Administration Guide

Enterprise Security

6.0

DOCUMENT ID: DC37004-01-0600-02

LAST REVISED: May 2004

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

Subject

This book discusses fundamental security concepts and provides instructions for setting up the Enterprise Security infrastructure so that the Security Officer can enable the types of security mechanisms needed to protect your enterprise's assets. It also describes how to implement a variety of Enterprise Security features. Your system's architecture, the nature of the back-end data that you want to protect, and the number and types of users you expect to access your enterprise environment determine which features you should implement.

Audience

This book is for anyone responsible for configuring and managing Sybase® Enterprise Security.

How to use this book

Chapter 1, "Introduction," describes several key aspects to a secure communications infrastructure and defines some common security-related terms.

Chapter 2, "Deploying Enterprise Security," describes how to deploy the Enterprise Security middleware in a BEA WebLogic server, or in an EASrever cluster.

Chapter 3, "Setting Up Security," provides step-by-step instructions for setting up the security infrastructure; creating organizations, users, groups, roles, and assets.

Chapter 4, "Using securetool," describes how to configure the Enterprise Security middleware, user interface, and database using the command line tool securetool.

Chapter 5, "Delegated Administration," describes how to configure your security system to support multiple security domains.

Chapter 6, "Auditing," provides instructions for setting up auditing to monitor user actions.

Chapter 7, "Setting up Security for Enterprise Portal," describes how to integrate the security mechanisms of J2EE, Enterprise Security, and EAServer, which provides an option to implement single sign-on capabilities.

Chapter 8, “Securing Accounts and Assets,” describes security enhancements that you can perform to secure your e-business system; for example, enabling account and asset locks, and implementing a password-strength verification component to verify passwords.

Chapter 9, “Proxy Authentication,” explains how to implement single sign-on to enterprise resources.

Chapter 10, “Configuring LDAP Authentication,” describes how to configure your security system to use an LDAP server.

Chapter 11, “Configuring the Web Server Plug-in,” describes how to set up the Web server plug-in to protect assets stored in the ACDB from unauthorized access, and how to configure the plug-in to use a secure listener.

Chapter 12, “Certificate-Based Authentication,” describes how to set up certificate-based authentication to authenticate Enterprise Portal clients, in a system that uses a Web server security plug-in and a redirector plug-in.

Chapter 13, “Using Proxy Servers,” describes how to set up Enterprise Portal to use a proxy server, which provides security, administrative control, and caching service.

Chapter 14, “Implementing a Secure Web Proxy,” provides instructions for configuring the proxy to control access to preexisting back-end Web applications, and deliver multiple applications, Web pages, and data stores as a single application, as well as implement single sign-on features.

Chapter 15, “Configuration Properties,” describes the global and domain-specific properties that configure Enterprise Security features.

Related documents

Enterprise Portal printed documentation Enterprise Security is included in the Enterprise Portal 6.0 package. The following Enterprise Portal documents are available on the *Getting Started with Enterprise Portal* CD:

- The *Enterprise Portal* installation guide for your platform explains how to install the Enterprise Portal software.
- The *Enterprise Portal* release bulletin for your platform contains last-minute information not documented elsewhere. You can also access the release bulletin from the Enterprise Portal installer.

Enterprise Portal online documentation The following Enterprise Portal documents are available in PDF and DynaText format on the *Enterprise Portal 6.0 Technical Library* CD:

- The *Enterprise Portal Developer’s Guide* includes developer-related topics for Enterprise Portal components, Portal Interface portlets, and Java Template Framework pages.

- The *Portal Interface User's Guide* describes the Portal Interface user interface and how to use Portal Interface to build and manage your enterprise's portal.

EAServer documentation EAServer is one of the applications servers into which you can install Enterprise Security, and it is included with Enterprise Portal. These EAServer documents are available in HTML format in your EAServer software installation, and in PDF and DynaText format on the *EAServer Technical Library* CD.

- *What's New in EAServer* summarizes new functionality in the latest version of EAServer.
- The *EAServer Feature Guide* explains application server concepts and architecture, such as components, transactions, and Web applications. This book also explains how to use the optional EAServer products such as Message Bridge for Java™ and the Web Services Toolkit.
- The *EAServer Programmer's Guide* explains how to create, deploy, and configure component-based applications, Web applications, Java servlets, JavaServer Pages, and how to use CORBA and Java APIs.
- The *EAServer System Administration Guide* explains how to manage EAServer with the Jaguar Manager plug-in, create new application servers, monitor servers and application components, define connection caches, and so on.
- The *EAServer Security Administration and Programming Guide* explains how to configure role-based security, configure SSL certificate based-security, implement custom security services for authentication, authorization and role-membership, and so on.
- The *EAServer Cookbook* contains tutorials and explains how to use the sample applications included with your EAServer software.
- The *EAServer API Reference Manual* contains reference pages for proprietary EAServer Java classes, ActiveX interfaces, and C routines. This document is available only online.

The *EAServer Installation Guide* for your platform explains how to install the EAServer software; it is available on the *Getting Started* CD.

The *EAServer Troubleshooting Guide* describes problems you may encounter running EAServer and possible solutions; it is available online—see the EAServer Troubleshooting Guide at http://www.sybase.com/detail?id=1024509.

jConnect™ for JDBC™ documents Enterprise Portal 6.0 includes the jConnect for JDBC driver to allow JDBC access to Sybase® database servers and gateways. The *Programmer's Reference jConnect for JDBC* is included on the *Enterprise Portal Technical Library* CD.

Note See the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

Other sources of information

Use the Sybase Getting Started CD, the Sybase Technical Library CD and the Technical Library Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

  Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.
- The Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Updates, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Technical Library Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.

5 Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software updates

❖ Finding the latest information on EBFs and software updates

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).

3 Select a product.

4 Specify a time frame and click Go.

5 Click the Info icon to display the EBF/Update report, or click the product description to download the software.

Conventions

The formatting conventions used in this manual are:

Formatting example	To indicate
commands and methods	When used in descriptive text, this font indicates keywords such as: • Command names • C++ and Java method or class names • Configuration property names
variable, *package*, or *component*	Italic font indicates: • Program variables, such as *myCounter* • Parts of input text that must be substituted, for example: `Server.log` • File names

Formatting example	To indicate
File | Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File | Save indicates "select Save from the File menu."
`package 1`	Monospace font indicates: • Information that you enter in Jaguar Manager, on a command line, or as program text • Example program fragments • Example output fragments
credentials	Bold font indicates that the term is described in the glossary.

Variables

The variables used in this manual to represent software installation directories are:

Term	Represents
JAGUAR	The EAServer installation directory
SECURITY	The Enterprise Security installation directory

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1

Introduction

This chapter provides an overview of fundamental security concepts with explanations of commonly used security terminology.

Overview

Web-based or “e-businesses” integrate and deliver critical enterprise information and applications and provide a wide range of connections between businesses and their partners, customers, employees, and suppliers.

The ability to reach this broad audience brings with it greater security demands. The potential risks of exposing private or privileged information, or high-value business transactions to accidental or deliberate intrusion can have a devastating impact on your business.

This chapter discusses primary security issues that you should consider when planning your security strategy, and describes Enterprise Security mechanisms designed to protect your e-business.

Data confidentiality, integrity, and availability

A successful e-business must devise a security strategy that focuses on three primary objectives: confidentiality, integrity, and availability of data and resources.

Your business needs should determine the level of emphasis you place on each objective. For example, national defense system security policies must place the greatest emphasis on confidentiality to protect classified and strategic information. A bank's funds-transfer system has a greater need for integrity to ensure accurate monetary balances. Finally, an emergency-medical system emphasizes availability to ensure that information and resources are accessible at all times and in many locations.

Although precautions can be taken to detect an unauthorized user, it is extremely difficult to determine if a legitimate user is purposefully doing something malicious. Therefore, the first layer securing sensitive data is preventing unauthorized individuals from accessing sensitive information.

Integrity ensures that information cannot be modified in unexpected ways. Loss of integrity results from human error, intentional tampering, or even catastrophic events. Inaccurate information can become useless or even dangerous.

Restricted availability prevents resources from being deleted or becoming inaccessible. This applies not only to information, but also to networked machines and other aspects of the technology infrastructure. Intentional attacks against computer systems often aim to disable data access or to steal the data. Limiting physical access to critical machines or data sources can eliminate accidents and internal mischief.

Similarly, protecting the network electronically is important when many entry points exist, especially from a public domain like the Internet. The next section discusses technical security issues that you should consider to secure your business environment.

Security mechanisms

Technological security measures that address your security threats can be categorized, in decreasing priority, as:

- Authentication
- Secure communications
- Access control and user authorization
- Auditing

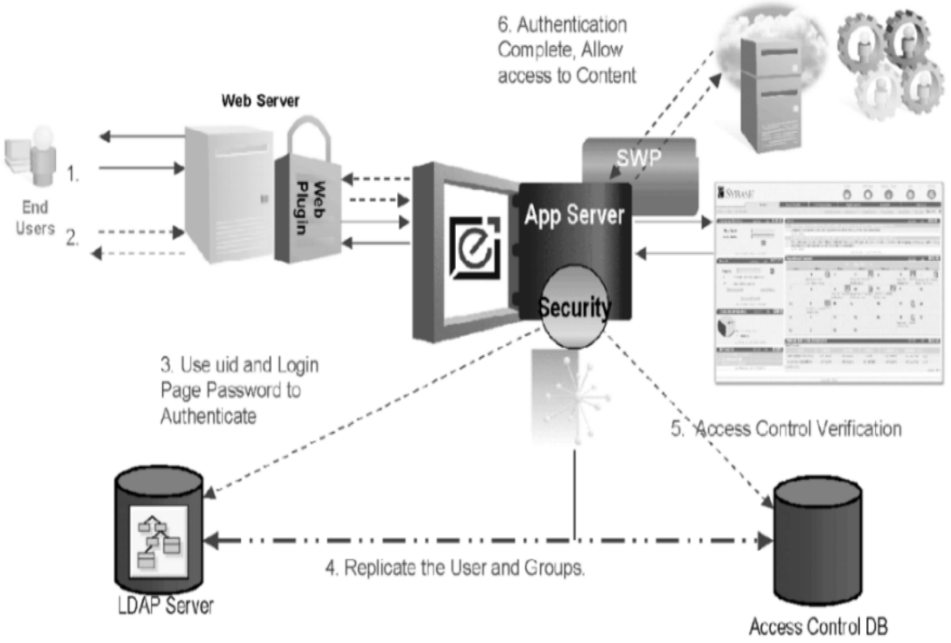
Authentication

Generally, the most commonly implemented form of protecting resources is verifying the identity of the person trying to enter the network system—authentication. Enterprise Security supports two types of user authentication—user name and password, and digital certificates.

Figure 1-1 illustrates the authentication process, where:

1. A user accesses the portal login page through a Web browser.
2. The user enters his or her user name and password.
3. An authentication request with the user name and password is passed to either an LDAP server or the ACDB.
4. If authentication is performed through LDAP, the information (success or failure) is replicated to the ACDB.
5. The system verifies that the user has permission to read the home page.
6. The user is authenticated and the home page displays.

Figure 1-1: Enterprise system architecture


6. Authentication Complete, Allow access to Content
Web Server
Web Plugin
1.
End Users
2.
SWP
App Server
Security
3. Use uid and Login Page Password to Authenticate
5. Access Control Verification
4. Replicate the User and Groups.
LDAP Server
Access Control DB

Name and password credentials

Typically, users enter a user name and password when they log in to the e-business system. The login process looks up the user ID (UID) in the Access Control Database (ACDB) and verifies the supplied password. If the authentication is successful, a session object is created that maintains the user credentials and presents them, as necessary, to other secured assets in the environment. See “Single sign-on support” on page 5.

However, a user name and password combination does not guarantee that the individual is really the person he or she claims to be. The password may have been stolen or given to someone.

User name and password pairs, also known as **credentials**, are considered a weaker form of authentication, while digital certificates are considered to be stronger.

Enterprise Security also supports using biometrics—the biological identification of a person—and smart cards—credit cards with built-in microprocessor and memory that are used for identification. Typically, your biometrics or smartcard vendor provides Web server plug-ins that allow you to use your biometrics device from your browser to unlock a keystore and present a certificate to the Web server for authentication. This must be configured by the customer—see your biometrics or smartcard vendor documentation for more information. Once configured, all that remains is to register the user’s certificate in the ACDB, and use the secure Web plug-in to complete the authentication into Enterprise Security.

Salted passwords

Enterprise Security stores passwords in a way that allows password verification while eliminating vulnerability to decryption and discovery of user passwords in the event that the system’s encryption key is compromised.

Salted passwords provide increased protection against precalculated dictionary attacks. Salting a password means adding a random (or pseudo-random) string of binary data to password data before it is encoded into the strings that are stored in the ACDB.

Digital certificates

A digital certificate is an electronic document used to identify an individual, a server, a company, or some other entity, and associate that identification with a public key. See “Public-key infrastructure” on page 6. A certificate provides generally recognized proof of an entity’s identity.

Authenticating a user by means of a digital certificate is similar to authenticating a user by means of user name and password.

A client application must first make an SSL-based connection to the application server. See “SSL handshake” on page 8. Once a session is established, Enterprise Security relies on the application server to perform the certificate authentication. The session identifies the client by the **distinguished name** (DN) embedded in the certificate, then looks up the DN in the ACDB. Again, if the authentication is successful, a session object is created and the DN is maintained and presented to other secured assets in the environment, as necessary.

This version of Enterprise Security allows the **Security Officer**, also known as the Portal Security Officer (PSO), to assign multiple digital certificates to a user’s accounts, which enables the system to authenticate the user with any of the certificates. You can assign certificates using Enterprise Security Manager—see “Managing certificates” on page 35.

Single sign-on support

Once a client has been authenticated into the Enterprise Security services, you can develop a single sign-on (**SSO**) solution, which allows clients to request access to protected assets within the environment without requiring that they reenter user credentials.

Single sign-on capability is provided by Enterprise Security as an agent for the client. If the back-end systems require a form of authentication other than a user name and password (for example, certificate-based), then single sign-on is not supported.

The Enterprise Security framework requires third-party applications to integrate with the framework for the SSO capability to persist. Enterprise Security provides APIs to allow for third-party applications to integrate with the SSO framework.

The Enterprise Security framework does not provide single sign-on features from operating-system level authentication mechanisms, or directory services, such as LDAP, UNIX or NT Active Directory, or other systems (Lotus Notes). In such a case, if the client has been authenticated by the operating system, additional authentication to the system is required.

Secure communications

While access controls and user authorization protect stored data and components, an important security factor in any e-business environment is protecting data while it is in transit.

TCP/IP is the primary transport protocol used in client/server computing, and is the protocol that governs the transmission of data over the Internet. TCP/IP uses intermediate computers to transport data from sender to recipient. The intermediate computers introduce weak links to the communication system where data may be subjected to:

- Eavesdropping – information remains intact, but privacy is compromised. For example, someone could learn your credit card number.
- Theft – the information never reaches the intended recipient.
- Tampering – information in transit is changed or replaced, then sent to the recipient. For example, someone could alter an order for goods.
- Impersonation – information passes to a person who fraudulently poses as the intended recipient.

The means to protect data while in transit are many, and the appropriate method depends on the sensitivity of the data and your users.

Public-key infrastructure

While a comprehensive discussion of public-key cryptography is beyond the scope of this document, the basics are described here to give you an understanding of how SSL secures Internet communication channels.

Several mechanisms, known collectively as **public-key cryptography**, have been developed and implemented to protect sensitive data during transmission over the Internet. Public-key cryptography consists of encryption and decryption, digital signatures, and digital certificates.

Encryption and decryption

Encryption is a process wherein a cryptographic algorithm is used to encode information to safeguard it from anyone except the intended recipient. **Decryption** is the process of decoding the information. Encryption and decryption allow two communicating systems to disguise information they send to each other. The sender encrypts information before sending it, and the receiver decrypts the information after receiving it. While in transit, the encrypted information is unintelligible to an intruder, making it less prone to theft.

Public-key encryption involves a pair of keys—a **public key** and a **private key**—associated with an entity that needs to encrypt and decrypt data. The public key is published, and the corresponding private key is kept secret. Data encrypted with your public key can be decrypted only with your private key. The reverse is also true—data encrypted with your private key can be decrypted only with your public key.

Digital signatures

Digital signatures are used for tamper detection and nonrepudiation. Digital signatures are created using a mathematical algorithm that generates a unique, fixed-length string of numbers from a text message; the result is called a **hash** or **message digest**.

To ensure message integrity, the message digest is encrypted by the signer's private key, then sent to the recipient along with information about the hashing algorithm. The recipient decrypts the message with the signer's public key. This process also regenerates the original message digest. If the digests match, the message proves to be intact and tamper free. If they do not match, the data has either been modified in transit, or the data was signed by an imposter.

Further, the digital signature provides **nonrepudiation**—senders cannot deny, or repudiate, that they sent a message, because their private key encrypted the message. Obviously, if the private key has been compromised (stolen or deciphered), the digital signature is worthless for nonrepudiation.

Secure Sockets Layer (SSL)

The Secure Sockets Layer protocol is a set of rules that govern server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information such as credit card numbers.

SSL handshake

Before the SSL connection is established, the server and the client exchange a series of I/O round trips to negotiate and agree upon a secure encrypted session. This is called the **SSL handshake**.

When a client requests a connection, the SSL-enabled server presents its certificate to prove its identity before data is transmitted. The certificate is issued by a certificate authority. See “Certificate authorities” on page 8. Essentially, the handshake consists of the following steps:

1. The client sends a connection request to the server. The request includes the SSL (or **Transport Layer Security**, TLS) options that the client supports.
2. The server returns its certificate and a list of supported **cipher suites**, which includes SSL/TLS support options, algorithms used for key exchange, and digital signatures.
3. A secure, encrypted session is established when both client and server have agreed upon a cipher suite.

For more specific information about the SSL handshake and the SSL/TLS protocol, see the Internet Engineering Task Force Web site at http://www.ietf.org.

Certificate authorities

Certificate authorities (**CA**s) are entities that validate identities and issue digital certificates. They can be either independent third parties or organizations running their own certificate-issuing server software. The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies.

In addition to a public key, a certificate always includes:

- The name of the entity it identifies
- An expiration date
- The name of the CA that issued the certificate
- The digital signature of the issuing CA
- A serial number

The CA’s digital signature allows the certificate to function as a “letter of introduction” for users who know and trust the CA but do not know the entity identified by the certificate.

Access control and user authorization

Once the identity of a user has been verified, the security system needs to control what information the user is allowed to see, modify, or which applications this user is allowed to execute. Controlling access to a data source is called access control; assigning permissions to users or groups of users to access secured assets is called authorization.

An **asset** can be a document, database information, another computer system, an application, or any other object within the enterprise's computer systems.

The process of implementing access control involves defining the information and the access permissions assigned to that information. Next, authorization to access the data is assigned to the user or a role assumed by the user.

Typical access permissions include read, write, update, create, and delete. A permission can also be the right to start or stop an application or access some other back-end system.

See "Managing a role's access permissions" on page 44.

Roles and groups

Roles enable you to enforce and maintain individual accountability. Enterprise Security provides system roles, such as PortalSecOfficer and PortalGuest, and user-defined roles, which are created by the PSO. The PortalSecOfficer role is initially granted to the "pso" user, and permits unlimited access to the security system. The PortalGuest role allows Enterprise Portal users to self-register.

Roles provide individual accountability for users performing operational and administrative tasks. Their actions can be **audited** and attributed to them.

The PSO can define role hierarchies such that if a user is granted one role, the user is also granted roles that it inherits from. For example, the "chief_financial_officer" role might contain both the "financial_analyst" and the "salary_administrator" roles. The Chief Financial Officer can perform all tasks and see all data that can be viewed by Salary Administrators and Financial Analysts.

Enterprise Security associates access control with roles (role-based access control policy). Roles can be granted to a single user or a group of users.

- A role can be granted to multiple users or groups.
- A user can have multiple active roles at a time.
- A role can inherit permissions from another role.

- Groups can be populated with multiple users from multiple organizations.

When you create new roles, keep in mind the following functionality:

- **Roles can be granted to multiple users** You can define a role in an organization and grant the role to multiple users by creating a group and populating the group with users to which you want to grant the same roles.
- **Roles can be granted to groups** You can define a “group” consisting of users from different organizations who need to share the same roles. You can then grant roles at the group level.

  A group can be defined at any organization level. For example, you can define a group for organization A. You can then populate the group with users from organization A and with users from the suborganizations of A.

  To populate a group with users from different organizations, create the group at a common root level.
- **Role permissions can be inherited** If the PSO wants to create a role that includes the permissions defined by another role, in addition to some new permissions, the role hierarchy eliminates duplicating the role definition and is more efficient.

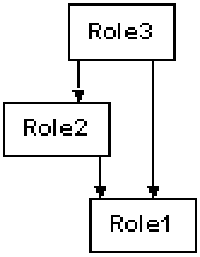
  For example, if you want Role1 to contain the permissions of Role2, then it is more efficient when you create Role1 to say that it inherits permissions from Role2, instead of redefining the permission set.

  Also, to modify Role2 later, you need not modify Role1 to reflect the changes because it inherits its permissions from Role2 and changes to Role2 are automatically reflected in Role1.

  For example:

  - If Role1 inherits from Role2 and Role2 inherits from Role3, Role1 implicitly inherits from Role3. This way even if you modify Role2 later to not inherit from Role3, Role1 still inherits from Role3.

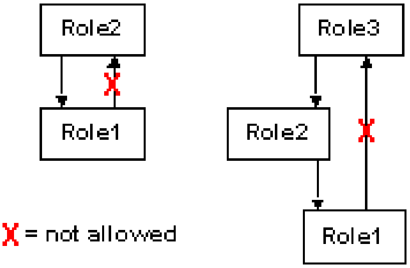
Figure 1-2: Roles hierarchy



- A role cannot inherit from a role that either directly or indirectly inherits from it—this prevents loops in the hierarchy.

  For example, if Role1 inherits from Role2, then Role2 cannot inherit from Role1. Also, if Role1 inherits from Role2 and Role2 inherits from Role3, then Role3 cannot inherit from either Role1 or Role2.

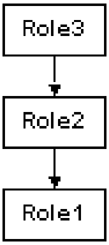
Figure 1-3: Roles inheritance restrictions



When users are granted a role, they get the privileges of all roles from which the granted role has inherited. However, users can assume only those roles that are explicitly granted to them. For example, if a user is granted Role1, which inherits from Role2, the user can assume Role1 but not Role2.

- If Role1 inherits from Role2, which inherits from Role3, you get:

Figure 1-4: Roles inheritance activation



If you remove the link between Role2 and Role1, Role1 no longer contains the privileges of Role3. If you want Role1 to possess the privileges of Role3 irrespective of the definition of Role2, define Role1 to inherit from both Role2 and Role3.

For information about how to create roles and groups, see “Setting up the security system” on page 30.

Authorization delegate

Enterprise Security stores authentication and authorization information in a JDBC-compliant database. If you are using Enterprise Security with Enterprise Portal, the database is installed in the DBMS (database management system) you select at installation.

Enterprise Security implements an authorization delegate, which allows you to store authorization information in a different data store. The authorization delegate establishes a connection to this data store and retrieves the authorization information.

Auditing

A well-balanced IT security policy should include complementary proactive and reactive components. The proactive component uses strong security controls such as those described above, while the reactive component includes auditing and monitoring those security controls. Both components are necessary to maintain effective security control.

The purpose of **auditing** is to track user actions and keep an audit trail that can be read by some intrusion detection system. There are two aspects of auditing.

The first aspect is to keep track of information about users who have been authenticated into the system, as well as the failed attempts of authentication.

The second aspect of auditing is to keep track of what information or which resources a particular user has accessed or failed to access. Also, the type of action, such as updating or deleting, is an important part of the access audit trail.

For information about configuring auditing, see Chapter 6, “Auditing.”

Enterprise Security architecture

Enterprise Security is managed by the PSO using Enterprise Security Manager, a graphics-based administration tool. The security architecture provides a uniform structure for user authentication and authorization for both Sybase and non-Sybase applications and components.

Enterprise Security includes:

- EAServer service components:
  - Connection manager
  - Object manager
- Access Control Database
- Enterprise Security Manager
- Security Management API
- Web server plug-in

This section describes each piece of the Enterprise Security architecture.

EAServer service components

Connection manager

The connection manager authenticates users to the enterprise, maintains their security session state, and interfaces to the ACDB. The connection manager can accept:

- User name and password authentication mechanisms
- Digital certificates
- PKI authentication mechanisms

The connection manager takes the authentication mechanism provided by the user and validates it against the user's record in the ACDB.

In addition, a security plug-in resides on the Web server. The security plug-in resides outside the firewall and manages the connection to the content management services, and sends login and request information to the connection manager for validation. See “Web server plug-in” on page 18.

After a successful login, the connection manager retrieves the user's profile from the ACDB and creates an authentication string, called the session handle, that exists for the duration of the user session. This session handle can be passed throughout the system and used wherever user authentication is required.

Object manager

The object manager is a service component in EAServer. It is used to manage the information in the ACDB and is the connection between Enterprise Security Manager and the ACDB.

Access Control Database

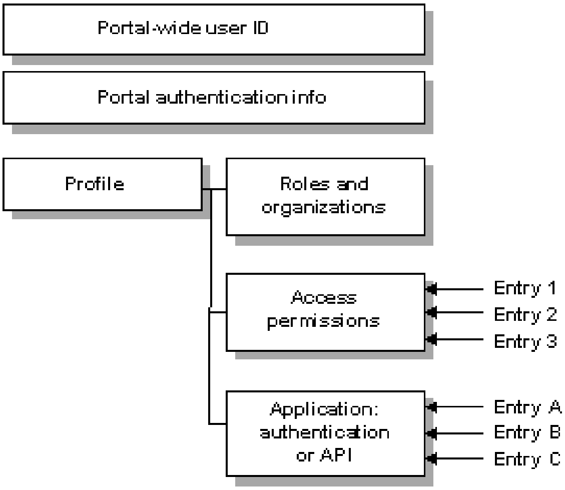
Every installation of Enterprise Security, whether in a standalone or multimachine environment, has a central database that stores all of the user's authorization and authentication information, such as user name and password credentials, digital certificates, and access permissions to the enterprise components. Unless you have installed and configured a third-party LDAP server to store authorization and authentication information, the database that stores this information is called the Access Control Database, or ACDB.

For information on configuring your environment to use a third-party LDAP server, see Chapter 10, “Configuring LDAP Authentication.” For information on configuring single sign-on capabilities through the Portal Interface, see Chapter 12, “Certificate-Based Authentication.”

In addition to the information that is automatically inserted into the database, there is configurable information that is defined and implemented by the PSO. The PSO configures user-specific and asset-specific information using Enterprise Security Manager—see “Enterprise Security Manager” on page 16.

The ACDB stores information for each user, role, and asset. Figure 1-5 shows the content of a user record.

Figure 1-5: Access Control Database record for a user



- Portal user ID – uniquely identifies the user throughout the portal. If you are using digital certificates, this field contains the distinguished name (DN) from the certificate.
- Portal authentication information – this field contains the information required to authenticate the user to the portal. It can be a password, a digital certificate, or a call to an authentication PKI.
- User profile – contains other information specific to the user, including:
  - The user’s default roles or organizational associations that can be used for access control decisions in place of the user’s actual identity.

  - A role can be used in two ways. It can be used in place of the user’s identity. In this case, the system may not need to know who the user is. A company affiliation can be sufficient. A role can also be used when access control decisions are made based on users’ identities and the roles they possess. For example, a branch manager role will have different permissions than a teller role.

- Information required for transparent authentication to legacy systems with their own security mechanisms.

The records for roles and assets are similar to those for users. You can define access permissions for roles and then assign roles to users, a process that simplifies security management for large user populations. By defining access permissions for roles to access different assets, you can specify the extent of what a user in that role can do.

Enterprise Security Manager

The PSO administers security from a graphical interface called Enterprise Security Manager. If you are using Enterprise Security with Enterprise Portal, the graphical interface is called “Portal Studio.” The PSO role is defined automatically when you install and configure Enterprise Security. The PSO role has all permissions and is assigned a default login. You can use this role to initially log in and create user names and passwords for security officers, the administrator, and to grant the appropriate roles. You can then invalidate or delete the default login to secure the product against intruders who possess the default login information.

See Chapter 3, “Setting Up Security” for information on using Enterprise Security Manager.

Security Management API

The Security Management API (SMAPI) allows you to manage security programmatically. You can also develop your own front-end tool that uses the SMAPI to manage your security data. The SMAPI replaces the Object Management API, providing greater functionality that is easier to use.

Enterprise Security provides a default implementation for managing the data in your ACDB. You can contact Sybase Professional Services to customize your implementation.

The SMAPI supports these features:

- Enables you to develop a password validation component to validate new passwords. A sample component is provided, which you can customize. See “Verifying passwords” on page 158 for more information.

  You can specify the number of days that a user’s password is valid before it expires. This is enabled using Enterprise Security Manager—see “Creating and managing user accounts” on page 32.

- Self-registration – new users can set up their own accounts and register with the secured system, using the SubjectManagement interface. Previously, the PSO was required to register all new users, and assign their roles and permissions. See “Self-registration group” on page 40.
- Internal assets – the SMAPI exposes some internal assets used to accomplish management tasks through the AssetManagement interface. Previously, only users who were granted the PortalSecOfficer role had access to these internal assets.
- Proxy authentication information management – users can manage their proxy authentication information that is defined at the user level. Administrators can also define proxy authentication information at the asset and role level. See Chapter 3, “Setting Up Security” and Chapter 9, “Proxy Authentication.”
- Subject account management – the SubjectManagement interface allows users to update their own account information. Users cannot delete their accounts; the PSO must delete any unwanted or unused accounts.

  The default implementation of the SubjectManagement interface supports access to and modification of data in the ACDB. If you store authentication and authorization information in another data store, you must write and deploy your own implementation of the provided APIs.

  For self-registration and personal account management, you must develop your own EJB client that uses the SubjectManagement interface. Enterprise Security Manager does not provide a user interface that allows users to update their own account.

SMAPI documentation To view the SMAPI documentation, open a browser, and access *docs/html/index.html* in your Enterprise Security installation; then, select the com.sybase.ep.security.management package.

Web server plug-in

Enterprise Security provides a Web server plug-in that protects URLs from unauthorized access. The Web server plug-in provides the bridge that allows you to define URL-type assets using Enterprise Security Manager, then map them to local Web addresses to provide protection to the document level.

The Web server plug-in is a shared library module that is loaded into the Web server to intercept HTTP requests to perform user authentication and access authorization as needed.

The plug-in consists of:

- A connection manager – interfaces with EAServer to perform authentication and authorization. A successful connection creates a session object, which maintains the connection cache that stores active session information.
- An asset cache manager – a cache that stores asset URLs. The cache is checked whenever a connection request is received to determine if the requested URL is unprotected. If the URL is not in the cache, the connection manager queries the ACDB before making the connection to determine what user authentication is required.

The plug-in also has a custom bean that instructs the asset connection manager to update the asset cache whenever a URL asset is created or updated in the ACDB. See Chapter 11, “Configuring the Web Server Plug-in” for instructions on configuring the plug-in.

BEA WebLogic support

Beginning with version 6.0, Enterprise Security supports the BEA WebLogic application server. You can install Enterprise Security as a security provider in the WebLogic server, which allows clients to use Enterprise Security credentials. WebLogic components can declaratively and programmatically perform security checks against Enterprise Security resources, such as roles. When Enterprise Security is installed in a WebLogic server, the Enterprise Security runtime components PortalSession and SMAPI are available to other beans in the server. WebLogic allows you to look up security components using either JNDI names or EJB references.

Planning the security system

Enterprise Security provides a comprehensive package of security services that encompasses all aspects of security, including authentication, authorization, and encryption. The type of security that you design is based on your system.

User populations for e-business systems can be very large, creating an environment in which traditional user account management and access control cannot function well.

Before setting up the Enterprise Security system, the PSO should carefully review the structure of the business enterprise, the assets that must be secured, the types of security required for each asset, the information needs of users, whether encryption and digital signatures are to be included, available hardware and software, failover requirements, and other issues that affect security and the secured system.

Note The system administrator has special access rights to install and configure Enterprise Security. The PSO should reset the administrator’s security access after installation and configuration are complete.

The following procedure summarizes the steps that are necessary to set up security and populate the ACDB. Chapter 3, “Setting Up Security,” provides details on each step.

❖ Setting up your security system

1. Create the organizational hierarchy for the security system. The root level organization is defined when you install and configure Enterprise Security. You can create suborganizations to mirror the organizational structure of your enterprise. Enterprise Security does not limit the number of suborganizations or the number of organizational levels. However, there can be only one root organization.

   The Enterprise Security installation also creates a default domain—which contains the root organization—and an associated security policy. If you create a suborganization, you can choose whether to create it in the default domain or in a new domain.

2. Create users.
3. Create groups and populate them with users.
4. Create roles for users and groups.
5. Create assets.

6 Define permissions so users can access the assets.

Other security notes

High availability

You can also configure security for high availability. You can configure an application server cluster and Adaptive Server in a companion environment to provide high availability in the event that either server goes down.

To configure an EAServer cluster, see the *EAServer System Administration Guide*, Chapter 6, “Clusters and Synchronization” and Chapter 7, “Load Balancing, Failover, and Component Availability.” To configure Adaptive Server in a companion server architecture, see *Adaptive Server Enterprise 12.5, Using Sybase Failover in a High Availability System*. Both books are available at http://www.sybase.com/support/manuals.

To configure a WebLogic cluster, see your BEA WebLogic documentation.

Multiple language support

The Enterprise Security log4j logging subsystem supports localized error messages. For information about log4j, see the Apache Jakarta Project Web page at http://jakarta.apache.org/log4j.

Also, all administration windows in Enterprise Security Manager support double-byte character sets, and can be localized.

Integration

The Enterprise Security framework works with third-party security frameworks, such as that provided by Netegrity. Sybase provides an authentication delegate sample, which you can customize to work with these third-party security frameworks.

CHAPTER 2 Deploying Enterprise Security

This chapter describes how to deploy the Enterprise Security middleware in an EAServer cluster, and how to deploy and configure Enterprise Security in a BEA WebLogic server.

If your application server is WebLogic and you ran the Enterprise Portal version 6.0 installer, you can skip this chapter, as the installer deploys and configures Enterprise Security automatically. Continue to Chapter 3, "Setting Up Security."

Deploying security in an EAServer cluster

If you install Enterprise Security into EAServer, and the server is the primary member of a cluster, you can deploy Enterprise Security to all servers in the cluster using the following procedure.

❖ Deploying security in a cluster

1. Copy the entire *Security* installation directory and its subdirectories from the primary EAServer machine to all the machines with secondary EAServer installations. Verify that the encryption key file *.enk* is replicated to all machines.
2. If the directory structure is not identical on all machines, you must update the script files in the *Security/bin* directory with the correct paths for each machine. On UNIX and Linux, the script files end with ".sh"; on Windows, the script files end with ".bat".

3 Use the securetool deploymw command to deploy the security middleware in each secondary EAServer installation—see deploymw on page 66.

Note All Enterprise Security installations must point to the same ACDB.

4 Synchronize the EAServer cluster—see Chapter 6, "Clusters and Synchronization," in the *EAServer System Administration Guide*. To access EAServer documentation, use a Web browser to open *html/docs/index.html* in your EAServer installation.

Deploying and configuring security in WebLogic

You can deploy Enterprise Security in a BEA WebLogic server using either:

- The installer—see the *Enterprise Portal Installation Guide* for your platform, or
- securetool—see "Deploying Enterprise Security in WebLogic using securetool," below.

To remove Enterprise Security, see "Removing Enterprise Security from a WebLogic server" on page 25.

❖ Deploying Enterprise Security in WebLogic using securetool

1 Set the BEA_HOME environment variable to the WebLogic installation directory.

2 Set the JAVA_HOME environment variable to point to a JDK 1.4 installation.

3 Deploy the middleware, using the securetool wls_deploymw command—see wls_deploymw on page 86.

4 Configure the middleware, using either:

- The securetool wls_configmw command—see wls_configmw on page 85, or
- The WebLogic Server Console—see "Configuring Enterprise Security using the WebLogic Server Console" on page 23.

5 Deploy Enterprise Security Manager, using the securetool wls_deploysm command—see wls_deploysm on page 88.

For complete information about securetool, see Chapter 4, “Using securetool.”

❖ **Configuring Enterprise Security using the WebLogic Server Console**

1. Using a Web browser, connect and log in to the WebLogic Server Console; typically, using this URL: http://localhost:7001/console.

2. Configure the Sybase security provider for the default WebLogic security realm:

   a If you configure the Sybase providers in the default realm, the system user already exists; it was created when you deployed Enterprise Security. If you use another realm, create a system user for this realm, using the same system user name as in the default realm. By default, the system user name in the default realm is “SybaseSecuritySystemIdentity.” Although the system user name for each realm must be the same, the passwords can be different.

   b In the left pane, open the Security | Realm folder. Select the default security realm; typically, “myrealm.”

   c In the right pane, select the Providers tab; then, select the Authentication tab. You should see two providers installed, “DefaultAuthenticator” and “DefaultIdentityAsserter.”

   d Select Configure a New Sybase Authenticator, then click Create.

   e Change the Control Flag setting to Optional, and click Apply.

   f Select the Details tab, and set the Provider URL to the WebLogic server URL, if it is different from the default, which is t3://localhost:7001.

   g In the left pane, select “myrealm.” Note the newly added SybaseAuthenticator instance.

   h From the list of configured authenticators, select DefaultAuthenticator. Change the Control Flag setting to Sufficient, and click Apply.

3. Configure the SybaseAuthorizer:

   a In the left pane, select “myrealm.” In the right pane, select the Authorizers tab.

   b Select Configure a New Sybase Authorizer, accept the defaults, and click Create.

   c Set the Provider URL to the WebLogic server URL, if it is different from the default, which is t3://localhost:7001.

d The SybaseAuthorizer uses the isAllowedAccess method to perform certain tasks, such as automatic session extension, which should be performed on each authorization invocation. This method does not actually verify whether a caller is allowed access. The default return value from isAllowedAccess is “PERMIT”; other possible values are “ABSTAIN” and “DENY.”

If you change the return value to “ABSTAIN,” you must reconfigure the Adjudication provider:

1 In the right pane, select the configured Adjudication provider, and choose the Details tab.

2 Uncheck the box titled “Require Unanimous Permit,” and click Apply.

4 Configure a new SybaseIdentityAsserter:

a In the left pane, select “myrealm.” In the right pane, select the Providers tab, then select the Authentication tab.

You should see these providers already installed: DefaultAuthenticator, DefaultIdentityAsserter, SybaseAuthenticator, and SybaseIdentityAsserter.

To enable the SybaseIdentityAsserter to perform certificate authentication, you must first enable two-way SSL on your WebLogic server. See the BEA WebLogic documentation.

b Select Configure a New Sybase Identity Asserter, and click Create.

c In the Available Active Types list, highlight X.509, and move it to the Chosen list. Click Apply.

d Select the Details tab, and verify that the Provider URL is set to “t3s://localhost:7002.” Click Apply.

5 Configure the Sybase role provider:

a In the left pane, select the default realm (myrealm). In the right pane, select the Providers tab, then select the Role Mapping tab. You should see an instance of the Sybase Role Mapper.

b Select Configure a New Sybase Role Mapper, accept the defaults, and click Create.

c. Select the Details tab, and in the Data Source JNDI Name field, enter the JNDI name of the data source that connects to the ACDB. The default value is set to the JNDI name of the data source that is created when Enterprise Security is installed.

   If you change the data source name, you must update the value in the Data Source JNDI Name field and the values in the deployment descriptors for the security modules.

6. Restart the WebLogic server.

For information about setting up authentication and role mapping, see “Setting up WebLogic authentication” on page 149.

❖ Removing Enterprise Security from a WebLogic server

To remove Enterprise Security from a WebLogic server, you can either run the uninstaller, or perform the following steps. For information about running the uninstaller, see the *Enterprise Portal Installation Guide* for your platform.

1. Remove the middleware using the securetool wls_removemw command—see wls_removemw on page 89.
2. Remove Enterprise Security Manager using the securetool wls_removesm command—see wls_removesm on page 90.

CHAPTER 3 Setting Up Security

This chapter describes the tasks that the PSO must complete to set up the Enterprise Security services.

Note If you plan to use PKI certificates for authentication and security, you must establish your security policy, install the proper hardware and software, and validate the issuance and acquisition of user certificates before you configure the Enterprise Security components.

Before you proceed with the installation, Sybase recommends that you follow the advice of the PKI vendor and a Sybase Professional Services representative to ensure that any existing LDAP products support your PKI certificate management plans.

Starting Enterprise Security Manager

The PSO administers security from a Web-based graphical interface called Enterprise Security Manager. Enterprise Security Manager is installed with the Enterprise Security services and is supported in a standalone EAServer or WebLogic environment, as well as with Enterprise Portal.

When Enterprise Security is installed with Enterprise Portal, the title on the main window is “Portal Studio.”

❖ Launching Enterprise Security Manager

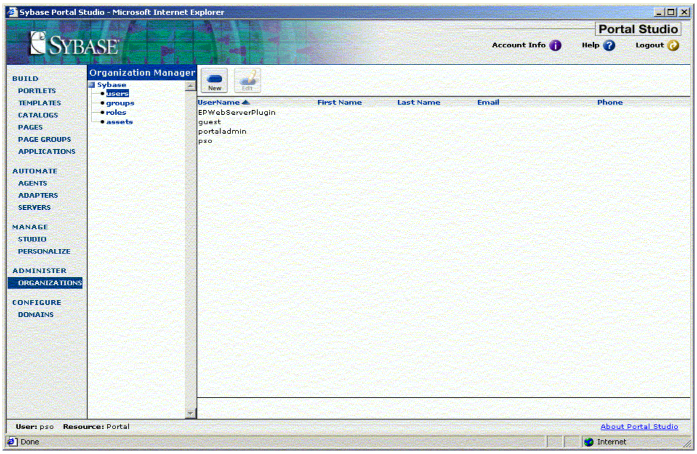
1. Enter this URL in your browser; *host* and *domain* identify where the application server is running, and *port* is the application server's HTTPS port number; the default for EAServer is 8081:

   ```
   https://host.domain:port/onepage/index.html
   ```

2. In the Login window, enter your user name and password, and click Login.

   If you accepted the defaults during installation, the user name is "pso" and the password is "123qwe".

Figure 3-1: Enterprise Security Manager



UserName ▲	First Name	Last Name	Email	Phone
EPWebServerPlugin				
guest				
portaladmin				
pso				

User: pso **Resource:** Portal

Once you are logged in, you see a multi-pane window that consists of:

- Status bar – the lower-left corner of the window displays the user name of the person logged in, and the name of the Enterprise Portal co-brand with which this user is associated; in Figure 3-1, the user name is "pso" and the co-brand is "Portal." For information about co-brands, see Chapter 11, "Creating Multiple Portals," in the *Enterprise Portal Developer's Guide*.
- Toolbar – in the upper-right corner of the window is the static toolbar, from which you can view your account information, access online help, and log out.

  Once you select from the menu in the left pane, an application-specific toolbar displays above the right pane; in Figure 3-1, Administer | Organizations is selected, and the application-specific toolbar consists of New and Edit buttons.

- The main window is divided into three panes. The selection you make in the left pane determines what displays in the center pane, and your selection in the center pane determines what displays in the right pane.

  The Enterprise Security menu options in the left pane allows you to select from:

  - Administer Organizations – create, edit, and manage users, groups, assets, permissions, and roles.
  - Configure Domains – configure security domains.

  For information about the Build, Automate, and Manage menu options, see the *Enterprise Portal Developer's Guide*.

Using Enterprise Security Manager, you can:

- Create security objects
- View a security object's properties
- Update a security object's properties
- Delete security objects

At the highest level is the root organization container, which is created when you install and configure Enterprise Security services. You can also create suborganizations under the root organization. In each organization and suborganization you can create these security objects:

- Users – Enterprise Security users. For each user, you can define a user profile, and proxy authentication information.
- Groups – groups of users. If you grant a role to a group, each user in the group has that role.
- Roles – a set of permissions to access assets. The permissions assigned to a role define what a user with that role can do in the secured system. You can grant multiple roles to users and groups.

  Each role can have multiple permissions assigned to it, which permit access to assets. Examples of permissions are READ, WRITE, DELETE, UPDATE, LIST, and GRANT.

- Assets – can include any object to which you want to restrict access. An asset can be a URL, an application, a database, a table in a database, or a column in a table. Every asset can have a custodian who is responsible for that asset. Figure 3-1 on page 28 displays the assets that are created automatically when you install Enterprise Portal.

Setting up the security system

This section contains detailed instructions for setting up the Enterprise Security system. You perform all the functions in this section from Enterprise Security Manager.

Change default passwords To secure your environment after the initial installation, Sybase strongly recommends that you change all default passwords.

To set up your security system:

1. Create the organizational hierarchy for the security system. See “Managing organizations and suborganizations” on page 30.
2. Populate the organizations and suborganizations with users. See “Creating and managing user accounts” on page 32.
3. Create groups and populate them with users. See “Creating and managing groups” on page 38.
4. Create roles for users and groups. See “Creating and managing roles” on page 42.
5. Create assets that you want to secure. See “Creating and managing assets” on page 46.
6. Define permissions so users can access the assets. See “Managing a role’s access permissions” on page 44.

Managing organizations and suborganizations

At the highest level of the object tree, you find the root organization, which is created during the installation and configuration of Enterprise Security. There can be only one root organization.

The Security Officer uses Enterprise Security Manager to create, view, update, and delete suborganizations. There is no limit to the number of suborganizations you can have, but they must all be below the root organization.

If you create a suborganization, you can choose whether to create it in the default domain or in a new security domain. Each security domain contains a set of **controlling assets**, which control access to the security objects in the domain. See Chapter 5, “Delegated Administration,” for information about security domains and controlling assets.

Table 3-1 describes the permissions you must have to manage organizations.

Table 3-1: Permissions required to manage organizations

Action	Permissions required
Create an organization	LIST and WRITE on the organization controlling asset, and LIST on the controlling asset of the domain, because you must select the domain in which to create the organization. To display a list of all domains, you must have LIST permission on the controlling asset of each domain.
View the organizations in a domain	LIST on the organization controlling asset.
View the properties of an organization	READ on the organization controlling asset and the domain controlling asset.
Update organization properties	UPDATE on the organization controlling asset.
Move an organization to a different domain	READ, UPDATE, and DELETE on the organization controlling asset in the current domain and WRITE on the organization controlling asset in the new domain.
Delete an organization	READ and DELETE on the organization controlling asset.

❖ Creating an organization

1. In the Organization Manager tree view, highlight the organization under which you want to create a suborganization, and click New.
2. In the Create New Organization dialog box, enter these values, and click OK:
   - Organization Name – name for the suborganization.
   - Security Domain – select the name of an existing security domain from the drop-down list.
   - Description – a description of the suborganization.

A container for the new suborganization displays in the center pane. The suborganization displays the same objects as its parent organization, but without any entries. Repeat this process for each suborganization you want to create.

❖ Viewing an organization's properties

- Highlight the organization's name in the Organization Manager tree view. The description and the name of the security domain that contains the organization display in the right pane.

❖ Updating an organization's properties

1. In the Organization Manager tree view, highlight the name of the organization you want to update, and click Edit.The Edit Organization dialog box displays.
2. Modify the organization's properties, and click OK.

❖ Deleting an organization

You can delete any organization except the root organization.

1. In the Organization Manager tree view, highlight the suborganization you want to delete.
2. In the right pane, right-click, and select Delete Organization.
3. Click Yes to confirm the deletion.

Creating and managing user accounts

Before a user can access any of the system assets, you must establish a user account and an account policy. An account policy contains information about a user's account and password. This information determines whether the user is allowed to log in to the secured system.

This section describes how to create user accounts and define their account policies.

Table 3-2 describes the permissions you must have to manage user accounts.

Table 3-2: Permissions required to manage user accounts

Action	Permissions required
Create a user account	WRITE on the subject controlling asset.
List the users in a domain	LIST on the subject controlling asset.
View the properties of a user account	READ on the subject controlling asset.
Update the properties of a user account	UPDATE on the subject controlling asset.
View a user's digital certificates	READ on the subject controlling asset.
Register or remove a certificate	UPDATE on the subject controlling asset.
Move a user account to a different organization	If the new organization is in the same domain, you need READ, DELETE, and WRITE on the subject controlling asset. If the organization is in a different domain, you need DELETE on the subject controlling asset in the current domain, and WRITE on the subject controlling asset in the new domain.
Manage a user's group memberships	READ and UPDATE on the group controlling asset.
Edit a user's roles	GRANT on the role controlling asset.
Display a user's access permissions	LIST on the asset controlling asset in each domain where the user has permission to access assets. For example, if a user has permission to access assets in three different domains, you need LIST permission on the asset controlling asset in all three domains.
Delete a user account	READ and DELETE on the subject controlling asset.

❖ Creating a user account

1 In the Organization Manager tree view, select the organization, highlight Users, and click New.

2 In the Create New User dialog, enter:

- Login Name – the name used to log in to the system.
- First Name – user's first name. This field is optional; however, if you enter either a first name or last name, you must enter both names.
- Last Name – user's last name. This field is optional; however, if you enter either a first name or last name, you must enter both names.

- Common Name – the name that displays in Enterprise Security Manager. A user’s common name should be unique throughout the security system.
- Password – a password for the user.
- Verify Password – reenter the password so the system can verify that it was entered correctly.
- E-Mail – the user’s e-mail address. This field is optional.
- Work Phone – the user’s work telephone number. This field is optional.

3 Configure the account policy by selecting from the following:

- Account is disabled – disables the account so the user cannot access the system until the PSO enables the account.
- Password never expires – makes the provided password valid indefinitely.
- Account never expires due to inactivity – keeps the account valid regardless of user activity.
- Account has fixed expiration date – select if you want the account to expire on a specific date. If you select this, enter the expiration date. If you do not set an expiration date, it is determined by the value you set on the security domain’s account properties tab—see “Configuring account properties for a security domain” on page 98.

4 Click OK.

Using multiple Enterprise Portal co-brands If your Enterprise Portal installation contains multiple co-brands, a Portal Interface user must have a separate user account with a unique user name for each co-brand.

Managing user accounts

In the Organization Manager tree view, select an organization, and highlight Users. The users that belong to this organization display in the right pane.

❖ Viewing and updating a user account

1 In the right pane, highlight the user name, and click Edit.

2 You can modify any field that is not dimmed. Click OK to save any changes.

Updating the EPWebServerPlugin password If you change the EPWebServerPlugin user’s password, and you plan to use the Web server plug-in, you must update the password in the *default_credential.txt* file, which is installed with your Web server plug-in—see Chapter 11, “Configuring the Web Server Plug-in.”

❖ Managing certificates

For information about how Enterprise Security uses digital certificates to authenticate users, see “Authentication” on page 3.

1. In the right pane, highlight the user name, right-click, and select Manage Certificates. The Manage Certificates dialog box displays the user’s existing certificates.
2. To add a new certificate, click New. The Register Certificate dialog box displays. Enter one of these values, then click OK:
   - The name of a base64 X.509 certificate file, or click Browse, and select the file name. By convention, certificate files have a *.cer* extension.

     If you have a certificate installed in Internet Explorer, you can export it, and register it here. To export a certificate:

     a. In Internet Explorer, select Tools | Internet Options.
     b. In the Internet Options dialog box, select the Content tab, then highlight the certificate, and click Export.
     c. In the Certificate Export wizard, select Base-64 Encoded X.509, click Next, then enter a file name for the certificate.
   - The certificate DN.

   The new certificate information displays in the Manage Certificates dialog box.

   To view an existing certificate, highlight the row, and click View.

   To delete an existing certificate, highlight the row, and click Delete.
3. Click OK.

You can also manage certificates using the SubjectManagement interface defined in SMAPI. To view the SMAPI documentation, open a browser, and access *docs/html/index.html* in your Enterprise Security installation.

❖ Moving a user between organizations

1. In the right pane, highlight the user name, right-click, and select Change Organization. The Change User Organization dialog box displays.
2. In the To New Organization list, highlight the organization to which you want to move the user, and click OK.

❖ Managing users’ group memberships

Before you can add users to a group, you must create the group—see “Creating a group” on page 38.

1. In the Organization Manager tree view, select the organization, and highlight Users.
2. In the right pane, highlight the user whose group memberships you want to edit, right-click, and select Edit Group Memberships. The Edit User Group Memberships dialog box displays.
3. In the left list box, select the organization. The groups that belong to this organization display in the middle list box.
4. To add the user to a group, highlight the group name, and click Add. The group name displays in the Members Of list box.

   Repeat this step for each group to which you want to add this user.
5. To remove the user from a group, highlight the group name in the Members Of list box, and click Remove.
6. When you are finished, click OK.

Accessing Portal Studio To enable users to access all the Portal Studio development features, grant them the StudioAdmin role.

To restrict Portal Studio users to a subset of the studio features:

1. Create a custom studio role—see “Creating a role” on page 43.
2. Assign appropriate studio permissions to the role—see “Managing a role’s access permissions” on page 44.
3. Grant the role to your studio users—see “Editing a user’s roles,” below.

❖ Editing a user’s roles

1. In the right pane, highlight the user name, right-click, and select Edit Roles. The Edit User Roles dialog box displays.

2 In the left list box, select the organization. The roles that belong to this organization display in the adjacent list box.

The Inherited Roles list box displays the roles that are granted to a group of which the user is a member. The user inherits these roles as a group member.

3 To grant a role to the user, highlight the role name, and click Add. The role name displays in the Granted Roles list box. The name of the organization to which each role belongs also displays, in parentheses.

Repeat this step for each role you want to grant to the user.

4 To revoke a role from the user, highlight the role name in the Granted Roles list box, and click Remove.

5 To save your changes, click OK.

❖ Listing a user's access permissions

To see which assets a user has permissions to access, and the access types:

- In the right pane, highlight the user, right-click, and select List Permissions. The List User Access Permissions dialog box displays each asset that the user has permission to access, the organization in which the asset belongs, and which permissions are assigned.

❖ Changing a user's password

If you change a user's password, and the user has a proxy user name and password defined, enter the new password on the Portal Studio AccountInfo tab; then, reenter the proxy user name and password—see "Registering proxy user names and passwords with Portal Studio" on page 214.

1 In the Organization Manager tree view, select an organization, and highlight Users.

2 In the right pane, highlight the user's name, right-click, and select Reset Password. The Reset Password dialog box displays.

3 Enter the new password twice, and click OK.

❖ Deleting a user account

Note If you delete a user's account, the user cannot log in to either Portal Interface or Portal Studio.

1 In the right pane, highlight the user you want to delete.

2 Right-click, and select Delete User.

3 Click Yes to confirm the deletion.

Creating and managing groups

Groups allow you to organize users in a way that is meaningful to your enterprise.

- If you grant a role to a group, all of the users in that group have that role. See “Managing a group’s roles” on page 40 for information about granting roles to groups.
- A group member may also have individual roles.

Table 3-3 describes the permissions you must have to manage groups.

Table 3-3: Permissions required to manage groups

Action	Permissions required
Create a group	WRITE on the group controlling asset.
List the groups in a domain	LIST on the group controlling asset.
View the properties of a group	READ on the group controlling asset.
Update the properties of a group	UPDATE on the group controlling asset.
Move a group to a different organization	If the new organization is in the same domain, you need READ, DELETE, and WRITE on the group controlling asset. If the organization is in a different domain, you need READ and DELETE on the group controlling asset in the current domain, and WRITE on the group controlling asset in the new domain.
Add users to, or remove users from, a group	READ and UPDATE on the group controlling asset.
Delete a group	READ and DELETE on the subject controlling asset.

❖ Creating a group

1 In the Organization Manager tree view, select the organization, and highlight Groups.

2 Click New. The Create New Group dialog box displays.

3 Enter a name for the group, and optionally, enter a description. Click OK.

Managing groups

In the Organization Manager tree view, select an organization, and highlight Groups. The groups that belong to this organization display in the right pane.

For information about the self-registration group, see “Self-registration group” on page 40.

❖ Editing a group’s properties

1. In the Organization Manager tree view, select the organization, and highlight Groups.
2. In the right pane, highlight the group, right-click, and select Edit Group. The Edit Group dialog box displays.
3. Modify the group’s properties, and click OK.

❖ Adding or removing users from a group

Before you can add users to a group, you must create the users—see “Creating and managing user accounts” on page 32.

1. In the Organization Manager tree view, select the organization, and highlight Groups.
2. In the right pane, highlight the group you want to edit, right-click, and select Edit Members. The Edit Group Members dialog box displays.
3. In the left list box, select the organization. The users that belong to this organization display in the middle list box.
4. To add a user to the group, highlight the user name, and click Add. The user name displays in the Members list box.

   Repeat this step for each user you want to add to the group.
5. To remove a user from the group, highlight the user name in the Members list box, and click Remove.
6. When you are done, click OK.

❖ Moving a group between organizations

1. In the right pane, highlight the group, right-click, and select Change Organization. The Change Group Organization dialog box displays.
2. Highlight the organization to which you want to move the group, and click OK.

❖ Managing a group's roles

If you add a user to a group that has been granted roles, the user inherits these roles. Every group member assumes a group's roles when they log in to the secured system.

Before you can grant a role to a group, you must create the group and the role. To create a group, see "Creating a group" on page 38. To create roles, see "Creating a role" on page 43.

1. In the right pane, highlight the group whose roles you want to edit, right-click, and select Edit Roles. The Edit Group Roles dialog box displays.
2. In the left list box, select an organization. The roles defined for this organization display in the center list box.
3. To grant a role, select a role in the center list box, and click Add. The role displays in the Granted Roles list. The name of the organization to which each role belongs also displays, in parentheses.

   To add another role that is defined within the same organization, repeat this step.

   To add a role that is defined in another organization, repeat steps 2 and 3.
4. To revoke a role from the group, select a role in the right list box, and click Remove.
5. Click OK.

❖ Deleting a group

1. In the right pane, highlight the group, right-click, and select Delete Group.
2. Confirm that you want to delete the selected group.

Self-registration group

New users of Sybase components can register their user information directly to the system, thus becoming a member of the self-registration group. Members of this group can access the assets that this group is allowed to access.

Every member of this group has identical permissions to access enterprise components. In an upgraded environment, all previously registered users maintain their access permission according to permissions granted to their original roles.

Enterprise Security supports only one self-registration group, which is, by default, installed into the root organization. This allows users to self-register in the root organization or any of the suborganizations.

Note If you are using Enterprise Portal, and you want roles to be granted automatically to users who self-register, grant these roles to the self-registration group—see "Managing a group's roles" on page 40.

❖ Changing the DN of the self-registration group

When you install and configure Enterprise Security, the self-registration group is added to the Enterprise Security Manager interface.

By default, the distinguished name (DN) of the self-registration group is SelfRegGroup. The PSO can change the DN by editing *security.properties*. If the PSO changes the default DN of this group, he or she must then create the group in the ACDB using Enterprise Security Manager; otherwise, attempts to self-register fail.

1. Using any standard ASCII text editor, open *security.properties*.The location depends on your application server:

   - EAServer – *JAGUAR/java/classes/com/sybase/ep/security*.
   - WebLogic – *BEA_ROOT/sybepsecurity/etc/com/sybase/ep/security*.

2. Search for this line:

```
selfRegistrationGroupName=gr\=SelfRegGroup,dc\=sybase,dc\=com
```

3. Change "SelfRegGroup" to the DN of your choice, and save the file.
4. Start Enterprise Security Manager.
5. In the middle pane, select the organization, highlight Groups, then click New.
6. Enter the group name. This name must match the name that you specified in *security.properties*. You can also enter a description, then click OK.

   The group is created, and viewable via Enterprise Security Manager.

7. Optionally, grant roles to the self-registration group. Follow the instructions for "Managing a group's roles" on page 40.

❖ Restricting the self-registration group to a suborganization

To restrict the self-registration group's access to a particular suborganization's assets, you must supply the group's full DN as part of the self-registration group name in *security.properties*, and configure the self-registration group in that suborganization.

- Following are examples of group DNs. The first defines a group in the root organization, and the second defines a group in a suborganization.

```
gr=NewSelfRegGroup,dc=Sybase,dc=com
```

```
gr=NewSelfRegGroup,ou1=subOrgA,dc=Sybase,dc=com
```

Creating and managing roles

For an overview of roles, and role hierarchy, see "Roles and groups" on page 9.

Table 3-4 describes the permissions you must have to manage roles.

Table 3-4: Permissions required to manage roles

Action	Permissions required
Create a role.	WRITE on the role controlling asset.
List the roles in a domain.	LIST on the role controlling asset.
View the properties of a role.	READ on the role controlling asset.
Update the properties of a role.	UPDATE on the role controlling asset.
Grant a role to, or revoke a role from, a user or group.	GRANT on the role controlling asset.
Assign permissions to a role to access an asset.	GRANT on either the asset or the asset controlling asset.
Move a role to a different organization.	If the new organization is in the same domain, you need READ, DELETE, and WRITE on the role controlling asset. If the organization is in a different domain, you need READ and DELETE on the role controlling asset in the current domain, and WRITE on the role controlling asset in the new domain.

Action	Permissions required
Display a role's access permissions.	LIST on the asset controlling asset in each domain where the role has permission to access assets. For example, if a role has permission to access assets in three different domains, you need LIST permission on the asset controlling asset in all three domains.
Delete a role.	READ and DELETE on the role controlling asset.

❖ Creating a role

1. In the Organization Manager tree view, select the organization, and highlight Roles.
2. Click New. The Create New Role dialog box displays.
3. Enter a name for the role. To enable EAServer to use implicit role mapping, each role name must be unique throughout the security system. See "Implicit role mapping" on page 146 for more information.
4. Optionally, enter a description, then click OK.

Managing roles

In the Organization Manager tree view, select an organization, and highlight Roles. The roles that belong to this organization display in the right pane.

❖ Editing a role's properties

1. In the right pane, highlight the role, right-click, and select Edit Role.
2. In the Edit Role dialog box, edit any of these fields, then click OK:
   - Role Name – the name of the role. To enable Enterprise Security to map roles automatically to Enterprise Portal roles, the role name must be unique throughout the security system. For more information, see "Implicit role mapping" on page 146.
   - Role DN – the distinguished name of the role.

     Note If you change a role's DN, and a role mapping exists for this role in the *security.properties* file, update the mapping to use the new DN.

   - Description – a description of the role.

❖ Moving a role between organizations

1. In the right pane, highlight a role, right-click, and select Change Organization. The Change Role Organization dialog box displays.
2. In the To New Organization dialog box, highlight the organization to which you want to move the role, and click OK.

❖ Managing a role’s access permissions

To enable users to access secured assets, assign permissions to roles, and grant these roles to users or groups. To manage permissions for a role:

1. In the right pane, highlight the role you want to edit, right-click, and select Manage Access Permissions. The Manage Role Access Permission dialog box displays.

   The Access Permissions Granted for the Role list at the bottom of the window displays the assets that this role can access, and which permissions it has.

2. In the left list box, select the organization. The assets that belong to this organization display in the adjacent list box.
3. To assign permissions, highlight an asset, then select the permission you want to add in the Available Permissions list, and click Add. The permission displays as an Assigned Permission. To add all permissions, select Add All.

   Note If you assign READ permission, assign LIST permission also so that users with READ permission can list the object using the Queries bean.

   Repeat this step for each asset that you want this role to have permission to access.

   To assign permissions to assets in other organizations, repeat steps 2 and 3.

   To remove a permission, highlight an asset, then select the permission you want to remove in the Assigned Permissions list, and click Remove.

4. To save your changes, click OK.

Note To manage access to the controlling assets, see “Managing permissions to access controlling assets” on page 102.

❖ Managing roles for users

1. In the Organization Manager tree view, select an organization, and highlight Roles.
2. In the right pane, highlight the role name, right-click, and select Manage User Roles. The Manage User Roles dialog box displays.
3. In the left list box, select the organization. The users that belong to this organization display in the adjacent list box.
4. To grant the role to a user, highlight the user name, and click Add. The user name displays in the Grant Role To list box. The name of the organization to which each user belongs also displays, in parentheses.

   The Inherited By list box displays the roles this user inherits via his or her group memberships.

   Repeat this step for each user to which you want to grant the role.

   To grant the role to users in other organizations, repeat steps 3 and 4.

   To revoke the role from a user, highlight the user name in the Grant Role To list box, and click Remove.
5. To save your changes, click OK.

❖ Managing roles for groups

1. In the right pane, highlight the role you want to edit, right-click, and select Manage Group Roles. The Manage Group Roles dialog box displays.
2. In the left list box, select an organization. The groups defined in this organization display in the center list box.
3. To grant the role to a group, select the group in the center list box, and click Add. The role displays in the Grant Role To list box. The name of the organization to which each group belongs also displays, in parentheses

   To grant the role to a group that is defined in another organization, repeat steps 2 and 3.

   To revoke the role from a group, select the group in the Grant Role To list box, and click Remove.
4. Click OK.

The PortalGuest role and the guest account

Enterprise Security includes a guest account for Enterprise Portal. After you install and configure Enterprise Security, the guest account allows users to log in to your portal. The login name for the guest account is “guest”; the password is also “guest”.

Warning! Do not delete the guest account. It is required for Enterprise Portal.

Enterprise Security automatically creates the PortalGuest role and grants the role to the guest account. The sole purpose of the PortalGuest role is to enable self-registration. The guest account does not have permission to access any assets. The system can have only one guest role. The guest role is valid for an indefinite period.

Creating and managing assets

An asset can represent a server, a portlet, a URL, or a resource in the back-end system, such as a database table.

Table 3-5 describes the permissions you must have to manage assets.

Table 3-5: Permissions required to manage assets

Action	Permissions required
Create an asset	WRITE on the asset controlling asset.
List the assets in a domain	LIST on the asset controlling asset.
View the properties of an asset	READ on the asset controlling asset.
Update the properties of an asset	UPDATE on the asset controlling asset.
Move an asset to a different organization	If the new organization is in the same domain, you need READ, DELETE, and WRITE on the asset controlling asset. If the organization is in a different domain, you need READ and DELETE on the asset controlling asset in the current domain, and WRITE on the asset controlling asset in the new domain.
Delete an asset	READ and DELETE on the asset controlling asset.

❖ Creating an asset

Use unique names when you create assets. Do not use the name of an existing root object for a new asset; for example, do not create an asset named “User”.

1. In the Organization Manager tree view, select the organization, and highlight Assets.
2. Click New. The Create New Asset dialog box displays.

   Enter the asset name, and optionally, enter a description. If the asset is a URL, select URL as the Asset Type.
3. Click OK. The name of the new asset displays in the right pane.

Managing assets

In the Organization Manager tree view, select an organization, and highlight Assets. The assets that belong to this organization display in the right pane.

❖ Editing an asset’s properties

1. In the right pane, highlight an asset, right-click, and select Edit Asset. The Edit Asset dialog box displays.
2. Edit any of these values, then click OK
   - Asset Name – name of the asset.
   - Asset DN – distinguished name of the asset.

- Asset Type – select the asset type from a list.
- Asset Description – an optional description of the asset.

To edit the asset's access permissions, see "Managing permissions to access an asset," below.

❖ Managing permissions to access an asset

To allow a user to access an asset, you must assign access permission to a role, and then grant the role to the user.

1. In the right pane, highlight an asset, right-click, and select Manage Access Permission. The Manage Access Permission on Asset dialog box displays.
2. In the left list, select the organization. In the adjacent list, select the role. The Available Permissions list box displays the permissions available for this asset. The Assigned Permissions list displays the permissions assigned to the selected role.
3. To assign permissions to this role, highlight a permission in the Available Permissions list, and click Add. The permission displays in the Assigned Permissions list.

   To assign all the available permissions to this role, click Add All.

   All of this role's assigned permissions to access this asset display in the Access Permissions Granted on the Asset list, at the bottom of the window.

   To assign permissions to access this asset to other roles, repeat steps 2 and 3.

   To remove permissions, highlight the permission you want to remove in the Assigned Permissions list, and click Remove.

   To remove all this role's permissions for this asset, click Remove All.

   For each role from which you want to remove access permissions, repeat steps 2 and 3.
4. To save your changes, click OK.

❖ Moving an asset between organizations

1. In the right pane, right-click, and select Change Organization. The Change Asset Organization dialog box displays.
2. Select the organization to which you want to move the asset, and click OK.

❖ Deleting an asset

1. In the right pane, highlight an asset, right-click, and select Delete Asset.

2 Confirm that you want to delete the asset.

Managing user-based proxy authentication information

Once an asset is part of the secured enterprise system, the PSO must assign READ access to the asset so users can use proxy authentication. This section describes how to create user-based proxy authentication information using Enterprise Security Manager. To create role- or asset-based proxy authentication information, you must use SMAPI—see Chapter 9, “Proxy Authentication.”

Table 3-6 describes the permissions you must have to manage user-based proxy authentication information.

Table 3-6: Permissions required to manage proxy authentication information

Action	Permissions required
Create user-based proxy authentication information	UPDATE on the subject controlling asset.
View proxy authentication information	READ on the subject controlling asset.
Edit proxy authentication information, including a user’s proxy authentication password	UPDATE on the subject controlling asset.
Delete user-based proxy authentication information	UPDATE on the subject controlling asset.

❖ **Managing user-based proxy authentication information**

1 In the Organization Manager tree view, select an organization, and highlight Users.

2 In the right pane, highlight a user, right-click, and select Proxy Authentication. The Manage User Proxy Authentication Information dialog box displays.

3 To create proxy authentication information for the user, see “Creating user-based proxy authentication information” on page 50.

To edit an entry, see “Editing user-based proxy authentication information” on page 50.

To delete an entry, highlight the entry, and click Delete.

For complete information about using proxy authentication to implement single sign-on to enterprise resources, see Chapter 9, “Proxy Authentication.”

❖ Creating user-based proxy authentication information

Before you can create proxy authentication information, users and assets must exist in the enterprise environment. To define a user’s proxy authentication information for an asset:

1 In the Manage User Proxy Authentication Information dialog box, click New. The Create User Proxy Authentication Information dialog box displays.

Enter:

- Asset DN – the distinguished name for the asset. To look up the DN, click Find.

  In the Choose Asset dialog box, select the organization to which the asset belongs, highlight the asset name, then click OK. The asset DN displays in the Create User Proxy Authentication Information dialog box.
- Service URL – the connection request uses this URL to establish a connection with the selected asset.
- User Name – the name used to log in to the asset specified by the URL.
- Password – a valid password for the user name.
- Verify Password – the same password.

2 Click OK to save your changes. The asset now has proxy authentication information associated with it.

Repeat this procedure for each asset for which you want to create proxy authentication information.

❖ Editing user-based proxy authentication information

1 In the Manage User Proxy Authentication Information dialog box, highlight the entry you want to change, and click Edit. The Edit User Proxy Authentication Information dialog box displays.

2 Edit any values you want to change. For a description of the fields, see “Creating user-based proxy authentication information” on page 50. Click OK to save your changes.

❖ **Changing a user's proxy authentication password**

1. In the Manage User Proxy Authentication Information dialog box, highlight the entry for which you want to change the password, and click Change Password. The Change User Proxy Authentication Password dialog box displays.
2. Enter the new password twice, and click OK.

Using the encryption/decryption key file

When you install Enterprise Security, a key file (*.enk*) is generated and stored as a hidden file in the *Security* directory.

The key file stores the key that is used to encrypt or decrypt information in the ACDB. Only the PSO can change the key file, so the PSO must protect this file from other users' read and modify attempts.

Enterprise Security includes a tool (securetool) that allows you to replace the key file with a new key file, then reencrypt the data that was encrypted with the old key. When you replace the key file, you must have the original encryption/decryption key.

❖ **Updating the security key file and reencrypting system data**

1. Restrict users from logging in to Enterprise Portal or EAServer until the key file is updated.
2. In the *Security\bin* directory, run:

   ```
   securetool changekey --enkfile <path_to_enk>\.enk
      --output_enkfile <path_to_enk>\.enk
      [--random_seed n]
   ```

   Where `path_to_enk` is the path to the *.enk* file, and *n* is an integer to use as a random seed.
3. Reencrypt the system data using the new encryption key:

   ```
   reencsysdata --appserver_url URL
      --username user_name  --password password
      [--init_ctx_factory initCtxtFactory]
   ```

Where *URL*, *user_name*, and *password* are the URL, user name and password for connecting to the security middleware. Optionally, provide an InitialContextFactory; the default is com.sybase.ep.security.naming.InitialContextFactory.

Warning! If you change the key file (*.enk*) in a clustered EAServer environment, you must manually copy the new key file to each machine that has Enterprise Security installed.

4 Lock the *.enk* file at the operating system level (using file access permissions) to prevent anyone from viewing or accessing the new key.

5 Remove any restrictions that you imposed to prevent users from logging in to the system. You may now allow user access based on the new key.

Enabling an authorization data store other than the ACDB

You can store authorization information in a data store other than the ACDB. The authorization delegate establishes a connection to the data store where the authorization information is stored, and retrieves authorization information.

To configure the authorization delegate to access an alternate data store:

1 Create a custom EJB that implements the com.sybase.ep.security.authorization.AuthorizationDelegate local interface, and deploy it to your application server. For information about this interface, see the SMAPI documentation. For information about implementing custom EJBs, see the *EAServer Programmer's Guide*.

   The default configuration included with the product uses the com.sybase.ep.security.authorizationdelegate/ACDBAuthDelegate component.

2 Disable implicit role mapping. Edit the *security.propeties* file, and set the value of the defaultRolemappingEnabled property to false.

3 Restart your application server for the changes to take effect.

4 In the com.sybase.ep.security.sessionsvcs/PortalSession component, reassign the ejb/AuthorizationDelegate local EJB reference to point to your newly created EJB component. For EAServer, see "Creating Web Applications" in the *EAServer Programmer's Guide*. For WebLogic, see the BEA documentation.

If you use a custom authorization delegate to access a data store other than the ACDB, Enterprise Security Manager will not work. As an alternative, you can create your own management tools to configure Enterprise Security services.

If you use the authorization delegate to access another data store, any programming that you have done using either the existing Security Object Management API or SCAPI (a framework that allows cross-platform development) will not work.

Note If you write a custom authorization delegate that does not use the standard ACDB schema, you must disable implicit role mapping—see "Implicit role mapping" on page 146.

CHAPTER 4 Using securetool

securetool is a command line interface that allows you to configure the Enterprise Security middleware and databases.

This chapter contains instructions on how to use securetool.

Working with securetool

Use the following scripts to run securetool:

- **UNIX or Linux** *$SECURITY/bin/securetool.sh*
- **Windows** *%SECURITY%\bin\securetool.bat*

securetool syntax

When using securetool, you can specify argument/value pairs either on the command line or in a file.

The syntax for securetool is:

```
securetool taskname [@response_file] [--argument value] [...]
```

Where:

- *taskname* is a securetool task described in “securetool tasks” on page 58. Task names are not case sensitive.
- *response_file* specifies the name of an optional text file that contains argument/value pairs. When using a response file, arguments that do not apply to the current task are ignored, so you can use the same response file for many different tasks.

- *argument* is a securetool argument. The arguments vary according to the task. For a list of the arguments for each task, see the individual task names under "securetool tasks" on page 58. On the command line, precede each argument with two hyphens; in a response file, do not use hyphens. Argument names are not case sensitive.
- *value* is the argument's value.

  You can specify both argument/value pairs and a response file on the command line. If conflicts occur, command line argument/value pairs override those in a response file.

Using securetool

This section describes how to use securetool to install and uninstall Enterprise Security, and how to upgrade the middleware using a sample response file. The response file includes all the argument/value pairs that are required to perform these tasks.

Sample response file

The following lines in a response file allow you to install and configure a new Enterprise Security installation, assuming there are locally installed versions of Adaptive Server and EAServer that use the default port numbers and path names:

```
rootorg_name=Sybase.com
enkfile=C:\Sybase\Security\.enk
output_enkfile=C:\Sybase\Security\.enk
security_dir=C:\Sybase\Security
easerver_dir=C:\Sybase\EAServer
shared_dir=C:\Sybase\Security\shared_libs
rootrog_dn=dc\=sybase.com
dns_domain=sybase.com
```

To use this response file with the following procedures, create a file called *configure_security.rsp* that contains the preceding lines.

❖ Manually installing Enterprise Security

To manually install Enterprise Security using *configure_security.rsp*, run the following commands.

1 To generate an encryption key file:

```
securetool genenk @configure_security.rsp
```

2 To create a new ACDB:

```
securetool createdb @configure_security.rsp
```

3 To deploy the Enterprise Security middleware into EAServer:

```
securetool deploymw @configure_security.rsp
```

4 To deploy Enterprise Security Manager into EAServer:

```
securetool deploysm @configure_security.rsp
```

❖ Upgrading the Enterprise Security middleware

To upgrade the middleware using *configure_security.rsp*, run:

```
securetool deploymw @configure_security.rsp
```

❖ Uninstalling Enterprise Security

To manually uninstall the software, perform the following tasks using *configure_security.rsp*.

1 To remove Enterprise Security Manager from EAServer:

```
securetool removesm @configure_security.rsp
```

2 To remove the middleware from EAServer:

```
securetool removemw @configure_security.rsp
```

This is necessary because typically, EAServer holds a database connection open.

3 To remove the database:

```
securetool removedb @configure_security.rsp
```

Before you remove the database, verify that there are no processes with database connections.

4 Manually delete the encryption key file.

5 Delete the *Security* directory.

securetool help

To display the list of securetool tasks and the command line syntax, enter:

```
securetool
```

To display a list of the valid arguments for any securetool task, enter:

```
securetool taskname help
```

Debugging information

Debugging information is written to the *securetool.log* file, in the same directory where you run securetool. Each time you run securetool, debugging information is appended to *securetool.log*, and the output is verbose, so you may want to monitor the size of this file, as it can grow quickly.

securetool tasks

This section contains information on securetool tasks; the task names are listed alphabetically. Each task has a brief description, a list of options, and an example of its usage.

Each task includes a table that lists the argument names, datatypes, default values (if they exist), and brief descriptions. The arguments that are required are identified by an asterisk to the left of the argument name.

Task name	Description
changekey	Creates a new encryption key file
createdb	Creates the ACDB
createschema	Creates the schema for the ACDB
deploymw	Deploys the security middleware to EAServer
deploysm	Deploys Enterprise Security Manager to EAServer
domainrules	Retrieves or modifies security domain rules
enc_dec_file	Encrypts or decrypts a file
genenk	Generates an encryption key file
populatedb	Populates the ACDB
querydb	Retrieves ACDB statistics
reencsysdata	Reencrypts system data
removedb	Removes the ACDB
removemw	Removes the security middleware from EAServer
removesm	Removes Enterprise Security Manager from EAServer
upgradedb	Upgrades the ACDB
	Configures the security middleware in WebLogic
wls_deploymw	Deploys the security middleware to WebLogic
wls_deploysm	Deploys Enterprise Security Manager to WebLogic
wls_removemw	Removes the security middleware from WebLogic

Task name	Description
wls_removesm	Removes Enterprise Security Manager from WebLogic

changekey

Description

Saves the old encryption key file, and generates a new encryption key file.

Syntax

changekey --enkfile *file_name* --output_enkfile *new_file_name*
[--random_seed *number*]

Argument	Datatype	Default value	Description
* enkfile	string		Existing encryption key file; must be readable.
* output_enkfile	string		Encryption key file to create; must be writable.
random_seed	integer		An integer to use as a random seed.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example saves the existing encryption key file (*.enk*) and creates a new encryption key file (*.enk2*) in the *C:\Sybase\Security* directory:

```
securetool changekey --enkfile C:\Sybase\Security\.enk
   --output_enkfile C:\Sybase\Security\.enk2
```

See also

genenk on page 73
reencsysdata on page 77

createdb

Description

Creates the ACDB.

Syntax

createdb --rootorg_dn *org_DN* --rootorg_name *org_name*
[--asadb_acdb_owner_group *acdb_owner*]
[--asedb_datadevice *datadevice*]
[--asedb_datafile *filename*]
[--asedb_datasize *dbsize*]
[--asedb_logdevice *logdevice*]

```
[--asedb_logfile logfile]
[--asedb_logsize logsize]
[--asedb_pagesize pagesize]
[--asedb_trunc_on_checkpoint true | false]
[--database_type sybase_ase | sybase_asa | oracle]
[--entldb_password password]
[--entldb_username user_name]
[--guest_password password]
[--jdbc_admin_password password]
[--jdbc_admin_username user_name]
[--jdbc_driver driverName]
[--jdbc_url jdbcURL]
[--oracle_data_tablespace dataSpace]
[--oracle_index_tablespace indexSpace]
[--populate_only true | false]
[--portaladmin_password password]
[--pso_password password]
[--psoemail email]
[--psoname psoName]
[--psophone phone]
[--psouid psoUID]
[--random_seed seed]
[--rootorg_address address]
[--rootorg_contact contact]
[--rootorg_desc description]
[--sybase_asa_servicename service_name]
[--webplugin_password password]
```

Argument	Datatype	Default value	Description
* rootorg_dn	string		Root organization DN. This is usually derived from the domain of the server. For example, a server in the domain MyCompany.com might have a root organization DN of dc=MyCompany,dc=com.
* rootorg_name	string		Root organization name.
asadb_acdb_owner_group	string	ACDB_owner	The ACDB group owner that is created when you install the database. This group owns the ACDB tables and procedures.
asedb_datadevice	string	acdbData	The Adaptive Server Enterprise device on which the database data segment is created. This device will be created if it does not exist.
asedb_datafile	string		Adaptive Server database data device file name.
asedb_datasize	integer	100MB	Adaptive Server database data device file size in MB.

Argument	Datatype	Default value	Description
asedb_logdevice	string	acdbLog	The Adaptive Server Enterprise device on which the database log segment is created. This device will be created if it does not exist.
asedb_logfile	string		Adaptive Server database log device file name.
asedb_logsize	integer	25MB	Adaptive Server database log device file size in MB.
asedb_pagesize	integer	2048	Adaptive Server database page size, which is used to determine the number of pages in existing devices. **Note** This must match the value on the Adaptive Server Enterprise database server; otherwise, devices are improperly sized. This value does not change the page size on the database server.
asedb_trunc_on_checkpoint	boolean	true	Adaptive Server database option to enable truncate on checkpoint.
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
entldb_password	string	dbopswd	Password of the user specified by entldb_username.
entldb_username	string	acdbdbo	User name for connecting to the security database. This user should have read and write privileges on the security database only.
guest_password	string	guest	Password for the guest user.
jdbc_admin_password	string		Password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	User name for connecting to the database for administrative purposes (creating databases, tables, and so on).

Argument	Datatype	Default value	Description
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	JDBC driver to use when connecting to the database. To load Oracle drivers automatically, add the name of the JAR file containing the Oracle drivers (typically, *classes12.zip*) to the SECURETOOL_CLASSPATH environment variable.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.
oracle_data_tablespace	string	ACDB_DATA	The tablespace on which the ACDB tables are created.
oracle_index_tablespace	string	ACDB_IDX	The tablespace on which the ACDB indexes are created/
populate_only	boolean	false	If using a JDBC database with an existing schema, set to true. In this case, only the initial data is inserted or removed.
portaladmin_password	string	sybase	Password for the Portal Administrator.
pso_password	string	123qwe	Password for the user specified by psouid.
psoemail	string		Security Officer’s e-mail address.
psoname	string	Portal Security Officer	Name of the Security Officer.
psophone	string		Security Officer’s phone number.
psouid	string	pso	Security Officer’s user ID.
random_seed	integer		An integer to use as the random seed.
rootorg_address	string		Root organization’s address.
rootorg_contact	string		Root organization contact name.
rootorg_desc	string		Root organization description.
sybase_asa_servicename	string	acdb	The SERVICENAME connection property to use when connecting to a Sybase ASA database.
webplugin_password	string	sybase	Web server plug-in user password.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example creates an ACDB for the root organization Sybase.com:

```
   securetool createdb --rootorg_name Sybase.com --rootorg_dn dc=sybase,dc=com
```

See also

createschema on page 63
populatedb on page 74
querydb on page 76
removedb on page 78
upgradedb on page 82

createschema

Description

Creates the schema for the ACDB.

Syntax

createschema
[--asadb_acdb_owner_group *acdb_owner*]
[--asedb_datafile *filename*]
[--asedb_datasize *dbsize*]
[--asedb_logfile *logfile*]
[--asedb_logsize *logsize*]
[--asedb_pagesize *pagesize*]
[--asedb_trunc_on_checkpoint *true* | *false*]
[--database_type *sybase_ase* | *sybase_asa* | *oracle*]
[--entldb_password *password*]
[--entldb_username *user_name*]
[--jdbc_admin_password *password*]
[--jdbc_admin_username *user_name*]
[--jdbc_driver *driverName*]
[--jdbc_url *jdbcURL*]
[--oracle_data_tablespace *dataSpace*]
[--oracle_index_tablespace *indexSpace*]

Argument	Datatype	Default value	Description
asadb_acdb_owner_group	string	ACDB_owner	The ACDB group owner that is created when you install the database. This group owns the ACDB tables and procedures.

Argument	Datatype	Default value	Description
asedb_datafile	string		Adaptive Server database data device file name.
asedb_datasize	integer	100MB	Adaptive Server database data device size, in MB. On an existing data device, this much space must exist on the device to create a new database of this size. **Note** The value of asedb_pagesize must be set correctly.
asedb_logfile	string		Adaptive Server database log device file name.
asedb_logsize	integer	25MB	Adaptive Server database log device file size, in MB. On an existing data device, this much space must exist on the device to create a new database log of this size. **Note** The value of asedb_pagesize must be set correctly.
asedb_pagesize	integer	2048	Adaptive Server database page size, which is used to determine the number of pages in existing devices. **Note** This must match the value on the Adaptive Server Enterprise database server; otherwise, devices are improperly sized. This value does not change the page size on the database server.
asedb_trunc_on_checkpoint	boolean	true	Adaptive Server database option to enable truncate on checkpoint.
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle

Argument	Datatype	Default value	Description
entldb_password	string	dbopswd	Password of the user specified by entldb_username.
entldb_username	string	acdbdbo	User name for connecting to the security database. This user should have read and write privileges on the security database only.
jdbc_admin_password	string		Password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	User name for connecting to the database for administrative purposes (creating databases, tables, and so on).
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	JDBC driver to use when connecting to the database. To load Oracle drivers automatically, add their names to the SECURETOOL_CLASSPATH environment variable.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.
oracle_data_tablespace	string	ACDB_DATA	The tablespace on which the ACDB tables are created.
oracle_index_tablespace	string	ACDB_IDX	The tablespace on which the ACDB indexes are created/

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example creates the schema for the ACDB using the default values:

```
securetool createschema
```

See also

createdb on page 59
populatedb on page 74
querydb on page 76
removedb on page 78
upgradedb on page 82

deploymw

Description

Deploys the Enterprise Security middleware to EAServer.

Syntax

```
deploymw --easerver_dir eas_dir --enkfile file_name
            --security_dir secure_dir --shared_dir share_dir
[--audit_topic topic_name]
[--database_type sybase_ase | sybase_asa | oracle]
[--easerver_host host_name]
[--easerver_password password]
[--easerver_port port]
[--easerver_restart true | false]
[--easerver_servername server]
[--easerver_username user_name]
[--entldb_password password]
[--entldb_username user_name]
[--jdbc_admin_password password]
[--jdbc_admin_username user_name]
[--jdbc_driver driver]
[--jdbc_url URL]
```

Argument	Datatype	Default value	Description
* easerver_dir	string		EAServer installation directory.
* enkfile	string		Encryption key file to use; must be a readable file.
* security_dir	string		Enterprise Security installation directory.
* shared_dir	string		Sybase shared directory.
audit_topic	string	AuditTopic	The preconfigured EAServer message service topic to use when publishing audit messages (if enabled).

Argument	Datatype	Default value	Description
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
easerver_host	string		Name of the machine where EAServer is running.
easerver_password	string		Password for the user identified by easerver_username.
easerver_port	integer	9000	EAServer connection port.
easerver_restart	boolean	true	Specifies whether to restart EAServer when this task completes.
easerver_servername	string	Jaguar	Name of the server in which to install the software.
easerver_username	string	jagadmin	User name for the EAServer administrator.
entldb_password	string	dbopswd	Password of the user specified by entldb_username.
entldb_username	string	acdbdbo	User name for connecting to the database. This user should have read and write privileges on the security database only.
jdbc_admin_password	string		Password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	User name for connecting to the database for administrative purposes (creating databases, tables, and so on).
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	JDBC driver to use when connecting to the database.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example deploys the Enterprise Security middleware to EAServer. The arguments provide the name of the encryption key file, the installation locations of Enterprise Security and EAServer, and the location of the Sybase shared directory.

```
securetool deploymw --easerver_dir C:\Sybase\EAServer
   --enkfile C:\Sybase\Security\.enk
   --security_dir C:\Sybase\Security
   --shared_dir C:\Sybase\Shared-1_0
```

See also

deploysm on page 68
removemw on page 80

deploysm

Description

Deploys the Enterprise Security Manager software to EAServer.

Warning! Running this command on a machine where you have Enterprise Portal installed removes your portal.

Enterprise Portal installs Enterprise Security Manager automatically.

Syntax

```
deploysm --dns_domain domain --easerver_dir eas_dir
        --security_dir secure_dir
[--easerver_host host_name]
[--easerver_port port]
[--easerver_restart true | false]
[--easerver_servername server]
[--easerver_password password]
[--easerver_username user_name]
[--http_port http_port#]
[--https_port https_port#]
[--overwrite yesno]
```

Argument	Datatype	Default value	Description
* dns_domain	string		DNS domain of the servers, used for constructing a fully-qualified host name.

Argument	Datatype	Default value	Description
* easerver_dir	string		EAServer installation directory.
* security_dir	string		Enterprise Security installation directory.
easerver_host	string		Name of the machine where EAServer is running.
easerver_port	integer	9000	EAServer connection port.
easerver_password	string		Password for the user identified by easerver_username.
easerver_restart	boolean	true	Specifies whether to restart EAServer when this task completes.
easerver_servername	string	Jaguar	Name of the server in which to install the software.
easerver_username	string	jagadmin	User name for the EAServer administrator.
http_port	integer	8080	The application server’s default HTTP port.
https_port	integer	8081	The application server’s default HTTPS port.
overwrite	boolean	false	If true, overwrites an existing Enterprise Security Manager.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example deploys Enterprise Security Manager to EAServer. The arguments provide the DNS domain, the EAServer and Enterprise Security installation directories, the EAServer HTTP and HTTPS port numbers, and the host name and IIOP port number where the middleware is installed.

```
securetool deploysm --dns_domain sybase.com --easerver_dir /work/EAServer
--security_dir /work/Security
```

See also

deploymw on page 66
removesm on page 81

domainrules

Description

Retrieves, changes, or removes security domain rules. To run this command, the user must have appropriate permission within the specified domain—see "Managing security domains and policies" on page 94.

Domain rule values of null differ from unspecified values. To set a rule value to null, use the string "<null>". If you retrieve a rule with a null value, the return value is also "<null>".

Syntax

```
domainrules --appserver_url URL --password password --username login
[--domain domainName]
[--init_ctx_factory ctxFactory]
[--input_property_file inputFile]
[--operation <get | set | remove | reset>]
[--output_property_file outputFile]
[--rule_name property]
[--rule_value value]
```

Argument	**Datatype**	**Default value**	**Description**
* appserver_url	string		The URL to connect to the Enterprise Security middleware.
* password	string		The password to use when connecting to the Enterprise Security middleware.
* username	string		The user name to use when connecting to the Enterprise Security middleware.
domain	string	DefaultDomain	The name of the domain on which to perform the operation (get, set, remove, or reset).
init_ctx_factory	string	com.sybase.ep.security. naming.InitialContextFactory	The InitialContextFactory to use when connecting to the Enterprise Security middleware.
input_property_file	string		A file that contains a list of rules and their values, in Java properties format; must be a readable file. Use an input property file when you want to set or remove multiple rules. To set or remove a single rule, specify rule_name and rule_value; do not use this argument.
operation	choice	get	The operation to perform: get, set, remove, or reset. The reset operation resets all domain rules to their default values. You cannot reset rules individually.

Argument	Datatype	Default value	Description
output_property_file	string		The name of the file where the requested rules are written in Java properties format. If not specified, property names and values are written to the console.
rule_name	string		The name of the rule to retrieve, set or remove. This argument cannot be used in conjunction with input_property_file. If you specify both, securetool displays a warning message and quits.
rule_value	string		Sets the rule to this value. If the operation is get or remove, or if you specify an input_property_file, this property is ignored.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

Example 1 This command line example sets the domain rules specified in the */work/SECURITY/defaultdomain.txt* file for the DefaultDomain, where *URL* is the URL to connect to the Enterprise Security middleware:

```
securetool domainrules --appserver_url URL --username pso --password 123qwe
   --operation set --input_property_file /work/Security/defaultdomain.txt
```

Example 2 This command line example sets the auditJMSEnable domain rule to true, which sends auditing notifications to a JMS message topic:

```
securetool domainrules --appserver_url iiop://hostname:9000
   --username pso --password 123qwe
   --operation set --rule_name auditJMSEnable --rule_value true
```

Warning! Do not set auditJMSEnable to true until after you set up both the message service and the message topic in your application server; otherwise, you will not be able to log in to Enterprise Portal—see “Setting up JMS auditing notifications for EAServer” on page 133.

Example 3 This example uses two commands to set the auditing filters, `auditIncludeFilter` and `auditExcludeFilter`, to define which events to audit for the domain:

```
securetool domainrules --appserver_url iiop://victory:9000
   --username pso --password 123qwe --operation set
   --rule_name auditIncludeFilter --rule_value "(ResourceClass=SYSTEM.*)"

securetool domainrules --appserver_url iiop://victory:9000
   --username pso --password 123qwe --operation set
   --rule_name auditExcludeFilter --rule_value "(Decision=Permit)"
```

See also

Chapter 5, “Delegated Administration”

enc_dec_file

Description

Encrypts or decrypts a file using the security middleware and user credentials.

Syntax

```
enc_dec_file --appserver_url URL
   --input_file read_file --output_file write_file
   --operation [encrypt | decrypt]
   --username user_name --password sm_password
[--init_ctx_factory initialCtxtFactory]
```

Argument	Datatype	Default value	Description
* appserver_url	string		The URL to use when connecting to the security middleware.
* input_file	string		The name of the file to encrypt or decrypt; must be a readable file.
* output_file	string		The destination file for the encrypted or decrypted data; must be a writable file.
* operation	choice		The operation to perform, encrypt or decrypt.
* username	string		The user name for connecting to the security middleware.
* password	string		The password for connecting to the security middleware.
init_ctx_factory	string	com.sybase.ep.security.naming.InitialContextFactory	The InitialContextFactory to use when connecting to the security middleware.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This example encrypts the input file *myFile.txt* and writes the output to the *myFile.txt.enk* file; the user name is “pso” and the password is “pso”:

```
securetool enk_dec_file
   --appserver_url iiop://victory:9000
   --operation encrypt --username pso --password pso
   --input_file myFile.txt --output_file myFile.txt.enk
```

See also

genenk on page 73
reencsysdata on page 77

genenk

Description

Generates a new encryption key file.

Syntax

genenk --output_enkfile *file_name*
[--random_seed *number*]

Argument	Datatype	Default value	Description
* output_enkfile	string		Encryption key file to create; must be writable.
random_seed	integer		An integer to use as the random seed.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example generates a new encryption key file (*.enk*) in the *C:\Sybase\Security* directory:

```
securetool genenk --enkfile C:\Sybase\Security\.enk
```

See also

enc_dec_file on page 72
reencsysdata on page 77

populatedb

Description Populates the ACDB.

Syntax

```
populatedb --rootorg_dn org_DN --rootorg_name org_name
[--database_type sybase_ase | sybase_asa | oracle]
[--guest_password password]
[--jdbc_admin_password password]
[--jdbc_admin_username user_name]
[--jdbc_driver driverName]
[--jdbc_url jdbcURL]
[--portaladmin_password password]
[--pso_password password]
[--psoemail email]
[--psoname psoName]
[--psophone phone]
[--psouid psoUID]
[--random_seed seed]
[--rootorg_address address]
[--rootorg_contact contactName]
[--rootorg_desc description]
[--sybase_asa_servicename service_name]
[--webplugin_password password]
```

Argument	Datatype	Default value	Description
* rootorg_dn	string		Root organization DN. This is usually derived from the domain of the server. For example, a server in the domain MyCompany.com might have a root organization DN of dc=MyCompany,dc=com.
* rootorg_name	string		Root organization name.
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
guest_password	string	guest	Password for the guest user.
jdbc_admin_password	string		Password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	User name for connecting to the database for administrative purposes (creating databases, tables, and so on).

Argument	Datatype	Default value	Description
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	JDBC driver to use when connecting to the database.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.
portaladmin_password	string	sybase	Password for the Portal Administrator.
pso_password	string	123qwe	Password for the user specified by psouid.
psoname	string	Portal Security Officer	Name of the Security Officer.
psophone	string		Security Officer’s phone number.
psoemail	string		Security Officer’s e-mail address.
psouid	string	pso	Security Officer’s user ID.
random_seed	integer		An integer to use as a random seed.
rootorg_address	string		Root organization’s address.
rootorg_contact	string		Root organization contact name.
rootorg_desc	string		Description of the root organization.
sybase_asa_servicename	string	acdb	The SERVICENAME connection property to use when connecting to a Sybase ASA database.
webplugin_password	string	sybase	Password for anyone using the Web server plug-in.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example populates the ACDB for the root organization Sybase.com:

```
securetool populatedb --rootorg_name Sybase.com --rootorg_dn
```

```
dc=sybase,dc=com
```

Note Before you populate the ACDB, you must create the ACDB schema—see createschema on page 63.

See also

createdb on page 59
createschema on page 63
querydb on page 76
removedb on page 78
upgradedb on page 82

querydb

Description

Retrieves the version number and additional status information of the ACDB.

Syntax

querydb
[--database_type *sybase_ase* | *sybase_asa* | *oracle*]
[--jdbc_admin_username *user_name*]
[--jdbc_admin_password *password*]
[--jdbc_driver *driverName*]
[--jdbc_url *jdbcURL*]

Argument	Datatype	Default value	Description
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
jdbc_admin_username	string	sa	The user name for connecting to the database for administrative purposes (creating databases, tables, and so on).
jdbc_admin_password	string		The password for the user specified by jdbc_admin_username.
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	The JDBC driver to use when connecting to the database.

Argument	Datatype	Default value	Description
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*: 5000/acdb	The JDBC URL to use when connecting to the database.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.
2	The database does not exist.
3	The database is not populated. Use populate_only option with createdb task.
4	The database schema is old. Upgrade the database.

Examples

This command line example retrieves the version number of the ACDB using the default values for the JDBC driver, JDBC URL, user name, and password:

```
securetool querydb
```

See also

createdb on page 59
createschema on page 63
populatedb on page 74
removedb on page 78
upgradedb on page 82

reencsysdata

Description

Reencrypts system data using the new encryption key. To perform this task, you must have update permission on the domain controlling asset in the domain that contains the root organization; typically, this is assigned only to the PSO.

Syntax

reencsysdata --appserver_url *URL*
--username *user_name* --password *password*
[--init_ctx_factory *initialCtxtFactory*]

Argument	Datatype	Default value	Description
* appserver_url	string		The URL to use when connecting to the security middleware.
* username	string		The user name for connecting to the security middleware.
* password	string		The password for connecting to the security middleware.
init_ctx_factory	string	com.sybase.ep.security.naming.InitialContextFactory	The InitialContextFactory to use when connecting to the security middleware.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example reencrypts the system data:

```
securetool reencsysdata --appserver_url URL --username pso --password 123qwe
```

See also

enc_dec_file on page 72
genenk on page 73

removedb

Description

Removes the ACDB.

Syntax

removedb
[--asadb_acdb_owner_group *acdb_owner*]
[--asedb_datadevice *datadevice*]
[--asedb_logdevice *logdevice*]
[--database_local *true* | *false*]
[--database_type *sybase_ase* | *sybase_asa* | *oracle*]
[--entldb_username *username*]
[--jdbc_admin_password *password*]
[--jdbc_admin_username *user_name*]
[--jdbc_driver *driverName*]
[--jdbc_url *jdbcURL*]
[--populate_only *true* | *false*]
[--sybase_asa_servicename *service_name*]

Argument	Datatype	Default value	Description
asadb_acdb_owner_group	string	ACDB_owner	The ACDB group owner that is created when you install the database. This group owns the ACDB tables and procedures.
asedb_datadevice	string	acdbData	The Adaptive Server Enterprise device on which the database data segment is created. This device will be created if it does not exist.
asedb_logdevice	string	acdbLog	The Adaptive Server Enterprise device on which the database log segment is created. This device will be created if it does not exist.
database_local	boolean		If true, removes the devices after dropping the database; otherwise, you must do this manually.
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
entldb_username	string	acdbdbo	The user name for connecting to the ACDB. This user should have read and write privileges on the ACDB only.
jdbc_admin_password	string		The password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	The user name for connecting to the database for administrative purposes (creating databases, tables, and so on).
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	The JDBC driver to use when connecting to the database.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	The JDBC URL to use when connecting to the database.

Argument	Datatype	Default value	Description
populate_only	boolean	false	Set to true if using a JDBC database with an existing schema. In this case, only the initial data is inserted or removed.
sybase_asa_servicename	string	acdb	The SERVICENAME connection property to use when connecting to a Sybase ASA database.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example removes the ACDB using the default values:

```
securetool removedb
```

See also

createdb on page 59
createschema on page 63
populatedb on page 74
querydb on page 76
upgradedb on page 82

removemw

Description

Removes the Enterprise Security software from EAServer.

Syntax

removemw --easerver_dir *eas_dir*
[--easerver_host *host_name*]
[--easerver_port *port*]
[--easerver_password *password*]
[--easerver_servername *server*]
[--easerver_username *user_name*]

Argument	Datatype	Default value	Description
* easerver_dir	string		EAServer installation directory.
easerver_host	string		EAServer connection host name.
easerver_port	integer	9000	EAServer connection port.
easerver_password	string		Password for the user identified by easerver_username.

Argument	Datatype	Default value	Description
easerver_servername	string	Jaguar	Name of the server from which to remove the software.
easerver_username	string	jagadmin	User name for the EAServer administrator.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example removes the Enterprise Security middleware from EAServer, and specifies */work/Sybase/EAServer* as the EAServer installation directory:

```
securetool removemw --easerver_dir /work/Sybase/EAServer
```

See also

deploymw on page 66
removesm on page 81

removesm

Description

Removes Enterprise Security Manager from EAServer.

Warning! If you have Enterprise Portal installed on the same machine as Enterprise Security Manager, running this command removes your portal.

Syntax

removesm
[--easerver_host *host_name*]
[--easerver_port *port*]
[--easerver_username *user_name*]
[--easerver_password *password*]

Argument	Datatype	Default value	Description
easerver_host	string	[*host name*]	Name of the machine where EAServer is running.
easerver_port	integer	9000	EAServer connection port.
easerver_username	string	jagadmin	User name for the EAServer administrator.

Argument	Datatype	Default value	Description
easerver_password	string		Password for the user identified by easerver_username.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example removes Enterprise Security Manager from EAServer:

```
securetool removesm
```

See also

deploysm on page 68
removemw on page 80

upgradedb

Description

Upgrades the ACDB.

Note If you are using a database other than Adaptive Server Enterprise, you must create the ACDB schema before you upgrade the ACDB—see createschema on page 63. For Adaptive Server Enterprise, the schema is created automatically.

Syntax

upgradedb --enkfile *file_name*
[--database_type *sybase_ase* | *sybase_asa* | *oracle*]
[--entldb_jdbc_admin_password *jdbcPassword*]
[--entldb_jdbc_admin_username *jdbcUsername*]
[--entldb_jdbc_driver *jdbcDriver*]
[--entldb_jdbc_url *jdbcURL*]
[--entldbreader_username *readerUsername*]
[--jdbc_admin_password *password*]
[--jdbc_admin_username *user_name*]
[--jdbc_driver *driverName*]
[--jdbc_url *jdbcURL*]
[--psoname *psoName*]
[--psouid *psoUID*]
[--random_seed *randomSeed*]
[--security_properties_file *propsFile*]

[--sybase_asa_servicename *service_name*]

Argument	Datatype	Default value	Description
* enkfile	string		Encryption key file name; must be readable.
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase • sybase_asa • oracle
entldb_jdbc_admin_password	string		Use only when upgrading from an Enterprise Security version 2.5.2 database. Password for the user specified by entldb_jdbc_admin_username.
entldb_jdbc_admin_username	string	sa	Use only when upgrading from an Enterprise Security version 2.5.2 database. User name for connecting to the ACDB. This user must have read and write privileges on the ACDB.
entldb_jdbc_driver	string	com.sybase.jdbc2.jdbc.SybDriver	Use only when upgrading from an Enterprise Security version 2.5.2 database. JDBC driver to use when connecting to the ACDB.
entldb_jdbc_url	string	jdbc:sybase:Tds:*host*:5000/acdb	Use only when upgrading from an Enterprise Security version 2.5.2 database. JDBC URL to use when connecting to the ACDB.
entldbreader_username	string	entldbreader	ENTLDB reader login; use only when upgrading from an Enterprise Security version 2.0 database.
jdbc_admin_password	string		Password for the user specified by jdbc_admin_username.
jdbc_admin_username	string	sa	User name for connecting to the database for administrative purposes (creating databases, tables, and so on).

Argument	Datatype	Default value	Description
jdbc_driver	Java class name	Depends on the value of database_type; for sybase_ase: com.sybase.jdbc2.jdbc.SybDriver	JDBC driver to use when connecting to the database.
jdbc_url	string	Depends on the value of database_type; for sybase_ase: jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.
psoname	string	Portal Security Officer	Name of the Security Officer.
psouid	string	pso	Security Officer's user ID.
random_seed	integer		An integer to use as a random seed.
security_properties_file	string		The path and file name of the *security.properties* file; must be readable. This value is required only when upgrading from Enterprise Security version 2.5.2.
sybase_asa_servicename	string	acdb	The SERVICENAME connection property to use when connecting to a Sybase ASA database.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example upgrades the ACDB using the encryption key file */work/Sybase/Security/.enk*:

```
securetool upgradedb --enkfile /work/Sybase/Security/.enk
```

See also

createdb on page 59
createschema on page 63
populatedb on page 74
querydb on page 76
removedb on page 78

wls_configmw

Description

Configures the Sybase providers in the default realm of a BEA WebLogic server.

Note After running this command, restart the WebLogic server.

Syntax

```
wls_configmw --wls_admin_password password
             --wls_admin_username user_name -- wls_dir directory
[--wls_admin_host host_name]
[--wls_admin_port port]
[--wls_servername server]
[--wls_ssl_port ssl_port]
```

Argument	Datatype	Default value	Description
* wls_admin_password	string		Password for the user identified by wls_admin_username.
* wls_admin_username	string		User name for the WebLogic administrator.
* wls_dir	string		WebLogic installation directory.
jdbc_url	string	jdbc:sybase:Tds:*host*:5000/acdb	JDBC URL to use when connecting to the database.
wls_admin_host	string	localhost	Name of the machine where WebLogic is running.
wls_admin_port	integer	7001	WebLogic connection port.
wls_servername	string	myserver	The name of the WebLogic server in which to install the software. Reads the value from the WLS_SERVERNAME_ARGDESC environment variable.
wls_ssl_port	string	7002	SSL port for the WebLogic server.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example configures the Sybase providers in the default realm of the WebLogic server. The arguments provide the WebLogic administrator’s user name and password, and the WebLogic server installation directory.

```
securetool wls_configmw --wls_admin_username admin --wls_admin_password bea
   --wls_dir C:\WebLogic\Server
```

See also

wls_deploymw on page 86
wls_deploysm on page 88

wls_deploymw

Description

Deploys the Enterprise Security middleware to a BEA WebLogic server.

Note After deploying Enterprise Security to a WebLogic server, the administrator must perform the additional configuration steps described in "Deploying and configuring security in WebLogic" on page 22.

Syntax

```
wls_deploymw --enkfile file_name --security_dir secure_dir
          --shared_dir share_dir --wls_admin_password password
          --wls_admin_username user_name -- wls_dir directory
[--database_type sybase_ase | sybase_asa | oracle]
[--entldb_password password]
[--entldb_username user_name]
[--jdbc_admin_password password]
[--jdbc_admin_username user_name]
[--jdbc_driver driver]
[--jdbc_url URL]
[--sybase_asa asa_service_name
[--wls_admin_host host_name]
[--wls_admin_port port]
[--wls_domain_dir directory]
[--wls_servername server]
[--wls_systemuser systemUser]
```

Argument	Datatype	Default value	Description
* enkfile	string		Encryption key file to use; must be a readable file.
* security_dir	string		Enterprise Security installation directory.
* shared_dir	string		Sybase shared directory.
* wls_admin_password	string		Password for the user identified by wls_admin_username.
* wls_admin_username	string		User name for the WebLogic administrator.
* wls_dir	string		WebLogic installation directory.

Argument	Datatype	Default value	Description
database_type	choice	sybase_ase	The database type. This value is used to determine the default values for the jdbc_url and jdbc_driver arguments. Acceptable values are: • sybase_ase (the default) • sybase_asa • oracle
entldb_password	string	dbopswd	Password of the user specified by entldb_username.
entldb_username	string	acdbdbo	User name for connecting to the database. This user should have read and write privileges on the security database only.
jdbc_admin_password	string		The password for the user identified by jdbc_admin_username.
jdbc_admin_username	string	sa	The user name for connecting to the database for administrative purposes (creating databases, tables, and so on).
jdbc_driver	Java class name	com.sybase.jdbc2.jdbc .SybDriver	Additional JDBC driver to register before attempting to connect to the database. If you add Oracle drivers to the SECURETOOL_CLASSPATH environment variable, they are loaded automatically.
jdbc_url	string	jdbc:sybase:Tds:*host*: 5000/acdb	JDBC URL to use when connecting to the database.
sybase_asa	string	acdb	The SERVICENAME connection property to use when connecting to a Sybase ASA database.
wls_admin_host	string	localhost	Name of the machine where WebLogic is running.
wls_admin_port	integer	7001	WebLogic connection port.
wls_domain_dir	string		WebLogic domain root directory; must be an existing directory. If you use standard WebLogic paths, this value is usually detected automatically.
wls_servername	string		The name of the WebLogic server in which to install the software. Reads the value from the WLS_SERVERNAME_ARGDESC environment variable.

Argument	Datatype	Default value	Description
wls_systemuser	string	SybaseSecurity SystemIdentity (all one word)	The user name for making privileged method calls. This user must be created within the WebLogic administration system before Enterprise Security is deployed. Sybase recommends that you create a strong password for this user.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example deploys the Enterprise Security middleware to WebLogic. The arguments provide the name of the encryption key file, the installation location of Enterprise Security, the location of the Sybase shared directory, the WebLogic administrator's user name and password, and the WebLogic server installation directory.

```
securetool wls_deploymw --enkfile C:\Sybase\Security\.enk
   --security_dir C:\Sybase\Security --shared_dir C:\Sybase\Shared-1_0
   --wls_admin_username admin --wls_admin_password bea
   --wls_dir C:\WebLogic\Server
```

See also

wls_deploysm on page 88
wls_removemw on page 89

wls_deploysm

Description

Deploys the Enterprise Security Manager graphical user interface software to a BEA WebLogic server.

Syntax

wls_deploysm --dns_domain *domain* --security_dir *secure_dir*
--wls_admin_host *host* --wls_admin_password *password*
--wls_admin_username *user_name* -- wls_dir *directory*
[--wls_http_port *http_port*]
[--wls_https_port *https_port*]

Argument	Datatype	Default value	Description
* dns_domain	string		DNS domain of the server; used to construct a fully-qualified host name.
* security_dir	string		Enterprise Security installation directory.

Argument	Datatype	Default value	Description
* wls_admin_host	string		WebLogic administration host.
* wls_admin_password	string		Password for the user identified by wls_admin_username.
* wls_admin_username	string		User name for the WebLogic administrator.
* wls_dir	string		WebLogic installation directory.
wls_http_port	integer	7001	The WebLogic application server's HTTP port.
wls_https_port	integer	7002	The WebLogic application server's HTTPS port.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example deploys Enterprise Security Manager to a WebLogic server. The arguments provide the DNS name of the server, the installation location of Enterprise Security, the name of the WebLogic admin host, the WebLogic administrator's user name and password, and the WebLogic server installation directory.

```
securetool wls_deploysm --dns_domain sybase.roma.com
   --security_dir C:\Sybase\Security --wls_admin_host cosmo
   --wls_admin_username admin --wls_admin_password bea
   --wls_dir C:\WebLogic\Server
```

See also

wls_deploymw on page 86
wls_removesm on page 90

wls_removemw

Description

Removes the Enterprise Security middleware from a BEA WebLogic server.

Note After you remove Enterprise Security from a WebLogic server, the administrator must perform some manual configuration—see "Removing Enterprise Security from a WebLogic server" on page 25.

Syntax

wls_removemw --wls_admin_password *password*
--wls_admin_username *user_name* -- wls_dir *directory*
[--wls_admin_host *host_name*]
[--wls_admin_port *port*]

Argument	Datatype	Default value	Description
* wls_admin_password	string		Password for the user identified by wls_admin_username.
* wls_admin_username	string		User name for the WebLogic administrator.
* wls_dir	string		WebLogic server installation directory.
wls_admin_host	string	localhost	Name of the machine where the WebLogic server is running.
wls_admin_port	integer	7001	WebLogic connection port.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example removes the Enterprise Security middleware from a WebLogic server. The arguments provide the WebLogic administrator's user name and password, and the WebLogic server installation directory.

```
securetool wls_removemw --wls_admin_username admin
--wls_admin_password bea --wls_dir C:\WebLogic\Server
```

See also

wls_removesm on page 90
wls_deploymw on page 86

wls_removesm

Description

Removes the Enterprise Security Manager graphical user interface software from a BEA WebLogic server.

Syntax

wls_removesm --wls_admin_password *password*
--wls_admin_username *user_name* -- wls_dir *directory*
[--wls_admin_host *host_name*]
[--wls_admin_port *port*]

Argument	Datatype	Default value	Description
* wls_admin_password	string		Password for the user identified by wls_admin_username.

Argument	Datatype	Default value	Description
* wls_admin_username	string		User name for the WebLogic administrator.
* wls_dir	string		WebLogic server installation directory.
wls_admin_host	string	localhost	Name of the machine where the WebLogic server is running.
wls_admin_port	integer	7001	WebLogic connection port.

Return value

Return value	Indicates
0	The command ran successfully; the result is true/success.
1	The command failed.

Examples

This command line example removes Enterprise Security Manager from a WebLogic server. The arguments provide the WebLogic administrator's user name and password, and the WebLogic server installation directory.

```
securetool wls_removesm --wls_admin_username admin
--wls_admin_password bea --wls_dir C:\WebLogic\Server
```

See also

wls_removemw on page 89
wls_deploysm on page 88

CHAPTER 5 Delegated Administration

This chapter describes how to implement delegated administration for your enterprise.

Overview

In many enterprises, there are people at different organizational levels that manage various types of security information. Enterprise Security supports managing different levels of security data using a technique called “delegated administration.” Delegated administration enables system administrators to define multiple domains within an enterprise, each with its own security policy.

Administrators of each security domain can configure a set of properties (for example, to enable auditing or password validation) independent of other domains. Domain-specific properties are stored in the ACDB. You can modify these property values using either Enterprise Security Manager or the SecurityDomainManagement interface. Other security properties can be configured for your entire security system, rather than for a specific domain; these global properties are stored in the *security.properties* file.

Default values for all security properties are set automatically during installation; these default values are defined in Chapter 15, “Configuration Properties.” Global properties are read by the security system when a session is initiated for the first time. Domain-specific properties are read as they are needed. All the security configuration information is cached in the Configuration bean, and used at runtime by all the security modules. Domain properties are refreshed based on a time interval, which you can configure using either Enterprise Security Manager, or programmatically using the SMAPI to set the propertyRefreshTimeInterval property. If you reconfigure global properties, you must restart the application server for the changes to take effect.

To view the SMAPI documentation, open a browser, and access *docs/html/index.html* in your Enterprise Security installation; then, select the com.sybase.ep.security.management package.

Managing security domains and policies

Each domain has a set of controlling assets, one for each security object type: subject, group, organization, role, asset, one that controls access to AccessTypes and AssetTypes, and a domain-controlling asset that controls access to the domain itself. When you create a new domain, the controlling assets, and the domain security officer (DSO) role for the domain are created automatically. The person who creates the domain is granted the DSO role, and has sole permission to access the controlling assets, and to perform all the administrative functions in the domain. The DSO must create the various security objects: organizations, roles, users, groups, and assets. The DSO can also either grant the DSO role to another role, or assign a set of permissions to access the controlling assets to another role, and grant the role to other users—see “Managing the controlling assets in a domain” on page 101.

To create a new domain, you must have write permission on the domain controlling asset in the domain that contains the root organization. To update a security policy or the rules for a domain, you must have update permission on the domain controlling asset in the domain to which the changes apply. To delete a domain, you must have delete permission on the domain controlling asset in the domain to be deleted.

Creating and managing security domains

An enterprise can have one or more security domains; each security domain has a security policy associated with it. The Enterprise Security installation automatically creates a root organization and a default domain. The default domain contains the root organization, and a predefined security policy is associated with the default domain.

In each domain, you can create one or more suborganizations to represent the departments within your organization—see “Managing organizations and suborganizations” on page 30.

Table 5-1 describes the permissions you must have to manage security domains.

Table 5-1: Permissions required to manage security domains

Action	Permissions required
Create a security domain.	WRITE on the domain controlling asset in the domain that contains the root organization.
List the properties of a security domain.	LIST on the domain controlling asset.
Update the properties of a security domain.	READ and UPDATE on the domain controlling asset.
List the organizations in a security domain.	LIST on the organization controlling asset.
Delete a security domain.	READ and DELETE on the domain controlling asset.

❖ Creating a security domain

1 In the left pane of Enterprise Security Manager, under Configure, select Domains, and click New.

2 In the Create New Security Domain dialog box, enter:

- Domain Name – the name of the new domain.
- Domain Policy Name – select the name of the class that implements the security policy. The com.sybase.ep.security.policy.impl.DefaultDomainAssets class implements the default policy.

For information about implementing a new security policy, see “Managing security policies” on page 103.

❖ Editing a security domain

To edit the domain name, domain policy name, or domain description:

1 In the middle pane of Enterprise Security Manager, highlight the domain you want to edit. In the right pane, right-click and select Edit Domain.

2 In the Edit Domain dialog box, edit the values you want to change, and click OK.

❖ Configuring general properties for a security domain

1 In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click and select Configure General Properties.

2 In the Configure Domain General Properties dialog box, enter these values, then click OK:

- Domain Property Refresh Time Interval – the number of seconds that define how often the system refreshes the domain-specific properties by reading their values from the ACDB.
- Enable Auditing – select to enable auditing for the domain. To specify which events to audit, see “Defining which events to audit using Enterprise Security Manager” on page 130.

  If you do not enable auditing, you need not enter the remaining audit-related values.
- Suspend Auditing When Unable to Log Audit Messages – select to turn off auditing when a system problem prevents logging the auditing information. Selecting this property prevents a failure in the auditing module itself from causing a transaction to roll back.

  If the auditing system fails because a domain property is configured incorrectly, you may need to reset the domain properties to their default values—see domainrules on page 70.
- Include User’s DN in Audit Records – select to include the subject DN in audit records.
- Notify Audit Events – select to send notifications of audited events to a JMS message topic.

Warning! Do not select Notify Audit Events until after you set up both the message service and the message topic in your application server; otherwise, you will not be able to log in to Enterprise Portal—see “Setting up JMS auditing notifications for EAServer” on page 133.

❖ Configuring lock manager properties for a security domain

1. In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click and select Configure Lock Manager.
2. In the Configure Domain Lock Manager Properties dialog box, enter these values, then click OK:
    - Enable Login Lock – select to enable the system to lock out users after a specified number of invalid login attempts.
    - Allowed Invalid Login Attempts – the number of invalid login attempts users are allowed before they are locked out of the system.
    - Login Lockout Duration – the duration of the lockout. Select one of:
        - Permanent Lock – account remains locked until an administrator unlocks it.
        - Minutes – account remains locked for the specified number of minutes.
    - Reset Login Lockout Counter After – the number of minutes during which the number of invalid access attempts are counted. For example, if you set this property to “60” and Allowed Invalid Access Attempts to “3,” then 3 invalid access attempts within 60 minutes locks your account. If only 2 invalid access attempts occur within 60 minutes, the counter is reset to zero at the end of 60 minutes. A subsequent invalid access attempt is counted as the first, not the third.
    - Successful Login Clears Invalid Attempt History – select to delete information about invalid login attempts when users successfully log in.
    - Enable Authorization Lock – select to lock out users after a specified number of attempts to access a security object for which they do not have access permission.
    - Allowed Invalid Access Attempts – the number of invalid access attempts users are allowed before they are locked out of the system.
    - Authorization Lockout Duration – the duration of the lockout. Select one of:
        - Permanent Lock – authorization remains locked until an administrator unlocks it.
        - Minutes – authorization remains locked for the specified number of minutes.

- Reset Authorization Lockout Counter After – the number of minutes during which unauthorized attempts to access security objects are counted.
- Terminate Session When Authorization is Locked – select to terminate users' sessions when their authorization is locked.
- Lock Login Ability When Authorization is Locked – select to prevent users from logging in when their authorization is locked. If you select this option, you must also select Enable Login Lock.

❖ Configuring password properties for a security domain

1. In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click and select Configure Password Properties.
2. In the Configure Domain Password Properties dialog box, enter these values, then click OK:
   - Password Duration – the number of days that passwords remain valid. The default is 0, which means passwords are valid indefinitely.
   - Time Window to Change Expired Password – the number of days after passwords expire that users are allowed to change their passwords. The default is 0, which means that users cannot change their passwords after they expire.
   - Enable Password Strength Verification – select to enable password verification using an existing password validation component—see "Verifying passwords" on page 158.

❖ Configuring account properties for a security domain

1. In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click, and select Configure Account Properties.
2. In the Configure Domain Account Properties dialog box, enter these values, then click OK:
   - Account Expires After Inactivity Duration – the number of days that an inactive account remains valid. The default is 0, which means inactive accounts remain valid indefinitely.
   - Account Expires After Duration – the number of days that any account (active or inactive) remains valid. The default is 0, which means all accounts remain valid indefinitely.

❖ Registering a security policy

Registering a security policy does not assign the policy to a specific domain.

1 In the middle pane of Enterprise Security Manager, highlight All Domains. In the right pane, right-click, and select Register Policy.

2 In the Register Policy dialog box, enter the name of the class that implements the security policy. For example, the name of the class that implements the default security policy is com.sybase.ep.security.policy.impl.DomainAssetsPolicy.

3 Restart the application server.

4 To apply this security policy to a domain, edit the domain, and set Domain Policy Name to the class name you specified in step 2—see “Editing a security domain” on page 95.

❖ Listing the organizations in a security domain

1 In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click, and select List Organizations.

2 The dialog box that opens displays a list of the organizations in the current domain.

Managing the DSO role

In each domain, the domain security officer can edit the DSO role, and grant the role to users and groups. Initially, the DSO role is granted only to the “pso” user.

Warning! Do not delete the DSO role. If you delete the DSO role before permission to access the controlling assets is granted to another role, no one can access anything in the domain.

Table 5-2 describes the permissions you must have to manage the DSO role.

Table 5-2: Permissions required to manage the DSO role

Action	Permissions required
List the properties of a DSO role.	LIST on the role controlling asset.
Update the properties of a DSO role.	READ and UPDATE on the role controlling asset.
Grant the DSO role to users or groups.	GRANT on the role controlling asset.

❖ Editing the DSO role

To edit the DSO role name, DN, or description:

1. In the Domain Manager tree view, expand the domain, and highlight Roles.
2. In the right pane, highlight the role, right-click, and select Edit Role.
3. In the Edit Role dialog box, modify the values you want to change, and click OK.

❖ Granting the DSO role to users

1. In the Domain Manager tree view, expand the domain, and highlight Roles.
2. In the right pane, highlight the role, right-click, and select Manage User Roles.
3. In the Manage User Roles dialog box, select the organization in the left list box. The users in this organization display in the adjacent list box.

   The users to whom the DSO role is granted display in the Grant Role To list box. The organization to which each user belongs also displays, in parentheses. The users who inherit the DSO role because they belong to a group that is granted the DSO role display in the Inherited By list box.
4. Select a user to whom you want to grant the DSO role, and click Add. To grant the DSO role to all users in this organization, click Add All.

   To grant the DSO role to users in other organizations, repeat steps 3 and 4.

   To revoke the DSO role from a user, highlight the user in the Grant Role To list box, and click Remove. To revoke the DSO role from all users in this organization, click Remove All.

   To revoke the DSO role from users in other organizations, repeat steps 3 and 4.

❖ Granting the DSO role to groups

1. In the Domain Manager tree view, expand the domain, and highlight Roles.
2. In the right pane, highlight the role, right-click, and select Manage Group Roles.
3. In the Manage Group Roles dialog box, select the organization in the left list box. The users in this organization display in the adjacent list box. The groups to whom the role is granted display in the Grant Role To list box. The organization to which each group belongs also displays, in parentheses.
4. Select a group to which you want to grant the DSO role, and click Add. To grant the DSO role to all groups in this organization, click Add All.

   To grant the DSO role to groups in other organizations, repeat steps 3 and 4.

   To revoke the DSO role from a group, highlight the group in the Grant Role To list box, and click Remove. To revoke the DSO role from all groups in this organization, click Remove All.

   To revoke the DSO role from groups in other organizations, repeat steps 3 and 4.

Managing the controlling assets in a domain

Each security domain contains a set of controlling assets that control access to the other security objects in the domain. Controlling assets check users' permissions when they try to perform an action on an object. For example, if John tries to update the properties of organization O, the organization controlling asset checks whether any of the roles granted to John have permission to update O; if so, John is permitted to update O, otherwise, John is not permitted to update O. Table 5-3 lists the controlling assets; *domain* represents the name of the domain.

Table 5-3: Security domain controlling assets

Controlling asset	Controls access to
SYBDOMAIN_*domain*_AccessAssetTypeCtrlAsset	AccessTypes and AssetTypes
SYBDOMAIN_*domain*_AssetCtrlAsset	Assets
SYBDOMAIN_*domain*_DomainCtrlAsset	The domain
SYBDOMAIN_*domain*_GroupCtrlAsset	Groups
SYBDOMAIN_*domain*_OrgCtrlAsset	Organizations

Controlling asset	Controls access to
SYBDOMAIN_*domain*_RoleCtrlAsset	Roles
SYBDOMAIN_*domain*_SubjectCtrlAsset	Users

Table 5-4 describes the permissions you must have to manage the controlling assets.

Table 5-4: Permissions required to manage controlling assets

Action	Permissions required
List the controlling assets in a domain.	LIST on the domain controlling asset.
View the properties of a controlling asset.	READ on the controlling asset.
Update the properties of, or the permission to access, a controlling asset.	UPDATE on the controlling asset.

❖ Editing controlling assets in a domain

To edit the name, DN, asset type, or description of a controlling asset:

1. In the Domain Manager tree view, expand the domain, and highlight Assets.
2. In the right pane, highlight the controlling asset, right-click, and select Edit Asset.
3. In the Edit Asset dialog box, modify the values you want to change, and click OK.

❖ Managing permissions to access controlling assets

To define which roles have permission to access the controlling assets:

1. In the Domain Manager tree view, expand the domain, and highlight Assets.
2. In the right pane, highlight the controlling asset, right-click, and select Manage Access Permission.
3. In the left list, select the organization. In the adjacent list, select the role. The Available Permissions list box displays the permissions available for this controlling asset. The Assigned Permissions list displays the permissions assigned to the selected role.
4. To assign permissions to this role, highlight a permission in the Available Permissions list, and click Add. The permission displays in the Assigned Permissions list.

To assign all the available permissions to this role, click Add All.

All of this role's assigned permissions to access this controlling asset display in the Access Permissions Granted on the Asset list, at the bottom of the window.

To assign permissions to access this controlling asset to other roles, repeat steps 3 and 4.

5 To remove permissions, highlight the permission you want to remove in the Assigned Permissions list, and click Remove.

To remove all this role's permissions for this controlling asset, click Remove All.

For each role from which you want to remove access permissions, repeat steps 3 and 5.

Managing security policies

A domain's security policy interprets the rules for managing security issues, such as auditing and password expiration. Enterprise Security provides a default security policy, which is associated with the domain that contains the root organization. This policy duplicates the functionality of earlier versions of Enterprise Security. The default security policy is implemented by the com.sybase.ep.security.policy.impl.DefaultDomainAssets class.

❖ Implementing a new security policy

1 Create a JAR file with a class that implements the SecurityPolicy interface.

2 Add the JAR to the EAServer CLASSPATH.

3 Add the package name to the value of the sybepsecurity Web application's com.sybase.jaguar.application.java.classes property:

a In Jaguar Manager, expand the folder for the server in which sybepsecurity is installed (typically, Jaguar).

b In the Installed Web Application folder, highlight sybepsecurity, right-click, and select Properties.

c On the Java Classes tab, append the location of your package to the existing value, which typically is:

```
com.sybase.jaguar.application.java.classes=jce1_2_1.jar,
sunjce_provider.jar,US_export_policy.jar,local_policy.jar,
log4j-1.2.8.jar, sybepsecurity_classes.jar,ldapjdk.jar,
jakarta-oro2.jar,com.sybase.ep.security.sessionsvcs.*,
com.sybase.ep.security.epauth.*,
com.sybase.ep.security.authdelegate.*,
com.sybase.ep.security.authorization.*,
com.sybase.ep.security.management.*,
com.sybase.ep.security.cachemgr.*,
com.sybase.ep.security.management.impl.acdbimpl.*,
com.sybase.ep.security.config.*,com.sybase.ep.security.audit.*,
com.sybase.ep.security.policy.*, com.sybase.ep.security.webmgmt.*
```

For example, if you create your class in the com.sybase.epstg.security.policy package, append this to the value in the Java Classes tab:

```
,com.sybase.epstg.security.policy.*
```

4 Register the new security policy using Enterprise Security Manager—see “Registering a security policy” on page 99.

Alternately, you can register the new policy using the registerPolicy method of the Configuration remote interface. Pass the fully-qualified class name to registerPolicy.

You can also use the Configuration remote interface to get a list of all the registered security policies.

5 Restart your application server.

❖ **Updating an existing security policy**

If you update an existing security policy:

1 Re-create the JAR file with the class that implements the SecurityPolicy interface.

2 Restart your application server.

SMAPI updates

Enterprise Security version 6.0 includes SMAPI methods that enable you to manage security domains, and associate them with security policies. The security properties that you can define for a specific domain are described in “Domain-specific properties” on page 242.

All the existing SMAPI interfaces have been modified to use the object ID, instead of the DN, as the primary key. Methods still accept a DN to maintain backward compatibility, but performance improves if you use the object ID.

SMAPI now allows clients to change the name of a security object and the organization to which it belongs. All security objects are created in the same security domain as the organization in which they are created.

Table 5-5 describes the methods that have been added to the existing SMAPI interfaces:

Table 5-5: New SMAPI methods

Method name	Description
boolean changePassword(String, String, String)	Added to the SubjectQueries remote interface. Allows users to change their password by supplying their user name, old password, and new password.
String getDN()	Added to the AssetManagement, GroupManagement, OrganizationManagement, RoleManagement, and SubjectManagement remote interfaces. Returns the DN of the security object (asset, group, organization, role, or subject). **Note** The format of a DN can vary. An algorithm, which in earlier versions of Enterprise Security successfully parsed a DN, may no longer work correctly.
Map[] getOrgHierarchy(SearchInfo)	Added to the OrganizationQueries remote interface. Returns a list of the organizations rooted at the current organization, and information about their hierarchy.
String getSecurityDomain()	Added to the AssetManagement, GroupManagement, OrganizationManagement, RoleManagement, and SubjectManagement remote interfaces. Returns the primary key of the security domain.
Map[] listAncestorRoles(SearchInfo)	Added to the RoleQueries remote interface. Returns the list of roles from which the specified role inherits.
Map[] listBySecurityDomain(String)	Added to the OrganizationQueries remote interface. Returns a list of all the organizations in the specified domain.
Map[] listDescendantRoles(SearchInfo)	Added to the RoleQueries remote interface. Returns the list of roles that inherit from the specified role.
Map[] listInfoByConditions(SearchInfo[])	Added to the ProxyAuthenticationInfoQueries remote interface. Returns proxy authentication information that satisfies the specified conditions.

Method name	Description
Map[] listInfoByConditions(SearchInfo[], Integer)	Added to the AssetQueries, GroupQueries, OrganizationQueries, and RoleQueries remote interfaces. Returns a list of the appropriate security objects (assets, groups, organizations, or roles) that satisfy the specified conditions.
Map[] listInfoByConditions(String[], SearchInfo[], Integer)	Added to the SubjectQueries remote interfaces. Returns a list of the subjects that satisfy the specified conditions.
Map[] listInfoByLockType(String[], int)	Added to the SubjectQueries remote interface. Returns a list of users whose accounts are locked.
Map[] listRootOrgs()	Added to the OrganizationQueries remote interface. Returns a list of all the root organizations. Currently, only one root organization can exist but future versions of Enterprise Security are scheduled to support multiple root organizations.
void moveToNewOrganization(String[], String)	Added to the AssetQueries, GroupQueries, RoleQueries, and SubjectQueries remote interfaces. Moves multiple security objects (assets, groups, roles, or subjects) to a new organization.
boolean setDN(String)	Added to the AssetManagement, GroupManagement, OrganizationManagement, RoleManagement, and SubjectManagement remote interfaces. Sets the DN for the security object.
boolean setName(String)	Added to the AssetManagement, GroupManagement, OrganizationManagement, RoleManagement, SecurityDomainManagement, and SubjectManagement, remote interfaces. Sets the object’s name.
boolean setOrganization(String)	Added to the AssetManagement, GroupManagement, RoleManagement, and SubjectManagement, remote interfaces. Moves the object to the specified organization.
boolean setSecurityDomain(String)	Added to the OrganizationManagement remote interface. Moves the organization to the specified security domain.

Interfaces that support custom AccessTypes and AssetTypes

This section describes the SMAPI interfaces that support creating and managing custom AccessType and AssetType security objects. AccessType and AssetType security objects do not have a DN, so their names must be unique throughout the security system.

Table 5-6 list the AssetTypeManagementHome interface methods and the permissions required to run them.

Table 5-6: AssetTypeManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the AccessAssetTypeCtrlAsset controlling asset in the domain that contains the organization in which this AssetType is being created.
findByPrimaryKey(String)	READ on the AccessAssetTypeCtrlAsset controlling asset.
findByName(String)	READ on the AccessAssetTypeCtrlAsset controlling asset.

Table 5-7 lists the AssetTypeManagementQueries interface method and the permission required to run it.

Table 5-7: AssetTypeManagementQueries interface

Method	Permission required
listInforByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the AssetTypes returned.

Table 5-8 lists the AssetTypeManagement interface methods and the permissions required to run them.

Table 5-8: AssetTypeManagement interface

Method	Permission required
getAccessTypes()	None.
setAccessTypes(String[])	UPDATE on the AssetType’s controlling asset.
removeAccessTypes(String[])	UPDATE on the AssetType’s controlling asset.
setName()	UPDATE on the AssetType’s controlling asset.
setSecurityDomain(String)	DELETE on the AssetType’s controlling asset in the existing domain, and WRITE on the AccessAssetTypeCtrlAsset controlling asset in the new domain.
getSecurityDomain()	None.
getDescription()	None.
setDescription(String)	UPDATE on the AssetType’s controlling asset.
getInfo()	None.
setInfo(Map)	UPDATE on the AssetType’s controlling asset.
getControllingAsset()	None.

Table 5-9 lists the AccessTypeManagementHome interface methods and the permissions required to run them.

Table 5-9: AccessTypeManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the AccessType’s controlling asset in the domain that contains the organization in which this AccessType is being created.

Method	Permission required
findByPrimaryKey(String)	READ on the AccessType’s controlling asset.
findByName(String)	READ on the AccessType’s controlling asset.

Table 5-10 lists the AccessTypeManagementQueries interface method and the permission required to run it.

Table 5-10: AccessTypeManagementQueries interface

Method	Permission required
listInforByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the AccessTypes returned.

Table 5-11 lists the AccessTypeManagement interface methods and the permissions required to run them.

Table 5-11: AccessTypeManagement interface

Method	Permission required
setName()	UPDATE on the AccessType’s controlling asset.
setSecurityDomain(String)	DELETE on the AccessType’s controlling asset in the existing domain, and WRITE on the AccessAssetTypeCtrlAsset controlling asset in the new domain.
getSecurityDomain()	None.
getDescription()	None.
setDescription(String)	UPDATE on the AccessAssetTypeCtrlAsset controlling asset.
getInfo()	None.
setInfo(Map)	UPDATE on the AccessType’s controlling asset.
getControllingAsset()	None.

Changes to permissions required to invoke SMAPI methods

This section describes changes to the required access permissions that clients must possess to invoke the SMAPI methods. For complete details of the SMAPI methods, use a Web browser to access *docs/html/index.html* in your Enterprise Security installation; then, select the com.sybase.ep.security.management package.

Table 5-12 describes the permissions required to run the AssetManagementHome interface methods.

Table 5-12: AssetManagementHome interface

Method	Permission required
create(java.util.Map assetInfo)	WRITE on the asset's controlling asset in the domain that contains the organization in which this asset is being created.
create(String, String)	WRITE on the asset's controlling asset in the domain that contains the organization in which this asset is being created.
create(String, String, String)	WRITE on the asset's controlling asset in the domain that contains the organization in which this asset is being created.
findByPrimaryKey(String)	READ on the asset or the asset's controlling asset.
findByDN(String)	READ on the asset or the asset's controlling asset.

Table 5-13 describes the permissions required to run the AssetManagementQueries interface methods.

Table 5-13: AssetManagementQueries interface

Method	Permission required
listAccessibleByRole(String)	LIST on the asset's controlling asset.
listByOrganization(String)	LIST on the asset's controlling asset.
listByOrganization(String, boolean)	LIST on the asset's controlling asset.
listByOrganizationAndType(String, String)	LIST on the asset's controlling asset.
listByType(String)	LIST on the asset's controlling asset.
listInfoByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the assets returned.
moveToNewOrganization(String[], String)	DELETE on the assets being moved, or their controlling assets, or the caller must be the owner of the assets, and WRITE on the asset control asset of the domain in which the specified organization exists.

Table 5-14 describes the permissions required to run the AssetManagement interface methods.

Table 5-14: AssetManagement interface

Method	Permission required
setName()	UPDATE on the asset or controlling asset.
setOrganization()	DELETE on the asset or controlling asset, and WRITE on the asset's controlling asset in the new organization.
setType(String)	UPDATE on the asset or controlling asset.
setRoleProxyAuthInfoPriorities(String[])	UPDATE on the asset or controlling asset.
setDataObject(java.lang.Object)	UPDATE on the asset.
setData(byte[])	UPDATE on the asset.
revokeAccess(String, String)	GRANT on the asset or controlling asset.
removeAssetAccessCtrlInfo()	GRANT on the asset or controlling asset.

Method	Permission required
removeAccessCtrlInfoForRole(String)	GRANT on the asset or controlling asset.
remove()	DELETE on the asset or controlling asset.
grantAccess(String, String)	GRANT on the asset or controlling asset.
getData()	READ on the asset.
getDataObject()	READ on the asset.
setInfo()	UPDATE on the asset or the controlling asset.

Table 5-15 describes the permissions required to run the DomainManagementHome interface methods.

Table 5-15: DomainManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the domain’s controlling asset in the domain containing the root organization.
findByPrimaryKey()	READ on the domain’s controlling asset.
findByName(String)	READ on the domain’s controlling asset.

Table 5-16 describes the permissions required to run the DomainManagement interface methods.

Table 5-16: DomainManagement interface

Method	Permission required
remove()	DELETE on the domain’s controlling asset.
setRules()	UPDATE on the domain’s controlling asset.
setPolicy()	UPDATE on the domain’s controlling asset.
removeRules()	UPDATE on the domain’s controlling asset.
setDefaults()	UPDATE on the domain’s controlling asset.
setInfo()	UPDATE on the domain’s controlling asset.

Table 5-17 describes the permission required to run the DomainQueries interface methods.

Table 5-17: DomainQueries interface

Method	Permission required
listInfoByConditions(SearchInfo[], Integer)	READ on the domain’s controlling asset.

Table 5-18 describes the permissions required to run the GroupQueries interface methods.

Table 5-18: GroupQueries interface

Method	Permission required
listByDefaultRole(String)	LIST on the controlling assets of the groups.
listByOrganization(String)	LIST on the controlling assets of the groups.
listByRole(String)	LIST on the controlling assets of the groups.
listBySubject(String)	LIST on the controlling assets of the groups.
listInfoByConditions(SearchInfo[],Integer)	LIST on the controlling assets of the groups that are returned.
moveToNewOrganization(String[],String)	DELETE on the groups' controlling asset in the existing organization and WRITE on the groups' controlling asset in the new organization.

Table 5-19 describes the permissions required to run the GroupManagementHome interface methods.

Table 5-19: GroupManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the group's controlling asset in the domain that contains the organization in which this group is being created.
create(String)	WRITE on the group's controlling asset in the domain that contains the organization in which this group is being created.
create(String, String)	WRITE on the group's controlling asset in the domain that contains the organization in which this group is being created.
findByPrimaryKey(String)	READ on the group's controlling asset.
findByDN(String)	READ on the group's controlling asset.

Table 5-20 describes the permissions required to run the GroupManagement interface methods.

Table 5-20: GroupManagement interface

Method	Permission required
addMember(String)	UPDATE on the group's controlling asset.
remove()	DELETE on the group's controlling asset.
removeMember(String)	UPDATE on the group's controlling asset.
setName()	UPDATE on the group's controlling asset.
setOrganization()	DELETE on the group's current controlling asset, and WRITE on the controlling asset in the new organization.
setInfo()	UPDATE on the group's controlling asset.

Table 5-21 describes the permissions required to run the OrganizationQueries interface methods.

Table 5-21: OrganizationQueries interface

Method	Permission required
listRootOrganizations()	LIST on the controlling assets of all the root organizations. Currently, only one root organization can exist; however, future versions of Enterprise Security are scheduled to support multiple root organizations.
listSuborganizations(String)	LIST on the controlling asset of the specified organization.
listByDomain()	LIST on the controlling assets of the organizations that are returned.
listRootOrgs()	LIST on the controlling assets of all the root organizations.
listInfoByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the all the organizations that are returned.

Table 5-22 describes the permissions required to run the OrganizationManagementHome interface methods.

Table 5-22: OrganizationManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the controlling asset of the parent organization.
create(String, String)	WRITE on the controlling asset of the parent organization.
findByPrimaryKey(String)	READ on the controlling asset of the specified organization.
findByDN(String)	READ on the organization or its controlling asset.

Table 5-23 describes the permissions required to run the OrganizationManagement interface methods.

Table 5-23: OrganizationManagement interface

Method	Permission required
remove()	DELETE on the controlling asset of the organization.
setDomain()	DELETE on the controlling asset of the current domain and WRITE on the controlling asset of the new domain.
setName()	UPDATE on the controlling asset of the organization.
setOrganization()	DELETE on the controlling asset and WRITE on the organization’s controlling asset in the new organization.
setInfo()	UPDATE on the controlling asset of the organization.

Table 5-24 describes the permissions required to run the ProxyAuthenticationInfoQueries interface methods.

Table 5-24: ProxyAuthenticationInfoQueries interface methods

Method	Permission required
listByAsset(String)	READ on the asset or the controlling asset.
listByRole(String)	READ on the role’s controlling asset.

Method	Permission required
listBySubject(String)	READ on the subject's controlling asset, or the caller must be the subject.
listInfoByConditions(SearchInfo[])	Permission required depends on the specified conditions. READ on the asset or the asset's controlling asset, or READ on the role's controlling asset, or READ on the subject's controlling asset, or the caller must be the subject.

Table 5-25 describes the permissions required to run the ProxyAuthenticationInfoManagementHome interface methods.

Table 5-25: ProxyAuthenticationInfoManagementHome interface

Method	Permission required
create(java.util.Map)	To create proxy authentication information, which associates these security objects with their controlling assets: • Assets – you need UPDATE permission on either the asset or the asset's controlling asset. • Roles – you need UPDATE permission on the role's controlling asset. • Subjects – you need UPDATE permission on the subject's controlling asset. A user can also create his or her own subject-level proxy authentication information.
findByPrimaryKey(ProxyAuthenticationInfoKey)	To get the remote interface that enables you to manage the proxy authentication information for these security objects: • Assets – you need READ permission on either the asset or the asset's controlling asset. • Roles – you need READ permission on the role's controlling asset. • Subjects – you need READ permission on the subject's controlling asset. A user can also access his or her own subject-level proxy authentication information.

Table 5-26 describes the permissions required to run the ProxyAuthenticationInfoManagement interface methods.

Table 5-26: ProxyAuthenticationInfoManagement interface

Method	Permission required
remove()	UPDATE on the controlling asset of the security object (asset, role, or subject) with which the proxy authentication information is associated. Users can delete their own subject-level proxy authentication information.
setInfo(java.util.Map)	UPDATE on the controlling asset of the security object (asset, role, or subject) with which the proxy authentication information is to be associated. Users can update their own subject-level proxy authentication information.

Method	Permission required
setPassword(String)	UPDATE on the controlling asset of the security object (asset, role, or subject) with which the proxy authentication information is associated, or UPDATE on the asset itself for asset-level proxy authentication information. Users can update their own subject-level proxy authentication password.
setUrl(String)	UPDATE on the controlling asset of the security object (asset, role, or subject) with which the proxy authentication information is associated. Users can update their own subject-level proxy authentication URL.
setUsername(String)	UPDATE on the controlling asset of the security object (asset, role, or subject) with which the proxy authentication information is associated, or UPDATE on the asset itself for asset-level proxy authentication information. Users can update their own subject-level proxy authentication user name.

Table 5-27 describes the permissions required to run the RoleQueries interface methods.

Table 5-27: RoleQueries interface methods

Method	Permission required
listAncestors(String)	LIST on the controlling assets of the roles that are returned.
listByFilter(java.util.Map)	LIST on the controlling assets of the roles that are returned.
listByOrganization(String)	LIST on the controlling assets of the roles that are returned.
listDefaultGrantedToGroup(String)	LIST on the controlling assets of the roles that are returned.
listDefaultGrantedToSubject(String)	LIST on the controlling assets of the roles that are returned.
listDescendants(String)	LIST on the controlling assets of the roles that are returned.
listExplicitlyGrantedToSubject(String)	LIST on the controlling assets of the roles that are returned.
listGrantedToGroup(String)	LIST on the controlling assets of the roles that are returned.
listGrantedToSubject(String)	LIST on the controlling assets of the roles that are returned.
listMutuallyExclusive(String, int)	This method is no longer supported.
listAncestorRoles(SearchInfo)	LIST on the controlling assets of the roles that are returned.
listDescendantRoles(SearchInfo)	LIST on the controlling assets of the roles that are returned
listInfoByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the roles that are returned.
moveToNewOrganization(String[], String)	DELETE on the roles’ controlling asset in the existing domain and WRITE on the roles’ controlling asset in the domain that contains the specified organization.

Table 5-28 describes the permissions required to run the RoleManagementHome interface methods.

Table 5-28: RoleManagementHome interface

Method	Permission required
create(java.util.Map)	WRITE on the role's controlling asset in the domain that contains the organization in which this role is being created.
create(String)	WRITE on the role's controlling asset in the domain that contains the organization in which this role is being created.
create(String, String)	WRITE on the role's controlling asset in the domain that contains the organization in which this role is being created.
findByPrimaryKey(String)	READ on the role's controlling asset.
FindByDN(String)	READ on the role's controlling asset.

Table 5-29 describes the permissions required to run the RoleManagement interface methods.

Table 5-29: RoleManagement interface

Method	Permission required
addInheritanceRelationship(String)	GRANT on the controlling asset of the specified role.
addMutualExclusionRelationship(String, int)	This method is no longer supported.
grantToGroup(String)	GRANT on the role's controlling asset.
grantToGroup(String, boolean)	GRANT on the role's controlling asset.
grantToSubject(String)	GRANT on the role's controlling asset.
grantToSubject(String, boolean)	GRANT on the role's controlling asset.
remove()	DELETE on the role's controlling asset.
removeAsDefaultFromGroup(String)	GRANT on the role's controlling asset.
removeAsDefaultFromSubject(String)	GRANT on the role's controlling asset.
removeInheritanceRelationship(String)	GRANT on the specified role's controlling asset.
removeMutualExclusionRelationship(String)	This method is no longer supported.
revokeFromGroup(String)	GRANT on the role's controlling asset.
revokeFromSubject(String)	GRANT on the role's controlling asset.
setName()	UPDATE on the role's controlling asset.
setOrganization()	DELETE on the role's controlling asset in the existing organization and WRITE on the role's controlling asset in the new organization.
setInfo()	UPDATE on the role's controlling asset.

Table 5-30 describes the permissions required to run the SubjectQueries interface methods.

Table 5-30: SubjectQueries interface

Method	Permission required
listByDefaultRole(String)	LIST on the controlling assets of the subjects that are returned.

Method	Permission required
listByGroup(String)	LIST on the controlling assets of the subjects that are returned.
listByLockType(int)	LIST on the controlling assets of the subjects that are returned.
listByOrganization(String)	LIST on the controlling assets of the subjects that are returned.
listByRole(String)	LIST on the controlling assets of the subjects that are returned.
listInfoByConditions(SearchInfo[], Integer)	LIST on the controlling assets of the subjects that are returned.
moveToNewOrganization(String[], String)	DELETE on the subject's controlling asset in the existing organization and WRITE on the subject's controlling asset in the new organization.
listInfoByLockType(String[], int)	LIST on the controlling assets of the subjects that are returned.

Table 5-31 describes the permissions required to run the SubjectManagementHome interface methods.

Table 5-31: SubjectManagementHome interface

Method	Permission required
create(java.util.Map)	Guest role, or WRITE permission on the subject's controlling asset in the domain that contains the organization in which this subject is being created.
create(String, String, String, String, String)	Guest role, or WRITE permission on the subject's controlling asset in the domain that contains the organization in which this subject is being created.
create(String, String, String, String, String, String))	Guest role, or WRITE permission on the subject's controlling asset in the domain that contains the organization in which this subject is being created.
findByPrimaryKey(String)	READ on the subject's controlling asset, or the caller must be the subject.
findByDN(String)	READ on the subject's controlling asset, or the caller must be the subject.
findByUid(String)	READ on the subject's controlling asset, or the caller must be the subject.

Table 5-32 describes the permissions required to run the SubjectManagement interface methods.

Table 5-32: SubjectManagement interface

Method	Permission required
isAccountDisabled()	READ on the subject's controlling asset, or the caller must be the subject.
isExemptFromInactivityExpiration()	READ on the subject's controlling asset, or the caller must be the subject.

Method	Permission required
isExemptFromPasswordExpiration()	READ on the subject's controlling asset, or the caller must be the subject.
isLocked(int)	READ on the subject's controlling asset, or the caller must be the subject.
lockAccount(int, int)	UPDATE on the subject's controlling asset.
remove()	DELETE on the subject's controlling asset.
resetLastLoginDate()	UPDATE on the subject's controlling asset.
setAccountDisabled(boolean)	UPDATE on the subject's controlling asset.
setEmail(String)	UPDATE on the subject's controlling asset, or the caller must be the subject.
setExemptFromInactivityExpiration(boolean)	UPDATE on the subject's controlling asset.
setExemptFromPasswordExpiration(boolean)	UPDATE on the subject's controlling asset.
setExpirationDate(java.util.Date)	UPDATE on the subject's controlling asset.
setInfo(java.util.Map)	UPDATE on the subject's controlling asset, or the caller must be the subject.
setName	UPDATE on the subject's controlling asset, or the caller must be the subject.
setOrganization()	DELETE on the subject's controlling asset in the existing organization, and WRITE on the subject's controlling asset in the new organization.
setPassword(String)	UPDATE on the subject's controlling asset.
setPassword(String, String)	Caller must be the subject.
setPhone(String)	UPDATE on the subject's controlling asset, or the caller must be the subject.
setTemporaryPassword()	UPDATE on the subject's controlling asset.
unlockAccount(int)	UPDATE on the subject's controlling asset.

Changes to sessionsvcs.PortalSession

The PortalSession interface has been updated to include GRANT and REVOKE access types, in addition to READ, WRITE, UPDATE, DELETE, and EXECUTE access types. All other access types are still supported for backward compatibility but are scheduled to be removed in a future version of Enterprise Security.

CHAPTER 6 Auditing

This chapter describes how to set up a secured environment to audit user activities.

Overview

Enterprise Security includes an auditing feature that allows you to monitor user actions.

You can configure auditing to:

- Send notifications of audited events to a JMS message topic.
- Direct the audit output to a file, a JDBC-compliant database, or a custom application.
- Use a database connection cache to improve performance and scalability by allowing client connections to share open connections. For information about EAServer connection caches, see “Using Connection Management,” in the *EAServer Programmer’s Guide*. For information about using a WebLogic connection cache, see your BEA documentation.
- Refuse authorization and authentication requests if errors occur when an audit message is logged, such as when the file system is full.
- Disable auditing in the case of an auditing system failure, allowing users to continue to log in to the system and perform operations normally while auditing is suspended.

To audit user actions within a specific domain, enable auditing for the domain, and specify which actions to audit. To write audit information to a database, set up the audit database, and enable the connection cache database Service Provider Interface (SPI). When an audited event occurs, the default SPI implementation generates an XML-formatted string, stamps the output with a Greenwich Mean Time (GMT) timestamp, and writes it to an audit log. If you write a custom SPI, neither the XML string nor the timestamp are required.

Object-level auditing

Object-level auditing provides detailed information about attempts to access any object in the security system. The objects that you can audit are known collectively as AssetTypes. The actions you can perform on these objects are called AccessTypes.

If auditing is enabled, the information below can be logged for all audit records:

- Session primary key
- Domain primary key
- Caller ID
- Caller DN
- Time
- Action
- Action decision (permit or deny)
- Object type
- Object DN (except if the object is a domain, or if action is Create)

Domains do not have a DN so the domain name is logged instead.

The following tables describe the AccessTypes and interface methods that trigger auditable events for each AssetType.

Table 6-1: SYSTEM.AccessType

AccessType	Interface method
DELETE	AccessTypeManagement.remove
SYSTEM.Create	AccessTypeManagementHome.create
UPDATE	AccessTypeManagement.set* methods, all

Table 6-2: SYSTEM.Account

AccessType	Interface method
SYSTEM.AuthorizationLock	PortalSession.checkAccess PortalSession.checkAuthorization PortalSession.isAuthorized SubjectManagement.lockAccount In addition, any of the SMAPI interface methods that check access can trigger this event, if a user attempts to perform an action for which he or she does not have permission, a specified number of times. To define authorization lock parameters, see “Configuring lock manager properties for a security domain” on page 97.
SYSTEM.AuthorizationUnlock	SubjectManagement.unlockAccount
SYSTEM.LoginLock	SubjectManagement.lockAccount In addition, a specified number of failed attempts either calling PortalSession.authenticate, or performing implicit authentication to EAServer. For information about implicit authentication, see “Implicit role mapping” on page 146.
SYSTEM.LoginUnlock	SubjectManagement.unlockAccount

Table 6-3: SYSTEM.Asset

AccessType	Interface method
DELETE	delete
READ	getData
SYSTEM.Authorization	PortalSession.checkAccess PortalSession.checkAuthorization PortalSession.isAuthorized In addition, any SMAPI interface methods that check access permissions, such as create, remove*, and set* methods.
SYSTEM.Create	create

AccessType	Interface method
SYSTEM.GrantAccess	grantAccess
SYSTEM.RemoveAssetACE	removeAssetAccessCtrlInfo
SYSTEM.RemoveRoleACE	removeAccessCtrlInfoForRole
SYSTEM.RevokeAccess	revokeAccess
SYSTEM.SetRoleProxyAuthInfoPriorities	setRoleProxyAuthInfoPriorities
UPDATE	setData
UPDATE	setDescription
UPDATE	setDN
UPDATE	setInfo
UPDATE	setName
UPDATE	setType
UPDATE	setOrganization

Table 6-4: SYSTEM.AssetType

AccessType	Interface method
DELETE	AssetTypeManagement.remove
SYSTEM.Create	AssetTypeManagementHome.create
UPDATE	AssetTypeManagement.set* methods, all

Table 6-5: SYSTEM.Domain

AccessType	SMAPI method
DELETE	delete
SYSTEM.Create	create
SYSTEM.SetDefaultRules	setDefaultRules
UPDATE	removeRules
UPDATE	setDescription
UPDATE	setInfo
UPDATE	setName
UPDATE	setPolicy
UPDATE	setRules

Table 6-6: SYSTEM.Group

AccessType	SMAPI method
DELETE	delete
SYSTEM.AddMember	addMember
SYSTEM.Create	create
SYSTEM.RemoveMember	removeMember

AccessType	SMAPI method
UPDATE	setDescription
UPDATE	setDN
UPDATE	setInfo
UPDATE	setName
UPDATE	setOrganization

Table 6-7: SYSTEM.Organization

AccessType	SMAPI method
DELETE	delete
SYSTEM.Create	create
UPDATE	setDescription
UPDATE	setDN
UPDATE	setInfo
UPDATE	setName
UPDATE	setParentOrganization
UPDATE	setSecurityDomain

Table 6-8: SYSTEM.ProxyAuthenticationInformation

AccessType	SMAPI method
DELETE	delete (asset-level)
DELETE	delete (role-level)
DELETE	delete (subject-level)
READ	getInfo
READ	getPassword
READ	getUrl
READ	getUsername
SYSTEM.Create	create (asset-level)
SYSTEM.Create	create (role-level)
SYSTEM.Create	create (subject-level)
UPDATE	setInfo
UPDATE	setPassword
UPDATE	setUrl
UPDATE	setUsername

Table 6-9: SYSTEM.Role

AccessType	SMAPI method
DELETE	delete

AccessType	SMAPI method
SYSTEM.AddRoleInheritance	addInheritanceRelationship
SYSTEM.Create	create
SYSTEM.GrantRole	grantToGroup
SYSTEM.GrantRole	grantToSubject
SYSTEM.RemoveRoleInheritance	removeInteritanceRelationship
SYSTEM.RevokeRole	revokeFromGroup
SYSTEM.RevokeRole	revokeFromSubject
UPDATE	setDescription
UPDATE	setDN
UPDATE	setInfo
UPDATE	setName
UPDATE	setOrganization

Table 6-10: SYSTEM.Session

AccessType	Interface method
SYSTEM.Login	PortalSession.authenticate In addition, attempts to authenticate implicitly to EAServer—see “Implicit role mapping” on page 146.
SYSTEM.Logout	PortalSession.disconnect PortalSession.remove In addition, this event is triggered when a portal session expires.

Table 6-11: SYSTEM.Subject

AccessType	Interface method
DELETE	delete
SYSTEM.Create	create
SYSTEM.DeleteCertificate	removeCertificate
SYSTEM.LockAccount	lockAccount
SYSTEM.RegisterCertificate	registerCertificate
SYSTEM.UnlockAccount	unlockAccount
UPDATE	resetLastLoginDate
UPDATE	setAccountDisabled
UPDATE	setDescription
UPDATE	setDN
UPDATE	setEmail
UPDATE	setExemptFromInactivityExpiration
UPDATE	setExemptFromPasswordExpiration

AccessType	Interface method
UPDATE	setExpirationDate
UPDATE	setExtraInfo
UPDATE	setFirstName
UPDATE	setInfo
UPDATE	setLastName
UPDATE	setName
UPDATE	setOrganization
UPDATE	setPassword
UPDATE	setPhone
UPDATE	setTemporaryPassword

Setting up auditing

To audit user actions, you must:

1. Enable auditing
2. Set up the audit-data store
3. Specify which events to audit

Optionally, you can set up auditing notifications to be sent to a JMS message topic.

Note To configure auditing for a domain, you must have UPDATE permission on the domain controlling asset.

Enable auditing

To enable auditing, use one of these tools:

- securetool—see “Enabling auditing using securetool,” below.
- Enterprise Security Manager—see “Configuring general properties for a security domain” on page 96.

- SMAPI to set the properties defined in Table 15-12 on page 244. To view the SMAPI documentation, open a Web browser, and access *SECURITY/html/docs/index.html*; then, select the com.sybase.ep.security.management package.

❖ Enabling auditing using securetool

For detailed information about securetool, see Chapter 4, "Using securetool."

- To enable auditing, change to the *SECURITY/bin* directory, and run:

```
securetool domainrules --appserver_url <protocol>://<host>:<port>
--username pso --password pso_password --operation set
--rule_name auditEnable --rule_value true [--domain <domain_name>]
```

Where:

- *protocol* is specific to your application server; use "iiop" for EAServer; use "T3" for WebLogic.
- *host* is the name of the machine hosting the Enterprise Security middleware.
- *port* is the IIOP or T3 port number.
- *pso_password* is the password for the PSO.
- To enable auditing for a domain other than the default, provide the `--domain` argument, and specify the name of the domain.

Events are audited when a security policy decision is made, even if the decision is made within an application-level transaction that is rolled back. If you prefer to roll back audit records when a security transaction rolls back:

1. Log the audit records in the ACDB.
2. Set the transaction attribute for the SecureAuditWriterBean methods to "Required."

To prevent a failure in the auditing module itself from causing a transaction to roll back, set the `auditSuspendOnFailure` property to true—see "Configuring general properties for a security domain" on page 96. For information about transactions, see Chapter 2, "Understanding Transactions and Component Lifecycles," in the *EAServer Programmer's Guide*.

Set up the audit-data store

You can store the audit information in a database or system file, or you can send the information to a custom Java class for processing.

❖ Specifying where to store audit information

1. Using any standard ASCII text editor, open the *security.properties* file. The location of the file depends on your application server:
   - EAServer – *JAGUAR/java/classes/com/sybase/ep/security*.
   - WebLogic – *BEA_ROOT/sybepsecurity/etc/com/sybase/ep/security*.
2. Set the following properties, which are defined in Table 15-1 on page 236:
   - auditSPI – dbconncache, file, database, or the name of a Java class.
   - auditLog – file name for primary or backup audit storage.
   - auditOverflowLog – secondary backup location.

❖ Setting up the audit database

If you set auditSPI to either “dbconncache” or “database,” you must set up an audit database.

1. Choose a database to host the auditing information. To maintain complete audit records, Sybase recommends that you use the ACDB. If you are using an Adaptive Server Enterprise database, and you need information on managing the database log, see Chapter 27, “Developing a Backup and Recovery Plan,” in the *Adaptive Server Enterprise System Administration Guide*, which is available on the Sybase Product Manuals Web page at http://sybooks.sybase.com/onlinebooks/group-as/asg1251e/sag/@Generic__BookView.
2. If you are using the ACDB, the SecDboCache is configured automatically when Security is installed, so you may skip this step.

   To use a database other than the ACDB, create a connection cache to the database, and link the com.sybase.ep.security.audit/SecureAuditWriter component’s jdbc/Audit resource reference to the new connection cache.

   If you are using EAServer, see these chapters:
   - Chapter 4, “Database Access,” in the *EAServer System Administration Guide* for information on creating connection caches.
   - Chapter 21, “Creating Web Applications,” in the *EAServer Programmer’s Guide* for information on configuring resource references.

If your application server is WebLogic, refer to your BEA documentation.

3. Choose a schema for your audit records. The following is the default schema, which can be created in an Adaptive Server Enterprise database:

```
create table Audit(recordID varchar(64),
                   timeStamp DATETIME,
                    auditData TEXT)
```

Note If you are using an Oracle database, you must use a name other than "Audit" for your audit table, which is a keyword in Oracle.

The information contained in the `auditData` field is specific to the object type and the AccessType—see "Object-level auditing" on page 120.

Table 6-12 defines the complete list of information you can audit. An asterisk identifies the default items.

Table 6-12: Auditable information

VALUES	Information to audit	Field name	Datatype	Can be null
1	* Audit record ID	recordID	varchar(64)	No
2	* Audit timestamp	timeStamp	• DATETIME in ASA and Adaptive Server Enterprise • DATE in Oracle	No
3	* XML audit record	auditData	• CLOB (character large object) • TEXT in Adaptive Server Enterprise • Reader in *java.sql*	No
4	Domain primary key	domainPK	varchar(30)	No
5	Subject primary key	subjectPK	varchar(30)	Yes
6	Subject DN	subjectDN	varchar(255)	Yes
7	Session primary key	sessionPK	varchar(64)	Yes
8	Resource class name (AssetType)	resourceClass	varchar(255)	No
9	Resource ID	resourceID	varchar(255)	No
10	Action (AccessType)	action	varchar(255)	No
11	Decision (permit or deny)	decision	• BIT • Boolean in *java.sql*	Yes

4. To audit more than the default items, update this statement in the *security.properties* file:

```
auditDatabaseInsertSql=INSERT INTO Audit(recordID, timeStamp, auditData)
VALUES ({1}, {2}, {3})
```

For each additional piece of information you want to audit, append the field name to the comma-separated list of fields for the Audit table, and add the corresponding VALUES number (from Table 6-12) enclosed by curly braces to the VALUES list. For example, to include the action and decision in each audit record, the `auditDatabaseInsertSql` statement should be:

```
auditDatabaseInsertSql=INSERT INTO Audit(recordID, timeStamp, auditData,
action, decision) VALUES ({1}, {2}, {3}, {10}, {11})
```

5 Create the Audit table in your database, and configure the following:

- Add row-level locking to the table.

  For an Adaptive Server Enterprise database, see Performance and Tuning: Locking at http://sybooks.sybase.com/onlinebooks/group-as/asg1251e/locking/@Generic__BookView.

- For each field whose value can be null, create a database column that supports null values. Adaptive Server Enterprise database columns support null values only if you specifically configure them to do so.

6 In *security.properties*, verify that auditSPI is set to this value:

```
auditSpi=dbconncache
```

Alternately, you can enable the standard JDBC database SPI by defining these values in *security.properties*; however, Sybase does not recommend using this SPI due to performance issues:

```
auditDatabaseJdbcDriver=com.sybase.jdbc2.jdbc.SybDriver
auditDatabaseJdbcUrl=jdbc:sybase:Tds:<host_name>:<port>/<db_name>
auditDatabasePassword=<password>
auditDatabaseUsername=<user_name>
auditSpi=database
```

If you enable the standard JDBC database SPI, you can skip step 2.

7 Restart your application server to enable the new SPI, and verify that authentication is working. If configuration errors exist, debugging information may be written to both *JAGUAR/bin/Jaguar.log* and *SECURITY/security.log*.

Specify which events to audit

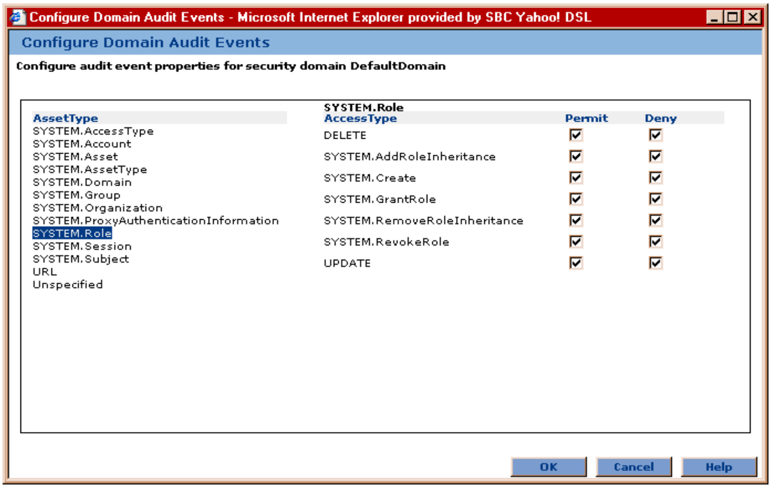
You can specify which events to audit within a domain either using Enterprise Security Manager, or by defining text expressions called filters. To set up auditing for many events, writing filters can be quicker than using Enterprise Security Manager.

Permissions required for defining which events to audit To define which events to audit, you must have UPDATE permission on the domain controlling asset.

❖ Defining which events to audit using Enterprise Security Manager

1 In the middle pane of Enterprise Security Manager, highlight the domain you want to configure. In the right pane, right-click, and select Configure Audit Events. The Configure Domain Audit Events dialog box displays.

Figure 6-1: Configure Domain Audit Events



Configure Domain Audit Events - Microsoft Internet Explorer provided by SBC Yahoo! DSL
Configure Domain Audit Events
OK
Cancel
Help

Configure audit event properties for security domain DefaultDomain

AssetType
SYSTEM.AccessType
SYSTEM.Account
SYSTEM.Asset
SYSTEM.AssetType
SYSTEM.Domain
SYSTEM.Group
SYSTEM.Organization
SYSTEM.ProxyAuthenticationInformation
SYSTEM.Role
SYSTEM.Session
SYSTEM.Subject
URL
Unspecified

SYSTEM.Role

AccessType	Permit	Deny
DELETE	☑	☑
SYSTEM.AddRoleInheritance	☑	☑
SYSTEM.Create	☑	☑
SYSTEM.GrantRole	☑	☑
SYSTEM.RemoveRoleInheritance	☑	☑
SYSTEM.RevokeRole	☑	☑
UPDATE	☑	☑

2 Select an AssetType from the list. The auditable AccessTypes (actions) for the AssetType display. To audit actions when they succeed, select Permit next to the action name. To audit actions when they fail, select Deny.

You can create and manage custom AssetTypes and AccessTypes using the SMAPI AssetTypeManagement, AccessTypeManagement, AssetTypeQueries, and AccessTypeQueries interfaces. To access the SMAPI documentation, use a browser to open *docs/html/index.html* in your Enterprise Security installation; then, select the com.sybase.ep.security.management package. If you create new AssetType or AccessType objects, refresh your application server to display the new objects in this dialog box.

To audit actions performed on custom AccessType objects, select the appropriate actions and decisions for the SYSTEM.AccessType object. To audit actions performed on custom AssetType objects, select the appropriate actions and decisions for the SYSTEM.AssetType object. For example, to audit the creation of a custom AccessType, you must enable the SYSTEM.Create event for the SYSTEM.AccessType object.

Note If auditing is enabled, Enterprise Security generates audit records for every audit request it receives; however, Enterprise Portal does not generate audit requests for the following AssetTypes:

- EPStudio (all AssetTypes whose names begin with “EPStudio”)
- Unspecified
- URL

If a custom client generates audit requests for these AssetTypes, Enterprise Security creates audit records for them.

❖ Defining which events to audit using filters

Auditing filters allow you to specify what information to audit by defining text expressions.

1 Define the values for the auditing filters using this syntax:

```
INCLUDE_FILTER =([key = value [,key = value...] ])
EXCLUDE_FILTER =([key = value [, key = value...] ])
```

Where the acceptable values for *key* are ResourceClass, Action, and Decision. For both filters, you can include as many *key*/*value* pairs as you want.

Note The terms “ResourceClass” and “AssetType” represent the same thing, as do the terms “Action” and “AccessType.”

The INCLUDE_FILTER expression defines the audit records that are eligible to be audited. The EXCLUDE_FILTER expression defines the audit records that you do not want to audit. If a record satisfies the INCLUDE_FILTER, it is checked against the EXCLUDE_FILTER. A record is audited only if it matches the INCLUDE_FILTER and does not match the EXCLUDE_FILTER.

For example, the filters below enable auditing for all events where the ResourceClass starts with "SYSTEM" or the Action is "Create," except when the Action is "AUTHORIZATION" and the Decision is "Permit."

```
INCLUDE_FILTER=(ResourceClass=SYSTEM.*)(Action=Create)
EXCLUDE_FILTER=(Action=AUTHORIZATION,Decision=Permit)
```

Note You cannot use a right or left parenthesis, an equal sign, or a comma in the value of a ResourceClass or Action.

Leading and trailing white space in key names and values is trimmed. White space within a name or value is retained.

Following are the default values for the filters, which audit all records generated by the Enterprise Security system and externally generated events:

```
INCLUDE_FILTER=(ResourceClass=SYSTEM.*)
EXCLUDE_FILTER=
```

2 Set the audit filters, `auditIncludeFilter` and `auditExcludeFilter`, using the domainrules task—see domainrules on page 70.

3 Restart your application server for the changes to take effect.

❖ **Reading audit information from the Audit database**

- To read audit information from the Audit database table, use a SQL statement to retrieve the data. For example, if you are using the default Audit table schema, to retrieve the information from the `auditData` field for all events that occurred after 1 September 2003, use this statement:

```
select auditData from Audit where timeStamp >
(select convert(datetime, "9/1/2003"))
```

Configure auditing notifications

Auditing notifications are sent to a JMS message topic in your application server.

❖ Setting up JMS auditing notifications for EAServer

1. In your application server, enable JMS support—see Chapter 8, “Setting up the Message Service,” in the *EAServer System Administration Guide*, and follow these steps:

    a. In Jaguar Manager, configure the message service, and select a database for persistent storage of JMS messages.

    b. Create a configured topic called “AuditTopic.”

        The Enterprise Security installation defines a resource environment reference in the com.sybase.ep.security.audit\SecureAuditWriter bean that points to a JMS topic called AuditTopic. To use another name for the topic, change the name of the jms/AuditTopic resource environment reference in the com.sybase.ep.security.audit\SecureAuditWriter bean.

2. To direct auditing information to the JMS topic, use one of these tools to set the value of auditJMSEnable to true for each domain in which you want to enable notifications:

    - Enterprise Security Manager—see “Configuring general properties for a security domain” on page 96.
    - securetool—see domainrules on page 70.

3. Restart EAServer for the changes to take effect.

❖ Setting up JMS auditing notifications for WebLogic

1. In your WebLogic application server, enable JMS support, select a database for persistent storage of JMS messages, and create a configured audit topic—see your BEA documentation.

2. Using a text editor, open the *weblogic-ejb-jar.xml* file, located in the *sybepsecurity/sybepsecurity.ear.exploded/ com.sybase.ep.security.audit.jar/META-INF* subdirectory of your WebLogic installation.

    Find the “resource-env-description” block of code, which is commented out. Uncomment it, and replace “jms.AuditTopic” with the JNDI name of the audit topic you created.

    Save and close the file.

3. In the same directory, open the *ejb-jar.xml* file.

    Find the “resource-env-ref” block of code, which is commented out, and uncomment it.

Save and close the file.

4 Restart your application server for the changes to take effect.

Receiving messages

If you enable auditing notifications for a domain, all audited messages are sent to the JMS topic. To receive messages, write a message-driven bean (MDB) that listens for messages on the audit topic. You can find a sample MDB in the *SECURITY/samples/src/samples/audit* directory.

Each JMS message is an instance of a javax.jms.TextMessage; the text portion is the XML audit record that is sent to the primary auditing destination. The following message properties are initialized, which enable you to filter the messages that are sent to your JMS topic:

Message property	Datatype
ResourceClass	String
Action	String
Decision (can be unspecified)	Boolean

If you are using EAServer, you can filter the messages sent to the audit topic by creating message selectors—see Chapter 8, “Setting up the Message Service,” in the *EAServer System Administration Guide*. If you are using WebLogic, see your BEA documentation.

Implementing a custom SPI

You can customize how your auditing information is handled by writing a Java class to manage the data:

1 Create a Java class that implements the com.sybase.ep.security.audit.AuditSPI interface.

2 Set the value of auditSPI to the name of the Java class—see Table 15-1 on page 236.

Enterprise Security provides the com.sybase.ep.security.audit.AuditImplBase abstract class, which implements the com.sybase.ep.security.audit.AuditSPI interface. The following code sample illustrates a valid auditing class that you can use:

```
import com.sybase.ep.security.audit.*;

public class MyAudit extends AuditImplBase
```

```
{
   protected void performAudit(String auditRecord)
                                 throws AuditException
   {
      System.out.println(auditRecord);
   }
}
```

Auditing Enterprise Portal events

You can audit Enterprise Portal events that are triggered by actions you perform in either Portal Studio or Portal Interface. Events are defined by an AssetType/AccessType pair. Table 6-13 describes the Portal Studio events that you can audit. Table 6-14 on page 137 describes the Portal Interface events that you can audit. To set up auditing for any of the Enterprise Portal events, see "Defining which events to audit using Enterprise Security Manager" on page 130.

Portal Studio auditing

Table 6-13 describes the Portal Studio events that you can audit, and the actions that triggers these events.

Table 6-13: Portal Studio auditable events

AssetType	AccessType	Action that triggers event
STUDIO.Applications	Studio.Create	Creating a new application. Triggers page and page group add and update actions, which occur immediately, bypassing the deferupdate global setting for the current user only.
	Studio.Preview	Clicking Preview in Application Builder. Displays the application's portlets, which triggers the portlet playback action and audit on those portlets.
	Studio.Update	Clicking Deploy. Triggers page group update action.
STUDIO.AutoFillAdapter	Studio.Create	Creating an autofill adapter in Personalize | Adapter, the adapterType attribute in the audit record shows the autofill type.
	Studio.Delete	Deleting a personalization adapter.
	Studio.Update	Clicking Edit in Adapter Builder.

AssetType	AccessType	Action that triggers event
STUDIO.Catalogs	Studio.Create	Creating a new catalog or saving an existing catalog. If the “requested catalog ID” is zero, the catalog is newly created. Does not audit the portlets added to the catalog.
	Studio.Update	Clicking Update Catalog.
STUDIO.Objects	Studio.Deploy	Clicking Deploy and performing deploy operation. All the Studio objects that are deployed (saved to *.ear* file) are audited as attributes.
	Studio.Export	Clicking Deploy and performing export operation. All the objects that are exported (saved to *.xml* file) are audited as attributes. The log content is listed as an attribute.
	Studio.Import	Clicking Deploy and performing import operation. All the objects that are imported (from the *.xml* file) are audited as attributes, which may trigger the page and page group update actions.
STUDIO.PageGroups	Studio.Create	Creating a new page group or saving an existing page group. If the “requested page group ID” is zero in the audit record, the page group was newly created. Does not audit pages and portlets added to the page group.
	Studio.Update	Clicking Update. If the global property deferupdate is turned on, the update happens when the user logs in; otherwise, the update occurs immediately. Either way, an audit record is generated. When the update happens, it triggers page and page group deletions and portlet additions.
STUDIO.Pages	Studio.Create	Creating a new page or saving an existing page. If the “requested page ID” is zero in the audit record, the page is new. This does not audit portlets added to the page.
	Studio.Update	Clicking Update. If the global property deferupdate is turned on, the update occurs when the user logs in; otherwise, the update occurs immediately. Either way, the audit record is generated. When the update happens, it triggers page and page group deletions and portlet additions.
STUDIO.Portlets	Studio.Create	Creating a new portlet or saving an existing portlet.
	Studio.Preview	Clicking Preview in Portlet Builder. The portlet preview is audited when a valid portlet ID is passed to UWPWindowApp.GetWindowContent.
	Studio.SearchReplace	Performing a search and replace action on a portlet. The pages and catalogs affected are listed as attributes in the audit record. If Update Pages is selected, this action also triggers the page and page group update action and audit.

AssetType	AccessType	Action that triggers event
STUDIO.Resources	Studio.Create	Creating a new resource ID.
	Studio.Delete	Deleting or undeleting a resource. Deleting a resource ID does not physically delete the resource; the resource is just marked for deletion from the database.
	Studio.Update	Clicking Edit in the Resource Builder.
STUDIO.Roles	Studio.Create	Creating a J2EE role reference.
	Studio.Delete	Deleting a J2EE role reference.
	Studio.Update	Clicking Edit in the Role Builder. If you undelete a J2EE role, an update action is generated.
STUDIO.Templates	Studio.Create	Creating a new template or saving an existing template.
	Studio.Preview	Clicking Preview in Template Builder.

Portal Interface auditing

Table 6-14 describes the Portal Interface events that you can audit and the actions that trigger these events.

Table 6-14: Portal Interface auditable events

AssetType	AccessType	Action that triggers event
PORTAL.PageGroups	Portal.Add	Adding a page group.
	Portal.Delete	Deleting a page group.
	Portal.Update	Changing a page group’s name or page order.
	Portal.UpdatePages	Updating a user’s page groups and pages after the user logs in. This happens when the global property deferupdate is turned on and a Portal Studio user initiates the update operation.
PORTAL.Pages	Portal.Add	Adding a page.
	Portal.Delete	Deleting a page.
	Portal.Update	Changing a page’s layout or name.
	Portal.Shared	Sharing a page with another user.
	Portal.List	Listing pages that are available to add.
	Portal.UpdatePages	Updating a user’s page groups and pages after the user logs in. This happens when the global property deferupdate is turned on and a Portal Studio user initiates the update operation.

AssetType	AccessType	Action that triggers event
PORTAL.Portlets	Portal.Add	Adding Portal Studio-created portlet from a catalog.
	Portal.Create	Creating a one-click capture portlet in Portal Interface using Create Portlet.
	Portal.Delete	Deleting a portlet.
	Portal.Update	Editing a portlet.
	Portal.Playback	Playing back a portlet on a page. For <DIV> tag portlets, if the portlet has already been played back once and cached, subsequent playbacks are not audited. However, refreshing a portlet audits portlet playback again.
SYSTEM.Session	SYSTEM.Login	Logging in to Portal Interface.
	SYSTEM.Logout	Logging out of Portal Interface.

Sample Portal Interface audit records

This section illustrates sample audit records for the Portal Interface events defined in Table 6-15. In the audit records, the AssetType is identified by the term “Resource Class,” and the AccessType is called “Action.” To enable auditing for these AssetType/AccessType combinations, see “Defining which events to audit using Enterprise Security Manager” on page 130.

Table 6-15: Portal Interface events

AssetType	AccessType
SYSTEM.Session	SYSTEM.Login
SYSTEM.Session	SYSTEM.Logout
PORTAL.Portlets	Portal.Playback

SYSTEM.Session/ SYSTEM.Login

```
<?xml version="1.0" encoding="UTF-8"?>
<audit:AuditLog
xmlns:audit="http://www.sybase.com/ep/xsd/audit/6.0">
<AuditRecord Action="SYSTEM.Login" Decision="permit"
Subject="908102000000001001" When="2003-08-12T23:41:02"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7fff">
<Resource Class="SYSTEM.Session"
ID="SYBSECPK28670A1655304788d5f6f888a4198000"/>
<Attribute Name="AuthenticationType"
Value="AUTH_BY_CREDENTIAL"/>
<Attribute Name="SubjectUID" Value="hwong1"/>
</AuditRecord>
```

SYSTEM.Session/
SYSTEM.Logout

```
<AuditRecord Action="SYSTEM.Logout" Decision="permit"
Subject="908102000000001001" When="2003-08-12T23:46:05"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ff6">
<Resource Class="SYSTEM.Session"
ID="SYBSECPK28670A1655304788d5f6f888a4198000"/>
<Attribute Name="LogoutTime" Value="2003-08-12T23:46:05"/>
<Attribute Name="LogoutType" Value="Terminated"/>
</AuditRecord>
```

PORTAL.Portlets/
Portal.Playback

```
<AuditRecord Action="Playback" Decision="permit"
Subject="908102000000001001" When="2003-08-12T23:41:16"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ffe">
<Resource Class="PORTAL.Portlets" ID="231"/>
<Attribute Name="taborder" Value="1"/>
<Attribute Name="rid" Value="1"/>
<Attribute Name="tabset" Value="1"/>
</AuditRecord>
```

CHAPTER 7

Setting up Security for Enterprise Portal

This chapter describes how the security functionality provided by Java 2 Enterprise Edition (J2EE) roles defined in the Enterprise Portal (EP) application, application server roles, and Enterprise Security roles is integrated using role mapping.

Overview

Enterprise Security supports J2EE security by providing a bridging mechanism that integrates J2EE, the application server, and Enterprise Security, securing all applications and components hosted on your application server.

EP objects (applications, catalogs, portlets, pages, and so on) can be secured using J2EE roles. Secure EP applications can then access other J2EE services and features on the application server. Enterprise Security provides the underlying login authentication and role authorization services, which enable the application server to determine whether clients have the appropriate access permissions.

Configuring EAServer security For instructions on configuring security profiles, EAServer roles, EAServer listeners, and EAServer identities, see Chapter 11, “Security Configuration Tasks,” in the *EAServer Security Administration and Programming Guide*.

EP maintains a set of J2EE roles that are used to control access to EP objects, such as portlets. These roles must also be known by the application server, either through implicit role mapping or by explicitly mapping the roles.

To access the portal, new users log in to Portal Interface. From Portal Interface, users register with the portal and become a member of the Enterprise Security self-registration group (SelfRegGroup). If you want roles to be granted automatically to users who self-register, grant these roles to the self-registration group—see "Managing a group's roles" on page 40. This assumes that you have created and deployed portlets that users can access. Enterprise Portal includes some sample pages and portlets that are protected by the PortalUser role. If you want users who self-register to have access to these objects, grant the PortalUser role to the self-registration group.

Integrating security models

J2EE, Enterprise Portal, and the application server have overlapping security models. By integrating the security functionality of these models, PortalSession authentication provides:

- Access to server components, subject to user roles, once the user has been authenticated by the server.
- High-availability access (failover) to other configured servers without requiring that the user log in to the new server.
- Stronger security against intrusion from clients who have not been authenticated by the portal.

Once the integration of the three security models is implemented, visitors to the secured system can access any component for which their user roles permit. All requests to applications that are hosted on the server are intercepted by the server, which performs access permission checks. If the user has the appropriate role, the request is processed; otherwise, an error is returned and the request is denied.

Further, when the user successfully logs in to the system, the PortalSession object persists so the user does not have to reenter login information when accessing other Web-based applications that are part of the secured system.

For EAServer, you can extend and maintain the authenticated session beyond the lifetime enforced by the server by using the methods CtsSecurity::SessionInfo::setName and CtsSecurity::AuthService::getCallerPrincipal. If these methods are implemented, then you must also handle the user authorization by implementing either a role service or authorization service. The internal role-checking performed by EAServer does not work unless you add the alternate user name to the authorized user's list for the role. Since the alternate user name that is set using the setName API can be dynamic, the role service or authorization service should work in tandem with the authentication service to authorize the user.

One-time authentication

Upon logging in to the secured system, clients present their credentials, either user name and password, or digital certificates. Once the authentication process is complete, the user's distinguished name (DN) information is stored in a PortalSession object and is available to any component within the application server that has role-based access restrictions.

The PortalSession object is active for a configurable "time-to-live" duration. If the user does not log out of the portal, and if the connection remains active without exceeding the time-to-live parameter, the PortalSession object remains available to all secured components. The default time-to-live duration is one hour.

Since the PortalSession object persists throughout the life of the connection, users do not have to submit credentials when accessing server applications, or when the server fails over to another configured server in the system. The supporting server assumes the persistent PortalSession object and the session continues.

The transparent nature of the PortalSession object allows users to access components without detecting your secured assets. Further, through EJBs, the system can perform specific and targeted access restrictions on assets. For example, you can configure an EJB to strip out certain restricted elements of a query result and return only the results of the query that are available to the user as specified by the user's roles.

This code demonstrates how to do this:

```
import javax.ejb.*;
import javax.naming.InitialContext;
import javax.rmi.PortableRemoteObject;
import com.sybase.ep.security.sessionsvcs.*;
```

```
public class TestComponentBean implements SessionBean {

// the various normal session bean methods must be defined...

// This component method will return different results
// depending on the user’s access to the asset
// "a1=PrivilegedAsset,o=Sybase,c=us"

    public int dynamicQueryMethod() {
        try {
           // retrieve the user’s portalsession object

           InitialContext ctx = new InitialContext();
           PortalSessionHome pshome =
              (PortalSessionHome)PortableRemoteObject.narrow(ctx.lookup(
              "com.sybase.ep.security.sessionsvcs/PortalSession"),
              PortalSessionHome.class);
           PortalSession callerSession = pshome.findByCurrentSession();

           // determine whether or not they are privileged
           boolean privilegedUser = false;
           String barrierAsset = "a1=PrivilegedAsset,o=Sybase,c=us";

           try {
              callerSession.checkAccess(barrierAsset,
                                         PortalSession.READ_ACCESS);

              // if the checkAccess succeeds, the user is privileged
              privilegedUser = true;
           } catch(com.sybase.ep.security.exceptions.SecurityException e) {
           // if a securityexception is thrown, the user is not privileged
           }

           // now build the query and return the results
           String query;

           if(privilegedUser) {
            // the privileged user has full access
              query = "SELECT COUNT(*) FROM UserTable";
           } else {
            // the non-privileged user returns a limited number of rows
              query = "SELECT COUNT(*) FROM UserTable WHERE Privileged = 0";
           }
           int retval = executeQuery(query);
           } catch(FinderException e) {
              throw new EJBException("Caller is not authenticated
                                   to the system");
           } catch(Exception e) {
          // we're not expecting any other exceptions
             throw new EJBException(e);    }
     }
        private int executeQuery(String query) {
     // get connection cache, execute query and return the result as an integer

    }
}
```

EAServer example

This example illustrates how a Java client connects to EAServer via the InitialContextFactory using Enterprise Security credentials:

```
Properties p = new Properties();

p.put(Context.INITIAL_CONTEXT_FACTORY,
      "com.sybase.ejb.InitialContextFactory");

// For certificate authentication, different properties will be set

p.put(Context.PROVIDER_URL, "iiop://[portalserver]:9000");
p.put(Context.SECURITY_PRINCIPAL, "epusername");
p.put(Context.SECURITY_CREDENTIALS, "eppassword");

InitialContext ctx = new InitialContext(p);
```

High availability

If your environment is configured with more than one application server instance for high-availability (failover) purposes, the failover server accesses the client credentials in the PortalSession object and the failover takes place transparently to the user.

Java component access

Since clients authenticate to the secured system with the PortalSession object, any Java component that is hosted on the application server can retrieve the PortalSession object using the following code:

```
InitialContext ctx = new InitialContext();
PortalSessionHome pshome =
ctx.lookup("com.sybase.ep.security.sessionsvcs/PortalSession");
```

```
PortalSession session = pshome.findByCurrentSession();
```

Bridging the security mechanisms

In earlier versions of Enterprise Security, you had to explicitly map EAServer roles to Enterprise Security roles, but beginning with Enterprise Security 6.0, mapping can be done automatically. You can still map roles explicitly if you choose. If you create new J2EE roles, explicitly mapping the J2EE roles to Enterprise Security roles requires:

- Mapping J2EE roles to EAServer roles
- Mapping EAServer roles to Enterprise Security roles

Roles not found warnings Although you need not explicitly map J2EE roles to EAServer roles or Enterprise Security roles, "roles not found" warnings are written in the server's log file if you do not. You can safely ignore these warnings, or you can eliminate them by explicitly mapping the roles.

Implicit role mapping

Beginning with Enterprise Security 6.0, role mappings are performed implicitly, so you no longer need to explicitly map custom J2EE roles to Enterprise Security roles in the *security.properties* file. Enterprise Security still checks for role mappings in *security.properties* before performing implicit role mapping.

For example, if a Portal Interface user attempts to access a portlet that is protected by PortletRole:

1. Enterprise Security first checks *security.properties* to see whether an explicit mapping exists for PortletRole. If a role mapping exists for PortletRole, Enterprise Security checks to see whether the Portal Interface user has this role, and permits or denies access to the portlet based on the result.

   This duplicates the functionality of earlier versions of Enterprise Security.

2 If a role mapping for PortletRole is not found in *security.properties*, Enterprise Security searches the ACDB for a role with the name PortletRole. If there is exactly one role with this name, Enterprise Security checks to see whether the user has this role, and permits or denies access to the portlet accordingly.

If there are zero, two, or more roles in the ACDB with the name PortletRole, a warning is written to *security.log* that says the system cannot create a default mapping, and access to the portlet is denied.

Implicit role mappings are performed dynamically, so you need not reboot the server after adding a new role.

Implicit role mapping is enabled by default. To disable this feature, edit the *security.properties* file to set the value of the defaultRolemappingEnabled property to false. If you disable implicit role mapping, you must explicitly map roles in *security.properties*—see “Mapping J2EE roles to EAServer roles” and “Mapping EAServer roles to Enterprise Security roles” on page 148.

If you write a custom authorization delegate that does not use the standard ACDB schema, you must disable implicit role mapping.

Mapping J2EE roles to EAServer roles

Roles define what users and groups of users are permitted to do, such as access components for read-only, read and write, and administration (start, stop, or execute methods).

When you install and deploy Enterprise Portal into EAServer, the J2EE roles defined in EP are automatically mapped to EAServer roles. If you add new EP roles, you should map the roles to EAServer roles. In the Servers folder in Jaguar Manager, the following folders each contain role-associated properties that you need to map to J2EE roles:

- Applications
- Web Applications
- Packages

For each of these items, click on the icon to expand the folder contents, then for each item in the folder:

1 Select the item, right-click, and select Properties.

2 Display the Role Mapping tab, and configure the mappings as follows:

a To add a logical J2EE role name, click Add, and enter the role name.

b To specify the mapped EAServer role, click in the right column, opposite the J2EE role to be mapped, then use the pull-down menu to choose the mapped role.

c Optionally, add a description of the role in the field labeled Description.

d Click OK to save the changes.

If necessary, define new EAServer roles to be used by callers of the component and map them to J2EE roles in the package properties.

For more information about role mapping and how to create roles in EAServer, see the *EAServer Security Administration and Programming Guide*.

Mapping EAServer roles to Enterprise Security roles

EAServer roles are mapped to Enterprise Security roles in the *security.properties* file. The following role mappings are created automatically by the Enterprise Security installer in *security.properties*:

```
easerverRolemap.epdefault_0.epdn=PortalWebPlugin
easerverRolemap.epdefault_0.jagrole=PortalWebPlugin
easerverRolemap.epdefault_1.epdn=PortalSecOfficer
easerverRolemap.epdefault_1.jagrole=PortalSecurityOfficer
easerverRolemap.epdefault_2.epdn=PortalAdmin
easerverRolemap.epdefault_2.jagrole=PortalAdmin
easerverRolemap.epdefault_3.epdn=PortalGuest
easerverRolemap.epdefault_3.jagrole=PortalGuest
```

In earlier versions of Enterprise Security, roles were mapped using their names; beginning with version 6.0, you must use the DN instead of the name. For each of these pre-mapped Enterprise Security roles, the role DN is the same as the role name.

❖ Mapping EAServer roles to Enterprise Security roles

1 Using any standard ASCII text editor, open the *security.properties* file, which is located in the *java/classes/com/sybase/ep/security* subdirectory of your EAServer installation, and use this syntax to add a role mapping pair:

easerverRolemap.epdefault_*n*.epdn=*secRoleDN*
easerverRolemap.epdefault_*n*.jagrole=*EAServerRole*

where:

- *n* is an integer that creates a unique mapping key (`epdefault_`***`n`***)
- *secRoleDN* is the DN of the Enterprise Security role
- *EAServerRole* is the name of the corresponding EAServer role

For example, the following lines map a role called NewRole using the Enterprise Security role DN and the EAServer role name:

```
easerverRolemap.epdefault_4.epdn=r1\=NewRole,o\=Sybase,c\=US
easerverRolemap.epdefault_4.jagrole=NewRole
```

2 Stop and restart EAServer for the changes to take effect.

Setting up WebLogic authentication

To configure Enterprise Security in WebLogic, the WebLogic administrator must configure the Sybase security provider for the default WebLogic security realm, using either:

- The securetool wls_configmw command—see wls_configmw on page 85, or
- The WebLogic Server Console—see “Configuring Enterprise Security using the WebLogic Server Console” on page 23

Once configured, the WebLogic server contains two authentication providers, and a SybaseIdentityAsserter provider, which handles certificate authentication. To enable the SybaseIdentityAsserter to perform certificate authentication, you must first enable two-way SSL on your WebLogic server. See the BEA WebLogic documentation.

The authentication providers include the default provider, which houses the standard WebLogic users and any users added to the WebLogic user database, and the Sybase provider. When authentication is required, the two authentication providers are queried in sequence. If a user cannot authenticate against the WebLogic user database, WebLogic attempts to authenticate the user against the Sybase provider.

As part of the authentication process, the Sybase provider adds principals to the WebLogic subject structure. Each principal represents one Enterprise Security role. The value of the Role Prefix field in the Sybase provider's Details tab specifies the prefix used when adding principals. For example, if the Role Prefix field is blank, the PortalAdmin user might have this principal added "r1=PortalAdmin,o=Sybase.com,c=us". If the value of Role Prefix is "SybaseRole", then the principal would be "SybaseRole:r1=PortalAdmin,o=Sybase.com,c=us". If the prefix is not blank, the colon is added implicitly to separate the prefix from the rest of the principal.

To map J2EE security roles to BEA principals:

1. In the WebLogic Server Console, launch the deployment descriptor editing application from the context menu of the J2EE entity whose deployment descriptor you want to edit.
2. Expand the WebLogic EJB JAR folder, then highlight the Security Role Assignments folder, right-click, and select Configure New Role Assignment.
3. In the Configuration dialog box, enter the role name, one or more principal names, and click Apply.

   Each role assignment maps a local J2EE security role to one or more WebLogic principals. When a user attempts to access a resource that has a role defined, the user must have at least one of the specified principals. Therefore, you can combine Sybase role principals with local role and group principals in the WebLogic security database.

Setting Enterprise Portal and session timeouts

The duration of Enterprise Portal sessions are controlled by the following timeout properties, which the administrator can adjust, based on the requirements of the enterprise:

- sessionDuration – defines a specific amount of time that a user's Enterprise Security PortalSession object remains active. The PortalSession is the token by which the portal determines a user's access rights. When a user remains logged in for the specified duration, the PortalSession times out, and the user must log in again.

- com.sybase.jaguar.server.authtimeout – defines the period of time, during which, if no user activity is detected, the EAServer authentication session times out, and the browser becomes an anonymous user. If activity continues, a user can remain logged in to the portal indefinitely.
- com.sybase.jaguar.webapplication.session-config – defines the EAServer HTTP session timeout. If the HTTP session times out first, the session is reauthenticated automatically, but information about the HTTP session may be lost.
- To limit everyone to a specific amount of time in the portal, set sessionDuration and com.sybase.jaguar.server.authtimeout to the same value. To limit activity, set com.sybase.jaguar.server.authtimeout to the preferred value, and set sessionDuration to twice that value. Setting the value of sessionDuration too high wastes resources.

❖ Setting the sessionDuration property

1. In any text editor, open the *securities.properties* file, which is located in the *java/classes/com/sybase/ep/security* subdirectory of your EAServer installation.
2. Search for this line:

   ```
   sessionDuration=3600
   ```

   Sybase recommends that you set the value of sessionDuration to 3600. The value of this property is expressed in seconds; the default value, 3600, equals one hour. This value defines the absolute time limit that a user can stay logged in to the same portal session. The time begins when the user logs in, and when time expires, the user must to log in again and establish a new session.
3. Save and close the file.

❖ Setting the com.sybase.jaguar.server.authtimeout property

1. Assuming EAServer is running, start Jaguar Manager.

   On UNIX or Linux:

   a. Set the JAGUAR environment variable to the location of your EAServer installation; for example:

      ```
      setenv JAGUAR /work/Sybase/EAServer
      ```

   b. Change to the *$JAGUAR/bin* directory, and run:

      ```
      jagmgr.sh
      ```

On Windows, select Select Start | Programs | Sybase | EAServer 4.2.2 | Jaguar Manager.

2 In the Jaguar Manager window, select Tools | Connect; then, on the connection screen, enter:

- User Name: `jagadmin`
- Password: the password for jagadmin; the default is an empty string
- Host Name: *your machine name*
- Port Number: `9000`

3 Click Connect.

4 Expand the Servers folder, then highlight Jaguar, and select File | Server Properties.

5 Select the All Properties tab, and scroll through the list of server properties to find:

```
com.sybase.jaguar.server.authtimeout
```

6 Double-click the property name, and set its value to 1800 (30 minutes).

7 Click OK.

8 Highlight Jaguar, and select File | Shutdown and Start.

❖ Setting the com.sybase.jaguar.webapplication.session-config property

1 In Jaguar Manager, expand these successive folders: Servers, Jaguar, Web Applications.

2 Highlight “onepage,” and select File | Web Application Properties. In the Web Application Properties dialog box, select the General tab.

3 Set the value of Session Timeout to an appropriate value; the default is 60 minutes.

4 Click OK to save your changes.

CHAPTER 8 Securing Accounts and Assets

This chapter describes basic security enhancements that further secure an e-business system.

Enabling account and asset locks

Now that you have created users, user accounts, assets, roles, and so on, you can enable Enterprise Security account lock features to secure user accounts and assets from unauthorized access.

User login lockout

Login control allows the System Administrator to specify the number of unsuccessful login attempts that can be made before the account is locked. When login control is configured, the user must provide a valid password within the allotted number of attempts. If the user fails to enter the correct password, the account is locked. The account can be locked for a configurable amount of time, or permanently locked, which requires that the security officer manually change the status of a user account from locked to unlocked.

Once the user makes a login request, a session starts and the request is sent to the security framework. Security forwards the request to either the ACDB authentication delegate or the LDAP authentication delegate, depending on how security is configured.

The delegates (either LDAP or ACDB) then contact an independent module, called the Lock Manager, which scans the login control policy to determine whether to allow the login to succeed.

The login control policy is stored in the ACDB. See Chapter 15, "Configuration Properties."

The login control policy contains five important parameters:

- Whether to enable login lock functions in the secured system. If you are using the LDAP delegate for authentication and want to leverage the account locking functionality of LDAP, set the value to false. Otherwise both the policies take effect and result in an undesirable situation.
- How many invalid login attempts are permitted.
- How long the account is locked once the login attempts have exceeded the limit. This can be a predetermined amount of time, set in minutes, or it can be a permanent lock, requiring the PSO to manually unlock the account.
- The length of time during which invalid login attempts are counted and decremented from the permitted number of attempts. After the specified amount of time, the count resets to zero. The time begins with the first failed login attempt.
- Whether the invalid login count should be reset to zero upon successful login. This compensates for the occasional user-input error.

The Lock Manager works with the login control policy and ACDB to fulfill the login lock features. When you install Enterprise Security, the ACDB is modified to contain a table that stores login lock information.

Table 15-11 on page 243 defines the properties that you can edit to configure the login lock component.

Permissions required to manage account locking To display a user's login lock information, you must have READ permission on the subject controlling asset. To lock or unlock a user's account, you must have UPDATE permission on the subject controlling asset.

Authorization lock

The purpose of an e-business system is to make the enterprise’s assets, applications, data, and services available to its users. As you extend your enterprise’s assets to users, there may be assets that you want to make available to all users, to make available to a selected group of users, or to keep entirely private.

This is accomplished by creating roles, defining access control elements to define access permissions of those roles, and granting the roles to the appropriate users or groups of users. This allows only authorized users to access the assets.

However, the PSO may want to detect access attempts by unauthorized users, even though such attempts fails. To further tighten security, the PSO can place a lock on a user’s account to prevent them from accessing any assets, including the ones to which he or she is authorized. An authorization lock prevents the user from accessing any assets.

Once a session has started, the user can attempt to access an asset within the system. The session object invokes the Lock Manager, which then scans the asset access control policy.

The asset access control policy contains six important parameters:

- Whether to enable authorization lock functions in the secured system.
- The number of unauthorized access attempts that can be made before the account is locked.
- The time duration for which the system increments unauthorized access attempts.
- How long the account is locked after exceeding the specified number of access attempts.
- Whether to terminate the user session once the user has exceeded the specified number of access attempts.
- Whether to lock the user account from login if and when the session is terminated.

Similar to login control (“Enabling account and asset locks” on page 153), the Lock Manager module regulates who can access your enterprise’s assets. Once a user makes the specified number of invalid access attempts on an asset to which he does not have the appropriate permissions, the user’s authorization ability is locked and he cannot access any assets, even those for which he does have access permissions.

The Lock Manager reads the access control policy and gathers data from the ACDB to determine authorization lock status for each user.

Table 15-11 on page 243 defines the properties that you can edit to configure asset authorization locks.

Locking a user’s account

This section describes how to lock and unlock a user’s account from Enterprise Security Manager.

Note Only the PSO can perform login lock and unlock, and authorization lock and unlock tasks.

❖ Locking a user’s account

1 In the Organization Manager tree view, select the organization and highlight Users.

2 In the right pane, highlight the name of the user, right-click, and select Edit User.

3 In the Edit User dialog box, select Account is Disabled, and click OK.

❖ Unlocking a user’s account

1 In the Organization Manager tree view, select the organization and highlight Users.

2 In the right pane, highlight the name of the user, right-click, and select Edit User.

3 In the Edit User dialog box, unselect Account is Disabled, and click OK.

Expiration of user accounts

Enterprise Security further tightens access control policies by implementing methods that allow system administrators to restrict user access by setting a fixed expiration date, or by denying access to users who have not logged in within a specific time period. By default, neither the expiration by date or inactivity restriction is enabled. Table 15-10 on page 243 lists the properties that configure user account expiration.

Fixed expiration date

You can specify a date after which the user is no longer allowed access. If you do not specify a value for this parameter, there is no expiration date for the user. You can set the fixed expiration date of a user using Security Manager, as well as programmatically using the SubjectManagement.setExpirationDate method.

Existing accounts do not have a fixed expiration date, even after upgrading Enterprise Security.

❖ Setting a fixed expiration date

1. Start Enterprise Security Manager—see “Launching Enterprise Security Manager” on page 28.
2. In the Organization Manager tree view, select the organization and highlight Users.
3. In the right pane, highlight the name of the user, right-click, and select Edit User.
4. In the Edit User dialog box, select Account Has Fixed Expiration Date, and enter the expiration date.
5. To save your changes, click OK.

Inactivity expiration

Activity is defined as a successful authentication to the system. If a user has not successfully accessed the system within the specified time range, the account is automatically deactivated, and the user cannot access the account until the security officer reactivates it.

By default, Enterprise Security system users, such as the “pso,” “portaladmin,” and “guest,” are not subject to inactivity restrictions.

The administrator can override the default behavior for any user in the system—including Enterprise Security system users—by selecting Account Never Expires Due to Inactivity, or by setting a fixed expiration date for any user in the system using Enterprise Security Manager.

❖ Disabling account inactivity restriction

1. In the Organization Manager tree view, select the organization and highlight Users.
2. In the right pane, highlight the name of the user, right-click, and select Edit User.

3 In the Edit User dialog box, select Account Never Expires Due to Inactivity, and click OK.

Verifying passwords

The Enterprise Security SMAPI allows you to develop a password-strength verification component to verify new passwords. Once a password-strength verification component has been deployed and configured, it is called automatically when:

- Passwords are set or changed using any of the GUI components: Enterprise Security Manager, Portal Studio, self-registration, and so on.
- Users are created using one of the SubjectManagement.create methods.
- Passwords are set using either the SubjectManagement.setPassword method or the SubjectManagement.setInfo(InfoConstants.PASSWORD) method.

If you are using the password validation component from a version of Enterprise Security earlier than 6.0, you can upgrade your existing component—see “Upgrading an existing password validation component” on page 160.

If a password verification fails, the information in an Exception object is passed to the client, and the current operation completes without modifying the underlying data store.

Configuring the sample password-strength verification component

A sample password-strength verification component is defined in the EJB-JAR file, *samples.jar*, which is located in the *samples/bin* subdirectory of your Enterprise Security installation.

The password-strength verification component implements the com.sybase.ep.security.management.PasswordStrengthVerifierHome home interface and the com.sybase.ep.security.management.PasswordStrengthVerifier remote interface. It also contains the code to enforce the following password constraints:

- Must be more than five characters

- Must contain at least one digit

If password verification fails, the information in the SMException object is passed to the client, and the current operation completes without modifying the underlying data store.

❖ Setting up the sample component in EAServer

1. Deploy *samples.jar* into EAServer. At a command line, change to the *Security/samples/bin* directory, and enter:

   ```
   jagtool deploy -type ejbjar samples.jar
   ```

2. Link the ejb/PasswordStrengthVerifier EJB reference in both the SubjectQueries and SubjectManagement beans to the newly defined password strength verifier. At a command line in the *Security/samples/bin* directory, enter:

```
jagtool ejbref
Component:com.sybase.ep.security.management/SubjectQueries -refname
ejb/PasswordStrengthVerifier -value samples/PasswordStrengthVerifier

jagtool ejbref
Component:com.sybase.ep.security.management/SubjectManagement -refname
ejb/PasswordStrengthVerifier -value samples/PasswordStrengthVerifier
```

3. Restart your application server.
4. Enable password strength verification using Enterprise Security Manager.
   a. In the Domain Manager, select the domain to which you want to apply the password-strength verification.
   b. In the right pane, right-click, and select Configure Password Properties.
   c. In the Configure Domain Password Properties dialog box, select Enable Password Strength Verification.

❖ Setting up the sample component in WebLogic

1. Deploy *samples.jar* to your WebLogic application server—see the BEA documentation.
2. Using a text editor, open *weblogic-ejb-jar.xml*, located in the *sybepsecurity/sybepsecurity.ear.exploded/ com.sybase.ep.security.management.jar/META-INF* subdirectory of your WebLogic installation, and replace both instances of “choose.your.passwordStrengthVerifier” with “samples/PasswordStrengthVerifier”.

Save and close the file.

3 Restart your application server.

4 Enable password strength verification using Enterprise Security Manager.

   a In the Domain Manager, select the domain to which you want to apply the password-strength verification.

   b In the right pane, right-click, and select Configure Password Properties.

   c In the Configure Domain Password Properties dialog box, select Enable Password Strength Verification.

Upgrading an existing password validation component

The component that was used to verify passwords in versions of Enterprise Security earlier than 6.0 was called the password validation component. The core interfaces for this component have been deprecated. When you upgrade to Enterprise Security 6.0, the installer automatically enables the old password validation component using a “compatibility” password-strength verifier, which bridges the old interfaces to the new APIs. To perform this task manually, use the following procedure.

❖ Bridging the version 5.0 interfaces to the 6.0 interfaces

1 Link the ejb/PasswordStrengthVerifier EJB reference in the SubjectQueries and SubjectManagement beans to the newly-defined password strength verifier.

   At a command line, change to the *EAServer/bin* directory, and enter each jagtool command on a single line:

```
jagtool ejbref
Component:com.sybase.ep.security.management/SubjectQueries
-refname ejb/PasswordStrengthVerifier -value
com.sybase.ep.security.management/CompatibilityPasswordStrengthVerifier

jagtool ejbref
Component:com.sybase.ep.security.management/SubjectManagement
-refname ejb/PasswordStrengthVerifier -value
com.sybase.ep.security.management/CompatibilityPasswordStrengthVerifier
```

2 Open the *security.properties* file, and set the value of the deprecatedPasswordValidationComponent property to the JNDI name of your custom password validation component.

3 Reboot the server.

Changing expired passwords

This method, added to the SubjectQueries interface, allows users to change their password after it expires:

```
boolean changePassword(String username, String oldPassword, String
newPassword) throws RemoteException, SMException
```

To call changePassword, users need not be authenticated; they need only to supply their user name, old password, and new password. You can call this method to change an expired password for a limited period of time after the password expires. To configure this time period, use Enterprise Security Manager—see “Configuring password properties for a security domain” on page 98.

If case of a failed authentication attempt, you can call the PortalSession::getAuthenticationFailureCode method, which returns a String that describes the reason for the failure. The authentication failure code persists with the unauthenticated session for the life of the session, but is overwritten by future authentication attempts on the same PortalSession instance. A new failure code indicates that a password has expired but the user still has the opportunity to change the password using the changePassword method. The failure codes are defined in the com.sybase.ep.security.sessionsvcs.AuthenticationFailureReasons interface, which is documented in Javadoc. Open a browser and access *docs/html/index.html* in your Enterprise Security installation

Salting passwords

Enterprise Security allows you to store passwords in a “salted” format. A salted password contains random characters, so the password as stored does not permit access to the system.

To enhance the security of stored passwords by enabling salting or password encryption, the administrator can set the password properties in the *security.properties* file. For example, to define MD5 as an allowable encoding type, use a text editor to open *security.properties*, and append “,MD5” to the passwordAllowedEncodings property value:

```
passwordAllowedEncodings = SHA,MD5
```

The location of *security.properties* depends on your application server.

- EAServer – *JAGUAR/java/classes/com/sybase/ep/security.*
- WebLogic – *BEA_ROOT/sybepsecurity/etc/com/sybase/ep/security*.

Passwords are encoded using industry-standard hash algorithms. Examples of these are “MD5” and “SHA”. In addition, the “TXT” proprietary encoding type is available, which does not actually encode passwords but leaves them in an unencoded format. The “TXT” encoding type’s primary use is to change the password of a user—typically, performed by the system administrator—when the user’s password is lost. For security purposes, users with TXT-encoded passwords cannot authenticate. The password* properties described in Table 15-5 on page 239 define acceptable password formats.

To enable TXT-encoded password authentication:

1 Using a text editor, open *security.propeties.*

2 Append “,TXT” to the allowedEncodings property value. For example:

```
passwordAllowedEncodings = SHA,TXT
```

3 Set the value of allowUnsaltedAuthentications to true.

Once access has been regained, change the user’s password using the administration tool, and disable TXT encoding.

For information about SHA, see the Secure Hash Standard at http://www.itl.nist.gov/fipspubs/fip180-1.htm. For information about MD5, see the MD5 Message-Digest Algorithm at http://www.ietf.org/rfc/rfc1321.txt.

CHAPTER 9 Proxy Authentication

This chapter describes how to implement single sign-on to enterprise resources.

Proxy authentication information identifies a user via user name and password (credentials), and the URL used to connect to the data source or service. When a user logs in to a system secured with Enterprise Security, the proxy authentication information can be retrieved from the ACDB and presented to the back-end data source to authenticate the user without requiring the user to reenter his or her user name or password.

To use proxy authentication information for single sign-on, the data source (asset) must first be defined in the ACDB. The proxy authentication mechanism determines whether a user’s credentials allow access to the back-end data source; it does not enforce the user’s privileges to read, write, update, or execute commands on the data source.

SMAPI provides management and administrative functions for managing proxy authentication information. You must write your own custom application to invoke the API that retrieves the proxy authentication information, and the application must call PortalSession.findProxyAuthenticationInfo.

Retrieving proxy authentication information

In SMAPI, the ProxyAuthenticationInfoManagement interface provides methods to read, update, and delete proxy authentication information for any asset.

SMAPI method changes In Enterprise Security version 6.0, the permissions required to run SMAPI methods have changed. Table 5-26 on page 113 describes the permissions required to run the ProxyAuthenticationInfoManagement interface methods.

All the SMAPI interfaces have been modified to use the object ID, instead of the DN, as the primary key. Methods still accept a DN to maintain backward compatibility, but performance improves if you use the object ID.

To view the SMAPI documentation, open a browser, and access *docs/html/index.html* in your Enterprise Security installation; then, select the com.sybase.ep.security.management package.

When building a single-sign on solution, an application that is designed to retrieve a user's proxy authentication information (also called credentials) uses the PortalSession.findProxyAuthenticationInfo method. This method accepts the asset's DN as a parameter and returns the user's credentials if the user has read permissions for the specified asset. The user must have read permission for the method to return the user's credentials.

findProxyAuthorization method The findProxyAuthorization method is supported for backward compatibility but is being replaced by the findProxyAuthenticationInfo method, which accepts an asset DN or a unique asset name. If the asset name is specified instead of the DN and the asset name is not unique, an ObjectNotFoundException is thrown. For more information, see the Javadocs in the *docs/html/index.html* directory of your Enterprise Security installation.

The findProxyAuthenticationInfo method searches for credentials in this order:

- User-based
- Role-based
- Asset-based

The search for user credentials stops as soon as the first set of credentials is found. For example, the methods search for user-based credentials first. If user-based credentials are not found, the methods then search for role-based credentials, and so on. If, however, user-based credentials are found, the search stops and the methods do not search for role- or asset-based credentials. You cannot use the PortalSession bean to modify the credentials.

If the user does not have read permission on the specified asset, no proxy authentication information is returned.

User-based proxy authentication information

Each user who is going to access secured assets via proxy authentication information can have user-based proxy authentication information in the ACDB. Additionally, each user can have multiple entries for any number of assets defined in the ACDB. If a user has user-based proxy authentication information defined for an asset, that authentication information takes precedence over asset-based proxy authentication information, and the role-based proxy authentication information that he or she assumed for the active session when logging in to the back-end resource.

Users can create, update, and delete their own user-based proxy authentication information. Also, the security officer (PSO role) can create, update, and delete user-based proxy authentication information for any user. To do this using Enterprise Security Manager, see “Managing user-based proxy authentication information” on page 49.

Role-based proxy authentication information

Any asset defined in the ACDB can have a list of roles that have role-based proxy authentication information that allows access to the asset. All users who are granted a role can access the proxy authentication information created for that role.

For example, if user “Bob” wants to access “AssetX,” he must have either user-based proxy authentication information for AssetX, or he must be granted a role that has the appropriate role-based proxy authentication information to access AssetX.

When multiple roles have proxy authentication information defined for the same asset, the roles must be assigned a priority order so a user who has more than one role that has proxy authentication information for any given asset can use the role-based proxy authentication information that has the highest priority. For example, Bob has two roles, Manager and Engineer, both of which have proxy authentication information pertaining to a back-end service. If only the Engineer role-based proxy authentication information should be used to authenticate to the service, the Engineer role should have a higher priority than the Manager role. The AssetManagement SMAPI interface provides the setRoleProxyAuthInfoPriorities method, which enables you to set the priority of the roles that have proxy authentication information defined.

If multiple roles have proxy authentication information defined for an asset, the following rules apply:

- If all the roles are granted explicitly to a user (user roles or group roles), then the proxy authentication information for the role with the highest priority is returned.
- If a parent role is not granted explicitly to the user, the child role's proxy authentication information always overrides the parent role's proxy authentication information.

For example, assume that there are three roles: role1, role2, and role3, and role3 inherits from role2. Proxy authentication information is defined for all three roles. The roles, in order of priority from highest to lowest, are role1, role2, and role3.

- If role1, role2, and role3 are granted to the user, proxy authentication information for role1 is returned because role1 has the highest priority.
- If role2 and role3 are granted to the user, proxy authentication information for role2 is returned. Although role3 is the child of role2, role2 is granted explicitly, and has higher priority.
- If only role3 is granted to the user, proxy authentication information for role3 is returned. role3's parent, role2, is not granted explicitly to the user, so the child role overrides the parent role.

  If role3 does not have any proxy authentication information defined, proxy authentication information for role2 is returned. A child role without proxy authentication information defined inherits the parent's proxy authentication information.

Only the PSO, can create, update, and delete role-based proxy authentication information.

Asset-based proxy authentication information

Each asset that allows user access via proxy authentication information is defined by a single entry in the ACDB. All users who have read permission on an asset can retrieve the proxy authentication information associated with the asset. Only the PSO can create, update, and delete asset-based proxy authentication information.

CHAPTER 10 Configuring LDAP Authentication

The authentication delegate is designed to allow the ACDB to access security information when a user has the Lightweight Directory Access Protocol (LDAP) data store for their system.

Overview

Enterprise Security uses a mechanism known as an “authentication delegate” to allow pluggable authentication capabilities. One implementation of this pluggable authentication mechanism is the LDAP authentication delegate, which can be used to direct authentication requests to an LDAP server. This delegate allows the replication of users, and their group and role memberships, from the LDAP server to the ACDB.

Authentication delegate

The authentication delegate is a mechanism used to synchronously and asynchronously migrate users, groups, and roles from the LDAP server into the ACDB. When subjects are modified, removed from, or added to the LDAP server, these changes are propagated to the ACDB during the next LDAP server replication to the ACDB. Users are never automatically removed from the ACDB; however, if a user is removed from the LDAP server, that user cannot authenticate, even if they have a record in the ACDB. Users are “replicated” one record at a time. The replicated record belongs to the authenticating user.

Note Changes made to the ACDB are not replicated to the LDAP data store.

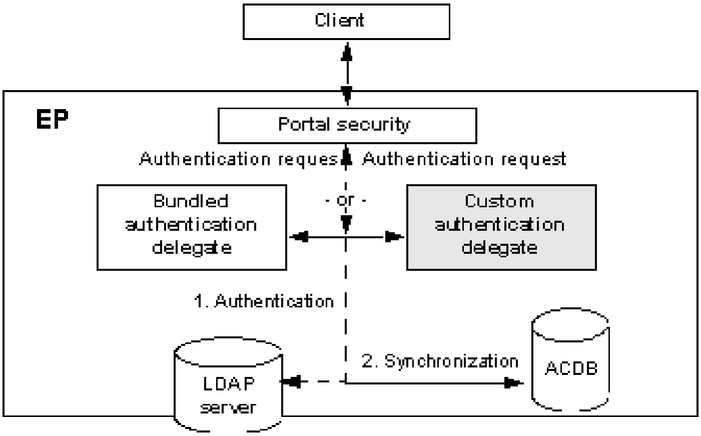
When a user logs in to the system:

1. The security services authenticate the user by accessing the LDAP server.
2. If the user is in the LDAP server, the information is replicated to the ACDB for that session and the user can access the secured system.

   If the user is not in the LDAP server, an error displays and the session does not open.

Users with zero-length passwords cannot authenticate Before the authentication delegate presents credentials to the LDAP server, it immediately rejects zero-length passwords. Therefore, any users in the LDAP server with zero-length passwords cannot successfully authenticate to the system.

Figure 10-1: Authentication process



LDAP information is replicated to the ACDB as needed. The authentication is verified against the LDAP server each time a user authenticates to the system, and any user data is replicated once the authentication is determined to be successful.

The LDAP server does not have to be on the same platform as the installed Sybase products.

The authentication delegate:

- Is configurable to one LDAP server.

- Establishes a cache of connections to the LDAP server. This reduces the time taken to authenticate users and decrease load on both the LDAP server and the ACDB.
- Provides flexible mapping from certificate binary and DNs to LDAP records. See "Certificate mapping" on page 176 for details.
- Maps credentials to a DN on the LDAP server.

  The credentials supplied can be mapped to any attribute on the LDAP server (user ID—uid—the most common, and default case). Once the DN is established, the delegate "binds" to the LDAP server using the DN and the supplied password. This process validates the user's password on the LDAP server, and if the LDAP server rejects the password, the authentication attempt fails.
- Once the user has been authenticated, the record is replicated to the ACDB. Ensure that the LDAP account specified by the ldap.connection.bindname and ldap.connection.bindpassword properties has sufficient privileges on the LDAP server to access all of the relevant fields within a user's LDAP record.

Note LDAP is not case sensitive, and ACDB is case sensitive. Therefore, if a user's `uid` attribute within LDAP is modified to change case, a new replicated ACDB record is created when the user attempts to authenticate and the old record is orphaned.

Subject replication

User records within LDAP must be instances of the object class inetOrgPerson, or a subclass, to function without extra configuration steps. If this is not the case, you must change the attribute mappings and the default search filter. For more information, see "Attribute mapping" on page 174, and the ldap.searchFilter property in Table 15-4 on page 238.

The attributes that are used by default, and replicated into the ACDB from the LDAP data store are:

- uid – user ID.
- cn – common name; typically, first name and last name.
- mail – e-mail address.

- telephoneNumber – business telephone number.
- givenName – first name.
- sn – last name (surname).

LDAP support includes the following functionality and restrictions:

- The LDAP authentication delegate replicates a single user record when authenticating a user.
- Group entries within the LDAP data store can be mapped to group and role entries within the ACDB.
- If a user within the LDAP data store is a member of an LDAP group that has a mapping defined to an ACDB group or role, that membership remains intact when the user is replicated to the ACDB. This is the only membership that is replicated; role membership of groups is not replicated.
- Group and role memberships are replicated from the LDAP server to the ACDB; however, the groups and roles themselves are *not* replicated. It is the administrator’s responsibility to create both the mappable entries in the LDAP data store, and the entries in the ACDB.

  Enterprise Security can map LDAP groups to Enterprise Portal groups or roles. To successfully map an LDAP group, it must be an instance of one of the following LDAP object classes:

  - groupOfNames
  - groupOfUniqueNames
  - groupOfURLs
  - group

  For definitions of the groupOfNames, groupOfURLs and groupOfUniqueNames object classes, see RFC 2256 - A Summary of the X.500(96) User Schema for use with LDAPv3 at http://www.faqs.org/rfcs/rfc2256.html. For a definition of the group object class, see the Microsoft Active Directory schema reference on the Microsoft Web site at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netdir/ad/active_directory_schema.asp.

  The mapping is defined internally using DN pairs. If an LDAP DN points to a group, the corresponding ACDB entry is defined by an ACDB DN.

- The administrator can add users to the LDAP group by adding their DN to the uniquemember attribute of the LDAP group, or in the case of dynamic groups, by modifying the memberurl attribute to include the appropriate search filter. These tasks are usually accomplished using the tools provided by the LDAP server vendor.

  When adding a user as a member of a LDAP group, use correct LDAP terminology for setting the user.
- Replication can optionally run at user-defined intervals or times through the use of customized scheduling software. A component method is provided that can be called to replicate users on demand. This method performs no authentication, and does not remove the need to perform an authentication check against the LDAP server the next time the user logs in.
- LDAP supports nested groups; however, nested group memberships are ignored by the current delegate implementation.
- Role and group memberships within Enterprise Portal can be mapped to the results of an LDAP search filter. This enables you to create default roles for users within a specific organization, or for all users. It also facilitates integration with LDAP servers on which a PSO may not have administrative privileges; for example, a Notes LDAP server, or a public LDAP server.

Consistency checks

When LDAP authentication components initialize themselves, they automatically perform a consistency check that reports on:

- Mappings with invalid values:
  - LDAP group does not exist
  - Enterprise Security role or group does not exist
  - Enterprise Security role or group mapped to more than one LDAP group
- Important roles within Enterprise Security that do not have mappings defined to the LDAP server: PortalAdmin, PortalSecOfficer, and PortalWebPlugin.

Do not disable consistency checker Although you can disable the automatic invocation of the consistency checker using a configuration parameter, Sybase strongly recommends that you do not.

The warnings generated by the consistency checker are important in the operation of the LDAP replicator. Possible reasons for warnings include: the LDAP group cannot be found, an EP role or group cannot be found, or important EP roles are not correctly mapped to LDAP entities.

Warnings generated by the consistency checker are written to *security.log,* and usually contain sufficient information for an administrator to resolve the problem.

Setting up LDAP support

To set up your security system to use an LDAP data store, you must configure the LDAP properties, and configure the authentication delegate in your application server.

To configure LDAP properties, edit the *security.properties* file, located in *JAGUAR/java/classes/com/sybase/ep/security*—see Table 15-4 on page 238 for the property descriptions. The values you provide depend entirely on your LDAP server and the structure of your ACDB.

Active Directory sample

This sample illustrates the minimum configuration for a Microsoft Active Directory server. The property values are appropriate for a Windows 2000 Server installation configured for the myorg.com domain. Based on this configuration, users can authenticate with the “username@myorg.com” user name token, and the password specified in the Active Directory administration tool:

```
// Specify the host name
ldap.connection.host=myserver

// Define the searchBase of the LDAP server, which typically
// contains the components of a domain name.
ldap.searchBase=dc=myorg,dc=com

// Many LDAP servers allow anonymous browsing, but the default
// Windows 2000 installation does not. Use the guest account,
// which generally has no password, to perform LDAP queries.
```

```
ldap.connection.bindname=guest@myorg.com

// Leave the password blank, unless it has been modified.
ldap.connection.bindpassword=

// The following two properties map the Enterprise Security
// attributes (attributeMappingNames) to the LDAP attributes
// (attributeMappingValues).
// Active Directory uses the attribute userPrincipalName, instead of uid.
ldap.attributeMappingNames=uid, cn, email, telephoneNumber, firstName,
lastName
ldap.attributeMappingValues=userPrincipalName, cn, mail, telephoneNumber,
givenName, sn
```

If you prefer to authenticate with only a user name, the sAMAccountName attribute may work as a substitute for userPrincipalName. The sAMAccountName attribute is supported for backward compatibility with older Windows versions. The revised property definition would be:

```
ldap.attributeMappingValues=sAMAccountName, cn, mail, telephoneNumber,
givenName, sn
```

❖ Configuring the authentication delegate in EAServer

1. In Jaguar Manager, expand these successive folders: Servers, <*Server Name*>, Installed Applications, sybepsecurity, Packages.
2. In the com.sybase.ep.security.sessionsvcs package, highlight the PortalSession component, and select File | Component Properties.
3. In the Component Properties dialog box, select the EJB Local Refs tab.
4. Highlight the "ejb/AuthenticationDelegate" EJB local reference, and set the Link Value to the JNDI component name of the authentication delegate; for example, "com.sybase.ep.security.authdelegate/LDAPDelegate".

❖ Configuring the authentication delegate in WebLogic

1. Using a text editor, open the *ejb-jar.xml* file, located in the *sybepsecurity/sybepsecurity.ear.exploded/com.sybase.ep.security.sessionsvcs.jar/META-INF* subdirectory of your WebLogic installation.
2. Below the line `<!-- authentication delegate -->`, set the value of `<ejb-link>` to "LDAPDelegate."

   If the value of `<ejb-link>` is set to "ACDBDelegate" the ACDB is used as the data store, instead of LDAP.

Mapping the LDAP entries to the ACDB

Enterprise Security provides three mapping implementations that transform data from Enterprise Security to LDAP and back, during the authentication process. Each mapping implementation performs a distinct function in the LDAP authentication process. The implementations are:

- Attribute mapping – determines how to populate the subject's ACDB record based on the data in the LDAP data store.
- Certificate mapping – takes certificate data presented by a client and determines the DN of the LDAP entry that corresponds to that certificate data.
- Organization mapping – determines the organization in which the replicated ACDB record should be placed, based on the LDAP entry.

This section describes these mapping implementations.

Attribute mapping

Attribute mapping defines how to populate subject records in the ACDB based on the data in the LDAP data store. To map attributes between LDAP and the ACDB, set the following properties in the *security.properties* file:

- ldap.attributeMappingNames – subject attributes in the ACDB that are populated when authenticating an LDAP user.
- ldap.attributeMappingValues – LDAP attributes that are stored in the ACDB during authentication.

An attribute mapping must define both properties, each property must contain the same number of attributes in a comma-separated list, and the corresponding attributes must be in the same positional order in both properties. The following attribute mapping is the default definition in *security.properties*:

```
ldap.attributeMappingNames=uid, cn, email, telephoneNumber, firstName, lastName
ldap.attributeMappingValues=uid, cn, mail, telephoneNumber, givenName, sn
```

ldap.attributeMappingNames

The ldap.attributeMappingNames property defines the attributes that are mapped to the subject table columns in the ACDB. The uid and cn attributes are required; other attributes are optional. Table 10-1 describes the attribute names, descriptions, and the SubjectManagement interface methods that return the attribute values from the ACDB.

Table 10-1: ldap.attributeMappingNames attributes

Attribute name	Description	SubjectManagement interface method
uid	The login name of the user.	getUid
cn	The user's common name.	getName
email	The user's e-mail address.	getEmail
telephoneNumber	The user's telephone number.	getPhone
firstName	The user's first name.	getFirstName
lastName	The user's last name.	getLastName
anyAttribute	A user-defined attribute name. Multiple user-defined attributes can exist, and for each, there must be a corresponding attribute in ldap.attributeMappingValues.	getExtraInfo This method returns a Map object that contains the values of all the user-defined attributes.

For more information about the SubjectManagement interface methods, use a Web browser to open *docs/html/index.html* in your Enterprise Security installation.

ldap.attributeMappingValues

The ldap.attributeMappingValues property defines the LDAP attributes that are retrieved from the LDAP data store and saved in the user's subject record during authentication. The position of the attributes in the mapping determines where the LDAP information is stored; for example, in the default definition, the sixth attribute of ldap.attributeMappingValues, sn, maps to the sixth attribute of ldap.attributeMappingNames, lastName. During authentication, the value of the LDAP attribute sn (surname) is stored in the ACDB subject table column that corresponds to lastName.

To use user name/password-based authentication, the LDAP user ID attribute, typically uid, must be a unique identifier that users present when logging in. The iPlanet Directory Server, Netscape Directory Server, and all other servers that support the inetOrgPerson object class, use the attribute name uid. For servers that do not support the inetOrgPerson object class, this name may be different; for example, the Microsoft Active Directory attribute name is userPrincipalName.

Certificate mapping

Certificate mapping is used to determine which LDAP record a user's certificate belongs to, and is used only when authenticating with certificates into an LDAP-enabled Enterprise Security installation. Enterprise Security ships two certificate mappers:

- Default – returns the certificate DN as the LDAP DN. This is useful if your certificate DNs match your LDAP DNs.
- AttributeMapper – allows various transformations to be done to search for a user's LDAP entry based on the certificate DN.

This section describes a published API that you can use to map certificate credentials to an LDAP record at runtime.

A certificate mapper must be a Java class that extends com.sybase.ep.security.authdelegate.ldap.CertificateMapper, and it must have a public constructor that takes one java.util.Properties parameter, which represents all of the properties stored in the *security.properties* file. When you develop new certificate mappers, you should create your own prefix (com.***clientName***.***propertyName***) tree within this file, which is passed into your mapper. You can override the following functions to implement new mapping functionality.

- Method 1:

```
public String getMappedDN(netscape.ldap.LDAPConnection conn,
       java.security.cert.X509Certificate certificate
throws com.sybase.ep.security.exceptions.AuthenticationFailedException;
```

- Method 2:

```
public String getMappedDN(netscape.ldap.LDAPConnection conn,
       String DN)
throws com.sybase.ep.security.exceptions.AuthenticationFailedException;
```

Both methods require an active connection to the LDAP server as the first parameter. The certificate mapper should release this connection after the method returns.

Method 1 requires an X.509 certificate instance as its second parameter. Using this class, the client can modify the certificate and independently verify the certificate's validity. This method is called in most circumstances.

Method 2 requires a DN that was extracted from the certificate at an earlier point in the authentication process. This method is called only if the full certificate instance is not available.

Both methods return the LDAP DN of the user who presented the certificate credentials. If an appropriate record cannot be found, an AuthenticationFailedException is thrown, and the user is denied access.

To configure a certificate mapper, specify a value for the ldap.certificateMapper property. The value you specify must be the Java class name of the certificate mapper you want to use. If you do not specify a value, the default mapper implementation value is used. The class you specify must be installed in a location identified by the EAServer BOOTCLASSPATH, typically *$JAGUAR/java/classes* on UNIX and Linux, or *%JAGUAR%\java\classes* on Windows.

Default mapper implementation

The default certificate mapper returns the DN extracted from the certificate as the LDAP DN.

Attribute mapper implementation

The attribute mapper is designed to extract a specific attribute from a certificate DN and search for the attribute in the LDAP data store to determine the LDAP record of the user to whom the certificate belongs. Any attribute and attribute value combined with the chosen search filter must return only one record on the LDAP server. If the query returns more than one value, the authentication request is denied. The attribute mapper returns the DN of the record returned by the LDAP query to the LDAP authentication delegate. The attribute mapper is implemented by the com.sybase.ep.security.authdelegate.ldap.AttributeMapper class.

To use the Attribute mapper, you must define the following properties:

- ldap.AttributeMapper.certificateAttributes – a comma-delimited list of attribute names that can be searched for in the supplied certificate. The attributes are extracted from the certificate DN. For example, given the following certificate DN:

  ```
  CN=John Doe,EMAIL=jdoe@org.com,O=ORGANIZATION
  ```

  These attributes can be extracted: CN, EMAIL, O. An extracted attribute value can be searched for within the LDAP server; both CN and EMAIL might be enough to uniquely identify the user.

- ldap.AttributeMapper.directoryAttributes – a comma-delimited list of the LDAP attributes that correspond to the ldap.AttributeMapper.certificateAttributes. Given the above sample, where the EMAIL attribute is extracted, the directory attribute would likely be "mail." The complete working sample properties for the above certificate is:

```
ldap.AttributeMapper.certificateAttributes=EMAIL, CN
ldap.AttributeMapper.directoryAttributes=mail, cn
```

When the user presents the above certificate to Enterprise Portal, it searches for a user in the LDAP server whose "mail" attribute is "jdoe@org.com." If no user is found with this attribute, then it tries to match based upon the CN attribute.

Organization mapping

Organization mappers are used to determine in which EP organization to place a user's record when initially replicating the user from LDAP to the ACDB. Enterprise Security ships two organization mappers:

- Default – places all subjects into the root organization.
- Pattern organization mapper – allows the specification of destination organizations by scanning through a list of regular expressions to see if the user's LDAP DN matches an appropriate expression.

An organization mapper must be a Java class that:

- Extends com.sybase.ep.security.authdelegate.ldap.OrganizationMapper
- Has a public constructor that accepts an instance of java.util.Properties as its only parameter
- Implements this method:

```
public String getOrganization(netscape.ldap.LDAPEntry entry);
```

where *entry* is the full LDAPEntry of the user who was authenticated, and the return value is an organization DN.

You can specify the organization mapper using the mapping properties in the *security.properties* file. See Table 15-4 on page 238. The default organization mapper places all users in the root organization.

To integrate completely with your LDAP server, you may need to specify additional configuration parameters. All of the LDAP configuration parameters are listed in Table 15-4 on page 238.

Using a remote authentication delegate

Enterprise Security 6.0 provides an authentication delegate implementation that acts as a bridge between the AuthenticationDelegate and RemoteAuthenticationDelegate interfaces.

❖ Configuring a remote authentication delegate in EAServer

1. In Jaguar Manager, expand these successive folders: Servers, <*Server Name*>, Installed Applications, sybepsecurity, Packages.
2. In the com.sybase.ep.security.sessionsvcs package, highlight the PortalSession component, and select File | Component Properties.
3. In the Component Properties dialog box, select the EJB Local Refs tab.
4. Highlight the “ejb/AuthenticationDelegate” EJB local reference, and set the Link Value to:

```
com.sybase.ep.security.authdelegate/RemoteProxyDelegate
```

5. Under Installed Applications, select “authdelegate.” Highlight the RemoteProxyDelegate component, and select File | Component Properties.
6. In the Component Properties dialog box, select the EJB Local Refs tab.
7. Highlight the “ejb/AuthenticationDelegate” EJB local reference, and set the Link Value to the name of a deployed bean that implements the RemoteAuthenticationDelegate interface.

For more information, access the Javadocs in your Enterprise Security installation. Using a Web browser, open */html/docs/index.html*, then select the com.sybase.ep.security.authdelegate package.

❖ Configuring a remote authentication delegate in WebLogic

1. Using a text editor, open the *ejb-jar.xml* file, located in the *sybepsecurity/sybepsecurity.ear.exploded/ com.sybase.ep.security.sessionsvcs.jar/META-INF* subdirectory of your WebLogic installation.

2. Below the line `<!-- authentication delegate -->`, set the value of `<ejb-link>` to “RemoteProxyDelegate.”

3. Using a text editor, open the *weblogic-ejb-jar.xml* file, located in the *sybepsecurity/sybepsecurity.ear.exploded/ com.sybase.ep.security.authdelegate.jar/META-INF* subdirectory of your WebLogic installation.

4. Find the string “custom.authentication.delegate” and replace it with the JNDI name of your custom authentication delegate.

CHAPTER 11

Configuring the Web Server Plug-in

Enterprise Security provides a Web server plug-in that protects assets—typically, URLs—from unauthorized access. This chapter describes how to set up the Web-server security plug-in, and how to configure the plug-in to use a secure listener.

Limitations Enterprise Security installations for EAServer on Linux, and Enterprise Security installations for WebLogic do not include the plug-in described in this chapter.

Overview

The Web server plug-in is a shared library module that is loaded into the Web server to intercept HTTP requests to perform user authentication and access authorization as needed.

The URL asset

The Web server plug-in is part of the Enterprise Security framework that secures assets stored in the ACDB.

The PSO enters documents to be secured (protected) by the plug-in into the ACDB as a URL asset; that is, the asset type is expressed as:

AssetTypeCN=*URL*

URL should point to the location of the document. Documents not entered into the ACDB can be accessed anonymously.

Processing incoming HTTP requests

The Web server plug-in intercepts HTTP requests after it finishes the URL translation phase. The Web server obtains the translated URL and checks it against the ACDB to determine whether it is protected. If the URL that the user specifies in the Web browser differs from the translated URL, the PSO must enter the translated URL into the ACDB. If the plug-in does not find the URL in the ACDB, it keeps this URL in a cache of unprotected URLs and returns control to the Web server so that the server can process the request as usual. If it does find the URL in the ACDB, it authenticates the user as necessary—see "User authentication" on page 182—and authorizes access based on the user's permissions.

The Web plug-in does not secure contents in a document. If a document contains hypertext links that a user does not have permission to access, the plug-in does not hide those links. The content management system should provide that functionality. If your environment does not have such a system, be aware of this behavior and name the links with caution.

User authentication

Note The only authentication methods currently supported are name and password credentials, and digital certificates.

A user can be authenticated by explicitly logging in to the system via a login page, or the user can browse a Web site without logging in, as long as they access only unprotected URLs.

When the plug-in intercepts an HTTP request and determines that the URL is protected, it follows this procedure to authenticate the user:

1. The plug-in tries to find a cookie (generated by the plug-in during authentication) in the HTTP request.
2. If there is no cookie, the plug-in looks for the client's certificate in the HTTP request.

- If the certificate is found, the plug-in attempts to authenticate the user.
- If the certificate is not found, the plug-in redirects the user to the login page to enter his or her user name and password.

3 The plug-in attempts to authenticate the user via the PortalSession EJB.

4 If authentication is successful, the security plug-in sets the session password in a PSPK_TEMPORARY cookie that is sent back to the Web browser. Any back-end application that manages security objects on behalf of the authenticated user must obtain this session password from the cookie. The security plug-in sets the PSPK_TEMPORARY cookie as follows:

   - For HTTP requests, the cookie is set as a regular (insecure) cookie, which is passed freely between the client and the Web server.
   - For HTTPS requests, the cookie is set as a secure cookie. As long as the client continues to use HTTPS, the cookie is transmitted between the client and the Web server. If the client requests a URL using HTTP, the cookie is not sent to the Web server.

   If the user cannot be authenticated, the plug-in ends the HTTP request processing and redirects the user to an error page indicating the failure.

5 Once authenticated, the user does not need to be reauthenticated to access protected URLs, as long as the session is valid; that is, the session has not timed out, and the user has not explicitly logged out of the system.

   If the user session terminates or becomes invalid, the plug-in removes the session from the internal cache and the user must log in to the system again to access protected URLs.

Access authorization

Once the Web server plug-in authenticates that a valid user is requesting access to a protected URL, it checks whether the user can access the URL based on the permissions that are assigned to the user’s roles.

- If the user is allowed access to the URL, the plug-in returns control to the Web server so the server can process the HTTP request as usual.
- If the user does not have permission to access the URL, the plug-in ends the HTTP request processing and redirects the user to an error page indicating access failure.

To perform access authorization, the plug-in must know exactly what to look for in the ACDB. If a given URL */A/B/C/D* is not found in the ACDB, the plug-in searches in this order:

- /A/B/C
- /A/B
- /A

When the URL is found, the plug-in checks access against the matching asset.

If a user does not have permission to access */A*, but does have permission to access */A/index.html*, the user must explicitly request */A/index.html* to gain access to that page.

For example, let's say you have a protected directory */ws_doc_root/DirA* with these files in the directory: *index.html*, *file1*, *file2*, *file3*. You want to lock *file1*, *file2*, and *file3* so that John cannot read them, but you want to give John permission to access *index.html*. Using Enterprise Security Manager, add the following URL assets:

```
/ws_doc_root/DirA
/ws_doc_root/DirA/index.html
```

Then, assign READ permission to access */ws_doc_root/DirA/index.html* to a role that John has. John can now access */ws_doc_root/DirA/index.html*. If John tries to access */ws_doc_root/DirA/file1*, */ws_doc_root/DirA/file2*, or */ws_doc_root/DirA/file3*, an access forbidden message displays in the browser.

If John tries to access */ws_doc_root/DirA* or */ws_doc_root/DirA/*, the plug-in determines that John does not have permission to access the directory and ends the request processing.

The security plug-in ignores the query string in a URL when performing authorization checks. For example, to determine whether `/index.html?arg=value` is secure, the plug-in checks the ACDB for `/index.html`.

Plug-in caches

Asset cache

The plug-in maintains an asset cache that keeps track of the unprotected URLs that have been accessed by all sessions. The asset cache is the first thing the plug-in checks when processing an HTTP request. If it finds the requested URL in the asset cache, the plug-in knows that the URL is not protected and does not perform the User authentication and Access authorization steps. This reduces the number of times the plug-in has to access the security EJBs that may be running on a remote host.

The asset cache uses a cache replacement policy. Whenever the cache is full and a new asset needs to be cached, the least recently used entry is removed from the cache. If a particular Web server handles many unprotected URLs, the plug-in should have a larger asset cache than a server that handles mostly protected URLs. This cache is synchronized with the ACDB through a TCP/IP connection between the plug-in and the Enterprise Security middleware. ACDB synchronization occurs whenever a URL asset changes in the ACDB.

The number of asset cache entries is configured via the UNPROTECTED_URL_CACHE_MAX_SIZE property in the plug-in configuration file *SybSecurityPluginConfig.txt*—see "Configuring the SybSecurityPluginConfig.txt file" on page 188.

Cache manager

The Web server plug-in's URL cache is the first level of caching. There is also a second-level, larger cache managed by the CacheMgr bean. The purpose of the second cache is to reduce the number of times the plug-in must access the ACDB. The *CacheManager.props* file is located in the *$JAGUAR/Repository/Component/com.sybase.ep.security.cachemgr* directory.

The assetCacheSize property determines the size of the cache, and the assetCacheFlushPercentage property determines how much of the cache is flushed when it if full. These properties are stored in the *security.properties* file.

Logging statistics

To help the Web server administrator properly configure the asset cache and connection cache size, the plug-in writes events to a log file that you specify in the plug-in configuration file. You can specify these properties in the configuration file:

- LOG_UNPROTECTED_URL_CACHE_ACTIVITY – if value is true, the plug-in logs the asset cache activities.
- LOG_CONNECTION_CACHE_ACTIVITY – if value is true, the plug-in logs the connection cache activities.

The session handle

The PortalSession plug-in now passes the PortalSession primary key through the HTTP header, *sybsession_pk*. All new applications should use this primary key to look up the PortalSession.

The method used to get the information depends on the application’s environment. For example, in a CGI script, the header is exported by Apache as $HTTP_SYBSESSION_PK.

If a chain of applications requires the primary key, the primary key must be passed from one application to another.

Installing a Web server security plug-in

Sybase does not provide the Web server. You must purchase and install the Web server separately.

You must install a plug-in on all machines with Web servers that run inside the e-business system.

Enterprise Security provides the following Web server plug-ins:

- Apache Web server plug-in for Solaris
- Netscape Enterprise Server plug-in for Solaris and Windows
- Sun ONE (iPlanet) Web server plug-in for Solaris and Windows
- Microsoft Internet Information Server (IIS) for Windows

Installing and configuring Apache on Solaris

Pre-installation tasks for the "root" user

If you plan to use the Apache Web server with the privileged HTTP and HTTPS ports, 80 and 443 respectively, the Apache Web server must run as the root user.

You can download a copy of the Apache Web server from the Apache Web page at http://httpd.apache.org.

❖ Configuring the Apache Web server to run as root

1 Change the ownership of the *httpd* (Web server executable) and *apachectl* (the command to start and stop the server) to the root user.

2 Set the setuid file protection bit.

3 Change to the *Apache-1_3/httpd/bin* directory, and grant execute permissions on these files to the group that the "sybase" user is in:

```
chown root apachectl httpd
chmod 4750 apachectl httpd
```

Now when you start the server, the setuid permission bit causes the Web server to run as root. Once Apache has successfully connected to the HTTP and HTTPS ports, it internally changes the value of userid to "nobody," reducing security risks.

The UNIX dynamic loader (*ld.so.1*) does not search the LD_LIBRARY_PATH when a process is running as root. Therefore, to use the Security plug-in for Apache (*mod_sybepsecure.so*), copy the EAServer client library (*libjcc.so*) to */usr/lib*:

```
cp $JAGUAR/client/lib/libjcc.so  /usr/lib
```

❖ Setting up the Apache Web server plug-in

Verify that you have installed the Web server components option before you complete this procedure.

1 Verify that the LD_LIBRARY_PATH environment settings contain:

- *JAGUAR/lib*
- *libjcc.so*
- The primary shared library, which should be in a folder within the plug-in's installation directory

2 Set JAGUAR_CLIENT_ROOT to *$JAGUAR*.

3 Set up the Apache Web server to load the plug-in:

a. Go to *$SYBASE/Security/lib/*.

b. Unjar *plugins.jar*.

c. Unjar *Apache.jar*.

d. Copy all *.html* files to the Apache Web server's document root directory, *htdocs*.

e. Copy **.so* to the *libexec* directory under the Apache installation.

f. Copy the sample plug-in configuration file *SybSecurityPluginConfig.txt* to $SYBASE, and edit it using the instructions in "Configuring the SybSecurityPluginConfig.txt file" on page 188.

g. Edit the Apache configuration file *httpd.conf*, and following this line:

```
#LoadModule foo_module libexec/mod_foo.so
```

Add the following lines:

```
LoadModule epsecure_module libexec/mod_sybepsecure.so
AddType sec-login .com_sybase_ep_seclogin
AddHandler epcontent-handler .com_sybase_ep_seclogin
```

4. Verify that EAServer and the security EJBs are running before you start Apache.

5. Start the Apache server. Change to the *http* directory of the Apache installation and enter:

```
bin/apachectl start
```

6. Configure other Web server plug-in properties in *SybSecurityPluginConfig.txt*, as necessary—see Table 11-1 on page 189.

Configuring the *SybSecurityPluginConfig.txt* file

To use the Web server plug-in, you must configure some of the properties that are defined the *SybSecurityPluginConfig.txt* file. At initialization time, the plug-in reads the parameters defined in Table 11-1 on page 189 from the *SybSecurityPluginConfig.txt* configuration file.

Before you start the Web server, set the following environment variables:

- JAGUAR_SERVER_IIOP_URL – the URL that represents the location of EAServer. Change the IIOP host to the host where EAServer is installed, and change the port to your EAServer port number.
- MISC_INFO_FILE – must point to the *default_credential.txt* file.

The password for the EPWebServerPlugin user is defined in *default_credential.txt* and used for credential checks. If you change the password in this file, you must also change the password using Enterprise Security Manager—see “Viewing and updating a user account” on page 34.

- Set these environment variables to the following values:

```
SYB_EP_WEB_VALID_LOGIN_PAGE = /syb_ep_web_valid_login.html
SYB_EP_WEB_INVALID_LOGIN_PAGE = /syb_ep_web_invalid_login.html
SYB_EP_WEB_LOGIN_PAGE = /syb_ep_weblogin.html
SYB_EP_WEB_LOGOUT_PAGE = /syb_ep_weblogout.html
SYB_EP_WEB_PLUGIN_ERROR_PAGE = /syb_ep_web_plugin_error.html
```

- SECURITY_PLUGIN_LOG_FILE_PATH – set to the full path of the *plugin.log* file.
- WEBSERVER_IP_ADDRESS – set to the IP address of the Web server.

Table 11-1: Security plug-in configuration properties

Property	Value

APACHE_PORT_AVAIL[*num*]	The number of each port available for the Apache servers. The number of entries for this parameter should match the value entered for MAX_APACHE_PORTS.
	For example, if MAX_APACHE_PORTS=5, there should be five entries in the plug-in configuration file to specify five different ports; for example:
	`APACHE_PORT_AVAIL1=3001`
	`APACHE_PORT_AVAIL2=3002`
	`APACHE_PORT_AVAIL3=3003`
	`APACHE_PORT_AVAIL4=3004`
	`APACHE_PORT_AVAIL5=3005`

CONN_CACHE_MAX_ENTRY_TIME	This property specifies how long (in seconds) an idle session can stay in the connection cache. If a session is idle for this amount of time, the session is flushed out of the connection cache. Typically, this value should be equal to the PortalSession bean timeout value. See the PortalSession bean property com.sybase.jaguar.component.timeout.
DEBUG	Set to true to enable debugging.
IPLANET_BIND_PORT	Specifies the port number the Web server could use to open a TCP/IP socket to communicate with EAServer. This TCP/IP socket is opened using the INADDR_ANY system macro (effective for all network cards for which the Web server is configured) in the Internet domain.

Property	Value
JAGUAR_SERVER_IIOP_URL	The URL that represents the location of EAServer.
LOG_UNPROTECTED_URL_CACHE_ACTIVITY	True or false. If true, an event is logged whenever a session is cached or removed. The number of entries in the cache at the moment is logged. If a user cannot connect due to a full connection cache, the failure is logged. Other information is logged as necessary. The default is “false.”
MAX_APACHE_PORTS	The number of ports available for the Apache servers. The Apache server spawns multiple child servers at runtime based on the server load. There should be more ports available if several child servers are spawned. Each child server binds to one port at start time. **Note** MAX_APACHE_PORTS must be equal to the value of MaxClients in the Apache configuration file, *httpd.conf*, which is included in your Apache installation *httpd/conf* directory. For example, if you increase the value of MaxClients to 250, then you must also set this value to 250, and you must create 250 entries for the APACHE_PORT_AVAIL[*num*] property.
SECURITY_PLUGIN_LOG_FILE	The location of the log file where statistics are recorded.
SET_COOKIE_MAX_AGE	True or false. If this parameter is set to true, the plug-in sets the Max_Age of the cookie to the value specified by SYB_SESSION_COOKIE_AGE. If this parameter is set to false, the plug-in does not set the Cookie Max_Age value. This leads to the cookie being nonpersistent in the client browser (recommended).
SYB_EP_WEB_LOGIN_PAGE	The login page URL relative to the Web server’s document root location.
SYB_EP_WEB_LOGOUT_PAGE	The logout page URL relative to the Web server’s document root location. The plug-in redirects clients to this page when either: • An authenticated session expires, or • The client’s browser presents a digital certificate for reauthentication. A subsequent attempt to access a protected URL reauthenticates the user.
SYB_QOP	Specifies the QOP that the plug-in uses. The only QOP currently supported is “sybpks_intl,” which is set as the default value.
SYB_SESSION_COOKIE_AGE	If SET_COOKIE_MAX_AGE is set to true, the plug-in sets the Max_Age value as specified. Value in seconds.

Property	Value
SYB_SESSION_COOKIE_DOMAIN	If “NOT_SPECIFIED”, the plug-in does not set the domain value, or else the domain has to be specified beginning with a period (for example, .sybase.com).
SYB_SESSION_COOKIE_NAME	Specifies the name of the cookie as it will be seen in the client browser if the appropriate browser options are enabled. After the plug-in authenticates the user, it creates an HTTP cookie and sends it back to the Web browser.
SYB_SESSION_COOKIE_PATH	Specifies the cookie path. Recommended value is “/”.
UNPROTECTED_URL_CACHE_FLUSH_SIZE	The number of entries to flush out of the URL cache when the cache is full.
UNPROTECTED_URL_CACHE_MAX_SIZE	The number of entries in the unprotected URL cache.
WEBSERVER_IP_ADDRESS	The IP address EAServer can use to open a TCP/IP client socket to the Web server. The security EJB opens a TCP/IP socket to this server and sends messages to instruct the plug-in to flush the unprotected URL cache when a URL asset is created, updated, or deleted in the ACDB.

Installing and configuring Netscape Enterprise Server and Sun ONE

Enterprise Security includes Web server plug-ins for the Netscape Enterprise Server (NES) version 6.1 and the Sun ONE (iPlanet) Web server version 6.1.

Enterprise Security does not include these Web servers. To use the Web server plug-ins, you must first obtain the server software and install it according to the Web server instructions.

❖ Setting up the NES or Sun ONE Web server plug-in on Solaris

1. After installing the Web server, create a *seclib* directory under *$NETSCAPE/https_servername*, where *$NETSCAPE* is the Web server installation directory, and *servername* is the name of your machine.
2. Create a JAGUAR_CLIENT_ROOT variable that points to *$JAGUAR*.
3. Create an LD_LIBRARY_PATH variable that includes the following paths:
    - *$JAGUAR/lib*
    - *$JAGUAR_CLIENT_ROOT*
    - *$NETSCAPE/https_servername/seclib*

4. Set up the Web server to load the plug-in:

   a. Change to *$SECURITY/lib*.

   b. Unjar *plugins.jar*.

   c. For Sun ONE, unjar *iPlanet.jar*; for NES, unjar *Netscape.jar*.

   d. Copy all *.html* files from the Netscape folder to *$NETSCAPE/docs*.

   e. Copy *libsybepsecure_ip.so* and *libcorba_jaguar_combat.so* to *$NETSCAPE/https_servername/seclib*, where *servername* is the name of the machine.

   f. Copy the sample plug-in configuration file *SybSecurityPluginConfig.txt* to *$NETSCAPE/https_servername*.

   g. Create a directory *unicode/sec_web/english* under *$SYBASE/locales*.

   h. If the $SYBASE variable is not defined, create it and point it to $JAGUAR.

5. Copy *plugins.lcu* to *$SYBASE/locales/unicode/sec_web/english*.

6. Open *SybSecurityPluginConfig.txt* located in *$NETSCAPE/https_servername*, and edit the file according the instructions in “Configuring the SybSecurityPluginConfig.txt file” on page 188.

7. Go to *$NETSCAPE/https_servername/config*, open the *magnus.conf* file in any text editor, and insert the following three lines after the last “Init” directive. Each “Init” directive must be written on a single line.

```
Init fn=load-modules
shlib=$NETSCAPE/https_servername/seclib/libsybepsecure_ip.so
funcs=”initialize_plugin,sec_path_check,sec_login"

Init fn=”initialize_plugin”
security_conf_file_path=$NETSCAPE/https_servername/
SybSecurityPluginConfig.txt

LateInit=yes
```

Where *$NETSCAPE* is the location of the Web server installation. The first directive loads the security plug-in in the Web server process space, and the second directive initializes the plug-in.

8. In the same directory, open the *obj.conf* file, and before this line:

```
AddLog fn=flex-log name=”access”
```

Add the service directive:

```
Service fn="sec_login" method="(GET|POST)"
type="magnus-internal/sec-login"
```

9 Define a new object type in *obj.conf* as the last object directive in the file:

```
<Object path="/*">
PathCheck fn="sec_path_check"
</Object>
```

10 For Sun ONE, add the following line to *obj.conf*:

```
NameTrans fn=”assign-name” from=”/onepage*” name=”EASProxy”
```

11 Allow digital certificate authentication:

a Before the following line in *obj.conf*:

```
PathCeck fn=“sec_path_check”
```

Add this line:

```
PathCheck fn=“get-client-cert” dorequest=“1”
```

b Enable Web server SSL—see your Web server documentation for information about how to do this.

12 Using any text editor, edit *mime.types* (found in the same Netscape directory as *obj.conf*). At the end of the file, insert the proprietary MIME type definition:

```
type=magnus-internal/sec-login exts=seclogin
```

You can insert this directive in any order relative to other existing MIME type definitions.

13 Start your Web server by running the start script in *$NETSCAPE/https_servername*.

If you have problems starting the Web server, set the owner of the start and stop scripts in the *$NETSCAPE/https_servername* directory to “root,” and change the permissions:

```
chown root $NETSCAPE/https_servername/start
chmod 4750 $NETSCAPE/https_servername/start
chown root $NETSCAPE/https_servername/stop
chmod 4750 $NETSCAPE/https_servername/stop
```

❖ Setting up the NES or Sun ONE Web server plug-in on Windows

1. After installing the Web server, create a *seclib* directory under *%NETSCAPE%\https_servername*, where *%NETSCAPE%* is the Web server installation directory, and *servername* is the name of your machine.

2. Verify that the PATH environment variable points to:

   - *%JAGUAR%\dll*
   - The primary shared library, *libsybepsecure_ip.dll*, which should be in *%NETSCAPE%\https_servername\seclib*.

3. Create a JAGUAR_CLIENT_ROOT variable that points to the EAServer installation directory.

4. Set up the Web server to load the plug-in:

   a. Change to *%SECURITY%\lib*.

   b. Unjar *plugins.jar*.

   c. For NES, unjar *Netscape.jar*; for Sun ONE, unjar *iPlanet.jar*.

   d. Copy all the *.html* files from the Netscape folder to *%NETSCAPE%\docs*.

   e. Copy *libsybepsecure.dll* to *%NETSCAPE%\https_servername\seclib*, where *servername* is the name of the machine.

   f. Copy the sample plug-in configuration file *SybSecurityPluginConfig.txt* to *%NETSCAPE%\https_servername.*

   g. Create a directory *unicode\sec_web\english* under *%SYBASE%\locales*.

      If the SYBASE variable is not defined, create it and point it to *%JAGUAR%*. You must restart your machine for the setting to take effect.

5. Copy *plugins.lcu* to *%SYBASE%\locales\unicode\sec_web\english*.

6. Edit the *SybSecurityPluginConfig.txt* file located in *%NETSCAPE%\https_servername*, and follow the instructions to update the environment variables described in “Configuring the SybSecurityPluginConfig.txt file” on page 188.

   Note When you edit these variables, use “/” instead of “\”.

7 Go to *%NETSCAPE%\https_servername\config*, open the *magnus.conf* file in any text editor, and insert the following three lines after the last “Init” directive. Each “Init” directive must be written on a single line.

```
Init fn=load-modules
shlib=%NETSCAPE%/https_servername/seclib/libsybepsecure_ip.dll
funcs=”initialize_plugin,sec_path_check,sec_login"

Init fn=”initialize_plugin”
security_conf_file_path=%NETSCAPE%/https_servername/SybSecurityPluginCo
nfig.txt

LateInit=yes
```

Where*%NETSCAPE%* is the location of the Web server installation. The first directive loads the security plug-in in the Web server process space, and the second directive initializes the plug-in.

8 In the same directory, open the *obj.conf* file, and before this line:

```
AddLog fn=flex-log name=“access”
```

Add the service directive:

```
Service fn="sec_login" method="(GET|POST)"
type="magnus-internal/sec-login"
```

9 Define a new object type in *obj.conf* as the last object directive in the file:

```
<Object path="\*">
PathCheck fn="sec_path_check"
</Object>
```

10 Allow digital certificate authentication:

a Before the following line in *obj.conf*:

```
PathCeck fn=“sec_path_check”
```

Add this line:

```
PathCheck fn=“get-client-cert” dorequest=“1”
```

b Enable Web server SSL—see the Netscape Enterprise Server documentation for information about how to do this.

11 In the same directory, open the *mime.types* file, and at the end of the file, insert the proprietary MIME type definition:

```
type=magnus-internal/sec-login exts=seclogin
```

You can insert this directive in any order relative to other existing MIME type definitions.

12. Restart your machine.
13. Start the NES or Sun ONE Web server from Start | Control Panel | Services. Locate Netscape Enterprise Server *hostname*, or Sun ONE *hostname*, and click Start.

Installing and configuring IIS 5.0 on Windows 2000

This section explains how to set up the IIS Web-server plug-in on Windows 2000.

Microsoft IIS is packaged with Windows 2000. You must install and configure IIS using the Microsoft instructions before you install and configure the Enterprise Security plug-in.

Note Make sure that the SYBASE environment variable points to the directory where you have installed Sybase Enterprise components.

This plug-in is supported on IIS 5.0 only.

❖ Configuring IIS to load the plug-in

1. Change to the *%SYBASE%\Security\lib* directory.
2. Unjar *plugins.jar.* You can use the WinZip utility.
3. Unjar *iis.jar*.
4. Copy all the *.html* files to the document root directory.

   When you install IIS 5.0, the installation process creates the default document root directory, *C:\\inetpub\wwwroot*.
5. Copy the *libsybepsecurity_iis.dll* file to the directory where you want the plug-in to reside.
6. Copy the sample plug-in configuration file, *SybSecurityPluginConfig.txt*, to *%SYBASE%*.
7. Copy the *plugins.lcu* file to *%SYBASE%\locales\unicode\sec_web\english*. The *plugins.lcu* file is created when you extract the *iis.jar* file. You may have to create this directory path if it does not already exist.

8 Edit the *SybSecurityPluginConfig.txt* file:

Note If you set the SYB_SESSION_COOKIE_DOMAIN variable as described below, you must add the domain information to all the SYB_EP_WEB_* variables and the JAGUAR_SERVER_IIOP_URL variable. For example:

```
SYB_EP_WEB_LOGIN_PAGE=http://myserver.mycompany.com
:80/syb_ep_weblogin.html
```

Includes the domain ".mycompany.com," whereas:

```
SYB_EP_WEB_LOGIN_PAGE=http://myserver:80/syb_ep_web
login.html
```

Does not include a domain name. If you do not include the domain name, be sure that the SYB_SESSION_COOKIE_DOMAIN variable remains set to the default of NOT_SPECIFIED.

a Add your *host:port* or IP Address:*port* to the JAGUAR_SERVER_IIOP_URL variable, where *host* is the name of the machine where EAServer is installed. For example:

```
JAGUAR_SERVER_IIOP_URL=iiop://host:9000
```

To retrieve the IP address for your Windows 2000 server, open a command prompt and type, `ipconfig`.

b For the following variables, replace all instances of "http_*servername*" and port numbers with the IIS 5.0 server name and port number.

When editing these variables, use the forward slash (/) instead of the back slash (\).

- SYB_EP_WEB_LOGIN_PAGE
- SYB_EP_WEB_INVALID_LOGIN_PAGE
- SYB_EP_WEB_VALID_LOGIN_PAGE
- SYB_EP_WEB_ERROR_PAGE

c Optionally, you can set the SYB_SESSION_COOKIE_DOMAIN variable to the domain name without the server name. For example:

```
.mycompany.com
```

Where *mycompany* is your company's domain name.

d Set the variable SECURITY_PLUGIN_LOG_FILE_PATH to the location of your log file. For example:

```
C:/inetpub/wwwroot/plugin.log
```

e Set the variable WEBSERVER_IP_ADDRESS to the IP address of the local server.

f Set the variable MISC_INFO_FILE to point to your *default_credential.txt* file that was created when you extracted the *iis.jar* file.

9 Configure IIS 5.0 to load *libsybepsecure_iis.dll* as an ISAPI filter. It must be installed as a global filter for all WWW sites. See the instructions in the Microsoft IIS 5.0 documentation. You must perform this step for each individual site you create in IIS.

Note *libsybepsecure_iis.dll* is high priority. Sybase recommends that you install it first in the list of ISAPI *.dll* files.

10 Configure application settings under the Home Directory tab under properties for the Web site (see the Microsoft IIS 5.0 documentation).

a Set the application name.

b Set Execute Permissions to “Scripts and Executables”.

c Set Application Protection to “low IIS process”.

d Click Configuration and:

1 Select the App Mappings tab, and click Add.

2 For the Executable: field, enter the full path to *<path>\libsybepsecure_iis.dll*.

3 For the Extension: field, enter `.com_sybase_ep_seclogin`.

4 Click the All Verbs button.

5 Select Script Engine and make sure that “check that file exists” is not selected.

6 Click OK.

Post-configuration tasks

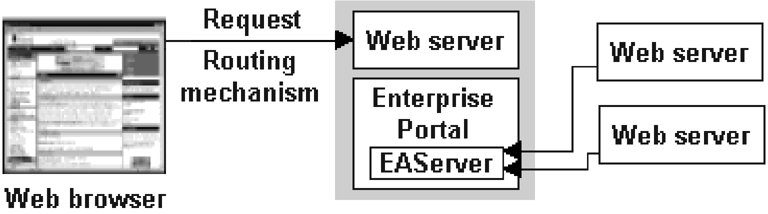
1 Verify that EAServer running the security EJBs is running before you start IIS 5.0.

2 Start IIS. If IIS 5.0 is already running, stop and restart it.

Using multiple Web servers

If you install Web servers on different machines, your setup may look something like the one shown in Figure 11-1.

Figure 11-1: Multiple Web server setup example



All of the Web servers should talk to the same server within EAServer and they should all have the same value for the JAGUAR_SERVER_IIOP_URL property in *SybSecurityPluginConfig.txt*.

Verifying that the security plug-in is active

Follow your Web server's instructions to set up and verify that both the HTTP and HTTPS listeners are active. Then, verify that the security Web server plug-in is active:

1 Open a Web browser.

2 Enter the URL; typically, this URL is:

```
http://Web_server_name:port_number/syb_ep_weblogin.html
```

The Web login page opens.

3 Log in using the default user name "pso" and password "123qwe".

4 You can create new user profiles using Enterprise Security Manager. To launch Enterprise Security Manager:

a Enter this URL in your browser; *host* and *domain* identify where EAServer is running, and *port* is the EAServer HTTP port number—8080, by default:

```
http://host.domain:port/onepage/loader.html
```

b In the Login window, enter your user name and password, and click Login. If you accepted the defaults during installation, the user name is "pso" and the password is "123qwe".

Using the IIOPS protocol

Enterprise Security allows a Web server plug-in to use IIOPS to establish a secure session with EAServer.

❖ Configuring the plug-in to use IIOPS

1 Verify that:

- The EAServer listener at the specified port is configured as an IIOPS listener.
- The security profile assigned to the listener has sybpks_intl set as the **quality of protection** (QOP).

See "Chapter 11, Security Configuration Tasks" in the *EAServer Security Administration and Programming Guide* for instructions.

2 In the *SybSecurityPluginConfig.txt* configuration file, set the JAGUAR_SERVER_IIOP_URL to use an IIOPS listener:

```
JAGUAR_SERVER_IIOP_URL=iiops://lowie:9003
```

This instructs the plug-in to connect to the machine named "lowie" using the IIOPS listener at port 9003.

Based on your environment, using IIOPS can sometimes diminish performance and may not be necessary. You can instruct the plug-in to use IIOP by setting JAGUAR_SERVER_IIOP_URL to an IIOP listener:

```
JAGUAR_SERVER_IIOP_URL=iiop://lowie:9000
```

The SYB_QOP property specifies the QOP that the plug-in uses. Currently, the only QOP supported is "sybpks_intl," which is set as the default value.

Encrypting passwords

To protect the credential information used by the Web server security plug-in, the credential information is moved out of the plug-in configuration file and placed in an encrypted file called *default_credential.txt*. The plug-in configuration file includes the MISC_INFO_FILE property, which points to the encrypted credential file; for example:

```
MISC_INFO_FILE=/work/default_credential.txt
```

The encrypted credential file contains:

```
JAGUARADMINNAME=jagadmin
JAGUARADMINPASSWORD =WEBPLUGIN
WEBPLUGINUSERNAME=EPWebServerPlugin
WEBPLUGINPASSWORD=sybase
SYBTOKENPASSWORD=sybase
```

The encrypted credential file is provided as part of the security administration component that is installed inside the firewall.

Enterprise Security provides a utility (webplugin_util) that allows you to encrypt and decrypt the credential file after changing the passwords. This utility is stored in the *bin* subdirectory of your Enterprise Security installation.

For Enterprise Portal customers, this utility was changed with the release of Enterprise Portal 2.5 (this has no effect on EAServer customers). If you encrypted your key file with an older version of the utility, you must use that same version of the utility to decrypt the file. You can then use the new version to reencrypt the file.

Warning! Because webplugin_util can be used to decrypt the credential file, you must store it on a trusted machine; that is, not on the machine where the Web server is running.

❖ **Running webplugin_util**

1. Copy the encrypted credential file from the Web server machine to the trusted machine where the utility resides.
2. At the UNIX or Windows command line, decrypt the credential file:

   ```
   webplugin_util -decrypt file_name
   ```

3. Edit the information in the credential file as necessary.
4. At the UNIX or Windows command line, reencrypt the credential file:

   ```
   webplugin_util -encrypt file_name
   ```

5 Copy the credential file back to the Web server machine and apply the appropriate permissions to the credential file at the operating system level.

The WEBPLUGINUSERNAME property specifies the subject UID the plug-in is running as. By default, WEBPLUGINUSERNAME is assigned the subject EPWebServerPlugin. To assign a subject other than EPWebServerPlugin to the plug-in, that subject must be granted the PortalWebPlugin role.

CHAPTER 12 Certificate-Based Authentication

This chapter describes how to configure certificate-based authentication into EAServer in a system using a Web server security plug-in and a redirector plug-in.

Limitations Enterprise Security installations for EAServer on Linux, and Enterprise Security installations for WebLogic do not include the plug-ins described in this chapter. However, you can enable certificate-based authentication for direct connections to either application server. For more information, see:

- EAServer – Certificate Authentication directly into Portal and Studio at http://www.sybase.com/detail_list/1,6902,48576,00.html.
- WebLogic – "Setting up WebLogic authentication" on page 149.

Configuring certificate-based authentication

To configure certificate-based authentication for systems that include a Web server security plug-in and a redirector plug-in, perform the following tasks, stopping after each step to make sure that things are working properly before proceeding.

1 Install and configure the Enterprise Security Web Server plug-in on your Web server. See Chapter 11, “Configuring the Web Server Plug-in.” You may want to use an insecure server that accepts client requests on port 80 before deploying this to your production environment.

Note To improve performance, Sybase strongly recommends that you install EAServer and the Web server on different machines. Typically, the portal generates numerous requests to the localhost HTTP listener, and if the Web server and EAServer are on the same machine, these requests are routed through the Web server, the security plug-in, and the redirector plug-in before EAServer receives them.

2 To use an LDAP server, instead of the ACDB:

a Configure Enterprise Security to use the LDAP authentication delegate—see Chapter 10, “Configuring LDAP Authentication.”

b Configure a certificate mapper to tell Enterprise Security how to find the LDAP user based on the certificate provided by the client—see “Certificate mapping” on page 176.

c Map the PortalWebPlugin role to a user or set of users in the LDAP server.

3 On the same server where the Enterprise Security Web Server plug-in is installed, install the EAServer redirector plug-in on Apache, iPlanet, or Netscape, and configure EAServer to accept Web server requests. For installation and configuration instructions, see Chapter 9, “Web Server Redirector Plug-In,” in the *EAServer System Administration Guide*. To access the HTML version, use a Web browser to open *html/docs/index.html* in your EAServer installation.

4 Set up the Web server to demand client certificates. See your Web server documentation for more information.

5 Edit *global.propeties.xml*, located in the *Repository/WebApplication/onepage/config* subdirectory of your EAServer installation, and set the values of the secure and secure_login properties to on. See the *Enterprise Portal Developer’s Guide* for more information about these properties.

6 Configure an EAServer HTTPS listener for accessing the portal—see “Configuring listeners” in Chapter 3, “Creating and Configuring Servers,” in the *EAServer System Administration Guide*. Set the listener’s security characteristic to a value that does not include the string “mutual_auth.”

7 Configure the redirector to support HTTPS—see “Configuring the redirector to support HTTPS connections” on page 207.

8 Configure the server’s HTTP configuration properties—see “Configuring EAServer HTTP properties” on page 207.

9 Create one or more URL assets that represent protected Web pages to force the Web server security plug-in to create a PortalSession object when a user attempts to log in to the portal—see “Creating an asset” on page 47.

❖ Configuring the redirector to support HTTPS connections

1 Edit the *conn_config* file in your redirector plug-in installation, and set the following properties:

- `Connector.WebApp /onepage=https://<`***`host`***`>:<`***`port`***`>/`

  Where *host* is the EAServer host and *port* is where EAServer accepts HTTPS requests.

- `Connector.Https.qop sybpks_simple`
- `Connector.Https.pin` ***`security_PIN`***

  Where *security_PIN* is the PIN used to connect to Jaguar Security Manager; the default is “sybase.”

2 Set up an HTTPS listener in EAServer for accessing the portal.

❖ Configuring EAServer HTTP properties

If you have multiple HTTP or HTTPS listeners, you must tell EAServer where to redirect requests.

1 In Jaguar Manager, expand the Servers folder, highlight the server you are using (typically, Jaguar), and select File | Server Properties.

2 Select the HTTP Config tab, and enter:

- Domain Name – enter the host and domain name of the Web server; for example, `abc.sybase.com`.

- Proxy HTTPS Port – enter the port number where the Web server accepts HTTPS requests.

  Note The Proxy HTTPS Port and Proxy HTTP Port numbers must match the values of the default_https_port and default_http_port properties in *global.properties.xml*, whose location depends on your application server:

  - EAServer – *JAGUAR/Repository/WebApplications/onepage/config*.
  - WebLogic – *BEA_ROOT/onepage/config*.

- Proxy HTTP Port – enter the port number where the Web server accepts HTTP requests.
- Set Proxy Protocol to HTTPS.

3 Click OK to save your changes.

CHAPTER 13 Using Proxy Servers

This chapter describes how to set up Enterprise Portal to use a proxy server.

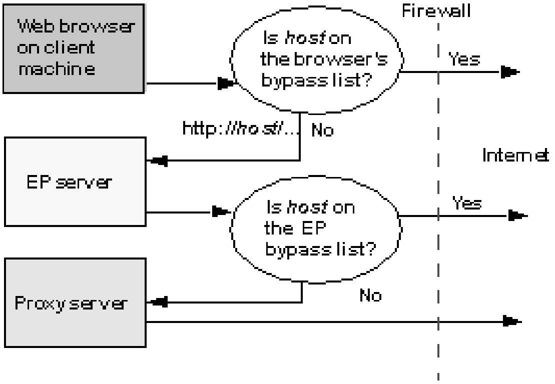
Overview

A proxy server is a server that acts as an intermediary between an internal user and the internet, which ensures security, administrative control, and caching service. A proxy server is associated with all or part of a gateway server that separates the enterprise network from the outside network, and a firewall server that protects the enterprise's network from outside intrusion. The proxy server administrator can choose to add access restrictions to a group of users by requiring user credentials during protocol connections. Most proxy servers support TCP/IP and protocols layered on top of it.

Enterprise Portal supports two proxy servers, Sun Open Network Environment Web Proxy Server 3.6 (Sun ONE) and Microsoft Internet Security & Acceleration Server 2000 (ISA 2000). The Sun ONE proxy server is supported on Solaris and Windows platforms. The ISA 2000 proxy server is supported on Windows platforms only. This version of Enterprise Portal supports basic authentication, which requires that clients present their credentials in the form of an encoded user name and password.

If you use a proxy server, Enterprise Portal routes HTTP requests and responses between clients and the proxy server, unless the host name is on a list that can bypass the proxy server. Figure 13-1 illustrates this scenario.

Figure 13-1: Routing requests through a proxy server


Web browser on client machine
Is *host* on the browser's bypass list?
Firewall
Yes
http://host/...
No
Internet
EP server
Is *host* on the EP bypass list?
Yes
Proxy server
No

To use a proxy server with Enterprise Portal, you must set up either a Sun ONE or Microsoft ISA 2000 Web proxy server, and configure proxy server support in EP.

Setting up a Sun ONE proxy server

This section describes how to download and configure a Sun ONE proxy server to run with Enterprise Portal. The steps are the same for both Solaris and Windows platforms, except where otherwise noted.

❖ **Installing and configuring a Sun ONE proxy server**

1 To install the trial version of the Sun ONE version 3.6 proxy server:

   a Using a Web browser, access the Sun Download Center at http://www.sun.com/software/download/products.

   b In the Download Center’s search list box, select Web, Portal & Directory Servers.

   c Scroll to the Web Servers section, and select Sun ONE Web Proxy Server 3.6.

   d Click Download, then follow the instructions to download and install the proxy server.

2 In the proxy server installation directory, start the admin server by running start-admin. The name of the process is “ns-admin.”

3 Open a Web browser, and connect to the Netscape Server Administration console using the host and port number that you specified when you installed the proxy server:

```
http://host:port/admin-serv/bin/index
```

4 Select Create New iPlanet Web Proxy Server 3.6. In the iPlanet Web Proxy Server Installation window, accept the default values, scroll to the bottom of the page, and click OK.

Note The value in the Server Port field must match the value of the proxy.port property in *global.properties.xml*, whose location depends on your application server:

- EAServer – *JAGUAR/Repository/WebApplications/onepage/config*.
- WebLogic – *BEA_ROOT/onepage/config*.

For more information, see the *Enterprise Portal Developer's Guide*.

5 Select Configure More About Your New Server.

6 In the Server Preferences window, click Server On.

7 Click Save and Apply.

8 In the upper-right corner, click Admin to return to the Administration console. Click Users and Groups.

9 In the Users and Groups window, enter the values below, then click Create User.

- First Name – the user's first name.
- Last Name – the user's last name.
- Full Name – the user's full name.
- User ID – this must match the value of proxy.user in *global.properties.xml*, and the value of Proxy User Name that you configure both on the Proxy tab in Portal Interface, and in the Change Proxy Configuration dialog box in Portal Studio.
- Password – this must match the value of proxy.password in *global.properties.xml*, and the value of Proxy Password that you configure both on the Proxy tab in Portal Interface, and in the Change Proxy Configuration dialog box in Portal Studio.
- Password – the same password.

- E-mail Address – the user’s e-mail address.

Repeat this step for each user that you want to create.

10 In the upper-right corner, click Server Administration to return to the Server Administration console.

11 Select the name of the server, then in the Server Preferences window, click Restrict Access.

12 In the Restrict Access window:

a Select “http://*” from the Editing drop-down list.

b If access control is off, select Turn On Access Control.

c Configure read access types (get, head, post, index, connect).

1 Select Deny, and click Edit Permissions.

2 In the Allow Access to a Resource window, scroll to the bottom of the window, and select Basic User Name and Password as the authentication method.

The user you just created should be in the list of users who are allowed access.

3 Click Done.

d Configure write access types (put, delete, move, mkdir, rmdir).

1 Select Deny, and click Edit Permissions.

2 In the Allow Access to a Resource window, scroll to the bottom of the window, and select Basic User Name and Password as the authentication method.

The user you just created should be in the list of users who are allowed access.

3 Click Done.

e Click OK.

13 Click Save and Apply.

Setting up a Microsoft ISA 2000 proxy server

This section describes how to download and configure a Microsoft ISA 2000 proxy server to run with Enterprise Portal. This proxy server runs on these platforms:

- Windows 2000 Server
- Windows 2000 Advanced Server
- Windows 2003 Server

❖ Installing and configuring an ISA 2000 proxy server

1 To install the trial version of the ISA 2000 proxy server, access the Microsoft Web page at http://www.microsoft.com/isaserver/howtobuy/default.asp, click Trial Software, click Download Now, and follow the installation instructions.

2 From the Windows Start menu, select Programs | Microsoft ISA Server | ISA Management. The ISA Management console displays.

3 Expand the Servers, highlight the name of the server you created when you installed the proxy server, right-click, and select Properties.

4 In the Properties dialog box, select the Outgoing Web Requests tab.

Note The value in the TCP Port field must match the value of the proxy.port property in *global.properties.xml*.

Highlight the name of the server, and click Edit.

5 In the Add/Edit Listeners dialog box, under Authentication, select Basic with this Domain, then click Select Domain.

6 Enter the name of the domain for the hosts that will originate the HTTP requests. Click Apply, then click OK.

7 In the ISA Management console, expand the server name, then expand Access Policy.

8 To create protocol rules to filter network traffic, highlight Protocol Rules, right-click, and select New.

Follow the instructions in the Protocol Rule wizard.

9 Close your Web browser.

Configuring proxy server support

Configuring Enterprise Portal to work with a proxy server requires:

1. Modifying global.properties.xml
2. Registering proxy user names and passwords with Portal Interface
3. Registering proxy user names and passwords with Portal Studio
4. Configuring a Web browser

❖ Modifying *global.properties.xml*

1. Using any standard ASCII text editor, open *global.properties.xml*, whose location depends on your application server:

   - EAServer – *JAGUAR/Repository/WebApplications/onepage/config*.
   - WebLogic – *BEA_ROOT/onepage/config*.

2. Enable proxy server support by setting the value of the proxy property to on:

   ```
   Property name=”proxy” value=”on”
   ```

3. Specify the host name or IP address of the proxy server; for example:

   ```
   Property name=”proxy.host” value=”10.22.85.198”
   ```

4. Set the value of proxy.port to the port number on which the proxy server listens for HTTP requests:

   ```
   Property name=”proxy.port” value=”5400”
   ```

   This value must match the port value that you configure in the Web proxy server; for Sun ONE, the field name is “Server Port”; for ISA 2000, the field name is “TCP Port.”

5. Specify a list of host names or IP addresses that can bypass the proxy server. Separate each host using a vertical bar. Typically, all machines inside the firewall should be on the list, including the EP server host. For example, if the local machine name is demo.sybase.com, the value of proxy.bypass_list should minimally include the following:

   ```
   Property name=”proxy.bypass_list”
   value=”demo.sybase.com|localhost|...”
   ```

Note Add the Enterprise Portal server host and localhost to the bypass list; otherwise, internal HTTP requests are routed to the proxy server then relayed back to EP, which diminishes performance.

If the EP server host and localhost are not on the bypass list, internal HTTP requests are routed to the proxy server then relayed back to EP, which diminishes performance.

❖ Registering proxy user names and passwords with Portal Interface

Every user must obtain a proxy user name and password from the proxy server administrator, and register them with Portal Interface.

1 To register a proxy username and password with Portal Interface, use a Web browser to access:

```
http://host.domain:port/onepage/index.jsp
```

Where *host* and *domain* identify the machine hosting the Enterprise Portal application, and *port* identifies the EAServer HTTP port.

2 If you are using Portal Interface for the first time, click Join Now.

On the Join Now tab, enter:

Field name	Description
First Name	First name
Last Name	Last name
E-mail Address	E-mail address
Phone Number	Telephone number
Member Name	Login name
Password	Login password
Confirm Password	Login password
Proxy Username	Proxy user name
Proxy Password	Proxy password
Confirm Proxy Password	Proxy password
Roles	Select one or more roles to grant to the user

If you already have a Portal Interface account:

a Enter your user name and password to log in to Portal Interface.

b Click My Info, then select the Proxy tab and enter:

- Password – login password.
- Proxy User Name – proxy user name.
- Proxy Password – proxy password.
- Confirm Proxy Password – proxy password.

3 Click Done.

❖ Registering proxy user names and passwords with Portal Studio

Portal Studio users must register their proxy user names and passwords in Portal Studio.

1 Using a Web browser, access this URL:

```
http://host.domain.com/onepage/index.html
```

Where *host* and *domain* identify the machine hosting the Enterprise Portal application.

2 Log in, then select Account Info. In the Account Info dialog box, select Proxy Configuration.

3 In the Change Proxy Configuration dialog box, enter:

- Password – login password.
- Proxy User Name – proxy user name; must match the User ID for one of the users that you created in the proxy server.
- Proxy Password – proxy password; must match the password for the same user that you created in the proxy server.
- Confirm Proxy Password – proxy password.

Click OK.

Configuring a Web browser

To route HTTP requests from a Web browser to the proxy server, you must configure a Web browser. Enterprise Portal currently supports Internet Explorer and Netscape Web browsers.

❖ Configuring an Internet Explorer Web browser

1 Open Internet Explorer, and from the browser menu, select Tools | Internet Options.

2 In the Internet Options dialog box, select the Connections tab, and click LAN Settings.

3 In the Local Area Network Settings dialog box:

a Select Use a Proxy Server for your LAN.

b In the Address field, enter the IP address of the proxy server.

c In the Port field, enter the port number on which the proxy server listens for HTTP requests.

d Select Bypass Proxy Server for Local Addresses.

e Click Advanced. The Proxy Settings dialog box displays.

f In the Exceptions list, enter the host names or IP addresses that can bypass the proxy server; separate entries with a semicolon. Include the EP host name in the list.

4 Click OK to save your changes.

❖ **Configuring a Netscape Web browser**

1 Open a Netscape window, and from the browser menu, select Edit | Preferences.

2 In the Preferences, expand the Advanced folder, and select Proxies.

3 In the Proxies property sheet:

a Select Manual Proxy Configuration.

b In the HTTP Proxy field, enter the IP address of the proxy server.

c In the corresponding Port field, enter the port number on which the proxy server listens for HTTP requests.

d In the No Proxy For field, enter the domain names that can bypass the proxy server; for example, enter “.sybase.com” to exclude clients in the Sybase domain from using the proxy server. Separate entries with a comma.

4 Click OK to save your changes.

Running Enterprise Portal with a proxy server

After you configure the Web browser to use a proxy server, the first time you attempt to access a URL that is not on the browser’s bypass list, a proxy server login window displays in which you must enter your proxy server user name and password. This should happen only once.

Debugging your proxy server configuration

This section describes how to determine whether the host names you specify to bypass, on either the proxy server or browser lists, are actually bypassing the proxy server.

Sun ONE proxy server

To debug the Sun ONE proxy server:

- In the *<SunOne>/<proxy_server_name>/logs/access* directory, run:

  ```
  tail -f <http_trace_file>
  ```

  Where *SunOne* is the Sun ONE proxy server installation directory, *proxy_server_name* is the name of the proxy server that you configured during installation, and *http_trace_file* is the name of the proxy server's HTTP trace file.

  You should see output similar to the following, although the format can be configured by the system administrator:

```
10.22.84.155 -- hhsi2 [04/Sep/2003:16:15:23 -0700]
"GET http://srd.yahoo.com:80/ HTTP/1.1" 200 38685 200 38685 -- ....
```

  Where 10.22.84.155 is the IP address of the host that sent the HTTP request; for Enterprise Portal, this should always be the EP host name. The "GET <*URL*>" string is the HTTP request.

  You can tell whether the proxy server is routing internal EP requests by the *URL*. If the *URL* is an EP servlet name, then we know the EP host name is not specified in the proxy.bypass property, and is therefore routed by EP to the proxy server. For example, if you see:

  ```
  GET <epHostName>:<ep port>/FWController ...
  ```

  Where "FWController" is an EP servlet, then we know that EP is routing internal requests to the proxy server.

  If the EP host is on the bypass list, you do not see any HTTP requests in the proxy server's HTTP trace file that include an EP host name, EP port number, or EP servlet in the URL.

For more help analyzing your system, see the *Sun ONE Web Proxy Server Deployment Guide* on the Sun Product Documentation Web page at http://docs.sun.com/source/817-0896-10/deploy.html.

Microsoft ISA 2000 proxy server

To debug the Microsoft ISA 2000 proxy server:

1. Start the ISA Management console by selecting Start | Programs | Microsoft ISA Server | ISA Management.
2. In the ISA Management console, expand these successive folders: Internet Security and Acceleration Server, Servers and Arrays, <*ServerName*>, Monitoring, Reports, Web Use. *ServerName* is typically set to the name of the host machine during ISA installation.

   This displays the Web Usage report. If the EP server host name displays in the Top WebSites section, then it is likely that neither the EP server host name nor the IP address are on the proxy server bypass list.

For more information about the trial version of the Microsoft ISA 2000 proxy server including a step-by-step tutorial, see the Microsoft ISA 2000 Trial Guide at http://www.microsoft.com/isaserver/evaluation/trial/ISAS_TrialGuide.doc.

CHAPTER 14 Implementing a Secure Web Proxy

You can configure Enterprise Security to use a Secure Web Proxy (SWP) to control access to preexisting back-end Web applications, and deliver multiple applications, Web pages, and data stores as a single application, and implement single sign-on features. This chapter describes how to install and configure an SWP, and discusses some precautions you must take to ensure that your users do not bypass the SWP security mechanism.

Note Configuring and implementing a Secure Web Proxy does not scale well for a large number of users. If you plan to use an SWP, you may need to add additional CPUs to handle the additional processing requirements.

Overview

A Secure Web Proxy functions as an interpreter or a gateway that intercepts client requests, routes the requests to the appropriate back-end application without allowing the client to access your resources directly, intercepts the server response, and if configured to do so, rewrites elements within the response to prevent users from accessing servers directly through embedded, absolute URLs.

As a gateway to back-end resources, an SWP serves several useful functions:

- Integrates content from multiple sources into a single application, delivering content to the user in a seamless manner.
- Provides the user an easy-to-use single point of entry to multiple resources located on a single, or multiple, sources.
- Allows the System Administrator to easily position enterprise data, protected by a firewall, for external access.
- Allows the System Administrator to control who is accessing data by implementing the optional authorization control.

Using the Secure Web Proxy

The user sends an HTTP connection request to an SWP. The HTTP request must contain:

- The URL that calls the SWP, such as http://*swp-server:port/webproxy/proxy*, where *swp-server* is the name of the server, *port* is the port number, and *webproxy/proxy* is an alias for the SWP as it was defined when it was deployed.
- The name of the application, such as “Sybase.” See “Defining applications” on page 225 for a definition of applications.
- The index of the application’s URL. The index is a number, 0 to *n*, that represents the sequential order of the URL in the URL index. The URL index is a comma-separated list of URLs that comprise an application. The user does not have to specify an index; however, if the application’s path data contains a number, the SWP interprets it as an index. Also, if an index number is not specified, the value defaults to 0.
- Any data that tells the SWP what the HTTP request is requesting, such as a Web page, an image, or an application. For example, *webpage.html* or *picture.gif*.

For example, the user may send this HTTP request:

```
http://swp-server:9008/webproxy/proxy/Sybase/0/webpage.html
```

An SWP scans the request, strips out the information that is used to connect to the SWP, (`swp-server:9008/webproxy/proxy`), parses and extracts the application name and the URL index (`Sybase/0`), and appends the requested data information (`webpage.html`) to generate a URL that can be interpreted by the back-end data server.

The result is:

```
http://www.sybase.com/webpage.html
```

When a back-end application sends a request to store or retrieve a cookie, an SWP intercepts the request and does not forward the request to the client. Instead, an SWP acts as a temporary cookie repository for application-level cookies. This makes the session appear seamless to users because they are not prompted to accept cookies from applications as they navigate through the different applications proxied by an SWP. When the user's session times out, all cookies stored for that session are deleted.

For example, if an application stores user-preference information in a cookie, an SWP intercepts those cookies and stores them for the duration of the session. When the session is closed, all user preferences are lost.

If the user has the appropriate permissions (see "Implementing single sign-on capabilities" on page 225), the URL is sent to the appropriate server. The server then returns the requested application as HTML or binary data.

If the response data is in binary format, an SWP returns the data unchanged. If the data is in HTML format, and if an SWP has been configured to do so, the SWP sends HTML data through filters to ensure the content does not contain any embedded links that allow the user to access the back-end data directly.

Using SWP filters

You can configure an SWP to scan for and rewrite requests containing client-side scripts that would otherwise bypass the SWP, and server responses that contain embedded absolute URLs. This helps to avert users from accessing back-end Web applications directly.

Rewriting application responses

When configured, an SWP scans HTML data, and rewrites HTML URLs. The SWP compares the URLs embedded in the HTML data to a list of URLs listed in the SWP properties file. See Table 14-2 on page 226.

If an SWP encounters a match in the HTML response data with a URL specified in the properties file, it rewrites the URL to be compliant with the SWP format.

For example, assume that the “Sybase” application was configured to proxy the following two URLs: http://www.sybase.com/ and http://my.sybase.com/. If the HTTP request, http://swp-server:9008/webproxy/proxy/Sybase/0/webpage.html returned an HTML response containing a link to http://my.sybase.com (for example, within an <Anchor> tag), the SWP rewrites the link to http://swp-server:9008/webproxy/proxy/Sybase/1/ where “1” refers to the index of the URL to which the link should proxy—in this case, http://my.sybase.com.

This is all transparent to the end user, except that the added use of CPU cycles can noticeably degrade response time. If an application uses only relative URLs, the parsing mechanism may not be necessary, and can be disabled. Test your applications after disabling parsing to ensure that the application contains only relative URLs. To disable parsing, see Table 14-2 on page 226, the Options.appname property.

When comparing URLs embedded in HTML documents to the URLs configured in the properties file, an SWP searches for the longest match first. This allows the rewriting script to rewrite related URLs simultaneously. However, the mechanism for searching, comparing, and rewriting URLs is CPU-intensive compared to an SWP configuration that does not rewrite embedded URLs. Be aware of the additional processing required for rewriting, and ensure that your infrastructure can handle the additional load.

Rewriting client-side scripting

Other items can control the flow of requests to an HTTP server as well as embedded URLs. For example, a client can access a Web page beyond the control of an SWP with a simple Java script like this:

```
<SCRIPT language=”JavaScript”>
     var webpage = “/absolute_url/”;
     document.location = webpage;
</SCRIPT>
```

where */absolute_url/* is the absolute path to the server, application, and requested data (Web page, image, and so on).

The above sample script bypasses an SWP and directs the client request directly to the source. To help address this, an SWP automatically adds a small script to the beginning of each HTML page that is proxied to the client, which can be used with certain limitations, to help block client-side scripting that redirects the user outside the control of the SWP; for example:

```
<SCRIPT language =”JavaScript”>
```

```
          function SybEPSSRewriter (href) {
             if (href.charAt (0) == '/')
                return "/security/proxy/Sybase/0" + href;
             return href;
          }
     </SCRIPT>
```

where *Sybase/0* is dynamically calculated for each page, depending on the origin of the page.

In this sample, the script assumes that all embedded URLs reference resources in the same index (`/0`). The SWP scans the page as it is proxied to the client and locates this pattern:

```
...
document.location
...
```

When the pattern is found, the SWP inserts calls within the client-side script to call the SybEPSSRewrite function, which dynamically rewrites URLs on the client side. The pattern is replaced with:

```
document.location = SybEPSSRewrite (webpage);
```

The result is:

```
<SCRIPT language="JavaScript">
     var webpage = "/absolute_url/";
     document.location = SybEPSSRewrite (webpage);
</SCRIPT>
```

For each page, you must determine the exact text of script that blocks the client-side Java script.

An SWP allows you to create patterns and pattern classes and store them in the SWP properties file. When script rewriting is enabled, an SWP scans client requests for patterns that match patterns defined in the properties file, and rewrites the script as necessary. See Table 14-1. Again, script rewriting is CPU-intensive. However, you can enable the script-rewriting filter at the application level, so not all applications are forced to use it.

Defining patterns

You can have as many patterns in a pattern class as you want. However, each pattern must have a unique index value. The order in which the patterns are processed within a pattern class is determined by the numeric order of the index values. For example, a pattern with an index value of 5 is tested before a pattern with an index value of 7. Each pattern class should be named with alphanumeric characters and no white spaces. For example: “Standard”.

Table 14-1 lists the properties that define patterns, and replacement patterns.

Table 14-1: Pattern class format

Property name	Default value	Description
Pattern.*pc*.*index*	none	This is the Perl regular expression pattern that, when found in a client-side script, calls the Java script rewriting function.
Pattern.*pc*.*index*.Substitution	$1 SybEPSSRewrite ($2)$3	The substitution pattern used when a match is found in the match pattern. The default value is appropriate for most cases.

Note *pc* and *index* are placeholders for the pattern class (pc) name and the index number (0 to *n*). You must replace these with the class name and a number.

Creating patterns and pattern classes

Patterns and pattern classes are stored in a Java properties file. You can use any standard ASCII text editor to add, modify, or delete patterns and pattern classes. As Java properties, patterns are subject to certain rules. For example, a backslash (\), must be escaped by another backslash (\\), or it is interpreted as a special character. You can escape other types of characters, such as single quotes, spaces, double quotes, and parenthesis.

The following example patterns are useful for many common Web sites. Backslashes are escaped as two backslashes.

```
Pattern.Standard.101=(.*?window\\.open\\s*\\(\\s*)(.*?)(,.*)
Pattern.Standard.102=(.*location\\.href\\s*\\=\\s*)(.*?)(;.*)
Pattern.Standard.103=(.*\\.location\\s*\\=\\s*)(.*)(;.*)
Pattern.Standard.104=(.*\\.action\\s*\\=\\s*)(.*)(;.*)
```

The patterns above conform to the `Pattern.pc.index` format, where the pattern class (pc) is “Standard” and the index ranges from 101 to 104.

Each pattern has three sets of data, defined by parenthesis. If there is a match in a client-side JavaScript with any of these patterns, the SWP replaces the pattern with the value defined in `Pattern.pc.index.Substitution`.

Implementing single sign-on capabilities

Single sign-on capabilities allow the user to access all the system assets for which they have access permissions with a single user name and password provided during the session initialization. If an SWP is configured for user authentication, it invokes an authenticator—a Java class API—that contacts the ACDB to retrieve user credentials that are stored in a proxy authentication Information record.

For each application to which you want to grant the user single sign-on access, you must create a proxy authentication information record. See Chapter 9, “Proxy Authentication.”

Enterprise Security provides three preconfigured authenticators. Using these authenticators requires Java, HTML, and HTTP programming knowledge. For more information on the authenticators, see the Javadoc HTML files, which are located in the *docs/html* subdirectory of your Enterprise Security installation.

Defining applications

The system administrator integrates resources, such as Web pages, data stores, and applications, into a single application. An application consists of one or more base URLs, which share the same session. For example, an application called “Sybase” might include the base URLs: http://my.sybase.com, http://www.sybase.com, and https://login.sybase.com.

You can define as many applications as you want. Each application is defined by the properties in Table 14-2.

Table 14-2: Application properties

Property name	Description
URLBaseRedirection.*appname*	A comma-delimited list of URLs that an SWP protects by redirecting client requests. This property is required.
DN.*appname*	The DN (distinguished name) of the asset, used to check for access permissions on this resource. The user must have read permissions to access this application. If no DN is specified, you cannot enable single sign-on capabilities, and this resource is protected from access only during the initial login validation.
Options.*appname*	"noparse" is the only valid value for this property. If you set this property, an SWP does not parse HTML responses before sending them to the client. Use this parameter only if you are certain that the application does not use absolute URLs.
Patterns.*appname*	A comma-delimited list of pattern classes that are used for script rewriting. The sample below uses only one pattern, "Standard." See "Defining patterns" on page 224.
Authenticator.*appname*	The fully qualified name of the Java class that is used as the AuthorizationService to access user credentials stored in the ACDB. See "Implementing single sign-on capabilities" on page 225.
AuthenticatorURL.*appname*	A configuration parameter passed to the authenticator. If there is no configured authenticator, this parameter is not used.
AuthenticatorParams.*appname*	A configuration parameter passed to the authenticator. If there is no configured authenticator, this parameter is not used.

In each of the property fields listed in Table 14-2, you must replace *appname* with the name of the application you are defining. For example, the following properties define an application named "Sybase."

```
URLBaseRedirection.Sybase=http://www.sybase.com/,http://my.sybase.com/
https://login.sybase.com/DN.Sybase=a1=Sybase.com,o=Sybase,c=usPatterns.Sybase
=StandardAuthenticator.Sybase=com.sybase.ep.security.webproxy.auth.
PostAuthenticator2AuthenticatorURL.Sybase=
https://login.sybase.com/login/check_passwordAuthenticatorParams.Sybase=
password=[password]&Submit=Login&refer=http%3A%2F%2Fmy.sybase.com
%2Fmysybase&error_code=&username=[user]
```

Installing and configuring a Secure Web Proxy

An SWP is a J2EE Web application that can be deployed to any J2EE 1.2-compliant application server, such as EAServer or WebLogic.

❖ Installing an SWP in EAServer

EAServer must be installed and configured before you install an SWP.

1. Start Jaguar Manager:
   - Windows – select Start | Programs | Sybase | EAServer 4.2 | Jaguar Manager.
   - UNIX or Linux – change to *$JAGUAR/bin*, and enter `./jagmgr`.
2. From the menu bar, select Tools | Connect | Jaguar Manager, and enter:
   - User name – `jagadmin.`
   - Password – password for jagadmin; the default is blank.
   - Host name – the name of the machine where EAServer is installed.
   - Port number – `9000`.
3. Once you are connected, right-click Web Applications in the left pane.
4. Select Deploy | J2EE War.
5. Change to *Security/lib*.
6. Highlight *webproxy.war*, and click Select, then Next.
7. When the wizard displays "Deployment Successful," click Close.
8. In Jaguar Manager, select Servers | Jaguar and right-click Installed Web Applications.
9. Select Install Web Application.
10. Select Install an Existing Web Application.
11. Select the "webproxy" Web application and click OK.
12. Refresh the Web application when prompted.

This installs the appropriate files into the appropriate directory in EAServer.

❖ Installing an SWP in a WebLogic server running on Solaris

WebLogic must be installed and configured before you install an SWP.

1. In *$BEA_HOME*, create a subdirectory called "webproxy."
2. Copy *$SECURITY/lib/webproxy.jar* to the *$BEA_HOME/webproxy* directory.
3. Change to the *$BEA_HOME/webproxy* directory, and unjar the *webproxy.war* file:

```
jar -xvf webproxy.war
```

4 Using a text editor, open the *web.xml* file, and:

   a Enter the absolute path to the *swp.properties* file. In the following line, replace *BEA_HOME* with the installation location of the BEA WebLogic server:

```
<env-entry-value>/BEA_HOME/webproxy/WEB-INF/swp.properties
</env-entry-value>
```

   b To use certificate authentication to SWP, add the following code block, after the `</session_config>` line:

```
<login-config>
   <auth-method>CLIENT-CERT</auth-method>
</login-config>
```

5 Log in to the WebLogic Server Console as the admin user, and deploy the "webproxy" Web application:

   a In the left pane, expand the Deployments folder, then click Web Application Modules.

   b In the right pane, click Deploy a New Web Application Module, then select "bea" as the location.

   c Select "webproxy," then click Target Module.

   d Click Deploy. When deployment is complete, you see "webproxy" under the Web Application Modules in the left pane.

Configuring an SWP

Configure an SWP using an external properties file, *swp.properties*. The properties file is installed when you deploy the WAR file, and its location must be identified by the propertyFile environment variable.

❖ Setting up the propertyFile environment variable

1 In Jaguar Manager, expand these successive folders: Servers, Jaguar, Installed Web Applications.

2 Highlight "webproxy" and select File | Web Application Properties.

3 In the Properties dialog box, select the Environment tab, and click Add. A new row is added to the Environment entries list. Enter these values:

- Entry – `propertyFile`.
- Type – `String`.
- Value – ***`path`***`/swp.properties`, where *path* is the full path to the properties file.

Restrict access to *swp.properties* *swp.properties* contains user names and passwords in plain text. Sybase recommends that you secure read and write access to this file.

If you deploy the WAR file to EAServer as a Web application named “webproxy,” a sample *swp.properties* file is installed in the *Repository/WebApplication/webproxy/WEB-INF* subdirectory of your EAServer installation.

To use an SWP, you must edit *swp.properties*. You can use any standard ASCII text editor to edit this file. Table describes the SWP configuration properties. By default, the ApplicationServer.java.naming.* properties are commented out in *swp.properties*. If the SWP is installed in a different application server container than Enterprise Security, uncomment these properties and supply a value for at least the ApplicationServer.java.naming.provider.url property.

SWP configuration properties

Property name	Description
ApplicationServer.java.naming.provider.url	Specifies the URL that the SWP uses to connect to a remote Enterprise Security installation.
ApplicationServer.java.naming.security.principal	Specifies the user name that the SWP uses to connect to a remote Enterprise Security installation. **Note** For an anonymous connection, the user name and password values can be empty.
ApplicationServer.java.naming.security.credentials	Specifies the password that the SWP uses to connect to a remote Enterprise Security installation.
WEBPLUGINUSER	User name to enable certificate authentication to an SWP via HTTPS listeners. This user must have the WebPluginRole.
WEBPLUGINPASSWORD	The password of the WEBPLUGINUSER.

Additional configuration options

When you configure an SWP, configure some Web-based pages to notify users if they have used an incorrect user name and password combination, if they have attempted to access resources for which they do not have permissions, and so on.

Table 14-3 lists property files that you can use to call HTML files that inform users of these errors.

Table 14-3: Web page configuration

Property name	Default value	Description
WeakAuthPage	*SybSecurityLogin.html*	The standard user name and password authentication form to access an SWP resource, when the user has not yet provided a valid user name and password.
InvalidLoginAccess	*SybSecurityInavlidLogin.html*	The page presented when the user enters an invalid user name or password.

If an invalid access attempt is made, an SWP returns the appropriate HTTP response code. You can map this response code to the Web page of your choice using EAServer parameters.

To ensure that users accessing your data resources are who they claim to be, you may want to enable the caching timeout settings. These settings determine how long a user session continues before an SWP revalidates the user's access permissions to the secured system, or to specific assets.

Table 14-4: Caching timeout properties

Property name	Default value	Description
assetCacheTimeout	60 (seconds)	The frequency at which an SWP revalidates user access to a Web asset.
sessionCacheTimeout	60 (seconds)	The frequency at which an SWP revalidates user access to the system.

Integrating an SWP into Enterprise Portal

Beginning with Enterprise Security 6.0, you can integrate an SWP into Enterprise Portal as a portlet. First deploy the SWP into EAServer as a standalone Web application—see “Installing and configuring a Secure Web Proxy” on page 226.

Once you verify that the SWP is working correctly as a standalone Web application, use Portal Studio to create a portlet that includes a JSP element with the SWP content you want. For complete information about creating and configuring portlets, see the *Enterprise Portal Developer's Guide*.

To configure the JSP element:

- In Portal Studio, set the following parameters in the JSP Element Definition window:
  - WAR File – `webproxy.war`.
  - Web App Display Name – `webproxy`.
  - Initial Resource – `proxy/[`***`application_name`***`]/`.

    Replace [*application_name*] with the name of the configured application from *swp.properties* that you want to display in the portlet.
  - Single Sign On Required – select this option.
  - 302 Redirect – select this option.

In *swp.properties*, verify that these properties are commented out:

- ApplicationServer.java.naming.provider.url
- ApplicationServer.java.naming.security.principal
- ApplicationServer.java.naming.security.credentials

Once this configuration is complete, you should see the appropriate content display in the preview window. See the *Security/security.log* and the server log file—for EAServer, this is typically *$JAGUAR/bin/Jaguar.log*—for error and diagnostic information.

CHAPTER 15 Configuration Properties

Enterprise Security features and the properties that control them are divided into two categories:

- Global properties – affect features that are common to your entire security system. Global properties are configured using a properties file.
- Domain-specific properties – affect the objects within a single domain. These properties are stored in the ACDB, and configured using either Enterprise Security Manager or SMAPI.

This chapter describes both the global properties, and the domain-specific properties.

Global properties

Global Enterprise Security properties are configured with the *security.properties* file. Most properties in the file have a default value, and any property value that you set in *security.properties* is maintained whenever you perform an upgrade of the Enterprise Security software.

Since all global security configuration properties are contained in this file, Sybase recommends that you make a backup copy of this file and store the backup in a safe location before you make any changes to it, then again after you have configured your security subsystem.

Additionally, you should protect the configuration information in *security.properties* so that users cannot disable any configured features. Enterprise Security provides a securetool command to encrypt and decrypt this file as necessary. See “Encrypting and decrypting the security.properties file” on page 234.

The location of *security.properties* depends on your application server.

- EAServer – *JAGUAR/java/classes/com/sybase/ep/security.*
- WebLogic – *BEA_ROOT/sybepsecurity/etc/com/sybase/ep/security*.

Encrypting and decrypting the *security.properties* file

This section describes how the PSO can encrypt and decrypt the *security.properties* file using securetool. For information about securetool, see Chapter 4, "Using securetool."

Warning! The keyFile property must remain in plain text in the *security.properties* file; otherwise, Enterprise Security services do not work.

❖ Encrypting *security.properties*

This procedure reads the *security.properties* file, and writes the encrypted contents to the *security.properties.enk* file. Both of these files must be in the same directory.

1 Make a backup copy of *security.properties*, and save it in a secure location.

2 Change to the *SECURITY/bin* directory, and run:

```
securetool enc_dec_file --operation encrypt
   --username pso --password pso_password
   --input_file <propspath>security.properties
   --output_file <propspath>security.properties.enk
```

Where *pso* and *pso_password* are the user name and password for the PSO, and <*propspath*> is the path to the *security.properties* file.

3 Edit *security.properties* and delete everything, except the properties that must be in plain text. The file must contain at least the keyFile property.

Note If a property is defined in both *security.properties* and *security.properties.enk*, the property value in *security.properties.enk* takes precedence over the value in *security.properties*.

❖ Decrypting *security.properties.enk*

- To decrypt the *security.properties.enk* file, change to the *SECURITY/bin* directory, and run:

```
securetool enc_dec_file --operation decrypt
   --username pso --password pso_password
   --input_file security.properties.enk
   --output_file security.properties
```

Where *pso* and *pso_password* are the user name and password for the PSO.

❖ Changing an encrypted property value

To change a property value that is encrypted in the *security.properties.enk* file:

1. Decrypt *security.properties.enk*.
2. Edit *security.properties* to change the property value.
3. Reencrypt *security.properties*.
4. Edit *security.properties* to remove the properties that should not remain in plain text.

Configuring global properties

This section describes how to configure the Enterprise Security global properties, which define the features that affect your entire security system.

To configure global security properties, use any standard ASCII text editor to edit the *security.properties* file located in the *java/classes/com/sybase/ep/security* subdirectory of your EAServer installation. If a property does not exist in *security.properties* and it has a default value specified in a configuration properties table, the default value is in effect. To modify the value of a property that is not specified in *security.properties*, edit the file and add it. Property names are case-sensitive. The order of the properties in the file does not matter.

If you make any changes to *security.properties*, you must restart EAServer for the changes to take effect.

The tables below describe all the properties in the *security.properties* file that you may need to edit to enable Enterprise Security features. Where applicable, the tables also show default values, and a description.

Table 15-1 lists the system-wide auditing properties that determine where auditing information is written. For more information about auditing, see Chapter 6, "Auditing."

Table 15-1: Auditing properties

Property name	Default value	Description
auditAuthenticationManagementOperations		Specifies whether to audit authentication-delegate calls to com.sybase.ep.security.management beans. This is used only for the LDAP authentication delegate and any custom authentication delegates that use the Management APIs to replicate data into the ACDB.
auditDatabaseConnCache		Connection cache created in EAServer to obtain a connection to the audit database server.
auditDatabaseInsertSql		The SQL statement used to insert a record into the Audit table.
auditDatabaseJdbcDrive		JDBC driver used to connect to the audit database server. Used only when auditSPI is set to database.
auditDatabaseJdbcUrl		The JDBC URL used to connect to the audit database. Used only when auditSPI is set to database.
auditDatabasePassword		The password that is used to connect to the audit database. Used only when auditSPI is set to database.
auditDatabaseUsername		The user name that is used to connect to the audit database. Used only when auditSPI is set to database.
auditKey	sybase_ep	Key value used internally both to generate and verify audit requests. It ensures that outside clients cannot add audit information to the audit trail. You can change the value.
auditLog	*audit.log* Currently created in the *$JAGUAR/bin* directory.	The name of the file in which auditing information is written. This is the backup destination when auditSPI is set to either dbconncache or database.

Property name	Default value	Description
auditOverflowLog	*auditOverflow.log* Typically created in the SYBASE directory; for example, on Windows: *c:\sybase\auditOverflow.log.*	The name of the file in which auditing information is written if an error occurs while writing to the primary audit destination. This is the secondary backup destination when auditSPI is set to either dbconncache or database.
auditSPI	file	• Set the value to "dbconncache" to write the audit output to a JDBC-compliant database using the EAServer connection cache feature. • Set the value to "database" to write the audit output to a JDBC-compliant database. • Set the value to "file" to write the audit output to a file. • Set the value to the name of a Java class to direct the output to your custom Java application. A sample Java class is illustrated in "Implementing a custom SPI" on page 134.

Table 15-2 lists the authentication delegate properties.

Table 15-2: Authentication delegate properties

Property name	Default value	Description
adminRoleDN		The DN for the admin role
defaultDSORoleDN		The DN for the default domain's security officer
guestRoleDN		The DN for the guest role
pluginRoleDN		The distinguished name (DN) for the plug-in role

Table 15-3 describes properties used to manage the asset cache.

Table 15-3: Cache manager properties

Property	Default value	Value
assetCacheSize	1000	Specifies the asset cache size.
assetCacheFlushPercentage	10	Specifies the percentage of the asset cache size at which the cache should be flushed.
cachemgrLog		Specifies the path to the log file. This file is for logging plug-in cache and ACDB synchronization problems.

Table 15-4 lists the properties that you must set to enable LDAP. For information about LDAP, see Chapter 10, “Configuring LDAP Authentication.”

Table 15-4: LDAP properties

Property name	Default value	Description
attributeList		User attributes that can be mapped to LDAP attributes.
ldap.AttributeMapper.certificateAttributes		A comma-delimited list of attribute names that can be searched for in the supplied certificate. The attributes are extracted from the certificate DN.
ldap.AttributeMapper.directoryAttributes		A comma-delimited list of the LDAP attributes that correspond to the ldap.AttributeMapper.certificateAttributes..
ldap.attributeMappingNames		The subject attributes that are populated when authenticating an LDAP user. The currently supported attributes are: • uid – the user’s login name; this is required. • cn – the user’s common name; this is required. • email – the user’s e-mail address. • telephoneNumber – the user’s telephone number. • firstName – the user’s first name. • lastName – the user’s last name.
ldap.attributeMappingValues		The LDAP attributes that correspond to the subject attributes specified in ldap.attributeMappingNames: • uid – the user’s login name; this is required. • cn – the user’s common name; this is required. • email – the user’s e-mail address. • telephoneNumber – the user’s telephone number. • givenName – the user’s first name. • sn – the user’s surname.
ldap.connection.bindname		The user name that should be used to initially establish the LDAP connection. Leaving this field blank means anonymous binding to the LDAP server.
ldap.connection.password		The password for the name specified in ldap.connection.bindname.
ldap.certificateMapper		Certificate mapper.
ldap.connection.host	none	The host name of the LDAP server. You can include multiple host names, delimited by a space character. You can also include the port number. For example: tsandee-pc:389. Port numbers within this property override the port number specified in the ldap.connection.port property.
ldap.connection.port	389	The port number at which the LDAP delegate connects.

Property name	Default value	Description
ldap.followReferrals	true	Determines whether or not referrals are followed automatically when encountered by the delegate.
ldap.groups		The properties that reside under this package define mappings from LDAP groups to Enterprise Security groups. The LDAP entry must be of the object class groupOfUniqueNames (static) or groupOfURLs (dynamic). [mapid] is a user-defined string that binds the two parameters together into one mapping.
ldap.connection.minpooled		The minimum number of LDAP connections in the pool.
ldap.connection.maxpooled		The maximum number of LDAP connections in the pool.
ldap.organizationMapper	none	The class name of a class that meets the qualifications specified in the Organization Mapping section. If this is not specified, any subjects who authenticate are placed in the root organization.
ldap.rebuildSubjects		Specifies whether to copy subject information from LDAP into the ACDB every time a user authenticates.
ldap.roles		The properties that reside under this package define mappings from LDAP groups to Enterprise Security roles. The LDAP entry must be of the object class groupOfUniqueNames (static) or groupOfURLs (dynamic). [mapid] is a user-defined string that binds the three parameters together into one mapping.
ldap.searchBase		This allows retrieval of authenticated users.
ldap.searchFilter	&({UID_ATTR}={UID}) (objectClass=inetOrgPerson)	The LDAP query which is used to retrieve a user's LDAP record. Normally, this value is dynamically calculated using the attributes.uid property and a user-supplied credential. The user credentials are dynamically substituted for instances of the string {UID} before executing the query. The value of the uid_attr property is substituted for the {UID_ATTR} string.

Table 15-5 list the properties to configure enhanced password security features. For more information, see Chapter 8, "Securing Accounts and Assets."

Table 15-5: Password properties

Property name	Allowed values	Default value	Description
passwordAllowedEncodings	Any valid encoding type. You can enter multiple values in a comma-delimited string.	SHA	Defines password encoding types. This effects validation for passwords already stored in the database. For example, to allow users whose passwords are stored using the MD5 encoding, the value must include "MD5".

Property name	Allowed values	Default value	Description
passwordAllowUnsaltedAuthentications	true or false	false	Specifies whether to allow users who have unsalted passwords defined in the database to be authenticated. This security check can be disabled when importing unsalted passwords from other sources; for example, from the iPlanet Directory Server, which stores unsalted passwords by default.
passwordDefaultEncoding	Any valid encoding type. Only a single type is allowed. Either "TXT" or the name of a message digest algorithm defined by the java.security. MessageDigest class.	SHA	Defines the default encoding of the passwords that are stored by a SMAPI or other internal routine, such as a PasswordUtils class routine.
passwordRandomSaltLength	An integer.	8	The number of bytes of random salt data to generate whenever passwords are generated. In general, higher values provide more secure storage of the password. This formula determines the total size in bytes of the encoded password, which must be less than 64: round4(*salt_bytes* * 1.34) + round4(*encoding_bytes* * 1.34) + 3 + *length_of_algorithm_name*) The round4 operator rounds up to the nearest multiple of 4. Assuming the algorithm name is "SHA" (20 encoding bytes) and the number of salt bytes is 8, the size of the encoded password would be: round4(8 * 1.34) + round4(20 * 1.34) + 3 + 3 = 12 + 28 + 3 + 3 = 46 bytes

Property name	Allowed values	Default value	Description
passwordRandomSaltsEnabled	true or false	true	Allows the administrator to define whether to generate random salted data to encode with the user's passwords, when encoding new passwords. This data is included in the encoded password string immediately after the encoding, within the same curly brackets; for example: `{SHA:base64encodedsalt}encodedPasswordData` If the value is false, unsalted passwords are always allowed to authenticate, and allowUnsaltedAuthentications is always true.

Table 15-6 defines the PortalSession properties.

Table 15-6: PortalSession properties

Property name	Default value	Description
keyFile		The path to the encryption key file.
oldKeyFile		The path to the old encryption key file.
sessionDuration	3600 (seconds)	The duration of a portal session. This value must be the same as the value of the EAServer com.sybase.jaguar.server.authtimeout property, which is defined in the *$JAGUAR/bin/Jaguar.props* file. If you update either property, you must update both properties.
sessionPurgeInterval	900 (seconds)	How often the service that removes expired sessions from the database runs.

Table 15-8 defines the role mapping properties.

Table 15-7: Role mapping properties

Property name	Default value	Description
defaultRolemappingEnabled	true	If true, EAServer attempts to perform role mapping implicitly between J2EE roles and Enterprise Security roles, which eliminates the need to add role mappings to the *security.properties* file. For more information, see "Implicit role mapping" on page 146.
easerverRolemap.epdefault_0.epdn	PortalWebPlugin	The DN of the Enterprise Security PortalWebPlugin role.

Property name	Default value	Description
easerverRolemap.epdefault_0.jagrole	PortalWebPlugin	The name of the EAServer role that maps to the Enterprise Security PortalWebPlugin role.
easerverRolemap.epdefault_1.epdn	PortalSecOfficer	The DN of the Enterprise Security PortalSecOfficer role.
easerverRolemap.epdefault_1.jagrole	PortalSecurityOfficer	The name of the EAServer role that maps to the Enterprise Security PortalSecOfficer role.
easerverRolemap.epdefault_2.epdn	PortalAdmin	The DN of the Enterprise Security PortalAdmin role.
easerverRolemap.epdefault_2.jagrole	PortalAdmin	The name of the EAServer role that maps to the Enterprise Security PortalAdmin role.
easerverRolemap.epdefault_3.epdn	PortalGuest	The DN of the Enterprise Security PortalGuest role.
easerverRolemap.epdefault_3.jagrole	PortalGuest	The name of the EAServer role that maps to the Enterprise Security PortalGuest role.

Table 15-8 defines the distinguished name (DN) of the self-registration group.

Table 15-8: Self-registration group name

Property name	Default value	Description
selfRegistrationGroupName	SelfRegGroup	The DN of the self-registration group. **Note** This is changed from Enterprise Security versions earlier than 6.0, where this property identified the group name.

Domain-specific properties

This section describes properties that can be configured for each security domain. These properties define the rules for managing security issues, such as auditing and password expiration. The default property values define the initial rules that apply to the default domain. You can modify the rules for a domain by configuring these domain-specific properties using either Enterprise Security Manager or SMAPI. For more information, see Chapter 5, “Delegated Administration.”

The property in Table 15-9 defines how often the domain-specific properties are refreshed.

Table 15-9: Refresh interval property

Property name	Default value	Description
propertyRefreshTimeInterval	60 (seconds)	Defines how often the properties for this domain are refreshed.

Table 15-10 describes the domain-specific account expiration properties.

Table 15-10: Account expiration properties

Property name	Default value	Description
defaultAccountExpirationDuration	0	The number of days any account (active or inactive) remains valid. If set to 0, accounts remain valid indefinitely.
inactivityExpirationDuration	0	The number of days an inactive account remains valid. If set to 0, inactive accounts remain valid indefinitely.

Table 15-11 describes the domain-specific account lock properties.

Table 15-11: Account lock properties

Property name	Default value	Description
allowedInvalidLoginAttempts	3	The number of invalid login attempts users are allowed before their account is locked.
allowedInvalidAccessAttempts	5	The number of invalid access attempts users are allowed before their account authorization is locked.
authCountTimeSpan	1440 (minutes)	The number of minutes during which unauthorized attempts to access security objects are counted.
authLockEnable	true	Set to true to enable the system to lock out users after a specified number of attempts to access a security object that they do not have permission to access.
authLockPeriod	-1	The duration of an authorization lockout.
closeSessionOnAuthLock	true	Set to true to terminate users' sessions when their authorization is locked.
loginClearHistory	true	Set to true to delete information about invalid log-in attempts when users successfully log in.
loginCountTimeSpan	60	The number of minutes during which the number of invalid login attempts are counted.
loginLockEnable	true	Set to true to enable the system to lock out users after a specified number of invalid log-in attempts.
loginLockOnAuthLock	true	Set to true to prevent users from logging in when their authorization is locked.

Property name	Default value	Description
loginLockPeriod	-1	The duration of a lockout. Specify one of: • -1 to lock the account until an administrator specifically unlocks it. • The number of minutes to keep the account locked.

Table 15-12 describes the domain-specific auditing properties. For more information about auditing, see Chapter 6, “Auditing.”

Table 15-12: Auditing properties

Property name	Default value	Description
auditEnable	false	Specifies whether auditing is enabled for this domain.
auditExcludeFilter		Specifies which events to exclude from auditing; applied after auditIncludeFilter.
auditIncludeFilter	(ResourceClass= SYSTEM.*)	Specifies which events to audit.
auditJMSEnable	false.	Specifies whether to send audit records to a JMS topic, in addition to the primary logging location defined by auditSPI.
auditSubjectDNEnable	false	Specifies whether to include the subject DN in auditing records. If set to true, the subject DN is added to the XML audit record column, and is available to insert in the Subject DN column—see Table 6-12 on page 128. If set to true, performance may be slower.
auditSuspendOnFailure	false	Specifies whether to suspend auditing when errors occur writing an audit record.

Table 15-13 describes the domain-specific password properties.

Table 15-13: Password properties

Property name	Default value	Description
passwordDuration	0	Specifies how many days a password remains valid after it is set. This value affects all users, except those explicitly excluded from this policy. You can also specify the password duration in months or years; for example: • 3m = three months. • 1y = one year.
expiredPasswordChangeWindow	0	The number of days after a password expires that users are allowed to change their password.

Property name	Default value	Description
passwordStrengthVerification	(String)null	The name of the password-strength verification component. See “Configuring the sample password-strength verification component” on page 158.

Resetting domain property values

If an invalid configuration property value makes it impossible to log in to the domain; for example, if you configure an audit filter incorrectly, you can reset all domain properties to their default values to restore access to the domain.

❖ Resetting domain configuration property values

1 Log in to the ACDB.

2 Reset all domain properties to their default values using this SQL statement:

```
UPDATE SecurityDomain SET Rules = NULL WHERE DomainName=”DefaultDomain”
```

To reset values in a domain other than the default, replace “DefaultDomain” with the name of the domain. Currently, you can reset all domain configuration values but not individual property values.

3 Restart your application server.

Glossary

access control

The process of controlling who has access to a data source.

ACDB

Access Control Database. A central database that stores all of the user's authorization and authentication information, such as user name and password credentials, digital certificates, and access permissions to the system components. The ACDB structures data in the form of a Lightweight Directory Access Protocol (LDAP)–compliant directory and is stored on the Adaptive Server, unless you have specifically configured a third-party database to store user information.

adapter

A component that provides an interface between an internal application and external applications or messaging systems. An adapter detects events and validates event contents.

API

An acronym for application programming interface. A set of routines, protocols, and tools for building software applications that enables programs to communicate with each other.

applet

A small program in an HTML-based program built with Java that a browser temporarily downloads to, and runs from, a user's hard drive. Java applets can be downloaded and run by any Java-interpreting Web browser, such as Microsoft Internet Explorer and Netscape Navigator. Java applets can be used to add multimedia effects, such as background music, real-time video displays, animations, and interactivity, such as calculators and games, to Web pages without having to send a user request back to the server.

application integration

Usually, a solution designed for a specific industry that allows multiple programs to work together seamlessly.

This is an approach that provides application logic and data to the application server via proxy components. During design, components are defined using metadata (sp_catalogs, COBOL copy books, IDL repositories) that associates the logic and data with a component name. Code generation and deployment into the application server are features of application integrators. When the component is available in the application server, any developer can use it without understanding the specifics of how it works.

application service provider Third-party companies that manage and distribute software-based services and solutions across a wide-area network from a central data center.

ASP Active Server Pages. An open, compile-free application environment in which Web developers can combine HTML, scripts, and reusable Active Server components. ASP technology enables server-side scripting for IIS with native support for both Visual Basic Scripting Edition and JScript.

asset Any object within the enterprise’s computer systems, including but not limited to a document, database information, another computer system, an application, and so on.

audit, auditing A method to provide individual accountability for users performing operational and administrative tasks. User actions are recorded in an audit log so the system administrator can see who is doing what while the user is logged in to the networked system.

authentication The process of verifying the identity of the person trying to enter a network system.

authorization The term used to describe the process of assigning permissions to users or groups of users to access system assets.

B2B An acronym for business-to-business. The ability of companies to deliver products, services, support, and information over the Internet to other companies with whom they do business.

B2Bi An acronym for business-to-business integration. B2Bi enables a business to integrate its computer systems with those of its business partners (suppliers, vendors, customers), eliminating redundant data entry, and speeding up order turnaround times.

bean A reusable software component. Beans can be combined to create an application.

binding The association of a client and a server.

broker A type of middleware that connects clients and servers. An example is an Object Request Broker.

buffered queue A message queue that resides in memory.

business object An application-level component you can use in unpredictable combinations. A business object is independent of any single application.

Business objects provide a natural way to describe application-independent concepts such as customer, order, competition, money, payment, car, and patient. They encourage a view of software that transcends tools, applications, databases, and other system concepts.

certificate authorities

Entities that validate identities and issue digital certificates. They can be either independent third parties or organizations running their own certificate-issuing server software. The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies.

cipher suites

As part of the SSL handshake, the client and server agree upon a common cipher suite. The cipher suite includes SSL/TLS support options, algorithms used for key exchange, and digital signatures.

class

In object-oriented programming, a category of objects. For example, there might be a class called shape that contains objects that are circles, rectangles, and triangles. The class defines all the common properties of the different objects that belong to it.

client/server

A network architecture in which one or more computers (servers) accept requests for services from one or more workstations (clients).

This may also refer to a back-end application (server) that accepts requests for information from a front-end application (client).

communications middleware

Software that provides inter-application connectivity based on communication styles such as message queuing, ORBs, and publish/subscribe.

communications protocol

A formally defined system for controlling the exchange of information over a network or communications channel.

component

In programming and engineering disciplines, a component is an identifiable part of a larger program or construction. Usually, a component provides a particular function or group of related functions.

In object-oriented programming and distributed object technology, a component is a reusable program building block that can be combined with other components in the same or other computers in a distributed network to form an application. Examples of a component include a single button in a graphical user interface, a small interest calculator, and an interface to a database manager.

Components can be deployed on different servers in a network and communicate with each other for needed services. A component runs in a context called a container. Examples of containers include pages on a Web site, Web browsers, and word processors.

connection pooling

Connection pooling is a performance optimization based on using collections of preallocated resources, such as objects or database connections. Pooling results in more efficient resource allocation.

connectionless communications

Communications that do not require a dedicated connection or session between applications.

continuous availability

The ability of a computer to stay up and running 24 hours a day, 7 days a week. Continuous availability requires that solutions are both highly reliable and quickly recoverable in the event of failure. See also high availability.

controlling assets

Special assets in a domain that control access to the other security objects in the domain. There are seven controlling assets in each domain, one for each security object type: asset, domain, group, organization, role, and user, and one that controls access to custom AccessType and AssetType objects.

CORBA

Common Object Request Broker Architecture. CORBA is a distributed-objects standard developed and defined by the Object Management Group (OMG). CORBA provides the mechanisms by which objects transparently make requests and receive responses, as defined by OMG's Object Request Broker (ORB). The CORBA ORB is an application framework in which objects can communicate with each other, even if they are written in different programming languages, are running on different platforms, reside at different locations, or were developed by different vendors.

credentials

User name and passwords pairs used for user authentication when logging in to a networked system.

data binding

The process by which a data source is linked to a Web page. You can present, manipulate, and update data on the client by linking data to HTML tags. Data binding is based on a component architecture consisting of three major pieces: DSO data consumers, the binding agent, and the table repetition agent. The DSO provides the data to the page, data-consuming HTML elements display the data, and the binding and table repetition agents ensure that both the provider and the consumer are synchronized. Data binding, combined with HTML 4.0 and the Document Object Model, is one of the contributing technologies to Dynamic HTML (DHTML).

Because the data binding is done on the client side, the data displayed on the Web page is kept separate from the HTML that displays the data. Data binding does this by treating HTML in a Web page as a template for data supplied by a data source object. Then, using the Dynamic HTML support, the data supplied by data objects is merged with the HTML template on the client, producing a complete HTML page.

data element

An element that contains no element references or code lists.

data mart

One or more databases designed to help managers make strategic decisions about their businesses. A data mart usually focuses on a particular subject or department rather than on an enterprise-wide application.

data store

A physical repository that resides on a database server.

data warehouse

A collection of data designed to help managers make strategic decisions about their business. A data warehouse contains a wide variety of data that presents a coherent picture of business conditions at a single point in time. Unlike a data mart, a data warehouse usually refers to a set of databases that are integrated across an entire enterprise.

database event

Database actions that change database states, that can be captured and re-created, and that cannot occur (or be recorded) simultaneously. These can include begins, rollbacks, or commits; inserts, updates, or deletes; blobs (Java object, image, or text); or stored procedure invocations that result in a change in the source database.

database middleware

Allows clients to invoke SQL-based services across multivendor databases. Database middleware is defined by de facto standards such as ODBC, DRDA, RDA, and so on.

DCE

Distributed Computing Environment. From the Open Software Foundation, DCE provides key distributed technologies such as RPC, distributed naming service, time synchronization service, distributed file system, and network security.

DCOM

Distributed COM. A protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. Based on the Open Software Foundation’s DCE-RPC specification, DCOM deploys across heterogeneous platforms and works with both Java applets and ActiveX components.

decryption

The process of unencoding information. Encryption and decryption allow communicating systems to disguise information they send.

digital certificate

An electronic document used to identify an individual, a server, a company, or some other entity and associate that identification with a public key. See PKI.

digital signatures

Digital signatures are created with a mathematical algorithm that generates a unique, fixed-length string of numbers from a text message; the result is called a hash or message digest. Digital signatures are used for tamper detection and nonrepudiation.

directory services

A way for clients to locate services. Usually contained in a single system image of available servers.

distinguished name

A name that uniquely identifies an entity. The distinguished name (DN) is embedded in a digital certificate. Enterprise Security identifies an entity by its DN for authentication to the system.

distributed database system

A computing system that contains a number of autonomous database management systems that are interconnected by a network and that cooperate with each other when performing data access and data capture tasks.

DOM

Document Object Model. The specification for how objects in a Web page (text, images, headers, links, and so on) are represented. The DOM defines what attributes are associated with each object, and how the objects and attributes can be manipulated. Dynamic HTML (DHTML) relies on the DOM to dynamically change the appearance of Web pages after they have been downloaded to a user’s browser.

DSO

A user who has the domain security officer role, and therefore has access to all the security objects in the security domain.

DTD

A document type definition is a specific definition that follows the rules of Standard Generalized Markup Language (SGML). A DTD accompanies a document and identifies what the codes (or markup) are that separate paragraphs, identify topic headings, and so on, and how each is to be processed. When a DTD is mailed with a document, any location that has a DTD “reader” (or “SGML compiler”) can process the document and display or print it as intended.

EAR

Enterprise archive file. Used to distribute a J2EE application. A standard JAR file with a “.ear” extension that may contain JAR and WAR files.

EDI

Electronic data interchange. The electronic communication of business transactions, such as orders, confirmations, and invoices between organizations.

EJB

Enterprise JavaBeans is an architecture for setting up program components, written in Java, that run in the server parts of a client/server. EJBs are specific Java components that meet the Java specifications for thread management, container support, and so on.

encryption

A process wherein a cryptographic algorithm is used to encode information to safeguard it from anyone except the intended recipient. Encryption and decryption allow communicating systems to disguise information they send.

enterprise A reference to all aspects of a large business organization—from manufacturing to finance, marketing to human resources. This term can also refer to an organization plus its partners, vendors, suppliers, and customers.

EP An acronym for Enterprise Portal. Enterprise Portal integrates all aspects of an organization's IT infrastructure and offers customers, partners, vendors, and employees a broad array of resources and services, including personalized information, online purchasing, e-mail, forums, search engines, and product support.

event An event is a notification that occurs in response to some action. It can be a change in state or as a result of the user clicking or moving the mouse, pressing a keyboard key, or other actions that are focus-related, element-specific, or object-specific. Programmers write code that respond to these actions. An event can also be an object that is imported, passed between processors, and exported to an external database.

extensible Capable of accepting new, user-defined commands.

extranet A network that allows partial access to authorized outsiders via valid user names and passwords.

firewall A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

gateway A hardware and/or software setup that performs translations between disparate protocols.

hash Also called a message digest, the hash is a unique, fixed-length string of numbers generated by a mathematical algorithm from a text message. The result is call a digital signature.

high availability The ability of a computer to stay up and running most of the time. Also, the ability to perform most administration tasks with users still connected and working with the data in the database. See also continuous availability.

HTTP HyperText Transport (or Transfer) Protocol is the set of rules that governs the exchange of text, graphic, sound, and video files on the World Wide Web.

HTTPS The secure version of HTTP.

IDL An interface definition language allows a program or object written in one language to communicate with another program written in an unknown language. For example, an Object Request Broker (ORB) uses an interface definition language to broker communication between two object programs.

IIOP Internet-Interoperable-ORB-Protocol is an object-oriented protocol that allows distributed programs written in different programming languages to communicate over the Internet. IIOP is the transport protocol for CORBA.

IIOPS The secure version of IIOP.

Internet A global network connecting millions of computers.

intranet A private network within an organization.

ISP An acronym for Internet service provider, which is a company that provides access to the Internet to companies or individual users for a monthly fee.

J2EE Sun software: Java 2 platform, Enterprise Edition.

Java Developed by Sun Microsystems, Java is an object-oriented programming language, similar to C++. Java-based applications, or applets, can be quickly downloaded from a Web site and run using a Java-compatible Web browser such as Microsoft Internet Explorer or Netscape Navigator. Java applets are the most widespread use of Java on the Web.

Java programs or source code files (.java) are compiled into a format known as bytecode files (.class). These files, once complied, can be executed by a Java interpreter. Most operating systems, including Linux, Macintosh, UNIX, and Windows have Java interpreters and runtime environments known as Java virtual machines.

JDBC JDBC is a data access interface based on ODBC and used with the Java programming language.

LDAP Lightweight Directory Access Protocol. LDAP is a software protocol that allows anyone to locate organizations, individuals, and other resources (files, devices, and so on) on the Internet or on a corporate intranet.

load balancing The even distribution of processing and communications activity across a computer network so no single device is overwhelmed.

mapper Sybase product that enables you to create transformations that transform XML documents based on one vocabulary into XML documents based on a different XML vocabulary.

marshalling, unmarshalling Data marshalling converts native datatypes into an intermediate data stream that can pass safely between process boundaries. Unmarshalling converts it from the intermediate data stream to a datatype required at the other end of a communication.

message A string of bytes that has meaning to the applications that use it. Messages are used for transferring information from one application to another between components in a single application. The applications can be running on the same platform or on different platforms.

message broker An intelligent intermediary that directs the flow of messages between applications. Message brokers provide a flexible communications backbone and provide such services as data transformation, message routing and message warehousing.

message digest Also called a hash; a unique, fixed-length string of numbers generated by a mathematical algorithm from a text message. The result is a digital signature.

message queuing A form of communication between programs. Application data is combined with a header (information about the data) to form a message. Messages are stored in queues, which can be buffered or persistent (see buffered queue and persistent queue). Message queueing is an asynchronous communications style and provides a loosely coupled exchange across multiple operating systems.

message routing A process that routes messages to applications based on business rules. A particular message may be directed based on its subject or actual content.

message warehousing A central repository for temporarily storing messages for analysis or transmission.

metadata Data that describes other data. Any file or database that holds information about another database’s structure, attributes, processing, or changes.

method In object-oriented programming, a procedure that is executed when an object receives a message. A method is really the same as a procedure, function, or routine in procedural programming languages. The only difference is that in object-oriented programming, a method is always associated with a class.

middleware Software that facilitates the communication between two applications. Middleware provides an API through which applications invoke services and it controls the transmission of the data exchange over the network. There are three basic types: communications middleware, database middleware, and systems middleware.

migration

When referring to data, migration describes the process of translating data from one format to another. When referring to a computing environment, migration describes the process of moving from one type of hardware or software to another.

nonrepudiation

Digital signatures provide nonrepudiation, that is, senders cannot deny, or repudiate, that they sent a message, because their private key encrypted the message.

object middleware

Allows clients to invoke methods or objects that reside on a remote server. This middleware revolves around OMG's CORBA and Microsoft's DCOM.

ODBC

Open Database Connectivity. ODBC is a Windows standard API that is used for SQL communication to connect applications to a variety of data sources. Access is generally provided through the Control Panel, where data source names (DSNs) can be assigned to use specific ODBC drivers.

ORB

Object Request Broker. Software that allows objects to dynamically discover each other and interact across machines, operating systems, and networks.

persistent queue

A message queue that resides on a permanent device, such as a disk, and can be recovered in case of system failure.

PKI

A public-key infrastructure allows users of an insecure public network, such as the Internet, to securely exchange data and money using a public and a private cryptographic key pair obtained and shared through a trusted authority.

portal

A Web site that offers users access to a broad array of resources and services, such as e-mail, forums, search engines, and online shopping malls.

private key

Part of the larger public-key infrastructure, a private key is kept secret and the public key is published. Typically, you use the private key to encrypt data before sending it over the Internet, and the recipient decrypts data with your public key.

PSO

A user who has the PortalSecOfficer role, and therefore has access to all security objects in the default security domain.

public key

Part of the larger public-key infrastructure, a public key is published, and the corresponding private key is kept secret. Typically, the public key is used to decrypt information that is encrypted with a private key before being sent over the Internet. See PKI.

public-key cryptography

Public-key cryptography consists of encryption and decryption, digital signatures, keys, and digital certificates. It is part of the larger public key infrastructure. See PKI.

publish

Make an event available to an external application by placing it on the external application’s queue.

publish/subscribe

A style of interapplication communications. Publishers can broadcast data to a community of information users or subscribers, which have issued the type of information they want to receive (normally defining topics or subjects of interest). An application or user can be both a publisher and subscriber.

queue

A list constructed and maintained so that the next data element to be retrieved is the one stored first.

For example, one application can put a message on a queue, and another application can retrieve the message from the same queue.

real time, real-time

The immediate processing of input, such as the ability of a computer to respond or process information immediately with no interruption.

replication

The process of copying data to remote locations. The copied (replicated) data is then kept synchronized with the primary data. Data replication is distinct from data distribution. Replicated data is stored copies of data at particular sites throughout a system and is not necessarily distributed data.

request/response

See publish/subscribe.

RMI

Remote Method Invocation is a set of protocols being developed by Sun’s Java Software division that enables Java objects to communicate remotely with other Java objects. RMI is a relatively simple protocol, but unlike more complex protocols such as CORBA and DCOM, it works only with Java objects. CORBA and DCOM are designed to support objects created in any language.

RPC

Remote procedure call. A form of application-to-application communication that hides the intricacies of the network by using an ordinary procedure call mechanism.

scalability

The ability of an information system to provide high performance as greater demands are placed upon it, through the addition of extra computing power.

Security Officer (PSO)

The Security Officer role is predefined in the Access Control Database. The Security Officer manages Enterprise Portal and EAServer security using the Enterprise Security Manager, a graphics-based administration tool.

The default PSO role has all permissions and is assigned to a default login. You can use this to initially log in and create user name and password combinations for security officers, administrator, and grant the appropriate roles. You can then invalidate or delete the default login to secure the product against intruders who possess the default login information.

server

A computer or software package that provides specific capabilities to client software running on other computers.

servlet

A servlet is a small, persistent, low-level program that runs on a server. The term was coined in the context of the Java applet, a small program that is sent as a separate file along with a Web (HTML) page.

Some programs that access databases based on user input must be on the server. These programs are most often implemented using a Common Gateway Interface (CGI) application. However, if a Java virtual machine is running in the server, servlets can be implemented in Java. A Java servlet can execute more quickly than a CGI application. Instead of creating a separate program process, each user request is invoked as a thread in a single daemon process, so the system overhead for each request is slight.

SNMP

Simple Network Management Protocol governs network management and how network devices and their functions are monitored. It is not necessarily limited to TCP/IP networks.

SOAP

Simple Object Access Protocol. SOAP provides a way for applications to communicate with each other over the Internet, independent of platform. Remote objects can give a program almost unlimited power over the Internet, but most firewalls block non-HTTP requests. SOAP, an XML-based protocol, gets around this limitation to provide intraprocess communication across machines.

In Enterprise Portal, the implementation of SOAP allows businesses to expose corporate software functionality to their customers with minimal firewall constraints, platform dependencies or complex development implementations involving DCOM or CORBA.

SOAP was developed by Microsoft, DevelopMentor, and Userland Software and has been proposed to the Internet Engineering Task Force (IETF) as a standard.

sockets

A portable standard for network application providers on TCP/IP networks.

SPI Service Provider Interface, the programming interface for developing Windows drivers to provide common access to services. An application (query, word processor, e-mail program, and so on) is written to a particular interface, such as ODBC or MAPI, and the developer of the service software (database manager, document manager, print spooler, and so on) writes to the SPI for that class of service

SQL Structured Query Language. The language used to process data in a relational database; supported by all major database management systems.

SSL Secure Sockets Layer. A set of rules that govern server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information.

SSL handshake A series of I/O round trips between a server and a client to negotiate and agree upon a secure encrypted session.

SSO An acronym for single sign-on. Single sign-on features allow a client to request access to protected assets within a portal without having to resubmit credentials or certificates for authentication.

stored procedure A program that creates a named collection of SQL or other procedural statements and logic that is compiled, verified, and stored in a server database.

systems middleware Software that provides value-add services as well as interprogram communications. An example is transaction processing monitors which are required to control local resources and also cooperate with other resource managers to access nonlocal resources.

TCP/IP Transmission Control Protocol/Internet Protocol—the network protocol for the Internet that runs on virtually every operating system. IP is the network layer and TCP is the transport layer. TCP/IP is the primary transport protocol used in client/server computing, and is the protocol that governs the transmission of data over the Internet.

thin client Thin client refers to the net PC or the network computer, personal computers for businesses that are centrally-managed, configured with only essential equipment, and do not have CD players, diskette drives, or expansion. Since the idea is to limit such computers to essential applications, they tend to remain “thin” in terms of the client applications they include.

trade relationship Business relationship between two trading partners in which EDI and XML documents are exchanged.

trading partner

Organization with which you trade (for example, a supplier or customer). Trading partners send and receive EDI and XML documents.

transaction log

The log of transactions kept by a database server.

transform

Process in which you convert a source document based on one XML vocabulary into a target document based on another XML vocabulary.

Transport Layer Security

A security protocol from the Internet Engineering Task Force (IETF) that is a merger of SSL and other protocols. TLS is backward compatible with SSL and uses Triple DES encryption.

trigger

A stored procedure that is automatically invoked on the basis of data-related events.

URI

Uniform Resource Identifier. A URI is compact string of characters for identifying an abstract or physical resource and provides a simple and extensible means for identifying resources. An example of a URI is a URL.

URL

Uniform Resource Locator. A subset of a URI, a URL is like a networked extension of the standard file name concept: you can point to a file in a directory, but that file and directory can exist on any machine on the network. They can also be served by any of several different methods. URLs can also point to queries, documents stored deep within databases, and so on.

WAR

Web application archive file. Used to distribute Web applications; it includes a deployment descriptor and Web components, and may contain server-side utility classes, HTML, image and sound files, applets, and client-side utility classes.

workflow

Software used to automatically route events or work-items from one user or program to another. Workflow is synonymous with process flow, although traditionally has been used in the context of person-to-person information flows.

XML

eXtensible Markup Language—a simplified subset of Standard Generalized Markup Language (SGML) provides a file format for representing data, a method for describing data structure, and a mechanism for extending and annotating HTML with semantic information.

As a universal data format, XML provides a standard for the server-to-server transfer of different types of structured data so that the information can be decoded, manipulated, and displayed consistently and correctly. In addition, it enables the development of three-tier Web applications, acting as the data transfer format between the middle-tier Web server and the client.

Index