workstation.
CIT102	Password minimum length	Password for Windows Server must be at least 10 characters in length.
CIT103	Minimum password age	Minimum password age must be greater than one day.

CITSSE100 ID	Requirement	Definition
CIT104	Maximum password age	Maximum password age must be fewer than 45 days.
CIT105	Number of failed logon attempts	The account lockout feature must be set to accept no more than two failed logon attempts to prevent brute-force password attacks on the system.
CIT106	Auditing for logon records	Audit success and failure logon events.

- ▶ CITSSE100 for IBM AIX servers (Table 8-3)

Table 8-3 CITSSE100 for IBM AIX servers

CITSSE100 ID	Requirement	Definition
CIT201	Auditing for privileged user records	Audit success and failure access events from privileged users.
CIT202	FTP without administrative privileges	FTP must not be accessible by privileged users (for example, root).
CIT203	rexec service	rexec (remote exec service) must be disabled on all servers.
CIT204	Users with a password assigned	Users must have a password assigned and stored in /etc/passwd to be granted access to the system.
CIT205	root remote connection through an encrypted channel	To access the AIX console, the root account must use an encrypted channel, such as Secure Shell (SSH).

- ▶ CITSSE100 for Linux Red Hat Enterprise Linux 5 (RHEL) servers (Table 8-4)

Table 8-4 CITSSE100 for RHEL 5 servers

CITSSE100 ID	Definition	Target
CIT301	Control of root user	Only the user root can have a privileged user account on an AIX system (UID=0).

CITSSE100 ID	Definition	Target
CIT302	FTP without administrative privileges	FTP must not be accessible by privileged users (for example, root).
CIT303	rexec service	rexec (remote exec service) must be disabled on all servers.

8.1.2 Tivoli Endpoint Manager security policy customization

The second step of defining the policy is mapping the *Corporate IT Security Standard Endpoint 100* (CITSSE100) to one or more checklist standards provided by Tivoli Endpoint Manager. The financial accounting company decided to use the DISA STIGs checklists as a base for its SCM implementation. Those checklists are available for all operating systems used by the financial accounting company.

We must identify a corresponding check within the DISA STIG checklists for each security requirement the company defined. The following list of tables presents the correlation between DISA STIG check (in columns STIG-ID, DISA Name, and Value) and the financial accounting company security requirement identifier (in column CITSSE100 ID).

We also use this mapping process to add new controls into the current CITSSE100, based on the financial accounting company DISA STIG checklist review. These controls are officially documented as part of the current security policy for the financial accounting company, which is now known as CITSSE100.

The new security controls, defined after the mapping process, are documented as *new check in the policy* (in column CITSSE100 ID) in Table 8-5. They each have an associated CITSSE100 ID in 8.1.3, “Designing a new policy model” on page 301, where we have the new *top/down* design strategy ready to be configured into Tivoli Endpoint Manager.

- Mapping CITSSE100 for workstation and DISA STIG (Table 8-5).

Table 8-5 CITSSE100 and DISA mapping for workstation

CITSSE100 ID	STIG-ID	DISA name	Value
CIT001	V-1122	The system configuration is not set with a password-protected screen saver (ScreenSaverIsSecure).	1 (Enabled)

CITSSE100 ID	STIG-ID	DISA name	Value
CIT002	V-17411	Firewall Standard Profile - Enable Firewall.	1 (Enabled)
CIT003	V-1145	Administrator automatic logon is enabled.	1 (Enabled)
CIT004	V-14267	Password required on resume from hibernate or suspend.	1 (Enabled)
New check in the policy	V-1107	Password uniqueness does not meet minimum requirements.	24
New check in the policy	V-1115	The built-in administrator account is not renamed.	=! Administrator

- Mapping CITSSE100 for Windows Server and DISA STIG (Table 8-6).

Table 8-6 CITSSE100 and DISA mapping for Windows Server

CITSSE100 ID	STIG-ID	Requirement	Value
CIT101	V-3340	Unauthorized shares can be accessed anonymously.	<none>
CIT102	V-6836	For systems that use a logon ID as the individual identifier, passwords are not at a minimum of 14 characters.	14
CIT103	V-1105	Minimum password age does not meet requirements.	1
CIT104	V-3373	The maximum age for machine account passwords does not meet requirements.	30
CIT105	V-1097	Number of allowed failed logon attempts does not meet requirements.	3
CIT106	V-6850	Auditing records are configured as required (Audit Logon events).	Audit success and failed events

CITSSE100 ID	STIG-ID	Requirement	Value
New check in the policy	V-1098	Time before failed logon counter is reset does not meet requirements.	60 minutes

- ▶ Mapping CITSSE100 for IBM AIX servers and DISA STIG (Table 8-7).

Table 8-7 CITSSE100 and DISA mapping for AIX servers

CITSSE100 ID	STIG-ID	Requirement	Value
CIT201	GEN000880	root account only with UID 0 - AIX 6.1.	Only root with UID 0
CIT202	GEN004900	ftputers file must not contain system users - AIX 6.1.	system, root, bin, sys, daemon, listen, nobody
CIT203	GEN003840	rexec service enabled - AIX 6.1.	disabled
CIT204	GEN000650	Password-protected enabled accounts - AIX 6.1.	true
CIT205	GEN001100	Unencrypted network root access - AIX 6.1.	sshd must be running.

- ▶ Mapping CITSSE100 for Linux RHEL 5 servers and DISA STIG (Table 8-8).

Table 8-8 CITSSE100 and DISA mapping for Linux RHEL 5

CITSSE100 ID	STIG-ID	Definition	Value
CIT301	GEN000880	root account only with UID 0 - RedHat/CentOS 5.	Only root with UID 0
CIT302	GEN004900	ftputers file must not contain system users - RedHat/CentOS 5.	system, root, bin, sys, daemon, listen, nobody
CIT303	GEN003840	rexec service enabled - RedHat/CentOS 5.	disabled

CITSSE100 ID	STIG-ID	Definition	Value
New check in the policy	GEN002160	Shells SUID - Red Hat/CentOS 5.	remove SUID bit
New check in the policy	LNx00140	Password protecting the GRUB Console Boot Loader.	password-protected
New check in the policy	GEN000700	Maximum password age for RedHat/CentOS 5.	90

8.1.3 Designing a new policy model

In the previous section, we created a mapping between the organizational requirements and the federal standards checklist. Now, we need to define the final version of the security policy for the financial accounting company. The financial accounting company, together with the IBM specialist, must review the recommended values provided by DISA STIG. Then, they update them if necessary, in accordance with the organizational security vision and risk mitigation control.

After this review, we present the final version of the corporate security policy to be implemented by using the Tivoli Endpoint Manager platform:

- ▶ CITSSE100 for Windows XP V1.0 (Table 8-9).

Table 8-9 Checklist for Windows XP

CITSSE100 ID	Control name	Required value
CIT001	The system configuration is not set with a password-protected screen saver (ScreenSaverIsSecure).	1 (Enabled)
CIT002	Firewall Standard Profile - Enable Firewall.	1 (Enabled)
CIT003	Administrator automatic logon is enabled.	1 (Enabled)
CIT004	Password required on resume from hibernate or suspend.	1 (Enabled)
CIT005	Password uniqueness does not meet requirements.	10
CIT006	The built-in administrator account is not renamed.	≠! Administrator

- ▶ CITSSE100 for Windows 7 V1.0 (Table 8-10).

Table 8-10 Checklist for Windows 7

CITSSE100 ID	Control name	Required value
CIT001	The system configuration is not set with a password-protected screen saver (ScreenSaverIsSecure).	1 (Enabled)
CIT002	Firewall Standard Profile - Enable Firewall.	1 (Enabled)
CIT003	Administrator automatic logon is enabled.	1 (Enabled)
CIT004	Password required on resume from hibernate or suspend.	1 (Enabled)
CIT005	Password uniqueness does not meet requirements.	10
CIT006	The built-in administrator account is not renamed.	≠! Administrator

- ▶ CITSSE100 for Windows 2003 V1.0 (Table 8-11).

Table 8-11 Checklist for Windows 2003

CITSSE100 ID	Control name	Required value
CIT101	Unauthorized shares can be accessed anonymously.	<none>
CIT102	For systems that use a logon ID as the individual identifier, passwords are not at a minimum of 14 characters.	10
CIT103	Minimum password age does not meet requirements.	1
CIT104	The maximum age for machine passwords is not set to requirements.	45
CIT105	Number of allowed failed logon attempts does not meet requirements.	3
CIT106	Auditing records are configured as required (Audit Logon events).	Audit success and failed events
CIT107	Time before failed logon counter is reset does not meet requirements.	30 minutes

- ▶ CITSSE100 for Windows 2008 V1.0 (Table 8-12).

Table 8-12 Checklist for Windows 2008

CITSSE100 ID	Control name	Required value
CIT101	Unauthorized shares can be accessed anonymously.	<none>
CIT102	For systems that use a logon ID as the individual identifier, passwords are not at a minimum of 14 characters.	10
CIT103	Minimum password age does not meet requirements.	1
CIT104	The maximum age for machine passwords is not set to requirements.	45
CIT105	Number of allowed failed logon attempts does not meet requirements.	3
CIT106	Auditing records are configured as required (Audit Logon events).	Audit success and failed events
CIT107	Time before failed logon counter is reset does not meet requirements.	30 minutes

- ▶ CITSSE100 for AIX6 V1.0 (Table 8-13).

Table 8-13 Checklist for AIX servers

CITSSE100 ID	Control name	Required value
CIT201	root account only with UID 0 - AIX 6.1.	Only root with UID 0
CIT202	ftpusers file must not contain system users - AIX 6.1.	system, root, bin, sys, daemon, listen, nobody
CIT203	rexec service enabled - AIX 6.1.	Disabled
CIT204	Password-protected enabled accounts - AIX 6.1.	True
CIT205	Unencrypted network root access - AIX 6.1.	sshd must be running

- ▶ CITSSE100 for RHEL5 v1.0 (Table 8-14).

Table 8-14 Checklist for Linux RHEL 5

CITSSE100 ID	Control name	Value
CIT301	root account only with UID 0 - Red Hat/CentOS 5.	Only root with UID 0
CIT302	ftputers file must not contain system users - Red Hat/CentOS 5	system, root, bin, sys, daemon, listen, nobody
CIT303	rexec service enabled - Red Hat/CentOS 5.	Disabled
CIT304	Shells SUID - Red Hat/CentOS 5.	Remove SUID bit
CIT305	Password protecting the GRUB Console Boot Loader.	Password-protected
CIT306	Maximum password age - Red Hat, CentOS 5.	90

8.2 Implementation

The financial accounting company must update the security policies to accomplish the internal and external compliance requirements and to improve its security configuration controls. However, the financial accounting company adopted a hybrid approach for its security configuration strategy, because the security team is not sufficiently confident with using a *top/down* approach only.

Based on this approach, we create two custom Fixlet Sites for each platform:

- ▶ The first Custom Site is used for the *top/down* approach, where the financial accounting company can monitor and remediate configuration issues based on the current security policy. Thus, the company knows whether it applies the security policy with the controls and rules as defined by the financial accounting company IT and security teams.
- ▶ The second Custom Site is used for the *bottom/up* strategy, where the company keeps all security configuration controls, provided by DISA STIG “as-is” to be validated by the financial accounting company. Thus, the financial accounting company can determine which controls must be part of the official security policy. After the controls are chosen, the controls are moved from the *bottom/up* view to the *top/down* view.

This approach is tracked either for six months or until the bottom/up Custom Site (the second Site for each platform) is empty, due to the security checks (controls) migration from one Custom Site to the other Custom Site.

We use the following naming conventions for each Custom Site:

- ▶ For the *top/down* approach, we use:
CITSSE100 for *<platform> <version>*
- ▶ For the *bottom/up* approach, we use:
<federal checklist best practice> chklist for *<platform>*

8.2.1 Create Custom Site for policies

Security configuration management tasks, and other activities related to Fixlets, must be managed inside of a Custom Site. This approach provides flexibility to the administrators, because it allows the administrators to assign permissions for different operators. It can also prevent the Site content from being updated in an uncontrolled manner.

Enabling SCM external Sites

First, we subscribe our Tivoli Endpoint Manager Server to external Sites that provide security configuration management content. This approach allows us to access the available best practices checklists implemented by federal standards, such as FDCC or DISA STIG.

For the financial accounting company Tivoli Endpoint Manager environment, we enabled the external Sites shown in Table 8-15. We can map the financial accounting company current security policy to the best practices content provided by the federal best practices standards from those external Sites (top/down approach). Or, we can review Sites content and select the desired Fixlets to add as the new security checks for the corporate security policy for the financial accounting company (bottom/up approach).

Table 8-15 External sites

External Fixlet Site	Managed environment
DISA STIG on Windows 2003 MS V6R1.18	Windows Server 2003
DISA STIG on Windows 2008 MS V6R1.11	Windows Server 2008
DISA STIG on Windows 7 V1R2	Windows 7 workstations
DISA STIG on Windows XP V6R1.18	Windows XP workstations
SCM Checklist for DISA STIG on RHEL 5	Linux RHEL 5 servers

External Fixlet Site	Managed environment
SCM Checklist for DISA STIG on AIX 6.1	AIX servers
Security Policy Manager	Windows
SCM Reporting	All machines

To enable gathering Site content from the IBM Fixlet Server, we need to open the License Overview dashboard and activate the desired items. We follow these steps:

1. Open the BigFix Management domain by clicking the appropriate button on the lower-left side of the Tivoli Endpoint Manager Console.
2. In the content tree, on the left side, we select **License Overview**.
3. The License Overview dashboard appears. Then, we locate the Security and Compliance section of the dashboard. A list of the Fixlet Sites that are available shows within our current Tivoli Endpoint Manager license.
4. We click **[Enable]** on the lines that match the list defined in Table 8-15 on page 306.

Figure 8-1 on page 308 depicts the status of Sites, which were enabled for our current environment. Under that table with those Sites, there are lines that list other Sites that are available for the current license.

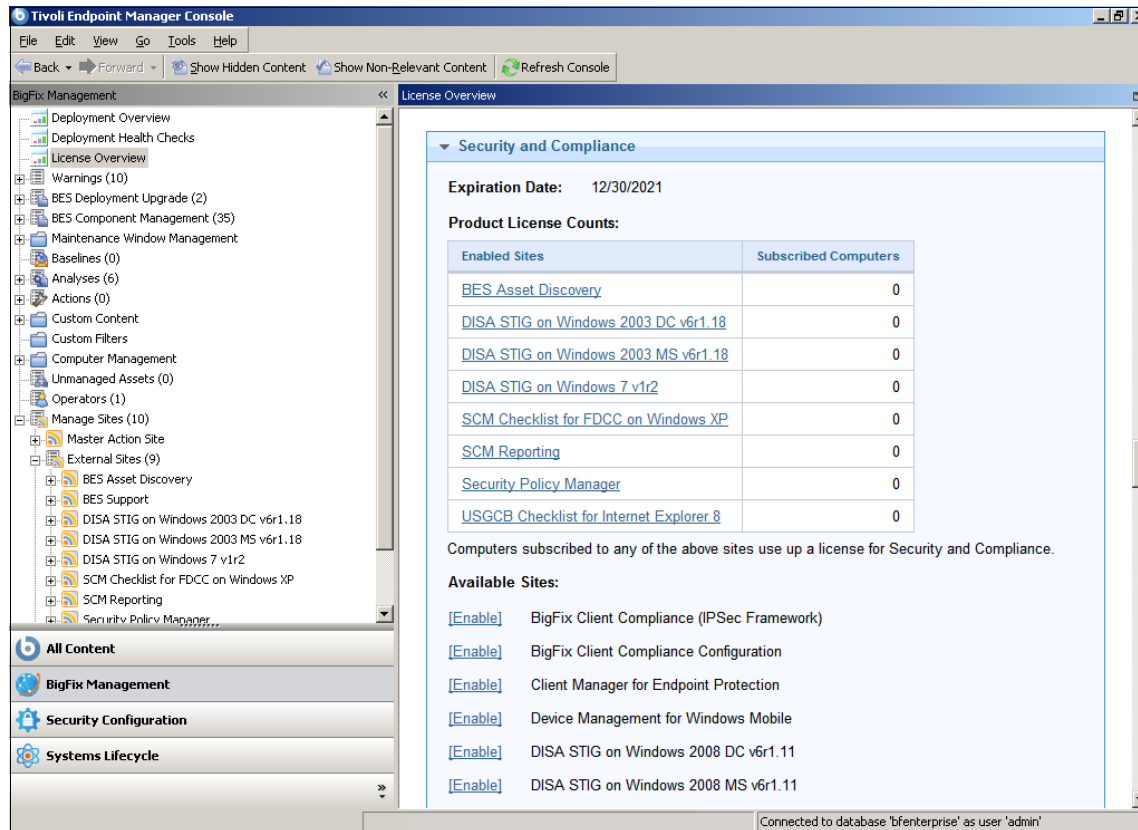


Figure 8-1 Subscribing the Tivoli Endpoint Manager environment to external Sites

For more details about how to subscribe to an external Site by using the License Overview dashboard or through the external Site masthead, see the *Tivoli Endpoint Manager Console Operator's Guide*, which you can obtain at the IBM Tivoli Endpoint Manager V8.2 Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v26r1/topic/com.ibm.tem.doc_8.2/Platform/Console/c_selecting_sites.html

Cloning the SCM checklist from the Custom Checklist Wizard

Previously, we made external Sites available in the financial accounting company Tivoli Endpoint Manager environment. Now, we need to prepare the financial accounting company-specific Custom Sites with only Fixlets that address the defined requirements.

We create one Custom Site for each operating system to manage. The number of Fixlets available in external Sites is about 200 Fixlets in each Site. Because we

plan to create several Custom Sites, the process looks complicated and time-consuming. It might also cause errors if performed manually. Tivoli Endpoint Manager provides a tool, the Custom Checklist Wizard, that simplifies the process of creating a Custom Site.

Continuous development: The *Custom Checklist Wizard* tool presented in this book is constantly enhanced. The user interface might change at any time for better usability or to contain more function. We try to provide comprehensive documentation about this tool, considering future updates. However, even if the user interface differs, the underlying concept of helping the administrator to create a Custom Site remains the same.

To start the Custom Checklist Wizard tool, follow these steps:

1. Open the Security Configuration domain by clicking the appropriate button on the lower-left side of the Tivoli Endpoint Manager Console.
2. In the content tree, choose **Configuration Management** → **Checklist Tools** → **Create Custom Checklist Wizard**.
3. A wizard starts in the Tivoli Endpoint Manager Console and lists all available security configuration management Fixlet Sites.

Figure 8-3 on page 312 shows the wizard after it is started. We look at the layout of the wizard.

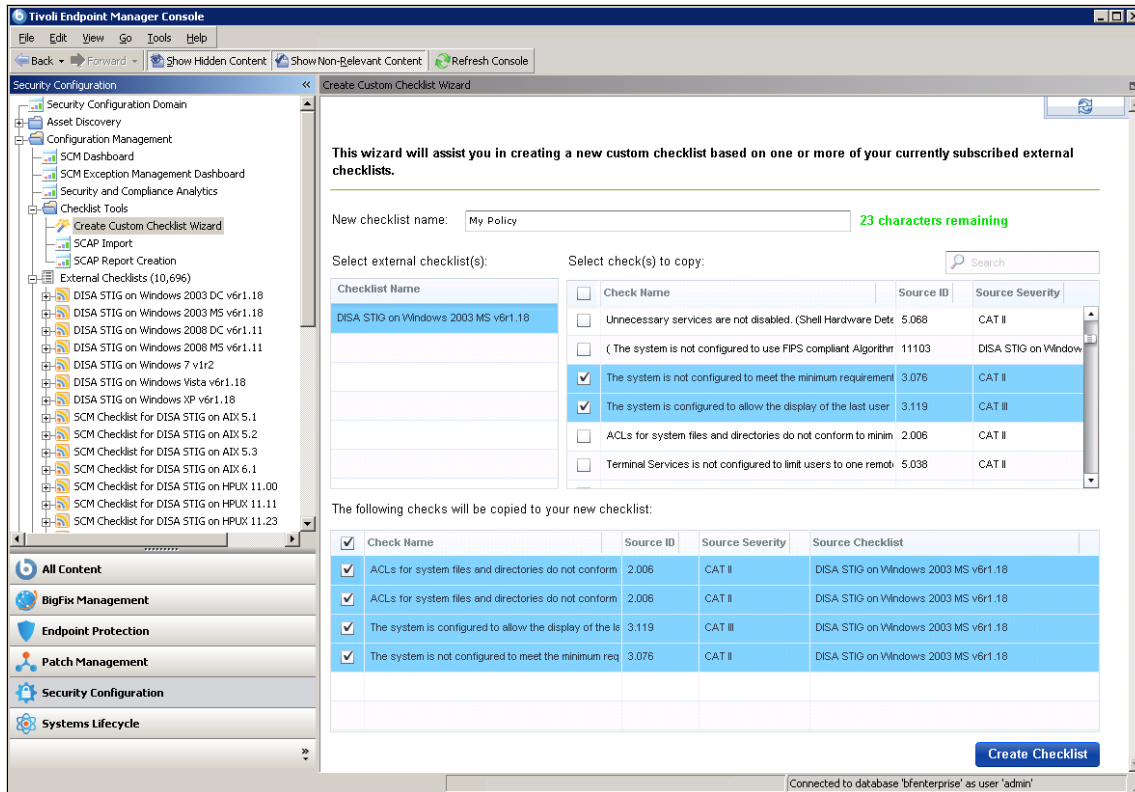


Figure 8-2 Custom Checklist Wizard form

The wizard layout consists of several lists and forms. To proceed to the next step, we need to be familiar with those controls. In the following list, we provide information about each of the visible controls:

- ▶ *New checklist name* field allows the user to enter the name for the Custom Site that is automatically created and filled with Fixlet, Task, and Analysis components as defined by the administrator. The Site name must be a maximum of 32 characters.
- ▶ *Select external checklist(s)* displays all Fixlet Sites that have security configuration management Fixlets. They display all Fixlet Sites that are available in the current Tivoli Endpoint Manager deployment.
- ▶ *Select check(s) to copy* displays Fixlets that are available in the currently selected Site in the previous list. The user can select Fixlets to be part of the newly created Site. The wizard marks selected objects with a check in the first column of the list.

There is also a Search field available, which allows instant filtering of the Fixlet list.

- ▶ *Final Fixlets list.* Based on the check boxes selection in the *Select check(s) to copy* list, the final Fixlet list is populated. This list can contain Fixlets from different security configuration management Sites. The *Source Checklist* column indicates the origin of the individual Fixlets.
- ▶ *Create Checklist* creates the Custom Site with the name provided in the first field, with the content defined in the final Fixlets list.

We now need to create a Custom Site for the financial accounting company. In “Enabling SCM external Sites” on page 306, we enabled the collection of several security configuration management external Sites. Now, we can start the Custom Checklist Wizard to prepare our content. We open the wizard in the financial accounting company environment.

Repeat for each policy: In the following section, we describe the Site creation steps for the *CITSSE100 for Windows XP V1.0 (CITSSE100 for WinXP v1.0)* policy. However, we must repeat the steps for each policy defined in 8.1.3, “Designing a new policy model” on page 301.

We follow these steps:

1. As a checklist name, we type CITSSE100 for WinXP v1.0 in the New checklist name field.
2. We select the external checklist called **DISA STIG on Windows XP v6r1.18**. The list of Fixlets is displayed in the list on the right side.
3. According to the policy defined in the Table 8-9 on page 302, we select the checks. The wizard automatically adds the selected items to the list at the bottom of the window.

Figure 8-3 on page 312 shows these steps.

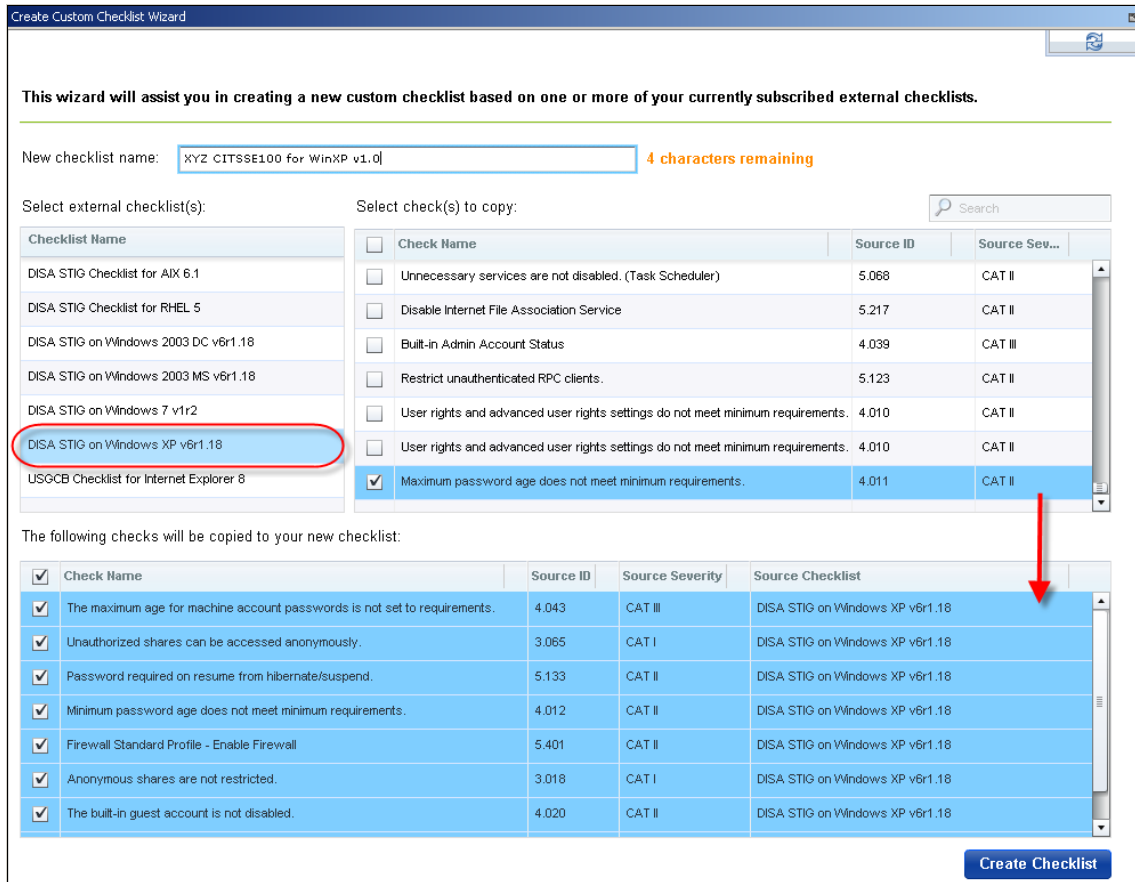


Figure 8-3 Custom Site creation with the Custom Checklist Wizard tool

4. To create a Custom Site, we click **Create Checklist**. The Site is created by the wizard, with all the required Fixlets and Analysis components.
5. The generated Site is not distributed to any of managed endpoints automatically. To allow evaluation and compliance reporting, we need to subscribe the computers. To allow easier management, we created an automatic computer group called All Windows XP workstations that aggregates the machines that run this operating system. This group must subscribe to the CITSSE100 for WinXP v1.0 Site. Figure 8-4 on page 313 shows the subscription settings.

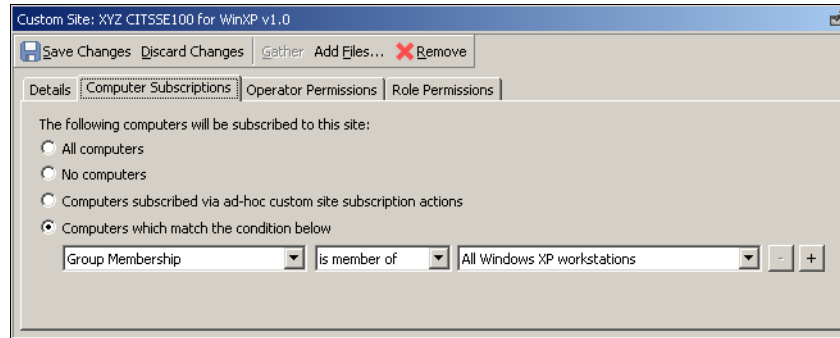


Figure 8-4 Site subscription settings for CITSSE100 for WinXP v1.0 checklist

6. After we click **Save Changes**, all computers that belong to the defined group receive the Site Fixlet and begin to report their compliance status.

We must repeat all the steps that we performed for the creation of the policy of Windows XP machines for the following remaining policies:

- ▶ CITSSE100 for Win7 v1.0
- ▶ CITSSE100 for Win2k3 v1.0
- ▶ CITSSE100 for Win2k8 V1.0
- ▶ CITSSE100 for AIX6 v1.0
- ▶ CITSSE100 for RHEL5 v1.0

To work through the *bottom/up* approach, we need to repeat the previous steps for each platform. In addition to using different names for the Sites, the only difference in content is in step 3. We must select all checks, by selecting the check box on the right side of Check Name header, as indicated in Figure 8-5 on page 314. We check these Custom Sites:

- ▶ DISA chklist for WinXP
- ▶ DISA chklist for Win7
- ▶ DISA chklist for Win2k3
- ▶ DISA chklist for Win2k8
- ▶ DISA chklist for AIX6
- ▶ DISA chklist for RHEL5

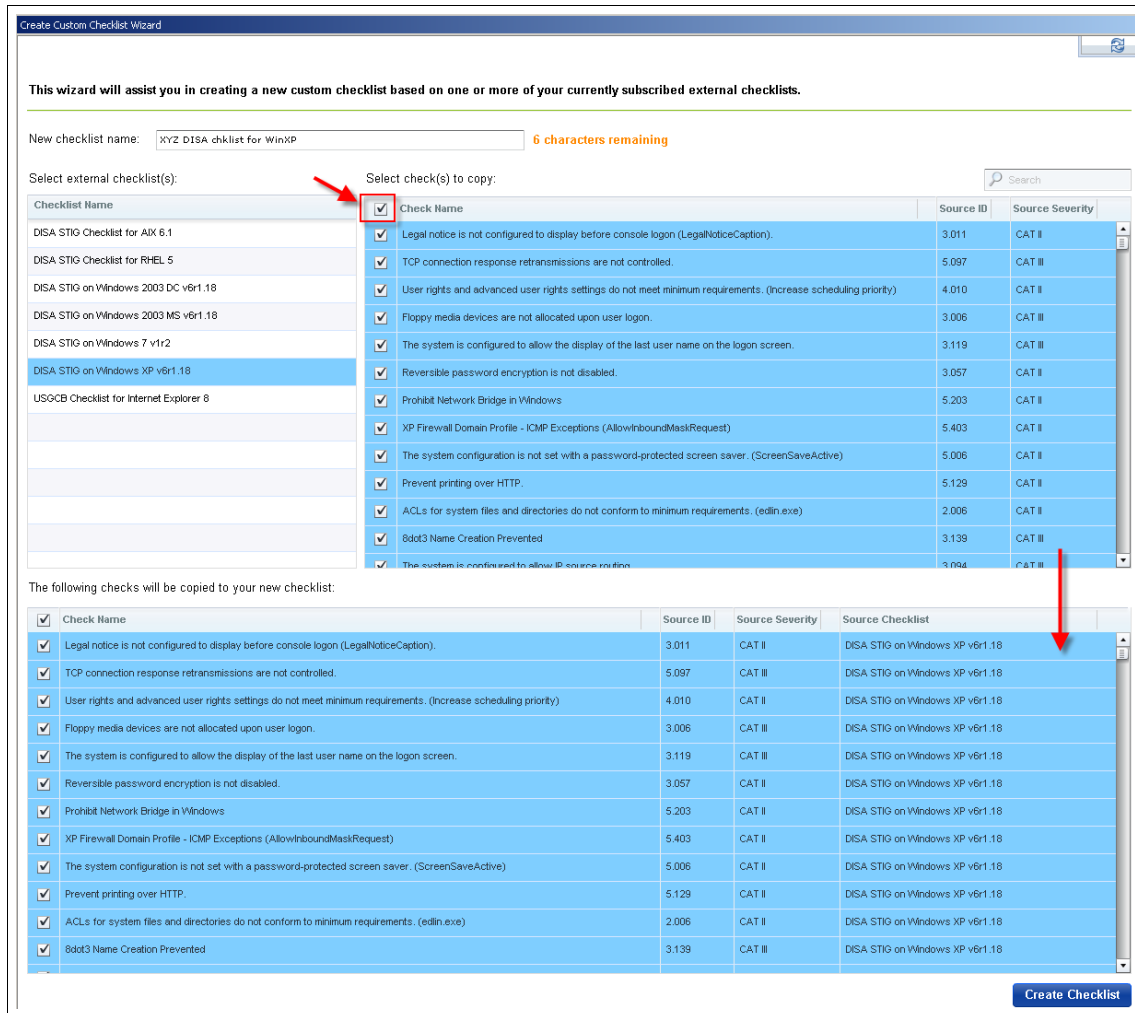


Figure 8-5 Custom Site creation, with all checks, with the Custom Checklist Wizard tool

8.2.2 Customizing SCM Fixlets

Ideally, the IBM provided policies and Fixlets can fit the requirements of the organization without any customization. However, areas always exist where changes are required, based on organization-specific policies, standards, or cultural aspects.

Because we are now familiar with Tivoli Endpoint Manager, we conclude that a *Fixlet* for the SCM module is supposed to perform compliance checking. We also conclude that *Tasks* are responsible for configurations and actions that need to

be executed in the environment. In many cases, Fixlets can use *Task action results* and, therefore, return different compliance states. But this solution is not usable. Tivoli Endpoint Manager offers another way to customize Fixlet behavior. In 4.4.2, “Security configuration management Fixlet design” on page 169, we introduced the *self-parameterized Fixlet*. Now, we show this type of Fixlet.

There are multiple Fixlets that allow customization among the DISA STIG checklists. As an example, we use the *Password uniqueness does not meet minimum requirements* Fixlet. Figure 8-6 shows the chosen Fixlet as a part of DISA STIG on the Windows XP v6r1.18 Fixlet Site.

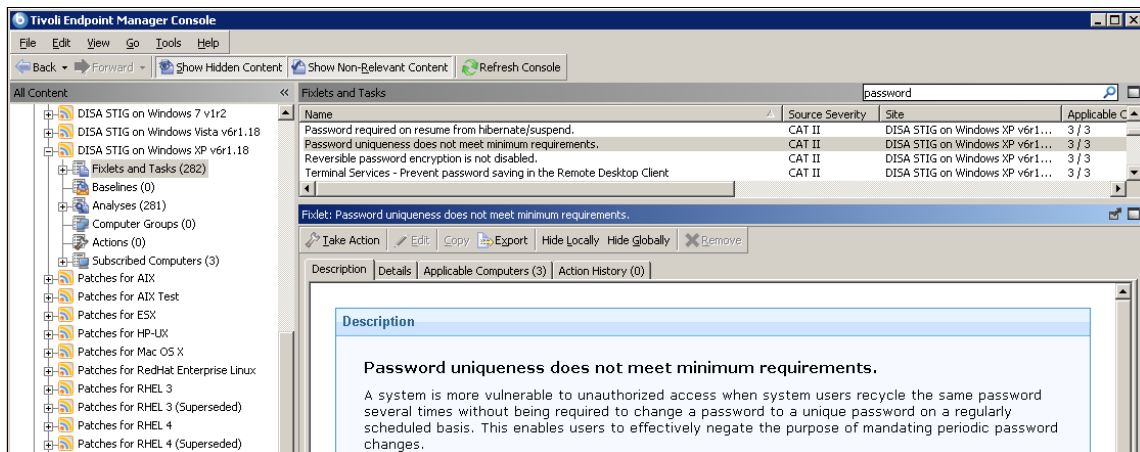


Figure 8-6 Password uniqueness does not meet minimum requirements Fixlet that is visible in the Console

As you can see in the Fixlet description, the operating system settings are checked regarding the amount of historical passwords that are stored. The Fixlet becomes *relevant* (which means, it reports non-compliance) in case the operating system settings do not fulfill the value defined in the Fixlet. We look at the full Fixlet description in Figure 8-7 on page 316.

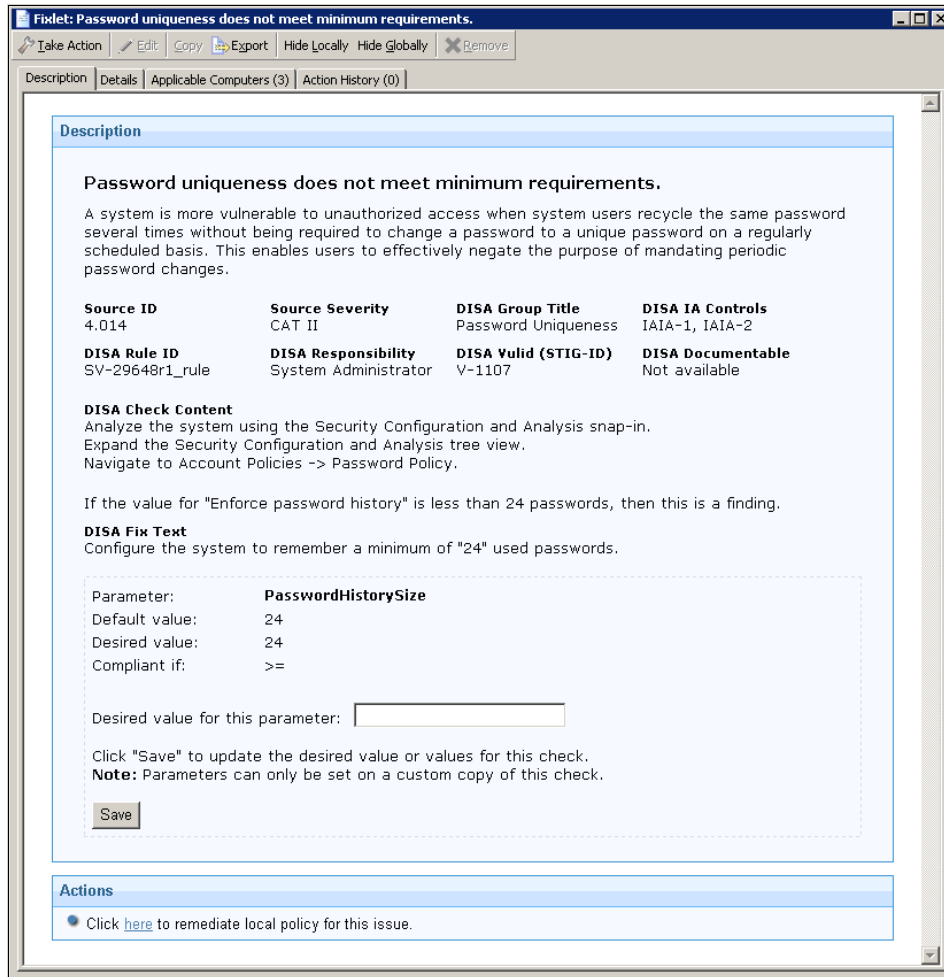


Figure 8-7 Self-parameterized Fixlet description

This Fixlet provides a significant amount of information, including a high-level description of the system settings that are checked. This Fixlet is a part of the DISA STIG standard implementation. Therefore, there are identifiers used in the standard that allow operators to match the Fixlet to the appropriate definition. There is also a description, *DISA Check Content*, that explains how to find the actual setting directly in the operating system configuration. The bottom part of the Fixlet description provides a section that defines the parameters used for the Fixlet customization.

Custom Sites: As visible in the Fixlet description in Figure 8-7 on page 316 and as introduced in 4.4.2, “Security configuration management Fixlet design” on page 169, self-parameterized Fixlets cannot be customized in the external Fixlet Site. The external Fixlet Site is provided by IBM and cannot be modified. To change any Fixlet setting, you need to create a Custom Site and then copy the Fixlet to that Site. You can change a Fixlet setting manually in the Fixlet list or use a tool, such as the *Create Custom Checklist Wizard*.

Customization

Before we customize the Fixlet, we look at the current settings and technical details of the Fixlet. The Fixlet checks the default value of 24. Figure 8-8 shows the relevance statement that is used for the compliance evaluation.

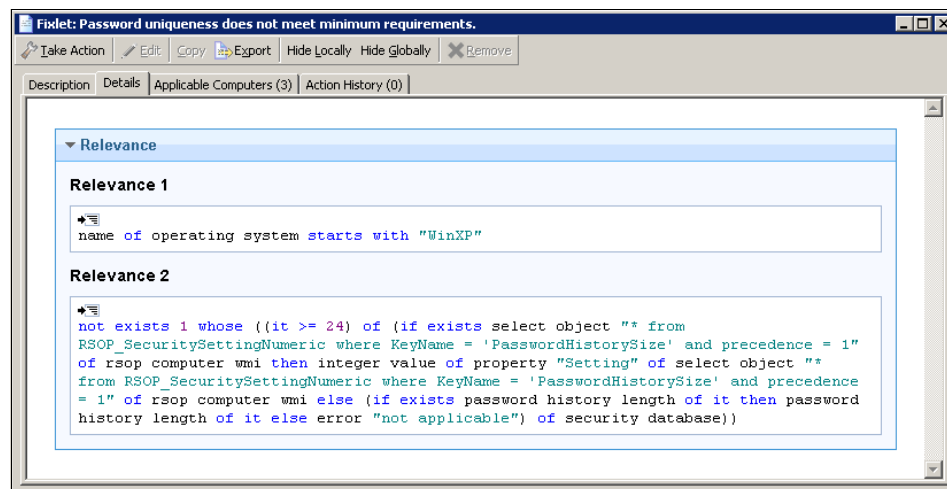


Figure 8-8 Password uniqueness Fixlet relevance clause

The *Relevance 1* statement targets the correct operating system. The statement called *Relevance 2* is used to check the value. The most important part is to check the following condition:

```
... whose ((it >= 24) of ....
```

The *it*¹ clause refers to the actual system settings. The condition checks whether it is greater than or equal to an integer value. This value is the same value that is defined in the Fixlet description.

¹ For more information about how to use the Relevance language, see this website: <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#wiki/Tivoli%20Endpoint%20Manager/page/Fixlet%20Authoring>

Because we already defined our custom checklist, we can now change the Fixlet parameters according to our policy. In Table 8-9 on page 302, we defined requirement CIT005 for the system setting of the password uniqueness equal to 10. Next, we must update the Fixlet with the new value that we want. We must enter the new number in the parameter field, as shown in Figure 8-9.

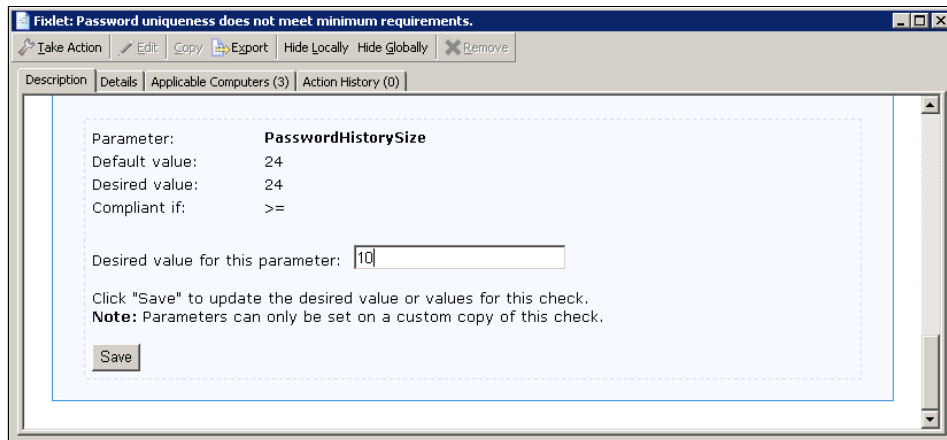


Figure 8-9 Defining the new Fixlet parameter

After we click **Save**, the Fixlet automatically updates itself. The new value is visible in the description next to the Desired value label. The updated Fixlet is automatically distributed to all endpoints that have this Fixlet assigned. For the Windows targeting Fixlets, no other action is required. After the updated Fixlet is delivered to the Tivoli Endpoint Manager Agents, the evaluation is executed against the new value. In the Tivoli Endpoint Manager Console, the updated Fixlet is displayed, as shown in Figure 8-10 on page 319.

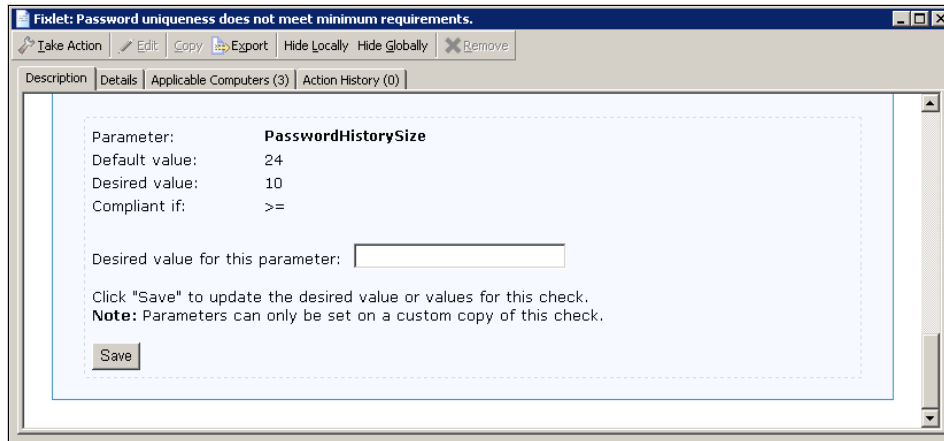


Figure 8-10 Fixlet with updated parameter

We can also review the Relevance statements by switching to the Details tab. As shown in Figure 8-11, the condition is updated to verify the system setting against the new value.

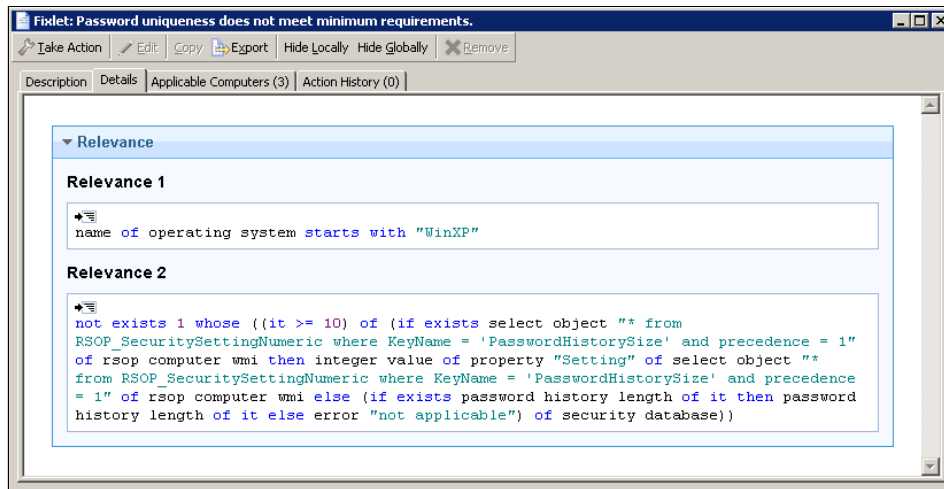


Figure 8-11 Password uniqueness Fixlet relevance clause with updated value

Remediation

Most of the compliance Fixlets are delivered by IBM with the appropriate remediation capabilities. After a Fixlet detects noncompliance, a Tivoli Endpoint Manager operator is able to initiate an Action to fix the detected vulnerability. Remediation steps are defined by using the *Action Script*² language. Figure 8-12

on page 320 shows the fix procedure defined for the password uniqueness Fixlet, *before* changing the parameter value.

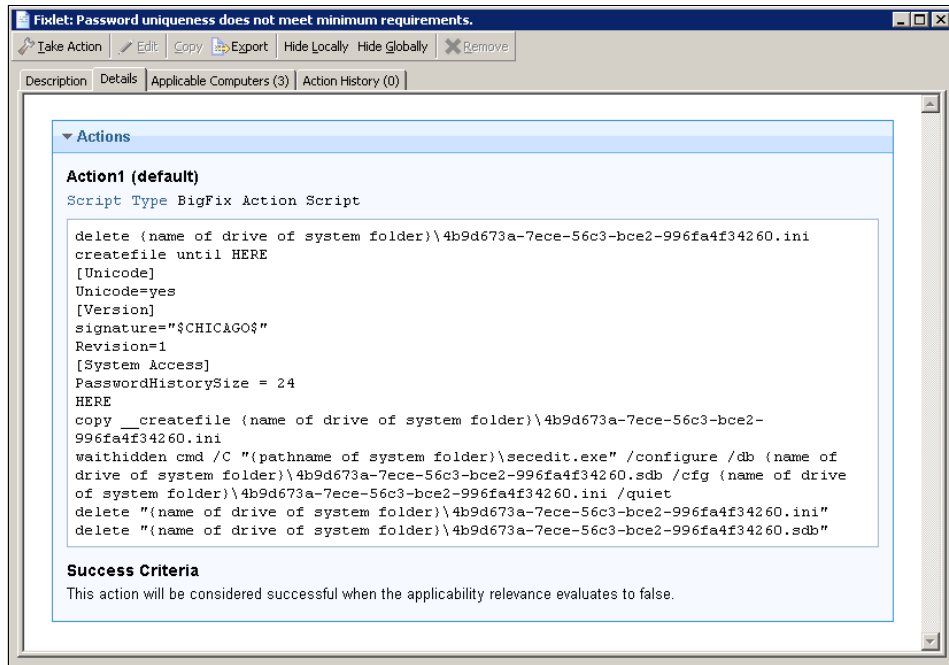


Figure 8-12 Password uniqueness Fixlet remediation Action Script

After the administrator updates the Fixlet parameter, as described in “Customization” on page 317, the remediation Action Script is also updated. We update the part of the script that is responsible for the setting:

```
[System Access]
PasswordHistorySize = 10
```

If the operator now starts to fix the issue that is reported by the Fixlet, the updated value is used.

8.2.3 Compliance evaluation

Tivoli Endpoint Manager provides a platform for continuously reporting the predefined policies of the entire system. In addition to observing the Fixlet status, operators must be aware of other aspects of the platform.

² For more information about how to use Action Scripts, see this website:
<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en/wiki/Tivoli%20Endpoint%20Manager/page/Fixlet%20Authoring>

Measured values analysis

In 4.4.2, “Security configuration management Fixlet design” on page 169, we introduced the concept of the logical link between Fixlet and Analysis. A Fixlet can provide only limited information about the checked aspect of the operating system. One of two Fixlet states, *relevant* or *not relevant*, might sometimes not provide the full explanation of what is happening in the IT endpoint environment. This full explanation is especially important in the security configuration management field, when an operator cares about the level of noncompliance rather than the noncompliance.

We look at the password uniqueness Fixlet introduced in 8.2.2, “Customizing SCM Fixlets” on page 314. If the Fixlet becomes relevant on a set of machines after we set the desired parameter value to 10, we do not know whether all relevant machines have a setting of 9, 5, or any other value. From the fixing priority point of view, if the noncompliant machines have the setting equal to 9, the severity of the issue might be low. If there is no password uniqueness setting defined, the vulnerability level is high.

Figure 8-13 shows the Analysis that is supposed to provide more detailed information about the system setting that is checked by the password uniqueness Fixlet.

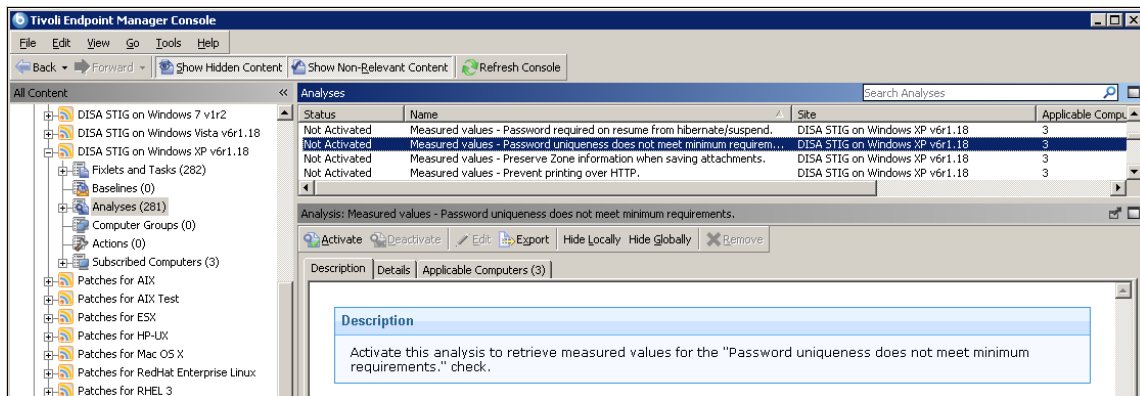


Figure 8-13 Activating Analysis

After it is activated, the Analysis evaluates the Relevance statement to gather the password setting from all workstations. The results are visible immediately in the Tivoli Endpoint Manager Console. Figure 8-14 on page 322 presents some sample values, gathered from a group of 163 computers. If we assume that the desired value of the setting is equal to 10, only five machines are not compliant with a password history size of 8.

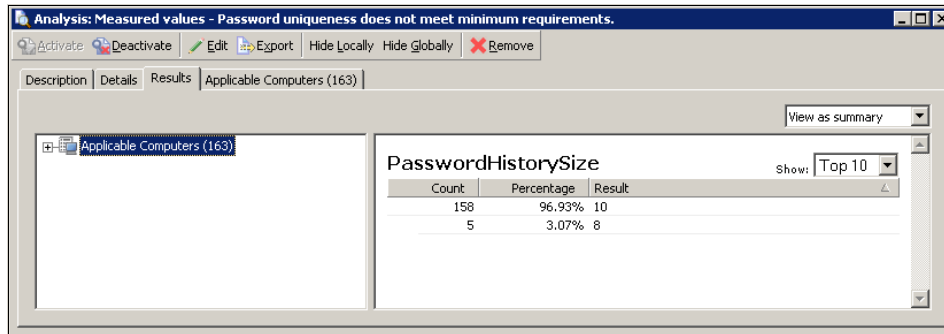


Figure 8-14 Measured values by the Analysis

The base Console is unable to show the logical link between the Fixlet and Analysis in an aggregated form. This functionality was introduced in the Security and Compliance Analytics reporting solution. It is presented in more detail in Chapter 9, “Phase IV: Security Compliance Analytics reporting” on page 357.

Use only what you need: Within each SCM Fixlet Site, IBM provides a set of Fixlets and correlated Analysis. Although Tivoli Endpoint Manager functionality allows many Analyses to be activated, it is suggested to use Analysis as an additional diagnosis tool. Operators must activate Analysis to further investigate noncompliance details, and when fixed, the Analysis must be deactivated. This approach prevents adding additional load on all platform components, starting with the Tivoli Endpoint Manager Agent, through your network, up to the Tivoli Endpoint Manager Relays and Tivoli Endpoint Manager Server.

UNIX Fixlets

The Tivoli Endpoint Manager platform uses the power of the *Relevance* language to gather information about various aspects of managed endpoints. Although with the Relevance language, you can address many requirements, differences exist among operating systems that are not fully supported by the Tivoli Endpoint Manager platform. At the time of writing this book, the Relevance language offers the best coverage for Windows systems. The Windows family is not large and differentiated, because the products derive many functionalities from a common source. For UNIX based operating systems, the situation is more complicated. There are many vendors and customized solutions. Often, there are differences even in the same components.

Fixlet authoring: For more information about documents related to the Relevance language and its features supported on various operating system platforms, see this website:

<https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Tivoli%20Endpoint%20Manager/page/Fixlet%20Authoring>

For the operator, this situation might not be convenient. The solution must provide a common way to manage both Windows systems and various UNIX based operating systems. Figure 8-15 depicts some of those differences.

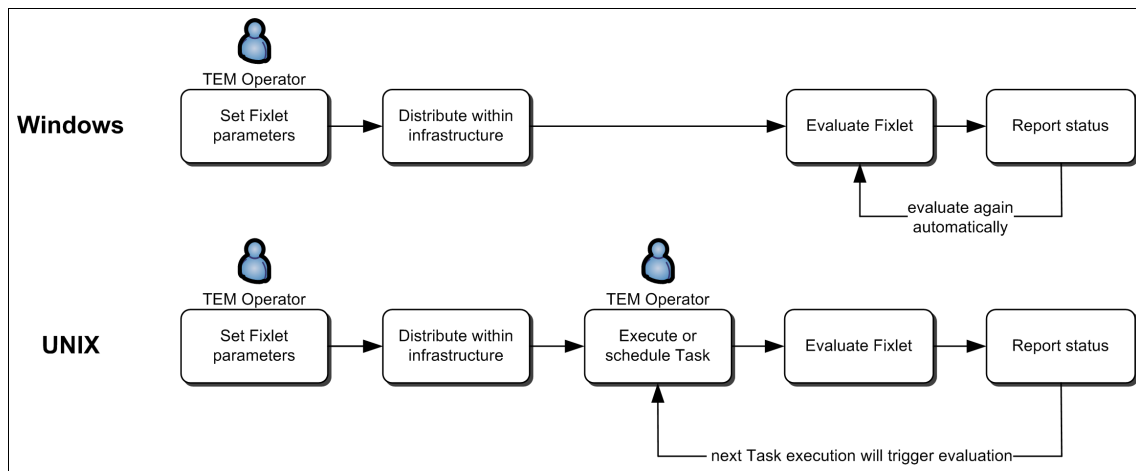


Figure 8-15 Windows and UNIX compliance evaluation flow

As we can see, both solution approaches use Fixlet Sites. Both approaches offer Fixlets to evaluate compliance state and measured value Analysis to gather more detailed information about the inspected aspect of the operating system. Both solutions offer self-parameterized Fixlets to allow Tivoli Endpoint Manager operators to update the Fixlet directly in its description and distribute to the endpoints. The main difference is at the time when the compliance evaluation takes place:

- ▶ Fixlets and Analysis dedicated to Windows systems use only the Relevance language. No matter what Actions are performed on the content (Site subscription or change of the parameters), a Fixlet is automatically distributed to an endpoint. And, the Fixlet is evaluated by the Tivoli Endpoint Manager Agent. The compliance information is instantly returned to the Tivoli Endpoint Manager Server. An operator can look at the resulting state directly in the Console.

- ▶ UNIX Fixlets are also distributed automatically to the endpoint, but it is the responsibility of the operator to enable the compliance evaluation. Within each UNIX Fixlet Site, there is a special Task that executes the compliance evaluation of all Fixlets available in the Fixlet Site. Every change to a Fixlet parameter or to the content of the Fixlet Site requires that the main Task is executed again. The execution can be started manually, or it can be scheduled with Tivoli Endpoint Manager platform-scheduling capabilities. After a Task is completed, Fixlets and Analysis report the compliance state and additional data back to the Tivoli Endpoint Manager Server, as in Windows systems.

SCM benchmark guides: For additional information for configuring and running security configuration management content for UNIX and Windows systems, see the following site:

<http://support.bigfix.com/resources.html#SCM>

8.2.4 Fixlet remediation

After Fixlets begin to report noncompliance, an operator needs to react according to predefined remediation actions.

For our scenario, we selected three policy controls. For these policies, we perform the entire lifecycle:

- ▶ Review noncompliance configuration on an endpoint, where the Fixlets (security checks) are true, based on the Relevance language.
- ▶ Check for noncompliance on Tivoli Endpoint Manager Console, focusing on the security configuration deviations that became “relevant”.
- ▶ Remediate the noncompliance policies by using predefined Actions, through the Tivoli Endpoint Manager Console.
- ▶ Finally, validate the security configuration update on the endpoint.

We can see these steps summarized in Figure 8-16 on page 325.

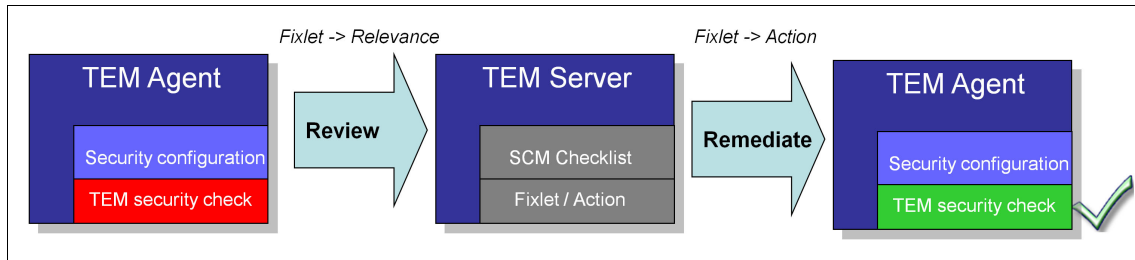


Figure 8-16 Remediation flow

Screen saver password policy

According to Table 8-9 on page 302, the *CIT001* check verifies whether the workstation screen saver is secured with a password. In this section, we explain the steps to fix this vulnerability when it is detected by a Fixlet.

Figure 8-17 shows a computer that is not protected by a password while the screen saver is visible.

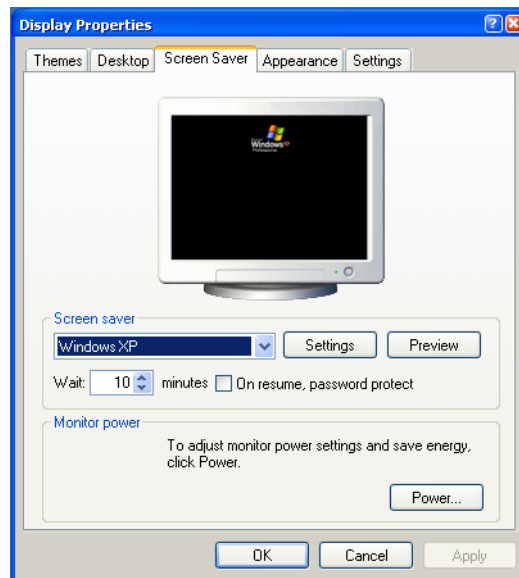


Figure 8-17 Workstation without a password-protected screen saver

Security risk: If a workstation owner walks away from a workstation for a while without a manual “lock” command, this machine becomes vulnerable for any malicious hacker that gains physical access to it. It is one of the most common security configuration issues for organizations worldwide.

When the SCM checks (Fixlets) are distributed to the endpoints, they evaluate the compliance posture by using the *Relevance* language. When a Fixlet detects a workstation as noncompliant, the workstation is reported to the Tivoli Endpoint Manager Server as “relevant”, and it needs attention from the operator. In this scenario, the machine, among others, is reporting to the Tivoli Endpoint Manager Console that it does not have the screen saver with password protection set, as shown in Figure 8-18.

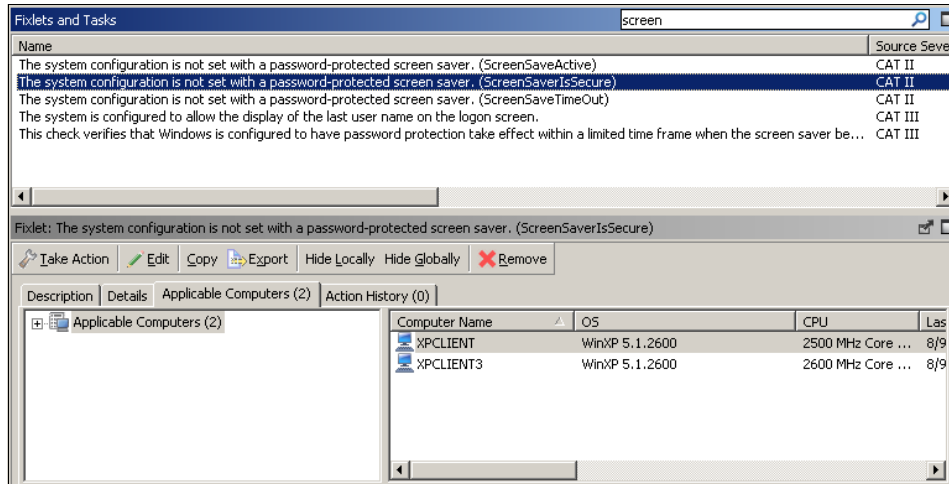


Figure 8-18 ScreenSaverIsSecure policy violation

Because this policy is one of the financial accounting company security policies that must be remediated, we must initiate an Action from the window that is shown in Figure 8-18.

Thus, implement the following steps from the TakeAction panel:

1. Click **Take Action**. In the following dialog, shown in Figure 8-19 on page 327, you need to select **XPCCLIENT3** as the Computer Name to be fixed.

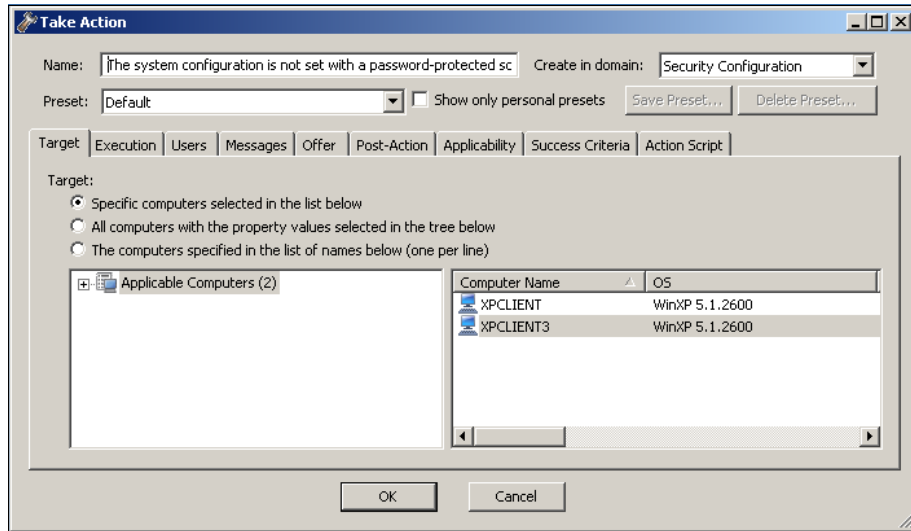


Figure 8-19 Remediation endpoint target

2. Navigate to the Execution tab to define the behavior of this Fixlet. The policy defines to run the Action automatically whenever the condition appears, or reappears. You must click **Reapply this action** and further select **whenever it becomes relevant again**, as shown in Figure 8-20. Finally, click **OK** to run the remediation.

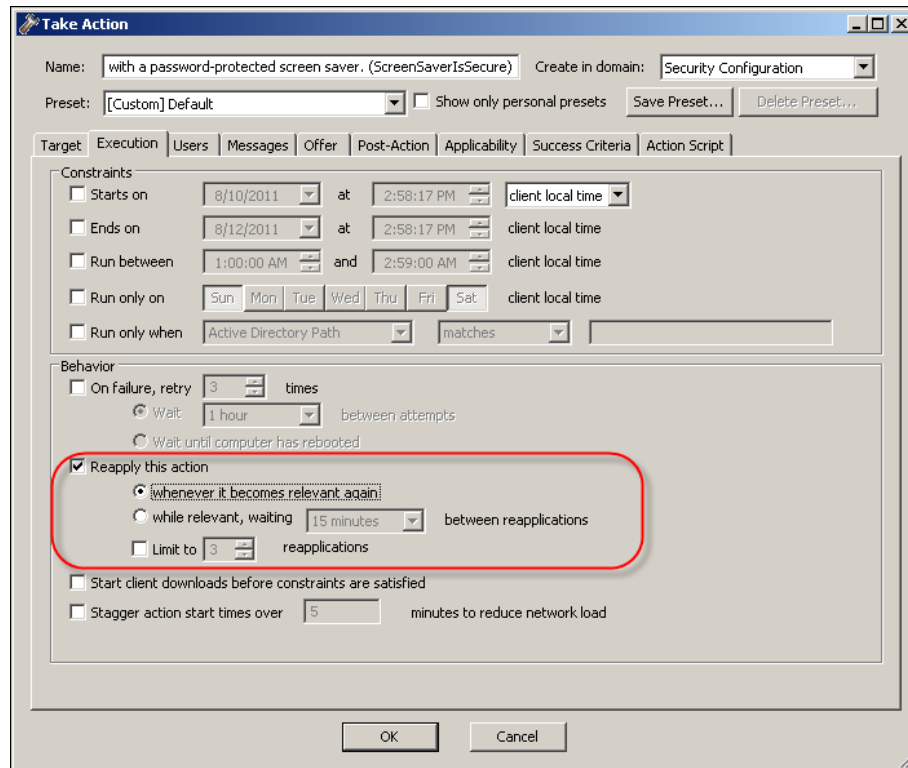
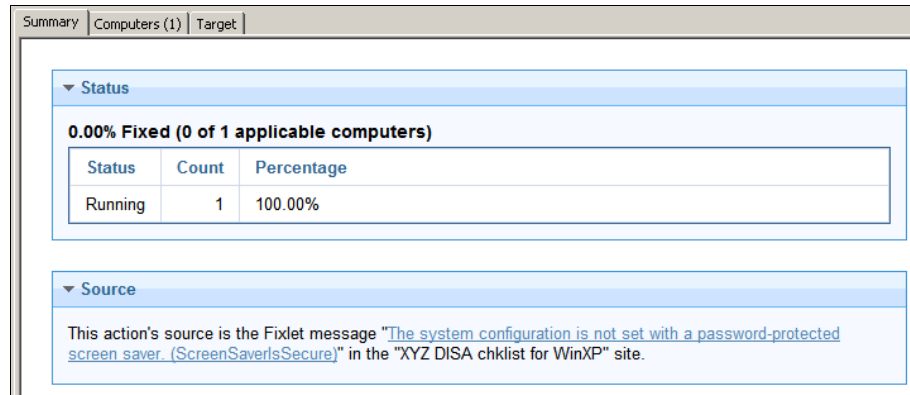


Figure 8-20 Defining the Execution behavior

Execution as a Policy: For any Action defined in the Execution tab, you can select “Policy” in the Preset drop-down list. An Action that becomes a Policy means that the Action is enforced whenever a change on a target machine violates the Policy. For a Policy, this evaluation and remediation are automatically executed by the Agent without any server or operator intervention. This evaluation and remediation are automatically executed, even if the target machine is disconnected from the financial accounting company network.

3. After you initiate the execution, an Action Information window is displayed. Here, you can view the status of the remediation, as shown in Figure 8-21.



The screenshot shows a window titled "Summary Computers (1) Target". It has two main sections: "Status" and "Source".

Status

0.00% Fixed (0 of 1 applicable computers)

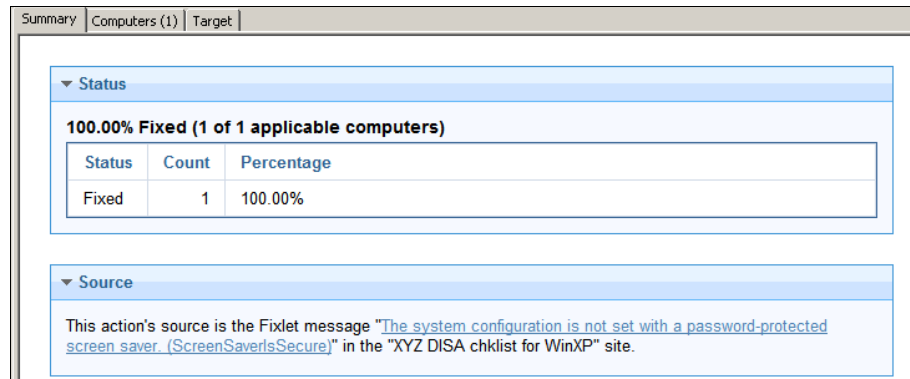
Status	Count	Percentage
Running	1	100.00%

Source

This action's source is the Fixlet message "[The system configuration is not set with a password-protected screen saver. \(ScreenSaverIsSecure\)](#)" in the "XYZ DISA chklist for WinXP" site.

Figure 8-21 Remediation running

4. After a while, the Status display changes and shows Fixed as the final message (Figure 8-22).



The screenshot shows a window titled "Summary Computers (1) Target". It has two main sections: "Status" and "Source".

Status

100.00% Fixed (1 of 1 applicable computers)

Status	Count	Percentage
Fixed	1	100.00%

Source

This action's source is the Fixlet message "[The system configuration is not set with a password-protected screen saver. \(ScreenSaverIsSecure\)](#)" in the "XYZ DISA chklist for WinXP" site.

Figure 8-22 Remediation fixed

5. If you look at workstation `xpclient3` again, you can see that it is protected by a password on the screen saver (Figure 8-23).

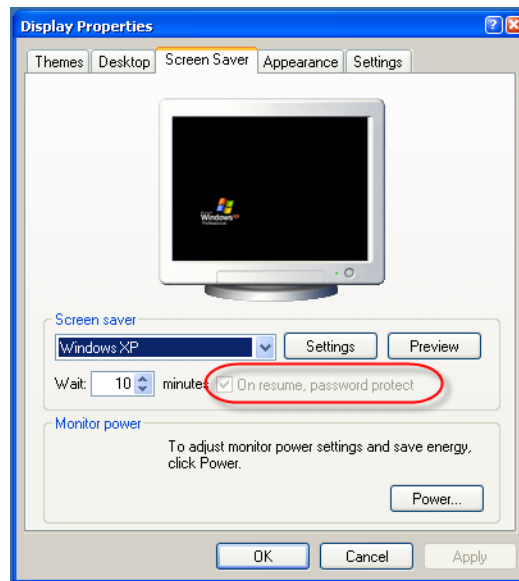


Figure 8-23 With a password-protected screen saver

Because this Fixlet was defined as “Reapply this action ... whenever it becomes relevant again”, either for a direct execution by an operator or by changing the execution preset to “Policy”, this configuration is reapplied. It is reapplied even if the workstation is not connected to the financial accounting company network.

Anonymous share must be disabled

According to the requirement *CIT101* in Table 8-11 on page 303, the system must verify whether anonymous share is enabled. In this section, we explain the steps to fix this vulnerability when it is detected by a Fixlet.

For the first part, it is necessary to check the current configuration of an endpoint for anonymous shares. Go to a Windows 2003 Server machine and navigate to **Start** → **Programs** → **Administrative Tools** and select **Local Security Policy**.

The Local Security Settings application opens. To visualize the current and default policy for anonymous shares, expand **Local Policies** and click **Security Options**. Then, scroll down the list of options until you see the policy “Network access: Shares that can be accessed anonymously”, as shown in Figure 8-24 on page 331.

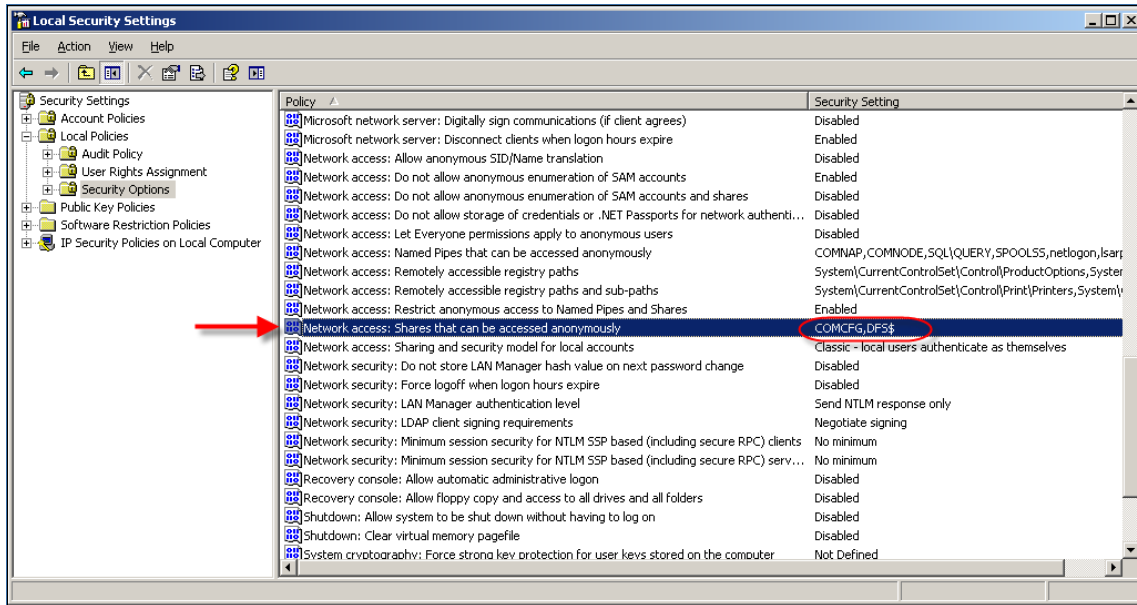


Figure 8-24 Default anonymous shares enabled

At the Tivoli Endpoint Manager Console, navigate to Fixlets and Tasks. There, you can see the appropriate Fixlet as *true*, or *relevant* (alarm), for this noncompliant security configuration (Figure 8-25).

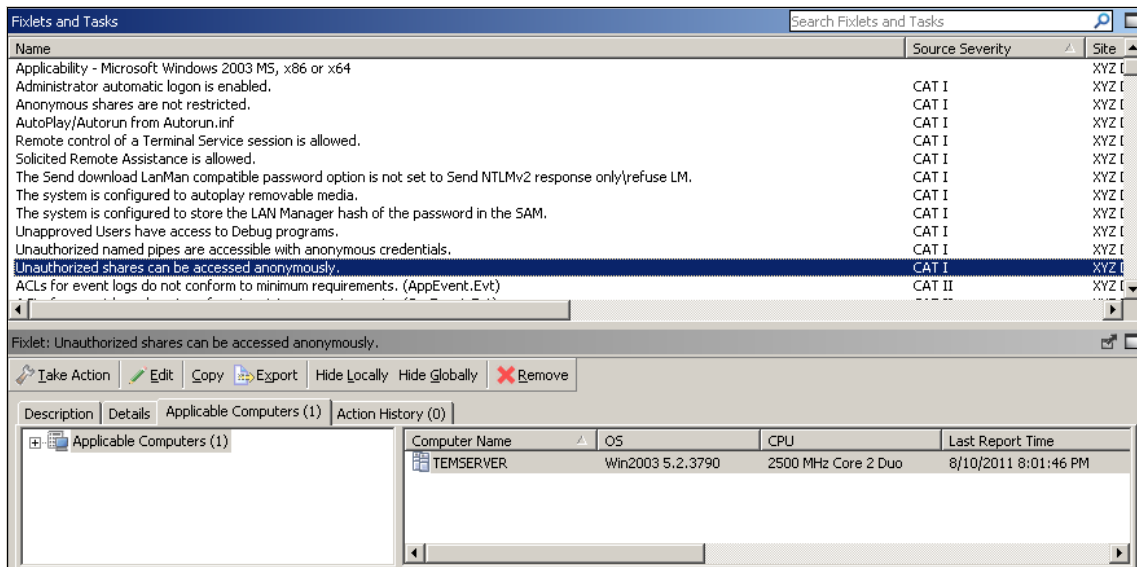


Figure 8-25 Anonymous shares Fixlet as true or relevant

To fix this condition, you need to implement the following steps (similar to “Screen saver password policy” on page 325):

1. In the Fixlets and Tasks dialog, click **Take Action**. In the following dialog, shown in Figure 8-26, you need to select **TEMSERVER** as the Computer Name to be fixed. When you click **OK**, the Action is executed.

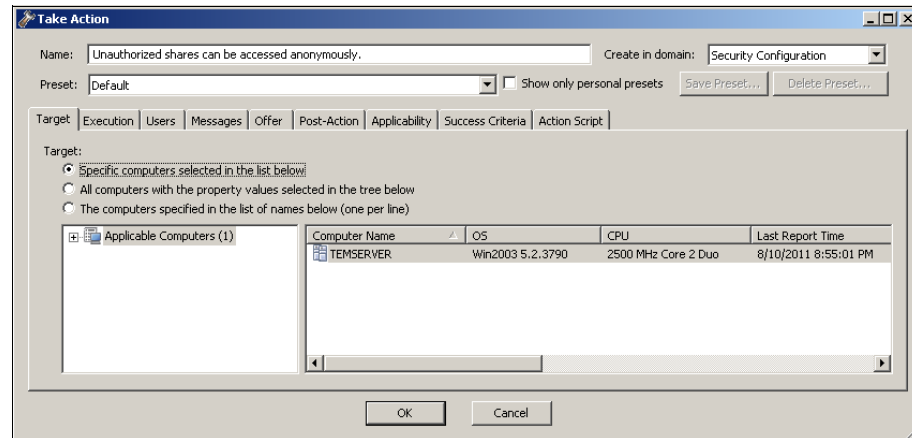


Figure 8-26 Take Action to remediate a security configuration issue

2. After the Action completes successfully, you can open the Local Security Policy application again on the endpoint. You can verify that no shares are enabled (Figure 8-27).

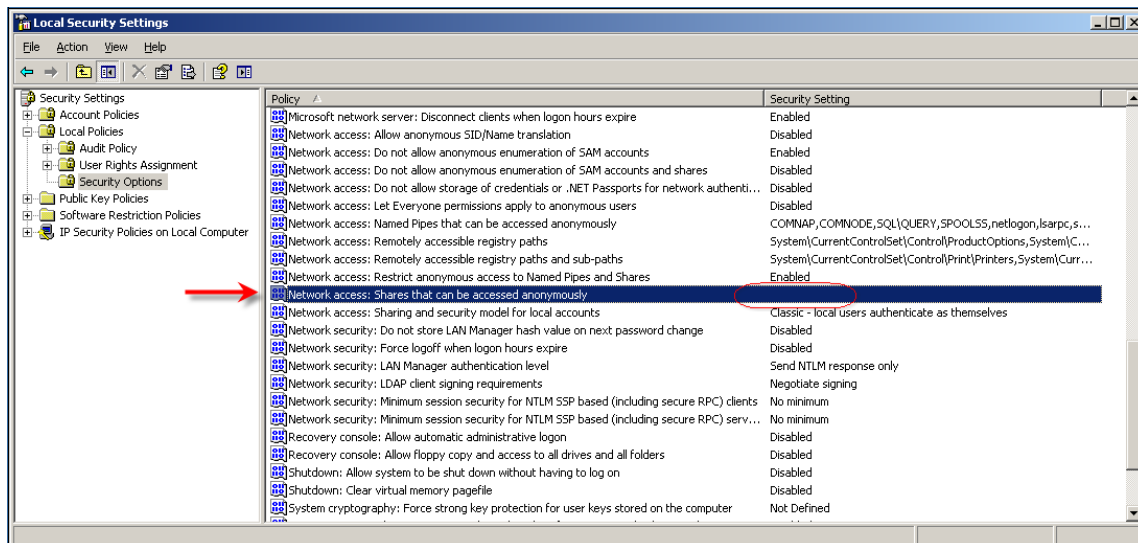


Figure 8-27 Anonymous shared disabled

Run this Action only one time: Because this Fixlet is targeted for a Windows Server, the financial accounting company leaves this Fixlet as the default Action behavior, which executes the Action one time. The company does not select a different Profile. The company selected this approach because this approach has more control over the server environment. Only Windows Server administrators can change any configuration settings in this environment.

Linux remediation

The third policy control to be remediated in our example is about a Linux RHEL 5 requirement. According to the requirement, which is *CIT306* in Table 8-14 on page 305, the password must be changed at least every 90 days on every Linux server system. In this section, we describe the necessary steps to fix the vulnerability detected by this Fixlet (security check).

As we described in “UNIX Fixlets” on page 322, Tivoli Endpoint Manager works differently for UNIX and Linux endpoints.

The Linux and UNIX evaluation executes a *Task* that runs each of the defined SCM controls in a batch, which differs distinctly from the real-time assessment used in a Windows environment. When the batch file runs, the results are evaluated on the desired endpoints. These results are logged and available to the corresponding Fixlet controls for evaluation. The Fixlets then use the *Relevance language* to examine the log and determine relevance. The results appear in the console, where compliance can be determined.

The financial accounting company deploys the Task called *Deploy and Run Security Checklist RedHat/CentOS 5*. It must be deployed before Tivoli Endpoint Manager can check the security configuration compliance on Linux and UNIX endpoints.

To execute this Task, select the Custom Site that hosts the Linux endpoint components. In this case, it can be either *CITSSE100 for RHEL 5 v1.0* or *DISA chklist for RHEL5*. Follow the steps:

1. Expand one of the Custom Sites and select **Fixlets and Tasks**.
2. Select the Task **Deploy and Run Security Checklist RedHat/CentOS 5**, as depicted in Figure 8-28 on page 334.

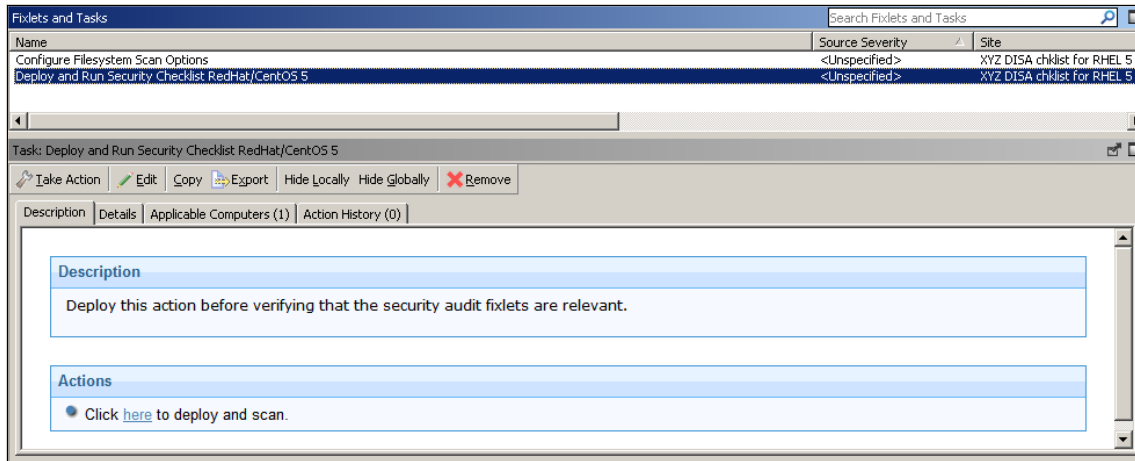


Figure 8-28 Linux main Task to deploy the checklist

3. Click **Take Action**.
4. In the Take Action dialog, select the Target endpoints by choosing the group membership **RHEL5**, as shown in Figure 8-29, and click **OK**.

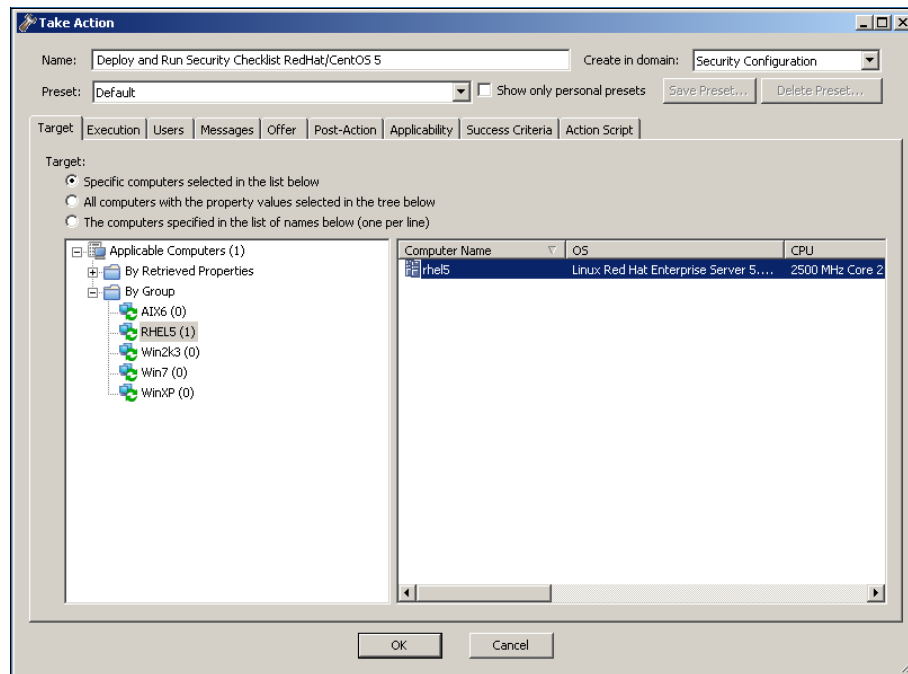


Figure 8-29 Linux target endpoint

This Task deploys the main SCM script called `runme.sh`, which prepares the Linux systems to be analyzed by Fixlets. As soon as each Fixlet becomes true, it appears on Tivoli Endpoint Manager Console as relevant.

- By listing the `/etc/shadow` file on your Linux host, shown in Figure 8-30, you can check that the maximum age is not set. The information is stored on field “5” for each user ID. At this time, there are only two users with the appropriate configuration set: `virtuser` and `operation`.

```

root@rhel5:/etc
File Edit View Terminal Tabs Help
virtuser:$1$sCsc3nEp$bjqai0Li0cRchl6ptRnm/:15202:0:90:7:::
operation:$1$0BLn.brK$gIqlPf1k2kEbEbcL2q2Md0:15202:0:90:7:::
paul:$1$feU3cL1w$aqU04QYVgcM99YX97HqBW/:15203:0:99999:7:::
michael:$1$wo5zRw09$2efLTyR8YPrsH4qZ4Mjdp0:15203:0:99999:7:::
john:$1$WrEbHWk0$S72Xp0ImtQ4z452SN.XUR.:15203:0:99999:7:::
mary:$1$xDVdIbPX$gnmd01hUtLL/8uL10CEDb1:15203:0:99999:7:::
jack:$1$L3vUyopy$tQQLfiwdefM9zQZlhkNHp1:15203:0:99999:7:::
smith:$1$Xav0Duul$EC/x9csZSDtFeuspwHrJB0:15203:0:99999:7:::
juca:$1$FX2Nh7Rq$93IoujHxqYFY4k.fvrtpv0:15203:0:99999:7:::
nilton:$1$TxfXtH0g$yvpmiUfbuXiTYpxzI05lA1:15203:0:99999:7:::
carl:$1$n.L3m6.J$YQrancB.Kicqtu99ad5kb/:15203:0:99999:7:::
barne:$1$2d4dr2e1$WKRPKPs9UuirRN9N0cbl.d.:15203:0:99999:7:::
mitiko:$1$d.aMl12Q$svI0vja2eDXK5aGZ3GIAuc1:15203:0:99999:7:::
[root@rhel5 etc]#

```

Figure 8-30 Linux account information on `/etc/shadow` file

In the Console, you can see the relevant Fixlet for this server, as shown in Figure 8-31.

Name	Source Severity
Library File Permissions - RedHat/CentOS 5	CAT II
Local Initialization Files Permissions - RedHat/CentOS 5	CAT II
Login Delay - RedHat/CentOS 5	CAT II
Maximum Password Age - RedHat/CentOS 5	CAT II
Minimum Password Age - RedHat/CentOS 5	CAT II
Minimum Password Age - RedHat/CentOS 5	CAT II
Minimum Password Length - RedHat/CentOS 5	CAT II
Password Complexity - alphabetic characters - RedHat/CentOS 5	CAT II
Password Complexity - alphabetic characters - RedHat/CentOS 5	CAT II
Password Complexity - numeric characters - RedHat/CentOS 5	CAT II
Password Complexity - special characters - RedHat/CentOS 5	CAT II
Remote consoles - RedHat/CentOS 5	CAT II
Required Network Services For Operation - RedHat/CentOS 5	CAT II
Reserved System Account GIDs - RedHat/CentOS 5	CAT II
Reserved System Account UIDs - RedHat/CentOS 5	CAT II
Root access encryption via ssh - RedHat/CentOS 5	CAT II
Root access encryption via ssh - RedHat/CentOS 5	CAT II
Root home directory permissions - RedHat/CentOS 5	CAT II
Sendmail Help Command - RedHat/CentOS 5	CAT II
Shells Permissions - RedHat/CentOS 5	CAT II
Single User Mode Password - RedHat/CentOS 5	CAT II
Support for .rhosts in PAM - RedHat/CentOS 5	CAT II

Figure 8-31 Tivoli Endpoint Manager Console: Linux maximum age password Fixlet

6. As with the previous Fixlets, to remediate this issue, you must navigate to the Take Action dialog and choose the targeted endpoints for this remediation.
7. After you receive the status “Fixed” in your Console, you can navigate to a Linux console and check the results in your /etc/passwd file. The result is shown in Figure 8-32.

```

root@rhel5:/etc
File Edit View Terminal Tabs Help
virtuser:$1$sCsc3nEp$bjqai0Li0cRcLh16ptRnm/:15202:0:90:7:::
operation:$1$soBLn.brK$gIqLpF1k2KEbEbC12q2Md0:15202:0:90:7:::
paul:$1$feU3cLlw$aqU04QYVgcM99YX97HqBW/:15203:0:90:7:::
michael:$1$wo5zRw09$2efLTyR8YPrsH4qZ4Mjdp0:15203:0:90:7:::
john:$1$WrEbHwK0$S72Xp0ImtQ4z4525N.XUR.:15203:0:90:7:::
mary:$1$xDVdIbPX$gnmd01hUtLL/8uL10CEdb1:15203:0:90:7:::
jack:$1$L3vUyopy$tQQLfiwdefM9zQZLhkNHp1:15203:0:90:7:::
smith:$1$XavQDuul$EC/x9csZSDtfEuspwhRJB0:15203:0:90:7:::
juca:$1$FX2Nh7Rq$93IoujHxqYFY4k.fvrtpv0:15203:0:90:7:::
nilton:$1$TxfXtH0g$yvpmiUfbuXiTYpxzIOs1A1:15203:0:90:7:::
carl:$1$N.L3m6.J$YQrancB.Kicqtu99ad5kb/:15203:0:90:7:::
barne:$1$2d4dr2e1$WkRPKPs9UuirRN9N0cbld.:15203:0:90:7:::
mitiko:$1$d.aMlL2QsvI0vja2eDXK5aGZ3GIAUc1:15203:0:90:7:::
[root@rhel5 etc]#

```

Figure 8-32 Linux with password age set

8.2.5 Practice

We described several preferred practices for an SCM design in 4.4, “Security configuration management solution design” on page 165. To complement these practices, we show several examples related to implementing an SCM solution that were used by the financial accounting company:

- ▶ Before enabling external Sites, validate whether there is any organization that makes available the content for all the platforms that you plan to manage. The financial accounting company uses content provided by DISA, because it can greatly help ease the management process.
- ▶ It is not possible to update control values directly on external Sites. Therefore, create at least one Custom Site for each managed platform and clone the related controls required for your organization from the external Site to the Custom Site. The financial accounting company created two Custom Sites for each platform, due to the top/down and bottom/up strategies. Depending on your situation, you might have a different Custom Site requirement. For example, you might need different policies for different countries, departments, or environments (such as test, quality assurance (QA), and production).
- ▶ Follow a naming convention to create Custom Sites that makes sense for your organization. This practice can better manage your environment. Document

this naming convention. Ensure that the operators who can create objects in Tivoli Endpoint Manager follow the naming convention.

- ▶ Use machine groups to manage Site membership rather than connecting machines directly to Sites. This approach improves the management process and can help maintain a more consistent Tivoli Endpoint Manager environment.
- ▶ The most common approach to enforce security controls for your endpoints is to first evaluate an actual security posture within your IT environment. Then, map your findings and requirements to an existing FDCC or DISA standard. For example, a top/down approach is the preferred approach for organizations with well-documented policies. For organizations without properly documented policies, a bottom/up approach is the only option. For a bottom/up approach, agree on a due date for the necessary controls so that you can have a policy and the appropriate controls implemented, documented, and monitored as soon as possible.
- ▶ Avoid merging checklists into one Custom Site. This approach can influence the administration and maintenance of the environment in future releases.

8.3 Project scope change

Often, new requirements are identified during a project implementation. Some of those requirements might gain priority over other activities that were planned at the beginning of the project. It is the responsibility of the project manager to redefine and negotiate this project scope change.

In this section, we do not intend to describe how to negotiate or develop a project change meeting. We show a scenario where the Tivoli Endpoint Manager architecture provides enough flexibility to add additional tasks to the project without significantly affecting the project timeline. The scope change for the financial accounting company occurred because of a recent incident related to its industry.

8.3.1 Monitoring anti-virus health

The financial accounting company was required to submit to a critical and recent Payment Card Industry - Data Security Standard (PCI-DSS) auditing process. The IT governance group was unable to provide sufficient information about PCI-DSS³ Requirement 5.

³ For more information, see the PCI document library:
https://www.pcisecuritystandards.org/security_standards/documents.php

PCI-DSS Requirement 5: Requirement 5 is part of the section to “Maintain a Vulnerability Management Program”. In particular, the requirement dictates that an organization must “use and regularly update anti-virus software or programs”.

Based on the outcome of the audit, the IT management and governance group asked to include a compliance report about the anti-virus deployment and security posture for the first iteration of the project.

Flexibility: The main objective of this project scope change is to demonstrate how Tivoli Endpoint Manager provides a flexible and fast way to add new functionalities. It is not necessary to install additional modules or components. We enable a new external Site from any licensed module, and with a few customizations, the new report goes into production.

Enabling Client Manager for Endpoint Protection external Site

Similar to the steps in “Enabling SCM external Sites” on page 306, the first step is to subscribe to an external Site. We need to gather Site content from an IBM Fixlet server. Follow the next steps:

1. In the Console domain selector, choose **BigFix Management** → **License Overview** to display an overview of the Available Sites for Security and Compliance. Click **Enable** for the *Client Manager for Endpoint Protection* Site in the List Panel on the license dashboard. This action adds the Site to the Enabled Sites, as shown in Figure 8-33 on page 339.

▼ Security and Compliance

Expiration Date: 12/30/2021

Product License Counts:

Enabled Sites	Subscribed Computers
BFS Asset Discovery	0
Client Manager for Endpoint Protection	3
DISA STIG on Windows 2003 DC v6r1.18	0
DISA STIG on Windows 2003 MS v6r1.18	0
DISA STIG on Windows 7 v1r2	0
DISA STIG on Windows XP v6r1.18	0
DISA STIG Checklist for AIX 6.1	0
DISA STIG Checklist for RHEL 5	0
SCM Checklist for FDCC on Windows XP	0
SCM Reporting	0
Security Policy Manager	2
USGCB Checklist for Internet Explorer 8	0

Computers subscribed to any of the above sites use up a license for Security and Compliance.

Available Sites:

- [\[Enable\]](#) BigFix Client Compliance (IPSec Framework)
- [\[Enable\]](#) BigFix Client Compliance Configuration
- [\[Enable\]](#) Device Management for Windows Mobile
- [\[Enable\]](#) DISA STIG on Windows 2008 DC v6r1.11
- [\[Enable\]](#) DISA STIG on Windows 2008 MS v6r1.11
- [\[Enable\]](#) DISA STIG on Windows Vista v6r1.18
- [\[Enable\]](#) Linux RPM Patching
- [\[Enable\]](#) Patches for AIX
- [\[Enable\]](#) Patches for ESX3

Figure 8-33 Enabling the Client Manager for Endpoint Protection external Site

- Now, we can see a new domain in the Tivoli Endpoint Manager domain panel called *Endpoint Protection* (Figure 8-34).

The screenshot shows a vertical list of domain icons and labels. From top to bottom, they are: 'All Content' (blue circle icon), 'BigFix Management' (blue globe icon), 'Endpoint Protection' (blue shield icon, highlighted with a red oval), 'Security Configuration' (blue gear icon), and 'Systems Lifecycle' (blue gear icon).

Figure 8-34 Tivoli Endpoint Manager domains

- Click **Endpoint Protection** to visualize and configure the external Site “Client Manager for Endpoint Protection”.

Important: The Tivoli Endpoint Manager for Core Protection offering, based on the Trend Micro solution, is associated with the Endpoint Protection domain, as well. However, the Client Manager for Endpoint Protection external Site is part of the Tivoli Endpoint Manager for Security and Compliance offering.

Subscribing the endpoints to be monitored

The operation of the Client Manager for Endpoint Protection external Site is similar to other Sites that we described in this book. It consists of Fixlets, Tasks, Analysis, Dashboards, and wizards.

To prepare this Site for our case, we execute the following steps:

1. Inside the Endpoint Protection domain, in the navigation tree, select **Sites** → **External Sites**. Then, navigate to the **Computer Subscription** tab.
2. Select the endpoints to be part of this Site.

Supported vendors: The anti-virus names that are shown in the Console reflect several of the supported vendors. There is no relation between the vendor name and the usage with the financial accounting company. We used several vendors to be vendor-agnostic.

3. Activate the Analysis, based on the anti-virus environment that is shown in Figure 8-35.

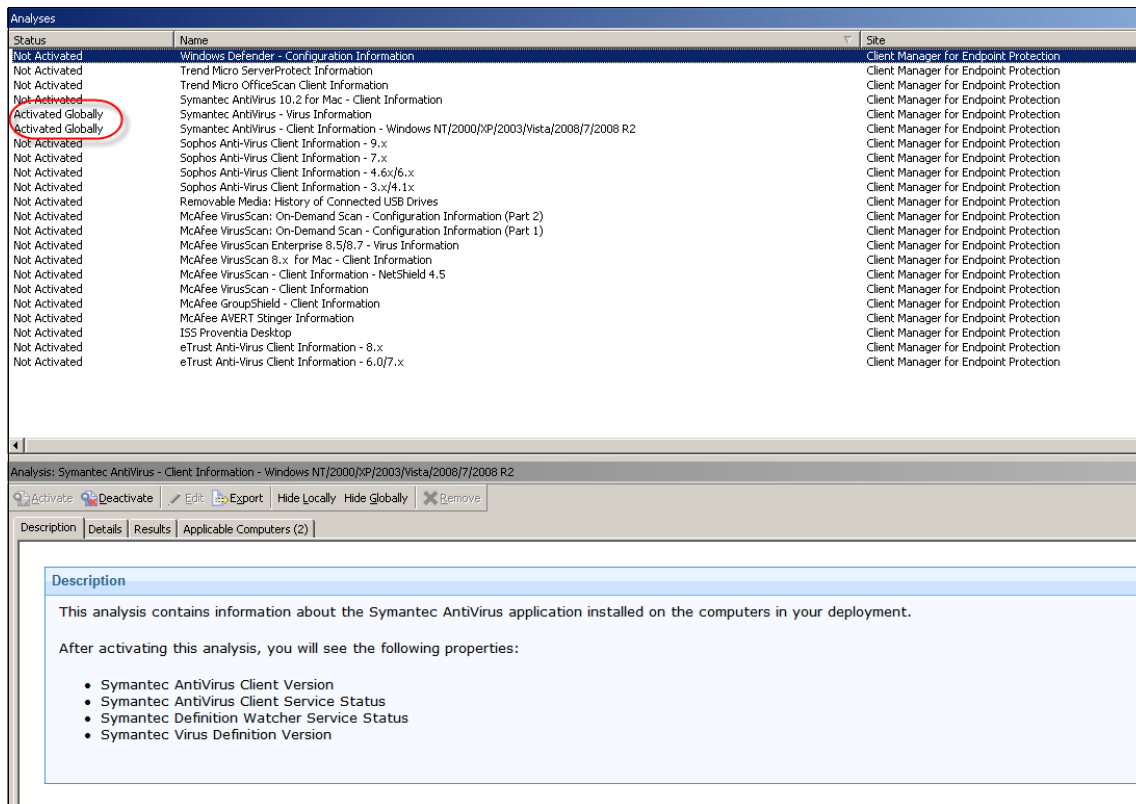


Figure 8-35 Client Manager for Endpoint Protection Analysis

Monitoring the anti-virus compliance

After you enable Client Manager for Endpoint Protection, subscribe the endpoints, and run the Analysis, you can view the status of the anti-virus deployment and security posture in the financial accounting company. You can then provide reports to the auditors to check on the current level of PCI-DSS compliance.

Anti-virus administration: The objective of this book is to present Client Manager for Endpoint Protection compliance reports. It is not our intention to show how to manage anti-virus deployment from Tivoli Endpoint Manager.

To access the real-time information related to anti-virus deployment (PCI-DSS Requirement 5.1) and information related to the health posture of the anti-virus

deployment (PCI-DSS Requirement 5.2), it is necessary to access the report *Client Manager for Endpoint Protection (CMEP) Overview*, as depicted in Figure 8-36.

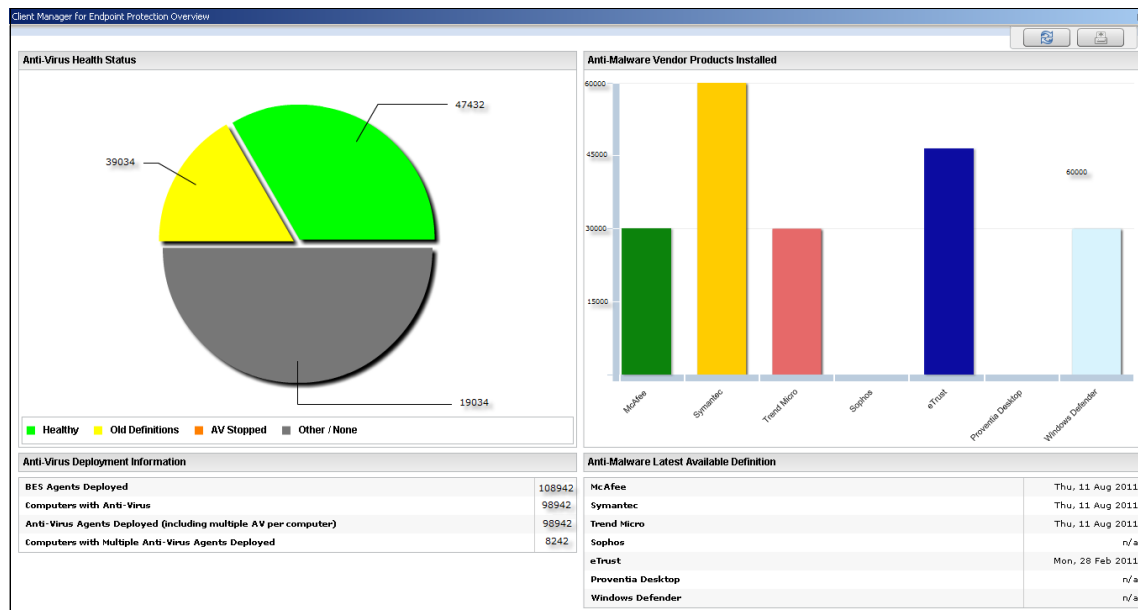


Figure 8-36 CMEP Antivirus Overview report

The *CMEP Overview* report provides a summary of the anti-virus health posture in the financial accounting company deployment. The left side of the overview window contains the anti-virus health status pie chart and anti-virus deployment information statistics. The right side contains the anti-virus deployment numbers graph by vendor, and the dates of the latest available anti-virus definition for each vendor product.

The anti-virus status information is the most important information that is provided, as shown in Table 8-16.

Table 8-16 Anti-virus status definition

Status	Definition
Health	This machine is adequately protected and up to date.
Old Definition	The virus definition needs to be updated.
AV stopped	The anti-virus application or service is not running.
Other/None	This machine uses an unsupported anti-virus, or no anti-virus is installed.

We must focus on any systems with a status other than “Health”. In our case, we discovered many systems with an “Old Definition”, “AV stopped”, or “Other/None” status. These statuses leave the financial accounting company in a noncompliance position. These statuses characterize a severe issue with the security control and process of the company, which must be investigated and remediated.

You initiate the investigation by clicking inside the graphic pie chart, where the status shows a problem. You can drill down and check the problematic machines. Next, you must prioritize which issue to address first.

Other/None

This anti-virus status does not necessarily imply a problem. Some operating systems do not use anti-virus, such as Linux and AIX. These machines are probably categorized with this status.

Thus, our suggestion is to look at the endpoints with a status of “Other/None”. You can filter the machines that must be excluded as “problematic” (Linux and AIX), and you can create a better view of the number of machines that actually pose a problem.

Deploy anti-virus

For machines that reported an anti-virus software that is not officially supported by the IT organization, use Tivoli Endpoint Manager to uninstall the discovered anti-virus software. Then, reinstall a supported version by using the Windows Software Distribution Wizard, which is part of the Tivoli Endpoint Manager for Lifecycle solution.

Anti-virus stopped and old definitions

Use the Console to fix problems on systems that run the Tivoli Endpoint Manager Agent.

If the Agent is not running on the endpoint, you can execute a Fixlet related to the anti-virus problem. Within the Endpoint Protection domain, navigate to the **Troubleshooting** → **Client Not Running** folder, as shown in Figure 8-37 on page 344.



Figure 8-37 Managing anti-virus stopped

For problems related to Old Definition files, you must use the Fixlets within the Manage Definition Updates folder, as shown in Figure 8-38.

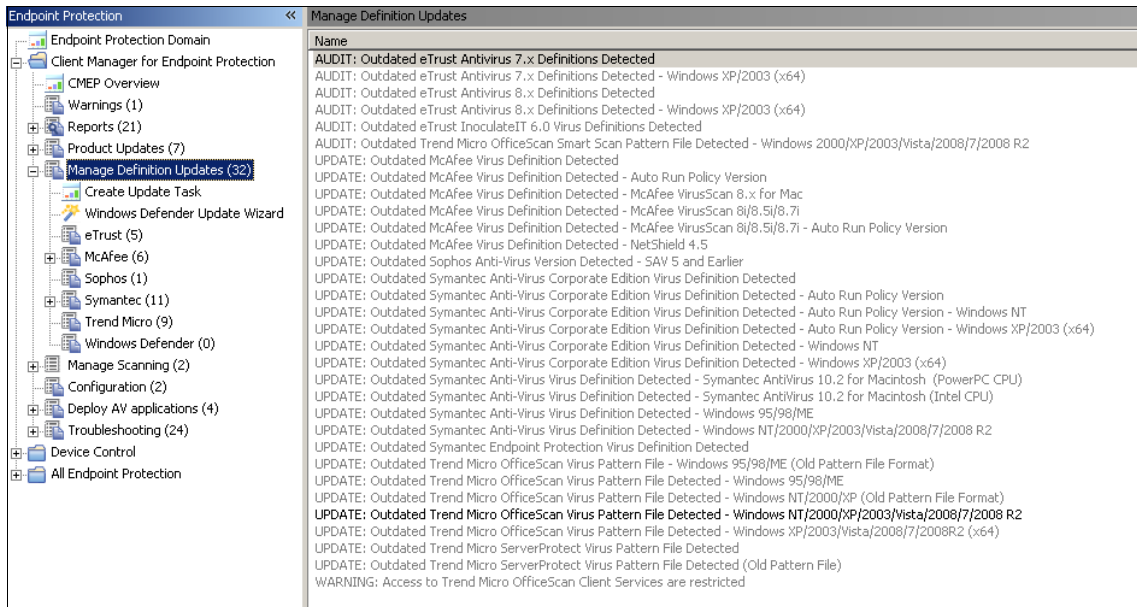


Figure 8-38 Managing old definition files

Report by vendor: You can also select individual vendors to display a customized pie chart and summary, which can help prioritize the correction options for a vendor. For example, you can select to view only the Trend Micro report. The dashboard displays the Trend Micro health status pie chart, the date of the latest definition release, and a list of related Analyses with either Activated or Not Activated status.

8.4 SCM administration and maintenance

One concern related to IT projects is how to keep the entire ecosystem (servers, applications, databases, processes, and networks) safe and up to date. Another concern is how to provide continuous improvements to stay aligned with current business requirements and future changes in those requirements.

These requirements do not differ in an endpoint management ecosystem. Sometimes, they become more intense, because this ecosystem is no longer confined to the premises of an organization but can be distributed worldwide.

To mitigate these inherent risks, we defined an endpoint management lifecycle guide for the financial accounting company administrators and users, after completing the initial professional services deployment project. The proposed lifecycle flow is shown on Figure 8-39 on page 346.

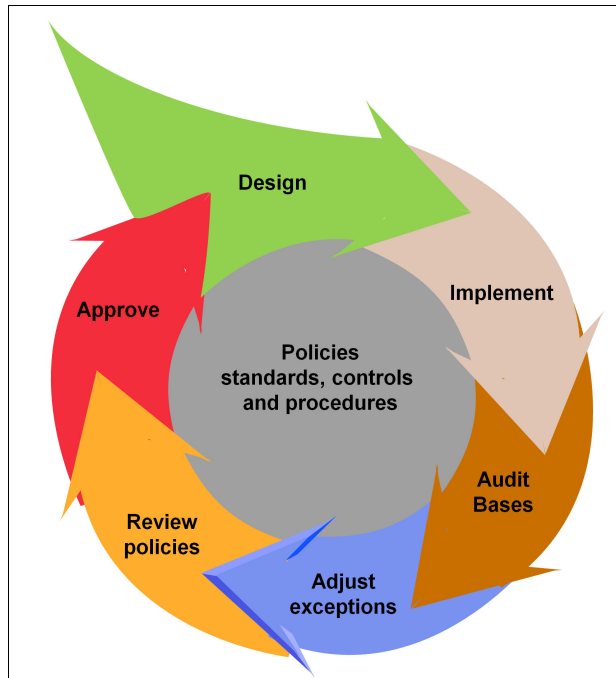


Figure 8-39 Endpoint management lifecycle

Every administration cycle in this flow is governed by the central circle in the diagram, named *Policies, standards, controls, and procedures*. Policies, standards, controls, and procedures must be well-documented at the beginning for the project design. You also might include the Business Requirements and Functional Requirements. However, from our standpoint, these requirements are external factors that connect with this flow from a peripheral aspect only.

This flow is completed one time during the professional services deployment. Then, the flow is maintained by the financial accounting company teams.

This lifecycle represents a continuous improvement process, because IT, business, and functional requirements change frequently. After the cycle completes for the first time, there are no more start or conclusion points. In the following sections, we describe all lifecycle steps:

► Design

After the corporate policy and the standards of the organization are defined or confirmed, it is necessary to map the security controls for the policy and standards to the Tivoli Endpoint Manager checklists. If the controls are not available on any current checklist used for production, it is necessary to validate whether the controls are available in any of the other checklists.

Always verify whether there are available updates by accessing the Security Configuration Domain dashboard. Look at the newsletters called *Tivoli Endpoint Manager Mailing Lists*⁴, where you can check for new content available for Tivoli Endpoint Manager.

► Implement

The financial accounting company might use more than one Custom Site for a specific purpose. However, it is important to have only one Custom Site in production at a time. To manage the Custom Sites, we suggest “versioning” as part of the naming convention, to differentiate what is in production at a certain time. When a new checklist version is ready, it can be moved to production and the old Custom Site (checklist) is removed. If necessary, you can keep both checklists in production for a few days to ensure that all actions run well. This approach can use more resources on endpoints.

► Audit Bases

It is important for the administrators or operators to always monitor and generate alerts and auditing information for SCM reports. When production operations begin, you want to report on compliance postures, execute security control, and identify what endpoints must be remediated based on priorities and risks. Later, we suggest that you use real-time reports for operations, such as to validate correct security controls, and use history reports for auditing, compliance, and management reporting.

► Adjust exceptions

Any management tool that generates many false positives can cause monitoring problems and can mask critical problems. *“If everything appears critical, how do we know what must be addressed first?”* To avoid this situation, you must adjust the exceptions configuration from time to time. We suggest this adjustment on a bimonthly base, or when the average percentage of noncompliant endpoints increases abnormally.

► Review policies

This cycle is critical to keep the endpoint management environment aggregating value to the organization and aligned with the business and functional requirements. We know that organizations are living organisms, and the business requirements can change based on several variables. The policies and standards to follow these requirements are constantly updated, as well. So, it is critical to have a process to analyze which security policy and standards changed since the last Tivoli Endpoint Manager implemented checklist. It is critical to see which updates on Tivoli Endpoint Manager are necessary to align the solution to those security policies and standards.

► Approve

⁴ You can access the Tivoli Endpoint Manager Product Mailing Lists at this website:
<http://www.ibm.com/support/docview.wss?uid=swg21505679>

After the policies revision is concluded, the next step is a formal process to approve those changes and to define a timeline to apply them to the Tivoli Endpoint Manager environment. After the changes are approved, the policies must be documented as security standards. For the financial accounting company, most standards are documented on the financial accounting company corporate security standard (CITSSE100). The standards are sent to the *Design* cycle to define how they must be implemented by using Tivoli Endpoint Manager.

8.5 Real-time reports

You can use the real-time reports feature with the Security and Compliance Analytics component, which is a historical information tool, to provide powerful reporting capability to Tivoli Endpoint Manager. The real-time reports feature is provided by *Tivoli Endpoint Manager Web Reports Server*, as introduced in 3.1, “Logical component overview” on page 64.

8.5.1 Tivoli Endpoint Manager real-time reports requirements

Tivoli Endpoint Manager Web Reports accesses the Tivoli Endpoint Manager database to obtain SCM information to provide more operational reports. The requirements for the first project phase relate to the automation of report distribution. In the second project phase, the financial accounting company designs other reports to gather additional information from the environment.

The real-time reports feature has the following requirements:

- ▶ Enable the administrators, operators, and auditors to access the Web Reports Console, as described in Table 8-17.

Table 8-17 Web Reports access

Name	Logon	Role
John Miller	jmillier	Administrator
Mark Evans	mevans	Operator
Monica	mparlangelo	Auditor
Mary Lord	mlord	Auditor
Patrick James	pjames	Operator

- ▶ Send the following reports by email:
 - Computer Compliance Summary for Windows 2003
 - Computer Compliance Summary for Windows XP
 - Analysis List
- ▶ Distribute the reports to key Tivoli Endpoint Manager operators and administrators, as shown in Table 8-18.

Table 8-18 Report distribution

Report name	Frequency	Receiver
Computer Compliance Summary for Windows XP	Daily	Mark Evans
Computer Compliance Summary for Windows 2003	Daily	Patrick James
Analysis List	Daily	John Miller

8.5.2 Tivoli Endpoint Manager real-time reports basic configuration

For the real-time report configuration, we use the following steps:

- ▶ “Enabling access to Web Reports Console”
- ▶ “Reports to distribute by email” on page 350
- ▶ “Distribute selected reports” on page 352

Enabling access to Web Reports Console

The first step is to create the users who access the Web Reports Console and then assign access roles for them.

By using the Web Reports Administration page, you can create new users, as shown in Figure 8-40. You need to specify the full name of the person, a user logon name, and password, and assign a role.

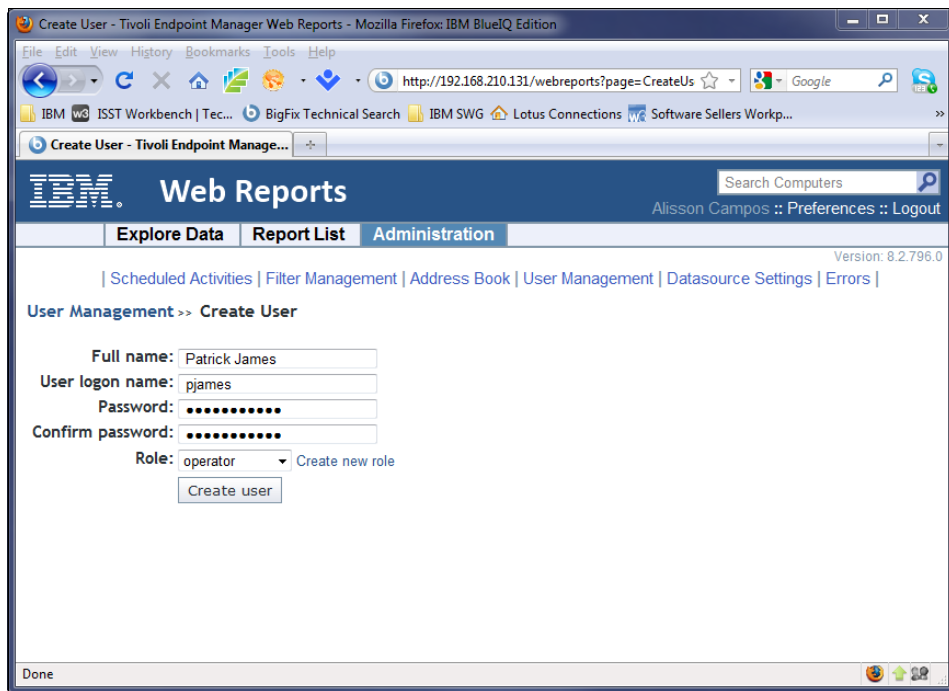


Figure 8-40 Web Reports Administration

Reports to distribute by email

Next, you must configure the environment to distribute the required reports. Use the following steps:

1. Use the Web Reports Console. Configure the email server by clicking **Administration** → **Address Book** → **Email Server settings**. Type the email Simple Mail Transfer Protocol (SMTP) server name and click **Test**. You receive an email, shown in Figure 8-41 on page 351, to confirm the configuration was correct.

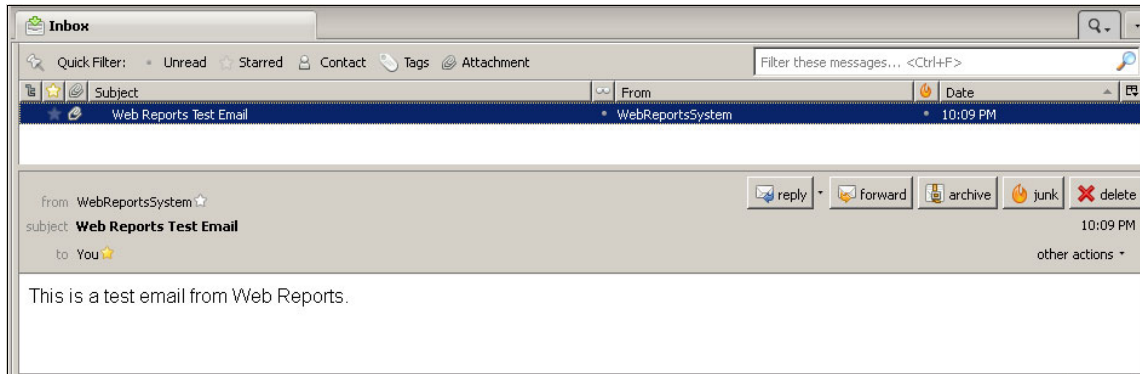


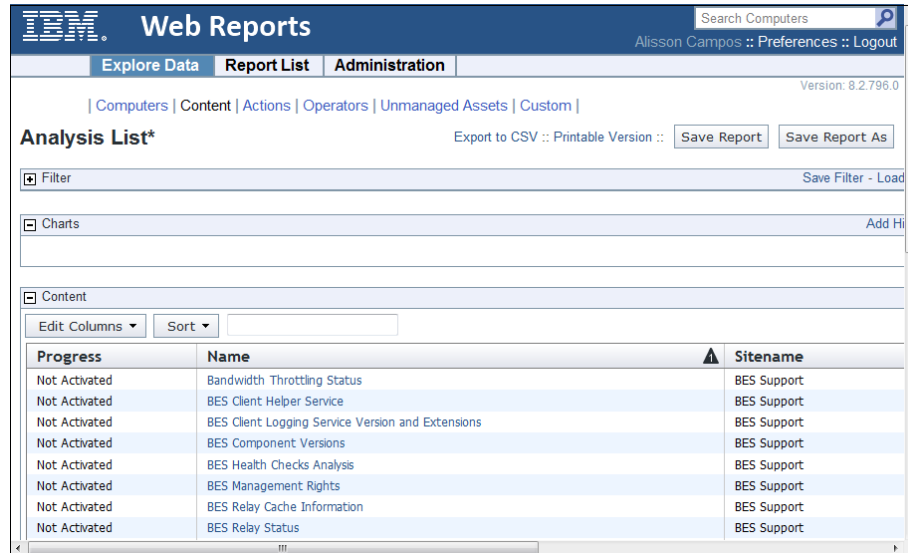
Figure 8-41 Email test to configure server to receive report

2. Back in the Web Reports Console, click **Save**.
3. Add the contact information for the people to receive the email by clicking **Add contact** on the same Administration menu. Then, provide the email addresses.

Distribute selected reports

For the financial accounting company, we distribute three reports daily:

1. The *Analysis List* report must be distributed to the administrator, John Miller, to check the key Analysis status for the Tivoli Endpoint Manager environment, as depicted in Figure 8-42.



The screenshot shows the IBM Web Reports interface. The main heading is "Web Reports" with a search bar for "Search Computers" and user information "Alisson Campos :: Preferences :: Logout". The navigation menu includes "Explore Data", "Report List", and "Administration". The current report is "Analysis List*", with options for "Export to CSV :: Printable Version :: Save Report" and "Save Report As". There are sections for "Filter", "Charts", and "Content". The "Content" section includes "Edit Columns" and "Sort" options. The main table has three columns: "Progress", "Name", and "Sitename".

Progress	Name	Sitename
Not Activated	Bandwidth Throttling Status	BES Support
Not Activated	BES Client Helper Service	BES Support
Not Activated	BES Client Logging Service Version and Extensions	BES Support
Not Activated	BES Component Versions	BES Support
Not Activated	BES Health Checks Analysis	BES Support
Not Activated	BES Management Rights	BES Support
Not Activated	BES Relay Cache Information	BES Support
Not Activated	BES Relay Status	BES Support

Figure 8-42 Analysis List report

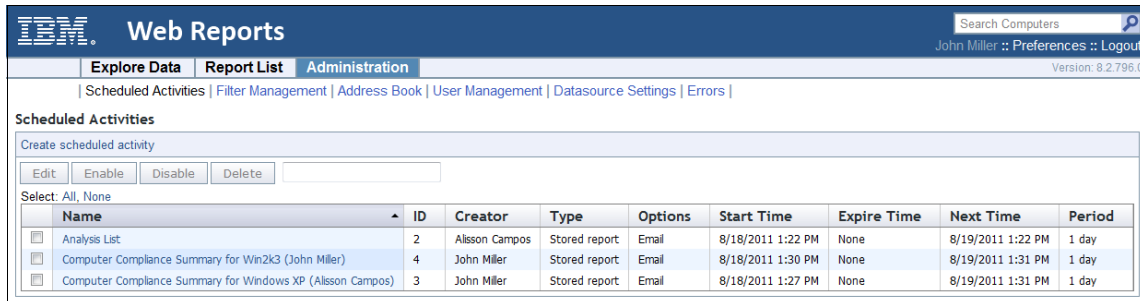
2. The *Computer Compliance Summary for Windows XP* report is distributed to Mark Evans. The *Computer Compliance Summary for Windows 2003* report is distributed to Patrick James. An example report is shown in Figure 8-43.



Figure 8-43 Computer Compliance Summary report

Follow these steps:

1. From the Web Reports main window, click **Administration** → **Scheduled Activities** → **Create scheduled activity**.
2. Define all parameters necessary for generating the report. For the financial accounting company, the reports are generated daily with no expiration. They are sent to operators and the administrator. The three resulting definitions are shown in Figure 8-44.



IBM Web Reports

Search Computers

John Miller :: Preferences :: Logout

Version: 8.2.796.0

Administration

Scheduled Activities | Filter Management | Address Book | User Management | Datasource Settings | Errors |

Scheduled Activities

Create scheduled activity

Edit Enable Disable Delete

Select: All, None

Name	ID	Creator	Type	Options	Start Time	Expire Time	Next Time	Period
<input type="checkbox"/> Analysis List	2	Alisson Campos	Stored report	Email	8/18/2011 1:22 PM	None	8/19/2011 1:22 PM	1 day
<input type="checkbox"/> Computer Compliance Summary for Win2K3 (John Miller)	4	John Miller	Stored report	Email	8/18/2011 1:30 PM	None	8/19/2011 1:31 PM	1 day
<input type="checkbox"/> Computer Compliance Summary for Windows XP (Alisson Campos)	3	John Miller	Stored report	Email	8/18/2011 1:27 PM	None	8/19/2011 1:31 PM	1 day

Figure 8-44 Scheduled report distribution

- After the reports are generated, they are distributed. The recipients receive an email with the attached report, as shown in Figure 8-45.

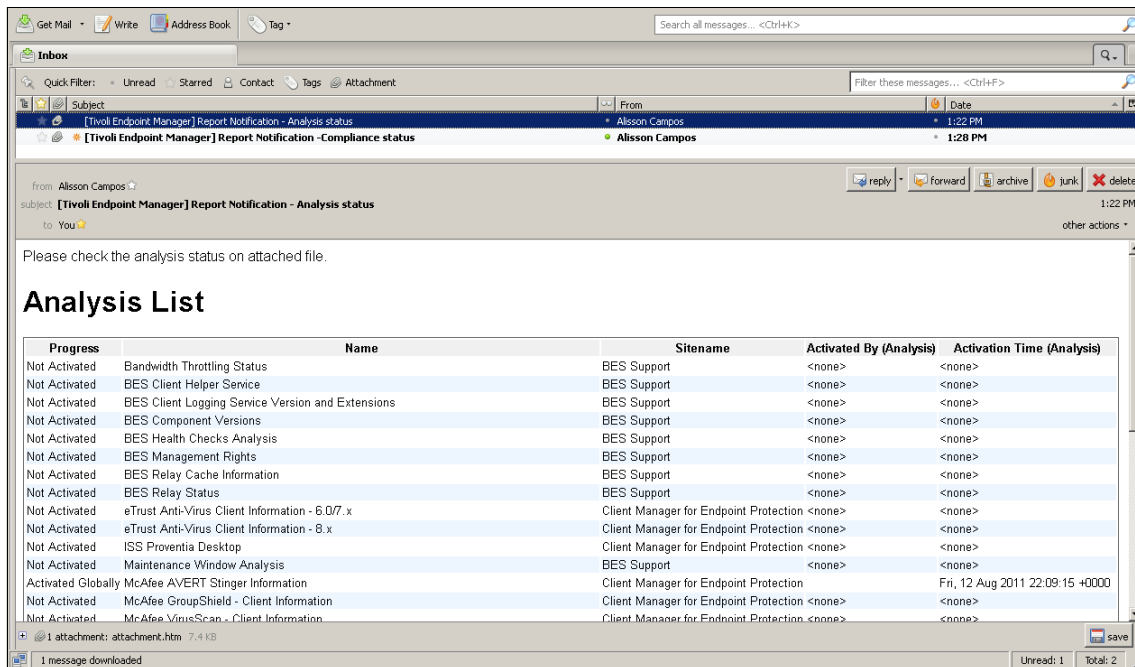


Figure 8-45 Example of a report received by email

For more information about how to create, configure, and distribute reports through Tivoli Endpoint Manager Web Reports, see the *Tivoli Endpoint Manager Web Reports Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v26r1/topic/com.ibm.tem.doc_8.2/Platform/Web_Reports/c_background.html

8.6 Conclusion

In this chapter, we described the approach of the financial accounting company to design and implement its security configuration and compliance management (SCM) module, by using IBM Tivoli Endpoint Manager for Security and Compliance.

First, we introduced the design approach for the financial accounting company of documenting at a high level the current organization standards and security requirements for the operating system platforms. By using this information, we

mapped the requirements to the DISA checklist and improved the number of controls. We defined the checks to implement on Tivoli Endpoint Manager.

Then, we created and customized the design approach. We chose several checks to demonstrate step-by-step how to implement the checks on Tivoli Endpoint Manager. We showed the interaction and how to fix the security control problems on the endpoints. We also described implementation practices to help improve environment administration and maintenance related to SCM within Tivoli Endpoint Manager.

Next, we suggested a project scope change scenario. The financial accounting company showed a noncompliance issue related to PCI regulation, Requirement 5 (anti-virus related). We showed how Tivoli Endpoint Manager flexibility helped to report the compliance status. We explained how we used Tivoli Endpoint Manager to fix the noncompliance issue for this PCI section.

We also briefly described and demonstrated SCM real-time reporting functionalities, based on the *Tivoli Endpoint Manager Web Reports Server*. We provided SCM information in real time to administrators, operators, and the management team.

Finally, we described the administration and maintenance of the endpoint management ecosystem. We explained that a security configuration and compliance solution is never finalized. It must be continuously improved. You must revisit the processes and technical configuration frequently so that they are always aligned with business and functional requirements.

In the final chapter of our business scenario, we look at Security Compliance Analytics reporting closely.



Phase IV: Security Compliance Analytics reporting

In this chapter, we describe the design approach of the financial accounting company to design the reporting solution for security configuration management. The Tivoli Endpoint Manager Analytics platform is used with the module called Security and Compliance Analytics. With this module, the financial accounting company can create reports based on the security configuration management content. We describe the solution in the following sections:

- ▶ “Design” on page 358
- ▶ “Implementation” on page 361
- ▶ “Usage” on page 366

9.1 Design

In 8.1.3, “Designing a new policy model” on page 301, we introduced the security policy CITSEE100 in many variants for multiple operating systems. The financial accounting company plans to create reports to show the current compliance status for its entire IT endpoint environment by using the Security Compliance Analytics solution.

As presented in 5.1.1, “Current IT infrastructure” on page 188, the financial accounting company currently manages more than 100,000 endpoints. The company plans to increase this number in the future (5.2, “Business vision” on page 194). The reporting design must address the requirement from a hardware performance, user, and access management perspective.

9.1.1 Hardware design

In 5.6, “Implementation approach” on page 205, we proposed the physical architecture for the financial accounting company Tivoli Endpoint Manager solution. Because we focus on the Security Compliance Analytics reporting system, Figure 9-1 shows the areas of interest.

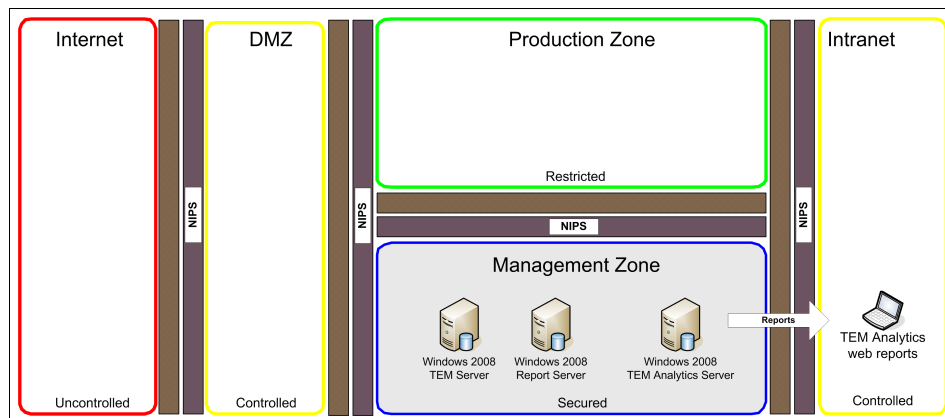


Figure 9-1 Tivoli Endpoint Manager Analytics part of the physical architecture

All machines used for reporting, including the source of data (Tivoli Endpoint Manager Server), user authorization information (Tivoli Endpoint Manager Web Reports server), and dedicated Tivoli Endpoint Manager Analytics server, are in the Management Zone. Access to the reporting web interface is granted from within the intranet only.

In 4.5, “Security and compliance analytics solution design” on page 178, we introduced several preferred practices to follow when implementing Security Compliance Analytics. The financial accounting company decided to set up a storage area network (SAN) solution with fast hard disk drives organized in a RAID 10 matrix. Because of the size of the managed environment, each critical database is in an independent storage location. Figure 9-2 shows a high-level design of the current solution.

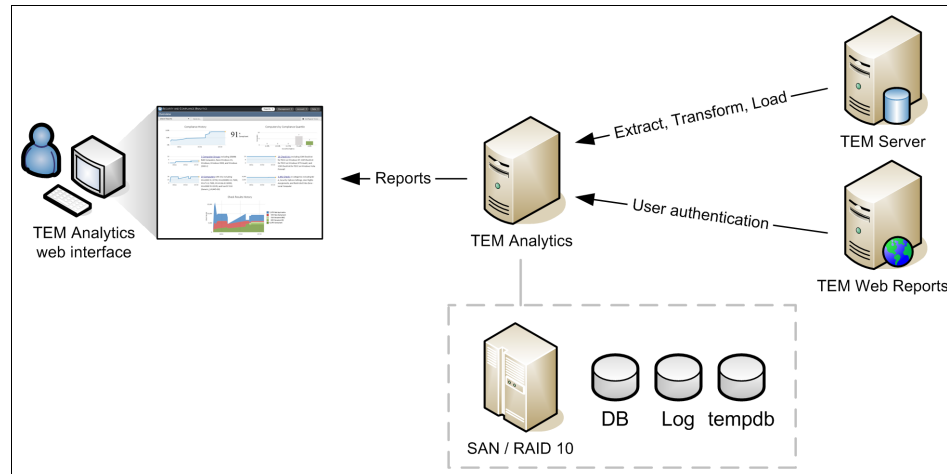


Figure 9-2 High-level analytics design for the financial accounting company

The Tivoli Endpoint Manager Analytics platform uses data from Tivoli Endpoint Manager Server through the extract, transform, and load (ETL) process. The platform can use Web Reports as an authentication source for user credentials at the time of the login request (this functionality is supported for native Web Reports user accounts only). Users can access the Tivoli Endpoint Manager Analytics interface by using a supported browser. The Tivoli Endpoint Manager Analytics server interacts with the SAN for the Microsoft SQL Server storage requirements. Firewall rules must be modified for specific components to enable the usage of the additional hosted service.

For additional information about hardware requirements and guidance for the Tivoli Endpoint Manager Analytics setup, see 4.5.2, “System and hardware guidelines for Analytics” on page 181.

9.1.2 Computer grouping

Tivoli Endpoint Manager Analytics uses computer groups to associate users to the machine scope, exceptions to computers, and aggregate values or trends over time on a defined subset of objects.

Computer groups can be organized in a hierarchical tree, which means that a child computer group inherits definitions from its parent group. For computer grouping in Tivoli Endpoint Manager Analytics, the financial accounting company decided to organize its endpoints into three groups. The three groups present a few sample operating systems. The financial accounting company uses this grouping approach for the rest of its managed operating systems:

- ▶ Geography:
 - Asia
 - Europe
 - North America
 - South America
- ▶ Production systems:
 - Windows XP production
 - Windows 7 production
- ▶ Test system:
 - Windows XP test
 - Windows 7 test

9.1.3 Users and roles

To address the separation of duty requirements, we must create a set of user accounts. Each account is given a set of privileges, organized for different access levels. The following list presents high-level descriptions of the roles:

- ▶ *System administrators* have full access to the solution. This group executes any functionality that is provided in the solution.
- ▶ *Report viewers* can run reports. No data modification is allowed for this group.
- ▶ *Configuration administrators* can modify or update the computer grouping. They can modify computer assignment for reporting.
- ▶ *Exception management access level* creates and updates the exceptions on a daily basis.
- ▶ *Maintenance access level* is for a specific group of users to import data manually, beyond the regular scheduled import settings.

9.1.4 Security configuration management content

The financial accounting company, together with the IBM consultant, create corporate security policies for various operating systems. We originally imported these policies from IBM provided Fixlet Sites, which are designed as an implementation of the Defense Information Systems Agency Security Technical

Implementation Guide (DISA STIG) standard. This security configuration management content is compatible with the Security Compliance Analytics platform. This standard can be automatically imported into the Tivoli Endpoint Manager Analytics database; no additional actions are required.

The Security Compliance Analytics reporting solution needs to import data from the main Tivoli Endpoint Manager database. The import process is scheduled to execute every 24 hours.

We now look at the implementation.

9.2 Implementation

We defined the requirements and completed the planning of the Tivoli Endpoint Manager Analytics rollout. Now, we need to set up the reporting solution, perform the initial data load, and configure the necessary parameters.

9.2.1 Installing the Security Compliance Analytics solution

IBM provides two methods to install the Tivoli Endpoint Manager Analytics platform¹. You can choose to perform an independent installation. Or, if you already have a Tivoli Endpoint Manager Agent deployed on the target machine, you can use the established infrastructure to start the installation process directly from the Tivoli Endpoint Manager Console.

Because the financial accounting company has Tivoli Endpoint Manager Agents already running on its machines, the company uses the second option.

By using the Security Compliance Analytics dashboard, shown in the Figure 9-3 on page 362, you can initiate the installation process.

¹ For the details of the Tivoli Endpoint Manager Analytics installation, see *Tivoli Endpoint Manager for Security and Compliance Analytics Setup Guide*:
http://support.bigfix.com/product/documents/dss/SCA_Setup_Guide.pdf

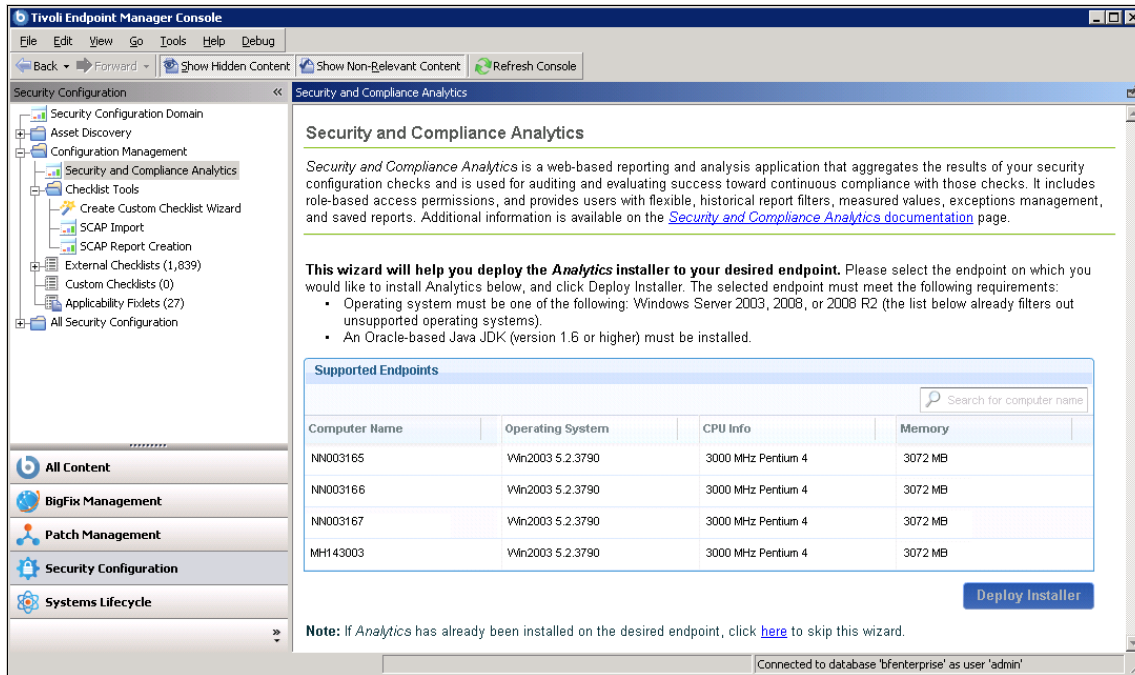


Figure 9-3 Security Compliance Analytics installation dashboard

Click the **Security Configuration** domain to access the dashboard. In the content tree on the left, select **Configuration Management** → **Security and Compliance Analytics**. Follow the instructions. By using the dashboard, you select a target machine and automatically create a Task to deploy and execute the Security and Compliance Analytics installer.

Oracle Java Development Kit Version 6: Security Compliance Analytics V1.1, available at the time of writing of this book, defines a prerequisite for successful installation. The Oracle Java Development Kit Version 6 Update 18 or newer must be on the endpoint *before* the Security Compliance Analytics installation.

Next, proceed to complete the installation and initial configuration by using the steps described in the *Tivoli Endpoint Manager for Security and Compliance Analytics Setup Guide*:

- ▶ Provide administrative user credentials.
- ▶ Provide the Tivoli Endpoint Manager Server location and database credentials. The credentials allow data retrieval from the server and transformation for the data warehousing.

- ▶ Provide (optionally) the Tivoli Endpoint Manager Web Reports server locations and credentials, to use the Web Reports authentication for the users. Tivoli Endpoint Manager Analytics, in addition to using its own authentication, can use the Web Reports database for authentication, as well.
- ▶ Manage the initial data import from the Tivoli Endpoint Manager database into the Security Compliance Analytics database.

After you complete these steps, you can see the first report, which shows the deployment compliance overview, as depicted in Figure 9-4.

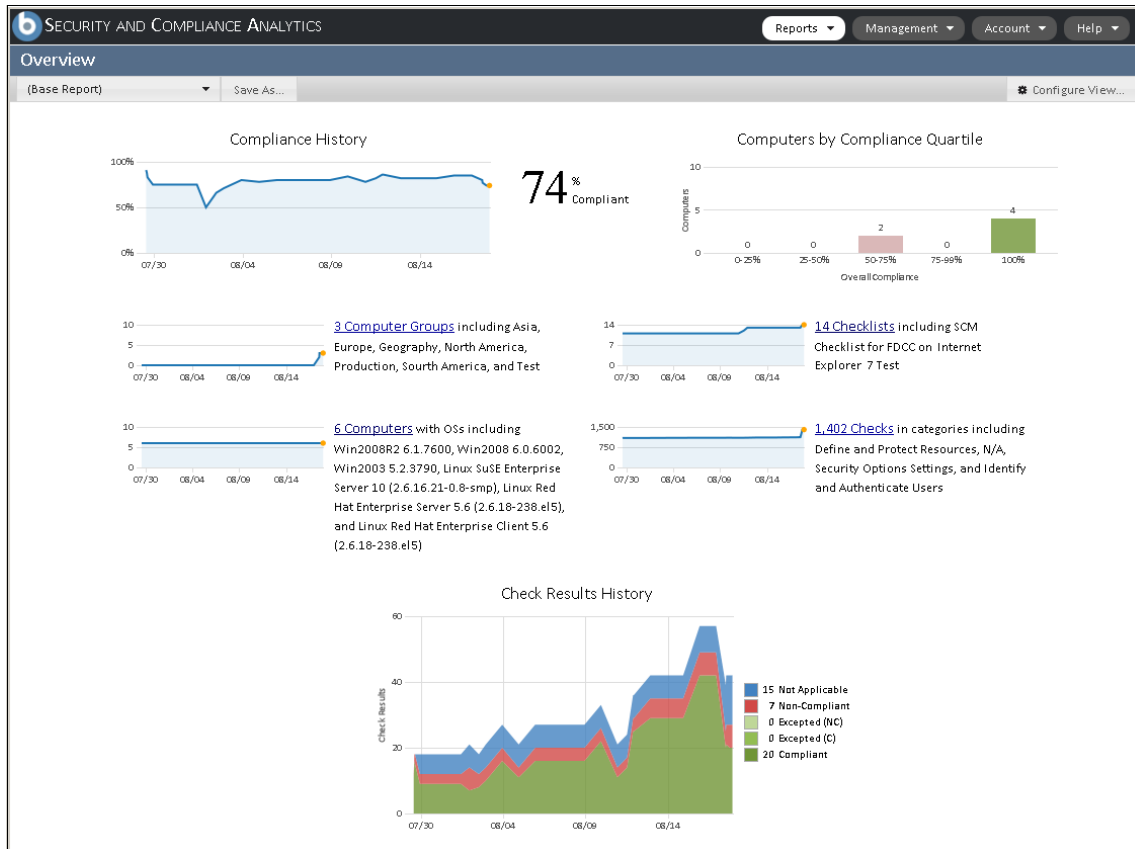


Figure 9-4 Security Compliance Analytics deployment overview

9.2.2 Implementing users, computer groups, and permissions

After the installation and initial data import, Security Compliance Analytics defines only one user with administrative authority. To create the other users, we need to define *user roles*. A user role can have several rights attached to it, and a

user can be assigned to a role. A single user can be assigned to multiple roles. The financial accounting company defines the roles (The Administrators role is already defined after the installation), as shown in Table 9-1.

Table 9-1 Security Compliance Analytics user roles

Role name	Description
Administrators	Can manage the system and create computer properties
Auditors	Can run reports
Computer Group Editors	Can define computer groups
Exception Managers	Can define exceptions
Import Managers	Can run data import

In the next step, you create *user accounts* and assign them to the defined roles. After you complete those steps, you need to create *computer groups* in the Security Compliance Analytics system.

User access scope is defined by creating a Security Compliance Analytics computer group. These computer groups are defined by *computer properties*. A computer property can be defined by a user with the appropriate privileges.

Computer group distinction: A computer group that is defined in the Security Compliance Analytics setup is not the same as a computer group that is defined in Tivoli Endpoint Manager Server. However, it is possible, and can be set up by an administrator, to define computer groups in a way so the same scope is covered..

Figure 9-5 on page 365 shows the computer groups defined by using the web interface in Security Compliance Analytics. We created these computer groups to address the requirements in 9.1.2, “Computer grouping” on page 359.

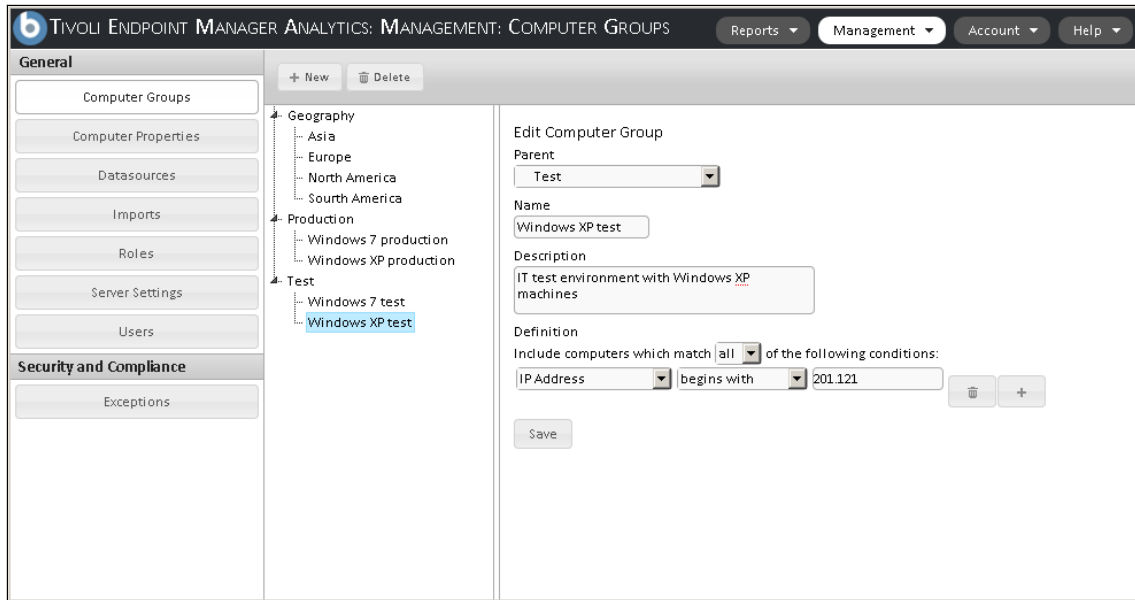


Figure 9-5 Computer groups defined in Security Compliance Analytics

We defined user roles and computer groups. Now, we must create user accounts and link them to roles and computers.

In addition to the users defined directly in Tivoli Endpoint Manager Analytics, it is possible to use Web Reports accounts as authenticated users. In the current release, Tivoli Endpoint Manager Analytics is unable to authenticate against an Active Directory. If users change their Web Reports accounts password, Tivoli Endpoint Manager Analytics requires the new set of credentials at the next login event. This requirement allows an inherited control of password complexity for all accounts imported into Tivoli Endpoint Manager Analytics from Web Reports. If the Web Reports account of a user is deleted, and the user is still logged in, the user can still run reports within the current session.

We created accounts, considering the privileges requirement defined by the financial accounting company. Figure 9-6 on page 366 depicts a sample user that is given only *auditors* authority. The user can run available reports only. The user is given access to the production computer group. This restriction means that the user can generate reports against computers that are used by employees, but not by the IT department.

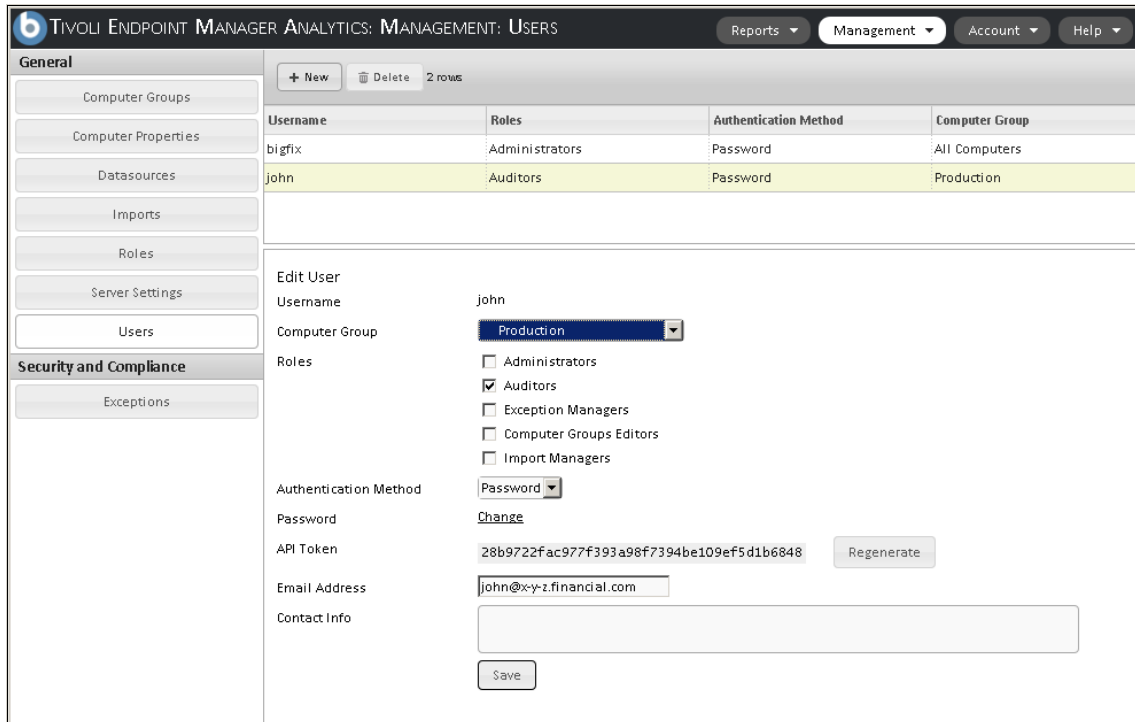


Figure 9-6 Sample user definition in Tivoli Endpoint Manager Analytics

Other user accounts were also created, according to the needs of the financial accounting company. The initial implementation is complete. Next, we describe the typical usage.

9.3 Usage

By using the Security Compliance Analytics solution, users can navigate and explore security configuration check results. Each computer in the deployment evaluates the appropriate checks, and each computer reports a *pass*, *fail*, or *not applicable* for each check. Each computer also reports computer properties and Analysis values. Analysis values are available only if they were activated in the Tivoli Endpoint Manager Console.

The security configuration management check results are aggregated by the Security Compliance Analytics server. They are augmented with computer properties and Analysis values to provide compliance overviews and drill-down

lists into the results. Figure 9-7 depicts the structure of the application reports and paths to the applicable data set.

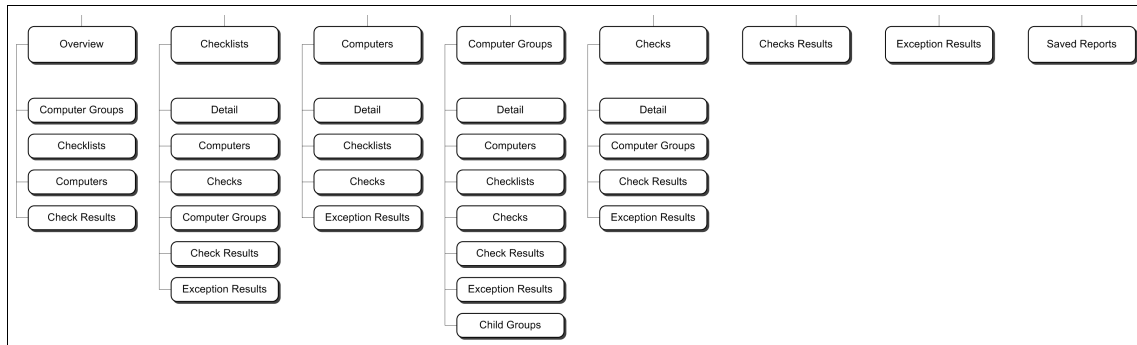


Figure 9-7 Security Compliance Analytics reports structure

The top level lists the general report branches that are available directly from the main menu of the application (Figure 9-8). The boxes underneath that hierarchy represent drill-down reports that can be opened after starting the main report. Certain detailed reports have the same name, but they are listed under different main report branches. The difference is the scope of the detailed report. For example, we can open two reports: *Computers* and *Checklists*. Both reports offer the *Checks* detailed reports. The difference is that, in the first case, the detailed report lists the checks available for a selected computer. In the second case, the detailed report lists the checks available within a selected checklist.

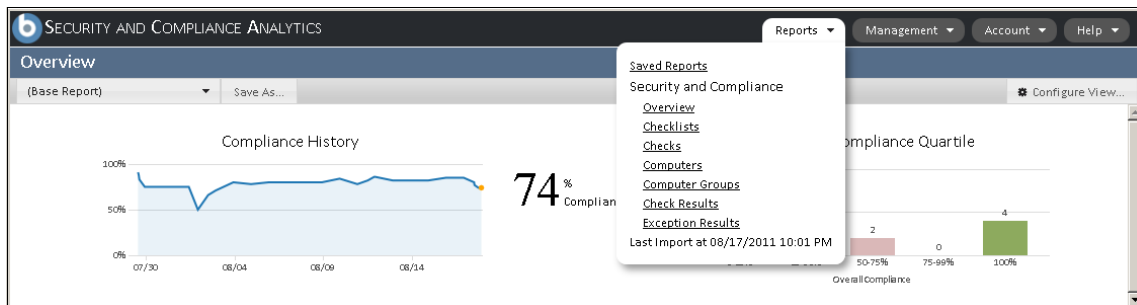


Figure 9-8 Reports menu expanded in Security Compliance Analytics

9.3.1 Report execution

The Security Compliance Analytics module for the Tivoli Endpoint Manager Analytics data warehouse solution provides compliance reports for the current environment. In 8.1.3, “Designing a new policy model” on page 301, we introduced the security configuration management implementation for the

financial accounting company. In this section, we continue this scenario and provide reporting examples. We focus on the CITSSE100 for WinXP V1.0 checklist. In particular, we describe the Password uniqueness does not meet minimum requirements Fixlet that we customized according to the organization policy definition.

Fixlet in Console and analytics web page

Figure 9-9 depicts the Fixlet description in its source form as it is provided in the DISA STIG for Windows XP V6R1.18 Fixlet Site. The Fixlet parameter contains the default value, so this capture is taken before the customization step.

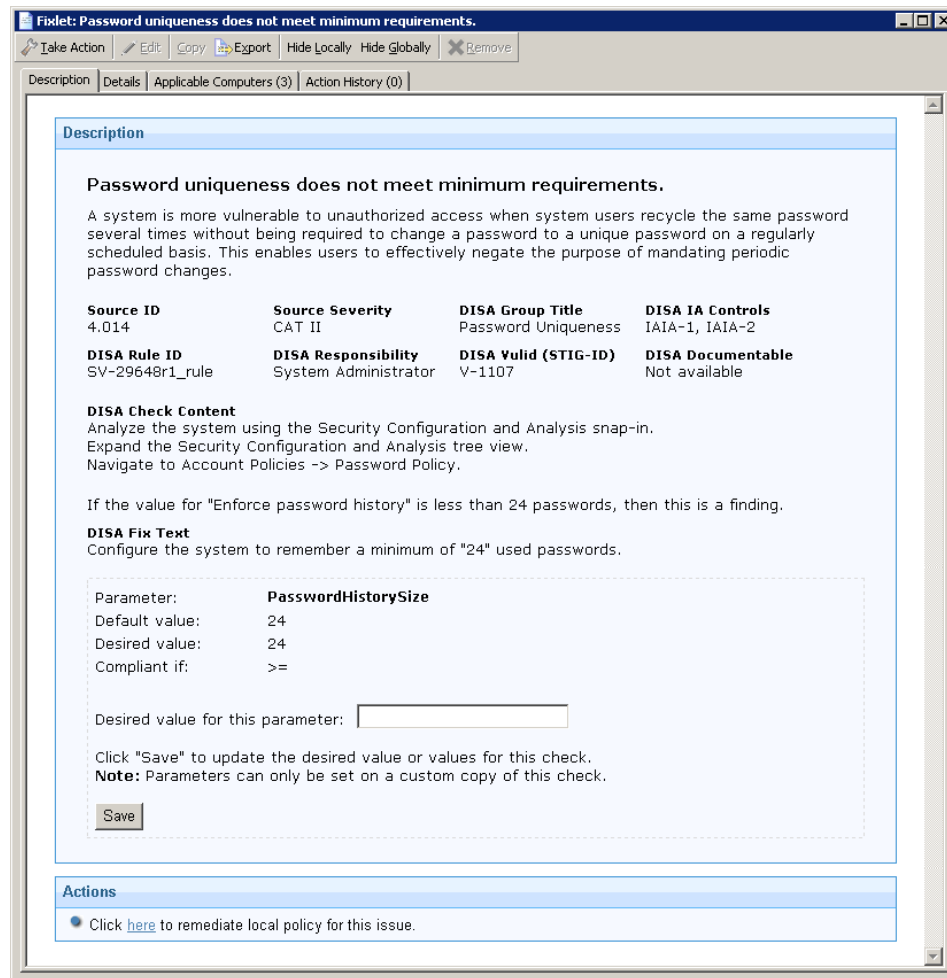


Figure 9-9 Fixlet description in Tivoli Endpoint Manager Console

After the Fixlet is imported into Security Compliance Analytics, you can see its description. To open the description page, starting from the Overview report, select **Reports** → **Checklists**. Then, you can locate the checklist that you want. After you click the checklist name, a list of all checks (Fixlets) is shown. Then, you locate a Fixlet and click its name. Figure 9-10 shows the Fixlet description.

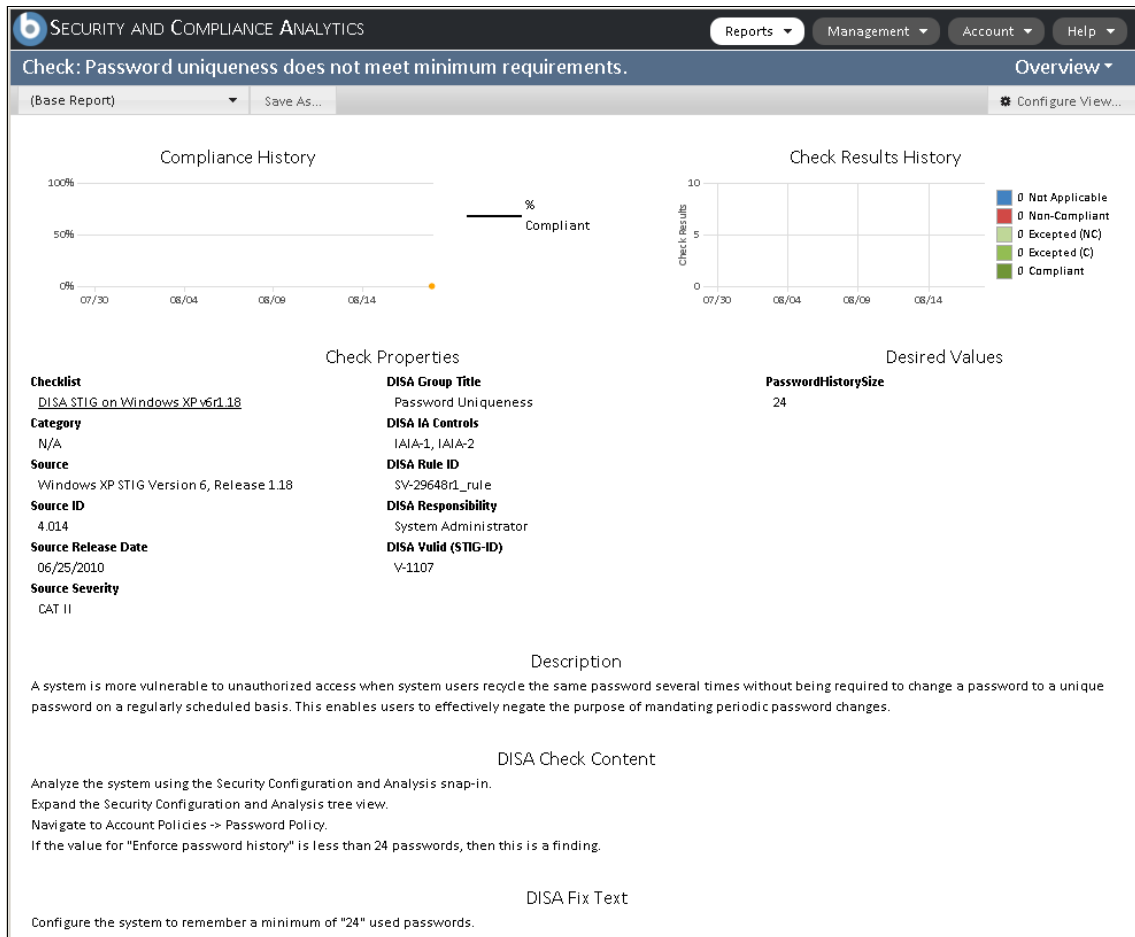


Figure 9-10 Fixlet description in Security Compliance Analytics

As you can see, all the information provided in the Fixlet description is also available in the web interface. Look at the *Desired Value* section on the right side. In a customized Fixlet, the Desired Value is equal to 10, according to the financial accounting company security policy.

From the environment overview to the checklist scope

After the user is authenticated, the environment overview page is displayed. The content of the sections depends on the assigned roles and computer groups. Figure 9-11 shows the overview report:

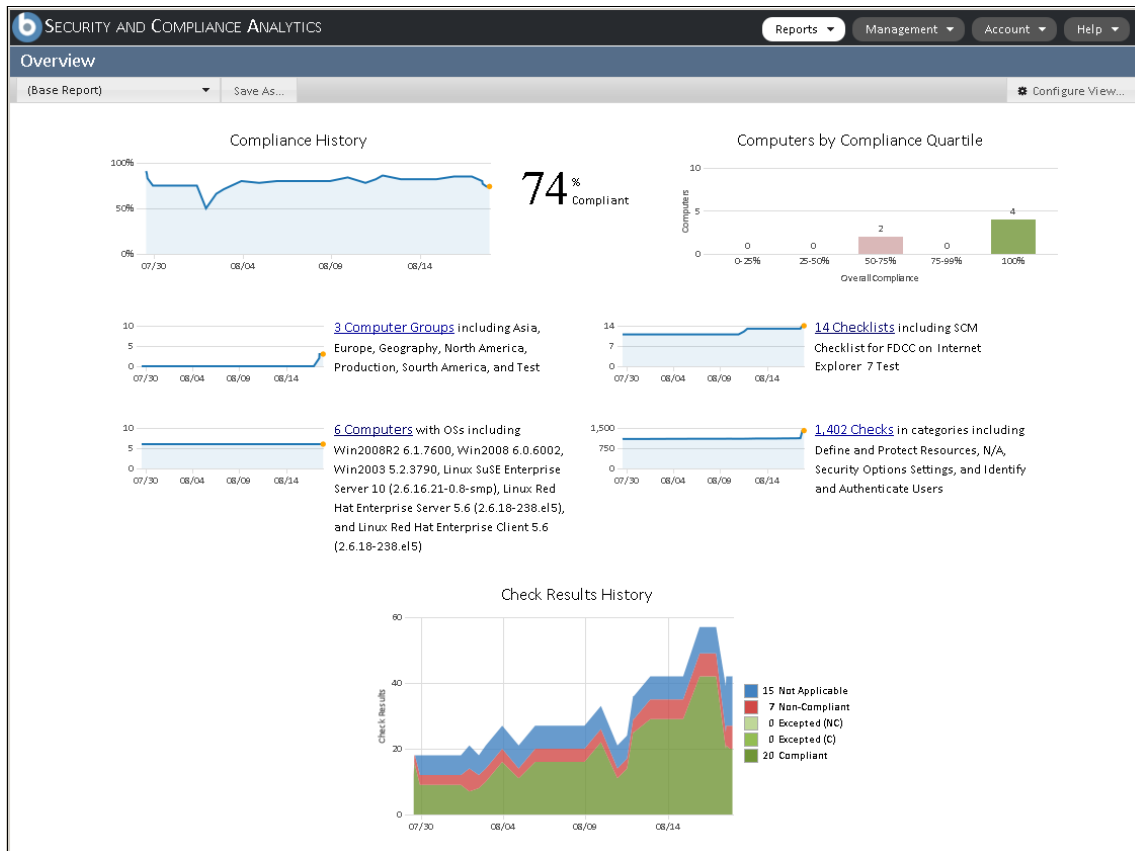


Figure 9-11 Environment overview page

The overview report contains the following sections:

- ▶ *Compliance History* represents the aggregate check results (compliant or non-compliant). It includes all computers within the environment that evaluate security configuration management content. This graph appears on the other overview type report and addresses the overview-specific scope.
- ▶ *Computers by Compliance Quartile* represents the computers grouped by compliance level.
- ▶ The deployment information provides additional information according to the overview report scope. This section shows the changes in the totals in a

predefined time frame. This data can represent the number of computers, computer groups, checklists, and the total number of checks.

- ▶ *Check Results History* shows the total number of evaluated checks and results over time. It uses colors to differentiate the specific types of check results:
 - *Not applicable*: A check that does not apply to a specific computer.
 - *Non-compliant*: A check that is non-compliant on a specific computer.
 - *Excepted (NC)*: A check that is non-compliant on a specific computer, but it is excepted through a manually created exception.
 - *Excepted (C)*: A check that is compliant on a specific computer, but it is excepted through a manually created exception.
 - *Compliant*: A check that complies with the checklist desired values.

From checklist overview to details

You want to collect the specific information about the compliance posture of the Windows XP machines according to the *CITSSE100 for WinXp v1.0* policy. Therefore, you must create the appropriate checklist reports. Follow the steps:

1. From the Overview report (Figure 9-11 on page 370), select the ***n* Checklists** link in the deployment information section. The ***n*** represents the number of checklists deployed within the environment.
2. Select the **Reports** → **Checklists** menu item to switch to the detailed report.
3. A list of all available checklists is displayed, as shown in Figure 9-12 on page 372. This report provides high-level compliance information, similar to the Overview report. However, in this case, the scope is limited to the Checklist level. The numerical and graphical controls provide the following information:
 - Graph of historical compliance percentage within the checklist.
 - Compliance percentage for each checklist.
 - Graphical representation of the percentage compliance levels, including states, such as *not applicable*, *compliant*, *noncompliant*, *excepted (NC)*, and *excepted (C)*.
 - Number of checks within the checklist.
 - Number of computers subscribed to the checklist.

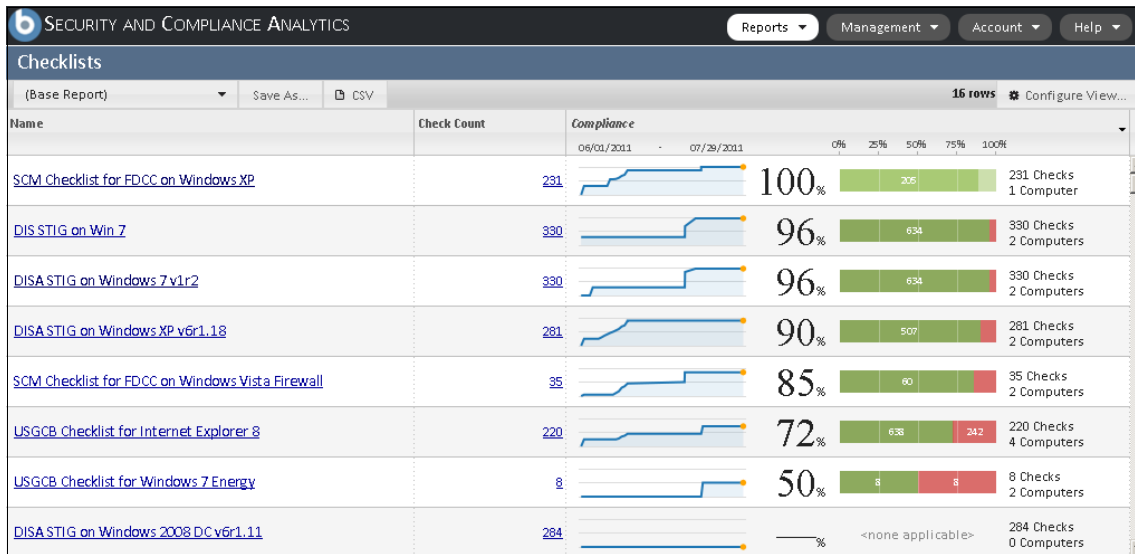


Figure 9-12 Checklists detailed report

Within this view, you can select any checklist report for more details.

Check results details

Because you want to view detailed compliance information about the CITSEE100 for WinXP V1.0 policy, you must switch to the appropriate view:

1. From the menu, select **Reports** → **Check results**. Tivoli Endpoint Manager Analytics displays the default view for that type of report. Because this report displays *all* Check Results from the endpoint environment, you must change the view to include only the machines, checklist, and aspects you want. The default view of the reports is shown in Figure 9-13.

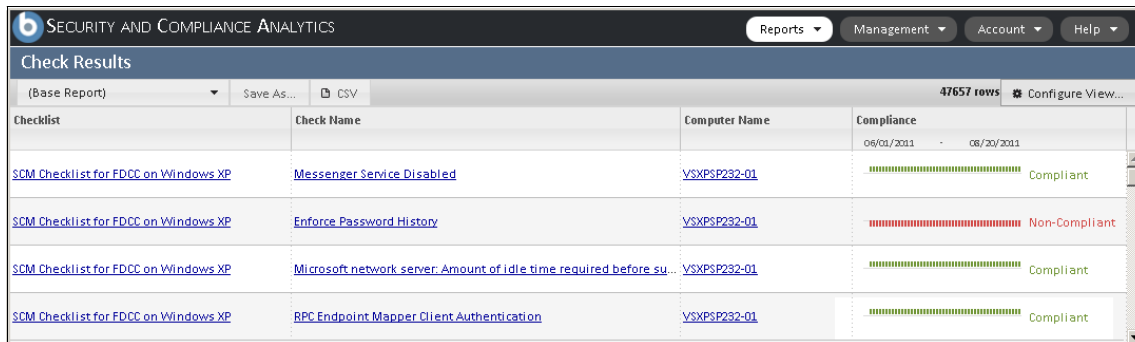


Figure 9-13 Default view of the Check Results report

Efficiency: Although the number of rows calculated by Tivoli Endpoint Analytics displayed in the upper-right report table header might be high, data is retrieved for only those lines that are visible on the window. After the operator scrolls through the report table, data is automatically retrieved from the database.

2. The Check Results report provides detailed information about a single check (Fixlet) within a checklist (policy) for a particular computer. You cannot get a compliance graph as shown in the previously described reports.
3. When you click a particular Computer Name link (from within Figure 9-13 on page 372), you can retrieve a detailed history overview, as shown in Figure 9-14.

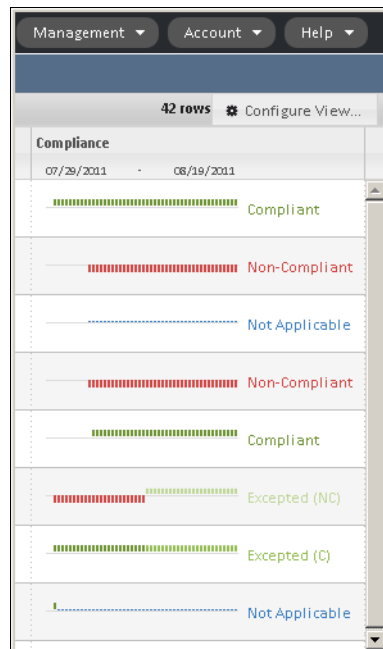


Figure 9-14 Single check compliance status and history

Within the depicted time frame (displayed as the column header), the control shows the last known state of the check on an endpoint. The control provides color and shape encoding of the historical state.

Customizing the view

With Security Compliance Analytics, you can customize the view and filter results in a detailed report. These options are available after you select **Customize View** in the table header on the upper-right side of each report. The available customization options for the Check Results report are depicted in Figure 9-15.

Configure View

Columns

Check

- Checklist
- Check Name
- Category
- Source
- Source ID
- Source Release Date
- Source Severity
- XCCDF Profile ID
- XCCDF Rule ID
- XCCDF Rule Weight
- XCCDF Benchmark ID
- DISA Group Title
- DISA IA Controls
- DISA Rule ID
- DISA Responsibility
- DISA Vulnid (STIG-ID)
- DISA Documentable
- Desired Values

Computer

- Computer Name
- Last Seen
- OS
- DNS Name
- IP Address
- Computer ID

Check Result

- State
- Compliance
- Measured Values

Time Range

All

Last 3 days

08/02/2011 to 08/14/2011

07/30/2010 08/01/2010 08/03/2010 08/05/2010 08/07/2010 08/09/2010 08/11/2010 08/13/2010 08/15/2010 08/17/2011

Filters

Include check results which match all of the following conditions:

Checklist in set

Submit

Figure 9-15 Customization dialog

The customization dialog contains these sections:

- ▶ Columns

Depending on the report type, in this section, you select the columns that you want displayed in the report. The section is typically divided into subsections if appropriate. Visible columns are marked with a selected check box.

- ▶ Computers

In this section, you can select the details you want to display for the endpoint.

► Time Range

In this section, you can define the time range that you want displayed in the report. You can choose between all the available data, the data that was collected in the last *n* days, or select a specific date range.

A pink bar at the bottom of this section indicates the selected time range. The orange dots show the execution times for the ETL processes.

► Filters

In this section, you can define the scope of data to display. Using this interface, you can define complex conditions. You can select particular checklists, individual computers, or groups of computers to filter the report view based on these computer properties.

The financial accounting company wants to know the compliance posture results for its CITSSE100 for WinXP V1.0 checklist and the Password uniqueness does not meet minimum requirements Fixlet. The company wants to evaluate the desired value of the password uniqueness (according to the policy, the Fixlet must define a value of 10). Then, the company wants to compare it to the actual values returned by the Analysis from the operating systems of the endpoints. To achieve these results, the company configures the columns and the filtering that are depicted in Figure 9-16 on page 376.

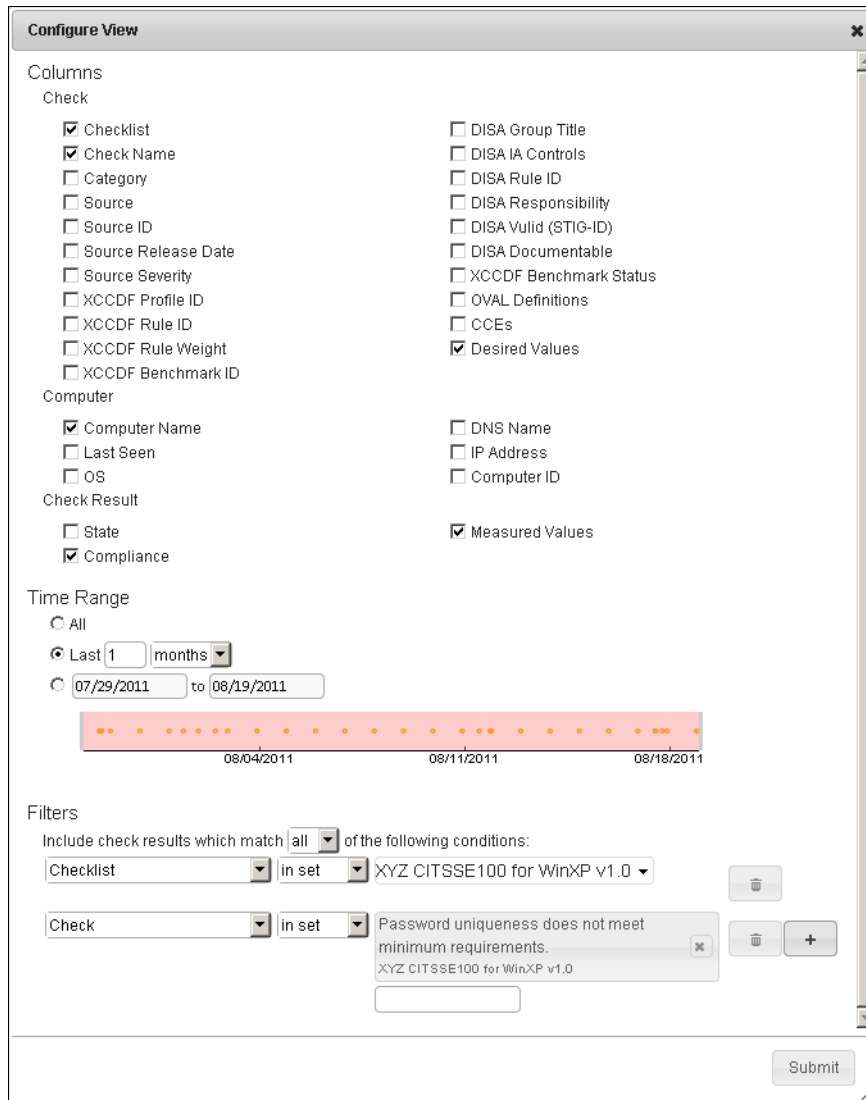


Figure 9-16 Customize the report for the financial accounting company

After the operator clicks **Submit**, Security Compliance Analytics applies the new view configuration and displays the results. A sample report is shown in Figure 9-17 on page 377. This report addresses the requirements of viewing the actual compliance posture and the detailed results for this particular check.

Checklist	Check Name	Computer Name	Desired Values	Compliance	Measured Values
XYZ CITSSSE100...	Password uniqueness does not meet...	XPCCLIENT	PasswordHistorySize: 10	07/19/2011 - 08/19/2011 Non-Compliant	PasswordHistorySize: 0
XYZ CITSSSE100...	Password uniqueness does not meet...	XPCCLIENT2	PasswordHistorySize: 10	Non-Compliant	PasswordHistorySize: 0
XYZ CITSSSE100...	Password uniqueness does not meet...	TEMCLIENT-XP	PasswordHistorySize: 10	Non-Compliant	PasswordHistorySize: 0

Figure 9-17 Customized Check Results view

Saving reports

The structure of the reports shown in Figure 9-7 on page 367 is complex. Manually navigating from the initial overview report to the report layout that you want by applying all the configuration steps every time you need the same report is frustrating. Therefore, for repetitive designs, with Tivoli Endpoint Manager Analytics, you can save your detailed reports.

After you configure a report to address your requirements, use **Save as** to save the report (see Figure 9-18).

Checklist	Check Name	Computer Name	Desired Values	Compliance	Measured Values
XYZ CITSSSE100...	Password uniqueness does not meet...	XPCCLIENT	PasswordHistorySize: 10	07/19/2011 - 08/19/2011 Non-Compliant	PasswordHistorySize: 0

Figure 9-18 Save your detailed report

You can provide a descriptive report name and optionally specify that the report is private. By selecting that check box, you limit the ability to see the report to only the user that created the report. Otherwise, the report is available to all users granted privileges to view reports. You can combine the ability to view the report with other current user privileges for computer groups and properties.

You can open the report by selecting the appropriate link in the Reports menu, as shown in the Figure 9-19 on page 378. Figure 9-19 on page 378 shows a list of saved reports. You can modify the name of the report or change the scope of visibility if you are granted that privilege.

The screenshot shows the 'TIVOLI ENDPOINT MANAGER ANALYTICS: SAVED REPORTS' interface. At the top, there are navigation buttons for 'Reports', 'Management', 'Account', and 'Help'. Below the header, there is a 'Delete' button and a '1 row' indicator. The main content is a table with the following data:

Name	Path	Username	Private
All XYZ checklists	/scm/checklists	reporter	No
Last month password uniqueness for CITSSE100 for Windows XP	/scm/check_results	reporter	No

Figure 9-19 Viewing saved report list

In addition to using saved report views, there is another way to directly access a required report. Tivoli Endpoint Manager Analytics encodes each detailed view within a particular URL. The report type, column configurations, scope, additional filtering, and other required aspects are represented by the URL hash string. By copying the URL and pasting it directly into a browser, you can open the report that you want without navigating any Tivoli Endpoint Manager Analytics menus. To access the data, you must be authenticated and authorized to view the particular report scope. The advantage of this approach is that the URL can be bookmarked within a web browser to view reports.

9.3.2 Exception management

Exception management is an inevitable aspect of every endpoint management deployment. Requirements change, solution deployments are delayed, and systems must conform to old policies and standards longer than expected. These cases require you to define and manage exceptions. The Tivoli Endpoint Manager Analytics reporting system provides a specific interface for exception management. It allows a privileged user to define the scope and time frame of the exception.

With Tivoli Endpoint Manager Analytics, you can delegate exception management. Exception management can be tied to individual operators and associated computer groups. An *exception administrator* can implement exceptions only on machines that administrator manages within a defined computer scope. This capability provides the delegation of exceptions in multiple tenant environments.

Consider the following organizational hierarchy of the financial accounting company:

- ▶ Financial accounting company:
 - Department A:
 - Office 1
 - Office 2
 - Department B

Operators can define exceptions at the financial accounting company level, which are applicable in both Department A and Department B. Department A can define exceptions for both Office 1 and Office 2. An operator in Department A or Office 2 can see the inherited control exceptions. That operator can define more control exceptions in its organization (if the appropriate permissions are granted). An operator in Department A or Office 1 can also see the inherited control exceptions; however, the operators in Office 1 and Office 2 cannot see each other's exceptions. No one in Department B can see exceptions in Department A, or vice versa.

Exceptions can be defined within constraints. They can be applied to associated computer groups or a single machine. Exceptions can be set to expire and can be defined against one or many checks.

Suppressing a single check

We examine a scenario where the policy is distributed among the organization infrastructure. All checks were gathered successfully from the endpoint, and the compliance posture evaluation and reporting are successfully implemented. After a change in the corporate IT security policy, the organization requires that password uniqueness requires a new value. This decision is distributed among the organization to all employees. Due to communication problems with offices in the Asia region, the organization was unable to announce the effectiveness of the new policy worldwide. Due to security reasons, the company decided that the new setting must be effective immediately. To avoid a 100% failing security posture for endpoints in Asia, the company must create an exception for that check for the Asia region. The time frame ends on 14 September 2011.

Figure 9-20 on page 380 depicts a sample exception definition.

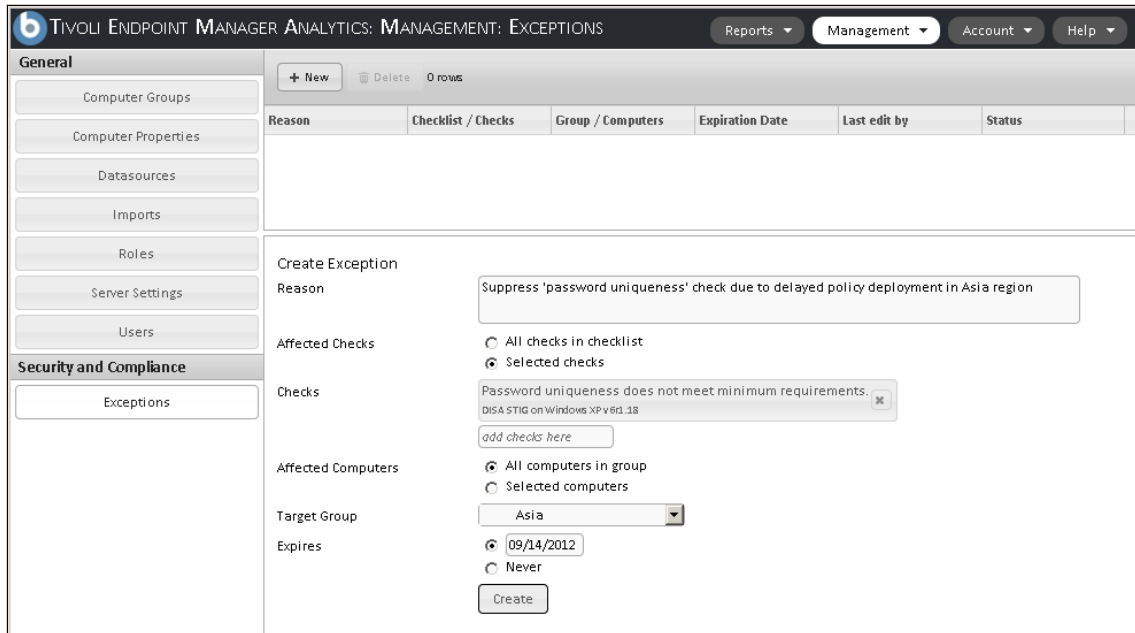


Figure 9-20 Exception definition

The exception contains the natural language description of the reason for the exception. This particular exception is configured for a single check, but it can either exclude whole checklists or groups of checks. As decided by the organization, all checked results from the endpoints in the Asia group are affected by this suppression. The exception is effective until 14 September 2011.

If there is a need, you can update an exception, for example, due to wrongly defined due dates, by using the same user interface. In addition to the information that is shown in Figure 9-20, the saved exception always maintains a history of its changes. If we need to change the expiration date of the previously defined exception to 7 September, the case is shown in Figure 9-21 on page 381. Examine the Exception History that is now displayed in the regular dialog.

TIVOLI ENDPOINT MANAGER ANALYTICS: MANAGEMENT: EXCEPTIONS Reports Management Account Help

General

+ New Delete 0 rows

Reason	Checklist / Checks	Group / Computers	Expiration Date	Last edit by	Status
Suppress 'password...	1 check	Group: Asia	09/07/2012	bigfix	Active

Edit Exception

Reason: Suppress 'password uniqueness' check due to delayed policy deployment in Asia region

Affected Checks: All checks in checklist Selected checks

Checks: Password uniqueness does not meet minimum requirements. DISA STIG on Windows XP v6.1.18

Affected Computers: All computers in group Selected computers

Target Group: Asia

Expires: 09/07/2012 Never

Exception History

Edit action	Action date	Reason	Checklist / Checks	Group / Computers	Expiration Date	Last edit by
create	08/17/2011 06:29 P	Suppress 'passwo1 check		Group: Asia	09/14/2012	excAdmin
edit	08/18/2011 12:01 P	Suppress 'passwo1 check		Group: Asia	09/07/2012	bigfix

Figure 9-21 Defined exception with updated history

After the exception is defined, a privileged user can run the reports. The expected result is that the compliance level is now higher for the checklist compliance, because the endpoints in Asia are not reported as noncompliant. See Figure 9-22 on page 382. The compliance level changed from 5% to 22%. This compliance level was calculated based on the number of computers and checks within the site. Also, notice the changes in the bar graph. The light green color indicates the percentage of excepted checks.

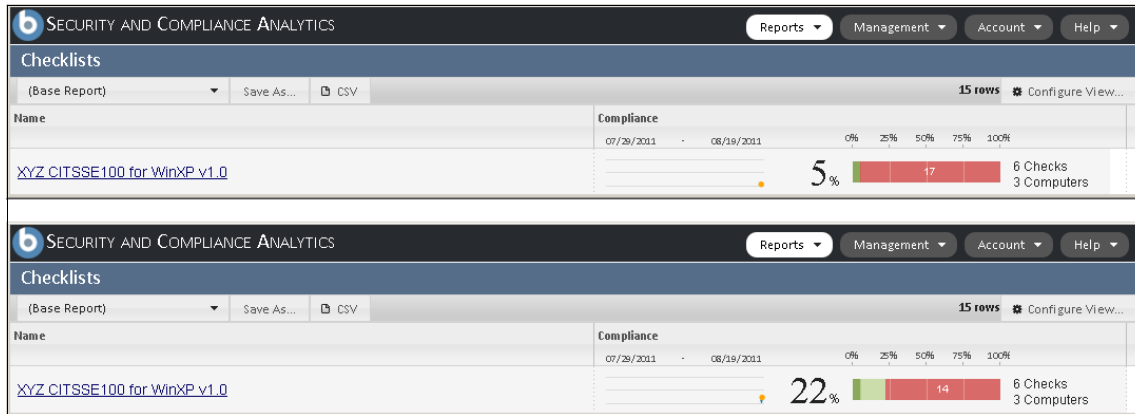


Figure 9-22 Checklist compliance level with (bottom) and without (top) effective exception

On the detailed view, the check is reported as *Excepted*. The characters in parentheses indicate the original state of the check:

- ▶ Compliant - (C)
- ▶ Non compliant - (NC)

Figure 9-22 shows an example of excepted checks, for various states before the exception was applied. If you want to see the state of the Fixlet, the detailed check results report displays the status by using a graphical representation (Figure 9-14 on page 373).

9.4 Conclusion

In this chapter, we followed the financial accounting company requirements for compliance reporting. We introduced the Tivoli Endpoint Manager Analytics platform with the Security Compliance Analytics module. We established a reporting environment and presented the reporting capabilities. We provided additional information about the maintenance of the reporting solution.



A

Service offerings

Organizations in today's ever-changing market need to improve service, reduce cost, and manage risk. Organizations need a dynamic infrastructure, which is instrumented, interconnected, and intelligent. A smarter approach requires speeding time to value, by using proven experience, reducing project risk to improve the return on investment (ROI) of IT, and expanding the business impact.

To get there faster, IBM Software Services for Tivoli provides consulting services, skills enablement, and world-class support. Our services are aimed at helping you move faster, which is key in today's rapidly evolving environment. With our services, you can get your Tivoli software solution into your organization quickly and effectively, putting it to work to solve key business challenges.

Guiding principles for Tivoli Endpoint Manager

Many IBM professionals, clients, and IBM Business Partners are interested in understanding and scoping Tivoli Endpoint Manager services projects. The IBM Security Tivoli Endpoint Manager team delivers hundreds of engagements. We deploy the infrastructure to small (fewer than 7,500 endpoints), medium (fewer than 100,000 endpoints), and large (more than 10,000 endpoints) accounts worldwide.

In this appendix, we outline some of the guiding principles that emerged over time that can contribute to a quick and effective implementation of the Tivoli

Endpoint Manager platform regardless of whether the emphasis is workstations, servers, or both. Above all else, it is key to be aware of three things about Tivoli Endpoint Manager deployments.

Scale and expertise

The platform is highly scalable (up to 300,000 endpoints managed by one server) and the installation of the server-side software is simple. You can install the server-side software with no training, *but* this step is only the first step. The successful deployment and configuration of the complete platform require significant expertise. For example, the traditional Tivoli Endpoint Manager service teams regularly deploy the solution to small clients with fewer than 7,500 endpoints and 5 - 10 locations with only 32 hours of services delivered remotely. For someone who is inexperienced in these deployments, this effort can easily exceed 200 hours. And, due to the power of the platform, it is possible that this inexperienced individual can introduce significant risk into the production environment. So, if you decide to forego professional services involvement in deployments, explore all training options available and certify the key personnel involved in the deployment to mitigate risk and avoid lost time.

Speed

Due to the scalability of the solution and the low requirements for dedicated hardware, the solution can be deployed rapidly. In fact, there are cases when the traditional Tivoli Endpoint Manager teams deployed the solution to over 100,000 endpoints over a single weekend. This example is not normal, however, and typical deployments spread out over 2 - 3 months. Most of this time is spent on non-technical deployment work, for example, getting agreement and access from key players, procuring hardware, and getting access to Relays). In cases where IBM services professionals participate in deployments, the actual on-site service presence over the course of this time is limited. It is typically 2 - 3 days for small deployments and 5 - 10 days for larger deployments.

Solutions, process mapping, and integrations

Due to scale limitations, competing solutions frequently impose a distributed (or federated) management paradigm on the organization. This paradigm can require excessive headcount and result in multiple points of failure. The Tivoli Endpoint Manager solutions, in contrast, all rely heavily on *distributed processing* from a centralized console. Tivoli Endpoint Manager services professionals work hard to help reap the benefits of this massive scale and centralized control. Unfortunately, it is sometimes difficult for organizations to fully understand the

importance of this new paradigm until the platform is fully deployed, at which point priorities frequently shift dramatically.

Due to this predictable shift in priorities, the Tivoli Endpoint Manager services teams encourage organizations to defer work on integrations and process mapping until after the platform is deployed. This strategy is often referred to as a *land and expand* strategy, and it is the hallmark of almost all successful Tivoli Endpoint Manager deployments. In cases where extensive process and integration work is done before platform deployment, we find that the *time to complete the deployment* is compromised. Also, after the platform is deployed and begins to be used as an *intelligence gathering* engine, many of the key assumptions that underlie the up-front process and integration work are destroyed.

As a result, we learned that most successful organizations deploy the platform rapidly, use it to gather intelligence for further decision-making, and then make investments in process definition and integrations as required. If there is concern that the budget might not be available to support post-deployment activities, we suggest allocation of reasonable funds for post-deployment hours to provide the organization with a reasonable level of comfort.

Phased implementation methodology

In support of the *land and expand* strategy, we suggest the following phased deployment strategy.

Phase I: Platform deployment

This phase consists of a comprehensive deployment of the Tivoli Endpoint Manager Agent to all endpoints in the organization. Success in phase I is a precursor to subsequent efforts, because all Tivoli Endpoint Manager solutions rely on a single Agent, regardless of the operating system. Phase I culminates in activating key Tivoli Endpoint Manager policies for *assessment only*, which eliminates the need of time-consuming change control procedures in most environments.

Phase II: Solutions implementation

This phase consists of a systematic adoption of each licensed solution based on priority, opportunity, and any key discoveries that are made by the assessment policies. Most of the Tivoli Endpoint Manager portfolio can be activated with no additional software installations. As a result, this phase contains the following key activities:

- ▶ Verification that the platform is ready to accept the incremental use case
- ▶ Assistance with preferred practice configuration and use of the solution
- ▶ Migrations and minor customizations that might be required to address unique business requirements

The Tivoli Endpoint Manager services teams frequently deliver *QuickStart services* that consist of platform deployment plus a series of *solutions workshops*. These services tend to greatly accelerate the learning curve of the operational teams of the organization and overall success with the solutions at a minimal cost.

Phase III: Integrations and custom reporting

Integrations and customized reporting requirements are addressed in phase III. Many organizations effectively used the Tivoli Endpoint Manager solutions for years with no investment in integrations. As the Tivoli Endpoint Manager solution set continues to mature, however, opportunities for dramatic operational or ROI impact are increasingly available. Especially, configuration management database solutions (CMDBs), ticketing systems, and asset management systems opportunities are increasing.

Custom reporting is also typically reserved for phase III. Unlike many competing solutions, investments in custom reporting tend to be modest. However, many organizations can achieve substantial benefits from a handful of strategic custom reports. Many straightforward reporting extensions are within the reach of Tivoli Endpoint Manager administrators. Other extensions require more advanced development skills.

Service offerings overview

Many Tivoli Endpoint Manager services are delivered through standardized offerings that are contracted for through predefined statements of work (SOW). These services are designed primarily to address phase I and phase II requirements. In each case, they are *pre-sized* at a level designed to support deployments of various sizes. All the standardized service offerings are designed to facilitate client self-reliance and longer-term success with the product. It is incumbent on organizations to assign qualified and focused resources to the project and that these resources continue on the project for at least 1 - 2 years.

Tivoli Endpoint Manager packaged service offerings and QuickStart continue to evolve over time. The following link is the best way to find the current offerings:

http://www.ibm.com/software/tivoli/services/consulting/offers-provisioning.html#provisioning_tem

Table A-1 on page 387 outlines the basic structure of the services offered.

Table A-1 Packaged services: Fall 2010

Class of offering	Offering name	Purpose
Platform Offerings	IBM QuickStart Deployment Service for Tivoli Endpoint Manager	Fixed fee platform deployment for up to 7,500 endpoints in up to 15 locations. See Table A-2 on page 388 to estimate scale for larger deployments. ^a
Solutions Offerings	IBM Lifecycle Management QuickStart Services for Tivoli Endpoint Manager	(48 hrs) Lifecycle Management Solution Implementation Services for Windows Patch Management, Asset Discovery, Software Distribution, and/or Power Management. Multiples of this offering might be required for some organizations.
	IBM Advanced Lifecycle Management QuickStart Services for Tivoli Endpoint Manager	(96 hrs) Lifecycle Management Solution Implementation Services for UNIX/Linux Patch Management, Data Security Standard (DSS) Software Asset Management, and OS Provisioning. Multiples of this offering might be required for some organizations.
	IBM Security and Compliance Quickstart Services for Tivoli Endpoint Manager	(96 hrs) Security and Compliance Solution Implementation Services for Endpoint Protection, Security Configuration and Vulnerability Management, Windows Patch Management, and UNIX/Linux Patch Management. Multiples of this offering might be required for some organizations.
	Tivoli Configuration Manager to Tivoli Endpoint Manager Migration Workshop	(32 hrs) Migration Planning Workshop for clients that want to replace Tivoli Configuration Manager with Tivoli Endpoint Manager.
Assistance/ Augmentation Services	IBM Expert Assistance Services for Tivoli Endpoint Manager (small)	(150 hrs) A pool of hours to assist a client with Platform Deployment, Solution Implementations, Custom Fixlet Development, Custom Reporting, and Mentoring/Coaching.

Class of offering	Offering name	Purpose
	IBM Expert Assistance Services for Tivoli Endpoint Manager (medium)	(300 hrs) A pool of hours to assist a client with Platform Deployment, Solution Implementations, Custom Fixlet Development, Custom Reporting, and Mentoring/Coaching.
	IBM Expert Assistance Services for Tivoli Endpoint Manager (large)	(600 hrs) A pool of hours to assist a client with Platform Deployment, Solution Implementations, Custom Fixlet Development, Custom Reporting, and Mentoring/Coaching.

a. Recommended relative effort calculations for platform deployments larger than 7,500 endpoints.

Additional effort due to the number of endpoints and locations can be estimated by using Table A-2.

Table A-2 Additional effort

Endpoints/locations	Additional effort (estimated)
Fewer that 7,500 endpoints Fewer than 15 locations	N/A
7,500 - 14,000 endpoints 15 - 30 locations	30 hrs.
15,000 - 30,000 endpoints 30 - 100 locations	60 hrs.
30,000 - 90,000 endpoints 100 - 500 locations	120 hrs.
90,000 - 250,000 endpoints 500 - 1,000 locations	180 hrs.
250,000 - 999,000 endpoints 1,000 - 3,000 locations	260 hrs.

Platform deployment services

Platform deployment services help organizations to establish a sound Tivoli Endpoint Manager deployment and begin to build the complementary administrator expertise required to keep their deployment healthy. Activities in deployment projects are generally broken into three phases.

Deployment planning

In this activity, the Tivoli Endpoint Manager services teams typically perform these services:

1. Initiate and lead a project planning and coordination call (approximately 1 - 1.5 hrs).
2. Deliver a pre-deployment planning checklist or email request to complete in preparation for attending the technical planning calls.
3. Initiate and lead a few pre-deployment technical planning calls (typically, less than 1.5 hrs each).
4. Deliver a deployment architecture and hardware sizing memo.
5. Document resource requirements for the duration of the project.
6. Validate the completion of pre-deployment activities before scheduling on-site deployment activities.

On-site deployment activities

In this activity, the Tivoli Endpoint Manager services team typically assists you with these tasks:

1. Install the Tivoli Endpoint Manager software on the main Tivoli Endpoint Manager Server.
2. Install and configure the Tivoli Endpoint Manager Console software on at least one device. This component can be installed on the main Tivoli Endpoint Manager Server.
3. Install and configure the Tivoli Endpoint Manager web server software on at least one device. This component can also be installed on the main Tivoli Endpoint Manager Server.
4. Select an appropriate Agent deployment mechanism and develop and test an Agent deployment package. Multiple packages, typically not in excess of three packages, can be provided, as well.
5. Deploy the Tivoli Endpoint Manager Agent to a handful of Relay devices, promote these devices to Relays, configure Relay selection parameters, and test Relay affiliation.

Deployment optimization activities

In this activity, the Tivoli Endpoint Manager services team typically assists you remotely to perform these tasks:

1. Complete the installation of Tivoli Endpoint Manager Agents on your endpoints.

2. Complete the implementation of a Relay infrastructure, including Relay selection and Relay affiliation strategies for your Tivoli Endpoint Manager platform.
3. Configure access rights for a few console operators and design an access provisioning strategy for any remaining console operators.
4. Develop a maintenance and performance monitoring plan for your Tivoli Endpoint Manager platform.
5. Develop a backup and recovery strategy for your Tivoli Endpoint Manager platform.

Solutions services

Solutions services help organizations to incorporate the preferred practices into their use of the key product functionality. Solutions services also help organizations to build up a body of competence in the operator community that facilitates ongoing success with the product. The primary focus of solutions services is the Solutions Workshop. The Tivoli Endpoint Manager services professionals assist you with these activities:

- ▶ Evaluate your current state and business requirements
- ▶ Educate you on preferred practices to use Tivoli Endpoint Manager in comparable environments
- ▶ Make key decisions about how to configure and use your Tivoli Endpoint Manager solutions.

Workshops take anywhere from 2 - 4 hours each and are typically done while on-site. Subsequent hours are delivered either remotely or on-site and focus on mentoring and coaching your Tivoli Endpoint Manager operators as they go into production. The solution services are grouped into Standard and Advanced services based on ranges of the complexity and effort of implementation.

Lifecycle Management services

Lifecycle Management offerings are ever-evolving. As of this writing, the following solutions are available for workshops under the standard lifecycle management offering:

- ▶ Windows Patch Management
- ▶ Asset Discovery
- ▶ Administrator-driven Software Provisioning
- ▶ Power Management
- ▶ Tivoli Remote Control

Advanced Lifecycle Management services

Advanced Lifecycle Management services are scoped slightly larger to accommodate greater complexity. As of this writing, the following solutions are available for workshops under the Advanced Lifecycle Management offering:

- ▶ UNIX/Linux Patch Management
- ▶ OS Provisioning
- ▶ Software Usage Analysis

Security and Compliance services

As of this writing, the following solutions are available for workshops under the standard Security and Compliance offering:

- ▶ UNIX/Linux/Windows Patch Management
- ▶ DSS Security and Compliance Analytics (SCA) Implementation
- ▶ Endpoint Protection

Advanced services offerings

The Tivoli Endpoint Manager services professionals have a wide array of expertise and experience. The advanced services offerings include the following key areas:

- ▶ Service Desk/Ticketing Integrations: IBM Tivoli Service Request Manager®, Remedy, Service Now, and Remedy Service Desk Express
- ▶ CMDB Integrations: IBM Tivoli Change and Configuration Management Database (CCMDB) and Atrium
- ▶ Event-Monitoring Integrations: Q1 Labs, ArcSight, and Splunk
- ▶ Asset Management Integrations: IBM Tivoli Asset Management for IT, Hewlett-Packard (HP), and CA
- ▶ Reporting Integrations: IBM Cognos®, SSRS, and Jasper
- ▶ Custom Development: Integration of third-party solutions (for example, HD Encryption, Application Whitelisting, Monitoring, Device Control, HIPS, and File Integrity Monitoring), other equipment manufacturer (OEM) product integrations, and Portal integration and portal development
- ▶ Advanced Software Distribution Capabilities: User self-service, advanced packaging, and managed services
- ▶ Security Configuration Management: Advanced security configuration management (SCM) content development, including OS, application, and database layers, plus mobile device management

Conclusion

With IBM Software Services for Tivoli, you get the most knowledgeable experts on Tivoli technology to accelerate your implementation and mitigate risk. We use a worldwide team of highly skilled consultants. These consultants offer broad architectural knowledge, deep technical skills, and preferred practices. They provide proven solution design consulting, rapid deployment, integration, configuration, and optimization of your Tivoli software solution. No matter where you are in your transformation to IT service management, IBM Software Services for Tivoli has the right services for you.



IBM deploys Tivoli Endpoint Manager internally

The IBM Chief Information Security Office (CISO) saw an increase in security risks to the IBM internal infrastructure. As the company grew through acquisitions and joint development activities with IBM Business Partners, the percentage of workers that connect from or work in an unprotected infrastructure increased, along with an increasing prevalence of endpoints that are not Windows endpoints. The number, type, and sophistication of security threats in the industry also rose, placing all corporations under greater risk.

The question for the CISO team was, *How do we effectively manage and protect our endpoints under these changing operating conditions?* For the strategist team, the answer was in a new security model.

The old model was insufficient for the changes that occurred, so the team approached endpoint management from a reactive correction perspective. Employees received workstation reports and if they missed a patch or their anti-virus definitions were out of date, they were provided with links to information about how to resolve the issue. To better protect the IBM infrastructure, they had to move to a model of continuous compliance with internal security policies that automatically remediates issues.

Continuous compliance with internal security policies

The CISO team focused on two key requirements. First, the team sought to deliver patches more quickly. The existing tools of the organization required the staff to repackage the security patches for deployment, which often delayed patch availability by up to 14 days.

Any delays in delivering patches can lead to an increased vulnerability period. The IBM team had to get out of the business of repackaging patches because it was inefficient and costly.

The second area on which the CISO team focused was continuous compliance with internal security policies. The previous model provided a point-in-time view of the status of an endpoint when the team ran the tools. So, there were periods of time when the real status was unknown. By using Tivoli Endpoint Manager, IBM gained real-time visibility into the status of endpoints. IBM can demonstrate that the infrastructure is continuously in compliance with the internal security policies.

A pilot program builds the business case

Changing the process for more than a half-million employees and business partners required a strong business case. So, the team launched a Proof of Concept (POC) with IBM Tivoli Endpoint Manager software.

It was the POC that led the CISO team to say *“This solution is the right answer for us.”* They tested the solution to confirm that it performed as expected. The next step was to launch a larger pilot, first to a few thousand endpoints, and then to about 18,000, to confirm scalability.

These pilots also enabled the staff to obtain the concrete data needed to gain executive agreement-in.

The data from the pilot provided the tipping point. The team estimated a 50% savings in endpoint support for security issues based on pilot results. After everyone saw this savings, the directive was to deploy this solution as fast as possible. The CISO team started in December 2010 and within six months deployed Tivoli Endpoint Manager on more than 550,000 endpoints worldwide. This deployment was the largest and fastest internal client deployment within the history of IBM.

The IBM deployment was organized into three geographic groupings: North America, Europe, and Asia Pacific. One dedicated Tivoli Endpoint Manager physical management server, an IBM System x® server with redundant arrays of storage disks (RAIDs), supports each geographic area. The System x platform provides the performance and optical storage to support high transaction rates and centralize management of about 250,000 endpoints for each geographic area.

Tivoli Endpoint Manager Relays enable the software to communicate with endpoints that do not have regular connectivity into the network, such as support systems used by employees to support IBM clients.

The team wanted to cover everything from servers to smartphones. The first focus was workstations; they started with Windows endpoints and are now moving to cover Mac and Linux systems. The CISO team also includes Tivoli Endpoint Manager in the standard build for any new IBM machine.

A 78% decrease in endpoint security issues

By using Tivoli Endpoint Manager, the team reduced the time and cost to monitor endpoints, apply patches, and implement new configuration settings and security software, such as firewall and antivirus solutions. Upon deployment, Tivoli Endpoint Manager identified which specific patches were missing for each individual endpoint, and automatically applied the required patches. Tivoli Endpoint Manager can target specific actions to an exact type of endpoint configuration or user type.

Tivoli Endpoint Manager also automatically remediates about 90% of the Windows requirements, which were previously addressed through workstation reports and manual employee corrective actions. And, the Tivoli Endpoint Manager administrators have real-time visibility to verify the status of each endpoint.

Patches are now available within 24 hours (previously, it could take up to 14 days for patch availability). The company realized a 60% reduction in patch cycle time with a higher rate of patch compliance (98% of required patches applied). While patches are available in 24 hours, the CISO team is quick to point out that they stage the distribution of new patches over a 48-hour period to minimize the risk of faulty patches.

They could have 98% distribution within 24 hours, but they deliberately slowed the process down to confirm that there are no problems with the patches or potentially *poison patches*. Updating half a million systems too fast would make

IBM a test site for software vendors, and the team does not want to incur that risk.

Millions in savings

Since deployment, the savings measured in the pilot were discovered to be conservative. The CISO team initially committed, based on the pilot, to a 50% decrease in internal security problems. This decrease results in about USD 10 million in savings in just the IBM internal support costs to deal with these security issues. After deploying Tivoli Endpoint Manager, IBM realized a 78% decrease in endpoint security problems in the first quarter of global use, which is better than our pilot estimate. This result drives savings well above the initial USD 10 million estimate. The team expects the savings to increase as they complete the deployment to all 750,000 endpoints.

In today's market, where businesses are constantly seeking ways to increase staff productivity, it is also significant is that only three full-time equivalents (FTEs) are needed to support more than 500,000 endpoints.

While the team's main focus is security, one of the important benefits is also the increased efficiency and the ability to effectively manage all endpoints with only three FTEs.

Advanced investigations for sophisticated security challenges

While the project started with a focus on patch management, the CISO team expanded its use of Tivoli Endpoint Manager software to help it investigate unique security problems. For example, recently, the team gathered intelligence about a threat to businesses that use a combination of changes to both dynamic live link (DLL) files and registry entries. DLL files contain code that can be called on by programs to execute a specific function, such as printing. Registry entries are collections of system settings vital in the stability of the computer operating system. Changes to these files can place companies at considerable risk for security breaches and security related-outages, so it is imperative that the CISO team can quickly confirm that IBM systems are not compromised.

One of the compelling benefits with Tivoli Endpoint Manager is that the CISO team can chain together a number of different conditions and see in minutes if any endpoints are at risk for a new security threat. The team no longer must search for each condition and then consolidate the results manually. And if the

team needs to remediate an issue, the team does not have to physically track down the machine, which can be challenging and expensive with systems spread across locations in almost every country. Tivoli Endpoint Manager is flexible enough that the IBM CISO team can use it to deliver or control technology for just about any problem.

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 following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*, REDP-4528
- ▶ *IBM Security Solutions Architecture for Network, Server and Endpoint*, SG24-7581

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website: ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ The IBM Tivoli Endpoint Manager V8.2 Information Center contains information describing the Tivoli Endpoint Manager products and features:
http://publib.boulder.ibm.com/infocenter/tivihelp/v26r1/index.jsp?topic=/com.ibm.tem.doc_8.2/welcome/welcome.html
- ▶ *Tivoli Endpoint Manager for Security and Compliance Analytics Setup Guide*:
http://support.bigfix.com/product/documents/dss/SCA_Setup_Guide.pdf

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Endpoint Security and Compliance Management Design Guide Using IBM Tivoli Endpoint Manager

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Endpoint Security and Compliance Management Design Guide Using IBM Tivoli Endpoint Manager



**Enterprise
integration for
endpoint security
and compliance
management**

**Complete
architecture and
component
discussion**

**Deployment scenario
with hands-on
details**

Organizations today are more widely distributed than ever before, which can make systems management tasks, such as distributing software, patches, and security policies, extremely challenging.

The IBM Tivoli Endpoint Manager platform is architected for today's highly diverse, distributed, and complex IT environments. It provides real-time visibility and control through a single infrastructure, single agent, and single console for systems lifecycle management, endpoint protection, and security configuration and vulnerability management. This platform enables organizations to securely manage their global IT infrastructures faster and more accurately, resulting in improved governance, control, visibility, and business agility. Plus, it gives organizations the ability to handle tomorrow's unforeseen challenges.

In this IBM Redbooks publication, we provide IT security professionals with a better understanding around the challenging topic of endpoint management in the IT security domain. We focus on IBM Tivoli Endpoint Manager for Security and Compliance and describe the product architecture and provide a hands-on design guide for deploying the solution.

This book is a valuable resource for security professionals and architects who want to understand and implement a centralized endpoint management infrastructure and endpoint protection to better handle security and compliance challenges.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**