Tivoli software

IBM

Deployment Guide Series:
IBM Tivoli Access Manager for e-business V6.0

- Full coverage of planning your access control management project
- Complete hands-on installation guidelines
- Based on best practices

Axel Buecker
Anji Greene

Redbooks

ibm.com/redbooks

International Technical Support Organization

Deployment Guide Series: IBM Tivoli Access Manager for e-business V6.0

July 2008

SG24-7207-01

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

Second Edition (July 2008)

This edition applies to Version 6.0 of IBM Tivoli Access Manager for e-business.

Contents

Notices vii
Trademarks viii

Preface ix
The team that wrote this book ix
Become a published author x
Comments welcome xi

Summary of changes xiii
July 2008, Second Edition xiii

Part 1. Architecture and design 1

Chapter 1. Business context 3
1.1 Introduction to Access Manager for e-business 4
1.2 Common business drivers 6
1.3 Common challenges 7
1.4 Conclusion 9

Chapter 2. Planning for customer engagement 11
2.1 Services engagement preparation 12
2.1.1 Implementation skills 12
2.1.2 Roles and responsibilities 13
2.1.3 Available resources 16
2.2 Services engagement overview 18
2.2.1 Executive Assessment 19
2.2.2 Demonstration system setup 20
2.2.3 Analyze solution tasks 21
2.2.4 Creating a contract 21
2.3 Defining solution tasks 23
2.3.1 Deployment tasks 24
2.4 Conclusion 25

Part 2. Customer environment 27

Chapter 3. Company profile 29
3.1 Business drivers and capabilities 31
3.2 Current IT environment 32
3.2.1 Organization 32
3.2.2 IT architecture 33

3.3 Conclusion 34

Chapter 4. Solution design 37
4.1 Defining the access control security policy 38
4.1.1 Secure domains 39
4.1.2 Objects to be secured 40
4.1.3 Permitted actions 41
4.1.4 TAMCO access control security policy 42
4.2 The TAMCO deployment 45
4.2.1 Network structure of the secure domains 46
4.2.2 Client access 49
4.2.3 Controlling Web resource access 50
4.2.4 Back-end resources 54
4.2.5 Management resources 55
4.2.6 Availability and scalability 56
4.2.7 Providing high availability 59
4.2.8 Common Auditing and Reporting Service 65
4.3 Deploying physical components 68
4.3.1 Prerequisite components 70
4.3.2 Components of an Access Manager system 71
4.4 Conclusion 73

Chapter 5. Installing the components 75
5.1 Installing and configuring prerequisites 77
5.1.1 Where to find the CD images 77
5.1.2 IBM Java Runtime 79
5.1.3 Global Security Kit (GSKit) 80
5.1.4 Setup of IBM Tivoli Directory Server 82
5.1.5 IBM WebSphere Application Server 91
5.1.6 Installing WebSphere Application Server V6.0 92
5.1.7 IBM HTTP Server 97
5.1.8 IBM Tivoli Directory Server Web Administration 102
5.1.9 IBM Tivoli Web Portal Manager (WPM) 104
5.1.10 IBM Tivoli Directory Integrator 106
5.2 Installing Access Manager base components 109
5.2.1 Base component installation 110
5.2.2 Installing WebSEAL 114
5.2.3 Installing the Common Auditing and Reporting Service 115

Chapter 6. Configuring IBM Tivoli Access Manager 123
6.1 Configuring the Access Manager base components 124
6.1.1 Configuring and populating the Access Manager user registry 124
6.1.2 Configuring Access Manager components 128
6.1.3 Common Auditing and Reporting Service configuration 140

6.2 Creating the TAMCO object namespace . . . 165
6.2.1 Populating the TAMCO object namespaces . . . 165
6.3 Configuring TAMCO single sign-on . . . 173
6.3.1 WebSEAL forms single sign-on configuration . . . 174
6.3.2 PeopleSoft single sign-on configuration . . . 175
6.3.3 Siebel single sign-on configuration . . . 178
6.3.4 WebSphere Portal single sign-on configuration . . . 188
6.4 Configuring TAMCO single sign-on across domains . . . 200
6.4.1 TAMCO failover cookie requirements . . . 200
6.5 Configuration for scalability and high availability . . . 203
6.5.1 Load balancing within the environment . . . 205
6.5.2 Access Manager Policy Server load balancing . . . 206
6.6 Conclusion . . . 206

Appendix A. Statement of work . . . 209
Building a security infrastructure solution . . . 210
Executive summary . . . 210
Project scope . . . 211
Assumptions . . . 212
IBM Business Partner responsibilities . . . 213
Customer responsibilities . . . 215
Deliverable materials . . . 216
Completion criteria . . . 217
Estimated schedule . . . 217
Charges . . . 218

Appendix B. Tips and tricks . . . 219
Importing and managing certificates on WebSEAL . . . 220
Method 1: HTTPS browser . . . 220
Method 2: IKEYMAN utility . . . 223
Firewall LDAP session timeout . . . 224
Installation Wizard problems . . . 225
Multiple network interfaces . . . 226
ACL problems after Tivoli Directory Server upgrade . . . 226
Validating and maintaining policy databases . . . 227
Using svrsslcfg for unconfiguration . . . 227

Glossary . . . 229

Related publications . . . 233
IBM Redbooks . . . 233
Other publications . . . 233
Online resources . . . 234
How to get IBM Redbooks . . . 234

Help from IBM 235

Index 237

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and 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®	iNotes™	PartnerWorld®
DB2 Universal Database™	Lotus Notes®	RACF®
DB2®	Lotus®	Redbooks®
Domino®	MQSeries®	Redbooks (logo)  ®
HACMP™	Notes®	Tivoli®
IBM®	OS/390®	WebSphere®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

J2EE, Java, JavaScript, JDBC, JDK, JRE, JSP, Solaris, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Internet Explorer, Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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

Preface

Deploying an access control solution for a medium-size business begins with a thorough analysis of the existing business and IT environment. After we fully understand the organization, its deployed infrastructure, and the application framework, we can define an applicable representation of these assets within an access control implementation.

This IBM® Redbooks® publication takes a step-by-step approach to implementing an access control solution based on IBM Tivoli® Access Manager for e-business. Part 1, “Architecture and design” on page 1 takes you through a general discussion of the business context around the use of Tivoli Access Manager for e-business. We also guide you through the necessary planning phases for a typical Tivoli Access Manager for e-business deployment. In Part 2, “Customer environment” on page 27, we introduce an example company profile with existing business policies and guidelines and build an access control solution design for this particular environment. We then describe how the new access control components can be integrated into the existing environment. We then explain how to execute the access control integration tasks that must be implemented in order to create a fully functional end-to-end solution.

This book does not introduce any general access control concepts or systematically explain all of Tivoli Access Manager’s components and capabilities. Instead, those details are thoroughly discussed in the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin 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 in the areas of software security architecture and network computing technologies. He holds a degree in Computer Science from the University of Bremen, Germany. He has 21 years of experience in a variety of 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.

Anji Greene is a Senior IT Architect in the Global TIM Competency team supporting ibm.com. Anji has been with IBM for nine years, focusing on distributed system administration, security infrastructure, and Web hosting. Before coming to IBM in 1998, Anji was employed as a UNIX® System Administrator at Sandia National Laboratories in Albuquerque, NM. She holds a BBA degree from Eastern New Mexico University.

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

Ryan Fanzone, Ron Williams, Pete Rubio
IBM US

Wade Wallace
International Technical Support Organization

Thanks to the following people for their contributions to the first edition of this book:

Peter Tuton, Christopher Hockings
IBM Australia

Ted Ralston, Sadu Bajekal, Sridhar Muppidi, Peter Calvert, Ivan Milman, Avery Salmon
IBM US

Jon P. Harry
IBM UK

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbooks publication dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will team with IBM technical professionals, Business Partners, and customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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 Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@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

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7207-01
for Deployment Guide Series: IBM Tivoli Access Manager for e-business V6.0
as created or updated on July 1, 2008.

July 2008, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

We have added two new chapters in Part 1 of the book:

- In Chapter 1, “Business context” on page 3, we discuss the business context for Tivoli Access Manager for e-business. After a short introduction to Access Manager for e-business, we describe the common business drivers that influence why and how Access Manager for e-business can be implemented in a given business context. Finally, we explain the challenges that you can expect to encounter when confronted with deploying Access Manager for e-business in an enterprise IT environment.
- In Chapter 2, “Planning for customer engagement” on page 11 we discuss the service engagement for Access Manager for e-business in general. This chapter is directed towards IBM Global Business Service agents and IBM Business Partners who assist customers in deploying Tivoli Access Manager for e-business.

We also added two new appendixes:

- In Appendix A, “Statement of work” on page 209 we provide a sample of what you might include in your statement of work contract.

- In Appendix B, “Tips and tricks” on page 219 we provide a few tips and tricks for common issues that may be encountered in an Access Manager for e-business installation. We also suggest some basic troubleshooting techniques and provide operational suggestions to improve your Access Manager for e-business environment.

Changed information

Due to increased experience with the product, we added and changed several technical details throughout Part 2 of the book.

Part 1

Architecture and design

In this part, we discuss the business context of Tivoli Access Manager for e-Business. We then describe how to technically architect the overall solution into an existing environment and introduce the logical and physical components.

1

Business context

In this chapter, we discuss the business context for Tivoli Access Manager for e-business. After a short introduction to Access Manager for e-business, we describe the common business drivers that influence why and how Tivoli Access Manager for e-business can be implemented in a given business context. Finally, we explain the challenges that you can expect to encounter when confronted with deploying Tivoli Access Manager for e-business in a large enterprise.

1.1 Introduction to Access Manager for e-business

Tivoli Access Manager for e-business provides a policy-based security infrastructure allowing users with varying needs and permissions access to a corporate Web environment. This can mean several things depending on the architecture needed. Authentication of users, control of access privileges, auditing, single sign-on, high availability, and logging are a few essential elements of any security management solution.

In order to understand what Access Manager for e-business does, you need to first understand some common problems that need to be addressed in any organization. The authentication process is complicated when the user accesses information from multiple computers or remote locations over the Internet. Users should be able to authenticate from a Web browser or a wireless device with no client software requirements. In addition, there are often hundreds of Web servers in a large enterprise, with users needing access privileges and in some cases different privileges for each server or application they access. This can lead to many architectural discussions that need to be considered when designing the security infrastructure. The more login IDs and passwords are required, the more help-desk time devoted to answering password calls is needed. In addition, if the authentication is decentralized, administrators are needed to manage the access controls for each individual server or possible application. This also requires additional development work and support from the application team. If the access control is de-centralized, consider how many entries must be added or removed when a user's access privileges change or when an employee joins or leaves the company.

Tivoli Access Manager for e-business addresses these potential problems and others by providing a security solution that:

- Manages access controls for all of the servers and applications centrally.
- Provides users with a single sign-on technology to the Web space that can span multiple sites or domains.
- Simplifies security management by providing a modular authorization architecture that separates security code from application code.
- Provides centralized auditing.

The authorization service in Access Manager for e-business is responsible for managing access controls through the use of access control lists (ACLs), protected object policies (POPs), and authorization rules. Tools such as the pdadmin command-line interface and the browser-based Web Portal Manager provide an administration interface for centrally updating the access controls.

The security entry control point is typically managed by Tivoli Access Manager WebSEAL or the Plug-in for Web servers component. The discussion here revolves around WebSEAL, however, most of the concepts are the same when using a Plug-in for Web servers component.

WebSEAL is a reverse proxy with built-in Access Manager for e-business authorization services. Like a traditional reverse proxy, WebSEAL protects the Web server by intercepting all requests to the back-end Web server and ensures that the data contained in the request is acceptable. If the data is acceptable, the reverse proxy retrieves the requested content from the Web server and forwards it to the original user. In this way, users never directly access any Web server. Like traditional reverse proxies, WebSEAL also provides Web acceleration techniques such as caching and compression and object filtering, which is designed to reduce Web site access times.

WebSEAL enables domain-wide single sign-on by leveraging its central and up-front position, establishing sessions with clients (browsers) that span multiple requests, across multiple back-end application servers. WebSEAL presents the sign-on page, performs authentication and authorization, then establishes a session with the requestor's browser. As a proxy, WebSEAL then connects to the back-end application server relaying the user's request and any identity information needed for back-end authorization.

WebSEAL uses the authentication and authorization services of Access Manager to make sure that the user is authenticated and has the right permissions to access the requested content. Single sign-on can be accomplished across multiple domains or hosts by using a function of WebSEAL either cross domain single sign-on (CDSSO) or e-community single sign-on (ECSSO). We will talk more about ECSSO in the TAMCO case study later in this book.

Access Manager for e-business accommodates a broad range of possible user-authentication mechanisms, including user IDs and passwords, client-side certificates, RSA SecurID tokens, and mobile and wireless identities. It supports e-business configurations common to many of today's enterprises, involving subsets of users requiring their transactions to be conducted in different languages. It also supports both the Wireless Application Protocol and the i-mode protocol. Customers with unique authentication requirements can use the plug-in authentication mechanism that comes with Tivoli Access Manager for e-business.

The customer set for this product includes any company or organization seeking to provide centralized security services for a set of Web-based application services.

1.2 Common business drivers

The predominant computing model today is the Web model. Typically this involves http and https transactions with applications on Web servers, application servers, or both. Many established client-server applications are being converted over to the Web model and virtually all new applications are Web-based. With the rapid introduction of new Web applications, each requiring user authentication and access to confidential information, companies must adopt a security management strategy or risk overwhelming their user population and consequently reducing security.

As a company grows and expands, inevitably more and more Web applications and services are needed. It is common for companies to host Web applications in different geographies, developed by different development groups and supported by different support teams. Unfortunately, it is also very common to find nonstandard login pages, different login IDs for applications, and varying authentication standards. The Web space can easily become disjointed if there are not standard development processes, a standard security model, and cross-team communication.

The trend in the market is for companies to move away from such disjointed views. Unified authentication and authorization functions are essential.

Security management today is driven by multiple initiatives:

- Increased demand for Web-based access to confidential information.

  Companies need assurance that confidential information is protected and proper measures have been taken to prevent intrusion detection.

- Established applications deployed on heterogeneous platforms with limited security enforced.

  Few organizations have the luxury of building their Web infrastructure from scratch. Most companies need tools that can blend new technology with their existing systems to provide security to all resources and applications accessed through the Web.

- Eliminate multiple logins.

  This greatly reduces administration costs and improves user productivity and experience.

- Auditing.

  All attempts to access corporate resources need to be audited to determine if the system is secure. In addition, this is a key element of achieving audit readiness and compliance with such regulations as Sarbanes-Oxley[1].

Sarbanes-Oxley requires companies to maintain and certify the validity of their records and disclosures of pertinent information.

- Faster application deployment.

  If there is an external security service in place, application developers do not have to code security into their applications. Besides improving deployment time, this also delineates application business logic from security enhancements, which is much more efficient for future upgrades.

1.3 Common challenges

Every Access Manager for e-business deployment comes with its own set of unique challenges. Here we list a few common hurdles that need to be considered during the architecture and design of a new Access Manager for e-business deployment:

- Do the users exist today in a single repository or multiple?

  In order to provide a global security policy within any organization, it is a best practice to consolidate and manage the users from within a single repository. These users can be managed independently within Access Manager for e-business by creating multiple domains or subdomains, however, Access Manager for e-business does not support multiple user repositories.

- What is the skill level of the operations team?

  Conduct a subjective analysis of the technical capabilities of the implementation and operational team. In preparing for production readiness, organizations should ensure they have qualified personnel to address changes, troubles, and basic maintenance.

- Are there any namespace migration challenges?

  Most organizations have an established Web site presence. There are namespace considerations that need to be explored during the design to ensure that existing/expected behavior is not broken through this process.

  For example, an organization that has designed their namespace on a redirect model might need to evaluate whether they want to continue with this or migrate to a more consolidated namespace. In a redirect model, the top-level Web site such as `www.mycompany.com` hosts the site navigation content and redirects the browser to a sub-domain such as support.mycompany.com to provide secured content. WebSEAL or the Access Manager for e-business plug-in in this scenario would most likely not

[1] To discover more about the Sarbanes-Oxley Act, visit the following Web site: http://www.soxlaw.com/

be the best candidate to provide redirection since the security is not enforced in redirection.

If during the design you determine that you need to change the namespace navigation model to consolidate behind the top-level domain, such as `www.mycompany.com/support` leveraging WebSEAL at the top-level, then special care is needed to make sure that bookmarks are not broken, performance is not impacted, and cross-geography routing (if any) does not create new security implications.

- Are the existing applications Access Manager for e-business-ready?

  A set of best practices will need to be developed and communicated to the application development teams for ease in migrating applications to an Access Manager for e-business security infrastructure.

  Items to consider for best practice application integration are:

  - Remove all forms logins. Back-end application servers should leverage Basic Authentication, LTPA, or TAI where applicable for ease of deployment and support of single sign-on.
  - Separate unprotected content from secured content by placing them in different directories or URIs. This makes it much simpler to apply the appropriate ACLs and POPs in the objectspace.
  - Develop standards for leveraging dynamic URLs and enable authentication or authorization for each dynamic URL defined.
  - Ensure that any credential or session data needed by the application is placed in a standard, common header for all applications (such as iv_user, iv_creds, and so on).
  - Do not hardcode host names or protocols. Applications should not code the http protocol in any links within the protected pages.
  - If a server relative link is used, the URI should be a net path and not include the protocol, for example, //example.com/myapp/page.html.
  - Applications using JavaScript™ should pay special attention to WebSEAL behavior in regards to relative, server-relative, and absolute URLs. More details on JavaScript considerations can be found in the *IBM Tivoli Access Manager for e-business Version 6.0 WebSEAL Administration Guide*, SC32-1687.

1.4 Conclusion

Access Manager for e-business can provide a wide variety of functions within any organization. It is imperative that the business drivers are fully explored and the unique challenges within each organization are identified in order to produce a flexible yet comprehensive solution that satisfies the business requirements.

The next chapter takes a closer look at how to begin an Access Manager for e-business service engagement.

2

Planning for customer engagement

In this chapter, we discuss the service engagement for Tivoli Access Manager for e-business in general. This chapter is directed towards IT security professionals who assist organizations in deploying Tivoli Access Manager for e-business.

2.1 Services engagement preparation

In this section, we describe resources that are available to help you to successfully deliver a solution.

2.1.1 Implementation skills

Successfully developing and deploying a Tivoli Access Manager for e-business solution requires the following skills:

Note: Skill levels mentioned in the following paragraphs are listed on a scale of 1-5, where 1 stands for least skills and 5 stands for most skills.

General skills:

- Basic operating system administrative skills for AIX®, Solaris™, Windows®, HP-UX, or Linux®
- Web server fundamentals
- Web application server fundamentals
- Reverse proxy concepts and fundamentals
- PKI fundamentals
- Security policy management concepts
- TCP/IP fundamentals
- Security communication protocols
- Networking concepts
- Firewall concepts
- Basic Web page development fundamentals (including security issues)
- Familiarity with industry standard reporting tools
- C, Java™, XML and application server (for example, WebSphere® Application Server) skills

Directory skills (LDAP, Active Directory®, and so on):

- Directory services fundamentals
- Organizing a directory tree
- Understanding of schema modifications and how to communicate those to the directory administrators

Tivoli Access Manager for e-business skills:

- Understanding of Tivoli Access Manager for e-business component architecture
- Understanding of the Tivoli Access Manager for e-business infrastructure communication as well as application communication
- Ability to troubleshoot Tivoli Access Manager for e-business issues

Depending on the target environment, there may be additional skills needed to understand the whole application environment.

2.1.2 Roles and responsibilities

The following are the recommended roles and desired skill level in defining the support requirements of Access Manager for e-business:

- Tivoli Access Manager for e-business administrators
- Application integrators
- Deployment/component specialists
- Solution specialists

Tivoli Access Manager for e-business administrators

- Perform User Creation/Deletion/Reset tasks.
- Perform Group Creation/Deletion/Reset tasks.
- Perform ACL Creation/Deletion/Reset tasks.
- Perform Junction Creation/Deletion/Reset tasks.
- Perform/Verify regular application-specific backups.
- Identify and Level-1 troubleshoot certificate expiration and junction failures.

Prerequisite skills:

- Level 3 familiarity with concepts of user name/password maintenance
- Level 3 familiarity with concepts of access control
- Level 3 understanding of UNIX command line
- Level 3 understanding of Web browser operation
- Level 2-3 understanding of URL composition (host name, path, and parms)
- Level 2 in technical problem identification and data capture (L1 support practices)

Incremental skills:

- Detailed understanding of applicable Tivoli Access Manager for e-business Command-Line Utility / WPM tasks
- User Creation and Deletion
- User lockout/reset
- Password reset
- Access Manager Backup Utility
- Basic understanding of Access Manager solution components and flows
  - LDAP is a store of user data.
  - WebSEALs receive and serve user Web requests.
  - Policy Server handles administration requests.
- Basic understanding of Access Manager nomenclature
- Junctions, ACLs, certificates, and SSL
- Detailed understanding of in-place password policies

Application integrators

- Work with application developers to test and board applications.
- Maintain Development Guidelines.
- Troubleshoot application Issues.
- Specify and assemble application boarding requirements, including junctions, ACLs, certificates, dynamic URL settings, and firewall rules.

Prerequisite skills:

- Level 4-5 in HTML and JavaScript
- Level 4-5 in one or more Web application platforms (WebSphere Application Server, IBM HTTP Server, and Domino®)
- Level 3 in SSL and certificate concepts
- Level 3 in technical troubleshooting
- Level 2-3 in TCP/IP addressing
- (optional) Level 3-4 in back-end Web development technologies (JSP™ and so on)
- (optional) Level 3-4 in IBM Web application development standards and guidelines

Incremental skills:

- WebSEAL page translation behavior
- Junction options and effects
- WebSEAL application integration guidelines
- WebSEAL tracing

Deployment/component specialists

- Contribute insight into the operation of specific components.
- Provide detailed information about technical and operation issues.
- Identify and maintain monitoring probes for processes and components.
- Monitor logs and support troubleshooting activities.
- Plan and execute patch and maintenance activities.
- Define or develop tools for maintenance, administration, and monitoring (in conjunction with Administrators and Solution Specialists)

Prerequisite skills:

- Level 4-5 in UNIX
- Level 4-5 in one or more of LDAP, Tivoli Access Manager for e-business, and network configurations
- Level 4-5 in technical troubleshooting
- Level 3-4 in hardware, software, and application deployment standards and practices
- Level 3-4 in component-specific backup and restore activities
- Level 3-4 in TCP/IP and DNS
- Level 3 in scripting (shell or Perl)

Incremental skills

- Detailed understanding of Tivoli security architecture
- Cross-training in other component skills (LDAP, Tivoli Access Manager for e-business, HACMP™, and load balancing)

Solution specialists

- High-level understanding of overall solution
- Understanding of inter-component communication
- Interface to Tivoli Support for severe problems
- Ongoing overall solution QA

- Backup integrity
- Monitoring effectiveness
- Scaling and capacity planning
- Defining or developing tools for maintenance, administration, and monitoring (in conjunction with administrators and deployment specialists)

Prerequisite skills:

- Level 5 in technical troubleshooting
- Level 4-5 in Tivoli Access Manager Components and Data Flows
- Level 3-4 in high-level TCP/IP, LDAP, HTTP, and SSL/Certificate concepts and activities
- Level 3-4 in UNIX
- Level 3-4 in high-availability and fault-tolerance concepts and practices
- Level 3 in scripting (shell or Perl)
- Level 2-3 in inter-SDC network communication concepts and practices

Incremental skills

- Detailed understanding of Tivoli Access Manager and related component architecture

2.1.3 Available resources

The prerequisite skills that we list in the previous section are those needed to customize or develop the solution. For each of these skills, there are a variety of resources available to help acquire the necessary skill level. The educational resources available are:

- Online Help

  Tivoli Access Manager for e-business provides online help and product manuals at the following Web site:

  http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliAccessManagerfore-business.html

- IBM Education Assistant

  IBM Education Assistant is a collection of multimedia educational modules designed to help you gain a better understanding of IBM software products and use them more effectively to meet your business requirements. Modules consist of the following types of content:

  - *Presentations* (many with audio) provide an overview of a product or technology or a more in-depth look at a particular product component or feature. Presentations are available in both Flash and PDF formats.
  - *Demonstrations* show you how to complete a specific task or configuration (in Flash format) and provide background information to help you understand the options available.
  - *Tutorials* provide instructions and all the files necessary to complete a practice lab scenario in your own environment.
  - *Additional resources* provide links to relevant external content.

  The IBM Education Assistant can be found at:

  `http://www.ibm.com/software/info/education/assistant/`

- Classroom Training

  IBM PartnerWorld® provides current information about available classes, their dates, locations, and registration.

  Additionally, check the Partner Education Web site, which serves as a single point of contact for all Business Partner education and training.

- IBM Technical Education Services (ITES)

  ITES offers a variety of classes at all knowledge levels to help you achieve any of the offering's prerequisite skills.

- IBM Redbooks

  You can access various practical and architectural information regarding IBM hardware and software platform from IBM Redbooks publications. An assorted list of applicable IBM Redbooks publications is listed in “Related publications” on page 233. You can download PDFs of IBM Redbooks publications from the following Web site:

  `http://ibm.com/redbooks`

- Problem-Solving Resources Online for Tivoli Access Manager for e-business administrators

  The Access Manager for e-business technical support team has compiled a list of online resources containing documents and information to aid in system administration of Access Manager for e-business. You can access the full Access Manager for e-business online knowledge inventory, subscribe to RSS feeds, download tools, and much more at:

  http://www.ibm.com/support/docview.wss?uid=swg21295179

2.2 Services engagement overview

You routinely rely on your skills and previous experience as a guide, but there are always some issues that might require some educated guesswork. The section provides information to help you minimize the guesswork that is involved in planning and implementing a solution by providing a framework and time estimates for the major tasks.

A typical services engagement consists of:

- Building an executive assessment
- Setting up a demonstration system or proof of technology
- Analyzing solution tasks
- Creating a contract known as a *statement of work*

The representative tasks and the time involved for custom solution execution are included in the following section. Because each customer has a unique set of needs, the actual set of tasks to accomplish and the time involved can vary. However, this list should help you understand the implementation details, size the solution more accurately for the customer, and ensure a profitable engagement for yourself.

It is important to work with your customers to understand their expectations. When you have gathered this data, document the tasks, deliverables, and associated costs in a *statement of work*. The statement of work acts as your contractual agreement with the customer for the duration of the project. Therefore, a detailed and well-defined statement of work is advantageous both to you and to your customer.

A good overall understanding of the solution scope is a crucial prerequisite to successfully selling, developing, and implementing it. As a solution provider, you must understand what is involved in developing such a solution before you can discuss it with your customer and size it for a cost estimate.

2.2.1 Executive Assessment

The *Executive Assessment* is a billable service that you can offer to your prospective clients. It offers a process designed to help you evaluate the business needs of a company that is planning to deploy a solution for e-business. It was created for IBM Business Partners to help you close a higher ratio of opportunities. It has been field-tested in markets all over North America and Europe and has received enthusiastic feedback.

The benefits of using the Executive Assessment in your sales process include:

- Earning additional service fees
- More effectively qualifying prospective clients
- Shortening the sales cycle
- Streamlining the development process
- Closing a much higher ratio of potential engagements

This toolset helps you ask the right people the right questions so that you get the information that you need to propose the appropriate solution. This assessment then helps you create a compelling business case that will persuade your prospect to buy the required hardware, software, and services from you in the shortest possible time.

This is a business-case assessment, not a technical assessment, so your audience should be business owners, line-of-business executives, marketing and sales managers, and finally, the IT manager. The business owner or line-of-business executive is likely to be the decision maker.

For their initial investment, your clients get:

- A business assessment prepared by a professional (you)
- A competitive analysis
- A prototype solution for their review
- A strategic and tactical proposal for justifying and implementing their solution for e-business

Over the course of the Executive Assessment, you determine who will be involved in the project, what they want to accomplish, when they plan to deploy, what plays a mission-critical role in their business, and how the project will be funded. Armed with this information, a competitive analysis, and a prototype solution, you will be able to justify their investment, build perceived value, present your recommendations in a way that is almost irresistible, and successfully close the contract.

Having the ability to recommend the correct course of action to your client has tremendous value. In a market where it is difficult for companies to find qualified Business Intelligence consultants, the Executive Assessment and resulting presentation gives you a chance to prove conclusively that you have the right technology and the right people to do the job.

2.2.2 Demonstration system setup

A demonstration system is typically set up in advance to show your customers the attributes of the solution. The demonstration system can be set up with a limited number of machines that are obviously separate from the system that will be used in production.

You can set up Tivoli Access Manager for e-business on a mobile computer that meets the minimum hardware requirements. It is common to set up several VMWare images in advance of the customer visit, which together demonstrates various server types and possible configurations of Access Manager for e-business.

The demonstration system allows your customers to not only see the software in action but also to evaluate whether the solution suits their particular needs. The starting point is assumed to be a VMWare image with the operating system and directory server installed (such as Tivoli Directory Server or Microsoft® Active Directory). The tasks of demonstrating a solution are shown here:

- Install and configure the Policy Server and Policy Proxy Server.
- Install and configure a WebSEAL server (or Plug-in).
- Install and configure the Web Portal Manager.
- Install and configure a test application tailored to represent the clients Web server environment (leveraging IBM HTTPS Server (IHS), WebSphere Application Server, or a local WebSEAL test application).
- Install and configure the Common Auditing and Reporting Services (CARS) server.
- Configure trust authentication using junctions.
- Assign users to groups with associated Access Control Lists.
- Demonstrate authentication leveraging forms or basic authentication (BA).
- Demonstrate log reporting/auditing capabilities.

2.2.3 Analyze solution tasks

After the customer has agreed to move forward with an Access Manager for e-business deployment in their environment, you need to decide what effort you must perform to implement it. These estimates are then collected and implemented into a contract or *statement of work*. We discuss these tasks in detail in 2.3, “Defining solution tasks” on page 23.

The tasks listed are a suggested list only with a suggested order. You might complete the tasks in a different order or might omit or add tasks depending on the environment in which you implement the solution. The overall success of the tasks and the required time can be influenced by the amount of skill and experience that you or your team have on the solution.

For the detailed task break down, see 2.3, “Defining solution tasks” on page 23.

2.2.4 Creating a contract

A contract or *statement of work* (SOW) is a binding contractual agreement between you and your customer that defines the service engagement that you must perform and the result that the customer can expect from the engagement. The contract should leave nothing in doubt.

This section helps you assemble the SOW. An example of a possible statement of work can be found in Appendix A, “Statement of work” on page 209.

A statement of work needs to include the following information:

- Executive summary of the solution, which is typically a short (less than a page) summary of the solution and its benefits. The executive summary describes the goals and intentions of the work scope. This may include a high-level overview of the project and the services necessary to meet the business requirements. Typically, the proof-of-concept as well as the production deployment are highlighted as deliverables in the executive summary. The executive summary is also a good place to introduce the methodologies that will be leveraged throughout the duration of the project. For example, IBM Global Services Method (GS Method) is the center point in the IBM Global Service project development approach. Use of the GS Method allows for a formally structured engagement with defined tasks, work products, and deliverables.You must specify any major restriction of the implementation, such as:
  - The solution is only implemented for internet-facing Web applications.
  - The solution will be implemented in phases.

- Project scope, which includes the major components and solution building blocks that will be implemented. It should cover the conceptual architecture of the solution and solution scope in general. This description is aimed at technical personnel to understand the implementation scope.
- Assumptions, which lists all the assumptions that are used to prepare the contract and provide task estimation. Any deviation to the assumptions that is used will definitely impact the scope of engagement and must be managed using the change management procedure. Typical changes would include cost changes or scope changes.
- Business Partner responsibilities, which lists all the responsibilities or major tasks that will be performed by you or your team to implement the solution.
- Customer responsibilities, which lists all the responsibilities or items that the customer must provide for you or your team to perform the engagement. If you cannot obtain any item in the customer responsibilities, then a Change Management procedure can be invoked.
- Staffing estimates, which lists the estimated personnel that must implement the solution.
- Project schedule and milestones, which shows the major steps, schedule and achievement calendar that can be used to check the project progress.
- Testing methodology, which lists the test cases to ensure that the project implementation is successful.
- Deliverables, which provides tangible items that the customer will get at the end of the service engagement, including:
  - Machine installation
  - Documentation
  - Training
- Completion criteria, which lists the items that, when provided to the customer, indicates that the engagement is successfully completed. For most of services engagement, this is probably the most delicate to define. Completion criteria can be too general so that you will be tied up with providing the customer on-going support for life. Alternatively, an inadequate completion criteria is often rejected by the customer fearing that you might back away from the engagement in an incomplete state.

We provide a sample statement of work in Appendix A, “Statement of work” on page 209.

2.3 Defining solution tasks

The key to a profitable services engagement is to identify the tasks that you must perform correctly and to allocate the necessary time to perform them. This section guides you on the tasks that you may need to perform for a Tivoli Access Manager for e-business solution implementation.

Your estimates for timing may not only depend on component selection driven by the business requirements but may also largely depend on the following factors:

- How many Access Manager for e-business components and resource managers are required?

  The core components of a user registry and Policy Server are required for a minimum installation. However, there are many various resource managers, such as WebSEAL and the Web server plug-in, and additional server types that may be needed as part of the corporate solution. Each of these need to be evaluated separately for tasks associated for deployment and sizing estimates.

  Optional components, such as the Common Auditing and Reporting Service (CARS) and the Session Management Server (SMS), both run as services of the IBM WebSphere Application Server. These technologies in some cases may require different skillsets to support and configure and will most certainly add time and increased complexity to the project.

- Will the deployment span multiple geographies?

  The complexity of developing a security solution greatly increases when defining cross-geographic flows, bandwidth requirements, failover requirements, networking requirements, single sign-on settings, and so on.

- Are multiple management domains necessary for separating and delegating administration of users, roles, and security policy?

  Delegated administration again increases the scope and complexity of supporting and configuring a security management infrastructure. Time needs to be dedicated to developing processes for partitioning the objectspace and defining tasks and permissions associated with the administrators of the shared policy environment.

The next section provides a description of the necessary tasks required for a Tivoli Access Manager for e-business deployment. Before we describe the tasks, however, we make the following general assumptions:

- You have a dedicated customer engineer that is available for the duration of the project.
- You have identified the pilot environment and defined the test criteria for the solution. In addition, the customer has signed off on the pilot environment and test criteria.
- You have set up user IDs and physical access for consultants prior to the kick off meeting.
- Documentation for the solution will be done off-site.

2.3.1 Deployment tasks

This section lists the required tasks for a Tivoli Access Manager for e-business deployment. You can use these tasks when creating a statement of work:

- Interview the personnel responsible for the Web application environment and systems administration to understand the environment Access Manager for e-business will be deployed in.
- Interview the personnel responsible for the existing directory structure or those responsible to determine the user directory structure definition and design.
- Detail the requirements.
- Detail the design of all Tivoli Access Manager for e-business components and architecture.
- Design the configuration of Web server junctions.
- Design the Web space.
- Design the protected object space.
- Design the access control lists and protected object policies.
- Design the user directory structure and Tivoli Access Manager for e-business user groups.
- Design the auditing and reporting systems.
- Design the test plan and test cases.
- Manage the deployment and communication.
- Document the project.
- Facilitate knowledge transfer and customer acceptance testing.

- Set up optional administrative training.

2.4 Conclusion

In this chapter, we have addressed the necessary prerequisites for the service engagement preparation by specifying the implementation skills, the different roles and responsibilities, and the required resources that are involved in or required for the engagement.

We have broken down the service engagement and looked at details for an Executive Assessment and a demonstration system setup. We also analyzed the solution tasks and discussed a good way to create a contract.

Finally, we listed the typical solution and deployment tasks.

With this information, you should be able to engage in a Tivoli Access Manager for e-business deployment.

Part 2

Customer environment

In this part, we take you through an example company profile with existing business policies and guidelines and build an access control solution design for this particular environment.

We then describe how the new access control components can be integrated into the existing environment. We then explain how to execute the access control integration tasks that must be implemented in order to create a fully functional end-to-end solution.

3

Company profile

Note: The company profile we use in this book also exists in the IBM Redbooks publication *Deployment Guide Series: IBM Tivoli Identity Manager*, SG24-6477. Looking at an overall time line, TAMCO's first IT security project is the deployment of the access control solution described in this book.

Established in 1967, Tamminen, Auramo, Makinen, & Co. (TAMCO) is currently one of the leading producers of sauna equipment in Europe. Originally, TAMCO operated only in Finland, but over the past decade, following Finland's entry into the European Union (EU), the company expanded its operations into the wider EU marketplace.

TAMCO started branches first in Germany in 1995, followed by the United Kingdom (UK) two years later. As their export revenue soared, each of these branches started smaller sales office in neighboring countries. Soon, sales representatives based in Germany were operating in Austria, Switzerland, France, Belgium, The Netherlands, and the Czech Republic. The UK office handles Ireland, as well as export business to Canada and Australia. The main offices in Helsinki are responsible for sales in the other Nordic countries and Russia.

By 1999, the company had grown from a small specialty firm, run by two carpenters and a blacksmith, into a multinational company with an annual turnover in the tens of millions Euros and hundreds of employees Europe-wide.

This growth has not come without problems. First, as they prepared for Y2K, TAMCO discovered that the absence of a centralized IT strategy at the Headquarters IT department meant that none of their branch offices had followed a standard build strategy for their IT infrastructure. Second, a number of the sales IT systems were earlier customized applications, while in other departments, modernization had produced a loosely organized and poorly secured set of Web-based applications.

The combination of the Y2K fixes (and associated costs) and the rapid adoption by the post-Y2K market of Web-hosted e-business convinced TAMCO's senior management that a new IT strategy built around a unified system providing customized Web-based services was mandatory.

In the first phase, TAMCO moved all their earlier custom applications onto standards-based Web-enabled application servers. They also added a centralized HR management system based on PeopleSoft® and a customer relationship management system from Siebel®. A centralized e-mail system built on Lotus® Domino replaced the home-built e-mail system in 2000. Lastly, a key element of the new strategy was to develop a company-wide directory service based on LDAP standards to provide a centralized white pages functionality that could be locally replicated for each major sales office location.

3.1 Business drivers and capabilities

Why did TAMCO come to decide they needed a centralized access control solution? Following the decision by TAMCO's senior management that a new IT strategy was needed, an internal TAMCO team was assigned to analyze the market-leading products in the identity management space. The TAMCO internal team conducted a trade study in which they developed a checklist of key capabilities against which they could measure these products, covering a broad range of topics from platform support to product architecture to a range of security and provisioning capabilities. The trade study resulted in two product sets that provided essentially equivalent capabilities, one led by a consortium of allied vendors, and one from IBM involving cross-pillar products from WebSphere, Tivoli, DB2®, and Lotus.

In order to test the claims made by each party, the study team conducted a Proof-of-Concept (PoC) with each of them. The PoC was designed to show the strengths and weaknesses of each solution set with respect to 11 key business drivers that TAMCO identified as essential to their internal and external business viability.

The 11 business drivers are:

- Asset value
- Legal/regulatory requirements
- Time-to-market
- Simple to use
- Risk tolerance
- Complex organizational environment
- Mission critical/availability
- Protecting the corporate image
- Complex IT environment
- Complex system
- High risk IT environment

The results of the PoC phase led TAMCO to choose the IBM Integrated Identity Management solution, a key component of which is the centralized authentication and authorization framework provided by IBM Tivoli Access Manager.

Key findings from the PoC showed that, among other positive results, adopting a centralized access control framework based on Access Manager would address the following key business drivers:

- Augment sales turnaround by allowing TAMCO to extend their IT environment more rapidly to upgrade to better sales and marketing applications without having to redo the security infrastructure.

- Improve time-to-market by relieving the small team of TAMCO developers from having to write security into applications.
- Reduce the overall risk TAMCO would incur by adopting the Web-based new IT strategy.
- Provide a strong story to regulators and TAMCO insurers with respect to their use of best security practices for authorization and auditing.
- Ease the task of extending their IT environment to include IT environments of suppliers and future acquisition partners.

Security capabilities that were identified and proven in the PoC were:

- Single sign-on to the TAMCO portal and back-end applications.
- Fine-grained authorization capability for portlets and JSPs at not only the URI level but also at the method level.
- A standards-based common auditing system plus the ability to generate standard and custom reports.
- Consistent application of access control policy across the TAMCO business units.

3.2 Current IT environment

The overall information technology environment that resulted from the new IT modernization includes the products and organization discussed in this section.

3.2.1 Organization

TAMCO offices are located in three countries: Finland, Germany, and UK, each of which is organized into the following structure:

- Sales
- Marketing
- Finance
- HR
- Distribution

Since the sauna equipment is manufactured only in Finland, the Finland office also includes a *manufacturing* department.

In addition, the Germany and UK offices include their own IT environment based on a central IT department in Finland, and each office administers its own local application set. Each of the departments is connected by a central TAMCO

intranet network. A standard set of applications is made available to each of the departments in each of the country offices through a WebSphere Portal. Each department in each country office is provided with the following applications as a part of the standard build:

- Lotus Notes® e-mail system hosted on WebSphere Portal as an iNotes™ portlet
- Human resources (HR) management by a PeopleSoft Application Server deployed on WebSphere Application Server
- Customer relationship management by a Siebel portlet hosted on the WebSphere Portal
- WebSphere Application Server

3.2.2 IT architecture

The current IT solution consists of three main components in each geography:

- An LDAP directory, which will also serve as the user registry for Access Manager, plus an LDAP client for remote access to the LDAP directory server. The master LDAP is centrally managed from Finland and replicated to the other two countries.
- A Lotus Domino mail system. This is centrally managed from Finland, but replicated to the offices in the other two countries.
  - Because TAMCO is an internationally operating company, the policy for the Domino users is to use English alphabet characters in their names only.
  - Access to the mail environment is implemented through WebSphere Portal using the iNotes portlet. This way everybody can access their e-mail through a regular Web browser without needing a fat client.
- WebSphere Portal servers and the application servers behind them. The WebSphere Portal servers and application servers do not require any administration from the user management point-of-view.

Figure 3-1 shows the IT system components in each country.

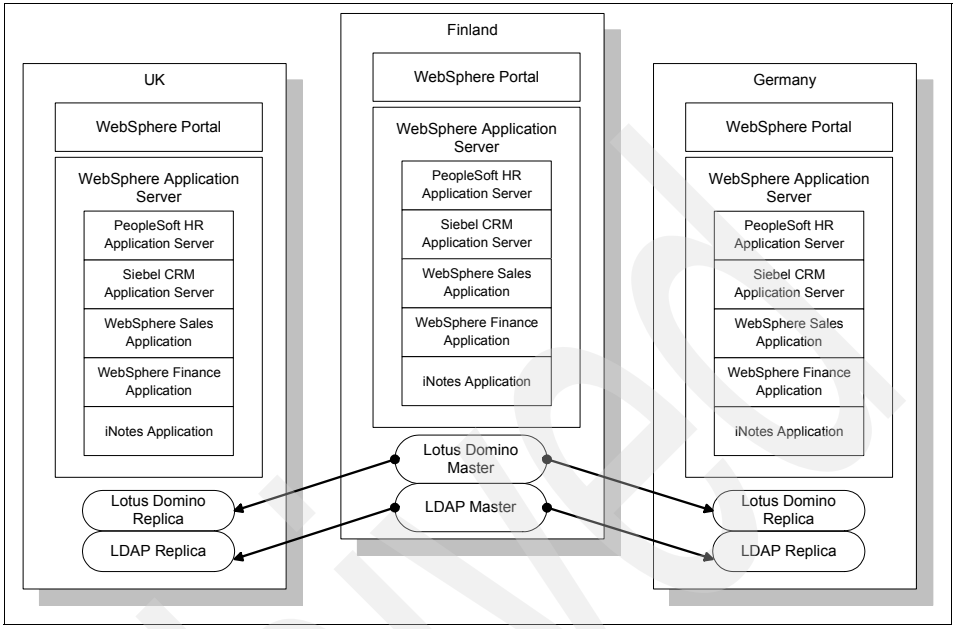

Figure 3-1 IT products by country

3.3 Conclusion

After we introduced the company profile of our case study, we discussed the business drivers for the targeted access control solution. Finally, we examined the current IT environment with its organization and IT architecture.

It was concluded that the TAMCO access control and management functions can be accomplished in the following manner:

- IBM Tivoli Access Manager for e-business controls access to the Web space using user accounts and group profiles. IBM Tivoli Access Manager will provide a global sign-on (GSO) resource to provide automatic forms based on single sign-on to all Web applications within TAMCO.
- The existing LDAP directory (IBM Tivoli Directory Server) will serve as the user registry for Access Manager.
- IBM Tivoli Directory Integrator will function as an Identity Manager endpoint, allowing for provisioning services to create, change, and delete user accounts defined in the PeopleSoft HRMS administration tool and password synchronization.

- IBM Tivoli Identity Manager will be the single point of management for all user accounts in this environment. Deployment considerations of Tivoli Identity Manager can be found in the IBM Redbooks publication *Deployment Guide Series: IBM Tivoli Identity Manager*, SG24-6477.

The following chapter takes a closer look at the solution design.

4

Solution design

In this chapter, we develop the TAMCO access control security policy and present the design for the Access Manager system to enforce it.

The key elements in this design are the management domains, the logical and functional components of Access Manager, and an overall security reference architecture to show how the Access Manager components fit. Detailed explanations of the context for this reference architecture can be found in section 2.1, “Common Security Architecture Subsystems,” in *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014. Details about the components and architecture for Tivoli Access Manager can be found in Part 2, “Managing access control,” in the same IBM Redbooks publication.

4.1 Defining the access control security policy

When planning the access control security capabilities, four important questions must be addressed prior to any deployment. The answers to these questions may already be available within the *corporate policy* for the organization.

Attention: *Policy* is a very common term and in many products you will find specific *policies* sections. These are the product-related policies that are covered in the practice or procedure documents. The *corporate policy* is not related to products and is a high-level document. Throughout this book, we are mostly referring to a certain portion of the policy concerned with access control guidelines and regulations.

The questions are:

- How many secure domains are needed?
- What objects are to be secured?
- What actions are to be permitted on these objects?
- What actions can users take?

Enforcing an access control security policy requires an understanding of the flow of access requests through the environment. The planning should identify proper roles (groups) and network topology (firewalls, routers, any subnets, servers, and so on). Deploying a Tivoli Access Manager security environment also requires identifying the optimal points within the network for installing software that evaluates user access requests, and grants or denies the requested access. *Optimal* means balancing trade-offs between performance, operations, and efficiency. Implementation of an access control security policy requires that you understand the quantity of users, data, and throughput that your network must accommodate. You must evaluate performance characteristics, scalability, and the need for failover capabilities.

Access Manager ships with a default access control security policy that protects all objects in a default secure domain. A set of administrative users and groups is established and granted a predefined set of permissions. This default security policy is an adequate starting point for defining the TAMCO access control security policy.

4.1.1 Secure domains

The default access control security policy that is provided on installation of Tivoli Access Manager for e-Business includes the creation of a default secure domain, called the *management domain*. This domain is used to manage the security policy of all domains and is available for managing other protected resources as well. In large organizations, however, you might want to define two or more domains. Each domain is given a name and is established with a unique set of physical and logical resources.

The security administrator can define the resources in a domain based on geography, business unit, or major organizational division within the enterprise. The security policy defined in the domain affects only the resources in that domain, which allows data to be partitioned and managed completely independently.

The choice of a single or multiple domains is primarily influenced by ease of administration and the administration practices of the staff. The easiest to administer is the single domain because of the economy of scale achieved with a many-to-one administration model applied to all resources and users within a single logical boundary.

Unfortunately, there are a number of factors that complicate this ideal picture. Scalability and availability concerns may dictate more than one domain, in part because of the desirability of local administration and in part because of network latency issues. A requirement to delegate administration so that no one individual or set of individuals has complete control over the environment is a factor in organizations that are worried about disgruntled employees or that need to minimize or remove the risk of a complete loss of service.

Questions to ask when it comes to deciding on a single or multiple domain structure are:

- How dispersed is my organization and what is the latency of my network?
- How diverse are the business units in terms of operations, equipment, platforms, and applications?
- How large is the user population and how dynamic are the roles and concurrent job functions? Can it be represented in a single large group or a few large groups, or in many small groups?
- How many different units within my organization do not share common applications or infrastructure?

A complete explanation of Access Manager's default security policy can be found in Chapter 4, "Default security policy", in *IBM Tivoli Access Manager Version 6.0 Administration Guide*, SC32-1686.

TAMCO has determined that each country's location should be locally managed and administered. Therefore, the IT environment in each country office will comprise its own Access Manager secure domain. Access Manager supports the ability to provide centralized authentication and authorization services to multiple domains.

More technical details about this domain and other accompanying specs are discussed in 4.2, "The TAMCO deployment" on page 45. For the purposes of this section, the properties of this secure domain can be carried over to the other two TAMCO domains for Germany and the UK.

4.1.2 Objects to be secured

The objects to be secured are the *resources* (typically servers and applications that reside on them) in the TAMCO intranet. These are referred to as *protected objects*. A *protected object namespace* is defined and administered by Access Manager. It provides a tree structure on which central authentication and authorization management functionality operate. Permitted actions are attached to the objects. The *policy* is expressed as *access control lists* (ACLs), which contain ACL entries that correspond to actions. Users of the system are defined and managed based on the permissions they have been assigned to request and access resources.

The objects to be secured in the TAMCO environment can be categorized as follows:

Servers WebSphere Application Server, WebSphere Portal, Lotus Domino, Access Manager Policy Server, Access Manager Authorization Server, Tivoli Directory Server, PeopleSoft Application Server, and Siebel server

Applications WebSphere portlets (Finance portlet), iNotes, PS/HRMS application, and Siebel Customer Relationship Management (CRM) application

At the time of installation, Access Manager will creates two default objectspaces: the Management objectspace (indicated as /Management) and the /WebSEAL objectspace.

4.1.3 Permitted actions

The actions that are permitted are determined by the access control security policy. For ease of administration, we adopt an approach to organize the resources to be managed into groups, and attach permissions to these groups. When a decision to authorize a request for access to a resource is made, the access decision compares the permissions that the user making the request has against the permissions for the group.

Access Manager for e-business ships with a set of default ACL policies appropriate for a generic three-tier Web application model. The default ACLs are default-root, default-management, default-config, default-gso, default-policy, default-domain, and default-management-proxy. These ACLs contain different action bits appropriate to each context. They are intended to provide a base security policy as a starting point.

The primary action group contains the action bits that are most likely to be needed for a typical Web-enabled three-tier environment. The primary action group is suitable for carrying out the authorizations needed to enforce the TAMCO security policy.

Note: A domain administrator can add or create ACL entries and manage the ACL policies by adding, removing, and modifying the action bits in the ACL policies. An ACL entry defines a user or group and which actions each can perform on a protected object before or after the ACL policy is attached to domain resources. Any change to the ACL entry affects only the access that these users and groups have against a specific domain resource to which the ACL policy is attached. For more information about ACL entries in an ACL policy, see Chapter 8, “Managing access control”, of *IBM Tivoli Access Manager Version 6.0 Administration Guide*, SC32-1686.

In addition to ACLs to express policy, Access Manager provides two other options: the *protected object policy* (POP) and *authorization rules*. A POP is a policy that is universally applicable or does not vary (such as *time of day* or *quality of protection* (SSL)). Consequently, it is powerful and needs to be used carefully. The authorization rules are boolean expressions that can express policy constraints, and which are evaluated by the Access Manager rules engine. Neither POPs nor authorization rules are needed to enforce the TAMCO security policy.

Access Manager provides a standard set of *permissions,* which are available once the Access Manager Authorization Server is installed and configured. The permissions correspond to *actions* such as add, execute, read, delete, and so on, which are represented by *action bits* (symbolized by uppercase and lowercase English letters). When Access Manager is installed and configured a *primary action group* is created that includes 18 default actions. An action group is a set of specific action bits.

Since the actions that users can perform depend on the groups they belong to, the ACLs on those groups, and the entries/action bits in those ACLs, it is necessary to establish the ACLs for each of the TAMCO groups. In the TAMCO case, there are five main groups whose members come from all three countries.

There is an additional group only in the Finland office (manufacturing), whose members are only the employees involved in sauna manufacture:

1. Finance
2. Marketing
3. Sales
4. Human Resources
5. Distribution

In addition, there will be three universal groups: administrators, employees, and executives. Administrators and executives will have additional access permissions, while employees will have more restricted permissions.

4.1.4 TAMCO access control security policy

Based on the above, the TAMCO overall access control security policy can be expressed in terms of the following two categories: *groups* and *resources*. Let us take a closer look at what makes up the groups and resources.

Groups

All *employees* of TAMCO will have basic read, write, and execute permissions to:

- Public information
- All employee-confidential information
- The iNotes Web-based e-mail client

Members of the *finance group* have read, write, and execute permissions to:

- The finance application portlet
- Read permission to the Siebel CRM application and finance application

Members of the *marketing group* have read, write, and execute permissions to:

- The Siebel CRM applications

Members of the *sales group* have read, write, and execute permissions to:

- The Siebel CRM application
- The sales pricing application (Portlet)

Members of the *human resources group* have read, write, and execute permissions to:

- The PeopleSoft Human Resource Management System (HRMS) application (deployed in WebSphere Application Server as a plug-in application on the PeopleSoft Application Server)

Members of the *manufacturing group* have read, write, and execute permission to:

- The design and process control applications

All TAMCO *executives* have read, write, and execute permissions to all TAMCO applications except the PeopleSoft HRMS, to which they have read-only permission.

Members of the *administrators group* have the full set of action bits included in the primary action group. Administration for each of the secure domains will be delegated, with a management ACL for each domain (for example, management-Germany and management-UK). These delegate administrators can then perform all the actions provided for in the primary action group, but only on the users in their assigned domain.

Resources

The goal of any security policy is to adequately protect business assets and resources that need to be protected. These could be any type of data object, such as files, directories, network servers, messages, databases, or Web pages. Then you must decide what users and groups of users should have access to these protected resources. You also need to decide what type of access to these resources should be permitted. Finally, you must apply the proper ACL policy to these resources to ensure that only the right users can access them.

The following are the resources in TAMCO:

Servers WebSphere Application Server, WebSphere Portal server, Siebel server, Lotus Domino server, PeopleSoft server, Access Manager Policy Server, and WebSEAL server.

Applications Financial portlet, PeopleSoft HRMS, Siebel CRM, and iNotes. (Access control for these applications will be based on the URI.)

Users Every employee of TAMCO is provisioned into a specific group representing his job role. The relevant ACL policy is then attached to the groups. Every employee is also a member of the group *employee*.

Table 4-1 shows the domain users and the groups they belong to. Table 4-2 shows the TAMCO groups with their associated ACLs and permissions.

Table 4-1 TAMCO users and groups

Finland domain users	Group	Germany domain users	Group	UK domain users	Group
Haari	finland-admin	Axel	germany-admin	Joe	uk-admin
Piia	HR/Finland	Gerhard	HR/Germany	Peter	HR/UK
Stiina	Sales/Finland	Oskar	Sales/Germany	Jill	Sales/UK
Jani	Marketing/Finland	Anna	Marketing/Germany	Jack	Marketing/UK
Teemo	Finance/Finland	Peer	Finance/Germany	Helen	Finance/UK
Sven	Distribution/Finland	Frank	Distribution/Germany	Fiona	Distribution/UK
Tommi	Executive/Finland	Angela	Executive/Germany	Blair	Executive/UK
Juha	Manufacturing/Finland				

Table 4-2 Groups with associated ACLs and permissions

Group	Access control list	Permissions
iv-admin	default-management	TcmdbsvaBtNWAR
Employee	default-root	T r
Sales	sales	T r x a m
Marketing	marketing	T r x a m
Finance	finance	T r x a m
Manufacturing	manufacturing	T r x a m

Group	Access control list	Permissions
Executive	default-management	T r
ivmgrd-servers	default-management	T s v g

4.2 The TAMCO deployment

The computing environment on which Tivoli Access Manager enforces access control security policies for authentication and authorization is called a *secure domain*.

The initial secure domain, called the *management domain*, is created when you install and configure the following Access Manager components:

- The *policy server* maintains the *master authorization database* for the management domain. In addition, it updates distributed authorization database replicas and maintains location information about other distributed Access Manager services.
- The *registry* provides a database of the user identities known to Access Manager. It also provides a representation of roles as Access Manager groups that are populated with users. The registry is typically implemented as an LDAP directory; however, other registries are supported, such as Microsoft Active Directory and IBM Lotus Domino.

In the TAMCO deployment there are three secure domains, one for each of the three *hub* countries of Finland, Germany, and the UK. The *management domain* is the computing environment encompassing the Finland IT environment. There are separate secure domains for Germany and the UK, each of which will have its own Access Manager Policy Proxy Server. The directory solution is being implemented using IBM Tivoli Directory Server. The master directory is located in the Finland central domain and local replicas of the registry will be configured in the Germany and UK offices, replicating against the master in order to achieve better performance and redundancy.

TAMCO already uses an LDAP directory to store user and group data; however, it has not been designed to be an enterprise directory. This directory is sufficient, however, to serve as the authoritative source for user identity and group membership information. Prior to configuring the Access Manager solution, we created an LDIF file (Lightweight Directory Interchange Format), which is an ASCII formatted file used for data integration and synchronization between LDAP directories. This file contains information about the currently existing users and their group memberships that we will use when we prepare the Access Manager

user registry for the initial user population in 6.1.1, “Configuring and populating the Access Manager user registry” on page 124.

When the LDIF file was created a script was run adding a new attribute of “gso-user” for each user in the LDIF file. We will use this attribute when we import these TAMCO users into the Access Manager user registry for purposes of configuring Access Manager’s single sign-on.

4.2.1 Network structure of the secure domains

Structurally, each of the secure domains will be organized as shown in Figure 4-1 on page 47, which depicts the network zones as they are described in IBM Method for Architecting Secure Solutions (MASS[1]). This structure provides a starting point for understanding where and how to deploy the IT components to satisfy the TAMCO access control security policy. It is important to the design to determine which resources should go into which network zone: uncontrolled, controlled, restricted, and secured.

Important: The design of network zones for a particular deployment does not prescribe a finite solution like the depicted zones in Figure 4-1. Some IT architectures might require more segregated zones, while others prefer a simpler solution with less network traffic control.

Direct access from an uncontrolled zone (like the Internet) to resources deployed in the controlled, restricted, and secured zones presents a significant security exposure. For this reason, back-end components are usually placed in an internal network “firewalled” from the Internet and the DMZ. The DMZ should only contain elements like the HTTP server or reverse proxy components exposed to direct browser access. Web applications and data are best placed within a restricted zone providing multiple protection mechanisms. This double-firewall architecture has become common, not only for controlling Internet application access, but increasingly for controlling access to critical computing resources by internal users as well.

[1] MASS has been developed and is used by IBM Global Service practitioners. More details about MASS can be obtained from Appendix A, “Method for Architecting Secure Solutions“, in *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

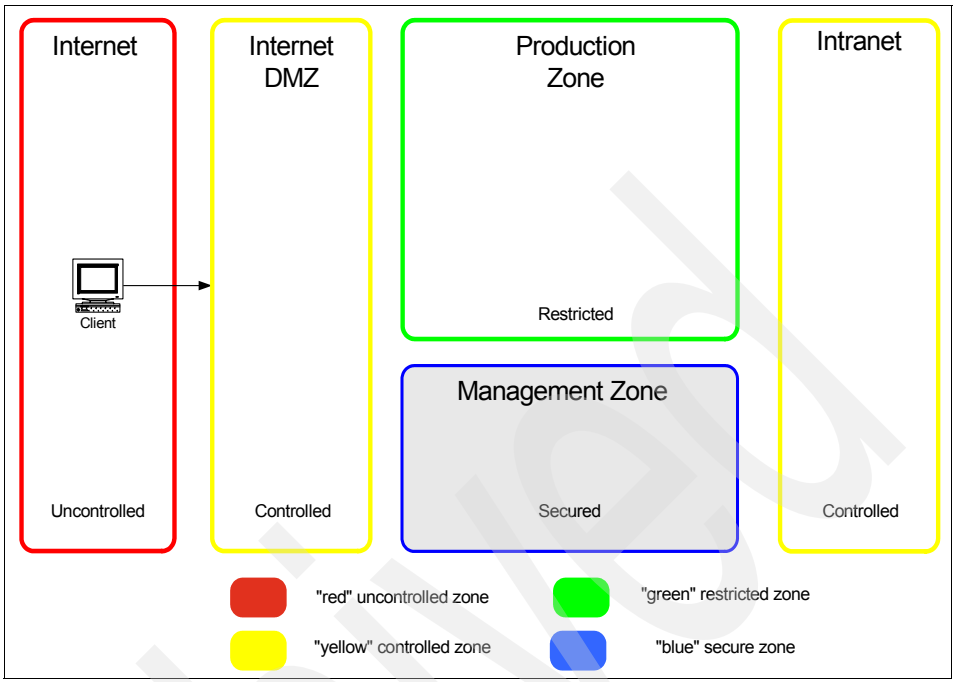

Figure 4-1 Network zones

Note: The breaks between each network zone indicate the use of a firewall that clearly delineates each perimeter from the next.

Placement of the components in these zones reflects both the access control security policy requirements and the levels of trust among the components, in particular the quality of protection provided by transport layer security. Figure 4-2 on page 48 shows the general component relationships and the transport classifications as a function of the security mechanisms used to achieve increased trust.

As you move from an uncontrolled zone to a secured zone, the *quality of protection* must increase as more sensitive resources are open to access. From a transport perspective, there are several mechanisms that can be employed. Typically, the Web environment secures the transport using Secure Socket Layer (SSL) tunnels and HTTPS protocol.

From a component-to-component interoperation perspective, the individual components use interprocess communications such as rpc and icmp to provide information between components, such as status and exchange of server-relevant data. There are several options for securing these paths. However, the typical choice in today's distributed environment is mutually authenticated SSL.

In addition to the use of cryptographically sealed communications such as SSL, a further measure of trust can be achieved by managing the degree of exposure by augmenting firewall policies with a proxy component. By setting up the DMZ to house only a component that acts like a buffer between the uncontrolled zone and the restricted and secured zones, then both the firewall port policies and the buffer component provide an effective chokepoint or controlled gateway.

A best practice approach for the placement of Access Manager components is also depicted in Figure 4-2.

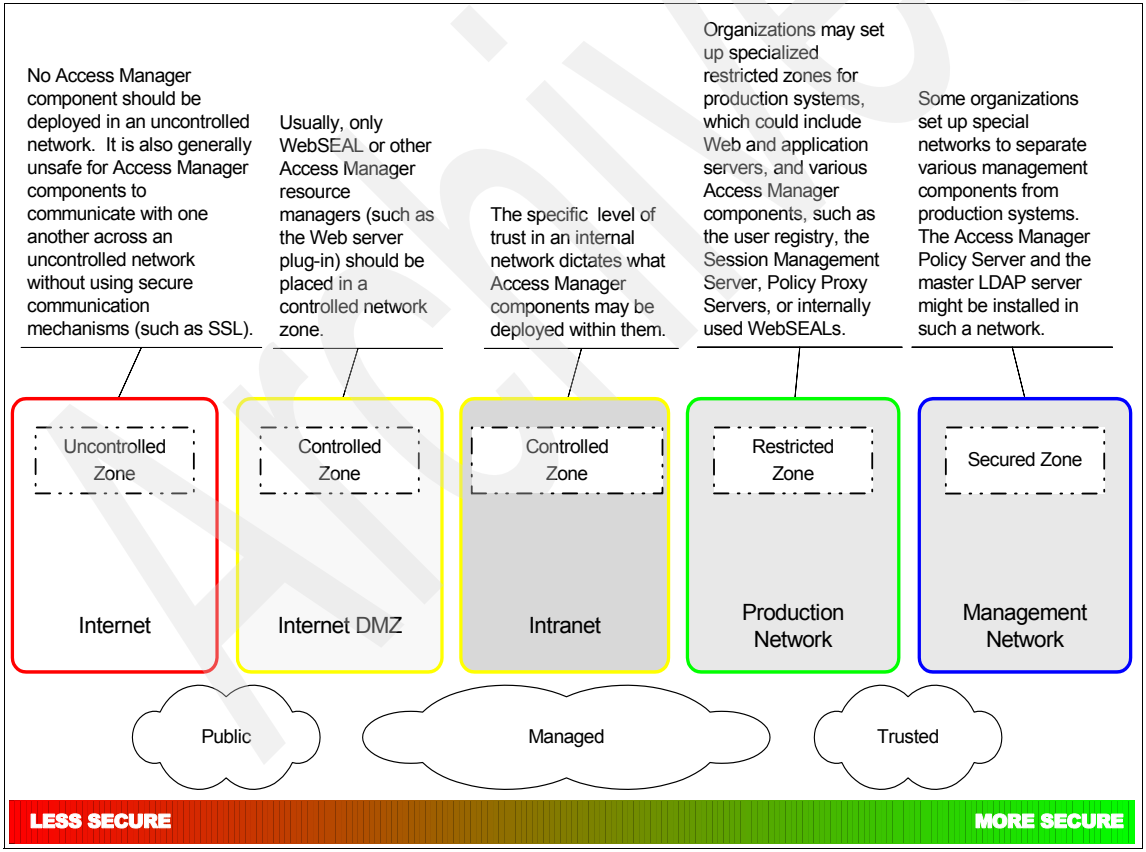

Figure 4-2 Network zones, transport classifications, and their level of trust

Let us discuss how to place the TAMCO resources and the necessary Access Manager components into the different network zones.

4.2.2 Client access

All Web-based client requests come into the TAMCO portal using HTTP or HTTPS protocols. The only distinction for us is whether the client request originates from the Internet or the intranet.

Internet client access

A firewall controlling traffic between the uncontrolled Internet and the controlled DMZ only allows ports 80 (HTTP) and 443 (HTTPS). The Internet client is typically a browser making HTTP/HTTPS requests to controlled servers in the DMZ. All browser-based access from the Internet is going to be channeled through the Access Manager WebSEAL component. All other Web application servers will be unaccessible. Based on the TAMCO access control security policy, WebSEAL will enforce proper authentication and authorization depending on the requested resource and the group membership of the authenticated user.

Another firewall located between the controlled DMZ and the restricted production zone only allows WebSEAL to communicate with back-end Web application servers using ports other than 80 or 443.

Intranet client access

A firewall controlling traffic between the controlled intranet and the restricted production zone (which houses all Web-based applications) only allows the ports 80 (HTTP) and 443 (HTTPS). The intranet client is also a browser making HTTP/HTTPS requests. However, these clients are not routed through the Internet DMZ but are going to be channeled through an Access Manager WebSEAL component that sits in the restricted zone (production network) next to the Web resources. Based on the TAMCO access control security policy, WebSEAL will enforce proper authentication and authorization depending on the requested resource and the group membership of the authenticated user.

Due to a risk assessment for intranet-based resource access, TAMCO has decided that there is no need for a separate internal DMZ. The WebSEAL server in the restricted zone will be the only HTTP/HTTPS component accepting incoming browser-based Web requests. All Web application servers are going to be configured to only allow incoming HTTP/HTTPS traffic from the deployed Access Manager WebSEAL components.

4.2.3 Controlling Web resource access

In addition to a firewall that provides filtering policies for network traffic on specific restricted ports, it is necessary to provide a way by which requests can be authenticated and which allows traffic to be routed and managed based on requests for resources inside the TAMCO network. Typically, this job is performed by an HTTP server configured to be a *reverse proxy*.

WebSEAL is a high-performance, multi-threaded HTTP server with built-in authorization services able to apply a fine-grained security policy to resources in the Tivoli Access Manager protected secure domain. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy. WebSEAL typically sits in the controlled Internet DMZ.

Similar to a traditional reverse proxy, WebSEAL protects the Web server by intercepting all requests to the back-end Web server and ensures that the data contained in the request is acceptable. Pre-configured ports on WebSEAL (typically 80 and 443) accept HTTP/HTTPS requests, evaluate them, and then retrieve the requested content from its own server or the back-end Web server to forward it to the original user. In this way, users never directly access the Web server. All other inbound ports are closed, thereby enabling WebSEAL to act as a *chokepoint*. In addition, WebSEAL also provides Web acceleration techniques such as caching and compression and object filtering that is designed to reduce Web site access times.

One of WebSEAL's key functions is to protect access to Web content and applications on the back-end Web application servers. WebSEAL works with the user registry to carry out authentication, credential creation, and authorization. Requests passing through WebSEAL are evaluated by the Tivoli Access Manager authorization service (typically part of WebSEAL deployment) to determine whether the user is authorized to access the requested resource.

Using information about the user from the authentication (user ID, certificate DN, and so on), WebSEAL builds a credential for the user. This credential is in the form of a Privilege Attribute Certificate (PAC) that can be extended to hold additional attributes.

WebSEAL connects to the back-end Web application servers using a *junction,* which is an HTTP mount point on the file system of the Web application server. WebSEAL passes authentication and authorization information across a junction. Content flowing over the junctions can be secured using SSL.

Traditional WebSEAL junctions

When using traditional junctions, depicted in Figure 4-3, WebSEAL is constantly listening for requests and filtering traffic, and appending the name of the junction to the end of the URL to facilitate routing of the request to the correct back-end server (defined when the junction was created). For simple environments, appending the junction name to the request URL is acceptable and useful.

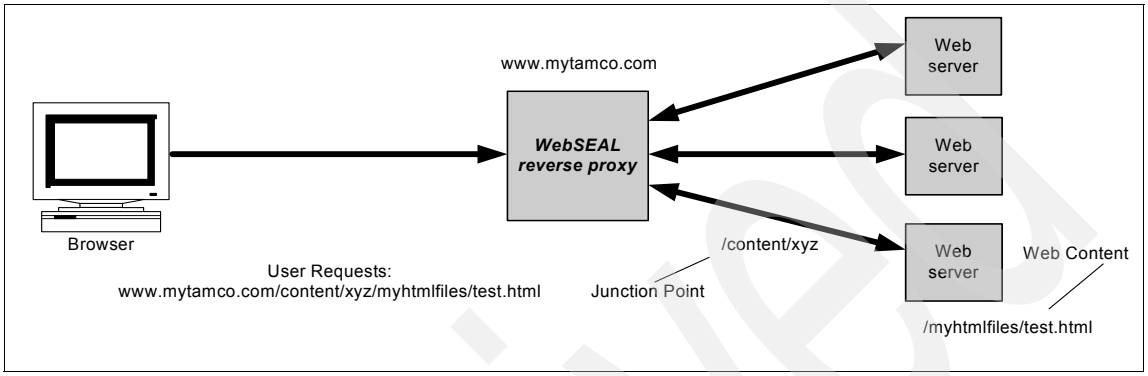

Figure 4-3 WebSEAL traditional request path and junction

However, for Web environments with more complexity, such as portals using virtual hosting with a requirement to preserve well-known organization-wide addresses (for example, bookmarks), or a single Web objectspace with single sign-on configured, path filtering can impose undesired changes. Therefore, WebSEAL junctions in Access Manager also provide support for *virtual hosting* and *transparent junctions* (no junction name required on the URL).

Virtual host junctions

Virtual host junctions preserve the traditional Web addresses that may already exist within an organization. For example, a company may have `www.myhr.com` for their HR system and `www.mypayroll.com` for their payroll system. Since these applications already exist and their Web addresses are known throughout the user community, the application of the traditional WebSEAL junction method would not benefit the corporation. Instead, resolving `www.myhr.com` and `www.mypayroll.com` to WebSEAL's IP address and allowing it to decipher which server to direct traffic to would be the most beneficial, as shown in Figure 4-4 on page 52. This capability is new to WebSEAL 6.0.

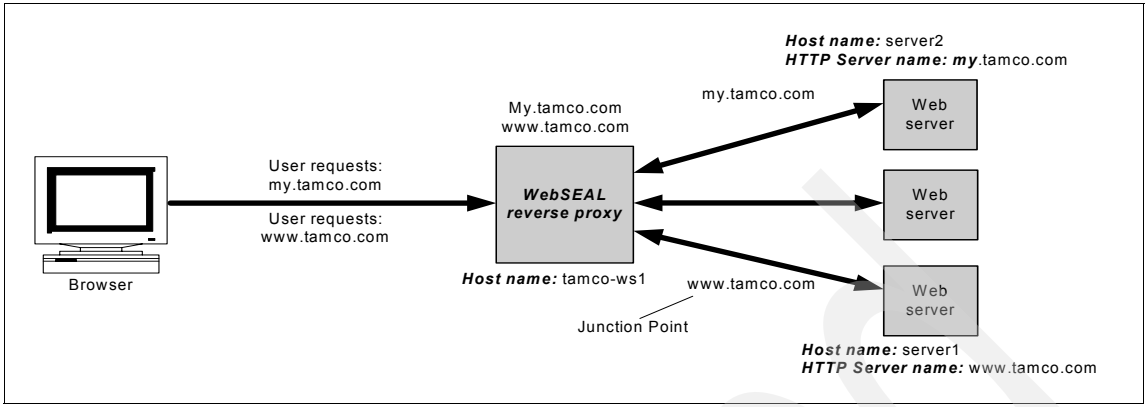

Figure 4-4 WebSEAL virtual host junctions

TAMCO is using virtual hosting with the MyTAMCO portal. In the initial deployment phase of the portal a year ago, TAMCO defined and published a number of Web addresses for requesting services, including providing links on their custom portal pages to those applications. In order not to introduce new names and addresses, they have decided to deploy the virtual host junctions.

In order to understand junctions, we now discuss the transparent path junctions, which is another new feature of junctioning in WebSEAL 6.0. TAMCO has determined that they do not need to use transparent path junctioning.

Transparent path junctions

In order to combine the benefits of both a single URL space for session management and single sign-on without the problems of path filtering, Access Manager for e-business is using the concept of *transparent junctions*. Transparent junctions remove the need for the junction name, such as /content/xyz, to be included in the Web address. Instead, transparent junctions are part of the existing URI space located on the back-end server.

In the example of `my.tamco.com`, the transparent junction would simply be /myhtmlfiles, as shown in Figure 4-5 on page 53. There is no need to add an extra junction name.

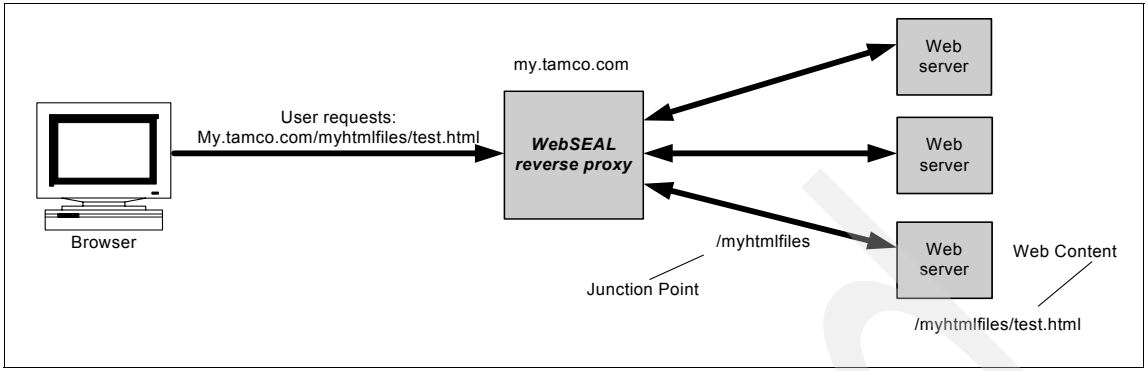

Figure 4-5 WebSEAL transparent junctions

Authentication and authorization

Being the entry point for all Web-based resource accesses, WebSEAL performs authentication and authorization functions by interacting with the user registry and Access Manager Policy Server or Policy Proxy Server, as shown in Figure 4-6 on page 54. The communication between WebSEAL and these servers is protected by using mutually authenticated SSL connections.

WebSEAL provides a wealth of different authentication mechanisms that are explained in Chapter 8, “Increasing availability and scalability”, in *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

TAMCO has decided to implement a forms authentication mechanism because all logins are taking place through their portal.

In addition to this method, access to the financial applications will only be granted if the authenticated user can provide an additional digital certificate through a *step-up authentication* process in WebSEAL. These certificates will only be distributed to employees in the financial department and executives.

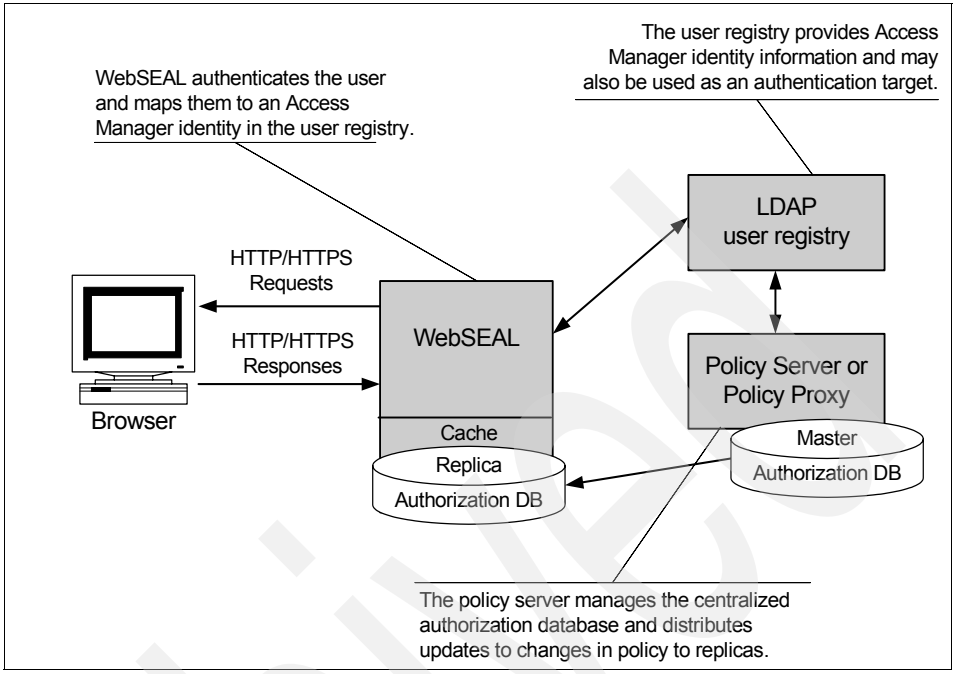

Figure 4-6 WebSEAL interaction with other Access Manager servers

WebSEAL is part of the tightly coupled trusted communication path with the back-end WebSphere Application Server and WebSphere Portal utilizing either the WebSphere Lightweight Third Party Authentication (LTPA) mechanism or the Trust Association Interceptor (TAI). Both mechanisms can be used to set up single sign-on capability.

4.2.4 Back-end resources

Several of the key TAMCO application servers sit behind the second firewall in a restricted zone that is called the production network.

This zone contains the Web application servers that host and serve the TAMCO applications. These servers reside in the restricted zone because it is important to secure the transport from and to these servers, both in terms of the request for service and the interprocess communications, in order to prevent uncontrolled access that can exploit vulnerabilities. WebSEAL in the DMZ (for Internet-based access) and the production network (for intranet-based access) provides both transport protection and a buffer to arbitrate access requests to the services provided by these application servers.

The Web application servers that will be deployed in the production zone are:

- WebSphere Portal V5.1
- WebSphere Application Server V6.0
- PeopleSoft Application Server V8.9 with IBM DB2 V8.1
- PeopleTools V8.46
- Siebel Customer Relationship Manager Server V7.7
- Lotus Domino Server V6.0

All these Web-based application resources will be configured to only accept incoming HTTP/HTTPS traffic originating from Access Manager WebSEAL servers on predefined ports and from predefined fixed IP addresses. These servers also accept user credentials passed on from the WebSEAL servers.

4.2.5 Management resources

The management zone contains IBM Tivoli Access Manager Policy Server or Policy Proxy Server and the master LDAP user registry deployed on IBM Tivoli Directory Server V6.0 and IBM DB2 UDB V8.2. It separates these components from any other network zone in order to provide an additional degree of trust through isolation.

In addition to protecting the transport communications with SSL, these servers are further protected by belonging to Access Manager groups with tightly restricted ACLs to restrict access to only a small number of trusted administrators and other Access Manager servers.

The Access Manager servers to be deployed in the management zone are the servers responsible for storing authentication data and attributes used in authentication and authorization. From here the necessary information, the Access Manager authorization database, and the LDAP directory are being replicated to the actual decision-making replicas.

TAMCO determined that the Policy Server should be placed in Finland, with Policy Proxy Servers located in Germany and UK. The Policy Server makes updates directly to the master LDAP server, which is also located in Finland. TAMCO decided to leverage the Policy Proxy Server in each of the secondary geographies to minimize Policy Server load and for increased security. As the name suggests, Policy Proxy Server is a proxy server used to isolate and protect the Policy Server from direct access. It acts as a client to the Policy Server.

The firewall protecting the primary Policy Server only has to allow inbound connections from the Policy Proxy Server rather than from all Tivoli Access Manager servers. Incoming connections to the Policy Proxy Server are authenticated at the Policy Proxy Server before being passed to the primary Policy Server. Administration requests from Tivoli Access Manager applications are also authenticated individually before being passed to the real Policy Server. In the TAMCO architecture, Tivoli Access Manager applications refers to the Web Portal Manager and the WebSEAL servers.

There are several advantages to allowing the Policy Proxy Server to intercept requests remotely. Only incoming Secure Sockets Layer (SSL) sessions to the Policy Server come from the physical Policy Proxy Server. This provides increased security. The ACL database is cached in memory for security. There is no authorization database stored on the disk of the Policy Proxy Server that can be read (or modified) if the Policy Proxy Server is compromised. In addition, the Policy Proxy Server offloads database replication tasks from the Policy Server by caching the Policy Server databases that it serves to Access Manager applications.

4.2.6 Availability and scalability

Since TAMCO will deploy most of its operational applications and sales and marketing information within the new Access Manager secure domain environment, we must discuss high availability and scalability.

Availability is the major concern that a failing part of the infrastructure will cause the overall solution to languish. This eventually leads to unsatisfied customers and decreasing business success.

Scalability describes the ability to instantaneously change and adapt the IT infrastructure in order to handle an increased number of information and transaction requests without reducing the quality of the online experience for customers.

TAMCO is looking for a way to guarantee the availability of the business applications around the clock. They have to provide a reliable e-business application infrastructure that is always responsive. A second concern is the constantly increasing number of customers visiting the Web site. TAMCO looks for future flexibility and ways to dynamically add functional empowerment of the single systems to better cope with new e-business opportunities.

Web servers and applications can and do fail. The reasons for failure vary: program code, unproven technologies, disk failures, and even human error. In Figure 4-7, WebSEAL, the LDAP user registry, Access Manager Policy Server with its authorization master database, Web Portal Manager, and each individual Web server can each represent a single point of failure.

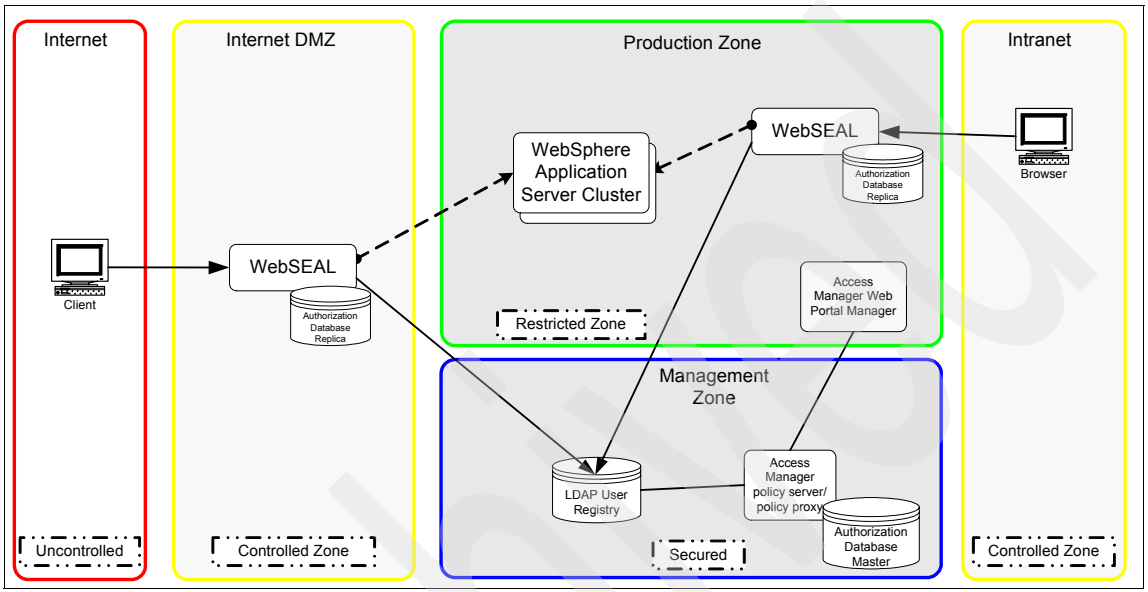

Figure 4-7 Basic component deployment

At this time, we assume that TAMCO has already deployed their Web applications in a WebSphere cluster environment, so we do not address Web application server availability and scalability.

What happens if the WebSEAL server fails? What happens if the user registry or the Access Manager Policy Server stops working? We now take a closer look at these individual components.

WebSEAL failure

If the WebSEAL portal to either the Internet or the intranet fails, and there is no operational replacement, the client attempting access will be denied access to the site. While the content and the application might be fully functional behind WebSEAL, the failure of the WebSEAL server leads the user to believe that the site is down.

User registry failure

If the user registry is down, WebSEAL will no longer be able to authenticate incoming users in order to access Web content and applications that are protected and require user authentication. WebSEAL and the Web servers may still be operational, but the client is unable to gain access and thus assumes that the site is down.

Access Manager Policy Server failure

Although failure of your policy server is not on your wish list, at least it does not affect the availability of your Web site. WebSEAL can still perform all necessary authorization operations because it uses the local cache mode, which means that the authorization service running on the WebSEAL machine uses a local authorization database replica. You only lose the ability to administer your Access Manager secure domain while your policy server is down.

The same guidelines apply if WebSEAL is configured to update the authorization database through the Policy Proxy Server instead of the Policy Server. WebSEAL leverages the local authorization database cache; however, authorization updates cannot be made until the master Policy Server is available.

Web Portal Manager server failure

The Web Portal Manager (WPM) provides the graphical user interface Web application for the Access Manager administrators. The Web application will not be affected if WPM is not available. The only impact is that the GUI administration of the Access Manager secure domain has to be postponed until the service is available again. However, in the absence of WPM, the pdadmin command-line tool is still available and in some cases is preferred for administration of the secure domain.

In addition to problems or failures of these components, sheer volume can affect availability as well. With the growth of the Internet and your business, the ability to handle the traffic to your site has changed the scope and appearance of the architecture. Internet sites can become unstable or even fail under severe load conditions.

4.2.7 Providing high availability

Adding replicas of crucial servers will increase TAMCO's availability. After depicting an overview of this configuration in Figure 4-8, we describe the different areas with their solutions.

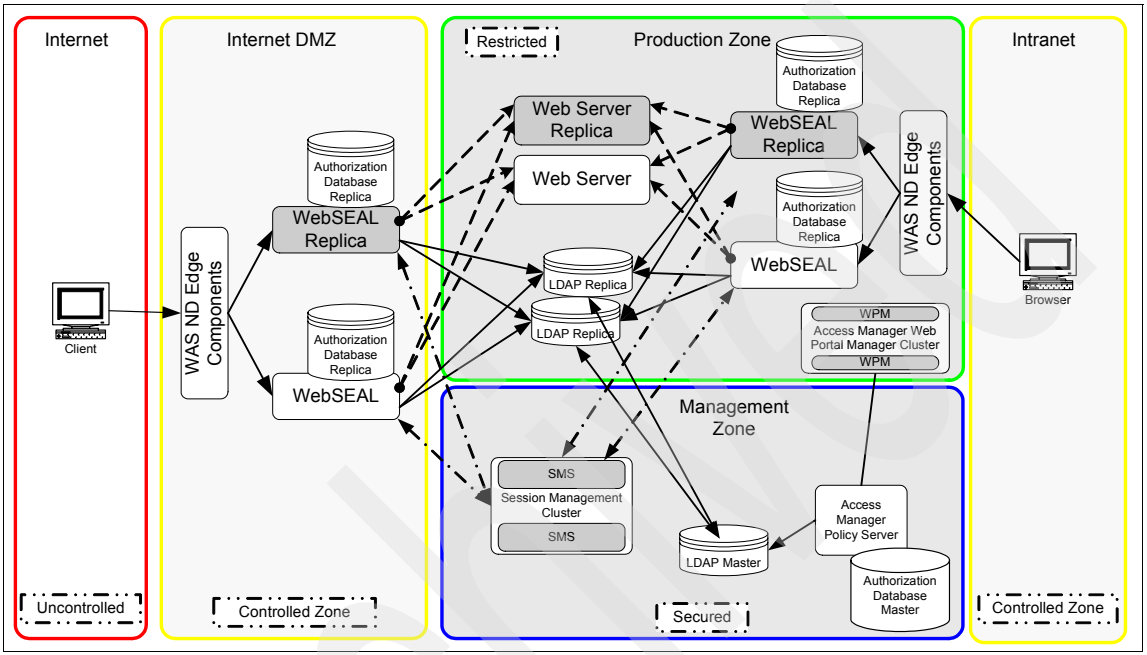

Figure 4-8 High availability deployment

WebSEAL availability

TAMCO has decided to increase the availability of the WebSEAL-controlled Web site with at least two front-end WebSEAL servers. Replicated front-end WebSEAL servers provide the site with load balancing during periods of heavy demand as well as failover capability. That is, if a server fails for some reason, the remaining replica servers continue to provide access to the site. Successful load balancing and failover capability results in high availability for users of the site. The load-balancing mechanism is handled by a component such as the IBM WebSphere Application Server Edge Components[2] or other third-party hardware or software based solutions.

[2] In today's WebSphere environment, the load balancing functionality is being delivered by the WebSphere Application Server Network Deployment V6.1 Edge Components. More information about these components can be obtained at http://www.ibm.com/software/webservers/appserv/was/network/edge.html.

The objective of a front-end load balancer is to use clustering technology to provide highly available hosting services that are robust and scalable, and to improve performance and server utilization by balancing requests across the available applications servers in the cluster.

Note: The load balancer should always be configured with *session affinity* enabled on the WebSEAL cluster. Once authenticated, session affinity ensures persistence to an individual WebSEAL server throughout the session life. Without this persistence, the browser could be redirected to another WebSEAL instance that does not contain the current session information. Using single sign-on methods, such as *failover cookie* or the *Session Management Server* (SMS), WebSEAL is able to regenerate the user's session; however, there is some impact associated with these methods and thus it is best to send all subsequent requests back to the same WebSEAL server unless there is a failover scenario.

More information about this configuration can be found in the section "Replicated front-end WebSEAL Servers" in *IBM Tivoli Access Manager for e-business WebSEAL Administration Guide Version 6.0,* SC32-1687.

Session availability

Deploying multiple WebSEAL servers (or Web Security Plug-Ins) introduces another field of concern: *session availability*.

When WebSEAL or the Web Security Plug-In replicas are needed, careful consideration needs to be made not only about the method for managing single sign-on, but also the method(s) used to manage sessions. The three configurations that we focus on in this guide are those that are provided by the Access Manager for e-business suite and do not require addition components.

TAMCO evaluated *failover cookie*, the *Session Management Server*, and *cross domain single sign-on*. However, custom solutions leveraging none or a combination of these methods may also be appropriate depending on the architecture design.

It should also be noted that many customers are embracing the open standards approach to enterprise single sign-on by adopting a ratified standard for passing security tokens between partners leveraging IBM Tivoli Federated Identity Manager. Details on Federated Identity Manager and Access Manager for e-business integration are covered in *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014 and *Federated Identity Management and Web Services Security with IBM Tivoli Security Solutions*, SG24-6394.

TAMCO has decided to leverage a cookie-based approach using the security server failover cookie as a means for single sign-on. TAMCO will look to deploy SMS when the e-business ordering system and customer shopping experience is made available for customers at a later time.

Failover cookie

The *failover cookie* is a server-specific cookie or a domain cookie. If the cookie is set at the domain level, the failover cookie can also provide single sign-on across multiple domains. For this to function, failover cookies need to be configured as a *domain cookie* and both hosted servers must share the same cryptographic key. This allows Web security servers hosted in the same DNS-domain to accept failover cookies created by other servers in the same domain with the same shared key.

The failover cookie is a mechanism for transparently re-authenticating the user in a different domain; it is not actually a mechanism for maintaining sessions. Failover cookies contain encrypted user authentication data that a WebSEAL server can use to validate a user's identity. The cdsso_key_gen utility is used to generate a key pair that can secure the cookie data. Failover cookie configuration requires the distribution of this secret key to all participating WebSEAL servers in the DNS-domain.

After a WebSEAL server receives this cookie, it decrypts the cookie (with the common key) and uses the user name and authentication method to regenerate the client's credential. The browser passes the failover cookie with each request and the WebSEAL uses the cookie to establish a user session. The user actually has a unique session on each WebSEAL server.

A failover cookie maintains the following information:

- User credential information
- Session inactivity timeout value
- Session lifetime timeout value
- Extended Attributes (if needed)

You need to carefully consider security when configuring the *failover cookie*, because since it behaves differently than a *session cookie*. If an attacker hijacks a session cookie, the session cookie is only valid until the WebSEAL server deletes the associated session. Failover cookies are valid until the lifetime or inactivity timeout in the failover cookie is reached. Therefore, it is good practice to to make sure the failover cookie lifetime does not exceed the session cookie lifetime. In addition, if the session is removed through pkmslogout, then so is the failover cookie. Failover cookies do allow the enforcement of session lifetime timeouts, inactivity timeouts, and pkmslogout.

The failover cookie is very cost effective (there is no additional hardware required) and simple to manage, which makes the failover cookie attractive for many environments. Failover cookies do not have to be implemented in an environment with a Session Management Server because the failover cookie takes responsibility for maintaining a user session.

Session Management Server (SMS)

Tivoli Access Manager for e-Business can be configured to take advantage a Session Management Server (SMS), which is a separate component running on WebSphere Application Server. SMS prevents a user's session from being destroyed when a WebSEAL server goes offline. This also keeps user login policy data consistent across multiple WebSEAL servers. Another advantage of this approach is that, unlike failover cookies, Session Management Server cookies are not cryptographic cookies containing user information. With the use of the Session Management Server, it is now possible to maintain a login history of the Access Manager environment.

The Session Management Server overcomes obstacles in relation to session management in a clustered WebSEAL environment that include limitations for policy enforcement, management, security, and the user experience. It also provides single sign-on between WebSEAL servers in a failover situation.

The Session Management Server provides the following benefits in a clustered environment:

- Distributed session cache to manage sessions across clustered WebSEAL server environments.
- Central point for maintaining login history information.
- Inconsistencies resolved between replicated WebSEAL servers in regards to session inactivity and session lifetime timeouts.
- Single sign-on and secure failover among replicated WebSEAL servers.
- Maximum number of concurrent sessions enforced across replicated WebSEAL servers.
- Single sign-on capabilities among other Web sites in the same DNS domain.
- Performance and high availability protection to the server environment in the event of a hardware or software failure.
- Administrators can view and modify sessions on WebSEAL servers.

As the Session Management Server runs within WebSphere Application Server to provide high availability, you should use WebSphere clustering, as shown in Figure 4-8 on page 59.

Cross domain single sign-on

Cross domain single sign-on (CDSSO) provides a mechanism for transferring user credentials between unique servers and domains. CDSSO allows movement of users between the domains with a single sign-on. When a user makes a request to a resource located in another domain, the CDSSO mechanism transfers an encrypted user identity token from the first domain to the second domain. The identity information in this token indicates to the receiving domain that the user is successfully authenticated in the first domain. The identity does not contain password information. The receiving server uses this token to build credentials in its own cache for that user. The user is not forced to perform an additional login.

With CDSSO, the user makes a request to a special link on a WebSEAL server, which then initiates the process to forward the request, along with credential information, to a WebSEAL server in a different Access Manager domain. If the user were to instead directly access the link in the target domain, he would have to authenticate to that domain. In order for CDSSO to work, the user must exist in both domains and have appropriate access controls in both.

A high level diagram of the CDSSO flow mechanism can be seen in Figure 4-9.

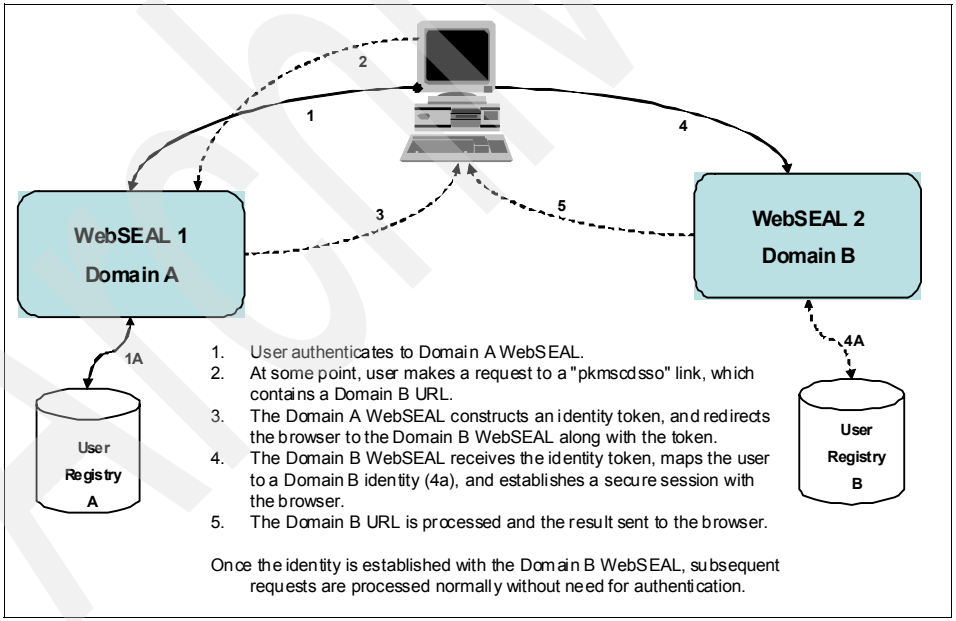

Figure 4-9 Basic CDSSO diagram

At TAMCO, the user IDs in both domains are the same, so there is no special programming interfaces that would be required to map the users. However, TAMCO has chosen to leverage virtual host junctions. At the time of this writing, CDSSO is not supported with virtual host junctions. If single sign-on is needed between separate DNS domains or Access Manager domains, and either virtual hosts or virtual host junctions are used, *e-community single sign-on* is the only technology supported for this type of functionality.

User registry availability

IBM Tivoli Directory Server supports the concept of master and replica LDAP servers. A master server contains the master directory from which updates are propagated to replicas. All changes are made and occur on the master server, and the master is responsible for propagating these changes to the replicas.

A replica is an additional server that contains a database replica. The replicas must be exact copies of the master. The only updates that replicas allow are through replication from the master. The replica provides a backup to the master server. If the master server crashes or is unreadable, the replica is still able to fulfill search requests and provide access to the data. You can deploy more than one replica in order to increase availability and scalability.

TAMCO has opted not to run in a multi-master LDAP mode, but rather accept the fact that the master LDAP directory might be unavailable for short periods of time. User updates, additions, and deletions will be affected in the absence of the master LDAP server; however, WebSEAL will still be able to authenticate users to one of the replicas.

If the master directory machine fails altogether, one of the replicas can then be promoted to become the master and another replica server can be deployed as required. They will deploy two LDAP replicas in the restricted production network.

Access Manager Policy Server

To provide redundancy for the Policy Server, a local standby Policy Server in Finland will be configured using high-availability (HA) clustering software with shared storage between the primary and standby servers. Guidelines on how to leverage HA clustering software for Policy Server redundancy are discussed in Chapter 27, “Using response files”, of the *Tivoli Access Manager for e-business Version 6.0 Installation Guide,* SC32-1361.

Web Portal Manager availability

The Web Portal Manager provides the administration GUI Web application for the Access Manager administrators. If the implementation requires a 24x7 availability of the Web administration interface, WebSphere clustering should be used to satisfy this requirement. Since Access Manager Web Portal Manager runs on the WebSphere Application Server, clustering of the application is supported by using WebSphere Application Server V6.0.2 Network Deployment.

TAMCO has decided that at this time they do not need the Web Portal Manager to be highly available.

This concludes the discussion on high availability. Next, we prepare for our deployment and take a closer look at the final architecture and the prerequisites for installation.

4.2.8 Common Auditing and Reporting Service

Tivoli Access Manager V6.0 offers a component for auditing and reporting called IBM Tivoli Common Auditing and Reporting Service (CARS). In the Common Auditing and Reporting Service context, *auditing* is defined as the process of maintaining detailed, secure logs of critical activities in a business environment. In essence, the underlying importance of common auditing and reporting is in proving compliance. Hence, the Common Auditing and Reporting Service reports are used for:

- External controls show compliance for various standards and legal requirements
- Internal controls show compliance to an organization's security policies
- Checking enforcement and effectiveness of IT controls, for accountability and vulnerability/risk analysis
- Forensic investigations of security incidents

The Tivoli Common Auditing and Reporting Service architecture is shown in Figure 4-10 on page 66 and consists of:

- The Common Auditing and Reporting Service server, which includes the event server feature and the operational reports feature
- The Common Auditing and Reporting Service client, which includes the Java and C client

For a detailed discussion of all CARS capabilities and reports, see Chapter 23, "Introducing IBM Tivoli Common Auditing and Reporting Service," in *Enterprise Security Architecture Using IBM Tivoli Security Solutions,* SG24-6014.

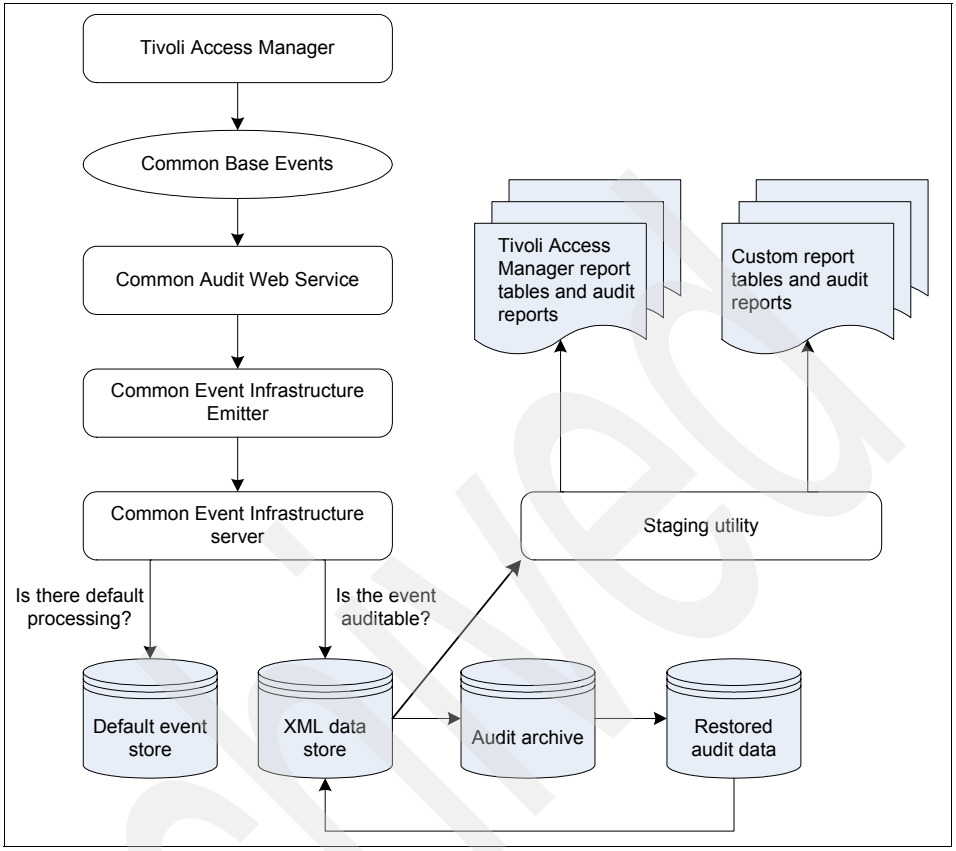

Figure 4-10 Common Auditing and Reporting Service architecture

Auditing

Auditing is the process of maintaining detailed, secure logs of critical activities in a business environment. Such critical activities could be related to security, content management, business transactions, and so on.

Based on how the audit data is used, management of the audit data has the following requirements:

- Collect and store large volumes of data for a long period of time.
- Stage the data periodically (daily or weekly) into report tables for audit reports.
- Archive the audit data for a long period of time (months or years) with archiving scheduled on a regular basis.

- Produce audit reports on recent and archived audit data. Such reports can be produced by customers using their reporting tool of choice or shipped as part of IBM products.
- Provide a process that is tamper resistant. That is, the audit data must be kept safe when it is generated, during transit, and when it is stored.
- Provide auditing functionality for changes to the configuration and policy for collecting audit data.

Reporting

The operational reports feature of the Common Auditing and Reporting Service provides a number of compiled reports that provide information about security-related activities that occur on your system.

The compiled Crystal Reports provided with the Common Auditing and Reporting Service include audit event history, password change activity, authentication event history, authorization event history, event details, resource access, and server availability reports. The compiled reports format allows you to run reports without having the Crystal Reports Designer installed on the system.

Since the Common Auditing and Reporting Service ships with Tivoli Access Manager V6.0, TAMCO has decided to deploy the centralized auditing and reporting solution.

4.3 Deploying physical components

Figure 4-11 shows the final set of functional components that satisfy the TAMCO requirements and Figure 4-12 on page 69 shows the IT architecture across the geographies.

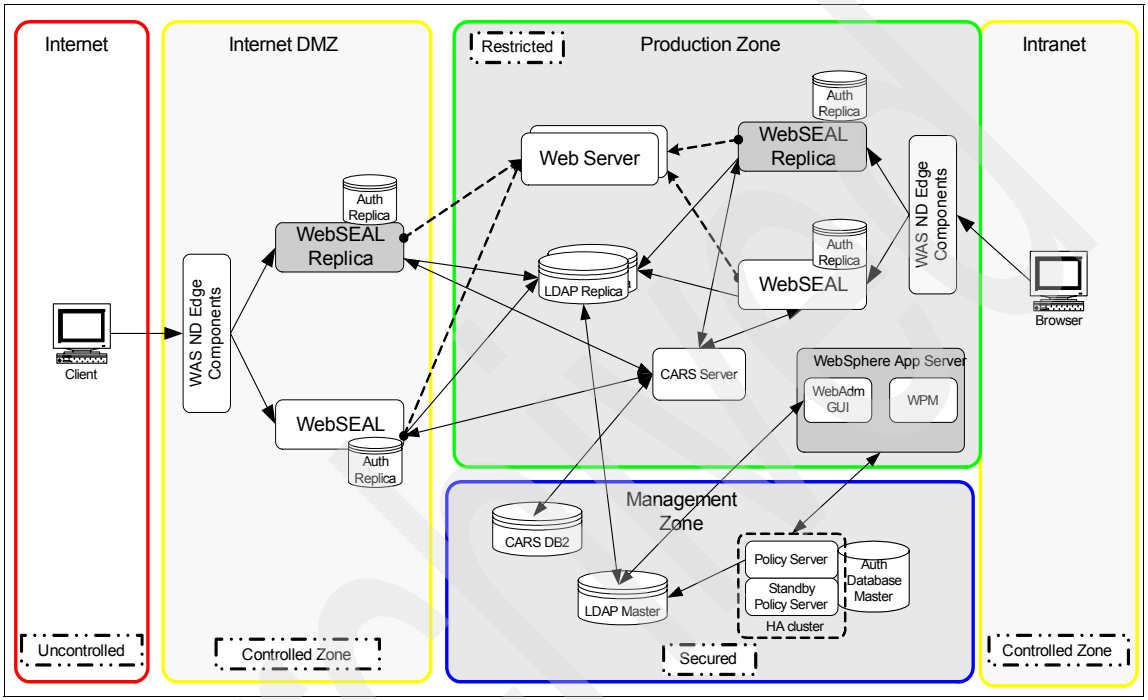

Figure 4-11 Final deployment architecture

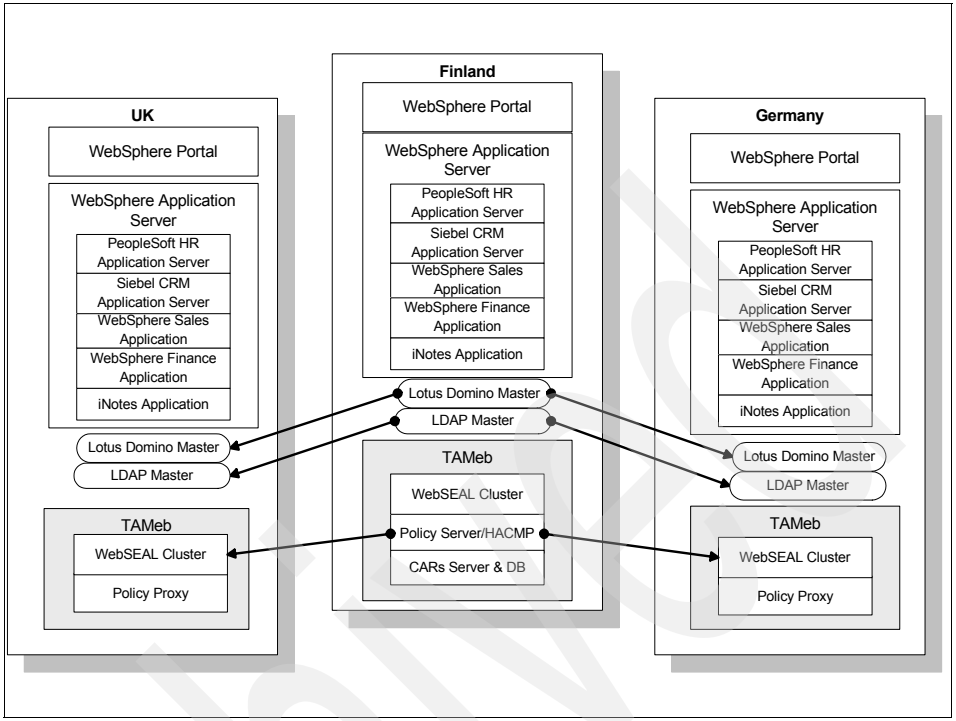

Figure 4-12 TAMCO IT architecture with Access Manager for e-business

Let us take a look at some of the prerequisites for the components we are going to deploy.

4.3.1 Prerequisite components

Figure 4-13 shows an overall picture of the relationship between the prerequisite software and the base Access Manager servers.

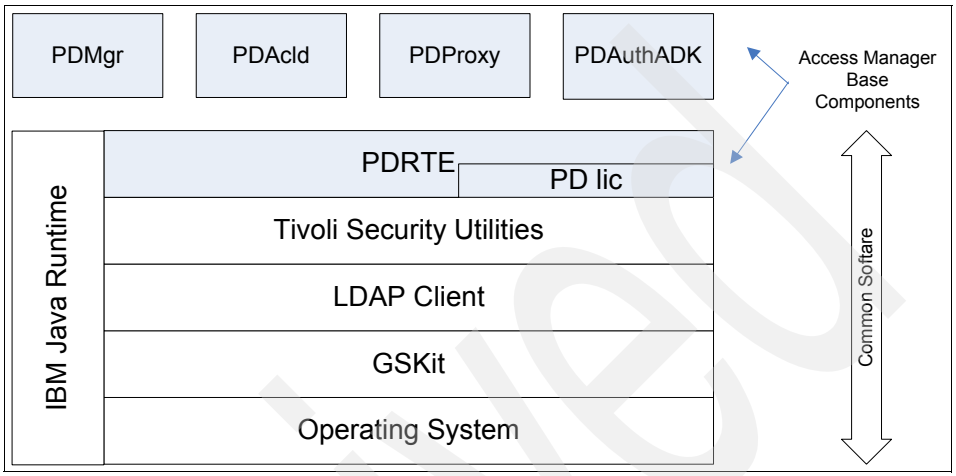

Figure 4-13 Access Manager prerequisite software and base components

Prior to the installation and configuration of the core Access Manager servers and clients, it is necessary to install and configure the following prerequisite products:

- IBM Java Runtime V1.4.2 SR2

  IBM Java Runtime provided with Tivoli Access Manager is required when installing and using language support packages and when using Tivoli Access Manager installation wizards. The Access Manager Runtime for Java component only supports IBM Java Runtime.

- IBM Global Security Kit (GSKit) V7.0.3.17

  The GSKit provides the cryptographic functions and libraries used for SSL and digital certificate management. GSKit will be installed on all hosts and clients using SSL:

  - WebSEAL master and replicas
  - WebSphere Application Server
  - Tivoli Directory Server

- IBM DB2 Universal Database™ Enterprise Server V8.2 (or V8.1 FP8)

  DB2 provides a full relational database as a back-end data store for IBM Tivoli Directory Server V6.0. An instance of DB2 will be stored with the master directory server and with each replica directory server.

- IBM Tivoli Directory Server V6.0.0.2 (equivalent to 6.0.0.1-TIV-ITDS-IF0001)

  The LDAP Directory is used as the Access Manager user registry. We will deploy one LDAP master directory server in the management network and two replicas in the production network.

- WebSphere Application Server V6.0 Refresh Pack 2

  An instance of WebSphere will be installed in each production network.

- IBM HTTP Server V6.0

  The Web server is required for improved robustness of communication between clients and WebSphere Application Server, and hence will be deployed on each machine hosting WebSphere Application Server.

- Load balancer

  TAMCO requires the use of a load balancer for high availability and failover to arbitrate traffic load from clients to the DMZ and from internal clients to the intranet. TAMCO has deployed a hardware appliance such as Cisco Content Server Switch or F5 Networks Big Iron/BigIP switch as their network load-balancing solution.

4.3.2 Components of an Access Manager system

The TAMCO solution needs the following Access Manager servers to be installed:

- Access Manager Policy Server (PDMgr)

  The Access Manager Policy Server needs to be configured to communicate with the LDAP master directory server.

- Access Manager Authorization Server (PDAcld)

  An Access Manager Authorization Server will be installed in the Finland domain only. Replicas of the authorization policy database will be maintained locally with each instance of the policy server.

- Web Portal Manager (WPM)

  Web Portal Manager needs to be configured to manage the policy server's protected object namespace. The components of WPM are shown in Figure 4-14.

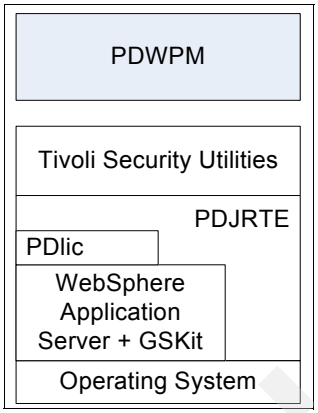

Figure 4-14 Components necessary for WPM installation

- WebSEAL servers

  Two replicated WebSEAL servers have to be deployed in the DMZ (facing Internet client requests) and the production network (facing intranet client requests). They have to accept incoming traffic distributed by the load balancers.

  Figure 4-15 on page 73 depicts all the components of the Access Manager Web security system (WebSEAL). The TAMCO solution does not use all of these components, and it is not necessary to install them all. We will be installing the PDWebRTE and PDWeb, which provide the base WebSEAL functionality.

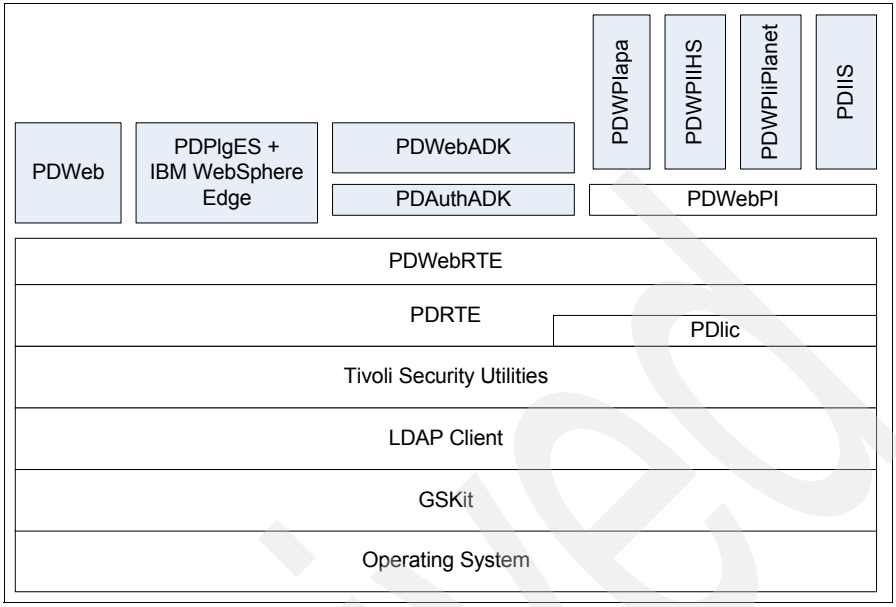

Figure 4-15 All Access Manager Web security components

4.4 Conclusion

In the beginning of this chapter, we defined the TAMCO access control security policy. Next, we examined the necessary disciplines for the solution deployment, covering network structure, client access, and the different resources, as well as high availability and a centralized auditing and reporting system. Finally, we discussed physical component prerequisites. In the next chapter, we take you through the installation details of all the components.

5

Installing the components

We developed the security requirements from the business drivers described in 3.1, “Business drivers and capabilities” on page 31. We determined the specific security management and administration capabilities in 4.1, “Defining the access control security policy” on page 38. Lastly, we described the Access Manager components that need to be deployed into the TAMCO solution in 4.2, “The TAMCO deployment” on page 45, and 4.3, “Deploying physical components” on page 68.

In this chapter, we describe the installation and configuration of the Access Manager prerequisite software and core components used in the TAMCO solution. The TAMCO environment is a mix of Linux and Windows 2000 platforms. We show the installation on a per-machine basis using the corresponding commands/install wizards.

In order to show the machines/platform configuration, we reproduce the environment shown in Figure 4-11 on page 68, but this time with machine host names (in red) and platform information. The result is depicted in Figure 5-1 on page 76.

As a general *best practice*, it is important to deploy the directory servers on machines with stability, reliability, and high performance in terms of both CPU speed and main memory. Linux servers can often provide this capability to a greater degree than Windows servers. Consequently, the directory server master and replicas are installed on Linux symetric multiprocessor machines.

Similarly, the load balancers require stability and reliable operation, plus networking optimized for throughput. TAMCO's load-balancing solution is hardware-based (Cisco CSS or F5 Big Iron switch).

In our case, WebSphere Application Server does not host a large number of applications, nor does the portal application (just PeopleSoft Application Server HRMS application), so WPM and the Directory Web Administration console can also be deployed on this instance of WebSphere Application Server.

The WebSEAL reverse proxy is replicated to satisfy scalability and availability needs, as discussed in "WebSEAL availability" on page 59. WebSEAL is hosted on Windows servers.

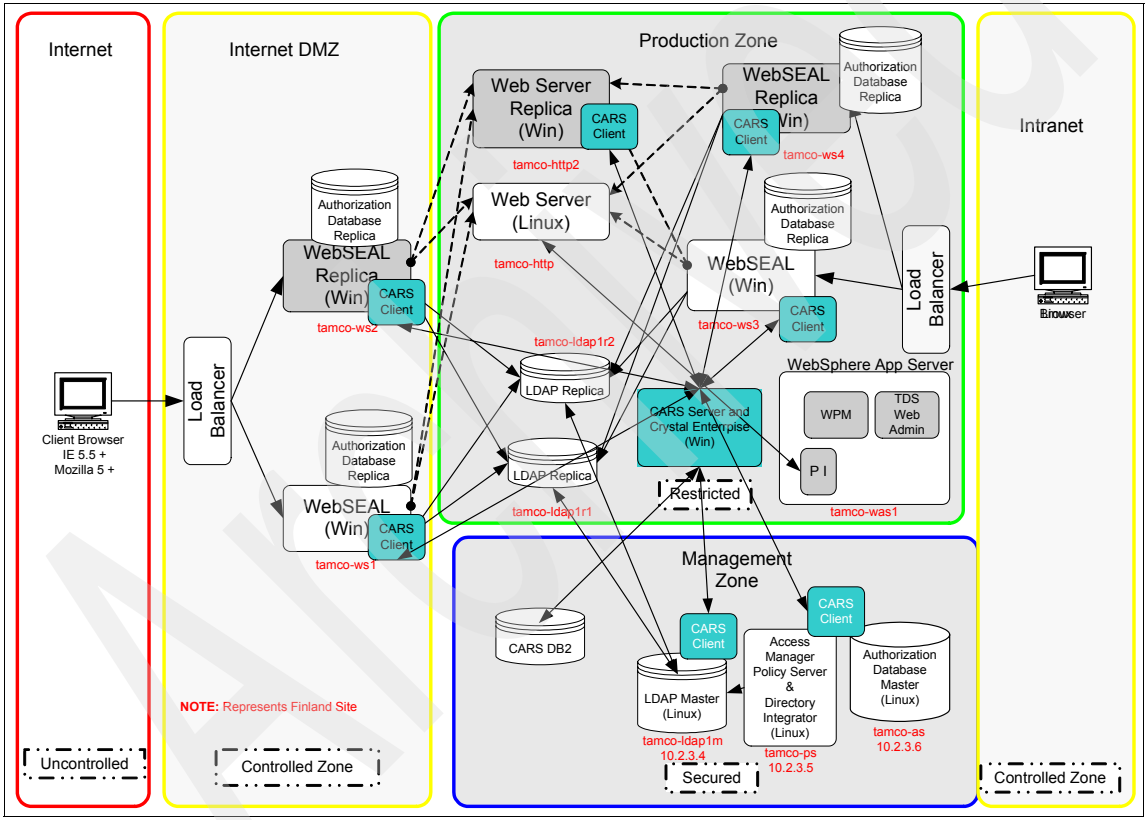

Figure 5-1 TAMCO Access Manager system deployment architecture by machine and platform

5.1 Installing and configuring prerequisites

Prior to installing the Access Manager core components, it is necessary to install the software that provides essential auxiliary functionality:

- IBM Java Runtime V1.4.2 SR 2
- IBM Global Security Kit (GSKit) V7.0.3.17
- IBM DB2 Database Enterprise Server V8.2
- IBM Tivoli Directory Server V6.0.0.2
- IBM WebSphere Application Server V6.0 Refresh Pack 2
- IBM HTTP Server V6
- IBM Tivoli Access Manager Web Portal Manager
- IBM Tivoli Directory Integrator V6.0

Access Manager prerequisite software and core components can be installed in one of three ways:

- An attended automatic install using installation wizards
- A step-by-step native installation method
- A software distribution method to avoid the need to download many CD separate images

For the TAMCO deployment, we use the native installation method, as it provides the most in-depth experience for TAMCO IT administrators for installation and configuration details they will need to understand.

5.1.1 Where to find the CD images

IBM makes the binary code for the Access Manager prerequisite, base, WebSEAL, and Web Portal Manager products available by download from a customer-accessible Web site as CD images (ISO 9660 .iso files).

Table 5-1 shows the names, versions, and Fix Pack levels at the time we wrote this book.

Table 5-1 Access Manager prerequisite software installed at TAMCO

Component name	Version and Fix Pack level
IBM DB2 UDB Enterprise Server Edition	8.2 (8.1 FP8); 8.1 FP9 for Red Hat 4.0
IBM Tivoli Directory Server	6.0.0.2 (6.0.0.1-TIV-ITDS-IF0001)
IBM Global Security Kit (GSKit)	7.0.3.17
IBM Java Runtime	1.4.2 SR2
IBM WebSphere Application Server base[a]	6.0 Refresh Pack 2
IBM HTTP Server	6.0

a. Versions Network Deployment and Network Deployment Edge at this same level may be used.

Table 5-2 shows the names, versions, and Fix Pack levels for the base Access Manager CDs, WebSEAL, and Web Portal Manager.

Table 5-2 Access Manager components and version levels installed at TAMCO

Access Manager component	Version and Fix Pack level
IBM Tivoli Access Manager Base	6.0
IBM Tivoli Access Manager Web Security	6.0
IBM Tivoli Access Manager Web Administration	6.0

Figure 5-2 on page 79 depicts an overview of prerequisites and Access Manager base servers. In the following section, we install these separate components.

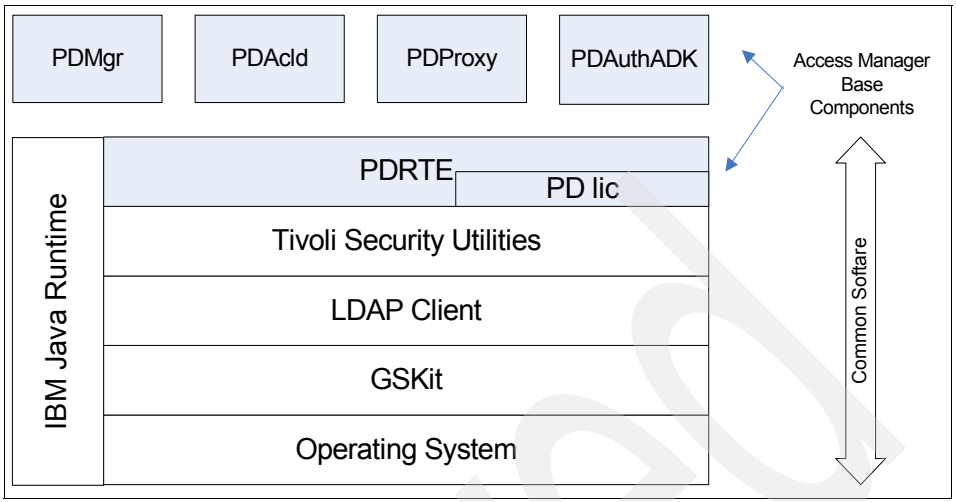

Figure 5-2 Access Manager prerequisite software and base servers

5.1.2 IBM Java Runtime

IBM Java Runtime is installed from the base Access Manager CD image. Mount the CD iso image at /media/cdrom and install an .rpm package:

1. Log in as the root user.
2. Issue the following commands to install the IBM JDK™ V1.4.2 package:

```
# cd /media/cdrom/
# cd linux_i386/
# ls IBMJ*
IBMJava2-SDK-1.4.2-1.0.i386.rpm
# rpm -ihv IBMJava2-SDK-1.4.2-1.0.i386.rpm
IBMJava2-SDK
################################################
# rpm -qa | grep IBMJava
IBMJava2-SDK-1.4.2-1.0
```

3. Verify that the correct Java version is installed by running the following command:

```
# java -version
```

Experience has shown that adding a particular Java Runtime Environment (JRE™) to the system's PATH can cause problems (one of which is the installation of WebSphere maintenance packs through its updateinstaller). In general, software that requires a JRE will provide its own JRE or give instructions on how to configure it to use an existing JRE on the system.

Many installation wizard programs require a JRE in the PATH. Here is a script that adds the JRE IBM Java V1.4.2 JRE to the path:

```
#!/bin/bash
JRE_PATH=/opt/IBMJava2-142/jre/bin
RE=".*$JRE_PATH"
found=`expr "$PATH" : "$RE"`
if [ "$found" = "0" ] ; then
export PATH=$PATH:$JRE_PATH
fi
```

Every BASH shell can automatically use IBM Java V1.4.2 JRE by creating a script called /etc/bash.bashrc.local with the same contents as above.

In order to install IBM Java Runtime on Windows, use the **install** command in the \Windows\Policy Director\Disk1\JRE directory on the base Access Manager CD or CD image.

The same advice applies regarding putting the Access Manager Java path in the Windows path environment variable. You do this by selecting **Control Panel** → **System** → **Advanced**.

5.1.3 Global Security Kit (GSKit)

GSKit is used across many IBM products to provide SSL certificate generation and management, which can sometimes result in version conflict. Although this has been minimized recently, it is still considered a *best practice* to identify the product that provides the highest version level of GSKit and install that package.

Let the following example guide you for this and even future installations with different version numbers:

> IBM HTTP Server 6.0 ships GSKit 7.0.3.6, but Tivoli Directory Server v6.0 ships GSKit 7.0.3.15. However, Tivoli Access Manager V6.0 requires GSKIT 7.0.3.17, which is shipped on the Access Manager CDs (and in the CD image files). Both IBM HTTP Server and Tivoli Directory Server can use the higher level version, so this is the one we will install.

IBM GSKit has to be installed on the following systems (basically, any system that participates in an SSL communication must have GSKit installed on it to be able to consume certificates):

- The Access Manager servers
- The WebSEAL master and replicas
- The Directory Server master and replicas
- The WebSphere Application Server
- The WebSphere Portal server

Installation on Linux

Do the following steps:

1. Log in as root and mount the IBM Access Manager V6.0 base CD or CD image.
2. Find the GSKit installation package, called gsk7bas, in the following directory:

   ```
   /linux_i386/gsk7bas
   ```

3. Enter the following commands:

   ```
   # rpm -ihv gsk7bas-7.0-3.17.i386.rpm
   gsk7bas
   ##################################################
   # rpm -qa | grep gsk
   gsk7bas-7.0-3.17
   ```

4. Verify the correct version is installed by running the command **gsk7ver**:

   ```
   # gsk7ver
   @(#)CompanyName:      IBM Corporation
   @(#)LegalTrademarks:  IBM
   @(#)FileDescription:  IBM Global Security Toolkit
   @(#)FileVersion:      7.0.3.17
   ```

Installation on Windows

To install:

1. Log in as administrator and insert IBM Access Manager V6.0 base CD or mount the CD image.
2. Find the GSKit installation package, called gsk7bas, in the following directory:

   ```
   <CD Drive>:ambaseWIN\windows\GSKit
   ```

3. Go to the ambaseWIN\windows\GSKit directory. Drag the setup.ins file over to the setup.exe. This launches the GSKit install wizard with the correct setup.iss settings.

   Warning: If you try to launch the setup.exe by double-clicking, you will get an error that indicates that setup could not find the .iss file.

4. Verify that the correct version is installed by running the command **gsk7ver**:

   ```
   # gsk7ver
   @(#)CompanyName:      IBM Corporation
   @(#)LegalTrademarks:  IBM
   @(#)FileDescription:  IBM Global Security Toolkit
   @(#)FileVersion:      7.0.3.17
   ```

5.1.4 Setup of IBM Tivoli Directory Server

Before we begin the installation, let us take a closer look at some general considerations. The procedure for the configuration of a working directory server is:

1. Create user IDs for the Directory Server instance owner and, for some installations, the database instance owner and the database owner.
2. Install Directory Server V6.0 and create a Directory Server instance.
3. Set the IBM Tivoli Directory Server administrator distinguished name (DN) and password for the Directory Server instance. This operation can be compared to defining the root user ID and password on a UNIX system.
4. If the Directory Server instance is not a proxy server, configure the database that holds the directory entries.

In order to install the directory server master, we carry out steps 1 through 4. The procedure for the installation of the replica servers is exactly the same. The configuration of replicas is done at a later stage and is described in Chapter 6, “Configuring IBM Tivoli Access Manager” on page 123.

We will be using the installation wizard for Directory Server V6.0 provided by Access Manager. This Java-based installer (ISMP installation wizard) does all of the above automatically, plus more. In addition to installing Tivoli Directory Server, the wizard installs DB2 UDB Enterprise Server V8.2 as the datastore for Directory Server V6.0, sets up a DB2 database, creates the Access Manager container under which Access Manager stores its data (the *secAuthority=Default* suffix), adds an LDAP suffix for users, and creates the base object for this user suffix.

Using the installation wizard is a quick and easy way to get a baseline directory server for Access Manager installed and configured. However, by itself, it is not adequate for a production zone deployment. The ISMP installation wizard for Directory Server V6.0 does not provide the granularity of control required for the TAMCO production environment (for example, it does not create a specific *uid* or *gid* for specific users).

During the configuration, we will create the correct TAMCO users and groups before installing the Access Manager Directory Server V6.0 user registry by following the procedure described in the following section.

Common user and group for Directory Server V6.0 installed files

There needs to be a required *system level* user that belongs to an appropriate system group created at the operating system level in order to ensure that the directory and DB2 will run properly. Thus, we have to create the following items using the operating system tools (for example, in Linux we use Yast2):

1. Group idsldap: This group name is required and is used as the *group owner* of the Directory Server V6.0 files shared by all Directory Server V6.0 instances.
2. User idsldap: This user name is required and is used as the *user owner* of the Directory Server V6.0 files shared by all Directory Server V6.0 instances. The primary group for this user should be *idsldap*.

User and group for Directory Server V6.0 instance

The Access Manager Directory Server V6.0 installer creates an instance of Directory Server V6.0, that is, a specific configuration of the Directory Server V6.0 product that uses the directory server installed binaries. Like all files on the system, the Directory Server instance files must have a user and group owner. The convention (and default) is to use *ldapdb2* for both the user and group name of instance files.

To access the common files, the instance user must be a member of group *idsldap*.

When the Directory Server server instance starts, it also starts DB2. However, in order to be sure DB2 starts properly, a user's primary group must also have root as a member. Therefore, we have to create the following Access Manager specific user registry user and group, and then edit their attributes as follows:

1. Group ldapdb2: This group name can be changed, if desired.
2. User ldapdb2: This user name can be changed, if desired.
3. Edit the ldapdb2 group and add root as a member.
4. Edit the idsldap group and add user ldapdb2 to the idsldap group.

When complete, a display of the groups should show the members shown in Figure 5-3 on page 84.

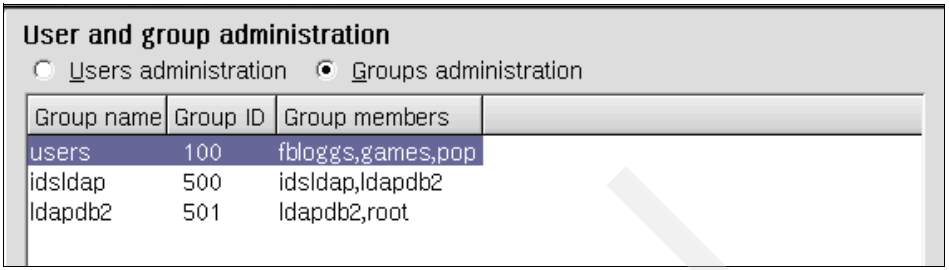

Figure 5-3 User and group information stored in DB2 as shown in Yast2

Install DB2 Universal Database Enterprise Edition

We use the directory installer script called *install_ldap_server* found at the root level on the Access Manager Directory Server CD to install the correct DB2 database that serves as the backing datastore for the TAMCO directory. When we invoke the script, it first checks to see whether the correct Java Runtime version is available. Since we just installed it earlier, we know that it is.

As this script runs, it automatically installs and configures an instance of DB2 Universal Database Enterprise Edition Version 8.2.

The DB2 install is done automatically with very little configuration other than setting the DB2 administrator ID and password. For complete control over the DB2 environment, users, and groups, use the native IBM Tivoli Directory Server 6.0 installation and configuration procedures documented in the Directory Server 6.0 publications, which can be found at:

http://publib.boulder.ibm.com/infocenter/tivhelp/v2r/index.jsp?toc=/com.ibm.IBMDS.doc/toc.xml

Install Tivoli Directory Server

The following steps are performed to install Tivoli Directory Server V6.0 as the Access Manager user registry on our Linux machines, as shown in Figure 5-1 on page 76:

1. Mount the CD or CD image, and browse the contents of the CD. You see the following directories and files:

```
dr-xr-xr-x    7 root     root         4096 Mar 28 23:13 EIC
dr-xr-xr-x    3 root     root         2048 Mar 28 23:13 LAP
 r-xr-xr-x    1 root     root          188 Mar 28 23:13 am_update_ldap.sh
dr-xr-xr-x    2 root     root         6144 Mar 28 23:13 common
dr-xr-xr-x    3 root     root         2048 Mar 28 23:13 help
 r-xr-xr-x    1 root     root       660310 Mar 28 23:13 install_ldap_server
 r-r-r--      1 root     root      9487993 Mar 28 23:13 install_ldaps_setup.jar
dr-xr-xr-x    4 root     root         6144 Mar 28 23:13 linux_i386
dr-xr-xr-x    2 root     root         4096 Mar 28 23:13 rspfile
```

```
dr-xr-xr-x    2 root     root         2048 Mar 28 23:13 spd
 r-xr-xr-x    1 root     root         2984 Mar 28 23:13 tamtblcp.ksh
```

2. Start the installation utility with the **`install_ldap_server`** command. First, a language selection window opens to allow you to select the language that is used for the rest of the installation.
3. The next window displays for accepting or rejecting the license. Accept the license.
4. The next series of windows to open are the windows for configuring the directory server. The first window (Figure 5-4 on page 86) depicts the DB2 configuration data. There are two important fields. The first is the DB2 administrator ID. It already exists because we created it above. Also note that the *Group for DB2 administrator* pull-down contains the value we previously created.
5. An *encryption seed* needs to be provided. It can only use ISO-8859-1 ASCII characters with values in the range of 33–126. The field has to contain a minimum number of 12, and no more than 1016 characters.

Figure 5-4 Directory Server configuration window for DB2 configuration

6. The next window, shown in Figure 5-5 on page 87, is used to configure the actual Directory Server instance with the information that is needed for future administration of the directory, which can be performed with standard LDAPAPI commands or by use of the Web Admin console. Provide the following information in this dialog:

   Administrator ID Specifies the administrative user for the LDAP Server. The convention for this user is cn=root.

   Administrator password
   The password for cn=root.

   Password confirmation
   Enter the password again.

User-defined suffix Defines a suffix that is created in the Directory Server for storing users and groups.

Local host name The fully qualified host name of the local computer.

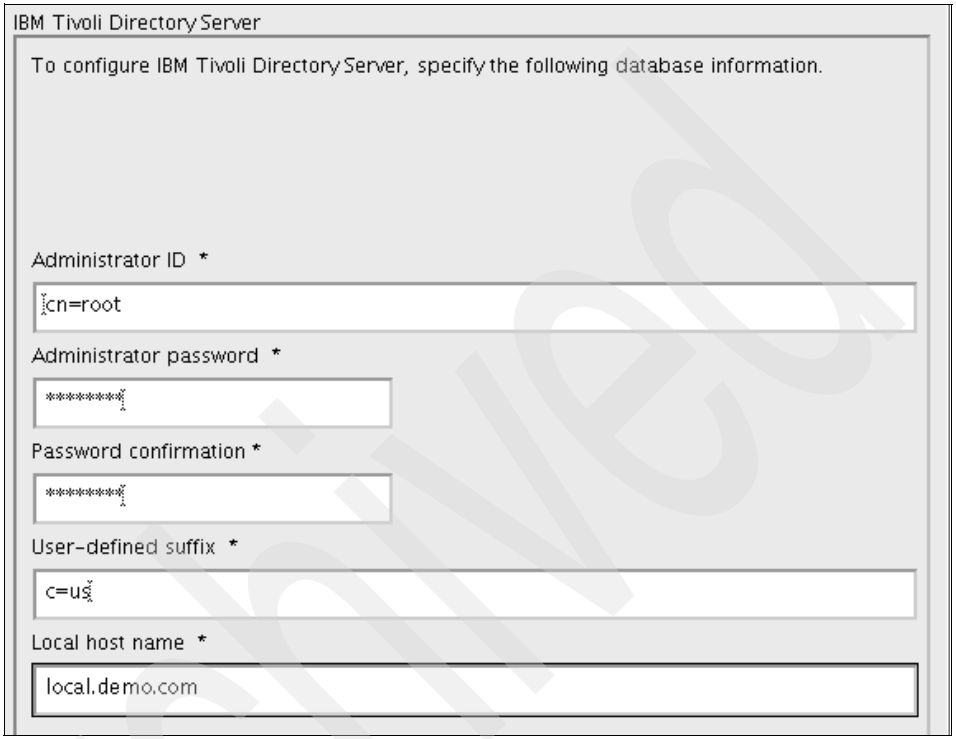

Figure 5-5 The Directory Server administration configuration

The next window (Figure 5-6) shows the Directory Server communication path and SSL configuration that is used for the unsecure and secure ports, the path to the SSL key file (must be a fully qualified path), the SSL key file password, and the label of the digital certificate.

Note the two check boxes: one for creating the SSL key file, and the other for enabling the Federal Information Processing Standards (FIPS) cryptographic algorithms for TLS V1. If you enable the FIPS algorithms, they *must* be enabled when configuring other Access Manager Runtime and base servers as well or the overall configuration will fail.

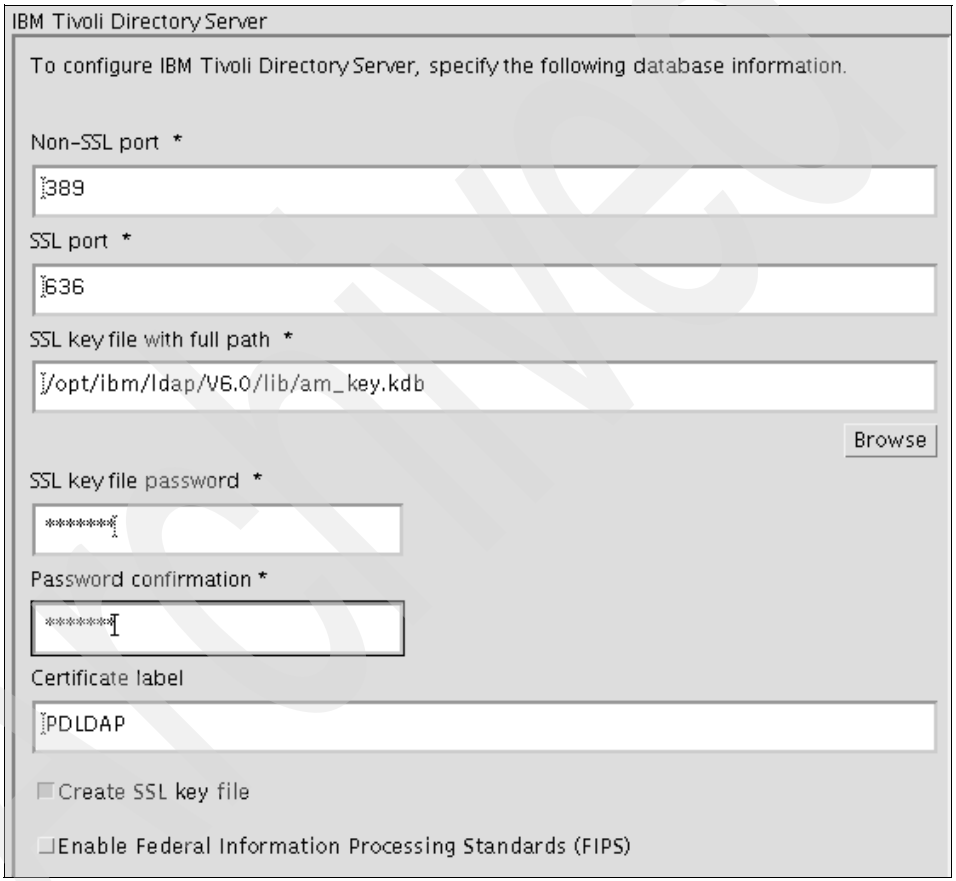

Figure 5-6 Directory Server SSL communication and configuration choices

7. The next window shows you the files and space needed for the DB2 and Directory Server instances. After this window, a summary window is displayed. This summary window is the last dialog that allows you to cancel the overall operation. When you choose **Next**, the installation and configuration displayed in the summary starts. It takes several minutes to complete. You can watch the process by running the following command:

   ```
   # tail -f /tmp/msg__ldaps_install.log
   ```

8. When the configuration finishes, you see the *Status Details* window, as shown in Figure 5-7. Click **Finish** to exit the installer.

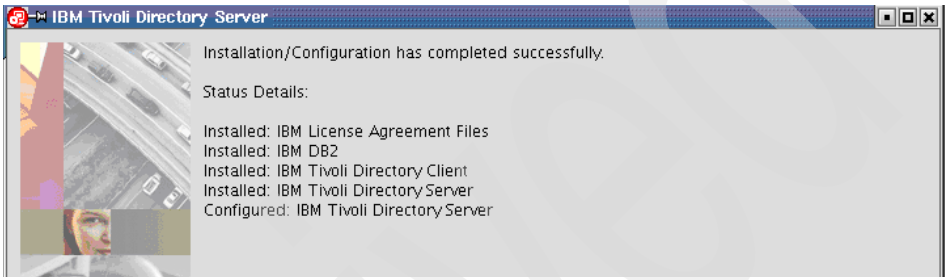

Figure 5-7 Successful installation/configuration of Directory Server

9. To verify the installation, run **ldapsearch** as follows:

   ```
   idsldapsearch -D cn=root -w -passw0rd -b "" -s base objectclass=*
   ```

   This command returns the whole contents of the directory.

10. A number of the Access Manager servers as well as WebSphere require the Directory Server to be running when they start. Therefore, it makes sense to configure the Directory Server to autostart. In Windows, you can change the autostart behavior of IBM Tivoli Directory Server V6.0 from *manual* to *automatic* using the *Services* dialog. In Linux, you can edit the /etc/inittab to comment out the following two lines:

    ```
    fmc:2345:once:/opt/IBM/db2/V8.1/bin/db2fmcd #DB2 Fault Monitor
    Coordinator
    ids0:2345:once:/opt/ibm/ldap/V6.0/sbin/ibmdiradm -I ldapdb2 >
    /dev/null 2>&1
    ```

11. Next add the following script into the /etc/.profile file:

```
#!/bin/ksh
echo "create IDS start script in /etc/rc.d..."
cat > /etc/rc.d/startIDSldap << catInput
idsslapd
catInput

chmod 755 /etc/rc.d/startIDSldap

echo "create link for level 3 and level 5 autoStart ..."
ln -s  /etc/rc.d/startIDSldap   /etc/rc.d/rc3.d/S580startIDSldap
ln -s  /etc/rc.d/startIDSldap   /etc/rc.d/rc5.d/S580startIDSldap
```

This concludes the installation of the Tivoli Directory Server component on the LDAP master *tamco-ldap1m* and both LDAP replicas *tamco-ldap1r1* and *tamco-ldap1r2*, as shown in Figure 5-1 on page 76. As previously said, we will configure both replicas at a later time.

IBM Tivoli Directory Server client

You must install IBM Tivoli Directory Server client on each system that runs a Tivoli Access Manager component. We install the Directory Server client on all machines in the TAMCO Access Manager environment. These machines are:

- The tamco-ws Access Manager WebSEAL server and replica machines
- The tamco-ps Access Manager Policy Server machine
- The tamco-pp Access Manager Policy Proxy Server machine
- The tamco-ldap Directory Server replicas
- The tamco-was1 machine hosting WebSphere Application Server
- The PeopleSoft and Siebel application server machines

The client application is provided on the CDs containing the Directory Server, the Access Manager base components, and the Access Manager Web Security (WebSEAL), so it can be installed when installing those components.

On Linux hosts, use the following commands to install the three client packages:

```
#rpm -ihv idsldap-clt32bit60-6.0.0-2.i386.rpm
idsldap-clt32bit60
##############################################################
#rpm -ihv idsldap-cltbase60-6.0.0-2.i386.rpm
idsldap-cltbase60
##############################################################
#rpm -ihv idsldap-cltjava60-6.0.0-2.i386.rpm
idsldap-cltjava60
##############################################################
```

On the Windows host for WebSEAL, run the setup.exe from the \windows\Directory directory. Next, select the IBM Tivoli Directory client, as shown in Figure 5-8, and click **Next**.

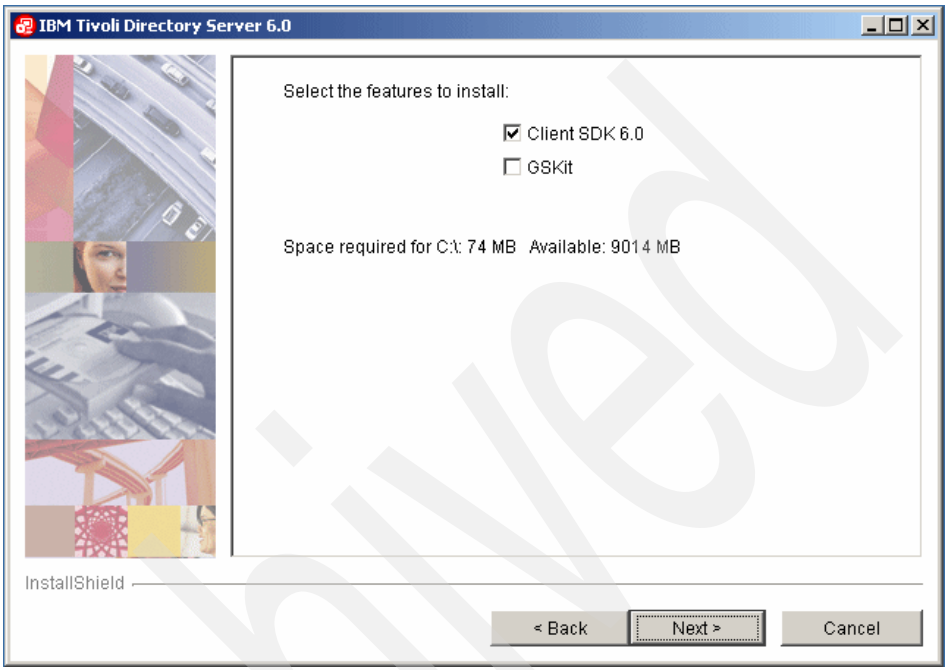

Figure 5-8 Directory Server client installation on Windows

The Directory Server client installation ends with a pop-up window indicating that the installation completed successfully.

5.1.5 IBM WebSphere Application Server

The Directory Server provides an optional Web-based administration tool that can be installed. This tool is provided as a .war file that is installed into the WebSphere Application Server container as an enterprise application. It runs as a WebSphere plug-in, and requires an instance of WebSphere Application Server. The installation package for the Directory Server that ships with Access Manager provides a version of WebSphere Application Server *Express*.

Unfortunately, the Express level of WebSphere Application Server is not adequate to run the Access Manager Web Portal Manager, the Web-based graphical user interface for Access Manager administration. Rather than have two instances of WebSphere running in the environment, it makes more sense to deploy the Directory Server Web Administration utility on the same WebSphere Application Server instance as Web Portal Manager.

Therefore, we install a base WebSphere Application Server V6.0 with Refresh Pack 2 that comes on the IBM Tivoli Access Manager Web Administration CD (amwpm.iso).

Directory Server Web Administration utility

The decision about deploying the Directory Server Web Administration tool depends on the preferences of the administrators of the directory and the type of administrative tasks envisioned. In the TAMCO environment, the main administrative tasks involving the Access Manager user registry involve searching and browsing the directory, not making changes directly to it. Hence, tasks such as schema extension, editing entries manually, and so on, are not needed. For these non-intervention tasks, any open source or freeware tool to browse an LDAP directory may prove adequate.

Access Manager Web Portal Manager (WPM)

WPM provides a Web-based graphical user interface for administering Access Manager users, groups, domains, object spaces, and ACL/POP/authorization rule management. It is not intended for bulkloading or importing large numbers of users or administration tasks involving large numbers. These are best done using LDAP utilities or PDADMIN commands, the Access Manager command-line interface for administration. WPM provides a subset of PDADMIN commands.

5.1.6 Installing WebSphere Application Server V6.0

There are two main steps in the installation of Access Manager's WebSphere Application Server. The first is to install the 6.0.0 binary. The second step is to use the *updateinstaller* to install the Refresh Pack 2.

Before going through the installation steps, it is a good idea to look at the WebSphere Application Server scenario most applicable to the TAMCO environment.

For performance reasons, it is not a good idea to overload a single instance of WebSphere Application Server with too many applications that are input/output or main memory intensive. In order to overcome the impact, WebSphere Application Server can be deployed with a separate Web server, which requires

the Web server plug-in for communication with the WebSphere Application Server.

Given that TAMCO is only deploying a few business applications (PeopleSoft and WebSphere Portal) plus two Access Manager environment-specific applications (WPM and the Directory Server Web Administration console) that are hosted on WebSphere Application Server, there is no need to deploy an additional Web server and Web server plug-in.

Step one: Install and configure base WebSphere

WebSphere Application Server in the TAMCO environment is being deployed on a Linux server called *tamco-was1*, according to Figure 5-1 on page 76. The installation of WebSphere starts by mounting the product CD or CD image. We install a stand-alone application server (called by default *server1*). A script is included on the CD root directory called *launchpad*, which requires either Microsoft Internet Explorer® V5.5 or later or Mozilla V1.4 or later, and is the starting point for the installation. This installation method creates one default profile.

WebSphere provides a profile creation wizard to create multiple stand-alone application servers if necessary. It is not necessary in each TAMCO domain to have more than one WebSphere Application Server, so we will only be installing a single application server.

Start the launchpad script. You are prompted to launch the ISMP installshield wizard, which progressively steps through several installation windows.

After going through the first few windows that ask for language and license acceptance, we come to the System prerequisites check window, as shown in Figure 5-9. If you fail this check, you can find the supported hardware and software described at the following Web site:

`http://www.ibm.com/software/Webservers/appserv/doc/latest/prereq.html`

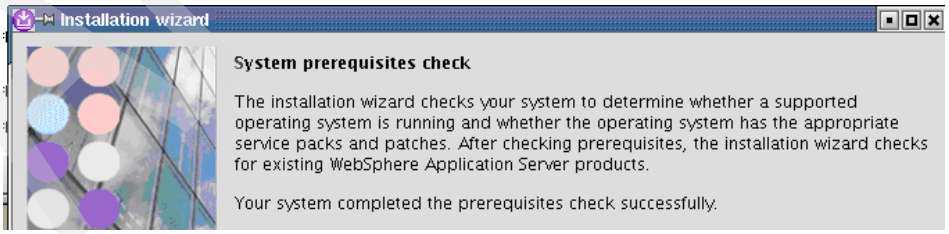

Figure 5-9 WebSphere installer System prerequisites check window

Once the installer has determined that the prerequisites have been met, click **Next** to display the Installation directory window (Figure 5-10).

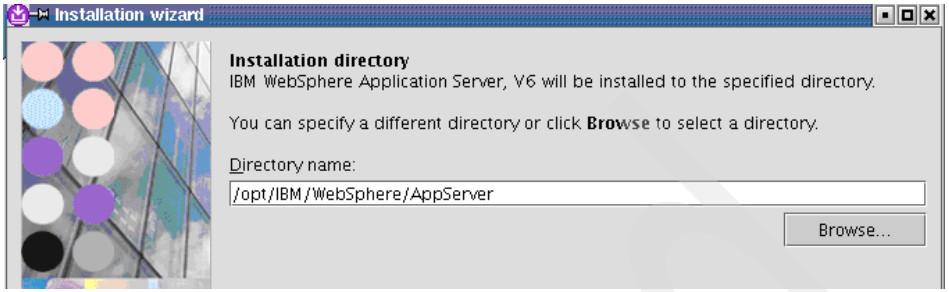

Figure 5-10 WebSphere default installation home directory

Here are two important facts for UNIX and Linux installations:

- Do not use symbolic links as the destination directory. Symbolic links are not supported.
- Spaces are not supported in the name of the installation directory.

Also note that the default system installation path is different from how it was in WebSphere Application Server V5.1.1 (the addition of IBM to the path).

We progress through several more windows that provide installation information. After this, a summary page is displayed, as shown in Figure 5-11.

Note: The next window is the last one you see before the installation begins, and the last chance to go back to make changes. Once the installation begins, it can take several minutes.

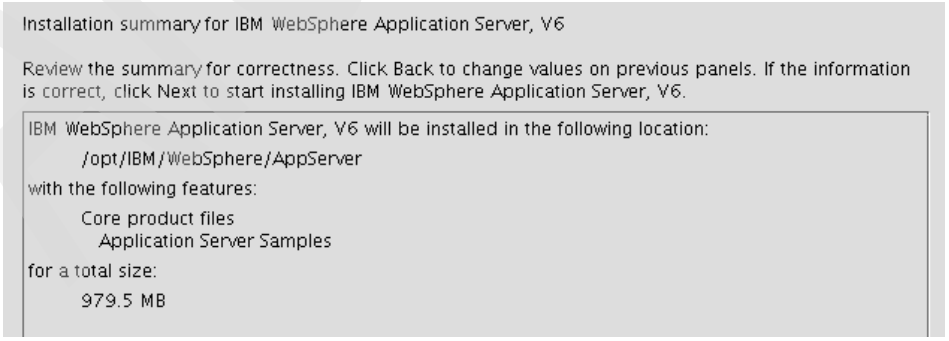

Figure 5-11 WebSphere installation summary

You can use the following command to follow the progress of the installation:

```
tail -f /opt/IBM/WebSphere/AppServer/logs/log.txt
```

Once the installation completes, you are presented with the window shown in Figure 5-12. Select **Launch the First steps console**.

Figure 5-12 WebSphere window confirming successful installation

When the First steps console is displayed, click the **Installation verification** link, as shown in Figure 5-13. Several checks are being performed, the most important of which is that the ivt script attempts to start the WebSphere Application Server instance.

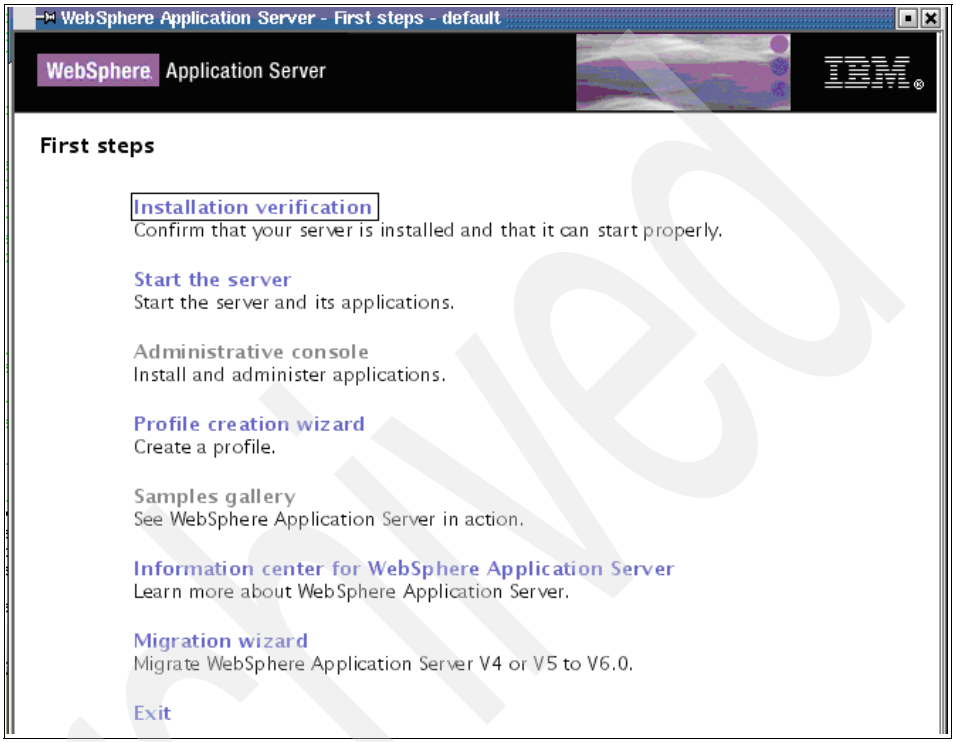

Figure 5-13 WebSphere Application Server first steps for installation verification

Installing the Refresh Pack 2

It is necessary to install Refresh Pack 2 onto the base WebSphere Application Server V6.0 in order to update certain classes and jar files needed by Access Manager. We use the *updateinstaller* used by WebSphere for applying fix packs and patches.

The Refresh Pack 2 is included in the installation images for Access Manager in the WebSphere directory under the Access Manager Base CD image.

Installing the Refresh Pack 2 uses the standard WebSphere *launchpad* process for installing maintenance packs.

5.1.7 IBM HTTP Server

While WebSphere Application Server comes with a built-in Web server, it is not as robust and flexible in production scenarios as a stand-alone Web server.

WebSphere Application Server V6.0 includes IBM HTTP Server V6.0 (IHS V6.0). Both WebSphere Application Server base and WebSphere Network Deployment ship IHS V6.0 on the CD.

In general, Web clients (browsers) should not connect directly to a WebSphere Application Server. Instead, they should access the application server through an HTTP Server that is configured with the WebSphere plug-in. There are two reasons for this:

- The WebSphere Application Server does not handle the high volume of TCP connect/disconnect traffic that is typical of client activity. The HTTP server with the WebSphere plug-in does a much better job of this and pools connections to the WebSphere Application Server.
- The load-balancing and failover logic to distribute Web clients across the members of an application server cluster resides in the HTTP Server plug-in.

Let us now install IBM HTTP Server V6.0. The IHS V6.0 package is provided on the Access Manager WebSphere V6.0 CD.

Start the ISMP installer by running the **install** command:

```
./install
```

The welcome window is shown in Figure 5-14. Click **Next** to continue.

Figure 5-14 IBM HTTP Server installer welcome window

Accept the license in the next window and continue to the Installation Completed Successfully window. If you do not get this window, then launch the WebSphere Application Server plug-in install window, uncheck the plug-in install, and click **Finish**.

Run the following command to start IBM HTTP Server:

```
./apachectl start
```

The default installation of IBM HTTP Server uses the local host name for the *ServerName* directive and a *Listen* directive that binds to port 80 for all IP addresses on the machine. When the server starts, it considers the name of the server to be the same as the host name of the system and accepts connections to any IP address:80.

When we install WebSEAL, we want WebSEAL to *own* the connections for `my.tamco.com:80`. This means that IHS must be restricted to only listen to one of the IP aliases created in the Create Virtual Interfaces dialog. The DNS name that we use for IBM HTTP Server is `ihs.fi.tamco.com`, and this should also be the name of the server (which is specified in the *ServerName* parameter of the Web server's httpd.conf file).

As mentioned earlier, Web clients (browsers) should not connect directly to a WebSphere Application Server. Instead, they should access the application server through a Web server that is equipped with the WebSphere plug-in.

The WebSphere plug-in examines each request received by the Web server and determines whether the requested URL corresponds to a WebSphere application. If there is a match, the plug-in forwards the request to the application server.

Using the WebSphere plug-in requires these high-level steps:

1. Install the WebSphere plug-in libraries.
2. Configure the Web server to load the WebSphere plug-in.
3. Configure the plug-in using the plugin-cfg.xml file that lists application URLs and the application server that contains them.
4. Enable the application server to update plugin-cfg.xml as applications are added or removed.
5. As usual, all this can be done using an installation wizard. First, we have to stop WebSphere Application Server using the following command:

   ```
   <WAS_install_path>/bin/stopServer.sh server1
   ```

6. When you run the **`../plugin/install`** command to start the installer, you see the window shown in Figure 5-15. Select the **Installation roadmap** to see a description of the Web server plug-in and roadmap for the various usage scenarios.

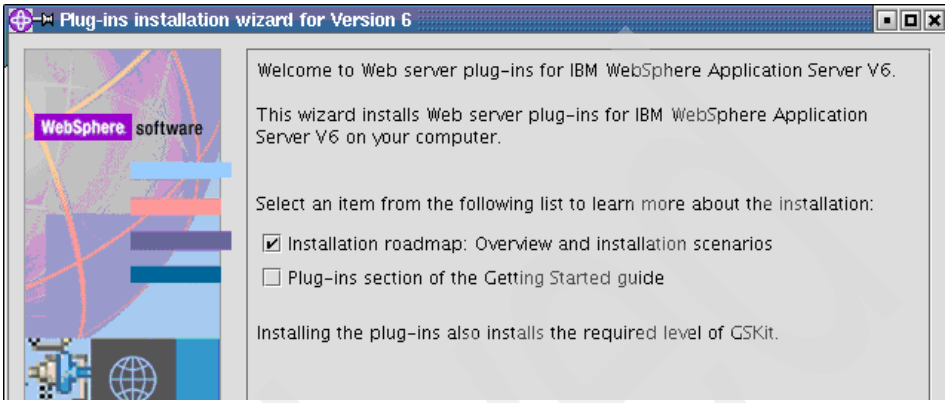

Figure 5-15 Web server plug-in roadmap

7. Do the system prerequisite check and, if successful, click **Next** to be taken to the window (shown in Figure 5-16) that allows the selection of different Web servers.

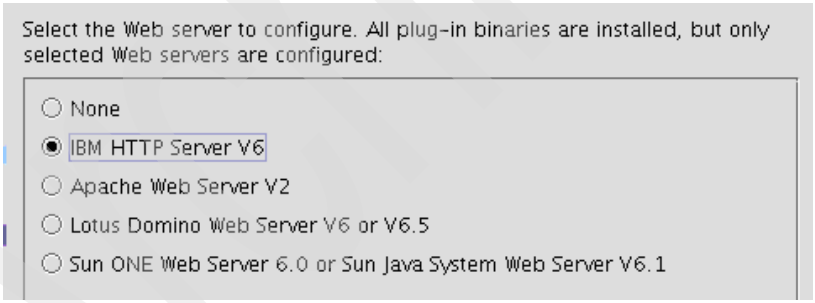

Figure 5-16 Web server installer options window showing supported Web servers

Note: The complete installation and configuration of the WebSphere plug-in is documented at:

`http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp`

8. Since we are deploying the Web server on the same physical machine as the WebSphere Application Server, you have to select the local choice, as shown in Figure 5-17.

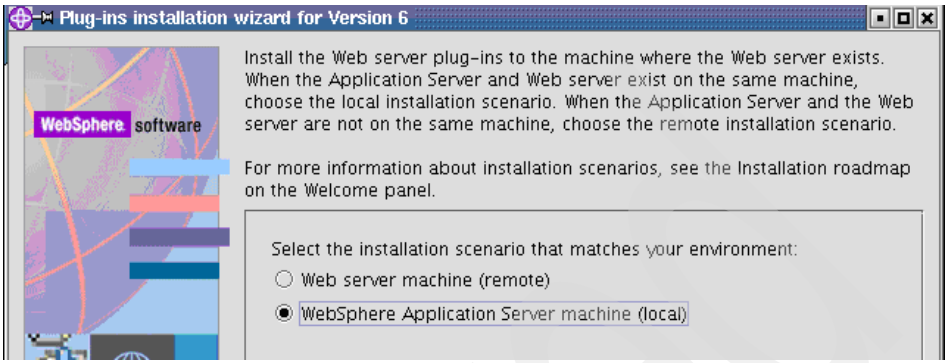

Figure 5-17 Location options for Web server plug-in

9. In the next window, accept the default installation directory (usually /opt/IBM/WebSphere/plugins). Next, you are asked to provide the installation directory where WebSphere Application Server is installed.

10. The next window will ask for the location of the Web server configuration file (httpd.conf). Use the **Browse** button to locate it (usually /opt/IBMIHS/conf/).

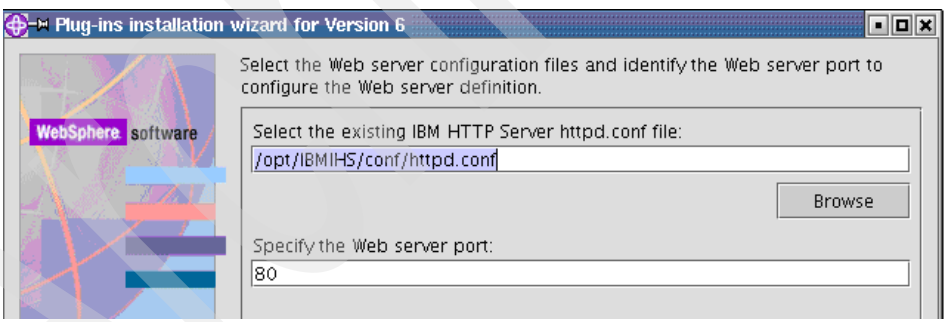

Figure 5-18 Specify the location of the httpd.conf file and the Web server listening port

11. Next you are asked to supply a unique Web server definition name, which the WebSphere installation document calls a *nickname*. Choose an appropriate name (Webserver1 is provided as the default).

12. When you click **Next**, you see the window in Figure 5-19. This window provides important configuration information.

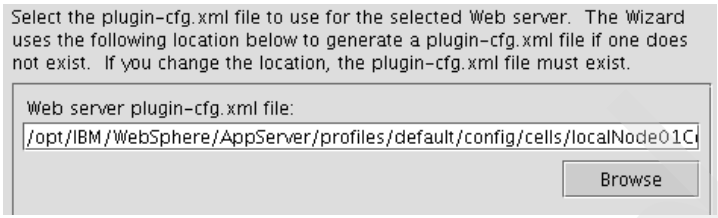

Figure 5-19 Location of the Web server plugin-cfg.xml file

13. The location of the plugin-cfg.xml file must be coordinated between the HTTP Server's configuration file and the WebSphere process for updating it. As changes are made to the applications that run on the WebSphere Application Server, the plugin-cfg.xml file must be re-generated.

    The installation wizard uses the following template for the location of the plugin-cfg.xml file:

```
<WAS-install>/profiles/<profile-name>
                /config/cells/<cell-name>
                     /nodes/<node_name_of_AppServer>

/servers/<Web_server_name>/plugin-cfg.xml
```

    Substituting the values for our environment results in:

```
/opt/IBM/WebSphere/AppServer/profiles/default/
         config/cells/localNode01Cell
             /nodes/Webserver1_node
                  /servers/Webserver1/plugin-cfg.xml
```

    The following path and file name will be used in the configuration statement added to the httpd.conf file:

```
LoadModule was_ap20_module
   /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
       WebSpherePluginConfig
           /opt/IBM/WebSphere/AppServer/profiles/
             default/config/cells/localNode01Cell/

nodes/Webserver1_node/servers/Webserver1/plugin-cfg.xml
```

    This configuration information has to be entered all on one line; it is reformatted here for readability.

14. Make sure the location is correct and click **Next**. Two summary windows appear next, and if the information is correct, click **Next** on both (the second **Next** actually starts the installation). At the end of the installation, you are presented with the information shown in Figure 5-20 confirming a successful installation of the WebSphere plug-in.

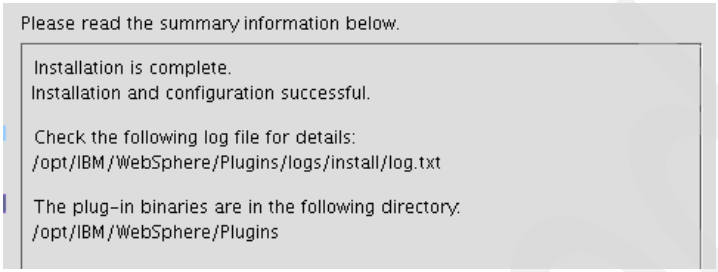

Figure 5-20 Web server installer window showing successful installation

5.1.8 IBM Tivoli Directory Server Web Administration

As discussed earlier, Directory Server V6.0 supplies a Web Administration application that runs on the WebSphere Application Server as an enterprise application. The package that includes this application is shipped on the Access Manager Web Administration CD rather than on the Directory Services CD.

Mount the CD and install the Directory Server Web Administration fileset using the following command:

```
# cd /media/cdrom/linux_i386
# rpm -ivh idsldap-Webadmin60-6.0.0-0.i386.rpm
idsldap-Webadmin60
##################################################
```

Now you have installed the Directory Server Web Administration application as a Web archive (.war) file in the Directory Server file system as /opt/ibm/ldap/V6.0/idstools/IDSWebApp.war. Next, this application has to be installed into your WebSphere Application Server as an Enterprise Application using the WebSphere Administration console using the *install new application* procedures.

Note: For more details, see *IBM Tivoli Directory Server Installation and Configuration Guide Version 6.1*, GC32-1560 at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.IBMDS.doc/toc.xml

Once the Directory Server Web Administration utility is installed, you have to connect to the Directory Server using the following URL to test the installation:

```
http://tamco.com/IDSWebApp/IDSjsp/Login.jsp
```

The login form, shown in Figure 5-21, is displayed. The first time you open the connection to the Directory Server, you need to configure console operations and add the correct Directory Server for administration. To do this task, you have to use the Administrator ID that is reserved for the Directory Server's root administrator *superadmin* with the password *secret*. This ID and password are rarely used once the relevant Directory Servers are configured.

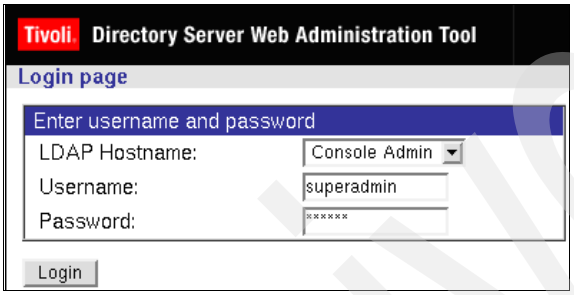

Figure 5-21 Login to the Directory Server Web Administration Tool

Once logged in as superadmin, click **Add Server**. The following dialog, shown in Figure 5-22, allows you to set the host name and ports, as well as enable SSL encryption. Set the host to tamco-ldap1m.fi.tamco.com and the port to 636.

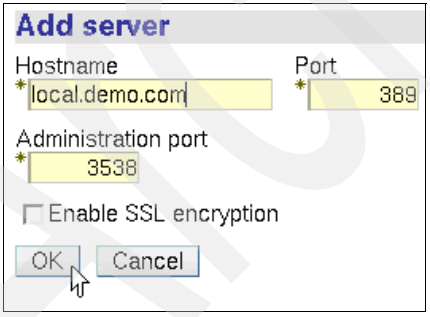

Figure 5-22 Configuring the Directory Server Administration Tool connection

To verify that you are able to connect to the Directory Server, click **OK** to return to the main menu and then select the **Logout** menu option. On the Logout Successful page, click the **You can re-login by clicking here** link to go to the login page again.

On the login window, use the drop-down menu to select the new option for tamco-ldap1m.fi.tamco.com:636. Enter the user name as cn=root and the password as passw0rd. Then click **Login** to get to the login page. Enter the following values:

LDAP Hostname	tamco-ldap1m.fi.tamco.com:636
Username	cn=root
Password	passw0rd

If the connection to the Directory Server is working, you will see the window shown in Figure 5-23.

Figure 5-23 Main navigation window in the Directory Web Administration console

This concludes the installation of the Directory Server Web Administration console.

5.1.9 IBM Tivoli Web Portal Manager (WPM)

Figure 5-24 on page 105 shows the components that have to be installed for WPM.

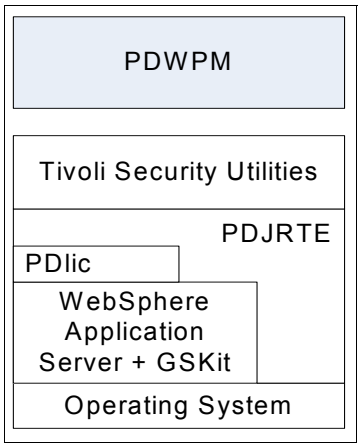

Figure 5-24 Access Manager's Web Portal Manager components

WPM installation code comes on its own CD called IBM Tivoli Access Manager Web Administration. As with the other Access Manager components, WPM installs with an ISMP installer.

Also, as with the Directory Server installation package, WPM installs as a .war file and hence requires hosting on our WebSphere Application Server. Therefore, the WPM CD includes the Express version of WebSphere. However, the WPM ISMP installer has the ability to recognize and use an existing WebSphere instance, so rather than installing the version of WebSphere Application Server that comes with the WPM installation code, we will use the existing WebSphere Application Server. When making this decision, care needs to be exercised to ensure that the version and patch level of the existing WebSphere meet the requirements of WPM. In the TAMCO case, we are using WebSphere Application Server V6.0.2 that ships with the Access Manager base, which exceeds the requirements of WPM.

Let us take a look at the content of the installation directory by executing the following commands:

```
# cd /media/cdrom/linux_i386/
# ls
.
..
PDWPM-PD-6.0.0-0.i386.rpm
PDlic-PD-6.0.0-0.i386.rpm
IBMJava2-142-ia32-SDK-1.4.2-1.0.i386.rpm
idsldap-Webadmin60-6.0.0-0.i386.rpm
PDJrte-PD-6.0.0-0.i386.rpm
migrate
```

First, we have to install IBM Java Runtime with the following command:

```
# rpm -ihv PDJrte-PD-6.0.0-0.i386.rpm
PDJrte-PD
################################################
```

Then we can install the Web Portal Manager with the following command:

```
# rpm -ihv PDWPM-PD-6.0.0-0.i386.rpm
PDWPM-PD
################################################
```

Since we have not installed and configured the Access Manager base servers that WPM will manage, we have to delay the WPM configuration until these servers are installed and configured. This is covered in Chapter 6, “Configuring IBM Tivoli Access Manager” on page 123.

5.1.10 IBM Tivoli Directory Integrator

Directory Integrator is shipped on the Access Manager Directory Server CD image. To install it, launch the setupLinux.bin from the CD directory.

The installer automatically looks for the correct Java version. If it finds it, you see the Welcome window shown in Figure 5-25 on page 107. If it does not find an adequate Java version, you need to install Java 1.4.2 or later.

Figure 5-25 Directory Integrator welcome window

Click **Next** and you are taken through several preliminary installation windows. First accept the license, and then accept the defaults for the installation path.

In the next window, you are asked to select the home directory to store your Directory Integrator solutions. You are given four options designed to provide you with the most flexibility. If you choose **Do Not Specify**, then the solution directory varies from run-to-run, storing the solutions in the directory from which Directory Integrator is run.

Select **Use a tdi subdirectory under my home directory**, as shown in Figure 5-26.

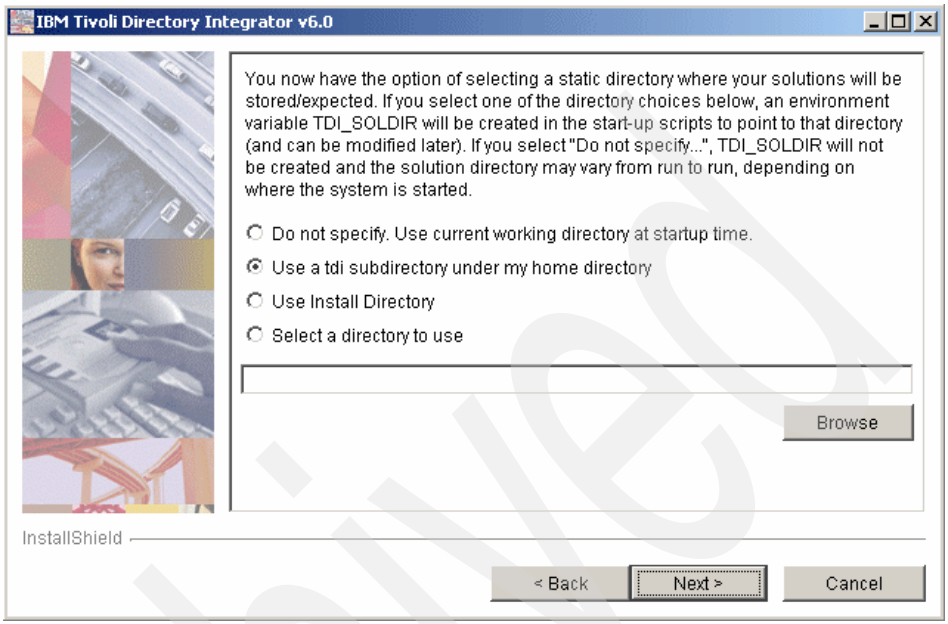

Figure 5-26 Directory Integrator solution directory options

Once you have made these choices, the next window to display is the summary window (Figure 5-27 on page 109). This is the last window before the actual installation begins.

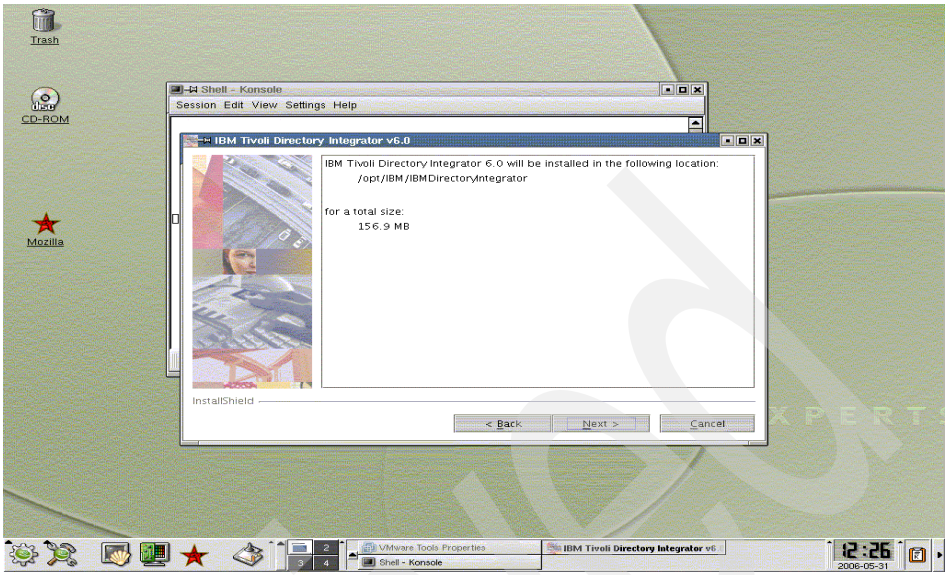

Figure 5-27 Directory Integrator summary window

Once you click **Next**, the installation begins. It takes a short time during which you see a progress bar. At the end of the installation, you see a success window.

Click **Finish** to complete the installation. At this point Directory Integrator is ready to use. Specific components for bulkloading and synchronizing the PeopleSoft HRMS Database with Directory Server are added to Directory Integrator and configured, as discussed in 6.1.2, “Configuring Access Manager components” on page 128.

5.2 Installing Access Manager base components

Once the prerequisite software is installed, we next install the base Access Manager servers, WebSEAL, and the Common Auditing and Reporting Services. These components are found on the CD or CD image called IBM Tivoli Access Manager Base CD (or ambase<platform>.iso) and IBM Tivoli Access Manager Web Security CD (amWebsec<platform>iso).

The base components (with package names) we are installing for TAMCO consist of:

- Access Manager License (PDlic)
- Access Manager Web Security utilities (TivSecUtil)
- Access Manager Runtime (pdrte)
- Access Manager Policy Server (pdmgrd)
- Access Manager Policy Proxy Server (pdmgrproxyd)
- Access Manager Authorization Server (pdacld)
- Common Auditing and Reporting Services

There are other Access Manager components included in the Access Manager base;, however, TAMCO does not intend to use them at this time. These are:

- Access Manager Application Development Kit, a set of C and Java APIs that are used for coding applications to use the Access Manager authorization service
- Access Manager Runtime for Java

In addition to the base components, we install WebSEAL, which is found on the IBM Tivoli Access Manager Web Security CD or CD image (amWebsec.iso). The components we install are:

- WebSEAL server (Webseald)
- IBM Network Authentication Service Toolkit, because TAMCO is using desktop single sign-on to Linux machines

As with the base CD, there are other components on the Web Security CD that apply to usage and scenarios not planned for the TAMCO environment.

5.2.1 Base component installation

Prior to installing the Access Manager Runtime and policy server, and a first step to any Access Manager V6.0 installation, it is necessary to install the Access Manager license and the Access Manager Web Security utilities. The license installation is checked when you start the Access Manager Runtime and policy server installations, which will terminate if the correct license file is not found. The runtime installation also requires the presence of the Web Security utilities, which provide the XML parsing and client utilities used by Access Manager. An overview of these dependencies is depicted in Figure 5-28 on page 111.

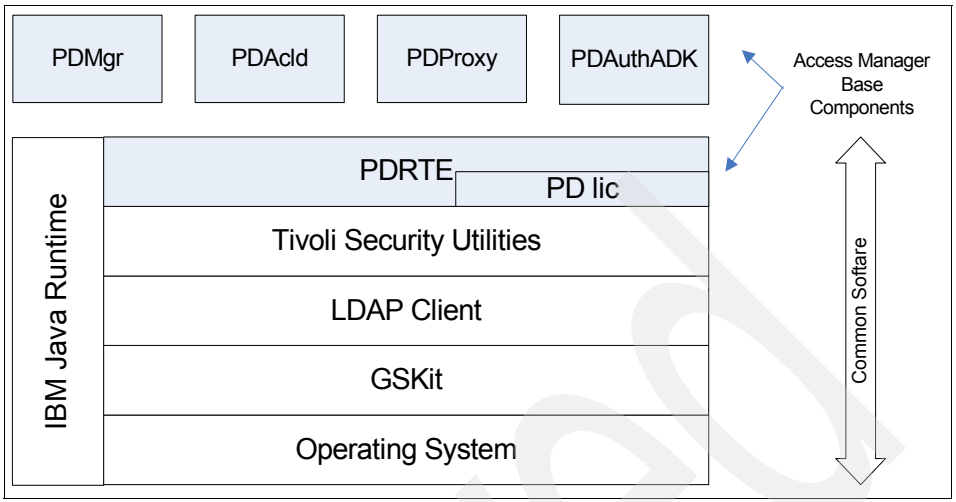

Figure 5-28 Access Manager component dependencies

Access Manager License (PDlic)

If you use the installation wizard to install the Access Manager Runtime, the Access Manager license component is installed automatically. If you use the native installation method, you are required to install this package manually.

Regardless of the installation method used, a directory is created in the Access Manager installation home directory that is populated with the correct license file and a script that displays the license information.

To install the license component on Linux, run the following **rpm** command:

```
rpm -ihv PDlic-PD-6.0.0-0.<platform version>.rpm
```

To install on Windows, run the **install** command on the CD in the /windows/Policy Director/Disk Images/Disk1/PDlic/ directory.

Access Manager Web Security utilities (TivSecUtl)

When you install the TivSecUtl package, the Axis XML parsing engine is installed to handle SOAP messaging and parse XML documents used by Access Manager.

Failure to install the Web Security utilities causes subsequent base Access Manager component installations to fail, so it must be installed on each platform hosting the Access Manager base components.

To install on Linux, run the package installer on the TivSecUtil package under the linux_<platform> directory on the CD or CD image:

```
# rpm -ihv TivSecUtl-TivSec-6.0.0-0.<platform>.rpm
```

To install on Windows, run the setup.exe on the ambaseWIN CD in the /windows/TivSecUtl/Disk Images/Disk1/ directory.

Access Manager Runtime

The Access Manager Runtime must be installed on every machine that hosts an Access Manager server. This includes WebSEAL.

To install the Access Manager Runtime on Linux, run the following command:

```
# rpm -ihv PDRTE-PD-6.0.0-0.i386.rpm
PDrte-PD
##################################################
```

To install on Windows, run the **PDRte** command in the /windows/Policy Director/Disk Images/Disk1/ directory.

Access Manager Policy Server

A single TAMCO Policy Server needs to be installed in the Finland region. The Policy Server will also have a warm standby leveraging linux HA clustering software (instructions for configuring the HA software are not provided here). The role of the Policy Server is crucial and requires a stable and reliable platform.

The host name of the policy server is tamco-ps.fi.tamco.com.

Run the following command to install the Access Manager Policy Server package:

```
# rpm -ihv PDMGR-PD-6.0.0-0.i386.rpm
PDMGR-PD
##################################################
```

Access Manager Policy Proxy Server

Because there can only be one Policy Server per security policy, the other two regions will have Policy Proxy Servers to offload requests from the Policy Server.

The host names of the two Policy Proxy Servers are:

- tamco-psde.de.tamco.com
- tamco-psuk.uk.tamco.com

Run the following command to install the Access Manager Policy Proxy package:

```
# rpm -ihv PDMgrPrxy-PD-6.0.0-0.i386.rpm
PDMGRPRXY-PD
################################################
```

Access Manager Authorization Server

The Access Manager Authorization Server is also installed from the Access Manager base CD. Like the policy server, the Authorization Server is crucial to the correct application of the access control policy because it provides the master authorization database from which the locally installed replicas get updates. It will be installed on a Linux platform.

Use the following command to install the Access Manager Authorization Server package:

```
# rpm -ihv PDAcld-PD-6.0.0-0.i386.rpm
PDAcld-PD
################################################
```

5.2.2 Installing WebSEAL

WebSEAL has to be installed from a separate CD image for Access Manager Web Security. To install WebSEAL, it is necessary first to verify that the prerequisite components indicated in Figure 5-29 are present.

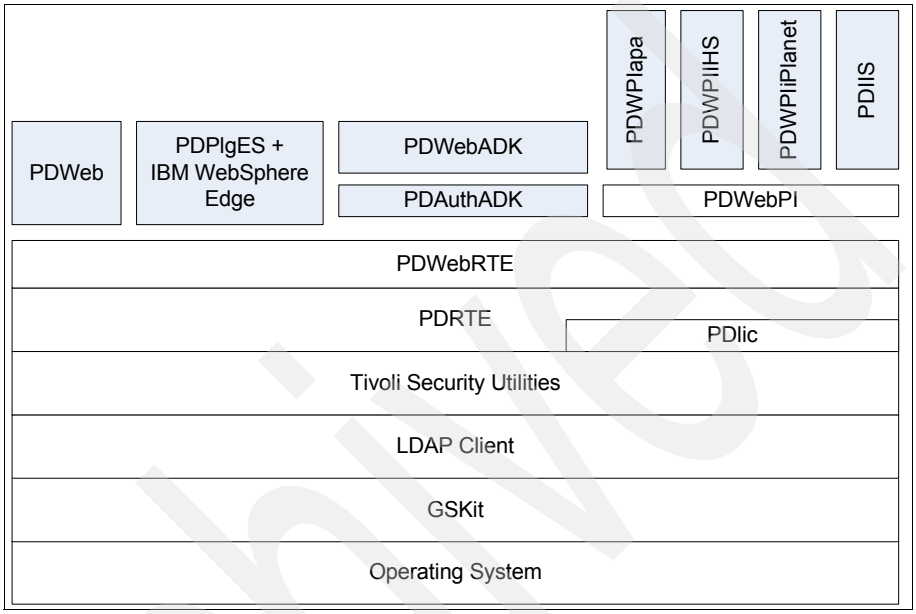

Figure 5-29 Access Manager Web Security components

Because we have just gone through an installation of the prerequisite software for the base Access Manager servers, we know that GSKit, an LDAP client, the PDLic, TivSecUtl, and PDRTE components are installed. The only remaining prerequisite software to install is the runtime component for WebSEAL, PDWebRTE, which contains shared authentication library files used for Web Security systems.

All other Web Security components shown in Figure 5-29 run on top of PDWebRTE.

Run the setup.exe to install the PDWebRTE on the Windows system that will host WebSEAL from the \windows\Policy Director\Disk Images\Disk 1\PDWebRTE\Disk Images\Disk 1\ directory.

The next step is to install the WebSEAL component PDWeb. Run the setup.exe on the Windows system hosting WebSEAL from the \windows\Policy Director\Disk Images\Disk 1\PDWeb\Disk Images\Disk 1\ directory.

This launches the installation wizard for WebSEAL. The familiar windows are displayed to choose the language, accept the license, and check the environment summary information collected by the wizard. After these steps have been completed, the window shown in Figure 5-30 is displayed. Accept the default.

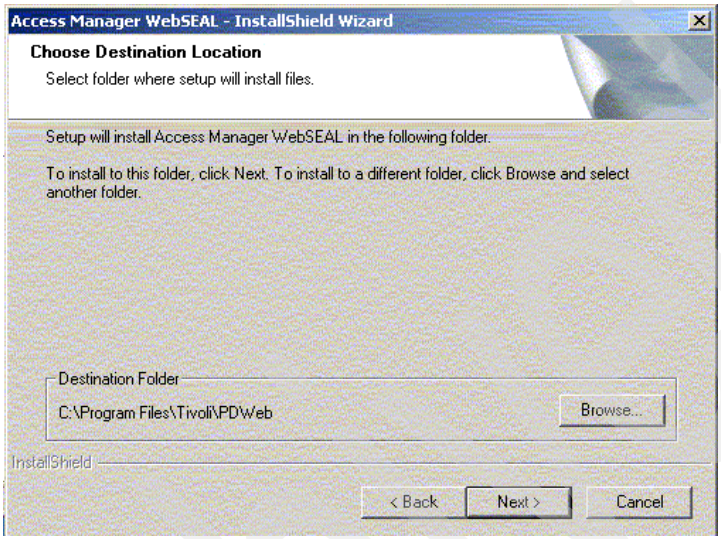

Figure 5-30 WebSEAL installer destination directory

Installation only takes a few moments. Click **Finish** in the final installation success dialog.

Verify that Access Manager WebSEAL has been added to the Add/Remove Programs list of currently installed programs.

5.2.3 Installing the Common Auditing and Reporting Service

The Common Auditing and Reporting Service is installed as a separate installation off the Access Manager V6.0 base CD. We install it on a Windows platform because TAMCO uses Crystal Reports as their standard audit report generator, and the Common Auditing and Reporting Services provide a Crystal Reports capability.

The first step in the installation is to make sure the prerequisite software that is used specifically by the Common Auditing and Reporting Services is installed and running. The two main prerequisites are:

- A DB2 instance
- WebSphere Application Server

Instances of both have already been installed. With respect to the use of WebSphere by the Common Auditing and Reporting Services, it is possible to use the existing WebSphere Application Server by adding Common Auditing and Reporting Services and the associated Event Server as enterprise applications in the same manner as we did earlier for WPM and the Directory Server Web Administration console.

However, the same is not true of the DB2 instance that we have defined earlier for Access Manager (*ldapdb2*). The database operations and load carried out by the Common Auditing and Reporting Services are more intensive than those carried out by the Directory Server. Therefore, it is necessary to define a new DB2 instance specifically for the Common Auditing and Reporting Services. We call this instance *carsdb2*.

Create a user and group for the DB2 instance

Use Linux utilities to create a carsdb2 group with the following properties. Make sure root is a member of this group:

Group name carsdb2

Group ID 101

A group ID less than 500 makes this a system group, and you are informed by the system that you are doing so.

Create a carsdb2 user with the following properties:

User name carsdb2

Some systems require a first name and last name as optional properties.

User login carsdb2

Enter a password for the carsdb2 user and confirm it. You then need to edit the user's properties as follows:

User id (uid) 101

Home directory /home/carsdb2

Login shell <shell of choice>

Default group membership
carsdb2

Create an instance using db2icrt

Create the DB2 instance where the Common Auditing and Reporting Service server stores events. The port number must be specified to allow this DB2 instance to be accessed over TCP/IP:

```
# /opt/IBM/db2/V8.1/instance/db2icrt -p 37000 -u carsdb2 carsdb2
```

It is also possible to specify a port name in the -p flag and then associate this alias with a port number in the /etc/services file.

As a check, start the DB2 instance by switching the user ID to carsdb2 and running the **db2start** command. If you get the message saying `DB2START processing was successful`, then you know the instance was created and is running properly.

Install Common Auditing and Reporting Server and Client

Before starting the installation wizard, it is always a good idea to verify that the required prerequisite software is present and working:

1. Log in as root and check that the DB2 carsdb instance is started and listening on the correct port (37000 is the one we defined) by connecting to the database using the following command:

   ```
   db2 get instance
   ```

   This command should return:

   ```
   The current database manager instance is: carsdb2
   ```

2. Then run the command:

   ```
   db2 get dbm cfg | grep SVCENAME
   ```

   This command should return:

   ```
   TCP/IP Service name                  (SVCENAME) = 37000
   ```

3. WebSphere needs to be stopped in order to install the Common Auditing and Reporting Service server. Run the following command to stop WebSphere:

   ```
   /opt/WebSphere/Appserver/bin/stopServer.sh server1
   ```

4. The Common Auditing and Reporting Service server installation wizard needs the WebSphere Java to work, so you need to run the following script to set up the shell environment correctly:

   ```
   . /opt/WebSphere/Appserver/profiles/default/bin/setCmdLine.sh
   ```

5. Start the Common Auditing and Reporting Service server installation wizard by executing the **srv_setupWin_server** command from the Access Manager base CD windows/CARS/server directory.

6. Several preliminary windows are displayed asking for your choice of language, the installation target directory, and the product to install (Common Auditing and Reporting Service Event Server). Accept the defaults.
7. The next window is displayed after the installer checks the system for the prerequisites. Enter the proper installation directory, as shown in Figure 5-31.

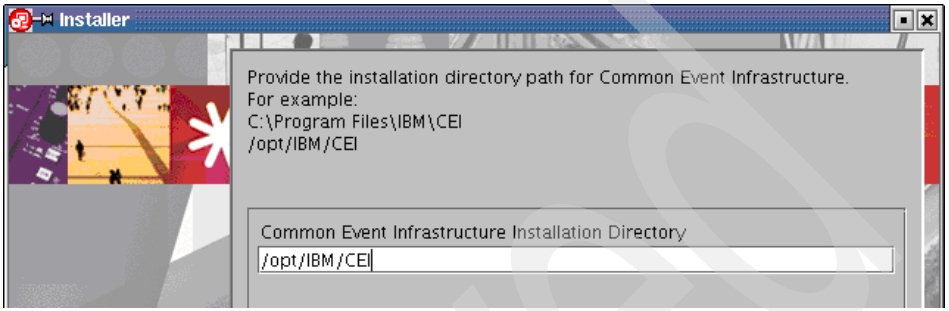

Figure 5-31 CEI installation directory path

8. The next window asks for a WebSphere administrator user ID and password (Figure 5-32 on page 119). These are required for this step even when WebSphere security is disabled. The values for these fields are required to proceed with the install. If WebSphere security is enabled, enter the actual values for your WebSphere administrator user and password. Otherwise, enter some *dummy* values (such as wasadmin and passw0rd) that will be ignored.

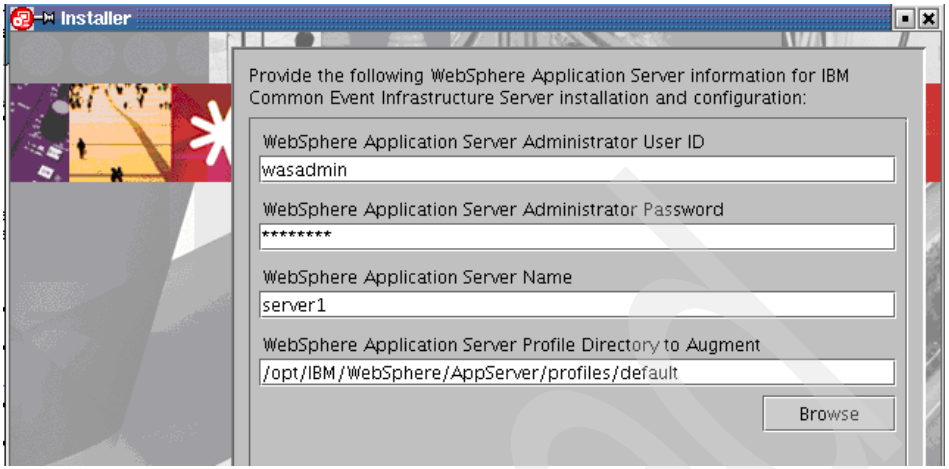

Figure 5-32 Setting the WebSphere administrator ID and password

9. Accept the default of server1 and either select **Browse** or enter by hand the path to the default WebSphere profile /opt/IBM/WebSphere/AppServer/profiles/default.
10. Click **Next** to continue to the database information window shown in Figure 5-33.

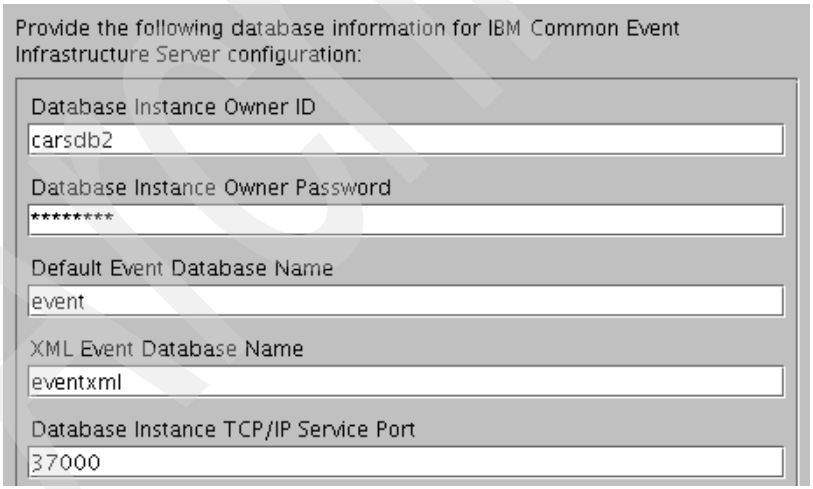

Figure 5-33 Information needed for the CEI Server configuration

11. Click **Next** and a summary information page is displayed. Click **Next** to start the installation. The installation should take a few minutes to complete. On completion, a window that says the Common Event Infrastructure has been installed is displayed. The next step is the configuration of the XML data store the CEI Server uses, as shown in Figure 5-34.

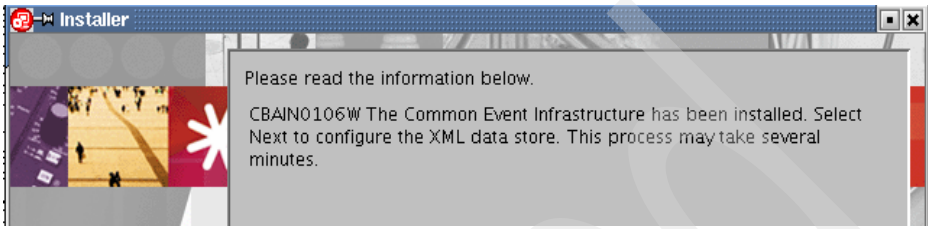

Figure 5-34 CEI installation completed and XML data store notification

When this process has finished, you see the window displayed in Figure 5-35.

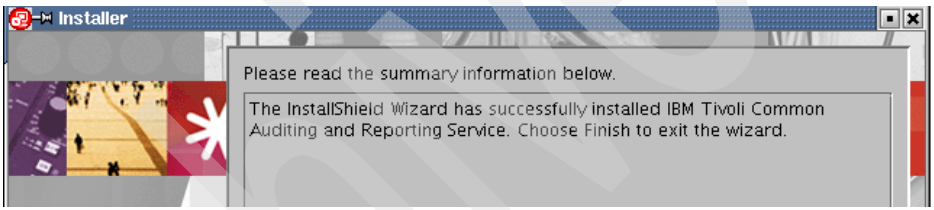

Figure 5-35 Successful installation window for CARS

12. The final step is to restart WebSphere using the following command:

```
# /opt/IBM/WebSphere/Appserver/bin/serverStart.sh server1
```

To verify the installation, you can launch the WebSphere Administration Console and select **Applications** → **Enterprise Applications**, which should show you the *CommonAuditService* and *EventServer* applications as installed and running. This is shown in Figure 5-36.

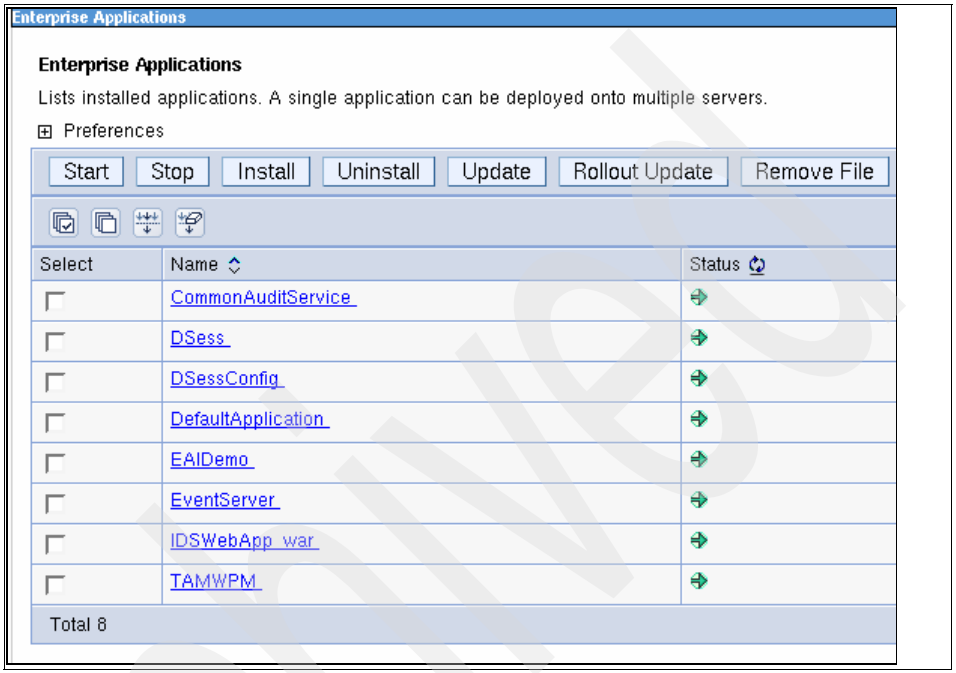

Figure 5-36 View of installed Enterprise applications in WebSphere administration

This concludes the installation of all necessary components for the TAMCO access control solution deployment. In the next chapter, we configure and integrate these components into the existing IT infrastructure.

6

Configuring IBM Tivoli Access Manager

In this chapter, we cover the configuration of the base Access Manager components and WebSEAL to conform to the TAMCO access control security policy. We have already configured the Access Manager prerequisite software in 5.1, “Installing and configuring prerequisites” on page 77. Now we address the tasks of configuring Access Manager base components: WebSEAL, the Common Auditing and Reporting Service, and Web Portal Manager. Once these components have been configured, we set up the object namespace, add Access Manager users, define the TAMCO ACL policies, and create the appropriate TAMCO groups. Access Manager is then integrated into the application environment with PeopleSoft, Siebel, and WebSphere Portal. This chapter concludes with a description of the deployment of Access Manager with the load-balancing hardware to satisfy TAMCO’s high availability requirements.

6.1 Configuring the Access Manager base components

Having installed all the prerequisites and the base Access Manager servers, WebSEAL, and Web Portal Manager, we now cover how to configure these components to enforce the TAMCO access control security policy.

6.1.1 Configuring and populating the Access Manager user registry

When we installed the Directory Server in 5.1.4, “Setup of IBM Tivoli Directory Server” on page 82 using the install_ldap_server wizard, we performed several tasks to configure the DB2 instance we are using as the datastore for the Directory Server (the DB2 administrator ID, the DB2 database name, the home directory of the DB2 database, and the encryption seed). We also configured the Directory Server administrator ID (cn=root) and password, a user-defined suffix (o=tamco, c=fi), a local host name (tamco-ldap1m.fi.tamco.com), and SSL parameters.

Configuring the user registry requires three additional updates to the directory configuration:

1. Install a schema definition specific to Access Manager.
2. Install a suffix in the directory that corresponds to the Access Manager container that holds Access Manager’s metadata.
3. Create the TAMCO domain-specific suffixes for Finland, Germany, and the UK domains under which the user and group data for each domain is stored.

The first task was done automatically when we installed Directory Server. To install the Access Manager suffix, launch the Directory Server Web Administration console in a browser using the following URL (make sure WebSphere Application Server is running):

```
http://localhost:12100/IDSWebApp/IDSjsp/Login.jsp
```

After a successful login, you are presented with the Directory Server Web Administration Tool welcome window shown in Figure 6-1.

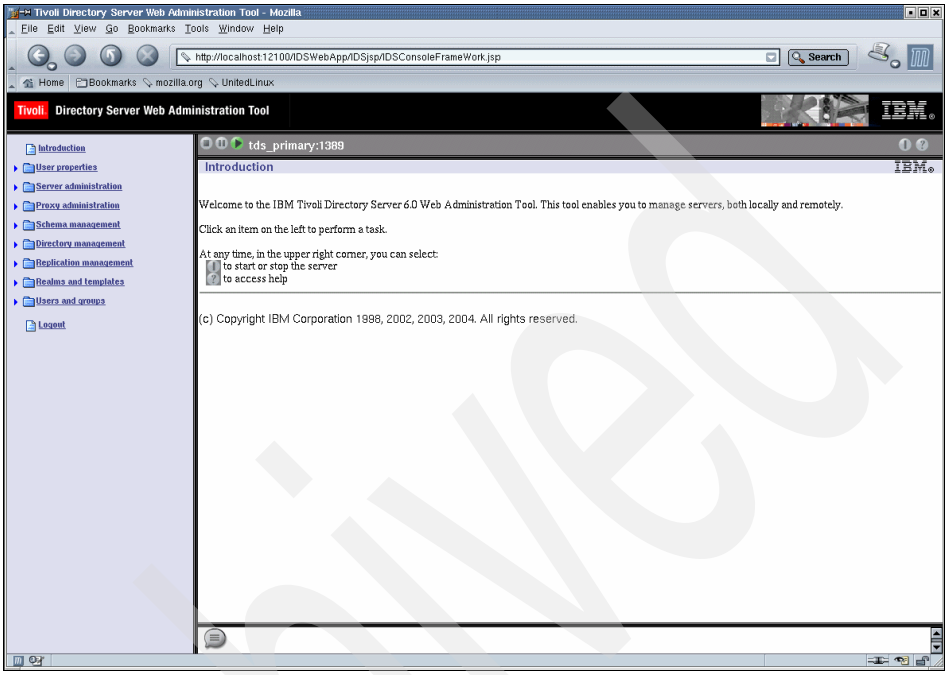

Figure 6-1 Directory Server Web Administration Tool welcome window

To create the Access Manager suffix, select **Server Administration** → **Manage server properties** → **Suffixes**. The window depicted in Figure 6-2 appears.

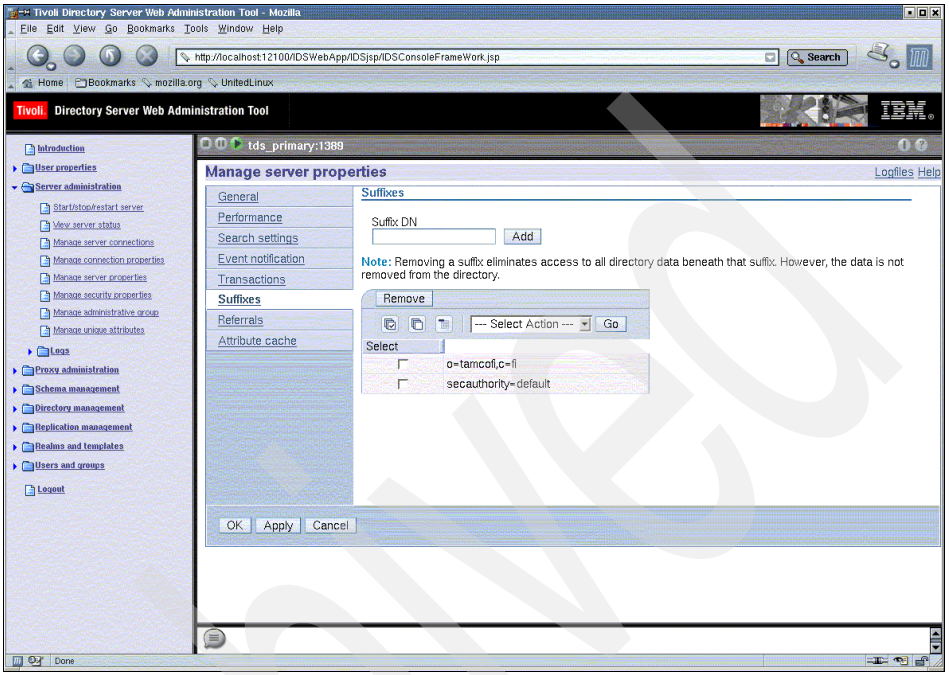

Figure 6-2 Directory Server Web administration console view of suffix creation

In the Suffixes window, add the following Distinguished Name (DN), which is a required value unique to Access Manager that cannot be changed (not case sensitive):

```
secauthority=default
```

We also add three TAMCO-specific suffixes for the containers in the directory that will hold TAMCO domain-specific user and group data:

```
o=tamcofi,c=fi
o=tamcode,c=de
o=tamcouk,c=uk
```

The final step is to populate the user registry with users. At the beginning of the project, in 4.2, “The TAMCO deployment” on page 45, we created an LDIF file of all the TAMCO employees. We now use this LDIF file to load the TAMCO users into the Access Manager user registry. We perform this load using the LDAP import utility *ldif2db*.

We have to look out for an additional security consideration, however, regarding users. The PeopleSoft Human Resources Management System stores its user data in a DB2 database, which is a separate instance from the Directory Server's DB2 instance. Over time, changes are made by the Human Resources staff to the user data in the HRMS system, and there is a danger that the user information in HRMS may get out of synchronization with the user data stored in the Access Manager user registry. Not all the attributes of the users' data stored in the HRMS database or in Directory Server need to be synchronized, because all but one of the attributes stored in the HRMS database are not relevant to security. The one security attribute we need to keep synchronized is the user passwords, because we are deploying an enterprise single sign-on solution. We will use the Directory Integrator tool bundled with Access Manager to keep these passwords synchronized (see "Synchronizing passwords" on page 128).

Bulkload

Now we proceed with our initial user population:

1. Issue the following command to load the employee data into the directory:

   ```
   bulkload -I idsldap -i /home/ibmuser/labdata/TAMCO_employees.ldif
   ```

   Where -I is the Directory Server instance name and -i is the name of the input file.

2. After a couple of minutes, the bulkload process should complete with the output shown in Figure 6-3.

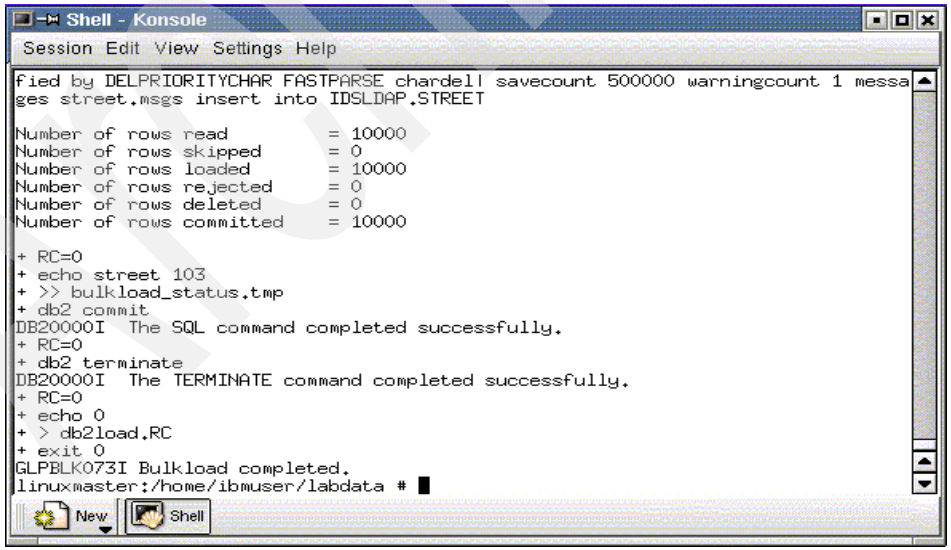

Figure 6-3 Output from bulkload command

Synchronizing passwords

We have a choice of mechanisms in order to synchronize password data. One is provided by Access Manager using IBM Tivoli Directory Integrator. The other is used for synchronizing passwords in an environment on which Access Manager and IBM Tivoli Identity Manager are integrated.

In this case, we are interested in synchronizing changes in passwords that are made through the PeopleSoft HRMS administration interface. To do this task, we use a Directory Integrator AssemblyLine that detects password changes in the HRMS DB2 database (through the DB2 Change Table) and pushes that change to the Access Manager user registry.

The Tivoli Directory Integrator password synchronization AssemblyLine does not come bundled with the Access Manager CDs, but Tivoli Directory Integrator is bundled with Access Manager. In order to run the password synchronization AssemblyLine, it is necessary to install Directory Integrator, and then add the password synchronization AssemblyLine, which can be downloaded from the Tivoli Directory Integrator internal support Web site:

`http://Web.cs.opensource.ibm.com/www/itdi/carrotpatch/pwsynch`

This solution uses one password interceptor synchronizing between two targets, the PeopleSoft HRMS DB2 database and the Access Manager Directory (user registry), which is the authoritative password store. This allows changes to be made in the PeopleSoft HRMS database record for user passwords that are automatically updated in the Access Manager user registry (Directory Server).

6.1.2 Configuring Access Manager components

We use the Access Manager command-line configuration utility pdconfig to configure the Access Manager run time, Policy Server, Policy Proxy Server, WebSEAL, and WPM components installed on Linux. We use the configuration console for the WebSEAL configuration since WebSEAL is hosted on Windows.

Access Manager runtime configuration

Since we are installing the Access Manager run time on both Linux and Windows platforms, instructions for both are provided.

- Linux

  Open a terminal session and run **pdconfig**. The choices in Figure 6-4 are displayed in an interactive session.

- Windows

  Select **Start** → **Programs** → **IBM Tivoli Access Manager** → **Configuration**.

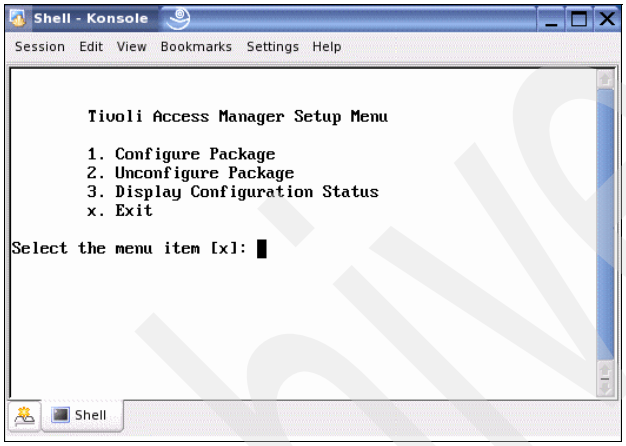

Figure 6-4 pdcopnfig main selection window

To configure:

1. Type 1 (Configure Package).
2. Next, select **Access Manager Runtime Configuration**.

   The following interactive response questions, shown in Figure 6-5 on page 130, are displayed in series. We are using LDAP as the Access Manager user registry, and specifying the host name of the Directory Server master in the Finland domain (tamco-ldap1m.fi.tamco.com). If we were using another directory, such as Active Directory, we would see two choices instead of only LDAP.

On successful completion of the Access Manager runtime configuration, you see a confirmation at the bottom of the dialogue.

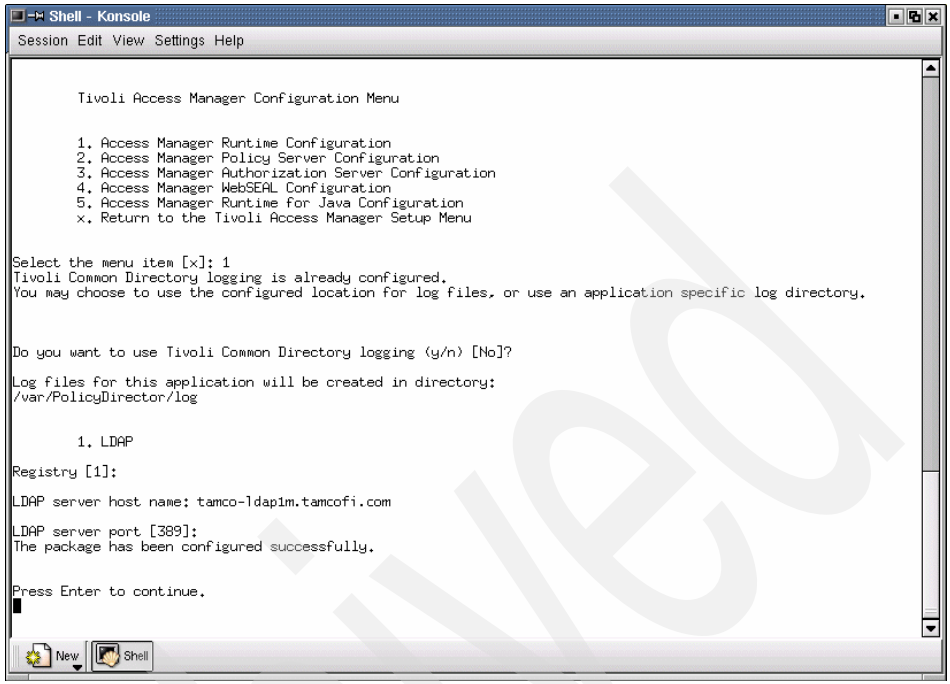

Figure 6-5 Access Manager runtime configuration completion

Access Manager Policy Server configuration

The Policy Server is installed on Linux. Therefore, we continue to use the pdconfig utility that is already running:

1. Type [x] (Return to Access Manager configuration).
2. Type 1 (Configure Package).
3. Select **Access Manager Policy Server Configuration**.
4. Answer the interactive questions shown in Figure 6-6 on page 131.

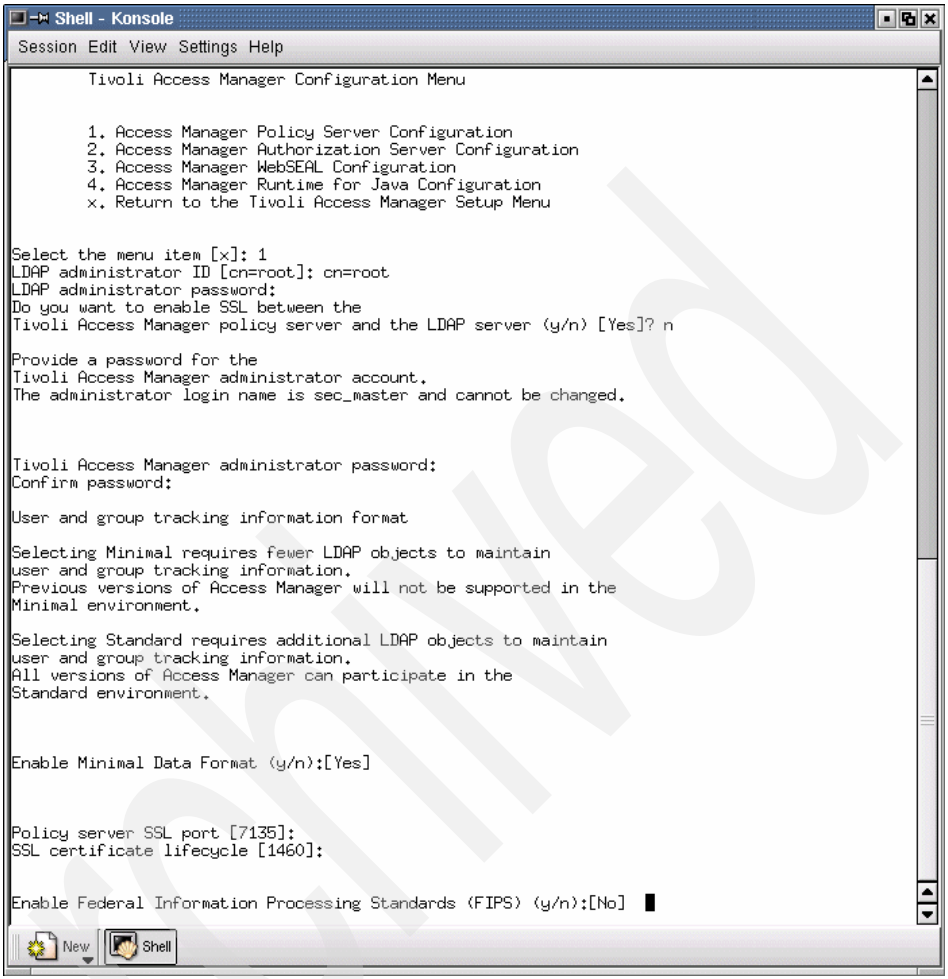

Figure 6-6 Access Manager Policy Server configuration completion

5. Figure 6-7 on page 132 shows the inclusion of the information about creating and storing the SSL certificate. When the Policy Server is configured, a default self-signed root certificate is created to be used for signing the SSL communication between the Access Manager servers. This default certificate is not appropriate for a production environment because it is self-signed; it should only be used for testing. Consequently, for the TAMCO environment we went through the registration and request process with a third-party certificate authority to receive a root certificate that we use with the GSKit utility to generate our keystore and derived server and client certificates.

Note the response required for the Enable Federal Information Processing Standards [FIPS]. We *enable* this option. While somewhat obscurely named, what this selection is doing is configuring the Access Manager Policy Server to use Transport Layer Security Version 1 (TLS) instead of SSL Version 3. TLS provides a somewhat stronger protection because the TLS specification has fixed some of the security weaknesses in the original SSL specification.

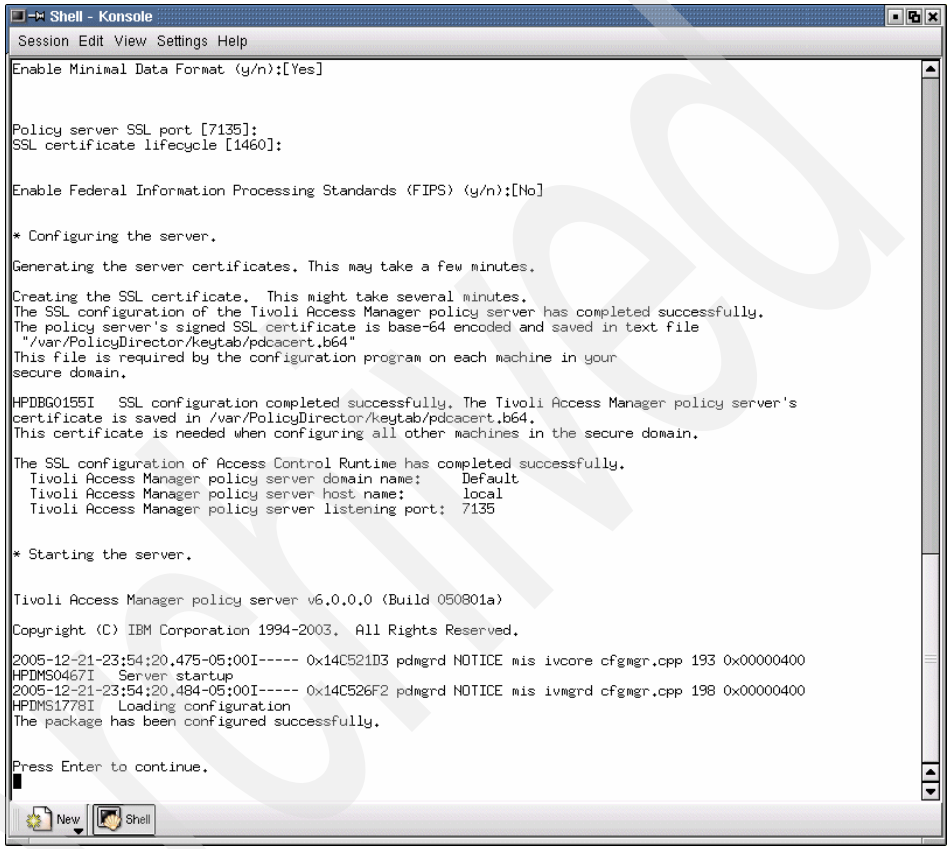

Figure 6-7 Access Manager Policy Server configuration

Access Manager Policy Proxy Server configuration

The Policy Proxy Server is installed on Linux. The policy proxy also is configured with the pdconfig utility:

1. Type [x] (Return to Access Manager configuration).
2. Type 1 (Configure Package).
3. Select **Access Manager Policy Proxy Server Configuration**.
4. Answer the interactive questions shown in Figure 6-8 on page 133.

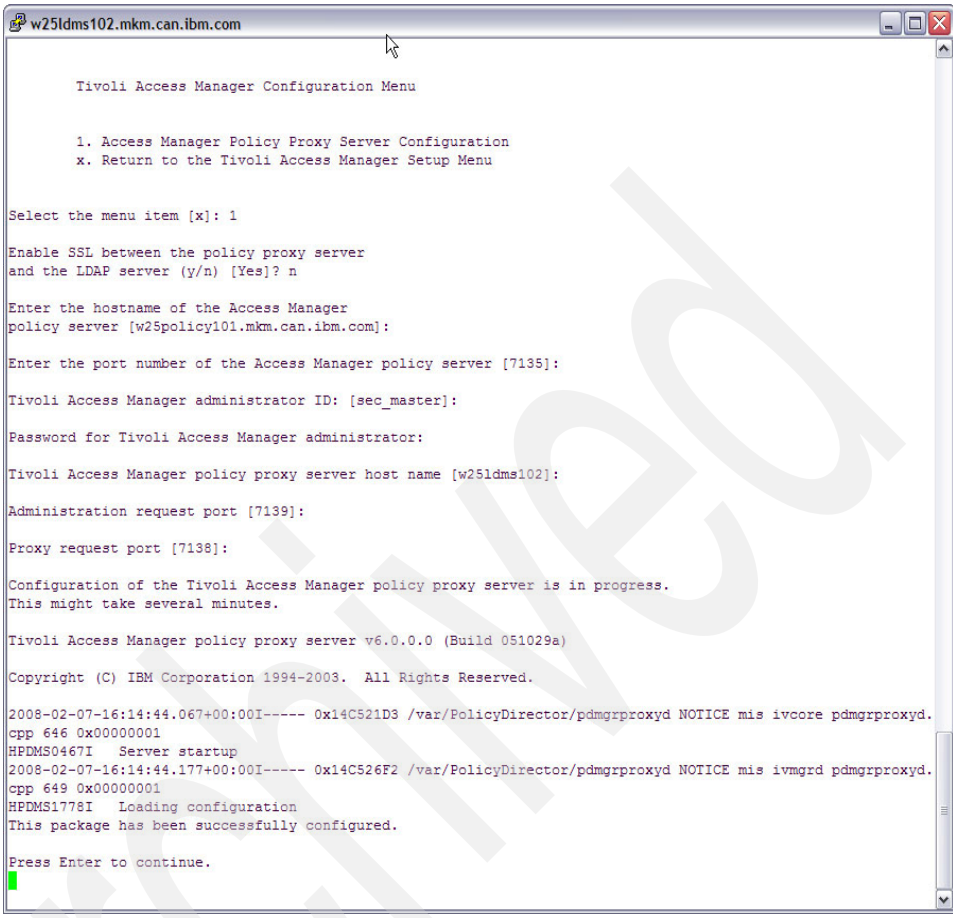

Figure 6-8 Access Manager Policy Proxy Server configuration completion

The Policy Proxy Server is now up and ready for other Access Manager for e-business servers to connect. In TAMCO, the WebSEAL servers will proxy all policy database updates through this server and thus requires additional configuration on the WebSEAL servers after installation.

Access Manager Authorization Server configuration

The Authorization Server is also installed on Linux. Therefore, we continue to use the pdconfig utility that is already running:

1. Type [x] (Return to Access Manager configuration).
2. Type 1 (Configure Package).
3. Select **Access Manager Authorization Server**.

4. Answer the interactive questions shown in Figure 6-9.

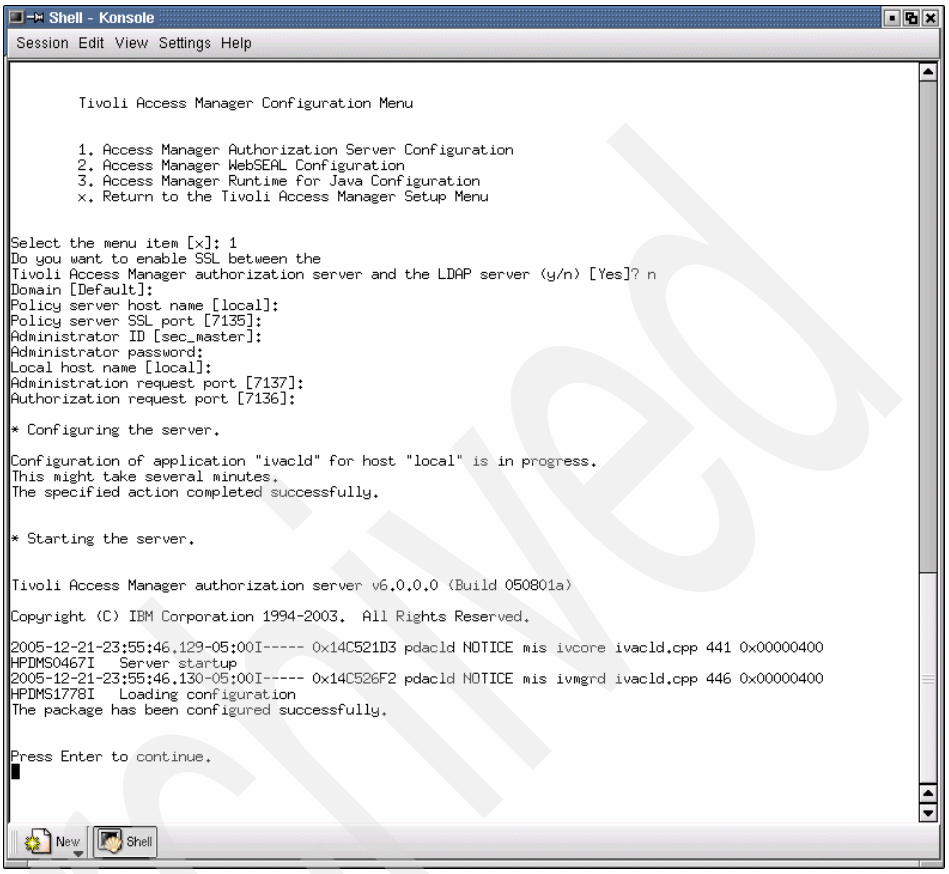

Figure 6-9 Access Manager Authorization Server configuration completion

WebSEAL configuration

We now move to the Windows machine that hosts the WebSEAL server (tamco-ws1.fi.tamco.com):

1. Select **Start** → **Programs** → **IBM Tivoli Access Manager** → **Configuration**. The Access Manager graphical configuration utility opens, as shown in Figure 6-10 on page 135. Note that it is also possible to enter the **`pdconfig`** command at the command prompt to launch this utility.

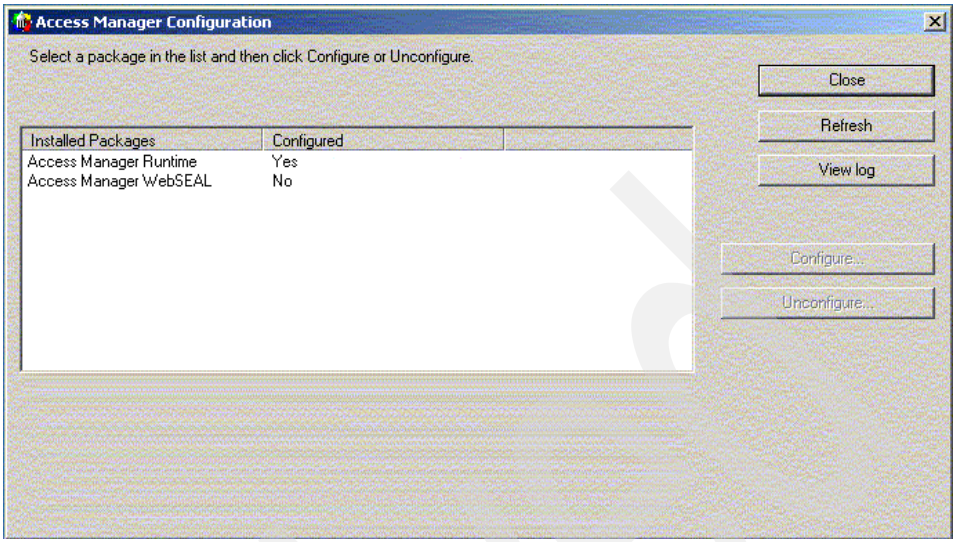

Figure 6-10 Access Manager WebSEAL configuration on Windows

We configure the master WebSEAL server for the TAMCO Finland domain. However, these instructions can be duplicated for the WebSEAL servers in the other two domains, with the proviso to use the correct domain names for the Germany and UK domains (tamcode and tamcouk, respectively).

2. If the Access Manager Runtime is not configured, it must be configured first. We already configured it earlier, so select the **Access Manager WebSEAL** component and click **Configure**.
3. The next several windows appear asking you to provide various data that defines the WebSEAL server to the Access Manager environment. Use the following data:
   - WebSEAL instance name: tamco-ws1 (Do not select logical network interface.)
   - WebSEAL server host name and listening port: tamco-ws1.fi.tamco.com 7234
   - Access Manager user registry: LDAP
   - Adminstrator ID and Password: sec_master/passw0rd
   - WebSEAL HTTP and HTTPS access ports: 80 and 443 (TAMCO has decided to use these well-known ports to minimize administration associated with opening additional http and https ports.)
   - Location and listening port of the Policy Server: tamco-ps.fi.tamco.com and 7135

4. A configuration progress window opens. Configuration should take a few minutes because the utility also starts the WebSEAL server once it is successfully configured, as shown in Figure 6-11.

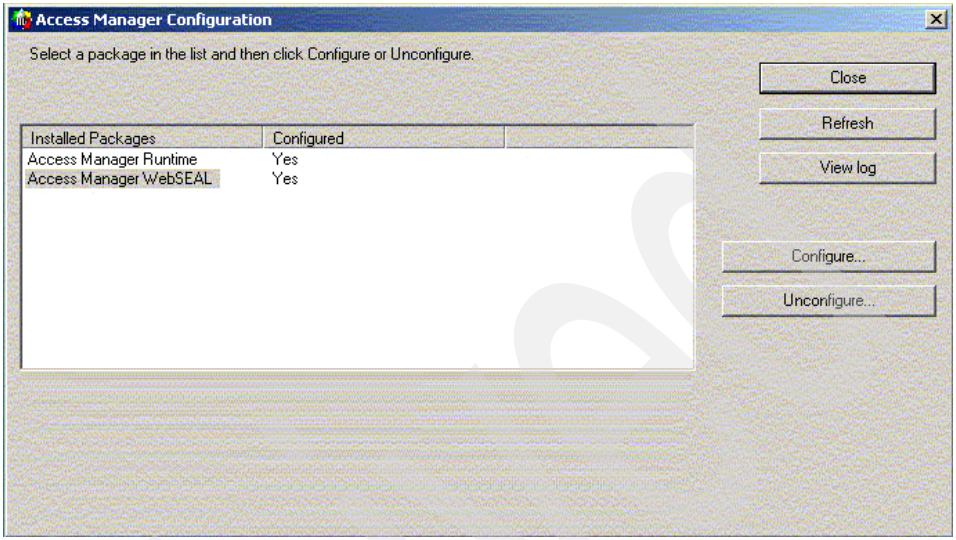

Figure 6-11 Successful WebSEAL configuration

5. For the WebSEAL servers configured outside of Finland, the WebSEAL configuration file will need to be updated to ensure that Policy Server requests are offloaded to the Policy Proxy server. There are two lines in the Webseald-default.conf file that will need to be updated to reflect the Policy Proxy Server host name and listening port (default 7138) as follows:

```
[manager]
master-host = tamco-ppde.de.tamco.com
master-port = 7138
```

WPM configuration

Now do the following:

1. Verify that the Directory Server, Access Manager Policy Server, and Authorization Server are running by entering the following commands:

```
local:~ # pdadmin -a sec_master -p passw0rd
pdadmin sec_master>
local:~ # pd_start status
Tivoli Access Manager servers
Server Enabled Running
------------------------------------------
pdmgrd                  yes       yes
pdacld                  yes       yes
pdmgrproxyd             no        no
local:~ #
```

2. Verify the status of WebSphere by running the following command on the WebSphere Linux machine:

   ```
   #/opt/IBM/WebSphere/AppServer/bin/serverStatus.sh server1
   ```

3. Check that the application server is active by launching the WebSphere Administration console in the browser using the following URL and then navigating to **Servers** → **Application Servers**:

   ```
   http://local.demo.com:9060/admin
   ```

The WPM configuration prompts for the local node used to deploy the WPM application to WebSphere. Note that, in Figure 6-12, this is *localNode01*.

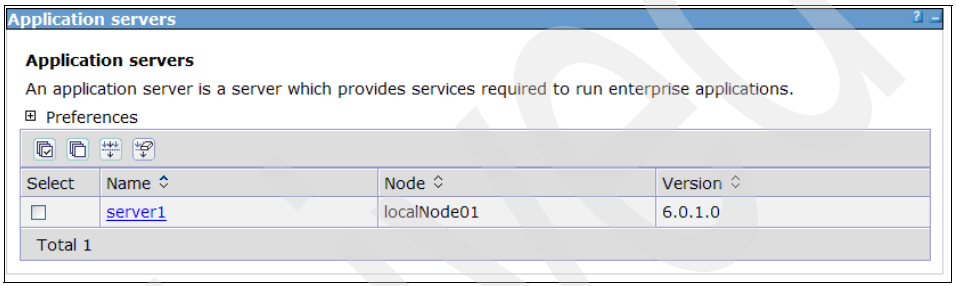

Figure 6-12 WebSphere Administration view of defined application servers

WebSphere Application Server V6.0 ships with its own level of the Access Manager runtime for Java in its JRE. This enables easier deployment of WebSphere-Access Manager integration components such as:

- Trust Association Interceptor
- Access Manager JACC provider
- Access Manager's GSO credential mapping

Web Portal Manager V6.0, however, requires the Access Manager V6.0 Runtime for Java. Because the Access Manager V6.0 Runtime for Java is backward compatible, the WebSphere Access Manager integration components listed above still work with the Access Manager V6.0 Runtime for Java.

1. *Source* the WebSphere environment variables using the setupCmdLine.sh utility in the *default* WebSphere profile:

   ```
   # .
   /opt/IBM/WebSphere/AppServer/profiles/default/bin/setupCmdLine.sh
   ```

   Note: To *source* the script, you must use the leading period (.) in the command above.

2. In the same command window, start the Access Manager pdconfig utility and then enter 1 for Configure Package.
3. Next enter 2 for Access Manager Runtime for Java configuration.
4. Reply to the interactive prompts as indicated below. <enter> means to simply hit the Enter key to accept the default shown in the prompt:

```
Specify the full path of the Java Runtime Environment (JRE) to
configure for Tivoli Access
Manager[/opt/IBM/WebSphere/AppServer/java/bin/../jre]:<enter>
Enter 'full' or 'standalone' for the configuration type[full]:
<enter>

Enter the hostname of the Access Manager policy server machine
[local]: local.demo.com

Enter the port number of the Access Manager policy server machine
[7135]: <enter>

Enter Access Manager Policy Server domain information [Default]:
<enter>

Tivoli Common Directory logging is currently configured.
You may enable this application to use Tivoli Common Directory
logging
using the currently configured directory for log files.

Do you want to use Tivoli Common Directory logging (y/n) [n]? y

Log files for this application will be created in directory:
/var/ibm/tivoli/common

Configuration of Access Manager Java Runtime Environment is in
progress.
This might take several minutes.
Configuration of Access Manager Java Runtime Environment completed
successfully.

Press Enter to continue. <enter>
```

5. While still in the pdconfig environment used in the previous section, select [x] to return to the Access Manager configuration, then enter 1 to select Configuration, and then enter 1 to select Access Manager Web Portal Manager Configuration.

6. Next, reply to the interactive prompts with the values indicated below:

```
Enter the IBM WebSphere Application Server or Deployment Manager
installation full path [/opt/IBM/WebSphere/AppServer]: <enter>

Enter the hostname of the Access Manager
policy server [local]: tamco.com

Enter the port number of the Access Manager policy server
[7135]:<enter>

Does the Access Manager domain contain an Authorization Server (y/n)
[y]? <enter>

Enter the hostname of the Access Manager authorization server
[tamco.com]: <enter>

Enter the port number of the Access Manager authorization server
[7136]: <enter>

Tivoli Access Manager administrator ID: [sec_master]: <enter>

Password for Tivoli Access Manager administrator: passw0rd

Enter the hostname of the IBM WebSphere Application Server
 or Deployment Manager[local]: tamco.com

Enter the SOAP Admin port number of the WebSphere
 Application Server or Deployment Manager [8880]: <enter>

Is WebSphere security enabled (y/n) [n]?

Enter the IBM WebSphere application server or cluster for
 deployment
[WebSphere:cell=localNode01Cell,node=localNode01,server=server1]:
<enter>

Enter the Web server for deployment [Webserver1]: <enter>

Configuration of Access Manager Web Portal Manager is in progress.
This might take several minutes.
Press ENTER to continue. <enter>
```

Note that the WPM configuration requires the use of an Authorization Server *only* when WPM uses authMethod=TAI as a means of single sign-on to WebSphere. When WebSphere security is enabled and WebSphere authentication is configured to use a Trust Association Interceptor (TAI), then authMethod=TAI allows WPM to silently acquire the user ID from the WebSphere security context. An Access Manager Authorization Server, however, is required for this support.

7. Confirm that WPM is active by entering the following URL in the browser:

   ```
   http://local.demo.com:9080/pdadmin
   ```

8. To enable use of WPM through the HTTP Sever, open the WebSphere Administration Console, select **Servers** → **Web Servers**, select **Webserver1**, and click **Generate Plug-in**, as shown in Figure 6-13. Then restart IBM HTTP Server with this command:

   ```
   /opt/IBMIHS/bin/apachectl restart
   ```

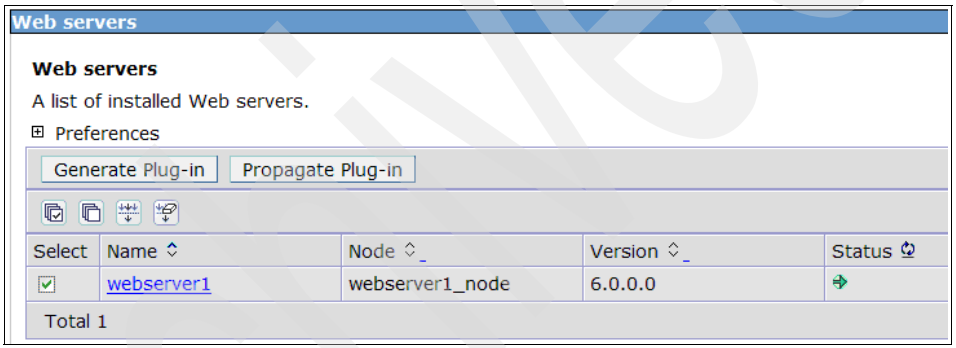

Figure 6-13 WebSphere administration console page for generating Web server plug-in

6.1.3 Common Auditing and Reporting Service configuration

Having installed the base Common Auditing and Reporting Service (CARS) components in “Install Common Auditing and Reporting Server and Client” on page 117, we now configure the operational reports we want to produce using TAMCO’s default reporting software, Crystal Enterprise 9.

Configure the event group in WebSphere Application Server

The Common Event Infrastructure (CEI) configuration is all done at the cell level. You will not see the entries created by the installation of the CARS server if you are working at the node level (which is the default).

1. Set the scope to Cell.

   a. In the WebSphere Administration Console, select **Resources** → **Common Event Infrastructure Provider**, as shown in Figure 6-14.

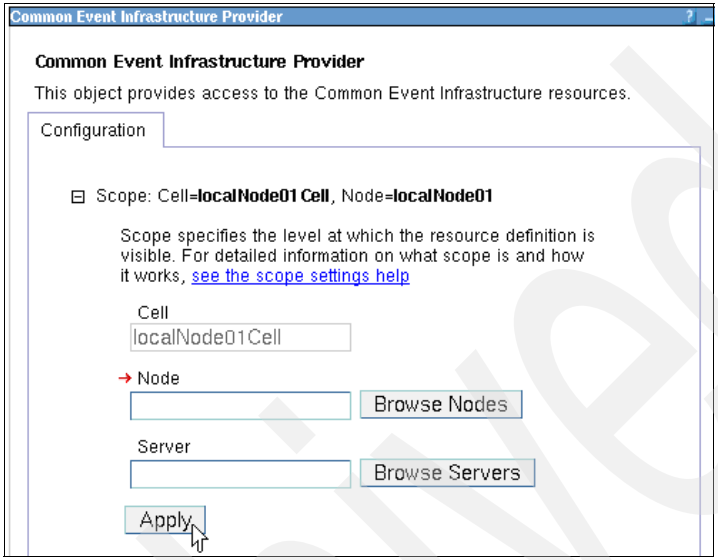

Figure 6-14 WebSphere Administration Console view of scope configuration

   b. Clear the Node field (as shown in Figure 6-14) and click **Apply**.

2. Configure the Common Auditing and Reporting Service Event Group.

   We need to define and configure a specific Event Group profile that is used to direct Common Auditing and Reporting Service events to a separate data store.

a. Select **Resources** → **Common Event Infrastructure Provider** → **Event Group Profile List** on the current page to display the table shown in Figure 6-15.

Figure 6-15 Common Event Infrastructure Event Group profile list

b. Click the **Event groups list** entry (as shown in Figure 6-15) to display the properties of the entry.

c. Select the **Event Group Profiles** option to display the Event Groups defined to CEI. Initially there is only one catch-all entry, *All Events*. Click **New** to create a new Event Group profile. Figure 6-16 on page 143 is displayed.

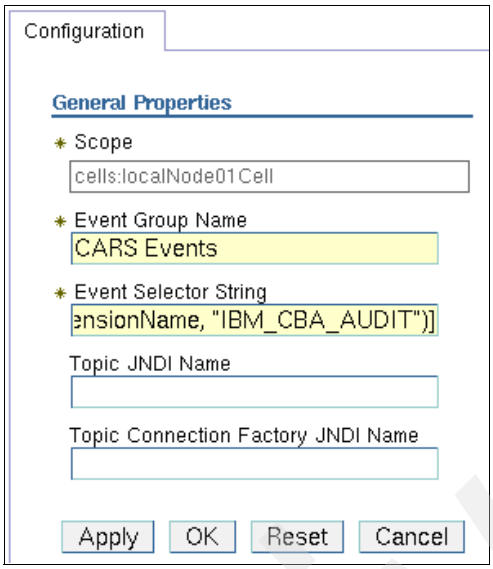

Figure 6-16 Create new Event Group profile

d. Set the event group name to be CARS Events and set the event selector string to be:

   CommonBaseEvent[starts-with(@extensionName, “IBM_CBA_AUDIT”)

e. Click **OK**. This takes you back to the table of event profiles.

f. Click the **CARS Events** entry that you just created and then click the **Custom properties** link. This displays an empty table.

g. Click **New** to create a new custom property. Complete the name (auditable) and value (true), as shown in Figure 6-17, and select type **java.lang.Boolean**.

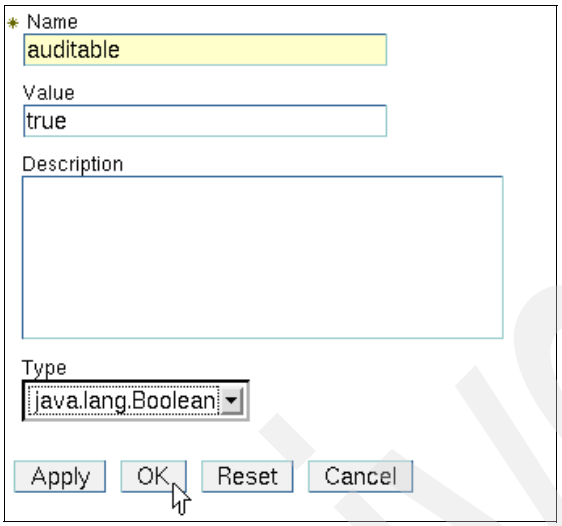

Figure 6-17 Event Group profile custom properties

h. Click **OK** to register the new custom property.

i. Click **New** again and create a second custom property. This time the name is default and the value is false. The type is java.lang.Boolean.

After you have done these steps, the table should look like Figure 6-18.

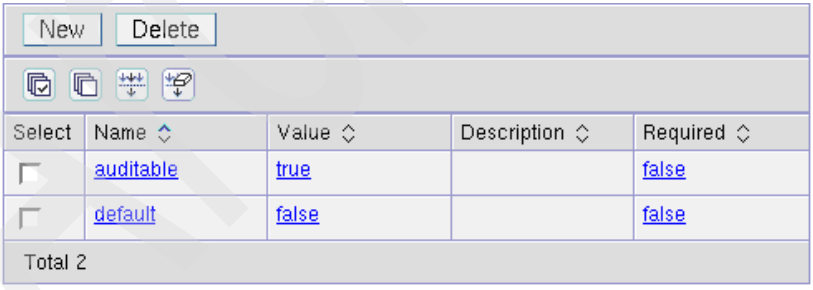

Figure 6-18 Completed CEI Event Group profile list

j. Click the **Save** link at the top of the page and then click the **Save** button to save the changes you have made so far to the WebSphere Application Server configuration.

3. Prevent Common Auditing and Reporting Service events from going to the All Events store.

   We now need to configure the All Events store so that Common Auditing and Reporting Service events are not sent to it. If we do not do this, then Common Auditing and Reporting Service events are stored in two places, which does not work with the staging of the data.

   a. Select **Resources** → **Common Event Infrastructure Provider** → **Event Group Profile List** → **Event groups list** → **Event Group Profile**.

   b. Click the **All events** profile and modify the Event Selector string to:

      CommonBaseEvent[@globalInstanceId and not(starts-with(“extensionName, “IBM_CBA_AUDIT”))]

   c. Then click **OK**. The table of profiles should now look like Figure 6-19.

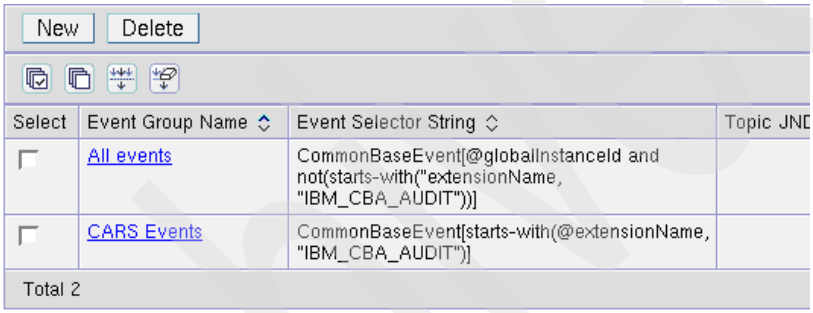

Figure 6-19 CEI Event Group profiles

   d. Click **All events** again and then select **Custom properties**. Create two new custom properties, auditable=false and default=true (reverse of last time). Both properties have the type java.lang.Boolean.

   e. Click the **Save** link at the top of the page and then click the **Save** button to save the changes you have made to the WebSphere Application Server configuration.

4. Configure CEI to compress events in the XML Event Store.

   Common Auditing and Reporting Service events are XML formatted text. These can optionally be compressed, which will save significant storage space.

   a. In the WebSphere Application Server Admin console, select **Resources** → **Common Event Infrastructure Provider** → **Data Store Profile**.

   b. Click **XML Common Event Infrastructure data store** in the table and then select **Custom Properties**.

c. Create a new custom property, compress=true. The type of this property must be java.lang.Boolean.

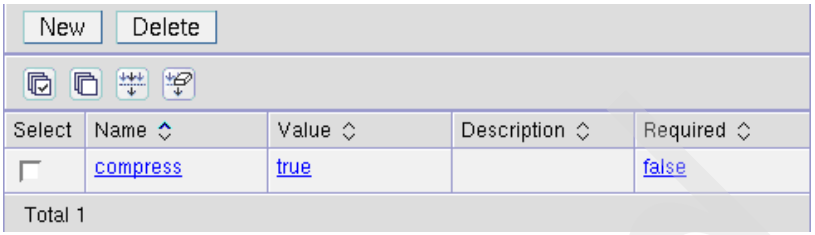

Figure 6-20 CEI Data Store profile

d. Save the configuration in the usual way.

To activate the changes we have made, the Common Auditing and Reporting Service EventServer application must be restarted in WebSphere Application Server. Perform the following three steps to do this:

1. In the WebSphere Application Server Administration console, select **Applications** → **Enterprise Applications**.
2. Put a check box beside both the CommonAuditService and EventServer applications and click **Stop**.
3. Put a check box beside both the CommonAuditService and EventServer applications again and click **Start**.

Configure the Java stored procedure in DB2

In order for reporting applications to read raw event data directly from DB2, a stored procedure is required that can decompress the XML data that was stored by the Common Event Infrastructure component.

In order for DB2 to be able to initialize a JRE, the libraries required must be linked in /usr/bin on our Linux machine. This is done by entering the following commands:

```
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libjava.so /usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/classic/libjvm.so
/usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libhpi.so /usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libjsig.so /usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libdbgmalloc.so
/usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libjitc.so /usr/lib
ln -s /opt/IBM/WebSphere/AppServer/java/jre/bin/libzip.so /usr/lib
ln -s /opt/IBM/WebSphere//AppServer/java/jre/bin/libxhpi.so /usr/lib
```

The Java stored procedure needs to use the WebSphere version of the JDK. To achieve this, modify the DB2 configuration using the following commands:

1. Source the DB2 environment in the current shell for DB2 commands:

   ```
   . /home/carsdb2/sqllib/db2profile
   ```

2. Then run the DB2 command to set the JDK_PATH configuration:

   ```
   db2 update dbm cfg using JDK_PATH /opt/IBM/WebSphere/AppServer/java
   ```

The Common Auditing and Reporting Service server ships with a script that installs the Java stored procedure into DB2. This script is called *ibmcardddinst.sh*. Use the same shell sourced to the DB2 environment as above to enter the following command:

```
/opt/IBM/Tivoli/CommonAudit/server/bin/ibmcarsddinst.sh -u carsdb2 -p
passwOrd -d /opt/IBM/Tivoli/CommonAudit/server/lib
```

Note that this command should be entered on a single line.

This is the expected output:

```
Database Connection Information
Database server        = DB2/LINUX 8.2.1
 SQL authorization ID   = CARSDB2
 Local database alias   = EVENTXML

DB20000I  The SQL command completed successfully.
DB20000I  The CALL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
The JAVA stored procedure for the drill down report installed
successfully.
```

As long as you get a success message at the end of the installation, then the installation should have been successful.

In some cases this command may fail if the DB2 instance was started by another process (for example, the ldapdb2 instance started by the Directory Server start up). In this case, try stopping the DB2 instance and then restarting it on the command line using **db2start**.

Verify IBMCARS_DD_REPORT procedure

Using the same shell so that the DB2 command line is still available and you are still connected to the EVENTXML database as user carsdb2, issue the following command:

```
# db2 "call IBMCARS_DD_REPORT(1)"
```

If this performs a search (even if 0 records are selected), then the procedure is installed and working correctly. If an error is returned that indicates that no valid procedure can be found, the procedure is not installed correctly.

Common Auditing and Reporting Service client installation

Every host from which we expect to gather event data and provide audit reports needs to have a Common Auditing and Reporting Service client. There are two clients to choose from: a Java API client and a C API client.

The reason there are two clients is because the Common Event Infrastructure only provides a Java interface for writing events, which poses a problem for Access Manager servers that are written in C.

To accept audit events created by a C application, the Common Auditing and Reporting Service server includes a Web Service along with the CEI in WebSphere. This Web Service accepts events over a Web Services connection and then forwards them to the CEI using the CEI Java APIs.

The Common Auditing and Reporting Service client is the C code that is called by the Access Manager servers, which then make a Web Services connection to the Common Auditing and Reporting Service server. Therefore, the client must be installed on every machine in the TAMCO environment that hosts Access Manager servers. In the TAMCO environment, this means that all of the Access Manager server machines have to have a Common Auditing and Reporting Service client (see Figure 5-1 on page 76).

The installation package for the Common Auditing and Reporting Service client can be found on the Access Manager Base CD image. There are packages for Windows (cli_setupWin.exe) and Linux (cli_setupLinux.bin). To run them successfully, it is necessary to have an appropriate Java Runtime available. We use the WebSphere Java by running the following command to set the WebSphere JRE in the shell:

```
. /opt/IBM/WebSphere/AppServer/profiles/default/bin/setupCmdLine.sh
# which java
/opt/IBM/WebSphere/AppServer/java/bin/java
```

Next we run the appropriate platform's Common Auditing and Reporting Service client installer. The installation can be done in either silent or interactive mode. The only complicated choice is whether to install both the Java and C API clients. Since we are using the Access Manager Java API, we install both clients. For the Java API client, the JNDI port setting option for the Common Auditing and Reporting Service server needs to be specified in the window shown in Figure 6-21 on page 149. We use the default port of 2809.

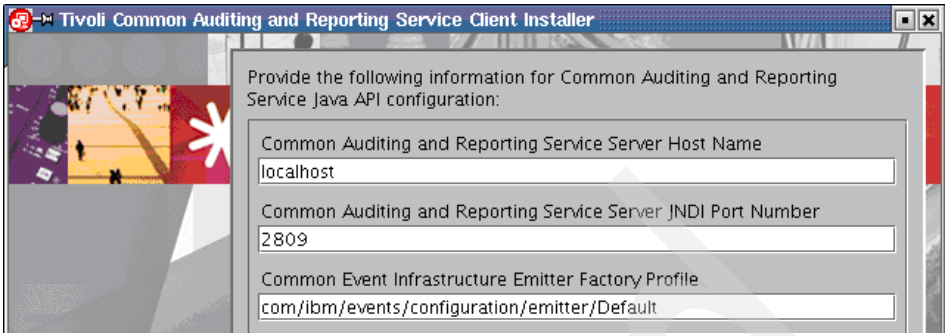

Figure 6-21 JNDI port setting option

When the installation of the client is complete, you will be presented with a success message.

Configuring for Crystal Reports

TAMCO uses Crystal Enterprise for its auditing and other report generation. Since TAMCO is using Crystal Enterprise Server 9, it is already installed, and the necessary prerequisite software is already present and configured (DB2 client and an HTTP server).

Since the Crystal Enterprise Server is running on Windows, it is likely that it is remote from the Common Auditing and Reporting Service server. It communicates with the DB2 database where the Common Auditing and Reporting Service data is stored using a DB2 client to provide ODBC access.

The Crystal Enterprise Server requires a Web server to front-end its Web application. An IBM HTTP Server provided by WebSphere is already installed and can be used for this purpose. Some manual configuration is required to integrate IBM HTTP Server with the Crystal Enterprise Server.

You can verify the installation of Crystal Enterprise by running the Configuration Manager by selecting **Start** → **Programs** → **Crystal Enterprise 9** → **Crystal Configuration Manager**.

Check that all of the services are started (except HTTP SSL and Web Proxy).

Make sure that the latest Crystal Service Pack is installed. You can check the Service Pack level in the Crystal Configuration Manager. Service Pack 2 or above is needed to fix a `rowset not found` error when you try to use the Crystal Reports that come predefined with the Common Auditing and Reporting Service. The latest service pack can be downloaded from:

http://support.businessobjects.com/downloads/updates/service_packs/enterprise.asp

Configure IBM HTTP Server for Crystal Enterprise Server

For details about the configuration, see the document entitled *How to configure the CGI Web Connector with IBM HTTP Server* on the Business Objects Web site. This document should be available at:

http://support.businessobjects.com/communityCS/TechnicalPapers/ce9_ihs_cgi_Web_connector.pdf.asp

Do the following steps for the configuration:

1. Copy the Web Connector file.

   Before you can configure the CGI Web Connector, copy the file wcscgi.cgi, which can be found in C:\Program Files\Crystal Decisions\Enterprise 9\win32_x86, to the cgi-bin directory of IBM HTTP Server.

   The default IBM HTTP Server cgi-bin directory is C:\Program Files\IBM HTTP Server\cgi-bin.

2. Modify IBM HTTP Server configuration.

   Edit the IBM HTTP Server configuration file, which is typically located in C:\Program Files\IBM HTTP Server\conf\httpd.conf, add new aliases and MIME types, and then associate the aliases with the wcscgi.cgi script.

3. Add new aliases to the configuration:

```
Alias /icons/ "C:/Program Files/IBM HTTP Server/icons/"
Alias /crystal/ "C:/Program Files/Crystal Decisions/Web Content/"
Alias /viewer/ "C:/Program Files/Common Files/Crystal
Decisions/2.0/crystalreportviewers/"
Alias /crystalreportviewers/ "C:/Program Files/Common Files/Crystal
Decisions/2.0/crystalreportviewers/"
```

4. Add MIME types:

```
AddType application/x-tar .tgz
AddType image/x-icon .ico
AddType Magnus-Internal/rpt .rpt
AddType Magnus-Internal/csp .csp
AddType Magnus-Internal/cri .cri
AddType Magnus-Internal/cwr .cwr
```

5. Associate MIME types with scripts:

```
# Format: Action handler-name /cgi-script/location
#
Action Magnus-Internal/rpt /cgi-bin/wcscgi.cgi
Action Magnus-Internal/cwr /cgi-bin/wcscgi.cgi
Action Magnus-Internal/csp /cgi-bin/wcscgi.cgi
Action Magnus-Internal/cri /cgi-bin/wcscgi.cgi
```

6. Restart IBM HTTP Server.
7. Edit the Crystal Enterprise HTML pages.
   a. Make a backup of the file C:\Program Files\Crystal Decisions\Web Content\Enterprise 9\ ePortfolio\default.htm.
   b. Make the following change to the default.htm file:

      Before:

      ```
      <HTML> <HEAD>
      <SCRIPT language="javascript"> location.replace ( "redirect.csp"
      ); </SCRIPT>
      </HEAD> </HTML>
      ```

      After:

      ```
      <HTML> <HEAD>
      <SCRIPT language="javascript"> location.replace (
      "en/default.htm" ); </SCRIPT>
      </HEAD> </HTML>
      ```

   c. Copy the modified file to the following directories (overwriting the existing version):
      - C:\Program Files\Crystal Decisions\Web Content\Enterprise9\Websamples
      - C:\Program Files\ Crystal Decisions\Web Content\Enterprise9\help
8. Test the Crystal configuration by launching the Crystal Launchpad in your Web browser using the following URL:

   ```
   http://localhost/crystal/enterprise9
   ```

9. You should see the Crystal Enterprise Launchpad in your browser.

   Check that the ePortfolio and Crystal Management Console links work correctly. You should be able to log into the Management Console with an administrative user name and no password (for example, leave the password box empty).

If everything looks good, then the customization was successful and Crystal Enterprise is ready for the Common Auditing and Reporting Service.

Installation of reports into Crystal Enterprise 9

The Access Manager Common Auditing and Reporting Service license for Crystal Enterprise only allows Crystal to be used to run predefined Common Auditing and Reporting Service reports. To do this task, we must run the Common Auditing and Reporting Service installer again and select the *Operational Reports* component to install:

1. Change the directories to the Access Manager Base CD and go to the Common Auditing and Reporting Service directory. Run the following executable to start the installer:

   ```
   srv_setupWin32.exe
   ```

   Because we have an appropriate JRE installed and in the classpath, this installer should run fine. If it does not, you must set your Java environment appropriately.

2. Move through the windows accepting the defaults until you get to the choice for either the Common Auditing and Reporting Service Event Server or Operational Reports. Check *only* the Operational Reports check box, as shown in Figure 6-22.

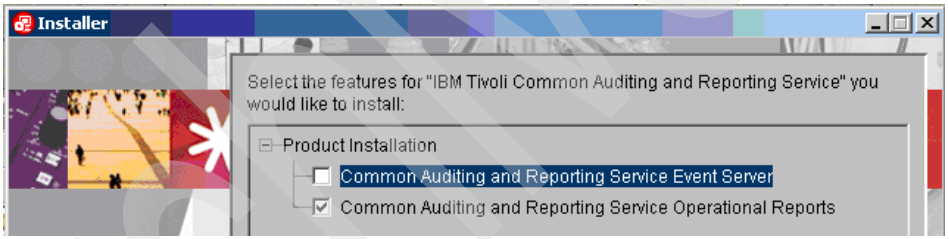

Figure 6-22 Installing the operational reports only

The APS server name is set to the host name of the local machine during installation. Use the name of the Windows machine for the APS server name so that you will remember it.

The APS server administrator ID is, by default, *administrator*, and it has no password.

3. Enter the DB2 user who owns the Common Auditing and Reporting Service Server (for example, *carsdb2*) and the carsdb2 user's password, as shown in Figure 6-23.

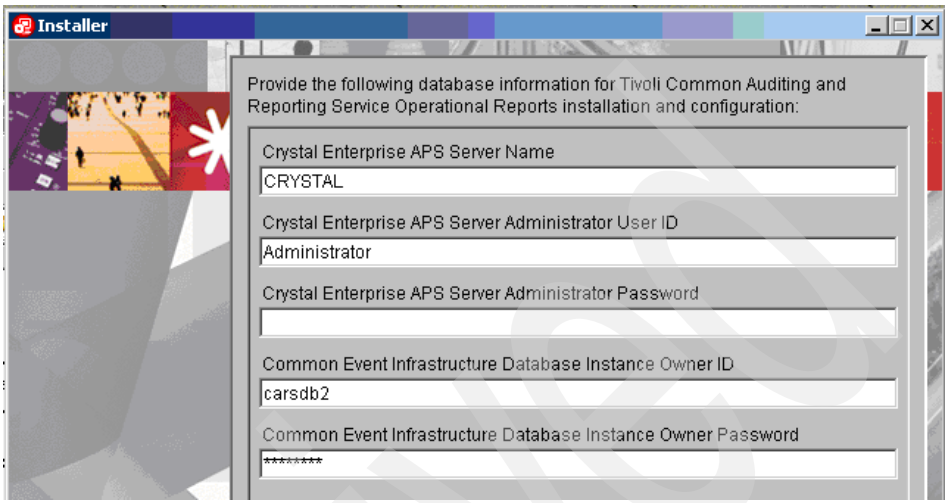

Figure 6-23 Installing the operational reports

4. Click **Next** and then **Finish**.

The Common Auditing and Reporting Service operational reports are now loaded into the Crystal Enterprise and can be accessed through the Crystal Web interface (although they will not work yet). You must log in to the Crystal *ePortfolio* as user Tivoli (no password) to see the reports.

Connect to Crystal Enterprise

In order to produce reports, the Crystal Enterprise server must be able to load data from the DB2 database being used by the Common Auditing and Reporting Service server. The connection is made by configuring an ODBC datasource in the DB2 client on the Crystal Enterprise machine:

1. Launch the DB2 Configuration Assistant by selecting **Start** → **Programs** → **IBM DB2** → **Set-up Tools** → **Configuration Assistant**.
2. Add a database.
3. When the DB2 configuration assistant is launched, you may be presented with a window asking whether you want to add a database. If this happens, click **Yes**. If you do not see this message, go to the Select menu and select the **Add database using Wizard…** option.

The wizard opens and you should see the choices shown in Figure 6-24. Select **Manually configure a connection to a database**.

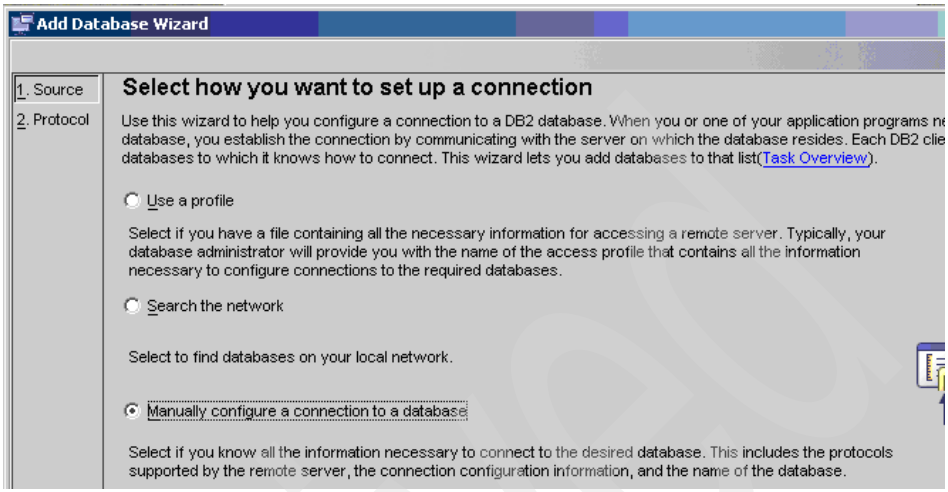

Figure 6-24 Add database wizard

4. Select **TCP/IP** and then supply the host name of the Common Auditing and Reporting Service server and the appropriate port number, as shown in Figure 6-25. (In our case, it is the default of 37000, which we configured the Common Auditing and Reporting Service DB2 database manager to listen on.)

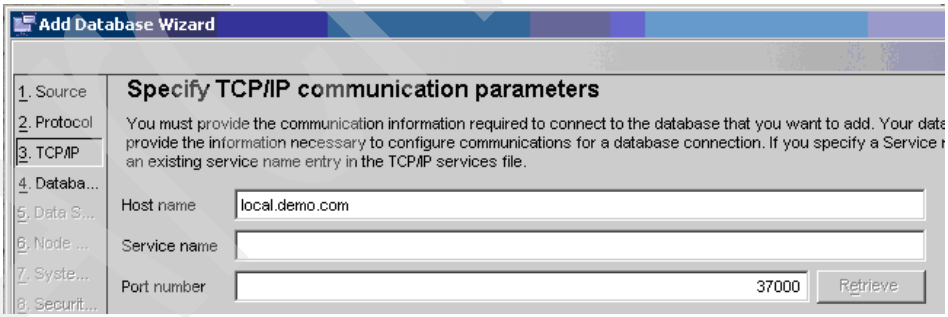

Figure 6-25 Add database wizard continued

5. In the next window, specify the Common Auditing and Reporting Service database we want to connect to. (EVENTXML is the default created by the autoconfiguration of the Common Auditing and Reporting Service server using the ibmcars.properties file.) The *database alias* should be added

automatically (the default is CARSSTG), but if it is not, you have to check it in the ibmcars.properties file (see CARS_HOME/server/etc).

6. Register this database as an ODBC resource because Crystal Enterprise uses ODBC to access the data. Accept the defaults for the data source with CARSSTG.
7. Next, select the appropriate operating system for the node options for the machine on which the CARSDB2 database instance is located (in the TAMCO case, it is on a Windows machine along with Crystal Enterprise; see Figure 5-1 on page 76).

   Provided the CARSDB2 is added successfully, you will see a dialog, shown in Figure 6-26, confirming the successful addition.

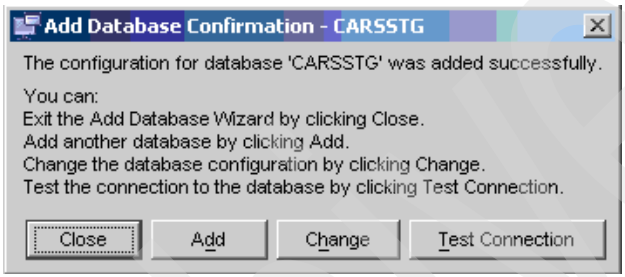

Figure 6-26 Add database wizard continued

8. Click **Test Connection**. The dialogue window in Figure 6-27 is displayed.

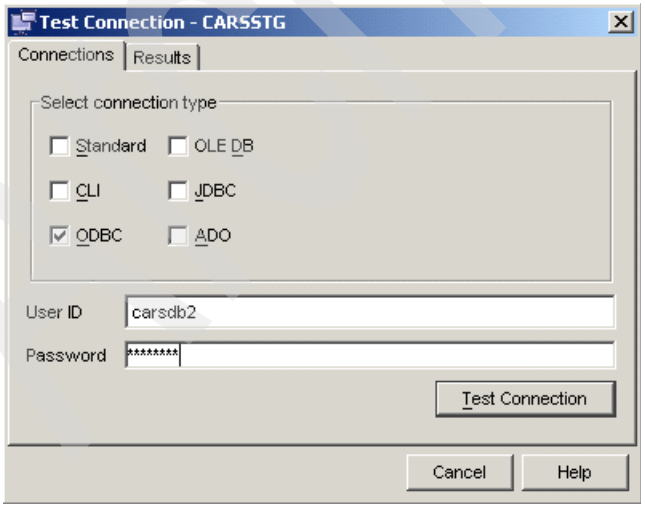

Figure 6-27 Add database wizard: test connection

9. Enter in the values shown in Figure 6-27 on page 155 and click **Test Connection**. You should see a message saying that the ODBC connection was successful.
10. Close the Add Database Confirmation window and close the DB2 Configuration Assistant.

The Crystal Enterprise system is now ready for reporting on Common Auditing and Reporting Service events. Now we need to configure Access Manager to send events to the Common Auditing and Reporting Service so that we can generate reports.

Configure Access Manager

Once the Common Auditing and Reporting Service client is installed, Access Manager components must be configured to use it to send audit events to the Common Auditing and Reporting Service server.

When an aznAPI application starts up, it looks for its audit configuration file to tell it whether to use the Common Auditing and Reporting Service. When Access Manager is installed, no audit configuration files exist and so no Access Manager components attempt to use the Common Auditing and Reporting Service.

The audit configuration file for an Access Manager server has a file name based on the server's name. It is important to create the file with exactly the right name or the aznAPI cannot initialize the Common Auditing and Reporting Service client.

The path for the audit configuration files is <PD_HOME>/etc/audit. This means that in our Linux environment the files will all be found in the /opt/PolicyDirector/etc/audit directory.

Looking in this directory, we can see that a number of template files already exist:

```
# ls /opt/PolicyDirector/etc/audit
.                              pdaudit.pdmgrd.conf.template
..                             pdaudit.pdproxy.conf.template
pdaudit.appsvr.conf.template   pdaudit.pdWeb.conf.template
pdaudit.pdacld.conf.template   pdaudit.pdWebpi.conf.template
```

To enable an Access Manager aznAPI application for the Common Auditing and Reporting Service, a utility called amauditcfg can be used. This creates an audit configuration file based on the appropriate template and completes it appropriately. It creates a configuration file called <PD_HOME>/etc/audit/pdaudit.<server name>.conf for the application.

In the case of the policy server (which has no server name), the file is called:

<PD_HOME>/etc/audit/pdaudit.pdmgrd.conf

Once the configuration is complete, Access Manager starts to use the Common Auditing and Reporting Service the next time it is restarted.

Note: Resource auditing will not occur unless an appropriate POP indicates that auditing is required.

Configure the Access Manager Policy Server

To enable an Access Manager application to use the Common Auditing and Reporting Service, we use the amauditcfg utility. This utility can be used to configure a lot of different options for the Common Auditing and Reporting Service client (for example, caching, SSL communication, and so on), but we will use the minimum options and let everything else use default values.

To see all the options of the amauditcfg utility, use the command:

```
/opt/PolicyDirector/sbin/amauditcfg -help
```

To successfully configure an Access Manager server for the Common Auditing and Reporting Service, the amauditcfg utility requires three parameters:

- -action config
- -srv_cfg_file <AM server aznAPI config file>
- -audit_srv_url <URL of the CARS Web Service>

To configure the policy server, use the following command:

```
# /opt/PolicyDirector/sbin/amauditcfg -action config -srv_cfg_file
/opt/PolicyDirector/etc/ivmgrd.conf -audit_srv_url
http://local.demo.com:9080/CommonAuditService/services/Emitter
```

Make sure you enter the above command on a single line.

You should see a few lines of output ending with:

```
Configuration completed successfully.
```

Configure the Access Manager Authorization Server

To configure the Access Manager Authorization Server, use the following command:

```
# /opt/PolicyDirector/sbin/amauditcfg -action config -srv_cfg_file
/opt/PolicyDirector/etc/ivacld.conf -audit_srv_url
http://local.demo.com:9080/CommonAuditService/services/Emitter
```

Configure Access Manager WebSEAL

To configure WebSEAL, use the following command:

```
# /opt/PolicyDirector/sbin/amauditcfg -action config -srv_cfg_file
/opt/pdWeb/etc/Webseald-default.conf -audit_srv_url
http://local.demo.com:9080/CommonAuditService/services/Emitter
```

Verify audit configuration files

The amauditcfg utility should have created audit configuration files for each of the Access Manager applications we have configured. Check for these files:

```
# ls /opt/PolicyDirector/etc/audit/*.conf
/opt/PolicyDirector/etc/audit/pdaudit.default-Webseald-local.demo.com.c
onf
/opt/PolicyDirector/etc/audit/pdaudit.ivacld-local.demo.com.conf
/opt/PolicyDirector/etc/audit/pdaudit.pdmgrd.conf
```

You can open these files if you want to see the configuration options that have been set.

Restart the Access Manager servers by running:

```
# pd_start restart
```

Verify correct initialization

You can verify that the Common Auditing and Reporting Service client has been initialized for an Access Manager application by checking for the presence of the cache files that are specified in the audit configuration file.

Check for the disk cache files by running:

```
# ls /var/PolicyDirector/cache/cars*
/var/PolicyDirector/cache/cars.default-Webseald-local.demo.com.dat.2005
-08-23-10-31-20.1
/var/PolicyDirector/cache/cars.ivacld-local.demo.com.dat.2005-08-23-10-
31-07.1
/var/PolicyDirector/cache/cars.pdmgrd.dat.2005-08-23-10-31-05.1
```

Verify that events have been stored in DB2

The starting of the Access Manager servers should have created some events in the Common Auditing and Reporting Service XML Event Store. The last check we will perform is to make sure that these events are there and that we can read one of them using the IBMCARS__DD_RECORD stored procedure.

Make sure that the DB2 command line is available:

```
# . /home/carsdb2/sqllib/db2profile
```

Run the following command to connect to the EVENTXML database as user carsdb2:

```
# db2 connect to eventxml user carsdb2 using passw0rd
Database Connection Information
Database server        = DB2/LINUX 8.2.1
 SQL authorization ID   = CARSDB2
 Local database alias   = EVENTXML
```

We can run the following command to list the Common Auditing and Reporting Service events that exist in DB2. It will output a table showing record ID and event type:

```
# db2 "select RECORD_ID,EXTENSION_NAME from cei_t_xml00"
```

If any events are found, then this means that the Common Auditing and Reporting Service client, server, and CEI configuration are all functioning.

You can use the stored procedure we configured earlier to read the XML Events from DB2. Use this command to provide a record ID (1 in this example):

```
# db2 "call IBMCARS_DD_REPORT(1)"
```

If events are not being written to DB2, first check the Common Auditing and Reporting Service client error log file. If this is clear, then check the WebSphere Application Server logs for errors from the Common Auditing and Reporting Service server and JDBC™ connectors.

At this point, the Common Auditing and Reporting Service is fully configured with Access Manager and ready to use.

Using the Common Auditing and Reporting Service

As noted above, we have made the Common Auditing and Reporting Service available, and through use of the administration of Access Manager by either pdadmin or WPM, and the normal authentication and authorization decisions, we have generated some events in Access Manager.

As a final step, we now need to configure Access Manager to properly utilize auditing through the Common Auditing and Reporting Service.

1. Turn on access control decision auditing.

   Access Manager only audits access control decisions if this is specified as a requirement in the Access Manager security policy. This is specified for a given object by the protected object policy (POP) that is effective on that object. By default, no POPs are attached in the objectspace and so no resource access auditing is performed.

2. Create a POP that specifies that auditing is required by running the following **pdadmin** commands:

```
# pdadmin -a sec_master -p passw0rd
pdadmin sec_master> pop create audit-all
pdadmin sec_master> pop modify audit-all set audit-level all
```

   This POP audits *all* access decisions. It is also possible, with different audit-level settings, to audit only successful or only failed access attempts.

3. Attach the POP to the objectspace.

   Our new POP can now be attached to any object we want to audit. Since POPs are inherited, we can attach this POP to the root object. This enforces auditing of all access decisions made by Access Manager. Run the following command:

```
pdadmin sec_master> pop attach / audit-all
```

Using Crystal Reports to view raw Access Manager events

Assuming that some access control decision activity has occurred, there should now be a number of audit events stored in the Common Auditing and Reporting Service XML Event Store. These are considered *raw events* and they are not that useful for reporting. However, we can attempt to view one of them to check that the Crystal Enterprise Server can successfully read from the Common Auditing and Reporting Service database.

1. Connect to the Crystal Enterprise Launchpad (using the browser on the Crystal Enterprise machine) using the following URL:

```
http://localhost/crystal/enterprise9
```

2. Click the **ePortfolio** link and log in as user Tivoli with no password.
3. You are presented with the dialog that shows a link to the Tivoli Common Auditing and Reporting Service reports.
4. Click the **Tivoli Common Auditing and Reporting Service Operational Reports** link and you are presented with a dialog that displays all the predefined Common Auditing and Reporting Service reports.
5. You can choose a report, for example, the *General Audit Event Details Report*, click the link, and then select **View** from the pop-up window, and then enter 1 as the record number to open the *raw event* report.

   Since the data is in the raw, the report should look something Example 6-1 on page 161.

Example 6-1 Raw event data output

```
<CommonBaseEvent creationTime="2005-08-15T15:45:48.636836Z"
extensionName="IBM_CBA_AUDIT_RUNTIME"
globalInstanceId="CE11DA0DA3A7825DA0E2489367C95D0692"
sequenceNumber="2"
version="1.0.1"><extendedDataElements name="outcome"
type="string"><values>CARSComplexType:CARSAuditOutcome</values><childre
n
name="result"
type="string"><values>SUCCESSFUL</values></children><children
name="majorStatus" type="int"><values>0</values></children><children
name="minorStatus"
type="int"><values>0</values></children></extendedDataElements><extende
dData
Elements name="action"
type="string"><values>start</values></extendedDataElements><extendedDat
aElem
ents name="resourceInfo"
type="string"><values>CARSComplexType:CARSAuditResourceInfo</values><ch
ildre
n name="nameInApp" type="string"><values>Not
Available</values></children><children name="nameInPolicy"
type="string"><values>Not Available</values></children><children
name="type"
type="string"><values>protectedResource</values></children></extendedDa
taEle
ments><extendedDataElements name="CARSEventVersion"
type="string"><values>6.0</values></extendedDataElements><sourceCompone
ntId
application="pdmgrd" component="IBM Tivoli Access Manager for
e-business, V
6.0" componentIdType="ProductName" location="local.demo.com"
locationType="FQHostname" subComponent="local.demo.com"
componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_component
Types
"/><situation categoryName="ReportSituation"><situationType
xsi:type="ReportSituation" reasoningScope="INTERNAL"
reportCategory="SECURITY"/></situation></CommonBaseEvent>
```

In order to make this more readable and organized, we need to *stage* the data using the Common Auditing and Reporting Service staging utility.

Staging events ready for reporting

So far, we have only looked at raw audit events in the Common Auditing and Reporting Service database. This is how the events are received from the Common Auditing and Reporting Service client and how they are initially stored by the Common Auditing and Reporting Service server. However, in order to get useful reports from the Common Auditing and Reporting Service, we need to stage the raw data so that it can be used by our reporting engine to generate meaningful reports.

Staging the audit data means parsing the raw data, extracting the important pieces, and putting these into staging tables in the database. The data is copied to the staging tables in a format that can be read directly into reports. This process is also called *shredding*.

The way in which the data is stored in the staging tables is specified in a configuration file (/opt/IBM/Tivoli/CommonAudit/server/etc/CARSShredder.conf). The default configuration allows the supplied operational reports to be run from Crystal Enterprise. Changing the staging configuration is beyond the scope of this book. See IBM *Tivoli Access Manager for e-business Version 6.0 Auditing Guide*, SC32-2202 for more details.

Staging is performed using a Java utility that is part of the Common Auditing and Reporting Service server installation. The easiest way to use this staging utility is to write a script that sets the correct CLASSPATH and executes the utility with the options you want.

The staging utility reads its configuration either from the command line or from the following file:

opt/IBM/Tivoli/CommonAudit/server/etc/ibmcars.properties

This file contains information such as the port to use for connection to DB2. The required information should have been added to the file when the Common Auditing and Reporting Service server was installed. The only information not stored in the file by default is the DB2 database password.

- Running the staging utility

  Run the staging utility to stage data into the report tables. It uses the modified CARSShredder.conf file. The staging utility command uses the following syntax:

  ```
  java com.ibm.cars.staging.Staging [-mode incremental]
  ```

  Optional parameters are:

  – -configurl value

  – -dbport value

- -dbname value
- -dbusername value
- -dbpassword value
- -batchsize value
- -numworkers value
- -progress value
- -help

For the parameters that are not specified on the staging utility command line, their values will be used according to what is set in the ibmcars.properties file. The parameters that you set on the command line will override any that you have set in the ibmcars.properties file.

An example staging utility command is shown below. It shows how to run historical staging beginning on January 1, 2004, at 10:00 p.m. through October 6, 2004 at 10:00 p.m.:

```
java com.ibm.cars.staging.Staging -mode historical -starttime "Jan
1, 2004 10:00:00 PM GMT" -endtime "Oct 6, 2004 10:00:00 PM GMT"
```

- Generating operational reports

  Do the following steps:

  a. First, connect to the ePortfolio application by connecting to the Crystal Enterprise Launchpad (using the browser on the Crystal Enterprise machine) at the URL http://localhost/crystal/enterprise9.

  b. Click the **ePortfolio** link.

  c. Log in as user Tivoli with no password and click the **Common Auditing and Reporting Service Operational Reports** link.

  d. Click **General Administration Event History** and then click **View**.

     This brings up a window where you must provide input parameters for the report generation. The most important one here is the *end date and time*. By default, this is likely to be in the past, which means you will not see many events.

  e. Click the calendar icon and select **Today** in the calendar page that is displayed.

  f. Click **OK** twice to run the report.

  A final example report output is shown in Figure 6-28 on page 164.

This concludes the configuration and usage of the Common Auditing and Reporting Service in conjunction with Tivoli Access Manager for e-business.

General Administration Event History

Report Date	11/17/2005
Report Time	08:30AM
Time Zone	System Default
Begin Time(GMT)	02/25/05 - 11:01PM
End Time(GMT)	02/25/05 - 11:03PM
Product Selected	All
Sort By	Timestamp

General Statistics

Total number of administration events: 242

General Administration Event List

Administrator	Domain	Number of Events	Event Type	Policy Resource Name	Resource Type	Action	Timestamp(GMT)
gp9user	Default	1					
			AUDIT_MGMT_REGISTRY	local		password Change	2/25/2005 11:02:36PM
gp9usr	Default	3					
			AUDIT_MGMT_REGISTRY	local		password Change	2/25/2005 11:01:49PM
			AUDIT_MGMT_REGISTRY	local		password Change	2/25/2005 11:02:10PM
			AUDIT_MGMT_REGISTRY	local		password Change	2/25/2005 11:02:57PM
root	Default	238					
			AUDIT_MGMT_POLICY	/OSSEAL	policy	apply	2/25/2005 11:01:29PM
			AUDIT_MGMT_POLICY	/OSSEAL	policy	apply	2/25/2005 11:01:29PM
			AUDIT_MGMT_POLICY	/OSSEAL	policy	create	2/25/2005 11:01:29PM

Figure 6-28 Example general administration event history report

6.2 Creating the TAMCO object namespace

The security policy is applied by attaching access control list (ACL) policies, protected object policies (POPs), and authorization rules to the objects within the object namespace that represents the physical resources to be protected. The Tivoli Access Manager authorization service makes decisions to permit or deny access to resources based on user credentials and the conditions specified by the security policy.

The following object namespaces were created during the installation of Access Manager:

- /Management
- /WebSEAL

These two object namespaces are adequate to support the TAMCO ACL policies and group protections (ACLs attached to TAMCO groups such as sales, HR, marketing, and so on) stipulated in 4.1, “Defining the access control security policy” on page 38. By using the default security policy that comes with Access Manager, it is possible to satisfy the basic minimums of the TAMCO access control security policy requirements.

6.2.1 Populating the TAMCO object namespaces

Access Manager represents resources within a domain using a virtual representation called the protected object namespace. The protected object namespace is the logical and hierarchical representation of resources belonging to a domain. The structure of the protected object space consists of two types of objects:

- Resource objects

  Resource objects are the logical representation of actual physical resources, such as files, services, Web pages, message queues, and so on, in a domain.

- Container objects

  Container objects are structural components that allow you to group resource objects hierarchically into distinct functional regions.

There are three TAMCO security domains (Finland, Germany, and UK), so there will be a protected object namespace for each. Since TAMCO is using a standard software build approach to their application inventory, each of the resource objects is the same in each domain.

As a general rule, resource objects belong to containers. Since the structure is a hierarchical tree, we start at the top with a *root container object*. Other containers are organized below root, and resources are leaf nodes below the container objects, as depicted in Figure 6-29.

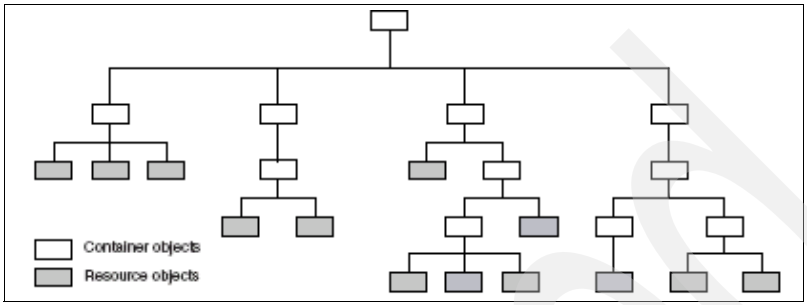

Figure 6-29 General object namespace tree

The root container for each domain is defined as /root<domain name>. Therefore, there will be a protected object space with /root-finland /root-germany and /root-uk.

Figure 6-30 shows the tree and nodes of the default /Management protected objectspace that is installed with Access Manager. When WebSEAL is installed, a /WebSEAL protected object space is also created, with a tree that includes objects contained on the file system of each application server in the WebSEAL domain. As there will be at least two WebSEALs per TAMCO domain (a master and replica for scaling and availability purposes), there will be a container under the WebSEAL named for each domain. The default WebSEAL ACL (default-Webseal) will be applied to each /WebSEAL protected object namespace.

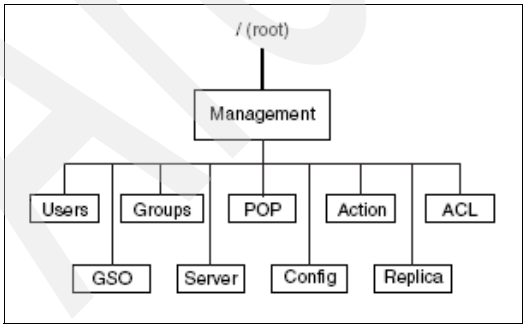

Figure 6-30 Hierarchical view of the management object space with default objects

In order to be able to enforce the TAMCO access control security policy, it is necessary to modify the ACL policies, resource objects, and group memberships by adding TAMCO-specific elements. We have a choice on how to perform these administrative actions: using the WPM or the pdadmin command-line interface. We will use WPM.

Open a browser and enter the link for WPM using the host name and port defined where WPM was configured:

```
http://<localhost>:9080/pdadmin
```

Figure 6-31 depicts a view of the /Management object namespace.

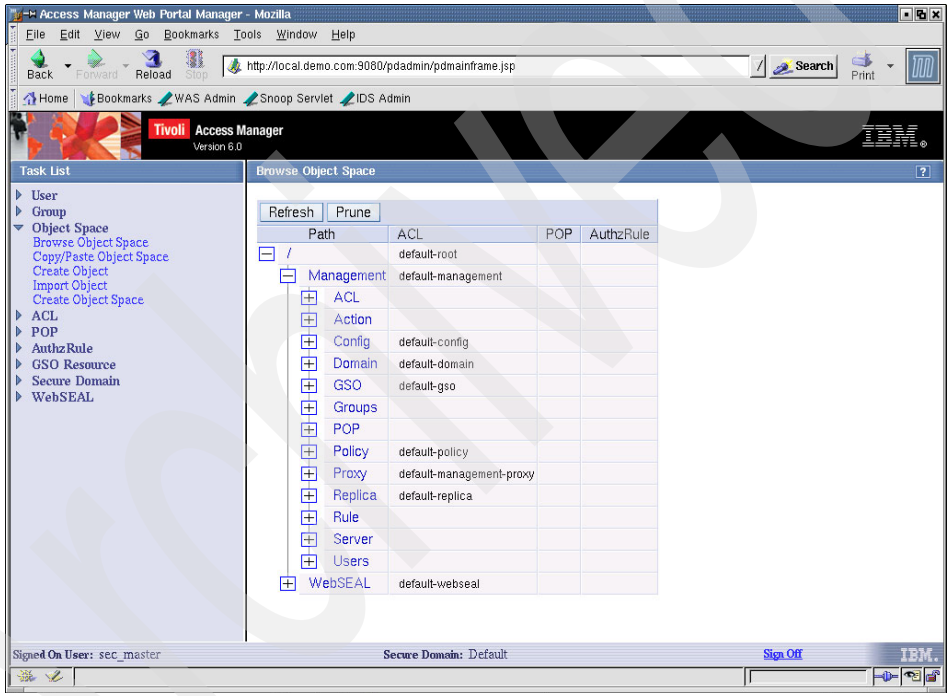

Figure 6-31 Web Portal Manager view showing the management objectspace

Users

According to the TAMCO access control security policy, TAMCO employees should be able to log in to the TAMCO portal using single sign-on (SSO). Therefore, the users in the TAMCO user registry who are TAMCO employees will have to be defined as *GSO users* in the object namespace.

There are also about 100,000 users who are external to TAMCO, either as customers, potential customers, suppliers, and contractors, who have established accounts with TAMCO. Prior to starting the IT modernization project, the TAMCO IT staff made an effort to prune the external user population of duplicates, invalid users, and other identities that no longer belonged in the TAMCO user population. Following this pruning effort, TAMCO IT staff created a second LDIF file with attributes of all the valid external users (tamco_external.ldif).

Defining users as GSO users

Since TAMCO has a policy to support single sign-on, it is necessary to define the TAMCO users in Access Manager as SSO users (users in Access Manager can be either SSO enabled or not). Access Manager's user registry stores these user definitions in a separate container called Global Sign-On (GSO), and the users belong to a group called GSO users. In addition, there is a container object in the Access Manager objectspace for GSO users (/Management/GSO), which gives administrators the ability to manage GSO resources. In addition, a default GSO access control list policy (default-gso) is provided when Access Manager is installed and configured.

Defining the TAMCO users to be GSO-users can be done using either the Web Portal Manager administration interface or by using **pdadmin** commands. We use the following **pdadmin** commands:

```
# pdadmin
pdadmin>login -a sec_master -p <password>
pdadmin sec_master>user create -gsouser <name> <dn> <cn> <sn>
<password>
```

It is also possible to import users from an existing registry, which in the TAMCO case we did by means of the Directory Server bulkload operation from an LDIF file in "Bulkload" on page 127. You remember that when we created the original LDIF file we included an attribute for each existing TAMCO user identifying the user as a GSO user (see 4.2, "The TAMCO deployment" on page 45), so the users are defined in the directory as GSO users. We now need to make these users Access Manager users (recognized by Access Manager). Use the following **pdadmin** command to import the users as Access Manager users:

```
pdadmin sec_master>user import -gsouser <username> dn [group_name]
```

dn is the full distinguished name that exists in the directory (for example, "cn=<name>,ou=Helsinki,o=TAMCO,c=Finland"), and [group_name] is optional, (it is an already existing group to which the user is being added).

Creating WebSEAL junctions

As noted in “Virtual host junctions” on page 51, TAMCO is using virtual host junctioning in order to avoid having to replace many already established and familiar bookmarked URLs throughout their environment.

A virtual host junction works by directing user requests to the junction based on the Host field in the HTTP request. The URI is different from a traditional junction that directs user requests to the junction based on a specific junction name prefixed to the requested back-end path as follows (emphasis added):

- Traditional junctions:

  "www.tamco.com/*sales*/<backend path>
  "www.tamco.com/*support*/<backend path>

- Virtual host junctions:

  "www.sales.demo.com/<backend path>
  "support.demo.com/<backend path>

The virtual host junction's main advantage is to remove the need for WebSEAL to filter requests and use cookies to distinguish server relative from server absolute URLs.

We configure the TAMCO WebSEAL environment to use the following configuration: Our WebSEAL *base* is `www.tamco.com` (hosted in Finland). This allows access to the default local junction.

There are three virtual host junctions (in each domain) where XX represents the geography FI (Finland), UK (United Kingdom), and GE (Germany):

- `www.peoplesoft.XX.tamco.com`
- `www.siebel.XX.tamco.com`
- `www.finance.XX.tamco.com`

Two additional virtual host junctions are created in Finland only:

- `my.tamco.com`
- `www.tamco.com`

Keep in mind that each one of these DNS domain names requires a CA-signed certificate to be created and imported into the WebSEAL keyring.

These junctions connect to different back-end servers running on different machines.

Note: Care needs to be taken in configuring WebSEAL with a single IP address and then associating the three DNS aliases with this address. This allows HTTP connections to the virtual hosts but also results in incorrect host names in the SSL certificates presented.

We configure a different IP address for each DNS alias. We utilize the support in WebSEAL V6.0 to allow it to associate a different SSL certificate with each interface, as a single WebSEAL instance can be configured with multiple interfaces. Each of these interfaces listens on a different IP address/port and can be configured with an independent SSL server certificate.

Do the following steps:

1. The first step is to make sure the correct key database (kdb) and stash files (sth) for the back-end application server certificates are in place. The kdb and associated stash files (files with.sth extension) for the PeopleSoft Application Server, the Siebel CRM server, and the WebSphere Portal server that were created in the relevant sections are used.

   In order to prevent a warning from appearing in the browser when connecting to WebSEAL over SSL, WebSEAL must present an SSL server certificate with a Common Name (CN) that matches the DNS alias requested by the client. A certificate with `cn=www.tamco.com` was created earlier for WebSEAL. In order for WebSEAL to use this SSL server certificate, the configuration must point to the kdb file and identify the certificate. All DNS domain names that are to be served through WebSEAL must have an associated CA certificate imported into the key database on each WebSEAL.

   Make the following changes to the WebSEAL configuration file:

   ```
   [ssl]
   Webseal-cert-keyfile = /var/pdWeb/www-default/certs/tamco-ws.kdb
   …
   Webseal-cert-keyfile-stash =
   /var/pdWeb/www-default/certs/tamco-ws.sth
   …
   Webseal-cert-keyfile-label = www.tamco.com
   ```

2. Second, we need to configure WebSEAL to use forms authentication. Go to the Webseald.conf file for each TAMCO domain and add the following value to the [server] stanza:

   ```
   [forms]
   forms-auth = both
   ```

Restart WebSEAL after making these changes by running the following command:

```
#pdWeb restart
```

3. Third, make sure each of the DNS aliases is associated with a different IP address.

   The reason why WebSEAL cannot send the correct SSL server certificate for each request is that it cannot determine the requested DNS alias until *after* the SSL session has been established. The request to set up the SSL session does not include any DNS information. It is only after SSL is connected and the HTTP request is sent that WebSEAL can read the host header and determine the DNS name. This is too late; SSL certificate exchange is already complete.

   In previous versions of Access Manager, WebSEAL could only be configured with a single SSL server certificate, which it would present on all IP addresses. It was also only possible to configure WebSEAL to listen on a single IP address or *all* IP addresses. To support multiple DNS aliases, multiple WebSEAL instances were required.

   In Access Manager V6.0, a single WebSEAL instance can be configured with multiple interfaces. Each of these listens on a different IP address/port and can be configured with an independent SSL server certificate.

4. Fourth, the PeopleSoft Application Server, the Siebel CRM server, and the MyTamco portal server each have different IP addresses. Hence, in this step we configure WebSEAL to listen on the individual IP addresses by modifying the [interfaces] stanza in the WebSEAL configuration file located at <x>/Program Files/Tivoli/PDWeb/etc/Webseald-<domain>.conf:

```
[interfaces]
peoplesoft =
network-interface=192.168.xx.101;http-port=80;https-port=443;certifi
cate-label=www.peoplesoft.fi.tamco.com
siebel =
network-interface=192.168.xx.102;http-port=80;https-port=443;certifi
cate-label=www.siebel.fi.tamco.com
finance =
network-interface=192.168.xx.103;http-port=80;https-port=443;certifi
cate-label=www.finance.fi.tamco.com
mytamco=
network-interface=192.168.xx.104;http-port=80;https-port=443;certifi
cate-label=my.tamco.com
```

Note: The [interfaces] stanza is new in WebSEAL V6.0. The values we entered tell WebSEAL which ports to listen on and which SSL configuration to use.

5. Lastly, because we are affecting the objects to be protected by WebSEAL, the /WebSEAL object namespace is modified to reflect the addition of the virtual hosts. (The new virtual hosts are shown in WPM when you browse the /WebSEAL object space.)

We use the pdadmin command-line interface to set up the virtual host junctions. It is also possible to do the same steps using WPM.

1. Start the pdadmin command-line interface and sign in as sec_master:

```
# pdadmin -a sec_master -p passw0rd
```

2. Create the TCP virtual host junction for each of the target servers (PeopleSoft, Siebel, and MyTamco WebSphere Portal), such as www.peoplesoft.fi.tamco.com. Note that it is the -v flag that provides the virtual host that triggers the virtual host junction. This also implies that the back-end server will answer requests for this virtual host:

```
pdadmin sec_master> s t default-Webseald-local.demo.com virtual
create -t tcp -v www.peoplesoft.fi.tamco.com -h
tamco-http1.fi.tamco.com -p 80 peoplesoft
Created Virtual Host Junction at peoplesoft
```

Note: Make sure the command is entered as a single line.

3. Next, we need to create an SSL virtual host junction for SSL traffic. The -g flag is used to specify the existing virtual host junction that is paired with this new one:

```
pdadmin sec_master> s t default-Webseald-local.demo.com  virtual
create -t ssl -v www.peoplesoft.fi.tamco.com -h
tamco-http1.fi.tamco.com -p 443 -g peoplesoft peoplesoftssl
Created Virtual Host Junction at peoplesoftssl
```

4. Once we have created these pairs, test the connectivity for both http and https by using a browser to connect to each of the addresses using both http and https:
   - www.peoplesoft.fi.tamco.com
   - www.siebel.fi.tamco.com
   - www.finance.fi.tamco.com
   - my.tamco.com
   - www.tamco.com

   http://www.tamco.com still connects you to the root junction of the base WebSEAL. After authenticating (with sec_master and passw0rd), you are presented with the default WebSEAL page.

Connecting using the other addresses should trigger the appropriate virtual host junction and, after authentication and authorization, the request should be forwarded to the back-end application server (as specified in the junction configuration). The host header is sent in the forwarded request and so the appropriate site page is returned from the appropriate application server.

6.3 Configuring TAMCO single sign-on

Web applications that are being protected by WebSEAL normally have security enabled on the back end and require the user to log in. In essence, there could be multiple logins: one for WebSEAL and one for the back-end server challenging the user unless we leverage one of WebSEAL's SSO mechanisms. After a user has been authenticated and authorized, WebSEAL can include information about the user when forwarding the request to the back end. This information can include details such as X.509 distinguished name, group memberships, or any other value required in order to provide back-end authentication. WebSEAL supports several mechanisms for forwarding requests to Web application servers.

TAMCO configures SSO to use what is called *forms single sign-on* (FSSO). Forms single sign-on authentication allows WebSEAL to log in an authenticated Tivoli Access Manager user to a back-end junctioned application server that requires authentication using an HTML form without the need for a second authentication.

Forms-based single sign-on is built on the following process:

1. WebSEAL interrupts the authentication process initiated by the back-end application.
2. WebSEAL supplies the data required by the login form and submits the login on behalf of the user.
3. WebSEAL saves and restores all cookies and headers.
4. The user is unaware of the second login taking place between WebSEAL and the back-end application.
5. The back-end application is unaware that the login form is not coming directly from the user.

The login form from the back-end application can be filled in with a variety of information from WebSEAL, such as:

- Static text
- GSO user name and password
- Values contained within a user's credential

The back-end applications that use forms authentication are PeopleSoft Application Server, Siebel CRM, and the WebSphere Portal (MyTamco portal). The SSO configuration in all three cases requires two steps:

1. Configure WebSEAL to support forms-based authentication and to provide the appropriate data over an HTTP header to PeopleSoft and Siebel servers, respectively, through an SSO junction.
2. Configure PeopleSoft, Siebel, and WebSphere Portal servers to use WebSEAL as their authentication source.

The reason for using forms authentication is that all three back-end application servers have a Web server plus an authentication servlet that serves an authentication form. Therefore, WebSEAL must be set up as the proxy for the servlet and be able to provide the form back to the back-end server.

6.3.1 WebSEAL forms single sign-on configuration

We have to modify certain stanzas in the WebSEAL main configuration file, Webseald-default.conf, to enable forms single sign-on. We also need to create a forms single sign-on configuration file and an SSO junction with the correct parameters.

A WebSEAL junction is required to connect WebSEAL to the PeopleSoft application server. The junction can be set up to use either the SSL (recommended) or TCP protocol. For details of the general WebSEAL junction creation command, refer to *IBM Tivoli Access Manager Version 6.0 Administration Guide,* SC32-1686. In addition to the standard procedures for junction creation, the PeopleSoft integration requires the following parameters:

1. Specify the -c iv_user flag to allow WebSEAL to pass the user ID to PeopleSoft in the form of an HTTP header.
2. It is no longer necessary to enable JavaScript filtering (for example, the -j flag is no longer required because of the new Access Manager support for virtual hosts in Version 6.0).
3. Specify the -S parameter to provide the path to the forms single sign-on configuration file, which must be created manually.
4. Ensure that the host name supplied with the -h parameter is the fully qualified DNS name (FQDN) of the PeopleSoft application server machine.
5. If the PeopleSoft application server is running on a *non-default* HTTP port, add the -v fqdn:port parameter. For example (entered as one line):

```
pdadmin> server task Webseald-server_name create -t tcp -h
peoplesoft_fqdn -p port_no -v peoplesoft_fqdn:port_no -c iv_user -j
/jct_name
```

So, specifically, you need to log in as sec_master and issue the server task command to create the junction:

```
# pdadmin> login -a sec_master -p password
pdadmin sec_master> server task tamco-ws.fi.tamco.com create -t ssl -h
tamco-http1.fi.tamco.com -c iv_user -S
/Program/Files/Tivoli/PDWeb/fsso/fsso.conf /psft
```

In the WebSEAL configuration file (Webseald-default.conf), we make changes to the following stanza:

```
[forms]
forms-auth=both
```

6.3.2 PeopleSoft single sign-on configuration

PeopleSoft provides a number of applications that perform various tasks related to human resources management and customer relationship management. The applications are deployed on the PeopleSoft Application Server, which itself is deployed as a enterprise archive (.ear file) in WebSphere Application Server. Access to PeopleSoft applications is provided through a PeopleSoft portal application that is also deployed in WebSphere and is administered and configured using PeopleTools utilities.

TAMCO uses the PeopleSoft Human Resources Management System (HRMS) application that is defined as a node of the PeopleSoft portal. As noted earlier, the PeopleSoft HRMS uses a relational database to store its data about users and groups.

The integration with Access Manager to achieve SSO involves configuring both the PeopleSoft Application Server and WebSEAL. In addition, it is necessary to add some functions to the PeopleSoft code to allow WebSEAL to take over the authentication and single sign-on that would normally be done by PeopleSoft.

The Access Manager - PeopleSoft integration is summarized in this section. The complete documentation can be obtained from the following Web site:

http://www.ibm.com/support/docview.wss?uid=swg24003606

Figure 6-32 shows the integration architecture and illustrates the following steps:

1. A client accesses PeopleSoft applications through WebSEAL.
2. WebSEAL intercepts the request, authenticates, and authorizes the user.
3. On successful authentication, WebSEAL passes the request to the PeopleSoft application, together with the authenticated user ID (iv-user), as part of the HTTP header.
4. PeopleTools is configured to accept and trust this user ID from WebSEAL using additional functions added to the PeopleSoft PeopleCode.
5. The PeopleTools Application Server logs in the user and grants user access.

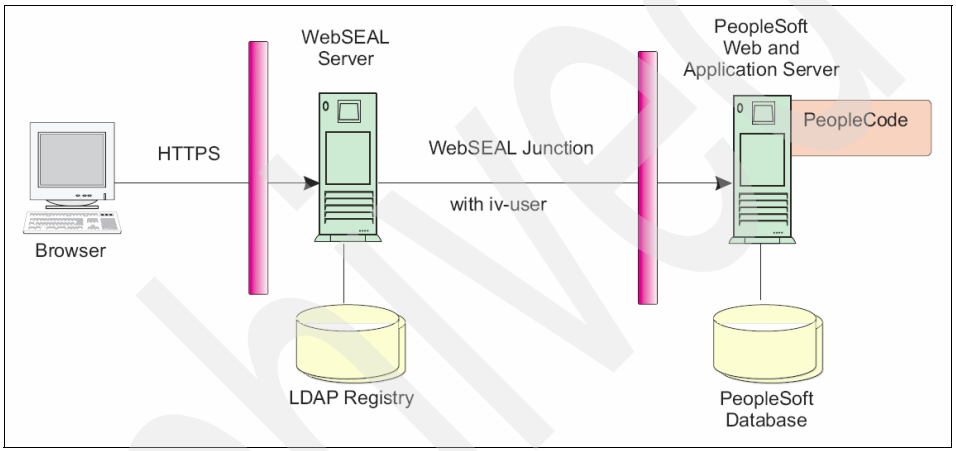

Figure 6-32 Tivoli Access Manager integration with PeopleSoft using WebSEAL

A new PeopleSoft user profile (we will use the sso_default_user, which maps to the PeopleSoft default user expected by PeopleSoft) must be created within the PeopleSoft database for an identity that the Web application server can use to authenticate to the PeopleSoft application server. This user is used to create a trust relationship between the PeopleSoft Web content and the PeopleSoft database.

To create the new user profile, perform the following steps described in the following sections.

Create a WebSEAL junction

In addition to the standard procedures for junction creation, this integration requires the following parameters:

1. Specify the -c iv_user flag to allow WebSEAL to pass the user ID to PeopleSoft in the form of a HTTP header.

2. Specify the -j parameter to enable JavaScript filtering.
3. Ensure that the host name supplied with the -h parameter is the fully qualified domain name (FQDN) of the PeopleSoft application server machine.
4. If the PeopleSoft application server is running on a non-default HTTP port, add the –v fqdn:port parameter. This must only be used for non-default HTTP ports (that is, not for port 80).

Update the PeopleTools configuration

PeopleTools must be configured by editing the appropriate Web profile as follows:

1. Log into PeopleSoft using a browser.
2. From the menu, select **PeopleTools** → **Web Profile** → **Web Profile Configuration**.
3. Click the **Search** button.
4. Click the appropriate Web profile (for example, DEV).
5. In the General tab:
   a. Uncheck **Compress Responses**.
   b. Uncheck **Compress Response References**.
   c. Clear the Authentication Domain field.
6. On the Security tab:
   a. In the Public Users section, select **Allow Public Users**. Enter the new user ID and password (created in the previous section) in the User ID and Password fields.
   b. In the Authenticated Users section, enter a value in the Inactivity Logout field (for example, 86400). This must be greater than the WebSEAL session idle timeout, otherwise a security vulnerability can be created.

Add and enable the custom SignOn Peoplecode provided with the Access Manager - PeopleSoft integration solution.

6.3.3 Siebel single sign-on configuration

TAMCO uses Siebel's Customer Relationship Manager (CRM) application, which is one of several e-business applications running on the Siebel Application Server, to manage their relationship data with their customers.

Siebel provides a security adapter interface to the e-business applications that permits the integration of an external user registry through the implementation of a third-party security adapter (in our case, through WebSEAL). This security adapter interface allows for external management of Siebel roles and credentials for enterprise systems.

This section provides a description of the steps required to configure both Tivoli Access Manager and Siebel to utilize the Tivoli Access Manager security adapter. User and role management are externalized to Tivoli Access Manager. The administrator can use Tivoli Access Manager user groups to delegate Siebel role memberships to Tivoli Access Manager users. Figure 6-33 on page 179 shows the basic architecture of the solution.

The complete integration documentation can be found at:

http://www.ibm.com/support/docview.wss?uid=swg24005918

There are three options from which to choose to implement the Access Manager-to-Siebel integration. All three implement SSO, but each have different impacts on performance and administration:

1. The Siebel LDAP security adapter, which integrates at the directory level and requires the Access Manager and directory IDs to match.
2. The Access Manager Security Adapter for Siebel, which allows a Siebel administrator to administer the server remotely and allows the administrator to use Access Manager user groups and delegate Siebel roles to Access Manager users.
3. The Access Manager Lightweight Security Adapter, which provides the functionality to store logon details for the Siebel Database in a local configuration file. This is useful in heavily firewalled architectures where communication from the Siebel server is undesirable, other than standard Web traffic. This solution is *lightweight* in that the user and role management is administered through Siebel. The adapter does not provide this functionality.

The second option, using the full Access Manager Security Adapter for Siebel Web Extension (SWE), fits the TAMCO case best because of the TAMCO requirement to have administrators remotely manage the servers in the three dispersed domains, and to have the ability to delegate administration.

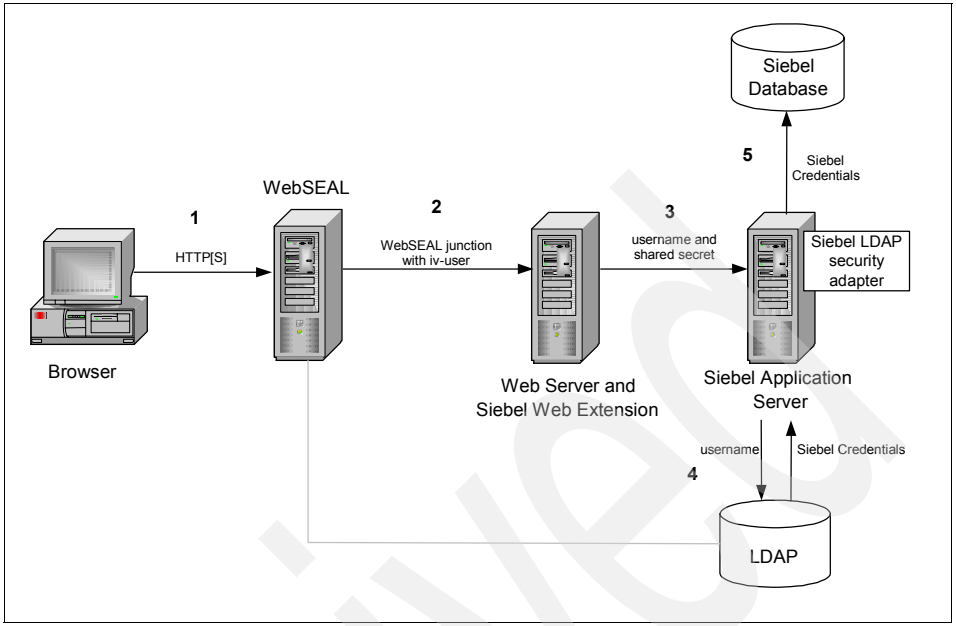

Figure 6-33 Tivoli Access Manager integration with Siebel

The flow shown in Figure 6-34 on page 189 shows an integration scenario architecture that supports the following process:

1. A client requests a Siebel resource.
2. WebSEAL authenticates the client, and passes the user name (as iv-user) in the HTTP header to the Siebel Web Extension (SWE).
3. The Siebel Web Extension passes the user name along with the shared secret to the Siebel Application Server.
4. The Siebel Application Server calls the security adapter, which returns to the Siebel server the credentials needed to log the user into the Siebel database. The credentials are stored in a local configuration file for the Siebel Application.
5. The credentials are then used to log into the Siebel database.

A WebSEAL junction is required to connect WebSEAL to the Siebel Web Extension.

WebSEAL junction

Use the pdadmin command-line interface to create a WebSEAL junction to the Web server that is running the Siebel Web Extension.

When creating the junction, specify the -c iv-user option to pass the Tivoli Access Manager user name in the IV-USER HTTP header. For example (entered as one line):

```
pdadmin> server task Webseald-server_name create -t ssl -h
siebel_server_name -p 443 -v siebel_server_fqdn -c iv-user
/junction_point
```

Where:

- *Webseald-server_name* is the WebSEAL server name, displayed in the exact format as displayed in the output of the **pdadmin server list** command.
- *siebel_server_name* is the DNS host name or IP address of the target back-end server.
- *siebel_server_fqdn* is the virtual host name of the back-end Siebel server.
- *junction_point* is the name given to the junction between WebSEAL and the back-end server.

Note: When creating the junction, we *do not* specify the -j option because script filtering is not necessary and because we are using the new virtual host junctions in Access Manager V6.0.

Enter the following command (on one line):

```
pdadmin sec_master> server task tamco-ws.fi.tamco.com create -t ssl -h
tamco-siebel.fi.tamco.com -c iv-user /Siebel
```

Note: This **pdadmin** command has to be used for the WebSEALs in the other two domains as well. Be sure to modify the FQDN appropriately.

As described in “Virtual host junctions” on page 51, WebSEAL supports virtual hosting. The use of virtual host junctions eliminates the limitations of URL filtering. Virtual host junctions allow WebSEAL to communicate with local or remote virtual hosts. WebSEAL uses the HTTP host header in client requests to direct those requests to the appropriate document spaces located on junctioned servers or on the local machine.

Local DNS servers need to be updated so that references to the Siebel Web Extension actually reference the WebSEAL server. Virtual hosting introduces some DNS and session management challenges. For details, refer to *IBM Tivoli Access Manager for e-business Version 6.0 WebSEAL Administration Guide,* SC32-1687.

As an example, you create the virtual host junction using the following command syntax:

```
pdadmin> server task instance-Webseald-server_name virtualhost create
-t tcp -h siebel_fqdn –p port_no -c iv_user vhost_label
```

Refer to the *IBM Tivoli Access Manager for e-business Version 6.0 WebSEAL Administration Guide,* SC32-1687 for more details on virtual host junctions.

By default, Siebel is configured to listen on port 80. Therefore, to ensure the virtual host junction works correctly, do one of the following:

- Configure WebSEAL and Siebel to use the same ports.
- Configure WebSEAL with an interface listening on the same ports as Siebel.

To configure a WebSEAL interface:

1. Open the Tivoli Access Manager WebSEAL configuration file.
2. Within the [interfaces] stanza, add an new interface and specify the options. For example, to configure an interface to listen on ports 80 (for http) and 443 (for https), enter:

   ```
   interface1 = http-port=80;https-port=443
   ```

3. Save and close the configuration file.
4. Restart WebSEAL.

Having configured WebSEAL, we now must configure the Siebel side of the integration, which means configuring three separate applications:

- The SWE
- The specific Siebel security adapter (In this case, the version we configure is the full Access Manager Security Adapter for Siebel 7.7/7.8.)
- The Siebel applications

Configuring the Siebel Web Extension

When setting up the junction in the previous section, the –c iv-user option was specified, causing WebSEAL to pass the user name of the authenticated user in the HTTP header. The Siebel Web Extension must now be configured to extract the user name from the HTTP header of the incoming request. It can then forward the user name to the Siebel server.

Configure the Siebel Web Extension as follows:

1. Edit the Siebel Web Extension configuration file eapps.cfg (located in the Siebel_install_dir/SWEApp/bin directory) to read the HTTP header and pass the user name to the Siebel Applications. For example, to enable SSO for the callcenter_enu and esales_enu applications, the sections for each of those applications in the configuration file eapps.cfg would be as follows:

```
[/callcenter_enu] ...
SingleSignOn = TRUE
UserSpec = IV-USER
UserSpecSource = Header
ProtectedVirtualDirectory = /callcenter_enu
[/esales_enu] ...
SingleSignOn = TRUE
UserSpec = IV-USER
UserSpecSource = Header
ProtectedVirtualDirectory = /esales_enu
```

   Note: The Siebel Web Extension explicitly trusts the information it receives in the HTTP request. For security reasons you should ensure that the only network path to the Siebel Web server's endpoint is through WebSEAL. This prevents users from accessing the Siebel Web Extension directly and masquerading as a different user. If this is not possible, the use of SSL junctions is recommended. Make the above changes for each Siebel application requiring Tivoli Access Manager protection and single sign-on capability.

2. Configure a shared secret that can be distributed between the Siebel Web Extension and the applications so that information can be trusted. The secret is specified in the Siebel configuration file, eapps.cfg, as follows:

   `[defaults] ... TrustToken = sso_shared_secret`

   Where *sso_shared_secret* are plaintext characters.

3. Next, set the user name for the anonymous user in the Siebel eapps.cfg configuration file for each Siebel application requiring protection. This user name represents the privileges of an anonymous user. It should match a user name found in the Siebel user table. The anonymous password must be set to any string. For example:

   `[defaults] ... AnonUserName=GUESTCST AnonPassword=secret`

4. Configure the Siebel Web Extension Server to disable compression:

   `[defaults] ... DoCompression = FALSE`

5. Set the Siebel session timeout to be at least 5 minutes greater than the WebSEAL inactive timeout. For example:

   ```
   [defaults] ... SessionTimeout=1800
   ```

6. After performing the above changes, save and close the file.
7. For Siebel 7.7 and 7.8 only:
   a. Open Siebel_install_dir/SWEApp/PUBLIC/lang/default.htm.
   b. Locate the following GotoUrl JavaScript function:

      ```
      function GotoUrl(url) { // Append the current hostname to the
      server request so that the server has // the top level host name.
      This is needed to support reverse proxy servers. url += "&SWEHo="
      + this.location.hostname; this.location =
      unescape(this.location.pathname) + url; }
      ```

   c. Change this to:

      ```
      function GotoUrl(url) { // Append the current hostname to the
      server request so that the server has // the top level host name.
      This is needed to support reverse proxy servers. url +=
      "&SWEHo=Siebel_Webserver_hostname/FQDN"; this.location =
      unescape(this.location.pathname) + url; }
      ```

      Where *Siebel_Webserver_hostname/FQDN* is the value used in the -h (or -v) parameter of the junction create command in WebSEAL.
   d. Save and close the file.
8. Restart the Web server.

Configure the Access Manager Security Adapter for Siebel 7.7 and 7.8

Edit the configuration for each Siebel application object to enable single sign-on and to specify a shared secret that matches the one entered in the Siebel Web Extension configuration file.

The following sections describe the steps required for configuring the full Access Manager Security Adapter.

Perform the following steps to configure the Tivoli Access Manager Adapter:

1. Start the Siebel client and log in as SADMIN, the Siebel Administrator.
2. From the Navigate menu, select **Site Map**.
3. Click **Administration - Server Configuration**.
4. Click **Enterprises - Profile Configuration**.

5. Click **New**.
6. Enter the following values:
   - Profile: TAM Security Adapter
   - Alias: TAMSecAdpt Subsystem
   - Type: InfraSecAdpt_Custom
   - Description: Tivoli Access Manager Security Adapter
7. Select **Menu**, then **Save Record**.
8. In the bottom frame, enter the parameters shown here, and keep the defaults of the remaining ones:
   - Security Adapter Dll Name: PDSiebel.dll
   - Config File Name: SIEBEL_HOME\siebsrvr\BIN\lang\PD.cfg
   - Config Section Name: PD Single Sign On: True
   - Trust Token: value
9. Create a new configuration file at SIEBEL_HOME\siebsrvr\BIN\lang\PD.cfg.
10. Create a stanza in the PD.cfg file called [PD].
11. In the new [PD] stanza, add the startup parameters as shown below. The complete set of configuration parameters is explained in the table on page 25 of the Access Manager - Siebel Integration Guide[1]:

```
[PD] AdminName = sec_master
AdminPassword = admin_pw
Domain = Default
DefaultCredentialUserID = username
ContextPoolSize = 3
RoleGroupPrefix = SiebelRole-
CredentialTypePrefix = SiebelCred-
Tracefile = path_to_tracefile
```

Configuring Siebel Applications for Siebel 7.7/7.8

The final piece of the puzzle is to configure the applications to make use of the Access Manager Security Adapter. Perform the following steps to configure the Siebel applications:

1. Start the Siebel client and log in as SADMIN, the Siebel Administrator.
2. From the Navigate menu, select **Site Map**.
3. Click **Administration - Server Configuration**.

[1] The Integration Guide is part of the integration download package and can be obtained at: http://www.ibm.com/support/docview.wss?uid=swg24005918.

4. Click **Servers - Components - Parameters**.
5. In the middle frame, select the application for which you wish to enable single sign-on (for example, Customer Relationship Manager).
6. In the bottom frame, select **Parameters**.
7. In the bottom frame, click **Query** and search for Security*.
8. Change the value of Security Adapter Mode to one of the following options:
   - LDAP for Scenario One
   - Custom for Scenario Two or Three
9. Change the value of Security Adapter Name to:

   TAMSecAdpt

When the above changes are made, the component group must be restarted. Either restart the entire Siebel server, or follow the steps below to restart only the modified application's component group:

1. From the Navigate menu, select **Site Map**.
2. Click **Administration - Server Management**.
3. Click **Servers - Component Groups**.
4. In the middle frame, select the appropriate component group (for example, Siebel Call Center).
5. In the bottom frame, select the appropriate component (for example, Call Center Object Manager).
6. Click **Shutdown**.
7. Wait until the component has been stopped. You may need to refresh the bottom frame by selecting **Query** and then **Go with an empty query**.
8. Click **Startup**.

Changing Siebel logout behavior

A final part of the integration is to ensure correct logout behavior for each Siebel application configured for SSO through Access Manager. Each application has to have its default Siebel logout behavior modified so that when users select **Log Off** from the Siebel application they will be logged out of Access Manager. This is important in order to ensure that there are no inobvious open Access Manager sessions.

The logout behavior is modified as follows:

1. Copy the TAMLogout.swt file to the Siebel_install_dir\siebsrvr\WEBTEMPL directory.

Note: The TAMLogout.swt file redirects to /pkmslogout, which only logs out Tivoli Access Manager users when using Forms Authentication. Where basic authentication is being used, users should be redirected to a suitable alternate location.

2. Stop the Siebel server process.
3. Start Siebel Tools.
4. Create a new project as follows:
   a. In the Object Explorer window, click **Project**.
   b. Create a new record by selecting **Edit**, then **New Record**.
   c. Enter TAM Logout as the new record's name, and select **Locked**.
5. Create a Web template as follows:
   a. In the Object Explorer window, click **Web Template**.
   b. Add a new record named TAM Logout.
   c. Enter TAM Logout for the project parameter.
   d. Specify Web Page Template for the type parameter.
6. Create a Web template file as follows:
   a. Expand the **Web Template** tree and click **Web Template File**.
   b. Add a new record named TAM Logout.
   c. Enter TAMLogout.swt for the file name parameter.
7. Create a Web page as follows:
   a. In the Object Explorer window, click **Web Page**.
   b. Add a new record named TAM Logout.
   c. Enter TAM Logout for the Project parameter.
   d. Select **TAM Logout** for the Web Template parameter.
8. Lock the application project for each project requiring the logout page to be changed, as follows:
   a. In the Object Explorer window, click **Project**.
   b. Locate the appropriate project and select **Locked** (for example, Siebel Universal Agent for Call Center, eSales for eSales).
9. In the Application window, select the Siebel module to be configured. (Each module must be configured separately.)

10. Scroll to the right and locate the Logoff Acknowledgement Web Page parameter. Make a note of this value before changing it. Select **TAM Logout** (the Web page created in step 6).
11. Repeat steps 9 and 10 above for all Siebel modules that need to be configured.
12. Compile the changes as follows:
    a. Select **Tools**, then **Compile Projects**.
    b. Select **Locked Projects**.
    c. Enter the name of the Siebel repository file (for example, Siebel_install_dir\siebsrvr\OBJECTS\lang\siebel.srf).
    d. Click **Compile**.
13. Unlock all locked projects.
14. Exit Siebel Tools.
15. Restart the Siebel server and the Web server.

Note: If the Siebel server is installed on a UNIX platform, and the Siebel Tools client is installed on a Windows machine, the newly compiled .srf file will need to be copied to the UNIX machine.

Testing the integration

When the above three configurations are complete, plus the logout behavior changed, it is useful to test the integration to make sure all is working as expected. Two test programs are included in the integration package for the Access Manager Security Adapter (the *pdadmin_pool_test* and the *PDSiebelTest*).

To test the integration:

1. Open a browser and access a Siebel application through WebSEAL and the junction that was created:
   – Standard junction:

     `http[s]://Webseal_fqdn/junction_point/siebel_app`

   – Transparent path junction:

     `http[s]://Webseal_fqdn/siebel_app`

   – Virtual host junction:

     `http[s]://siebel_fqdn:port/siebel_app`

Where:

- *Webseal_fqdn* is the fully qualified domain name of the WebSEAL server.
- *junction_point* is the junction created in the WebSEAL configuration step above.
- *siebel_app* is the Siebel application that was junctioned to (for example) callcenter_enu.

2. WebSEAL displays an authentication challenge. Log in using the Tivoli Access Manager user name and password. Remember that this user name must also have a corresponding account in the Siebel user table.
3. Upon successful authentication, the Siebel application will be displayed. Some Siebel applications will display a logout option in the top right-hand corner, verifying that SSO has been achieved.

6.3.4 WebSphere Portal single sign-on configuration

The TAMCO portal (MyTamco) provides a basic user interface that allows access to back-end application functionality, depending on the user's role, which is represented by being a member of one of the TAMCO Access Manager groups. This authorization is currently determined by the user's role assignment within the WebSphere Portal server.

SSO to the TAMCO portal is an essential requirement. As with the PeopleSoft and Siebel integrations with Access Manager to provide SSO, the simplest way to provide portal SSO is to use WebSEAL as a reverse proxy to the portal. The full guide to the integration can be found at the following Web site:

`http://www.ibm.com/support/docview.wss?uid=swg24005919`

SSO from a portlet hosted in a WebSphere Portal server is typically done using credential data stored in the credential vault. In most deployments, the data in the credential vault needs to be kept synchronized with the credentials stored in the target user registries. WebSphere Portal provides a facility where its credential vault data can be stored in the Access Manager *GSO lockbox*, rather than the usual location of a relational database table. This integration is done transparently to the portlets, such that a portlet is unaware of whether the credential data is coming from the portal database or from Access Manager.

By integrating the credential vault with Access Manager, a set of credentials can be stored in one place and used by either the WebSEAL component of Access Manager or a portlet, as needed. Moreover, Access Manager provides a command-line interface, and (Java and C) APIs for maintaining the credential data that allows a product, such as IBM Tivoli Directory Integrator, to be used to

keep the credential data automatically synchronized with the back-end user registries.

From a security perspective, storing the credentials in Access Manager implies that the credentials are now transported between Access Manager and WebSphere Portal servers through a mutually authenticated (using X.509 certificates) SSL connection.

From a system availability and maintenance perspective, it is generally preferable to use an identity assertion or credential propagation style SSO, rather than storing, maintaining, and using separate credentials to do a *screen-scraping* approach to SSO, such as that provided by the WebSphere Portal APIs.

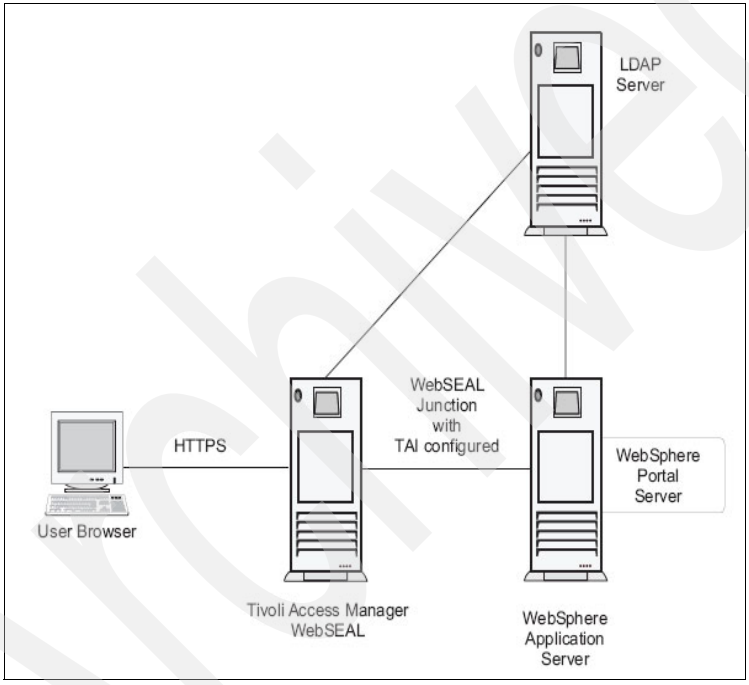

Figure 6-34 Components of the SSO solution to WebSphere Portal

Like the other integrations using Access Manager to facilitate SSO, this integration uses WebSEAL as a reverse proxy in front of the WebSphere Portal server.

The architecture shown in Figure 6-34 on page 189 supports the following process for achieving SSO. It should be noted that the *Trust Association Interceptor* approach is used to establish trust between the WebSEAL server and the WebSphere Application Server, into which WebSphere Portal is deployed, thereby establishing a trust relationship with the portal as well:

1. A client accesses the WebSphere Portal server corporate public /portal page through an Access Manager WebSEAL junction.
2. The user clicks the login link and is redirected to the secured WebSphere Portal server personalized /myportal page using SSL.
3. WebSEAL intercepts the request, then authenticates and authorizes the user.
4. Trust Association Intercepter (TAI) has been configured between WebSEAL and WebSphere Application Server, so the WebSphere Application Server accepts the user from WebSEAL based on the data contained in the iv-user HTTP header. The WebSphere Portal server personalized page is displayed.
5. When the user clicks the logoff link, the user is taken back to the corporate public /portal page.

As with the other Access Manager SSO integrations, this integration involves configuring WebSEAL and the WebSphere Portal such that WebSEAL performs the authentication of requests to the portal. This means that WebSEAL and portal share the same user registry, so that any existing users as well as new users avoid the need for reauthentication to WebSphere.

Importing users

The first step is to import any existing users into the Access Manager user registry. In addition, the portal administrative users (wpsadmin, wpsbind, and the portal administrators group wpsadmins) must also be imported into the Access Manager registry.

Configuring Trust Association Interceptor

Once this import is done, we configure the TAI so that any users authenticated by WebSEAL will be trusted by the WebSphere Application Server without re-authentication.

The first step is to create a trusted user account (this account should be used only for the TAI). Run the following commands:

```
pdadmin> user create trusted_user_ID user_DN cn sn password
pdadmin> user modify trusted_user_ID account-valid yes
```

For example:

```
pdadmin> user create tai_trusted_user cn=tai_trusted_user,o=acme,c=au
tai trusted_user xxxxxx
pdadmin> user modify tai_trusted_user account-valid yes
```

Modifying WebSphere Application Server properties

Next, it is necessary to modify two of the WebSphere Application Server properties files, the *trustedservers.properties* and the *Webseal.properties* files, to enable the TAI feature.

- trustedservers.properties

  Uncomment the options shown below in **bold**:

  ```
  #Use this property to specify the types of reverse proxy servers
  #that will be loaded at run time
  #com.ibm.Websphere.security.trustassociation.types=Webseal For each
  #type of reverse proxy servers specified in
  #com.ibm.Websphere.security.trustassociation.types, specify the
  #class file that implements the associated #interceptor for it.
  com.ibm.Websphere.security.trustassociation.Webseal.interceptor=com.
  ibm.ws.security.Web.WebSealTrustAssociationInterceptor

  #Optionally, specify a properties file for any of the reverse proxy
  #servers type specified above. The properties file must end with
  #".properties". For example, Webseal36.properties. However, do not
  #include this extension as shown below. Moreover, you can only do
  #this if the interceptor class extends
  #WebSphereBaseTrustAssociationInterceptor and both init() and
  #cleanup() methods were implemented. The init() method should read
  #and parse the contents of the properties file.
  com.ibm.Websphere.security.trustassociation.Webseal.config=Webseal
  ```

- Webseal.properties

  Uncomment the options shown below, and follow the comments within the configuration file to configure the value of each option:

  ```
  #Uncomment and use this property to specify the header name(s) you
  #expect to exist in the HTTP Request.
  com.ibm.Websphere.security.Webseal.id=iv-user,iv-creds

  #Uncomment and use this property to specify where you expect the
  #WebSeal server(s) to be.
  com.ibm.Websphere.security.Webseal.hostnames=Webseal_fqdn,
  Webseal_hostname
  ```

```
#Uncomment and use this property to specify the port(s) from which
#the WebSeal server(s) receive user requests.
com.ibm.Websphere.security.Webseal.ports=443,80

#For WebSeal 3.71, if basic login (user name/password) is used,
#uncomment and use this property to specify the ID that the Webseal
#server must use to validate trust with the interceptor. NOTE: For
#WebSeal 3.6, do not uncomment this line.
com.ibm.Websphere.security.Webseal.loginId=trusted_user_id
```

Enable WebSphere Application Server security

The next step is to enable WebSphere security. To enable the Trust Association Interceptor, the WebSphere Application Server security must be enabled and configured against the LDAP registry that Tivoli Access Manager is configured to use.

Perform the following steps to achieve this task:

1. Open the WebSphere Administrative Console (`http://<hostname>:9060/admin`).
2. Select **Security Center** from the Console menu.
3. On the General tab, click **Enable Security** and click **Apply**.
4. Click the **Authentication** tab and perform the following actions:
   a. Select **Lightweight Third Party Authentication (LTPA)** as the Authentication Mechanism.
   b. Select the **Enable Single Sign On (SSO)** check box and supply your Internet DNS domain in the Domain field.
   c. Select the **Enable Web trust association** check box.
   d. Select the **LDAP** radio button.
   e. Enter an LDAP user DN in the Security Server ID field (for example, cn=wasadmin,o=acme,c=au).
   f. Enter the password of the chosen LDAP user DN in the Security Server Password field.
   g. Type the host name of the LDAP server used by the Tivoli Access Manager in the Host field.
   h. For Directory Type, select **custom**.
   i. Select the **Advanced** button. In the LDAP Advanced Properties window, update the following fields:

      ```
      User Filter:
      (&(uid=%v)(objectclass=inetOrgPerson))
      ```

```
Group Filter:
(&(cn=%v)(|(objectclass=groupOfNames)(objectclass=
groupOfUniqueNames)))
Group Member ID Map:
groupOfNames:member;groupOfUniqueNames:uniqueMember Select the OK
button.
```

j. Specify the port that the LDAP server is listening on (typically 389).

k. Enter the DN suffix in the Base Distinguished Name field (for example, o=acme,c=au).

l. Enter the LDAP bind DN in the Bind Distinguished Name field.

> **Note:** The Tivoli Access Manager configuration secures the LDAP namespace by modifying the LDAP access control list of all suffixes that are defined. This may include the suffixes used by the WebSphere Application Server to store users and groups. To avoid problems when the WebSphere Application Server searches for users, this LDAP bind DN and password must have sufficient access rights within the directory (at least under the subtree specified by the base DN) to perform searches in both the user and group subtrees within the directory.

m. Enter the LDAP bind DN password in the Bind Password field.

5. Click **Apply**.
6. Click **OK**.
7. If you are using SSL, add a host alias for WebSphere to intercept SSL connections as follows:
   a. Click **Virtual Hosts**.
   b. Under Aliases, click **Add**.
   c. Enter the SSL port number (typically 443) as follows: **:ssl_port_number* (for example, *:443).
   d. Click **Apply**.
8. If you are using SSL, regenerate the Web server plug-in as follows:
   a. Expand **Nodes**.
   b. Right-click the WebSphere server name.
   c. Click **Regen Webserver Plugin** on the context menu.
   d. Restart the Webserver used with WebSphere.
9. Close the Administrative Console.
10. Restart WebSphere.

The next time the Administrative Console is started, a logon box is displayed. To open the console, supply the user name and password of the Security Server ID, as specified in the above steps.

Creating WebSEAL junctions to WebSphere

Two WebSEAL junctions need to be created, one using TCP and the other using SSL. The TCP junction is for unauthenticated access to public content, while the SSL junction provides secured access to personalized portal content (for example, MyTamco pages).

WebSEAL configuration to facilitate filtering of portal content

To allow WebSEAL to filter the content from WebSphere Portal correctly, the following WebSEAL options must be updated:

1. Locate the WebSEAL configuration file, Webseald.conf, located in the directory WebSEAL_install_dir\etc\.
2. Within the [forms] stanza, set the following option:

   ```
   [forms] forms-auth = https
   ```

3. Within the [server] stanza, set the following option:

   ```
   [server] dynurl-allow-large-posts = yes
   ```

4. Within the [script-filtering] stanza, set the following option:

   ```
   [script-filtering] script-filter = yes
   ```

Note: Tivoli Access Manager supports multiple secure domains, allowing multiple WebSEAL servers within different secure domains. When modifying the WebSEAL configuration file, the file name is formatted as Webseald-domain.conf (for example, Webseald-*default*.conf, where *default* is the name of the default Tivoli Access Manager domain).

Configuring WebSphere Portal logout pages

On successful user logon to WebSEAL and, in turn, single sign-on (SSO) to WebSphere Portal, session cookies from WebSphere Portal are stored in the browser's memory. Unless the user physically closes the browser, the session between the browser and WebSphere Portal remains open, even if the user logs out of Tivoli Access Manager. If another user were to re-use the same browser window to log on to WebSEAL and access WebSphere Portal, WebSphere Portal might assume that the new user was the same as the previous user.

To solve this problem, the integration package includes two HTML logout pages:

1. wpslogout.html

   This file redirects the user back to the WebSphere Portal corporate public /portal page. It is designed specifically for WebSphere Portal.

   To configure wpslogout.html, follow these steps:

   a. Locate the wpslogout.html file in the integration package. Copy this file to the directory WebSEAL_install_dir\www\lib\html\locale\.

   b. Modify the new wpslogout.html file, updating the Webseal_jctname variable to the value of the WebSEAL TCP junction name that you created for WebSphere Portal. For example, if the TCP junction name is /wps_tcp_jct, update the variable as follows:

   ```
   // IBM Tivoli Access Manager Integration
   // for WebSphere Portal Server
   //
   // Update the Webseal_jctname variable below with the value of
   the
   // WebSEAL TCP junction name that you created for the WebSphere
   Portal // Server. For example, if the TCP junction name is
   "/wps_tcp_jct",
   // update the variable as:
   // Webseal_jctname = "/wps_tcp_jct";
   Webseal_jctname = "/wps_tcp_jct"; logout.html
   ```

   Note: WebSEAL returns the logout.html page when a user logs out using the pkmslogout function. This function is not available when WebSEAL has been configured to use basic authentication (BA). Hence, this section is not relevant if WebSEAL is set to BA mode.

2. logout.html

   This file is similar to the logout.html file that is supplied with the default WebSEAL installation. It includes JavaScript that erases the predefined browser session cookie from WebSphere Portal upon logout from WebSEAL. The logout.html page is designed to take the user back to the regular Tivoli Access Manager WebSEAL page.

   To configure the WebSEAL logout page with the required JavaScript:

   a. Locate the logout.html file in the integration package.

   b. Use this logout.html file to overwrite the existing logout.html file in the directory WebSEAL_install_dir\www-domain\lib\html\locale\.

   c. Restart the WebSEAL server.

If you have previously modified the default logout.html file, you can still add the required JavaScript functionality without abandoning your modifications. To do this:

a. Use an editor to open the logout.html file in the integration package.

b. Copy the script element (that is, the code contained by the script start and end tags):

```
<script language="Javascript">
....
</script>
```

c. Use an editor to open your existing logout.html file in the directory WebSEAL_install_dir\www-domain\lib\html\locale\.

d. Paste the code from step 2 above before the HTML tag:

```
In the body tag, add the parameter:
onLoad=delete_all_cookies('/',exception_list) For example: body
bgcolor=#FFFFFF text=#000000
onLoad=delete_all_cookies('/',exception_list)
```

e. Save the file.

f. Restart WebSEAL.

Securing WebSphere Portal resources

In order to secure the WebSphere Portal personalized and administrative portal pages, you must create a number of Access Manager access control lists (ACLs) and a protected object policy (POP). Table 6-1 and Table 6-2 on page 197 show summaries of the required ACLs and POP.

Table 6-1 Required ACLs for WebSphere Portal

ACL	Description
WPS_Admins_only	Restrict access to WebSphere Portal configuration to designated administrative users.
WPS_No_Access	Block access to WebSphere Portal personalized portal pages through the TCP junction.
WPS_Unauthenticated_view	Allow unauthenticated access to the corporate public portal page.
WPS_Authenticated_view	Restrict access to WebSphere Portal personalized portal pages to Access Manager authenticated users.

Table 6-2 Required POPs for WebSphere Portal

POP	Description
SSL_only	Restrict access to WebSphere Portal personalized pages to SSL connections only.

To create and apply these ACLs and POP, perform the following steps:

1. Locate the Webseal_wps_sso_acls.pdadmin file in the integration package.
2. In a command window, log in to pdadmin as the security administrator and pipe in the file. For example:

```
pdadmin –a sec_master –p password integration_files_dir\
Webseal_wps_sso_acls.pdadmin
```

The ACLs and POP created above must now be attached to specific locations in the Access Manager protected object space. Enter the following commands in pdadmin, modifying the host name and junction names appropriately:

1. In a command window, log in to pdadmin as the security administrator.
2. Enter the following commands in sequence (each command is entered as one line):

```
pdadmin> acl attach /WebSEAL/hostname/wps_tcp_jct
WPS_Unauthenticated_view
pdadmin> acl attach /WebSEAL/hostname/wps_tcp_jct/wps/config
WPS_Admins_only
pdadmin> acl attach /WebSEAL/hostname/wps_tcp_jct/wps/myportal
WPS_No_access
pdadmin> acl attach
/WebSEAL/hostname/wps_tcp_jct/wps/doc/en/infocenter/help
WPS_Unauthenticated_view
pdadmin> acl attach /WebSEAL/hostname/wps/doc/en/infocenter/help
WPS_Unauthenticated_view
pdadmin> acl attach /WebSEAL/hostname/wps_ssl_jct
WPS_Unauthenticated_view
pdadmin> acl attach /WebSEAL/hostname/wps_ssl_jct/wps/config
WPS_Admins_only
pdadmin> acl attach /WebSEAL/hostname/wps_ssl_jct/wps/myportal
WPS_Authenticated_view
pdadmin> pop attach /WebSEAL/hostname/wps_ssl_jct/wps/myportal
SSL_only
```

Configure WebSphere Portal login/logout functions

The final two steps in the integration require modifying the WebSphere Portal login.jsp file so that it redirects the user from the corporate public /portal page to the secured personalized /myportal page, where the user is asked to authenticate to Access Manager. To do this task, follow these steps:

1. Back up the original login.jsp file located in the directory WebSphere_Portal_Server_install_dir\app\wps.ear\wps.war\screens\html\.

1. Locate the login.jsp file in the integration package.
2. Use this login.jsp file to overwrite the existing login.jsp file.
3. Modify the login.jsp file, updating the Webseal_jctname variable to the value of the WebSEAL SSL junction name that you created for the WebSphere Portal. For example, if the SSL junction name is /wps_ssl_jct, update the variable as follows:

```
// IBM Tivoli Access Manager Integration
//
// for WebSphere Portal Server //
// Update the Webseal_jctname variable below with the value of
/ the WebSEAL SSL junction name that you created for the WebSphere
Portal
// Server. For example, if the SSL junction name is "/wps_ssl_jct",
// update the variable as:
//    Webseal_jctname = "/wps_ssl_jct";
//
Webseal_jctname = "/wps_ssl_jct";
```

The ErrorNotLoggedOut.jsp file provided by WebSphere Portal needs to be replaced with the one provided in the integration package. This file is used to return the user to the previous file in the browser history following an inactivity timeout. To do this task, follow these steps:

1. Back up the original ErrorNotLoggedOut.jsp file located in the directory WebSphere_Portal_Server_install_dir\app\wps.ear\wps.war\screens\html\.

1. Locate the ErrorNotLoggedOut.jsp file in the integration package.
2. Use this ErrorNotLoggedOut.jsp file to overwrite the existing ErrorNotLoggedOut.jsp file.

The ConfigService.properties file also needs to be modified so that the WebSphere Portal logout calls the WebSEAL pkmslogout function and redirects the user to wpslogout.html. The wpslogout.html then removes all the session cookies and takes the user back to the WebSphere Portal corporate public /portal page.

To do this task, follow these steps:

1. Open WebSphere_install_dir\lib\app\config\services\ConfigService.properties in an editor.
2. Modify the following lines:

```
redirect.logout = true
redirect.logout.ssl = true
redirect.logout.url =
https://Webseal_fqdn/pkmslogout?filename=wpslogout.html
```

Testing the integration

To test the integration, follow these steps:

1. Open a browser and access the WebSphere Portal corporate public /portal page through WebSEAL with the following URL:

   ```
   http://WebSEAL_hostname/tcp_junction_name/WebSphere_Portal_Server_public_path
   ```

   For example, `http://www.Webseal.acme.com/wps_tcp_jct/wps/portal`.
2. Click the login link and notice how you are redirected to the WebSphere Portal personalized /myportal page through the WebSEAL SSL junction.
3. An authentication comes from the Access Manager WebSEAL server. Log in using the Access Manager user ID and password.
4. Upon successful authentication, the WebSphere Portal personalized /myportal page should be displayed without any WebSphere Portal-specific authentication.
5. Click the logoff link. Notice that you are taken back to the WebSphere Portal corporate public /portal page that you entered in step 1.

At this point you have completed the integration and Access Manager WebSEAL is providing SSO functionality for WebSphere Portal.

6.4 Configuring TAMCO single sign-on across domains

Single sign-on across domains is an area that is growing in significance, largely spurred by the popularity of the Web services revolution. Access Manager for e-business includes toolkits with a number of design alternatives for customers who want to implement cross domain single sign-on, but who do not have a need for the management capabilities and the breadth of protocol and token type coverage that Tivoli Federated Identity Manager provides. These include cross-domain single sign-on, e-community single sign-on, failover cookie, and SMS.

In a segmented organization like TAMCO, we have a requirement to configure the WebSEAL servers with a mechanism to allow single sign-on across the Access Manager domains. In TAMCO, the Access Manager domains are also identified by unique DNS domains. This kind of domain crossing depends on trust between the domains because one domain needs to accept the authenticated entities being passed from another domain.

The ability for a user to access resources in a secure domain depends on the user acquiring a credential in that domain. Normally a credential is built after the user authenticates. In the cross-domain environment, some other way has to be found for WebSEAL to build a credential for the user.

In TAMCO, we are leveraging the *failover cookie* to accomplish this task because of its simplistic design and support for virtual host junctions. As the TAMCO infrastructure grows, the strategic cross-domain solution is to leverage SMS.

6.4.1 TAMCO failover cookie requirements

The only way to determine the correct cross-domain settings is to identify the session and failover requirements.

TAMCO identified that a user's session should not be valid for longer than eight hours across all applications. Since we are utilizing the failover cookie, this process will need to ensure that the user's session lifetime gets passed to subsequent servers as a user traverses between one tamco.com domain to another.

For Example, a typical user pattern during an average workday might look like this:

- 8:00 am - www.peoplesoft.fi.tamco.tamco.com
- 9:00 am - my.tamco.com
- 10:30am - www.finance.fi.tamco.tamco.com

- 12:30 pm - www.siebel.ge.tamco.tamco.com
- 4:01 pm - my.tamco.com

Based on TAMCO's requirements, the user should be challenged for login one time at 8:00 am and not rechallenged again until 4:01 pm. Notice that there are several failover transactions that must occur as the user browses across different WebSEALs.

Here are the basic WebSEAL settings needed to support this design:

1. The failover cookie uses the Web-host-name to determine the cookie domain (These directives can be found in the Webseald-default.conf file.):

   ```
   [server]
   Web-host-name = www.tamco.com
   ```

2. Failover cookies should only be sent over https:

   ```
   [failover]
   failover-auth = https
   ```

3. Generate a new failover key (one time only):

   ```
   $ cp -p /opt/pdWeb/etc/cdsso.keys /opt/pdWeb/etc/cdsso.keys.orig
   $ /opt/pdWebrte/bin/cdsso_key_gen /opt/pdWeb/etc/cdsso.keys
   ```

4. Copy this new key to all WebSEAL servers in the SSO domain (including FI, GE, and UK).

5. Change cdsso.keys permissions/ownership:

   ```
   $ chmod 644 /opt/pdWeb/etc/cdsso.keys
   $ chown  root:system /opt/pdWeb/etc/cdsso.keys
   ```

6. Load the failover encryption library, configure keyfile, enable domain, and cookie lifetime (8 hours by minutes):

   ```
   [authentication-mechanisms]
   failover-password = failoverauthn.a

   [failover]
   failover-cookies-keyfile = /opt/pdWeb/etc/cdsso.keys
   failover-cookie-lifetime = 480
   enable-failover-cookie-for-domain = yes
   failover-require-lifetime-timestamp-validation = no
   failover-require-activity-timestamp-validation = no
   failover-include-session-id = no
   ```

7. When a user authenticates, WebSEAL tracks the age or lifetime of the user entry in the session cache. When the current system time exceeds the timestamp value, WebSEAL invalidates the user's entry in the session cache (including the user credentials) and will force a reauthentication transaction on the page request.

   In addition, WebSEAL can be configured to add the session lifetime timestamp to the failover cookie. This effectively preserves the session lifetime across failover events.

   Note: The successful use of this feature is dependent on synchronization of clocks between WebSEAL servers. If clock skew becomes great, sessions will expire.

   Here are the attributes to use:

```
[failover-add-attributes]
session-lifetime-timestamp = add

[failover-restore-attributes]
session-lifetime-timestamp = preserve

8. Be sure that the failover-cookie-lifetime corresponds to the
session timeout (8 hours by seconds).

[session]
timeout = 28800
```

There are many variations to consider when making the failover cookie active in a global environment. These settings are discussed in detail in the *IBM Tivoli Access Manager for e-business Version 6.0 WebSEAL Administration Guide,* SC32-1687.

Figure 6-35 on page 203 depicts the TAMCO cross-domain WebSEAL configuration with the virtual host junctions listed in each geography. The failover cookie name is PD-ID and can be found at the domain level of tamco.com. Since each of the TAMCO DNS domains are part of tamco.com, the failover cookie can be leveraged to provide SSO across the domains. The failover cookie would not be able to support environments hosting many different DNS domain names.

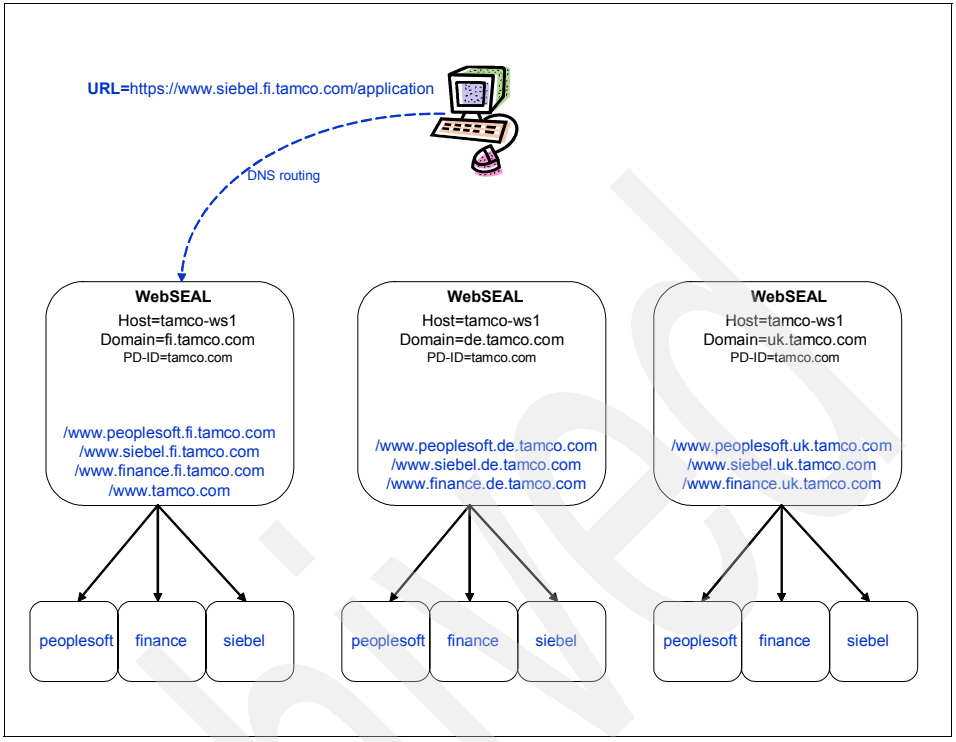

Figure 6-35 Virtual host junctions with failover cookie

6.5 Configuration for scalability and high availability

As noted in 4.2.6, “Availability and scalability” on page 56, TAMCO is using a hardware-based load-balancing appliance to meet their availability requirements. While the hardware appliance does the greatest share of the load balancing, the Access Manager components also contribute load balancing techniques. This section discusses the configuration of the Access Manager components to work with the hardware-based load balancing solution.

The TAMCO environment can be considered a typical three-tier architecture (see deployment architecture in Figure 4-11 on page 68). As such, the load-balancing requirements to achieve TAMCO’s high availability requirements encompass three main considerations:

- To provide reliability and fault tolerance in case of a single server failure in a clustered configuration.

- To provide a mechanism to spread client traffic load in a manner that preserves performance across multiple destination servers.
- To provide a mechanism by which a deployment may be scaled through the addition of servers without the need for downtime in the solution.

It is important to understand the traffic flow associated with the deployment of the TAMCO Access Manager solution. There are three protocols, LDAP, authorization communications, and HTTP, that are the source of the primary traffic load. Figure 6-36 shows the main flow patterns in a way that makes it easier to see than in the deployment architecture in Figure 4-11 on page 68.

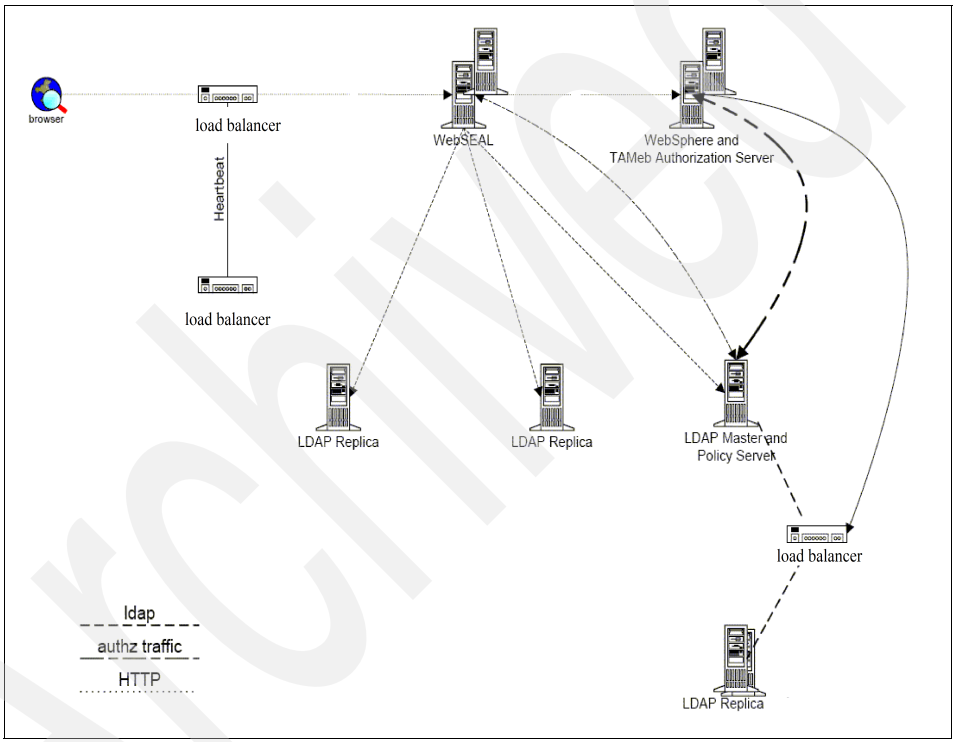

Figure 6-36 Traffic load communication protocol flow in TAMCO deployment

HTTP traffic shows the path for HTTP traffic through the environment. This traffic is secured by SSL, which has implications for load-balancing designs in the environment, because typically the use of SSL has some impact on the establishment of a session and session flow.

LDAP traffic represents the connections that are made between LDAP clients and LDAP servers within the environment. The TAMCO environment has one master LDAP server and one replica. This traffic is also secured by SSL.

Authorization traffic represents the authorization policy database replication performed by Access Manager to minimize the impact of updates to the database. The content of this traffic is proprietary and secured by SSL.

In an enterprise system, high availability should be applied as follows:

- Between the load balancer and WebSEAL
- Between the LDAP master and replica and WebSEAL
- Between WebSEAL and WebSphere Application Server
- Between WebSphere Application Server and the LDAP master

However, the TAMCO environment is not as demanding as this.

6.5.1 Load balancing within the environment

Load-balancing client HTTP requests in the TAMCO case uses a dedicated hardware load balancer with *stickiness*[2] based on client IP address.

In configuring the front-end cluster on the load balancer, the preferred forwarding method is Media Access Control (MAC) packet forwarding. MAC is light, efficient, and well proven as a technology. However, it is limited by the requirement that the load balancer servers and the destination servers be configured on the same subnet.

With MAC forwarding, the dispatcher load balances the incoming request to the selected server and the server returns the response directly to the client without any involvement of the dispatcher. With this forwarding method, the dispatcher only looks at the inbound client-to-server flows. It does not need to see the outbound server-to-client flows. For MAC forwarding, the load balanced servers such as WebSEAL must be configured with the cluster address aliased on the loopback adapter.

Back-end HTTP load balancing is performed by a combination of WebSEAL and the load balancer. The load balancer uses connection pooling and WebSEAL implements junction load balancing using a least-busy arbitration among multiple application servers on a single junction. Session affinity is not required with the PeopleSoft application server. Do not use non-terminating SSL load balancing.

Directory Server load balancing uses connection pooling, which provides the means by which Access Manager components can perform load balancing with the Directory without having to rely on a front-end load balancer appliance.

[2] The term *stickiness* refers to a method used by load balancers to keep track of request route information within a given session, in particular to which servers to route specific requests.

Any component that leverages Access Manager will load balance (in accordance with the priority set within the configuration) across the configured directory infrastructure for Directory Server requests. This includes WebSEAL, the Access Manager Authorization Server (pdacld), and the Web server plug-ins (WebPI).

The configuration file that manages this Directory Server configuration has an entry within the [aznapi-configuration] stanza that defines the replicas. This configuration entry is called *ldap-replica*. It is possible to configure multiple Directory Servers within this stanza, either as read-only or as writable Directory Servers (called peers). The software will handle the load balancing and failover of Directory Server connections as well as refreshing the context when required.

6.5.2 Access Manager Policy Server load balancing

Figure 6-36 on page 204 shows the typical Access Manager deployment with a load balancer.

As noted above, Access Manager itself manages the accessibility of the Directory Server master and replica infrastructure. The TAMCO directory architecture does not require any separate dedicated load balancers for the Directory Servers. It can rely on the Access Manager innate load balancing. However, a load balancer can be used to assist in the configuration of an active/passive Access Manager Policy Server configuration to provide failover availability.

A replica of the policy server can be created such that on encountering unrecoverable problems with the Access Manager Policy Server, it can be booted, and switched to by the load balancer for all administration requests. There is no notion of active-active configurations of the policy server, where there are two policy servers configured in a domain, hence the active-standby is the only configuration alternative.

6.6 Conclusion

At the outset, TAMCO determined that the following value would be added to their business by adopting a centralized authentication and authorization framework. In particular, through an in-depth security architecture analysis in conjunction with their general IT modernization effort, TAMCO identified the following needs for improving their business efficacy, efficiency, and security:

- Augment sales turnaround by allowing TAMCO to extend their IT environment more rapidly to upgrade to better sales and marketing applications without having to redo security infrastructure.

- Improve time-to-market by relieving the small team of TAMCO developers from having to write security into applications.
- Reduce the overall risk TAMCO would incur by adopting the Web-based new IT strategy.
- Provide a strong story to regulators and TAMCO insurers with respect to their use of best security practices for authorization and auditing.
- Ease the task of extending their IT environment to include IT environments of suppliers and future acquisition partners.

Security capabilities that were identified and proven in the Proof of Concept were:

- Single sign-on to the TAMCO portal and back-end applications
- Fine-grained authorization capability for portlets and JSPs at not only the URI level but also at the method level
- A standards-based common auditing system plus the ability to generate standard and custom reports
- Consistent application of access control policy across the TAMCO business units

The adoption and implementation of an IBM Tivoli Access Manager V6.0 security architecture, as described in this document, has proven to satisfy these requirements, and provided these key value-adds to the TAMCO business:

- Optimized user experience: Secure Web single sign-on (single domain or cross-domain), personalization, self care, and so on
- Tivoli world-class management/administration: Web-based delegatable administration of users and policies, integrated customer self-care, and unified view of protected object namespace
- Breadth of coverage: Ranges across:
  - IBM SWG portfolio: Web SSO and common security model across WebSphere, MQSeries®, Domino, and OS/390® with RACF® integration
  - The key third-party Web vendors, including Siebel, SAP®, BEA, BroadVision, Plumtree, Epicentric, PeopleSoft, Vignette, Oracle®, KANA, i2, and Ariba
- Time to value, time to market for Web application development: A security services architecture built on standards-based technology (for example, J2EE™), supporting the ability to separate application logic and security

A

Statement of work

This appendix provides a sample of what you might include in your *statement of work* contract.

Building a security infrastructure solution

The content of the Statement of Work should include activities to do the following:

- Assist in the design and implementation of a security infrastructure based on IBM Tivoli Access Manager.
- Install and configure Access Manager for e-business in a development and pilot environments.
- Integrate a Web application sampling to the extent necessary to demonstrate product functionality.

Executive summary

This Statement of Work describes the services to assist the customer with installing and configuring IBM Tivoli Access Manager (Access Manager for e-business) in a development environment and demonstrating its functionality through a pilot release.

The *IBM Business Partner* proposes to leverage IBM Global Services Method as the center point of the project development approach. The method uses work products to define “what” are the project artifacts produced during the project lifetime. The method also uses processes to define “how” the work products are produced in a timely manner, and “how” phases, activities, and tasks of the project are performed.

The activities defined in the SOW as well as the project estimates are based upon information that is currently available regarding the customer's business needs and goals for this project. Progressive elaboration of the details of these business needs and goals is inherent to this type of project and may result in a need to modify estimated tasks duration and costs or add new tasks in accordance with defined project change control procedures.

The *IBM Business Partner* is pleased to present the attached Statement of Work to assist the customer in this endeavor. Based on our understanding of the *customer's* needs, the *IBM Business Partner* is confident that its approach, qualifications, and experience will best address your concerns.

Project scope

The *customer* has expressed a desire to implement the IBM Tivoli Access Manager for e-business. Due to the importance and sensitivity of the data handled by the Access Manager for e-business application, the *customer* has requested assistance in installing and configuring this new solution. Therefore, the *IBM Business Partner* proposes a plan to complete the product installation and configuration, along with a solution design, which will provide the *customer* with a design for utilizing Access Manager for e-business in controlling access to the corporate Web space.

This Statement of Work describes the services to assist *the customer* with installing and configuring Access Manager for e-business first in a development environment in order to demonstrate its functionality through a pilot release in a production environment. This Statement of Work also covers the migration of one Web application to the Access Manager for e-business production environment.

This document seeks to provide a view of the solution vision that is common and agreed to by all of the above. It does not address detailed requirements, solution architecture/designs, or deployment plans, other than at a high level. Such specifics will be discussed within other documents, as appropriate.

At the customer's request, the project scope has been divided into four phases (A through D). The phases are:

A. Planning

B. Design: Architecture and Documentation

C. Development and Pilot Deployment

D. Production Deployment

We take a team approach to delivering our services. The team will be comprised of both *customer* and the *IBM Business Partner* personnel. During the process, we will keep you informed of the progress of our work through frequent status meetings and continuous interaction with the *customer*.

At the conclusion of each phase, the *IBM Business Partner* will complete a Value Assessment. The Value Assessment will consist of a review of the Phase Deliverables between the *IBM Business Partner* and the *customer* Project Manager for acceptance and concurrence to proceed with the next Phase.

Other considerations for inclusion in the Project Scope are:

- Assess *the IBM Business Partner*'s computing environment to prepare for the implementation of Access Manager for e-business.
- Assist the customer with the definition and creation of the domain and objectspace policies.
- Provide (remote) guidance to the customer project team during intermediate deployment.
- Provide product training.
- Add here whatever is offered to the customer.

Additionally, a description of the different phases of the implementation project if applicable.

Assumptions

The statement of work should also include all assumptions. A few key assumptions you may want to consider for an Access Manager for e-business deployment are:

1. The *customer* will make personnel available for facilitated sessions and meetings as required.
2. The *customer* personnel who will be assigned to this project will have the technical skills necessary to participate in this project.
3. Information Technology (IT) and user personnel will be available as described in the *customer* responsibilities.
4. IBM Tivoli Access Manager for e-business <Current Release> will be used. The Access Manager Policy Server, IBM Directory Server and WebSEAL reverse proxy server will be implemented on Linux <or platform chosen>.
5. Tivoli Access Manager for e-business will be deployed using the bundled components using Access Manager for e-business's version of WebSphere Application Server and Tivoli Directory Server.
6. The *customer* will supply all required hardware for the Tivoli Access Manager installation and configuration.
7. The *customer* will supply all the required software for the Tivoli Access Manager installation and configuration
8. The security infrastructure will be accessed by only employees. Customers, partners, and other user types are out of scope during this engagement.

9. The *customer* is required to provide SSL server certificates, as required, within the environment.
10. Tivoli Access Manager installation will initially take place in a development environment in order to demonstrate functionality through a pilot release.
11. The security infrastructure deployed in a development environment does not require formal change management procedures.
12. User names and passwords will be used to authenticate users to Access Manager.
13. In the development environment, Access Manager will be installed in single server environments and will not require high availability.
14. The Web applications to be integrated with Access Manager support basic authentication or forms-based login and therefore will support single sign-on from Access Manager.
15. Integration of Access Manager with the selected intranet Web application environment during the pilot phase will be limited to single sign-on integration.
16. Integration of Access Manager with the selected intranet Web application in the production environment will be limited to course-grain access control.
17. Work will be performed at the *customer's* facility in and some work may be performed at IBM locations.
18. The *customer* will provide services under this statement of work during normal business hours, Monday through Friday, excluding holidays.

In addition to the above assumptions that affect the entire project, assumptions specific to each activity/phase of the project should be included within the section of this statement of work describing each activity.

Note: Insert any additional assumption about specific security issues that the customer has here.

IBM Business Partner responsibilities

The *IBM Business Partner* responsibilities may be broken into two or more sections. Project Management and Solution Implementation recommended tasks are listed here. In addition, the *IBM Business Partner* might also be responsible for tasks such as purchasing software and hardware, general consulting, and negotiating financing options with the *customer*.

Project management

The *IBM Business Partner* will provide project management for the *IBM Business Partner* responsibilities in this Statement of Work. The objective is to establish a framework for project communications, reporting, procedural, and contractual activity. The *IBM Business Partner* Project Manager will be responsible for this task.

The following subtasks will be performed:

- Be the primary *IBM Business Partner* liaison with the *customer* Project Manager.
- Prepare a project plan, which identifies and assigns tasks to both *IBM Business Partner* and *customer* project participants, identifies major milestones for the efforts of the project team, identifies estimated dates on which they occur, and indicates critical path.
- Review the Statement of Work, project plan, and the contractual responsibilities of both parties with *customer* Project Manager and project team.
- Review areas of risk and containment plans with the *customer* Project Manager.
- Maintain regular project communications with the designated *customer* Project Manager.
- Measure and evaluate progress against the Project Plan.
- Resolve deviations from the Project Plan.
- Implement the Change Control Procedure in conjunction with *customer* Project Manager.
- Coordinate and manage the technical activities of *IBM Business Partner* project personnel.

Solution implementation

The *IBM Business Partner* has the following solution implementation responsibilities:

- Install and configure IBM Tivoli Access Manager for e-business V6.0 servers, including all prerequisite software and networking environment.
- Install and configure additional components according to the Project Plan.
- Integrate Web or custom applications with Access Manager for e-business.
- Provide Access Manager for e-business solution documentation.
- Perform stress testing and tuning (if needed).

Customer responsibilities

The successful completion of the implementation also depends on the customer's participation and full commitment. This section therefore should include customer responsibilities as precisely as possible.

A successful implementation project is predicated upon the following customer responsibilities.

Project management

Prior to the start of a statement of work, a designated person from the customer must be assigned. This designated representative or Project Manager will be the focal point for all communication with the IBM Business Partner relative to this project and who will have the authority to act on the customer's behalf in matters regarding this project. His or her responsibilities include:

- Managing the customer's personnel and responsibilities for the project
- Serving as the interface between IBM and all customer departments participating in the project
- Participating in project status meetings
- Obtaining and providing information, data, and decisions
- Resolving deviations from the estimated schedule, project plan, or statement of work
- Helping resolve project issues and escalating issues within the customer's organization as necessary

Other responsibilities

Within this section of the statement of work it should be documented that the customer's staff is available at the agreed time. Also, the customer needs to ensure that the staff has the appropriate skills and experience.

Accurate information is key for such projects. It should be agreed that all information disclosed to the IBM Business Partner will be true, accurate, and not misleading in any material respect.

It also has to be the customer's responsibility to make the final selection of the solution and technical architecture. Given this, all prerequisite hardware and software to be used during the project should be supplied by the customer.

Specific responsibilities could include:

- Retaining overall responsibility and ownership of the IBM Tivoli Access Manager Implementation
- Designating skilled operations personnel to work with the IBM Business Partner as appropriate in installation and testing of Access Manager for e-business
- Providing hardware and software pre-requisites required for setting up Access Manager for e-business development and production environments
- Providing required network connections between Access Manager for e-business server and customer Web applications
- Appointing technical personnel to participate in Access Manager for e-business education sessions as needed
- Providing all data and information required for implementation, such as organizational structure model, user directory structure definition, and Web application standards
- Providing suitable workspace with telephone access for the IBM Business Partner team while working on customer premises
- Providing user IDs, passwords, and IP addresses as required, which enables the IBM Business Partner to perform the service
- Providing information to allow estimates on current and future system workload and performance expectations

Laws, regulations, and statutes

The customer is responsible for the identification of, interpretation of, and compliance with any applicable laws, regulations, and statutes that affect the customer's applications or business.

Data file content and security

The customer must be responsible for the actual content of any data file, selection, and implementation of controls on its access and use, and security of the stored data.

Deliverable materials

The following items will be delivered to customer under this Statement of Work:

- Project Plan
- Biweekly Status Reports
- Customer Tivoli Access Manager Micro Solution Design

- Customer Tivoli Access Manager Solution Test Manual
- Customer Tivoli Access Manager Installation Manual
- Customer Tivoli Access Manager System Administration Guide
- Customer Tivoli Access Manager Stress Test Results (if applicable)

Completion criteria

You need to list the completion criteria here. You have to engage with the customer to get a proper sign-off of the project with an appropriate completion criteria.

The *IBM Business Partner* will have fulfilled its obligations under this Statement of Work when any of the following occurs:

- The *IBM Business Partner* accomplishes the tasks described in "IBM Business Partner responsibilities" on page 213, including delivery of the materials listed in "Deliverable materials" on page 216.
- The *IBM Business Partner* provides the number of hours specified in this Statement of Work or in any subsequent Change Authorization.
- The *customer* or *IBM Business Partner* terminates the Project in accordance with the provisions of the *IBM Business Partner* Customer Agreement.

You can also include specific issues and resolutions explicitly in the completion criteria. You have to be aware of these additional completion criteria for your customer.

Estimated schedule

The services to be performed in this Statement of Work are estimated to complete within <NWEEKS> weeks from the start of this Statement of Work, requiring a minimum of <number> full time *IBM Business Partner* consultants. Figure A-1 shows a sample project schedule.

ID	Task Name	Start	Finish	Duration
1	Perform customer interviews	6/19/2006	6/21/2006	3d
2	Prepare/execute demo	6/22/2006	6/29/2006	6d
3	Write up recommendation	6/30/2006	7/3/2006	2d
4	Close the engagement	7/4/2006	7/5/2006	2d

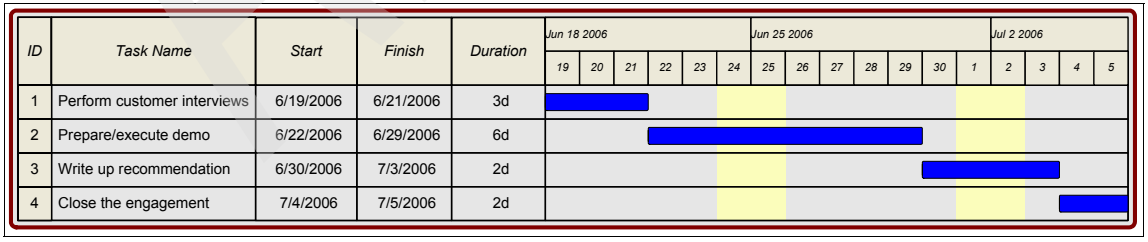

Figure A-1 Project schedule (sample)

Charges

This engagement will be conducted on a time and materials basis.

The *IBM Business Partner* will provide up to a total number of <NHOURS> hours for this Service at an hourly rate of $XXX.

The estimated professional services charges for this Statement of Work are $XXX and are exclusive of any travel and living expenses and any applicable taxes. This price does not include any hardware or software costs associated with the purchase of the customer's selected identity management solution.

The *customer* will be billed actual travel and living costs.

The hours specified above are the IBM Business Partner estimate based upon the information available at this time. The *IBM Business Partner* will notify the *customer* as soon as practical of any changes in its estimates.

B

Tips and tricks

In this appendix, we provide a few tips and tricks for common issues that may be encountered in an Access Manager for e-business installation. We also suggest some basic troubleshooting techniques and provide operational suggestions to improve your Access Manager for e-business environment. Be sure to review the Release Notes document, as it offers a section on the latest known limitations, problems, and workarounds.

The *IBM Tivoli Access Manager Problem Determination Guide*, SC32-1701 offers more detailed descriptions on how to tackle problematic situations.

Importing and managing certificates on WebSEAL

When configuring SSL junctions on WebSEAL to the back-end application servers, you must be sure to import the back-end CA root certificate into the WebSEAL gskit key database. Access Manager for e-business provides several known CA signer certificates (such as Verisign and Thawte); however, if you are using self-signed certificates on the back-end junctioned servers or if the CA-signed certificate is not included in the default list, then you will run into problems trying to create your junction.

Below we have identified the steps necessary to make sure that your back-end signer certificate is properly installed in the local WebSEAL key database:

1. Extract a copy of the junctioned or back-end server certificate (two methods for doing this are shown in “Method 1: HTTPS browser” on page 220 and “Method 2: IKEYMAN utility” on page 223).
2. FTP the certificate to all WebSEAL server(s).
3. Import the participating Web server root certificate into each WebSEAL's key database.
4. Restart each WebSEAL server.

Method 1: HTTPS browser

This is the simplest method and it works with all back-end server types.

For this example, we are using Internet Explorer; however, a similar process can be followed using Mozilla or other browsers. Using https, browse to the back-end host name (`https://www.hostname.com`) and double-click the padlock that appears near the lower right corner of the browser window. Select **View Certificates** and the Certificate window appears, as shown in Figure B-1 on page 221.

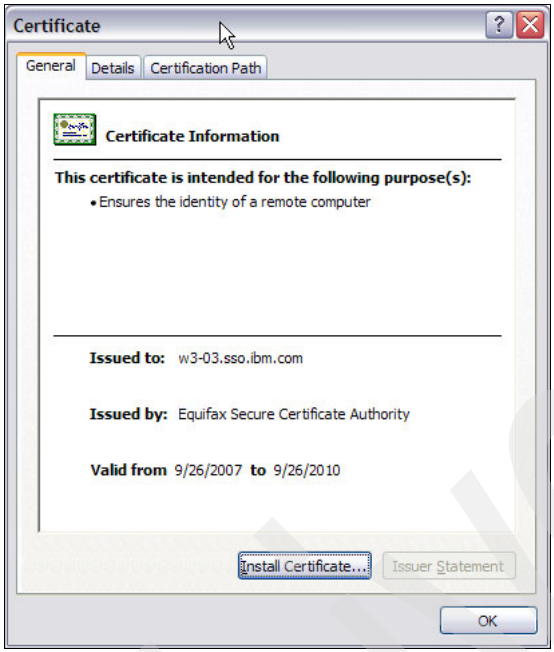

Figure B-1 Certificate information window

Under the Certification Path tab (shown in Figure B-2 on page 222), you can see the certificates available for export.

- Certificate List
  - Equifax Secure Certificate Authority
  - w3-03.sso.ibm.com

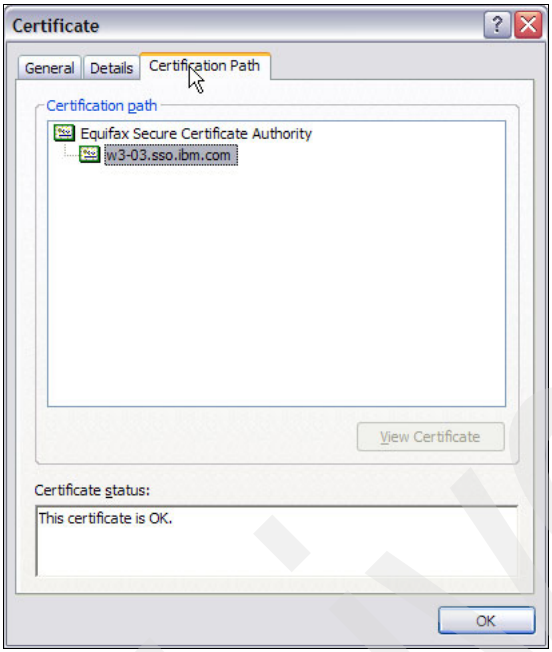

Figure B-2 Certification Path dialog

If there is only one certificate in the list, then this is normally a “self-signed certificate”. Once you have highlighted the parent certificate name in the Certification Path Window, click the **View Certificate** button and go to the Details tab, as shown in Figure B-3 on page 223.

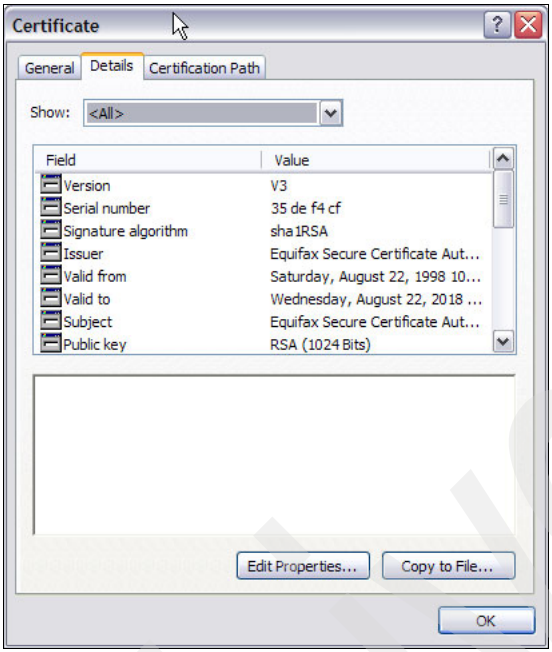

Figure B-3 Certificate Details

From here, click the **Copy to File** button. This launches the Certificate Export Wizard. Save the exported certificate as Base-64 encoded X.509 to a .cer file on your local machine, for example, C:\Equifax_Secure_Certificate_Authority. This is the file that you need to import into the WebSEAL key database.

Method 2: IKEYMAN utility

You can also export the certificate using the IBM Key Management Utility (IKEYMAN) utility, which is included with GSKit. IKEYMAN needs to be executed on the back-end servers presenting the certificate to WebSEAL. Do the following steps:

1. List the default certificate in key database (you need to know the keyfile database password):

```
export JAVA_HOME=/usr/java131
gsk7cmd -cert -getdefault -db /keyfilename.kdb
A password is required to access this key database.
Please enter a password:

Label: ihs
Key Size: 1024
```

```
Version: X509 V3
Serial Number: 3f78fd6f
Issued By: ihs
ibm
us
Subject: ihs
ibm
us
Valid From: Sun Sep 28 23:50:07 EDT 2003 To: Tue Sep 28 23:50:07 EDT
2004
Fingerprint: 61:5B:93:F3:4E:F1:93:D4:81:AA:74:35:F4:75:A3:34
Signature Algorithm: 1.2.840.113549.1.1.4
Trust Status: enabled
```

2. Extract the default certificate:

```
<syntax>
-cert -extract -db <filename> -pw <password> -label <label> -target
<filename> -format <ascii | binary>
gsk7cmd -cert -extract -db /Web/security/ihs.kdb -label "w00was02"
/tmp/w00was02.arm
A password is required to access this key database.
Please enter a password:
```

3. FTP or copy the certificate (w00was02.arm) to all WebSEAL server(s) and store them into the following directory:

   $ /var/pdWeb/www-default/certs

4. Import the participating Web server root certificate into each WebSEAL's key database:

```
export JAVA_HOME=/usr/java131
gsk7cmd -cert -add -db /var/pdWeb/www-default/certs/pdsrv.kdb -file
/tmp/w00was02.arm -label "w00was02"
```

5. Restart each WebSEAL server:

```
pdWeb restart
```

Firewall LDAP session timeout

It is important to consider LDAP session timeout settings when configuring WebSEAL on one side of a firewall with the IBM Tivoli Directory Server user registry on the other side. If the LDAP client timeout settings are greater than the firewall timeout, this will most certainly cause intermittent problems. The problem surfaces when a WebSEAL user tries to re-login to an active session and a browser timeout occurs. This can be especially problematic when all of the

available Directory Server connections have been closed by the firewall and all browser requests are receiving timeout errors.

The solution is to increase the idle session timeout on the firewall, and also implement LDAP timeout attributes in the [ldap] stanza of the Webseald-default.conf file: timeout, search-timeout, and authn-timeout.

Installation Wizard problems

Listed here are the most common installation problems encountered when using the installation wizards:

- Insufficient disk space for temporary files

  The installer can use the –is:tempdir option to explicitly specify a different location for the temporary files.

  For example, on AIX:

  ```
  install_ammgr -is:tempdir /var/scratch3
  ```

- Installation wizard failed

  Things to try:

  – Ensure that the local firewall is disabled.

  – Ensure the correct JRE version is installed.

  – Enable the Java console in a separate window using the –is:javaconsole option:

  ```
  install_amrte -is:javaconsole
  ```

  – Write the Java console output to a log file using the –is:log option:

  ```
  install_amproxy -is:log "c:\AccessManagerLogs\amproxy_ismp.log"
  ```

  – Remove the ibmjcaprovider.jar file.

  The installer may receive a prompt to remove the ibmjcaprovider.jar file and restart the installation. Be sure to not rename the file.

Multiple network interfaces

In UNIX operating systems, when there are multiple network interfaces, there may be more than one route to the Policy Server. In this scenario, the operating system may select a different route for each communication.

The Policy Server may not be able to identify each client and may produce an error similar to the following one:

```
'The server lost the client authentication, because of session
expiration.'
```

This issue can be avoided by using one of the following two solutions:

- Change the operating system routing table so that the same route is always selected.
- Set the PD_FIXED_CLIENT_IP environment variable to the IP address of a valid network interface on the operating system.

ACL problems after Tivoli Directory Server upgrade

The migration of Tivoli Directory Server is performed by following the instructions in the *IBM Tivoli Access Manager for e-business Version 6.0 Upgrade Guide*, SC32-1703:

Details are included to back up the current data with the db2ldif utility, upgrade the Tivoli Directory Server, and restore the data with the bulkload utility.

Tip: In the bulkload utility, you should specify the -A yes option to have it properly process Access Control List (ACL) updates.

However, if the ACLs are not loaded properly, Access Manager will not have the authority to perform the user and group tasks needed.

You can create the ACLs manually by following the "Applying Access Manager ACLs to new LDAP suffixes" procedure in the *IBM Tivoli Access Manager Version 6.0 Administration Guide,* SC32-1686.

Using the Web Administration Tool, apply the ACLs to all existing LDAP suffixes and secAuthority=Default entries below all defined users in the LDAP server. Applying these ACLs will restore the proper authority to allow Tivoli Access Manager to continue.

Validating and maintaining policy databases

There is a utility that is provided with the Access Manager for e-business Policy Server installation that can be used to validate and maintain the Access Manager policy database and database replicas. The utility is called pdacld_dump and is located under the installation directory in the /sbin subdirectory. It has the following functions:

- Display all database contents into readable text. The command is:

  ```
  pdacld_dump -f /var/PolicyDirector/db/Events.db
  ```

- Display Summary reports.
  - Database sequence number line produced.
  - This changes each time the database is updated.
  - Invalid objects indicates database validity.

  The command is:

  ```
  pdacld_dump -f /var/PolicyDirector/db/Events.db -s
  ```

- Repair a damaged policy database.
  - Examines the database for any corrupted content, defragments the database, and produces a valid, updated version of the database.
  - Requires a stop of all Access Manager services, swaps the old database for the new one, and restarts Access Manager services.

  The command is:

  ```
  pdacld_dump -f /var/PolicyDirector/db/Events.db -r \
  /var/PolicyDirector/db/RepairEvents.db
  ```

Using svrsslcfg for unconfiguration

If your Access Manager server ends up in a partially configured state and the pdconfig unconfiguration method is not working, there is a method that can be used to force an unconfigure. There can be many reasons an unconfigure is not working; be sure to check the basics before doing a force.

Here are a few basics to check:

- Is the Policy Server up and running? Check the logs for any problem indicators on the Policy Server.
- Did the Policy Server get reconfigured? If yes, then all of the server objects were removed when that happened.

If the Policy Server was reconfigured, the pdconfig unconfiguration will most likely give this error:

```
HPDBA0205E   The certificate presented by the SSL partner could not
be successfully validated.
```

For WebSEAL, simply move /opt/pdWeb/etc/Webseald-default.conf and /var/pdWeb/keytab-default/default-Webseald.kdb to another location and retry pdconfig/unconfiguration.

For Policy Proxy Server, remove PDMgrPrxy from the /opt/PolicyDirector/.configure directory.

- Is the Directory Server server up and working correctly? Validate that Access Manager ACLs are still in place.

  Once you are sure that you need to do a forceful unconfiguration, here are the steps to clean up the Directory Server directory and the Access Manager master authorization database. Make sure you back up your environment first.

  If the server shows up in pdadmin as follows:

  ```
  pdadmin > s l
  default-Webseald-w25wisg101
  ```

  then this can be removed from the objectspace as follows:

  ```
  touch /tmp/conf
  /opt/PolicyDirector/bin/svrsslcfg -unconfig -f /tmp/conf -n
  default-Webseald/w25wisg101
  ```

  Use the **pdacld** command to look make sure all references were removed from the object:

  ```
  /opt/PolicyDirector/sbin/pdacld_dump -f
  /var/PolicyDirector/master_authzn.db >/tmp/dump_auth
  cat /tmp/dump_auth|grep w25wisg101
  ```

  You can also search LDAP for any related objects:

  ```
  ldapsearch -h {ldapserver}-Dcn=root -w {cnrootpw}-b
  "cn=securitydaemons,secauthority=default"
  uid=default-Webseald/w25wisg101
  ```

Glossary

access control list (ACL) In computer security, a list with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is a list that is associated with a file that identifies the users who can access the file and identifies the users' access rights to that file.

authorization rule Part of the security policy that define conditions that are contained in an authorization policy. An authorization rule is used to make access decisions based on attributes such as user, application, and environment context.

certificate In computer security, a digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority.

certificate authority (CA) An organization that issues certificates. A CA creates digital signatures and public-private key pairs. The CA guarantees the identity of the individual who is granted the unique certificate and guarantees the services that the owner is authorized to use, to issue new certificates, and to revoke certificates that belong to users and organizations who are no longer authorized to use the services. The role of the CA s to authenticate the entities (users and organizations) involved in electronic transactions. Because the CA guarantees that the two parties that are exchanging information are really who they claim to be, the CA is a critical component in data security and electronic commerce.

container object A structural designation that organizes the object space into distinct functional regions.

cookie Information that a server stores on a client machine and accesses during subsequent sessions. Cookies allow servers to remember specific information about clients.

demilitarized zone (DMZ) In network security, a computer or network that uses a firewall to be isolated from, and to serve as a neutral zone between, a trusted network (for example, a private intranet) and an untrusted network (for example, the Internet). One or more secure gateways usually control access to the DMZ from the trusted or the untrusted network.

digital signature Information that is encrypted with a private key and is appended to a message to ensure the recipient of the authenticity and integrity of the message. The digital signature proves that the message was signed by the entity that owns, or has access to, the private key or shared secret symmetric key.

directory schema The valid attribute types and object classes that can appear in a directory. The attribute types and object classes define the syntax of the attribute values, which attributes are required, and which attributes are optional.

distinguished name (DN) (1) The name that uniquely identifies an entry in a directory. A distinguished name is made up of an attribute-value pairs, separated by commas.
(2) A set of name-value pairs (such as cn=common name and c=country) that uniquely identifies an entry in a digital certificate.

DMZ See demilitarized zone.

entitlement A data structure that contains externalized security policy information. Entitlements contain policy data or capabilities that are formatted in a way that is understandable to a specific application.

global sign-on (GSO) A flexible single sign-on solution that enables the user to provide alternative user names and passwords to the back-end Web application server. Through a single login, global sign-on grants users access to the computing resources they are authorized to use. Designed for large enterprises consisting of multiple systems and applications within heterogeneous, distributed computing environments, GSO eliminates the need for users to manage multiple user names and passwords. See also single sign-on.

GSO See global sign-on.

junction A logical connection that is created to establish a path from one server to another.

LDAP See lightweight directory access protocol.

lightweight directory access protocol (LDAP) An open protocol that uses TCP/IP to provide access to directories that support an X.500 model and that does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). For example, LDAP can be used to locate people, organizations, and other resources in an Internet or intranet directory.

management domain The default domain in which Tivoli Access Manager enforces security policies for authentication, authorization, and access control. This domain is created when the policy server is configured.

namespace (1) In XML, a uniform resource identifier (URI) that provides a unique name to associate with all the elements and type definitions in a schema.
(2) Space reserved by a file system to contain the names of its objects.

object space A virtual representation of the resources to be protected. See also namespace.

policy database The database that contains the security policy information for all resources in the domain. Each domain has its own policy database.

policy server The Tivoli Access Manager component that maintains the master policy database, replicates this policy information throughout the secure domain, and updates database replicas whenever a change is made to the master policy database. The policy server also maintains location information about other Tivoli Access Manager and non-Tivoli Access Manager resource managers that are operating in the secure domain.

POP See protected object policy.

protected object The logical representation of an actual system resource that is used for applying ACLs and POPs and for authorizing user access. See also protected object policy and protected object space.

protected object policy (POP) A type of security policy that imposes additional conditions on the operation permitted by the ACL policy to access a protected object. It is the responsibility of the resource manager to enforce the POP conditions.

protected object space The virtual object representation of actual system resources that is used for applying ACLs and POPs and for authorizing user access. See also protected object and protected object policy.

resource A hardware, software, or data entity that is managed.

resource manager (1) An application, program, or transaction that manages and controls access to shared resources, such as memory buffers and data sets.
(2) Any server or application that uses the authorization API to process client requests for access to resources.

role A definition of the access permissions that a user or process has and the specific resources that the user or process can modify at those levels. Users and processes are limited in how they can access resources when that user or process does not have the appropriate role.

scalability The ability of hardware, software, or a distributed system to maintain performance levels as it increases in size and increases in the number of users who access resources.

Secure Sockets Layer (SSL) A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

security policy (1) A written document that defines the security controls that you institute for your computer systems. A security policy describes the risks that you intend to minimize and the actions that should be taken if someone breaches your security controls.
(2) In Tivoli Access Manager, the combination of ACL policies, authorization rules, and protected object policies attached to objects to make them protected objects.

single sign-on (SSO) The mechanism that allows a user to log on once and access multiple applications through a single authorization challenge. Using SSO, a user does not need to log on to each application separately. See also global sign-on.

SSL See Secure Socket Layer.

SSO See single sign-on.

uniform resource identifier (URI) The character string used to identify an abstract or physical resource on the Internet. A URI typically describes how to access the resource, the computer that contains the resource, and the name of the resource. The most common form of URI is the Web page address, which is a particular subset or URI called uniform resource locator (URL). See also uniform resource locator.

uniform resource locator (URL) A character string that represent resources on a computer or in a network, such as the Internet. The URL includes the abbreviated name of the protocol used to access the information resource and the information used by the protocol to locate the resource.

URI See uniform resource identifier.

URL See uniform resource locator.

virtual hosting The capability of a Web server that allows it to appear as more than one host to the Internet.

Web Portal Manager (WPM) A Web-based graphical application used to manage Tivoli Access Manager security policy in a secure domain. An alternative to the pdadmin command-line interface, this GUI enables remote administrator access and enables administrators to create delegated user domains and assign delegate administrators to these domains.

WebSEAL A high performance, multi-threaded Web server that applies a security policy to a protected object space. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy.

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

For information about ordering these publications, see “How to get IBM Redbooks” on page 234. Note that some of the documents referenced here may be available in softcopy only.

- *Deployment Guide Series: IBM Tivoli Identity Manager*, SG24-6477
- *Enterprise Business Portals with IBM Tivoli Access Manager*, SG24-6556
- *Enterprise Business Portals II with IBM Tivoli Access Manager*, SG24-6885
- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- *Federated Identity Management and Web Services Security with IBM Tivoli Security Solutions*, SG24-6394
- *Identity Management Design Guide with IBM Tivoli Identity Manager,* SG24-6996
- *Understanding LDAP - Design and Implementation*, SG24-4986

Other publications

These publications are also relevant as further information sources:

- *IBM Tivoli Access Manager for e-business Version 6.0 Administration C API Developer Reference,* SC32-1692
- *IBM Tivoli Access Manager for e-business Version 6.0 Auditing Guide*, SC32-2202
- *IBM Tivoli Access Manager for e-business Version 6.0 BEA WebLogic Server Administration Guide*, SC32-1688
- *IBM Tivoli Access Manager for e-business Version 6.0 Installation Guide,* SC32-1361

- *IBM Tivoli Access Manager for e-business Version 6.0 Performance Tuning Guide*, SC32-1704
- *IBM Tivoli Access Manager for e-business Version 6.0 Plug-in for Web Servers Administration Guide*, SC32-1690
- *IBM Tivoli Access Manager for e-business Version 6.0 Problem Determination Guide*, SC32-1701
- *IBM Tivoli Access Manager for e-business Version 6.0 Release Notes,* SC32-1702
- *IBM Tivoli Access Manager for e-business Version 6.0 Upgrade Guide*, SC32-1703
- *IBM Tivoli Access Manager for e-business Version 6.0 WebSEAL Administration Guide,* SC32-1687
- *IBM Tivoli Access Manager Version 6.0 Administration Guide,* SC32-1686
- *IBM Tivoli Access Manager Version 6.0 Administration Java Classes Developer Reference*, SC32-1692

Online resources

These Web sites and URLs are also relevant as further information sources:

- Tivoli product documentation Web site

  `http://publib.boulder.ibm.com/tividd/td/tdmktlist.html`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

access control
  decision auditing 159
  policy 32, 113, 207
  security policy 38, 42, 45
  solution
    business driver 31
Access Control List
  see ACL
access port 135
accountability 65
ACL 4, 40–41, 165
action 42
  bit 42
administrator
  skills 13
application integrator
  skills 14
asset value 31
auditing 4, 6, 32, 65, 115, 149, 207
  POP 157
authentication 4, 53
  certificate based authentication 53
  forms authentication 53
  mechanisms 5
  step-up authentication 53
authorization 53
  database 45, 113
  rule 41, 165
Authorization Server 71
  configuration 133
  installation 113
authorization service 4
availability 31, 56
  user registry 64
  Web Portal Manager 65
  WebSEAL 59–60

B

base component
  installation 110
bulkload utility
  ACL problems 226
business context 3
business driver 6, 31

C

CDSSO
  see cross domain single sign-on
cdsso_key_gen utility 61
certificate
  based authentication 53
  management 70, 220
Certificate Authority 131
client access 49
Common Auditing and Reporting Service 65
  Access Manager configuration 156
  client installation 148
  installation 115
  staging raw data 162
Common Event Infrastructure
  configuration 140
compliance 6, 65
confidentiality 6
container objects 165
cookies 169
corporate image 31
corporate policy 38
credential vault 188
cross domain single sign-on 5, 63
  configuration 200
cryptographic algorithm FIPS 88
cryptographic functions 70
Crystal Enterprise 140
  configuration 149
Crystal Reports 115

D

DB2 115
  instance creation 117
  stored procedure 146
default ACLs 41
demonstration system 20
deployment skills 12
deployment tasks 24
directory

skills 12
suffix 124
Directory Integrator 106
password synchronization 128
Directory Server 45, 55, 71
availability 64
best practice 75
client installation 90
installation 82
schema definition 124
SSL 70, 103
SSL configuration 88
upgrade tips 226
Web Administration installation 102
DMZ 46, 50, 72
firewall 49
domain 40
cookie 61

E

e-community single sign-on 5, 64
Edge Server 59
e-mail system 33
engagement overview 18
executive assessment 19
executive summary 21

F

failover 38, 71
failover cookie 60–61
configuration 200
failure situations 57
Federated Identity Manager 60
FIPS 132
cryptographic algorithm 88
firewall 46, 49
forms authentication 53, 170
forms single sign-on 173

G

Global Security Kit
installation 80
Global Sign-On
see GSO
group 42
GSKit 70
installation 80
GSO 168
credential mapping 137
users 167

H

high availability 56, 71
configuration 203
HTTPS 47

I

IBM DB2 70
IBM Global Security Kit 70
IBM HTTP Server 71
installation 97
IP alias 98
IBM Java Runtime 70
installation 79
IBM Tivoli Common Auditing and Reporting Service
see Common Auditing and Reporting Service
IBM Tivoli Directory Integrator
see Directory Integrator
IBM Tivoli Directory Server
see Directory Server
IBM Tivoli Federated Identity Manager
see Federated Identity Manager
IBM WebSphere
see WebSphere
implementation skills 12
iNotes 33
installation
tips 225
Internet DMZ 50
IP alias 98

J

JACC provider 137
Java
runtime 70
stored procedure 146
junction 51, 169
transparent path junction 52
virtual host junction 51

K

key database 170

L

LDAP
  session timeout 224
  suffix 124
  suffix for Access Manager 82
LDIF 45
ldif2db 126
legal requirements 31, 65
license component
  installation 111
Lightweight Directory Interchange Format
  see LDIF
listening port 135
load balancing
  MAC packet forwarding 205
Lotus Notes 33

M

management
  domain 39, 45
  network 71
  zone 55
master authorization database 45, 113
Media Access Control
  packet forwarding 205
metadata 124
mission critical 31
multiple domains 39
multiple network interfaces 226
mutually authenticated SSL 48, 53

N

namespace considerations 7
network
  DMZ 46
  multiple interfaces 226
  topology 38
  transport classifications 47
  zone 46

O

object namespace 40, 72
  configuration 165
operational report 67
organization 32

P

packet forwarding 205
password
  synchronization 128
pdacld_dump 227
pdconfig 128
  unconfiguration 228
PDWebRTE
  installation 114
PeopleSoft
  Application Server 33
  integration 175
  user store 127
performance 38
permission 42
pkmslogout 61
plugin-cfg.xml 101
policy 38
policy database
  maintenance 227
Policy Proxy Server
  configuration 132
  installation 112
Policy Server 45, 55, 71
  Common Auditing and Reporting Service configuration 157
  configuration 130
  failure 58
  installation 112
  SSL 131
  standby 64
POP 4, 41, 157, 165
port 135
pre-requisite components 70
primary action group 42
Privilege Attribute Certificate 50
production network 54, 71
protected object namespace 165
  see object namespace
protected object policy
  see POP

Q

quality of protection 41, 47

R

Redbooks Web site 234
  Contact us xi

redundancy
Policy Server 64
registry 45
regulatory requirements 31
reporting 32, 65, 115, 149, 207
resource 40, 43
objects 165
restricted zone 54
reverse proxy 5, 46, 50
replication 76
risk
analysis 65
tolerance 31
role 38, 45
runtime
configuration 129
installation 112

S

scalability 38, 56, 203
schema definition 124
secAuthority=Default 82
secauthority=default 126
secure domain 40, 45
Secure Socket Layer
see SSL
security
capabilities 32, 207
infrastructure design 4
management 4
policy 38
services engagement overview 18
session affinity 60
session cookie 61
Session Management Server 60, 62
Siebel 33
Access Manager Security Adapter configuration 183
integration 178
single sign-on 4, 168, 173
configuration 173
cross domain configuration 200
WebSphere Portal 188
sizing estimates 23
skills 7
for deployment 12
resources 16
SOAP message 111
solution
tasks 23
SSL 47, 55, 70
configuration 88
standby
Policy Server 64
stash file 170
statement of work 18, 21, 209
step-up authentication 53
suffix 124
svrsslcfg
unconfiguration 227
synchronization
password 128
user data 127

T

tasks 23
deployment 24
time of day 41
time-to-market 31
TLS 132
transparent path junction 52
Transport Layer Security
see TLS
trust 47
Trust Association Interceptor 137, 140, 190

U

user registry 71
availability 64
failure 58
populating the ... 124
user repository 7

V

virtual host junction 169, 200
virtual hosting 51
vulnerability 65

W

Web Portal Manager
availability 65
failure 58
installation 104
Web Security Utilities 110
installation 111

WebSEAL 5, 72
availability 59–60
certificate 170
certificate management 220
cluster
session affinity 60
configuration 134
cookies 169
deployment 49
failure 57
filtering 169
forms single sign-on configuration 174
installation 114
junction 50, 169
LDAP session timeout 224
PeopleSoft junction 176
replication 76
session affinity 60
Siebel junction 180
single sign-on 5, 173
SSL 70, 170
transparent path junction 52
virtual host junction 51
WebSphere
... Edge Server 59
Portal 33
WebSphere Application Server 115
HTTP Server 97
installation 92
ivt script 96
load-balancing 97
plug-in 98
plugin-cfg.xml 101
profile creation wizard 93
SSL 70
UNIX symbolic links 94
WebSphere Application Server Express 91
WebSphere Portal
credential vault 188
integration 188
WPM
configuration 136
single sign-on 140

X

XML
parsing 111
XML Common Event Infrastructure data store 145

Deployment Guide Series: IBM Tivoli Access Manager for e-business V6.0

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Deployment Guide Series:
IBM Tivoli Access Manager for e-business V6.0

Full coverage of planning your access control management project

Complete hands-on installation guidelines

Based on best practices

Deploying an access control solution for a medium-size business begins with a thorough analysis of the existing business and IT environment. After we fully understand the organization, its deployed infrastructure, and the application framework, we can define an applicable representation of these assets within an access control implementation.

This IBM Redbooks publication takes a step-by-step approach to implementing an access control solution based on IBM Tivoli Access Manager for e-business. Part 1 takes you through an example company profile with existing business policies and guidelines and builds an access control solution designed for this particular environment. In Part 2, we describe how the new access control components can be integrated into the existing environment. Then we explain how to execute the access control integration tasks that must be implemented in order to create a fully functional end-to-end solution.

This book does not introduce any general access control concepts, nor does it systematically explain all of Tivoli Access Manager's components and capabilities. Instead, those details are thoroughly discussed in the IBM Redbooks publication *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014.

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

SG24-7207-01 ISBN 073843101X