IBM Security Identity Manager
Version 6.0.0.18

Security Topics

IBM Security Identity Manager
Version 6.0.0.18

Security Topics

Table of contents

Table list v

Chapter 1. WebSphere security and IBM Security Identity Manager 1

Chapter 2. External user registry for authentication 3

Chapter 3. Secure environment practices 5

Chapter 4. Disabling the HTTP port . . . 7

Chapter 5. Secure sockets layer communication 9
SSL terminology 9
One-way and two-way SSL authentication 10
SSL in a clustered environment 11
SSL implementations 12

Chapter 6. Certificate file types 13

Chapter 7. Securing of communication with adapters 15

Chapter 8. Securing of communication with custom applications 17

Chapter 9. Secure communication with supported middleware 19
Example SSL configurations 19
Preparation for SSL configuration 20
Creating a certificate 20
Configuring SSL for the database server 22
Configuring SSL for the directory server 22
Configuring SSL on the WebSphere Application Server 23
Configuring the IBM Security Identity Manager Server 25
Testing SSL communication between servers . . . 26
Configuration of the HTTP server for additional security and performance 28
SSL for the IBM HTTP server and WebSphere Application Server plug-in 28
Configuring SSL for the IBM HTTP server . . . 28
Configuring SSL for the plug-in 29

Chapter 10. Configuration of single sign-on 31
Configuration of IBM Security Identity Manager for single sign-on with WebSphere Trust Association Interceptor and IBM Security Access Manager WebSEAL 31
Account mapping 32
Changing the logoff page 33
Creating a user in IBM Security Access Manager that WebSEAL uses to connect to the backend server 35
Defining IBM Security Access Manager Accounts 35
Defining IBM Security Access Manager groups 37
Adding IBM Security Access Manager user account to a group 37
Defining a junction that points to IBM Security Identity Manager Server 38
Defining IBM Security Access Manager ACLs . . 40
Granting access to the IBM Security Access Manager ACLs 40
Associating the WebSEAL junction to the ACLs 41
Configuring WebSphere Application Server to point to IBM Security Access Manager 42
Configuring the Trust Association Interceptor . . 45
Configuring IBM Security Identity Manager to use single sign-on 46
Configuring WebSEAL. 46
IBM Security Identity Manager web services in a single sign-on environment 47
Installing on a system where the IBM Security Identity Manager is installed 48
Installing on a separate system than where the IBM Security Identity Manager is installed . . . 50
Starting the SSO application 53
Testing the SSO application 53
Building the SSO application 53
Preparing the WebSphere Application Server . . 55
Accessing IBM Security Identity Manager consoles 56
Frequently used commands to configure single sign-on 57

Chapter 11. Security layer configuration around the data model and reports . . 61
Authentication and authorization for IBM Cognos reports 61
User authentication setup by using LDAP 61
Configuring an LDAP Namespace for IBM Directory Server 61
Creating users in an LDAP 63
Access control definition for the reports and reporting packages 64
Restricting administration access and adding an LDAP user to system administrator role 64
Creating a role and adding LDAP users as members 65
Defining an access to the report by using a role 66
Defining an access to the reporting package by using a role 66
References for IBM Cognos report security configuration 67

Chapter 12. Setting the session timeout interval for IBM Security Identity Manager 69

Chapter 13. Setting the session timeout interval for the IBM Security Identity Manager Service Center 71

Index 73

Table list

1. Practices for a secure IBM Security Identity Manager environment 5
2. Example SSL configurations 19
3. Files in the /certs directory 21
4. Custom properties 25
5. Logoff pages 34
6. LDAP advanced mapping values 62

Chapter 1. WebSphere security and IBM Security Identity Manager

IBM® Security Identity Manager uses WebSphere® Application Server security.

IBM Security Identity Manager uses WebSphere security to enforce authentication and role-based authorization. WebSphere security includes administrative security. WebSphere administrative security must be enabled before IBM Security Identity Manager is installed. WebSphere security supports security domains, which can be used to define a security scope that is not global, but applicable to a specific application. WebSphere also supports application security. IBM Security Identity Manager requires WebSphere application security. You can enable application security at the security domain level, or you can enable it at the global security level.

When you install IBM Security Identity Manager, you select either the default custom registry that is provided with IBM Security Identity Manager, or you select an external user registry. If you choose the default custom registry, the installation program automatically creates a security domain that has application security enabled. If you select an external user registry, you must manually enable application security for the security domain that IBM Security Identity Manager uses.

The external user registry can operate at the global security level, or can be part of a specific security domain. If you select an external user registry that is used for global security, then you must enable application security for global security. If you select an external user registry that is associated with a security domain, then you must enable application security for that security domain.

IBM Security Identity Manager does not require Java™ 2 Security, but you can optionally enable it. You can turn on Java 2 security in order to address system resource usage. System resources include writing to the file system, listening on a socket, and calls to APIs. Java 2 security is configured in a was.policy file.

For more information about how to configure WebSphere Application Server security for an IBM Security Identity Manager deployment, see the topic "WebSphere security configuration" in the *IBM Security Identity Manager Installation Guide*.

For information on using an external user registry, see Chapter 2, “External user registry for authentication,” on page 3.

Chapter 2. External user registry for authentication

You can choose to use an external user registry instead of the default custom registry.

IBM Security Identity Manager provides a default custom registry. You do not have to use this registry for authentication. You can choose to use an external registry. An external user registry is any other registry that can be configured with WebSphere Application Server. You can use an existing registry or configure a new one.

The IBM Security Identity Manager installation program prompts you whether you want to use the custom registry.

- If you use the custom registry, the IBM Security Identity Manager installation program programmatically creates a security domain, enables application security, and configures it to the IBM Security Identity Manager custom registry.
- If you use an external registry, you must manually configure application security.

If you want to use an external user registry, review how IBM Security Identity Manager uses WebSphere Application Server security. See the following documentation:

- Chapter 1, “WebSphere security and IBM Security Identity Manager,” on page 1
- "WebSphere security configuration" in the *IBM Security Identity Manager Installation Guide.*

To use an external user registry, you must complete specific configuration tasks.

The tasks are specific to either Configuration of a new installation, Reconfiguration of an existing installation, or Upgrade from a previous version.

Configuration of a new installation

The IBM Security Identity Manager documentation describes how to configure external user registry for a new installation.

- Before you install IBM Security Identity Manager, you must configure the existing external user registry. You must also configure a WebSphere security domain, unless you previously configured one for WebSphere Application Server with the user registry, or you previously configured WebSphere Application Server global security.

  See the installation and configuration instructions in the topic "Preinstall configuration for authentication with an external user registry" in the *IBM Security Identity Manager Installation Guide.*

  Note: If you do not have an existing user registry, you must create and configure one. See the topic "User registry configuration for external user registry" in the *IBM Security Identity Manager Installation Guide*.
- During the IBM Security Identity Manager installation, you must choose *not* to use the custom registry.

  In the "Installation of IBM Security Identity Manager" chapter of the *IBM Security Identity Manager Installation Guide*, see the topic that fits your scenario:

  - "Completing the installation wizard pages for WebSphere single server deployments"
  - "Completing the installation wizard pages for WebSphere cluster deployments"
- After installing IBM Security Identity Manager, complete the instructions in the topic "Postinstall configuration of an external user registry for authentication" in the *IBM Security Identity Manager Installation Guide*.

  Note: To complete the configuration for external user registry, you must modify the value for the WebSphere account repository attribute on the Service Information page for the ITIM Service. Be sure to complete the instructions in the topic "Configuring the WebSphere account repository setting" in the *IBM Security Identity Manager Installation Guide*.

Reconfiguration of an existing installation

If you installed IBM Security Identity Manager to use the default custom registry and want to switch to an external user registry, you must reconfigure middleware to support authentication with an external user registry. You must add required users to the external user registry and reconfigure the WebSphere security domain.

Follow the instructions in the topic "Reconfiguration for authentication with an external user registry" in the *IBM Security Identity Manager Installation Guide*.

Upgrade from a previous version

If you upgrade IBM Security Identity Manager from a previous release, and want to use an external user registry, you must add the new attribute errepositoryservice to the ITIM Service form.

Be sure to follow the upgrade instructions in the topic "Processes and settings that are not preserved, or require manual upgrade" in the *IBM Security Identity Manager Installation Guide*.

External user registry example deployment

The IBM Security Identity Manager product distribution includes three documents that describe an example deployment with an external user registry. The following documents are included in the directory *ISIM_HOME*`/extensions/6.0/doc/authentication/`:

- *Example installation of IBM Security Identity Manager with an external LDAP user registry*

  An example configuration, including screen captures of middleware configuration.
- *Example reconfiguration of IBM Security Identity Manager to use an external LDAP user registry*

  An example reconfiguration, including screen captures of middleware configuration.
- *Configuring and using IBM Security Identity Manager with an external user registry*

  Configuration tips for best practice.

Chapter 3. Secure environment practices

These practices can help ensure a secure IBM Security Identity Manager environment.

Table 1. Practices for a secure IBM Security Identity Manager environment

Given sensitive data in these areas	Ensure that these practices occur
Database data	Restrict operating system access to database files. Limit the privileges of the operating system accounts (administrative, root-privileged, or DBA) to the least privileges needed. Change the default passwords. Enforce periodic password changes. See the security information in the database documentation for more details.
Database logs	Restrict operating system access to log and trace files. Limit the privileges of the operating system accounts (administrative, root-privileged, or DBA) to the least privileges needed. Change the default passwords. Enforce periodic password changes. See the security information in the database documentation for more details.
Database backups	Store database backups at safe and secure locations. Guard against leaks or exposure of sensitive and confidential information. See the security and backup information in the database documentation for more details.
LDAP data	Securely handle any LDAP data that contains sensitive information. Sensitive information includes disabling anonymous read, enabling SSL, and restricting access to privileged and authorized operating system and application users. See the security information in the LDAP directory server documentation for more details.
LDAP logs	Restrict access to log files in the `log` directory of the directory server to privileged and authorized operating system and application users. This restriction is especially important if you enable audit logging for the directory server. See the security information in the directory server documentation for more details.
LDAP backups	If LDIF files contain sensitive information, store them safely and handle them securely.
IBM Security Identity Manager logs	If Security Identity Manager logs in the *path*`/ibm/tivo../../common/CTGIM` directory contain sensitive information, restrict access to them.
Directories under *ISIM_HOME*	If the data, configuration, and installation logs contain sensitive information, restrict access to the directories in *ISIM_HOME*.

Table 1. Practices for a secure IBM Security Identity Manager environment (continued)

Given sensitive data in these areas	Ensure that these practices occur
Network traffic	Restrict network traffic to what is required by the deployment. If you write your own application and use an IBM Security Identity Manager API to retrieve sensitive data, encrypt the data before you send it over the network.
WebSphere Application Server security	Enable security on WebSphere Application Server and disallow running WebSphere Application Server with a non-root account.

Chapter 4. Disabling the HTTP port

When IBM Security Identity Manager 6.0 application is deployed on the WebSphere Application Server, SSL and non-SSL connections are both enabled by default. For enhanced security, run the application only on an SSL connection.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Use this task to disable the default HTTP port (9080) by removing the non-SSL port from the virtual host that is configured with Security Identity Manager in the WebSphere Application Server.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Click **Environment** > **Virtual Hosts**.
3. Click **default_host**.
4. Click **Host Aliases** in the **Additional Properties** section.
5. Select the row of default HTTP port (9080) on which Security Identity Manager is deployed.
6. Click **Delete** to remove the port from the virtual host.
7. Click **Save** to save the changes to the master configuration.
8. Update the global web server plug-in configuration.
   a. Click **Environment** > **Update global Web server plug-in configuration**.
   b. Click **OK** to update the plug-in configuration file.
9. Restart the WebSphere Application Server.

Results

The port is removed from the “default_host” virtual host. The HTTP port is unusable for all the applications that are configured to use this “default_host” virtual host. Because Security Identity Manager is configured by default to use the "default_host" virtual host, the user cannot access the Security Identity Manager application through this non-SSL HTTP port. Access must be made through the SSL port.

Note:

For other methods and for cluster environments where load balancers are used, see the WebSphere Application Server or the IBM HTTP Server documentation.

Chapter 5. Secure sockets layer communication

The industry-standard Secure Sockets Layer (SSL) protocol uses signed digital certificates from a certificate authority (CA) for authentication. It provides secure communication in an IBM Security Identity Manager deployment.

SSL provides encryption of the data that is exchanged between the applications. An application that acts as an SSL server presents its credentials in a signed digital certificate to verify to an SSL client that it is the entity it claims to be. You can also configure an application that acts as an SSL server to require the application that acts as an SSL client to present its credentials in a certificate. This method completes a two-way exchange of certificates.

SSL terminology

These terms apply to Secure Sockets Layer communication for IBM Security Identity Manager.

SSL server
Listens for connection requests from SSL clients. For example, theIBM Security Directory Server might be an SSL server that listens for connection requests from the IBM Security Identity Manager Server and the WebSphere Application Server.

SSL client
Issues connection requests. For example, the computer on which the IBM Security Identity Manager Server and the WebSphere Application Server are installed is the SSL client, which issues connection requests to the IBM Security Directory Integrator.

Signed certificates
Is an industry-standard method of verifying the authenticity of an entity, such as a server, client, or application. Signed certificates are issued by a third-party certificate authority for a fee. Some utilities, such as the iKeyman utility, can also issue signed certificates. A certificate authority or CA certificate must be used to verify the origin of a signed digital certificate.

Signer certificates (certificate authority certificates)
Must be used to verify the origin of a signed digital certificate. When an application receives a signed certificate from another application, it uses a CA certificate to verify the originator of the certificate. Many applications, such as web browsers, are configured with the CA certificates of well-known certificate authorities. This practice eliminates or reduces the task of distributing CA certificates throughout the security zones in a network.

Self-signed certificates
Contains information about the owner of the certificate and the owner of the signature. Basically, it is a signed certificate and CA certificate in one. If you choose to use self-signed certificates, you must extract the CA certificate from it in order to configure SSL.

SSL keystore
Is a key database file designated as a keystore. It contains the SSL certificate.

Note: The keystore and truststore can be the same physical file.

SSL truststore
Is a key database file designated as a truststore. The SSL truststore contains the list of signer certificates (CA certificates) that define which certificates the SSL protocol trusts. Only a certificate issued by one of these listed trusted signers is accepted.

Note: The truststore and keystore can be the same physical file.

One-way SSL authentication
Requires a keystore and certificate only on the SSL server side (such as the Security Directory Server) and a truststore only on the SSL client side (such as the Security Identity Manager Server).

Two-way SSL authentication (client-side authentication)
Requires a keystore with a certificate and a truststore that contains the signer of the certificate that issued the other certificate on both the SSL server and client.

One-way and two-way SSL authentication

Configuring communication between an SSL server and client can use one-way or two-way SSL authentication. For example, the SSL client is the computer on which the IBM Security Identity Manager Server is installed, and the SSL server is the IBM Security Directory Server.

One-way authentication creates a truststore on the client and a keystore on the server. In this example, CA certificate "A" exists in the truststore on the SSL client and also in the keystore on the SSL server.

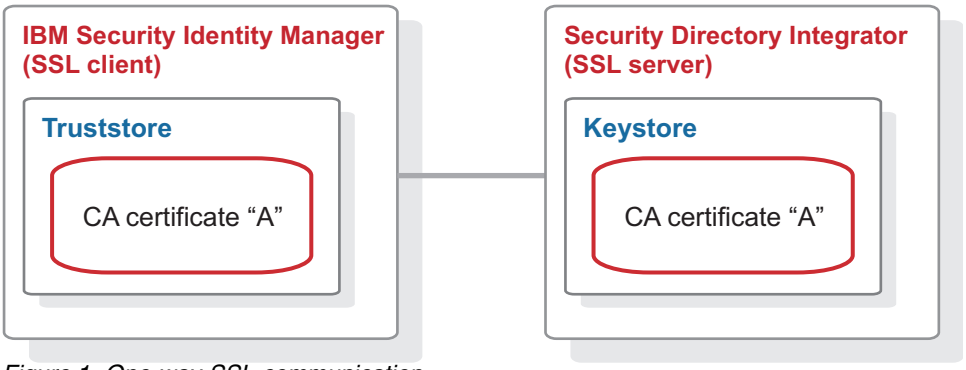

Figure 1. One-way SSL communication

Two-way authentication creates a truststore and a keystore on both the client and the server. In this example, there is a CA certificate "A" in the truststore and a CA certificate "B" in the keystore on both client and server.

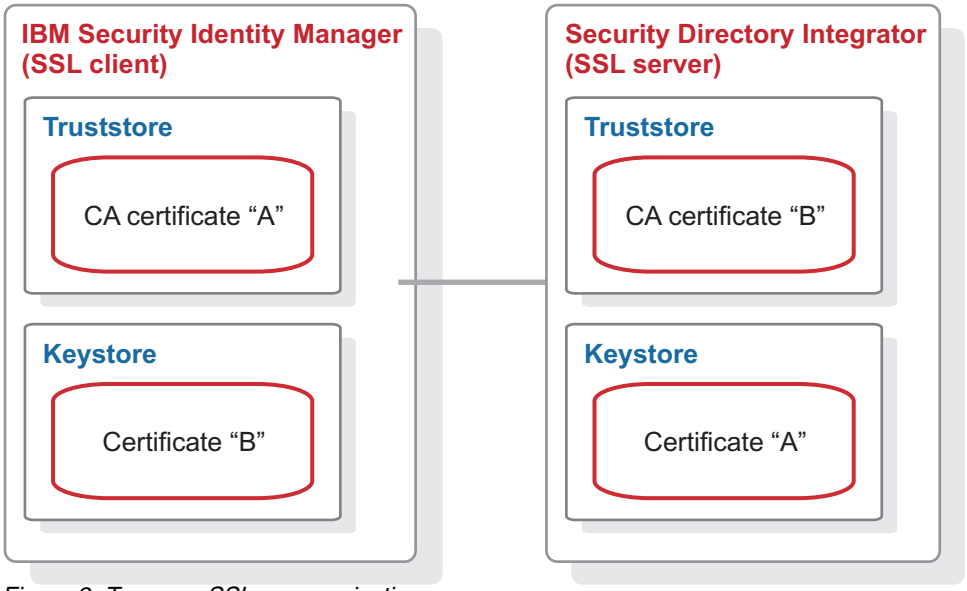

Figure 2. Two-way SSL communication

For more information about configuring SSL communication between the IBM Security Identity Manager Server and an IBM Security Identity Manager adapter, see the installation and configuration guide for the adapter.

SSL in a clustered environment

Communication between an SSL server and client can use one-way or two-way SSL authentication. For example, the SSL client is the computer on which the IBM Security Identity Manager Server is installed. The SSL server is the IBM Security Directory Server.

To configure one-way SSL authentication for a group of IBM Security Identity Manager servers in a WebSphere Application Server cluster, take one of the following actions:

- Import the appropriate CA certificates into the truststore used by each application server member of the cluster.
- Set up the truststore of one application server with CA certificates. Copy the contents of that truststore to the other application server members in the cluster.

To configure two-way SSL authentication for a cluster, you must configure each application server member to use separate truststores and keystores. The truststore can be the same common file with CA certificates. You can copy the file from one application server member to the truststores of the other application server members. You also copy files for one-way SSL authentication. Each keystore must be unique to each application server member and contain only the application server (client) certificate and private key.

Alternatively, you can implement a less secure configuration by setting up a common keystore with a single certificate and private key. For example, you might set up a less secure configuration to test connectivity. In this scenario, you:

1. Export the certificate and private key to a temporary file (for example, to a `PKCS12` formatted file).
2. Import that file into the keystore of each application server member in the cluster.

SSL implementations

IBM Security Identity Manager Server uses several implementations of the SSL protocol.

IBM Security Identity Manager Server uses these implementations of the SSL protocol:

IBM Global Security Toolkit (GSKit)
Used by the WebSphere Application Server, IBM Security Directory Server, and IBM Security Identity Manager adapters.

IBM Java Secure Socket Extension (JSSE)
Used by the IBM Security Identity Manager Server and by IBM Security Directory Integrator

OpenSSL
Used by IBM Security Identity Manager Server Adapter Development Kit (ADK)-based adapters.

Chapter 6. Certificate file types

Certificates and keys are stored in several types of files.

Files that store certificates and keys can have the following formats:

.pem A privacy-enhanced mail file with a file extension of `.pem`. It begins and ends with the following lines:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

A privacy-enhanced mail format supports multiple digital certificates, including a certificate chain. If your organization uses certificate chaining, use this format to create CA certificates.

.arm A file with an extension of `.arm` contains a base-64 encoded ASCII representation of a certificate. It includes its public key but not its private key. The IBM Key Management utility generates and uses an `.arm` format. Specify this format to extract a self-signed certificate from the computer on which it was generated to the computer that uses it as the CA certificate.

.der A file with an extension of `.der` contains binary data. This format can be used only for a single certificate, unlike a file with a privacy-enhanced mail format, which can contain multiple certificates. Specify this format to extract a self-signed certificate from the computer on which it was generated to the computer that uses it as the CA certificate.

.pfx (PKCS12)
A `PKCS12` file has an extension of `.pfx`. It contains a certificate (CA-issued certificate or self-signed certificate) and a corresponding private key. Use this format to transfer the contents of a keystore to a separate computer. For example, you can create and install a certificate and private key with the key management utility. You can then export them to a `PKCS12` file and import the file into another keystore. This format is also useful for converting from one type of SSL implementation to a different implementation. For example, you can create and export a `PKCS12` file with the IBM Key Management utility and then import the file on another computer with the **`OpenSSL CertTool`** utility.

Chapter 7. Securing of communication with adapters

The IBM Security Identity Manager Server uses either SSL or Secure Shell (SSH) communication to communicate securely with supported adapters.

Figure 3 illustrates how you can configure secure communication links.

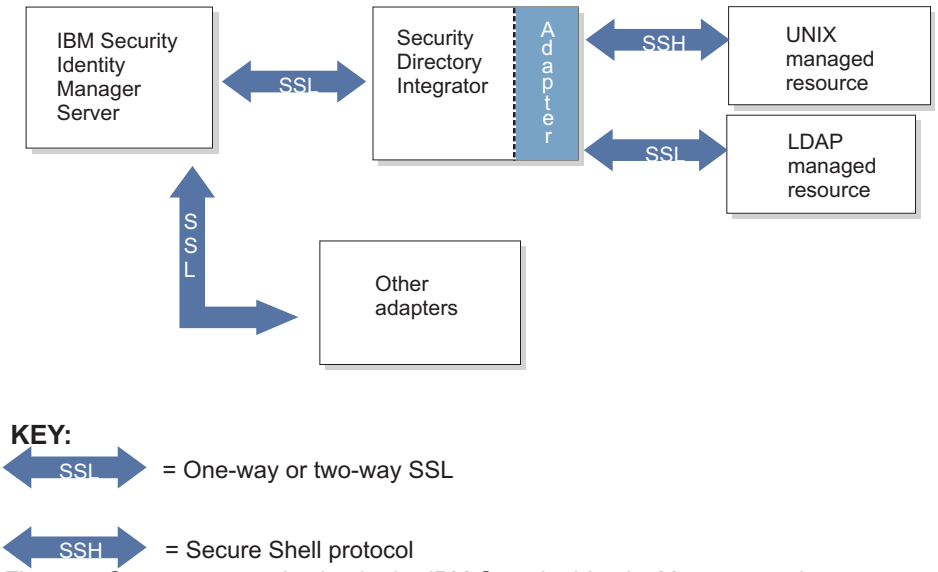

Figure 3. Secure communication in the IBM Security Identity Manager environment

Managed resources can communicate with the IBM Security Identity Manager adapters with the following protocols:

SSL Configures adapters, such as Windows Server Active Directory or Lotus Notes®, to use SSL authentication to communicate with the IBM Security Identity Manager Server. Not all adapters use the same SSL configuration. For more information, see the installation and configuration guide for the specific adapter.

Secure Shell (SSH)
Is used between the adapter and managed resource. The SSH protocol requires no configuration. The use of SSH between the adapter and managed resources cannot be disabled. However, configuration of SSH might be required on the managed resource. For more information, see the IBM Security Identity Manager adapter installation and configuration guides for UNIX and Linux.

Chapter 8. Securing of communication with custom applications

If you develop custom applications to access the IBM Security Identity Manager Server, these applications must adhere to the programming guidelines described in this section.

These guidelines ensure that:

- Security boundaries built into the IBM Security Identity Manager Server are observed strictly.
- Only authorized application programming interfaces (APIs) are used for communication between the server and custom applications.
- Appropriate roles are assigned to users and user groups that use custom applications to access IBM Security Identity Manager functions.

IBM Security Identity Manager shields its core functions with a layer of managed enterprise Java beans (EJBs). These EJBs are in an unprivileged layer of the IBM Security Identity Manager, which is illustrated in Figure 4.

When the IBM Security Identity Manager communicates with a client application, every managed EJB method takes a signed token from the caller. The token verifies the caller identity, except when the method does the authentication. The caller obtains this signed token after authentication with the IBM Security Identity Manager Server.

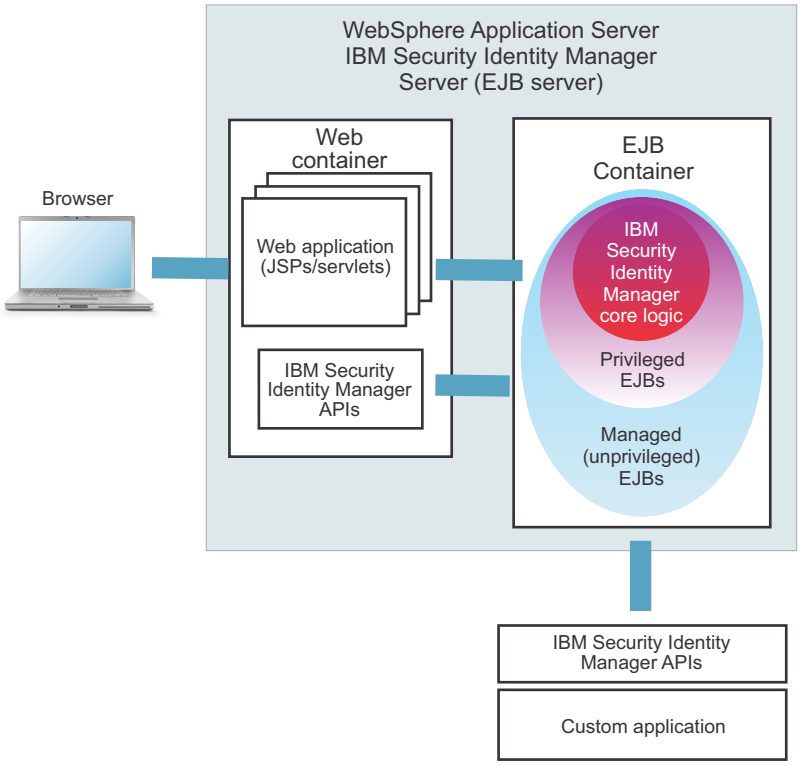

Figure 4. Security layers in IBM Security Identity Manager Server

The following types of custom applications can be created to communicate with the IBM Security Identity Manager Server:

Stand-alone Java client
Deployed as a WebSphere Application Server thin client.

Web application
Deployed outside of WebSphere Application Server. A web application can start only a specific subset of IBM Security Identity Manager Server APIs.

Enterprise application, same Java virtual machine (JVM)
Deployed in the same server instance (`enrole.ear`) as the IBM Security Identity Manager Server .

Enterprise application, separate JVM
Deployed on the same computer as the IBM Security Identity Manager Server, but runs as a separate JVM process.

Servlets
Deployed on a separate computer that runs WebSphere Application Server. Servlets are not deployed in the context of a web application.

When you develop custom applications to communicate with the IBM Security Identity Manager Server, use the following rules to ensure secure communication:

- Allow only published APIs to access the managed EJBs in the unprivileged area.
- Allow custom applications to use only the functions that the APIs provide.
- Ensure that the computer on which the IBM Security Identity Manager Server runs is always secure.

WebSphere Application Server uses roles to manage access to application components and other objects, including user and group names. Use the following guidelines for assigning roles in custom applications that interface with IBM Security Identity Manager Server.

`ITIM_SYSTEM`
This role is defined when the IBM Security Identity Manager Server is deployed into WebSphere Application Server. **`ITIM_SYSTEM`** is used by IBM Security Identity Manager Server components. It is authorized to call all EJB methods in both privileged and unprivileged layers. Do not assign any principal names or user IDs to this role without prior consultation with an IBM representative.

`ITIM_CLIENT`
This role is authorized to call only managed EJB methods in the unprivileged layer. Map to this role the users, and user group names, and other principals that perform less restricted tasks in the IBM Security Identity Manager Server.

Chapter 9. Secure communication with supported middleware

The IBM Security Identity Manager Server uses SSL for secure communication with supported middleware, such as a directory server.

Your configuration might be similar to the cluster configuration example in Figure 5.

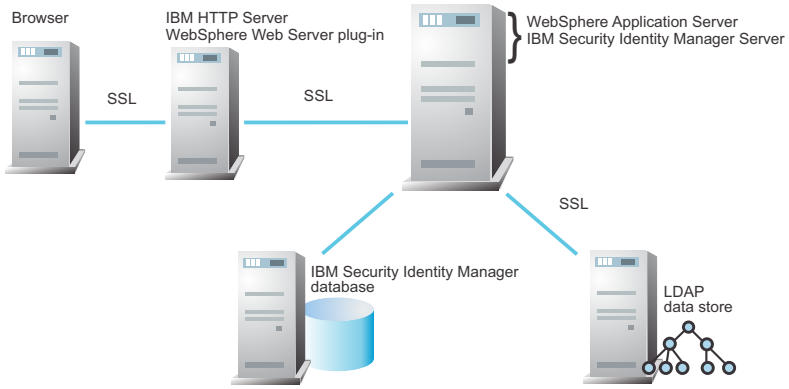

Figure 5. Initial configuration

After initial installation, you might configure secure communication links between IBM Security Identity Manager Server and these applications:

- Database server
- Directory server
- HTTP server
- Web browser
- Other supported middleware

Example SSL configurations

Example SSL configurations include secure communication between IBM Security Identity Manager Server and the directory server and between an HTTP server and a web browser.

In Table 2, the first application is the SSL client, and the second application is the SSL server:

Table 2. Example SSL configurations

SSL client	SSL server	One-way SSL	Two-way SSL
IBM Security Identity Manager Server	LDAP directory server	✓	
HTTP server (IBM HTTP Server)	IBM Security Identity Manager Server	✓	✓
Web browser	IBM Security Identity Manager Server	✓	

Your site might require additional configuration for SSL authentication with the IBM Security Identity Manager Server.

Preparation for SSL configuration

Before you configure SSL for secure communication, install and configure IBM Security Identity Manager Server. Then, locate IBM Global Security Kit (**GSKit**) to generate certificates.

Complete these tasks:

1. Install and configure the IBM Security Identity Manager Server and required supported middleware, including the directory server. This example assumes that a cluster configuration exists and that the directory server is on a separate computer.
2. Ensure that the initial configuration is running correctly. For more information, see *IBM Security Identity Manager Configuration Guide*.
3. Locate the IBM Global Security Kit (**GSKit**), which is included in the IBM Security Directory Server that the initial configuration installs. For example, locate the `/path/local/ibm/gsk7/bin` directory on the computer that has the Security Directory Server, where *path* is a value such as `usr`.

   The `GSKit` package provides the iKeyman key management utility, `gsk7ikm`. Use the utility to create key databases, public-private key pairs, and certificate requests. The following steps assume that you use the iKeyman utility to create self-signed certificates for secure communication. Alternatively, you can use the WebSphere Application Server administrative console to create a self-signed certificate.

   A self-signed digital certificate is a temporary digital certificate that you issue to yourself with yourself as the certificate authority (CA). When you complete testing, replace the self-signed certificate with a certificate signed by a CA certificate from a well-known certificate authority.

Creating a certificate

Use the iKeyman utility to create a self-signed certificate and extract the certificate to make it available for secure communication.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The iKeyman utility is in the IBM Security Directory Server.

Procedure

1. Start the iKeyman utility. For example, type the `gsk7ikm` command in the `/usr/local/ibm/gsk7/bin` directory
2. If the iKeyman utility cannot locate Java, run this command: **export JAVA_HOME=opt/IBM/ldapv6.1/java/jre**
3. On the IBM Key Management page, select **Key Database File** > **Open** > **New**.
4. Select a default database type of CMS.

5. In the **File Name** field, type a name for the CMS key database file. For example, type: `LDAPSERVER_TEST1234.kbd`

   For example, the value specifies *application_serverhostname* where *application* is the directory server, and *serverhostname* is the computer that has the directory server.
6. In the Location field, specify a location to store the key database file. For example, type `/certs`.
7. Click **OK**.
8. On the Password menu:
   a. Type and then confirm a password, such as `Pa$$word1`.
   b. Specify the highest password strength possible.
   c. Specify **Stash the password to a file?**.
   d. Click **OK**.
9. Select **Create** > **New Self Signed Certificate** and specify a label that matches the CMS key database file name, such as `LDAPSERVER_TEST1234.`

   This example uses the same name (`LDAPSERVER_TEST1234`) for both the certificate name and the key database file that contains the certificate.
10. Type `IBM` in the **Organization** field, accept the remaining field default values, and click **OK**. A self-signed certificate, including public and private keys, now exists.
11. For subsequent use with clients, extract the contents of the certificate into an ASCII Base-64 Encoded file. Complete these steps:
    a. Select **Extract Certificate**.
    b. Specify a data type of Binary DER Data.

       A file with an extension of `.der` contains binary data. This format can be used only for a single certificate. Specify this format to extract a self-signed certificate.
    c. Specify the name of the certificate file name you created, such as `LDAPSERVER_TEST1234.der`.
    d. Specify a location, such as `/certs`, in which you previously stored the key database file
    e. Click **OK**.
12. Verify that the `/certs` directory contains the following files:

Table 3. Files in the `/certs` directory

File	Description
`LDAPSERVER_TEST1234.crl`	Not used in this example.
`LDAPSERVER_TEST1234.der`	The certificate.
`LDAPSERVER_TEST1234.kbd`	Key database file that has the certificate.
`LDAPSERVER_TEST1234.rdb`	Not used in this example.
`LDAPSERVER_TEST1234.sth`	Stash file that has the password

Note: If you use an existing or newly acquired certificate from a CA, copy it to the `/certs` directory on root file system of the directory server.

Alternatively, you can use the WebSphere Application Server administrative console to create a self-signed certificate:

a. Select **Security** > **SSL certificate and key management** > **Manage endpoint security configurations** > **{Inbound | Outbound}** > **ssl_configuration** > **Key stores and certificates** > **[keystore]**. From Additional Properties, click **Personal certificates**.
b. Click **Create a self-signed certificate**

What to do next

For more information, see:

- Topics on securing directory communications in the *IBM Security Directory Server Administration Guide* at

  http://www.ibm.com/support/knowledgecenter/SSVJJU/welcome
- *IBM Global Security Kit Secure Sockets Layer Introduction and iKeyman User's Guide* at

  http://www.ibm.com/support/knowledgecenter/SSPREK_6.1.1/com.ibm.itame.doc_6.1.1/ss7cumst.htm?cp=SSPREK_6.1.1%2F0-3-5

Configuring SSL for the database server

Configure the database server to use SSL for secure communications.

About this task

For information about configuring SSL for the database server, see Security configuration of the database server.

Configuring SSL for the directory server

Use an LDIF file to configure SSL on the directory server and to specify a secure port.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. If the directory server is not running, start the server. For example, on UNIX, type this command:

   ```
   /opt/IBM/ldap/V6.1/sbin/ibmslapd -I itimldap
   ```

   Where *-I* specifies the instance.
2. Create an LDIF file, such as `ssl.ldif`, with the following data:

   ```
   dn: cn=SSL,cn=Configuration
   changetype: modify
   replace: ibm-slapdSslAuth
   ibm-slapdSslAuth: serverauth
   -
   replace: ibm-slapdSecurity
   ibm-slapdSecurity: sslonly
   -
   replace: ibm-slapdSslKeyDatabase
   ibm-slapdSslKeyDatabase: /certs/LDAPSERVER_TEST1234.kdb
   ```

Note: The empty lines that contain only the - (hyphen) character are required for LDIF file formatting.

To change the secured port from the default port number 636, add these additional lines:

```
replace: ibm-slapdSecurePort
ibm-slapdSecurePort: 637
```

3. Place the LDIF file in the following directory:

   `/opt/IBM/ldap/V6.1/bin`

4. Run the **idsldapmodify** command, which modifies the password policy by adding the LDIF file to the process.

   `idsldapmodify -D cn=root -w passwd -i ssl.ldif`

   -D Binds to the LDAP directory, which is `cn=root` in this example.

   -w Uses the *passwd* value, which is the directory server administrator password, as the password for authentication.

   -i Reads the entry modification information from an LDIF file instead of from standard input. In this example, the file is named `ssl.ldif`.

   A successful result produces a message similar to the following one:

   `Operation 0 modifying entry cn=SSL,cn=Configuration`

5. Test the directory server to confirm that it is listening on the default secure port 636. Follow these steps:

   a. Stop the directory server. Type `/opt/IBM/ldap/V6.1/sbin/ibmslapd -k:`.

   b. Start the directory server. Type `/opt/IBM/ldap/V6.1/sbin/ibmslapd -I itimldap`.

      Where **-I** specifies the instance.

   c. Determine whether the directory server is listening on port 636. For example, display statistics for the network interface with the directory server by typing `netstat -an |grep 636`.

      A return message that indicates the port is listening might be this example:

      ```
      tcp    0   0 9.42.62.72:636  0.0.0.0:*   LISTEN
      ```

Configuring SSL on the WebSphere Application Server

Set up the WebSphere Application Server to enable SSL communication between IBM Security Identity Manager and the directory server.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The topic guides you through the configuration of SSL by using the default `cacerts` file that is provided by the WebSphere Application Server Java SDK. Take note that this `cacerts` file is likely to be overwritten each time the Java SDK is upgraded or a fix pack is applied. To save your settings, backup the `cacerts` file before you upgrade the Java SDK and then restore the file after upgrade is completed. Otherwise, you can store the certificates in a different keystore file that is not the default `cacerts` file.

Procedure

1. Manually copy all files from the /certs/*LDAPSERVER_TEST1234* directory on the directory server to an identical /certs directory on the WebSphere Application Server root.
2. On the WebSphere Application Server, change to the /opt/IBM/WebSphere/AppServer/bin directory.
3. Start the iKeyman utility. For example, on UNIX, issue the command
   ikeyman.sh
4. Click **New Key Database File**. In the New page, complete the following steps.
   a. In the **Key database type** field, select **JKS**.
   b. In the **File Name** field, browse for a file name such as /opt/IBM/WebSphere/AppServer/java/jre/lib/security/cacerts. The cacerts file is installed by default with the WebSphere Application Server.
   c. In the **Location** field, type /opt/IBM/WebSphere/AppServer/java/jre/lib/security/ and click **OK**.
5. On the **Password** menu, type changeit, which is the default password, and click **OK**.
6. On the **Replace existing file** menu, select **Yes**.
7. Type and confirm the password Pa$$word1. Select the highest possible password strength, and click **OK**.
8. Click **Add** to add a certificate from a file.
9. On the **Add CA's Certificate from a File** menu, complete these steps and click **OK**.
   a. Specify a data type of Binary DER Data.
   b. Browse for the certificate name, such as *LDAPSERVER_TEST1234.der* in this example.
   c. Type a value for the location, such as /certs/.
10. Type a label for the certificate, such as ITIM2LDAP, which is a convenience for remembering the purpose of the certificate on the WebSphere Application Server. Click **OK**.
11. Examine the list of signer certificates to ensure that it contains the ITIM2LDAP certificate.
12. Exit the iKeyman utility.
13. Start the WebSphere Application Server administrative console to enable SSL communication between IBM Security Identity Manager and the directory server.
    a. Open the WebSphere Application Server administrative console. For example, type
       https://test1234:9043/ibm/console/logon.jsp
    b. Log on as the WebSphere Application Server administrator.
    c. On each cluster member server record, click **Servers** > **Application servers** > **Server1** > **Java and Process Management** > **Process Definition** > **Java Virtual Machine** > **Custom Properties**.
    d. In the Applications Servers page, select **New** to specify these custom properties:

Table 4. Custom properties

Name	Value	Description
javax.net.ssl.trustStore	/opt/IBM/WebSphere/AppServer/java/jre/lib/security/cacerts	Directory server certificate truststore for IBM Security Identity Manager
javax.net.ssl.trustStorePassword	Pa$$word1	Password that you initially specified for the self-signed certificate.
javax.net.ssl.trustStoreType	jks	Default keystore format available with Java VM

e. Click **Save**.

14. From the top-level menu of the **Appliance Dashboard**, click **Configure** > **Advanced Configuration** > **Application Server Certificate Management** to display the Application Server SSL Certificate page. The Application Server SSL Certificate page displays the certificate details.

Configuring the IBM Security Identity Manager Server

Configure the IBM Security Identity Manager server to communicate with the computer and port on which the directory server listens for secure communication.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. On the computer that has the Security Identity Manager Server, edit the *java.naming.provider.url* property that specifies the LDAP connection.
   a. In the /opt/IBM/isim/data directory, edit the enRoleLDAPConnection.properties file.
   b. In the enRoleLDAPConnection.properties file, change the *java.naming.provider.url* property to specify the computer and port number on which the directory server is listening. For example, type the host name and secure port of the computer that has the directory server:
      ```
      java.naming.provider.url=ldap://test1234:636
      ```
   c. Change the *java.naming.security.protocol* property to specify the SSL communication. For example:
      ```
      java.naming.security.protocol=ssl
      ```
      Alternatively, specify the protocol as ldaps instead of ldap in the *java.naming.security.protocol* property. For example:
      ```
      java.naming.provider.url=ldaps://test1234:636
      ```
      Note: The property change indicates the Security Identity Manager Server to use SSL for communicating with LDAP.
2. Save and close the enRoleLDAPConnection.properties file.
3. Restart the WebSphere Application Server.

4. From the top-level menu of the **Appliance Dashboard**, select **Configure** > **Advanced Configuration** > **Update Property** to display the Update Property page.

Testing SSL communication between servers

Test the SSL communication between IBM Security Identity Manager Server and IBM Security Directory Server.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Test that the Security Directory Server is listening. In the $*TDS_INSTALL_HOME*/bin directory on the computer where Security Directory Server is installed, type the following command on one line. For example:

   ```
   ldapsearch -b dc=com –K /certs/LDAPSERVER_TEST1234.kdb
   –p 636 -s base "objectclass=*"
   ```

   LDAPSERVER_TEST1234.kdb is the name of the key database.

   The result has entries for the top-level schema similar to the following entries:

   ```
   dc=com
   objectclass=top
   objectclass=domain
   dc=com
   ```

2. On the computer on which the Security Identity Manager Server is installed, test the SSL connections.
   a. Access the /opt/IBM/isim/bin directory.
   b. To provide additional security, copy ldapConfig.lax to ldapConfig.lax.backup and then edit the ldapConfig.lax file.

      Note: You use the ldapConfig.lax.backup file later in the process to hide the truststore password.

   c. The next statements provide the values of the truststore that the WebSphere Application Server uses for SSL communication between the Security Identity Manager Server and the Security Directory Server.

      In the file, add the following statement *as one line*:

      - For UNIX systems:

        ```
        lax.nl.java.option.additional=
        -Djavax.net.ssl.trustStoreType=jks
        -Djavax.net.ssl.trustStore=/certs
        -Djavax.net.ssl.trustStorePassword=Pa$$word1
        -Djava.ext.dirs=
        /products/IBM/WebSphere/AppServer/java/jre/lib/ext:
        /products/IBM/WebSphere/AppServer/plugins:
        /products/IBM/WebSphere/AppServer/lib:
        /products/IBM/WebSphere/AppServer/lib/ext
        ```

        In the preceding example, there are required spaces:

        ```
        -Djavax.net.ssl.trustStoreType=jks(SPACE)
        -Djavax.net.ssl.trustStore=/certs(SPACE)
        -Djavax.net.ssl.trustStorePassword=Pa$$word1(SPACE)
        ```

      - For Windows systems:

```
lax.nl.java.option.additional=
-Djavax.net.ssl.trustStoreType=jks
-Djavax.net.ssl.trustStore=
C:\Progra~1\IBM\WebSphere\AppServer\java\jre\lib\security\cacerts
-Djavax.net.ssl.trustStorePassword=Pa$$word1
-Djava.ext.dirs=
C:\Progra~1\IBM\WebSphere\AppServer\java\jre\lib\ext;
C:\Progra~1\IBM\WebSphere\AppServer\plugins;
C:\Progra~1\IBM\WebSphere\AppServer\lib;
C:\Progra~1\IBM\WebSphere\AppServer\lib\ext
```

In the preceding example, there are required spaces:

```
-Djavax.net.ssl.trustStoreType=jks(SPACE)
-Djavax.net.ssl.trustStore=
C:\Progra~1\IBM\WebSphere\AppServer\java\jre\lib\security\cacerts(SPACE)
-Djavax.net.ssl.trustStorePassword=Pa$$word1(SPACE)
```

d. Save the `ldapConfig.lax` file.

e. In the `runConfig.lax` file, add the same statements that you added to the `ldapConfig.lax` file. Save the `runConfig.lax` file.

f. Start the **ldapConfig** utility that the IBM Security Identity Manager Server provides.

g. In the Test LDAP connection page, complete these steps:

1) Enter the administrator DN (`cn=root`) and password.

2) Type the host name and port on which the directory server listens for secure communication. For example, type `test1234` for the host name and `636` for the port number.

h. Click **Test** to test secure connection to LDAP and click **Cancel** to quit the **ldapConfig** utility.

- If the test is successful, continue to the next step.
- If the test fails, return to the previous configuration steps with the `ldapConfig` connection test error message as guidance.

i. Restart the WebSphere Application Server.

j. Delete the `ldapConfig.lax` file and rename the `ldapConfig.lax.backup` file to `ldapConfig.lax`. If you want to keep the `ldapConfig.lax` file that was used during this process, blank out the truststore password that it contains.

3. To confirm that secure communication is configured, log in with your user ID and password to the IBM Security Identity Manager Server.

   A successful login indicates that you configured SSL communication between the IBM Security Identity Manager Server and the directory server.

4. If your login does not succeed, an error message on the login screen indicates that the directory server is not available. After you analyze the LDAP configuration log and the IBM Security Identity Manager Server log, try the configuration steps again.

   Additionally, you might determine whether:

   - Statements were correctly added to the `ldapConfig.lax` file.
   - Required spaces were entered in the `ldapConfig.lax` file.
   - The path to the truststore file is valid.
   - The truststore file is corrupted.

Configuration of the HTTP server for additional security and performance

For additional security and better performance, configure an HTTP server, such as the IBM HTTP Server, on a stand-alone computer that is external to any other IBM Security Identity Manager component.

By default, SSL is set to **off** in the IBM HTTP Server. To enable SSL, you must specify SSL directives (properties) in the `httpd.conf` server configuration file. A configuration that provides additional security and performance is similar to Figure 6

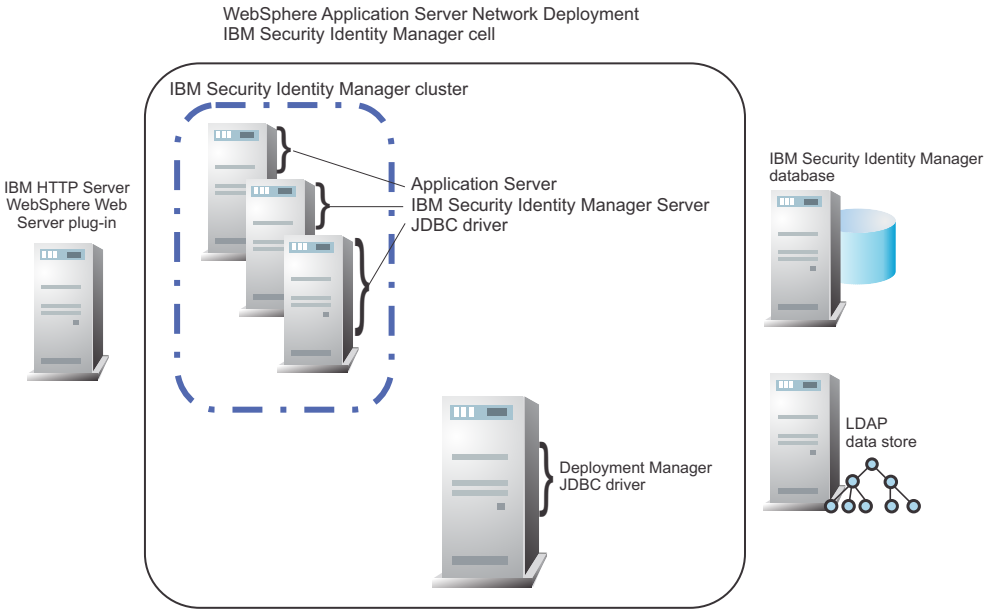

Figure 6. HTTP server configuration for increased security

SSL for the IBM HTTP server and WebSphere Application Server plug-in

The external IBM HTTP Server forwards HTTP requests that are sent to it to the internal HTTP transport of the WebSphere Application Server web container through the WebSphere Application Server plug-in.

To secure this communication, you must enable SSL for the IBM HTTP Server and configure the WebSphere Application Server plug-in to communicate securely with the WebSphere Application Server web container.

Configuring SSL for the IBM HTTP server

To configure the use of SSL, you must specify SSL directives (properties) in the `httpd.conf` file in the IBM HTTP Server. By default, SSL is set to `off` in the IBM HTTP Server.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

See the configuration information in the "Securing with SSL Communications" topic in the "Securing IBM HTTP Server" section of the WebSphere Application Server product documentation website at http://www.ibm.com/support/knowledgecenter/SSAW57/mapfiles/product_welcome_wasnd.html.

For enhanced security, do not use RC4 ciphers. Use the strongest cipher suites that the browser and web server support. To set specific ciphers, see the "Setting advanced SSL options" section of the WebSphere Application Server documentation at http://www.ibm.com/support/knowledgecenter/SSAW57/mapfiles/product_welcome_wasnd.html.

About this task

To enable SSL on the IBM HTTP Server, use the configuration information to complete these steps:

Procedure

1. Use the IBM HTTP Server iKeyman utility graphical user interface or command line to create a CMS key database file and self-signed server certificate.
2. Enable SSL directives in the `httpd.conf` configuration file for the IBM HTTP Server.
   a. Uncomment the `LoadModule ibm_ssl_module modules/mod_ibm_ssl.so` configuration directive.
   b. Create an SSL virtual host stanza in the `httpd.conf` file by using the following examples and directives:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
<IfModule mod_ibm_ssl.c>
  Listen 443
  <VirtualHost *:443>
    SSLEnable
  </VirtualHost>
</IfModule>
SSLDisable
KeyFile "c:/Program Files/IBM HTTP Server/key.kdb"
```

   Note: On Windows platforms:
   - The load module name is `LoadModule ibm_ssl_module modules/mod_ibm_ssl.dll`.
   - Always specify the address with the port on the Listen directive. To add the Listen directive in `httpd.conf` by using the default address 0.0.0.0 to listen on IPv4 port 443, type `Listen 0.0.0.0:443`.
3. Stop and start the IBM HTTP Server.
4. Test the configuration with a browser in an HTTPS session to the IBM HTTP Server (`https://ihs_host`).

Configuring SSL for the plug-in

After you enable the IBM HTTP Server for SSL, configure the WebSphere Application Server plug-in so that the IBM HTTP Server can communicate securely with the application servers. You must ensure that SSL was enabled for the

WebSphere Application Server web container by pointing your browser to a URL such as `https://dm_host:9043/ibm/console`.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Set up the IBM HTTP server on a stand-alone computer that is external to any other IBM Security Identity Manager component. For more information, see the topic "Selecting a web server topology diagram and roadmap" on the following website:

http://www.ibm.com/support/knowledgecenter/SSAW57_7.0.0/as_ditamaps/welcome_nd.html

The installation and configuration of the plug-in registers the web server with the WebSphere Application Server deployment manager, and the IBM HTTP Server becomes a managed web server. You can manage a managed web server with the WebSphere Application Server administrative console.

About this task

The application server profile to which you point during the WebSphere Application Server plug-in installation and configuration is the deployment manager itself in this topology. It creates a key file called `plugin-key.kdb` in the `app_server_root/profiles/dm_profile/etc` directory. The `plugin-key.kdb` file contains the certificates of all federated application servers.

Push the key file to the managed web server so the plug-in can establish secure application with the application servers. For more information, see the topic "Configuring the Web server plug-in for Secure Sockets Layer" on the web site:

http://www.ibm.com/support/knowledgecenter/SSAW57_7.0.0/as_ditamaps/welcome_nd.html

Procedure

1. Create a directory on the web server host for storing the key ring file that is referenced by the plug-in and associated files. For example, create a *plugin_install_root*`/etc/keys` directory.
2. On the WebSphere Application Server administrative console, click **Servers** > **Web servers**.
3. Select the web server name.
4. Click **Plug-in properties**.
5. Click **Manage keys and certificates** to access configuration options for your keys and certificates. By default, you can change the password that protects the keystore.
6. Click **OK**.
7. Click the web server **keystores** button to copy the keystore and to stash files to a managed web server.

Chapter 10. Configuration of single sign-on

Single sign-on services provide a seamless experience for a user who accesses a number of applications in the enterprise.

You can enable single sign-on for the IBM Security Identity Manager administrative console, the Self-service console, and the Identity Service Center applications with IBM Security Access Manager.

After you configure single sign-on, a user logs on to IBM Security Access Manager web security one time. The identity of the user is propagated to IBM Security Identity Manager, which eliminates the need for another login.

This function requires IBM Security Access Manager to enable single sign-on with IBM Security Identity Manager.

1. IBM Security Access Manager provides user authentication and coarse-grained authorization before it allows access to IBM Security Identity Manager.
2. IBM Security Identity Manager then applies fine-grained access control with its own Access Control Item (ACI).

You can configure IBM Security Access Manager and IBM Security Identity Manager for single sign-on with either

- WebSEAL
- IBM Security Access Manager plug-in servers

Before you configure single sign-on with WebSEAL, you must install and configure IBM Security Access Manager and WebSEAL.

Configuration of IBM Security Identity Manager for single sign-on with WebSphere Trust Association Interceptor and IBM Security Access Manager WebSEAL

Trust Association Interceptor and WebSEAL authentication eliminates the need for a separate password to access IBM Security Identity Manager.

To configure single sign-on with Trust Association Interceptor and WebSEAL, complete the following steps:

1. Define how IBM Security Access Manager maps its accounts to IBM Security Identity Manager accounts during authentication.
2. Create a user in IBM Security Access Manager that WebSEAL can use to connect to the backend server.
3. Create a junction that points to the IBM Security Identity Manager server.
4. Define two IBM Security Access Manager ACLs to control access to IBM Security Identity Manager. Define one ACL for the IBM Security Identity Manager Administrator application. Define ACLs for the IBM Security Identity Manager Self Service application and the Identity Service Center application.
5. Configure WebSphere to point to IBM Security Access Manager.
6. Configure the Trust Association Interceptor.
7. Configure IBM Security Identity Manager to use single sign-on.

8. Configure WebSEAL.

Account mapping

Single sign-on, account mapping occurs between IBM Security Access Manager and IBM Security Identity Manager during login authentication.

When a user accesses IBM Security Identity Manager with WebSEAL and single sign-on, the user must specify a IBM Security Access Manager user account and password. IBM Security Access Manager checks if the user is authorized to access IBM Security Identity Manager.

If the authentication and authorization are successful, the IBM Security Access Manager user account is passed in the `iv-user` HTTP request header to IBM Security Identity Manager. IBM Security Identity Manager passes the information in the HTTP request header to IBM Security Identity Manager for further processing. IBM Security Identity Manager uses the IBM Security Access Manager user account to find a matching user account in the IBM Security Identity Manager directory.

Typically, IBM Security Access Manager and IBM Security Identity Manager user accounts are identical. If they are identical, the IBM Security Identity Manager user can log in to IBM Security Identity Manager.

If they are not identical, you can configure IBM Security Identity Manager user account mapping. There are two configuration options. They are controlled by the `enrole.authentication.idsEqual` attribute in the `enRoleAuthentication.properties` file in the *ISIM_HOME*`/data` directory:

`enrole.authentication.idsEqual=true`
: No mapping is attempted. The IBM Security Access Manager user account passed in the `iv-user` HTTP request header must be identical to an IBM Security Identity Manager user account defined in the IBM Security Identity Manager directory for the user to log in to IBM Security Identity Manager.

  If the policy in your installation is that all IBM Security Identity Manager user accounts must have matching IBM Security Access Manager user accounts, specify `enrole.authentication.idsEqual=true` to avoid the unnecessary mapping processing and overhead.

`enrole.authentication.idsEqual=false`
: The IBM Security Access Manager user account passed in the `iv-user` HTTP request header searched the IBM Security Access Manager directory for a matching IBM Security Identity Manager user account:

  - If an identical IBM Security Identity Manager is found, the user can log in to IBM Security Identity Manager.
  - If an identical IBM Security Identity Manager account is not found, then IBM Security Identity Manager attempts to locate a matching IBM Security Identity Manager user account with the following mapping logic:

    The IBM Security Access Manager user account in the `iv-user` HTTP request header searches the IBM Security Identity Manager directory for a IBM Security Access Manager user account.

    If an identical IBM Security Access Manager user account is found in the IBM Security Identity Manager directory, it searches for the IBM Security Identity Manager Person entity that owns the IBM Security Access

Manager user account. If an owning IBM Security Identity Manager Person entity cannot be located, the user cannot log in.

If the IBM Security Identity Manager Person entity that owns the matching IBM Security Access Manager user account is found, then a search is performed for anIBM Security Identity Manager user account owned by that entity. If an IBM Security Identity Manager user account owned by the IBM Security Identity Manager Person is found, then the user can log in to IBM Security Identity Manager with that IBM Security Identity Manager user account. Otherwise, the user cannot log in.

Changing the logoff page

IBM Security Identity Manager comes with different files that can be specified as the logoff page for its Console and self-service GUI.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

The files are in the following directories:

- $*WAS_HOME*/AppServer/profiles/$*PROFILE_NAME*/installedApps/$*NODE_NAME*/ITIM.ear/itim_console.war/j../../common
- $*WAS_HOME*/AppServer/profiles/$*PROFILE_NAME*/installedApps/$*NODE_NAME*/ITIM.ear/itim_self_service.war/j../../common directory

Where $*WAS_HOME* is the directory where the WebSphere Application Server is installed.

To configure a different logoff page than the default page, modify the `ui.properties` or `SelfServiceUI.properties` files.

See the following table about Logout pages.

Table 5. Logoff pages

File name	Description

`websealLogout.jsp`	This sample file is the most secure. Use it when you want the following combined behavior when the user clicks the **Logoff** button:
	• o End the logon session.
	• o End the logon session; the **pkmslogout** function is started.
	Note: The **pkmslogout** function works only for clients with an authentication mechanism that does not supply authentication data with each request. For example, **pkmslogout** does not work for clients that use Basic Authentication, certificates, or IP address information. In these cases, you must close the browser to log out. The **pkmslogout** function provides this information to the user in a message on the logout page.
	
	You can edit this file to customize the sample logoff function. Set the values:
	For Console UI:
	`enrole.ui.logoffURL=/itim/console/j../../common/websealLogout.jsp`
	For SelfService UI:
	`enrole.ui.logoffURL=/j../../common/websealLogout.jsp`

`ssoLogout.jsp`	Use this sample file for the following combined behavior when the user clicks the **Logoff** button:
	• End the current logon session and provide a link to return to IBM Security Identity Manager.
	• Remain logged in to IBM Security Access Manager; the `iv-user HTTP` header information is still available. For example, this action provides for continued use of a portal page or a return to IBM Security Identity Manager without a logon prompt.
	
	You can edit this file to customize the sample logoff function.
	For Console UI:
	`enrole.ui.logoffURL=/itim/console/j../../common/sso_logout.jsp`
	For SelfService UI:
	`enrole.ui.logoffURL=/jsp/logon/SSOLogoff.jsp`

Procedure

1. Open the IBM Security Identity Manager $*ISIM_HOME*`/data/ui.properties` file in a text editor. To configure the Logoff page for `SelfService UI`, open the `SelfServiceUI.properties` file.
2. For the `enrole.ui.logoffURL` property, specify one of the logoff pages that are described in Table 5.

   Note: The `ssoLogout.jsp` and `websealLogout.jsp` files are sample files. They show the sample code required for the IBM Security Identity Manager logout button when WebSEAL single sign-on is enabled. You can edit these files, including language, for any functions appropriate to your environment.

Creating a user in IBM Security Access Manager that WebSEAL uses to connect to the backend server

You must create a IBM Security Access Manager user that is used to configure the single sign-on.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

IBM Security Access Manager must be installed.

About this task

Use the **pdadmin** command to create a user in IBM Security Access Manager that can be used by WebSEAL. For this task, the user name is `sso`. You can also use the web interface to create the user.

Perform this task on the server where IBM Security Access Manager is installed.

Procedure

1. Start the utility by typing `pdadmin` at a command prompt. The **pdadmin** command is located in the */PolicyDirectory Installation path*`/bin` directory.
2. Log in to a secure domain as the `sec_master` administration user to use the utility.
   a. At the command prompt, type `login`.
   b. Type `sec_master` when prompted for a user ID.
   c. Specify the associated password at the **Enter Password** prompt.

   For example:

   ```
   pdadmin> login
   Enter User ID: sec_master
   Enter Password: password
   pdadmin>
   ```
3. To create the `sso` user, type the following command on one line at the command prompt.

   ```
   pdadmin sec_master> user create sso cn=sso,cn=Users,secAuthority=Default
   sso sso password
   ```

   sso Specifies the user name you want to create. In this case, the user is `sso`.

   cn=sso,cn=Users,secAuthority=Default
   Specifies the full LDAP distinguished name (DN).

   password
   Specifies the password for the user.
4. To make the user account valid, type this command

   ```
   pdadmin sec_master> user modify sso account-valid yes
   ```

Defining IBM Security Access Manager Accounts

For users that access IBM Security Identity Manager, you must define IBM Security Access Manager user accounts in addition to Security Identity Manager user accounts.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Use Security Identity Manager to provision the IBM Security Access Manager user accounts.

This example defines `myaccount` as an identical user account for both applications. Use identical user accounts for both the IBM Security Access Manager and IBM Security Identity Manager. Otherwise, you must configure the user account mapping.

Procedure

1. On the computer on which IBM Security Access Manager is installed, start the IBM Security Access Manager utility. Type `pdadmin` at a command prompt. This prompt can be on the IBM Security Access Manager Authorization Server or the IBM Security Access Manager Policy Server. You can also use IBM Security Identity Manager to provision IBM Security Access Manager user accounts.
2. Take the following steps:
   a. Log in to a secure domain as the `sec_master` administration user to use the utility.
   b. At the command prompt, type `login`.
   c. Type `sec_master` when prompted for a user ID.
   d. Specify the associated password at the **Enter Password** prompt.

   For example:

   ```
   pdadmin> login
   Enter User ID: sec_master
   Enter Password: password
   pdadmin>
   ```
3. Define the example `myaccount` user account on IBM Security Access Manager with the **`user create`** command.

   `user create [-gsouser][-no-password-policy] user_name dn cn sn password [groups]`

   Where:

   -gsouser
   Enables global sign-on.

   -no-password-policy
   Enables the administrator to create the user with an initial password that is not checked by the existing global password policies.

   user_name
   Specifies the name of the user.

   dn
   Specifies the registry identifier assigned to the user you want to create. The format for a distinguished name is like:

   ```
   cn=Mary Jones,ou=Austin,o=IBM,c=us
   ```

   cn
   Specifies the common name assigned to the user you want to create. For example, Mary.

   sn
   Specifies the family name of the user. For example, Jones.

password
Specifies the new user account password.

groups
Specifies a list of groups to which the new user is assigned.

For example, type:

```
user create "myaccount" "cn=FirstName LastName,o=ibm,c=us"
"FirstName LastName" "LastName" password
```

4. To make the user account valid, type `user modify "myaccount" account-valid yes`.

Defining IBM Security Access Manager groups

Use the **pdadmin** utility to define two IBM Security Access Manager groups.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Create a group for the users who need access to the following IBM® Security Identity Manager user interfaces:

- Administrative console.
- Self-service user interface.
- Identity Service Center.

Procedure

1. Create a group `ITIM-Group` for users who need administrative access, by typing the following command:

   ```
   group create ITIM-Group cn=ITIM-Group,o=ibm,c=us ITIM-Group
   ```

2. Create a group `ITIM-Self-Service-Group` for users who need self-service access, by typing the following command:

   ```
   group create ITIM-Self-Service-Group cn=ITIM-Self-Service-Group,o=ibm,c=us
   ITIM-Self-Service-Group
   ```

3. Create a group `ITIM-ISC-Group` for users who need Identity Service Center access, by typing the following command:

   ```
   group create ITIM-ISC-Group cn=ITIM-ISC-Group,o=ibm,c=us
   ITIM-ISC-Group
   ```

Adding IBM Security Access Manager user account to a group

You must add IBM Security Access Manager user accounts to IBM Security Access Manager groups.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Add user account `myaccount` to the group `ITIM-Group` by typing this command:
   ```
   group modify ITIM-Group add "myaccount"
   ```
2. Add the IBM Security Access Manager user account `myaccount` to the group `ITIM-Self-Service-Group` by typing the following command:
   ```
   group modify ITIM-Self-Service-Group add "myaccount"
   ```
3. Add the IBM Security Access Manager user account `myaccount` to the group `ITIM-ISC-Group` by typing the following command:
   ```
   group modify ITIM-ISC-Group add "myaccount"
   ```

What to do next

Use IBM Security Identity Manager to provision.

Defining a junction that points to IBM Security Identity Manager Server

Create a WebSEAL junction that points to the IBM Security Identity Manager Server with the **pdadmin** utility.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

IBM Security Access Manager must be installed.

Procedure

1. Start the utility by typing `pdadmin` at a command line.
2. Log in to a secure domain as the `sec_master` administration user to use the utility.
   a. At the command line, type the text as `login`.
   b. Type the ID as `sec_master` when prompted for a user ID.
   c. Specify the associated password at the **Enter Password** prompt.

   For example:
   ```
   pdadmin> login
   Enter User ID: sec_master
   Enter Password: password
   pdadmin>
   ```
3. Locate the name of the WebSEAL server to create the junction. To determine the name of the WebSEAL server that is defined in IBM Security Access Manager, issue the `server list` command. The information that is returned contains the name in the following format:
   ```
   webseald-server_hostname
   ```

   where *server_hostname* is the WebSEAL server name.

   Note: If you install multiple WebSEAL server instances on the same workstation, the name format is *server_ instancename*-webseald-*server_ hostname*. For example:

```
pdadmin sec_master> server list
        amwpm-tam60-server
        ivacld-tam60-server
        default-webseald-tam60-server
pdadmin sec_master>
```

4. Issue the `server task create` command to create the junction. The command format is as follows.

   `server task` *webseal_server_name* `create` *options* */junction_name*

   webseal_server_name
   Name of the WebSEAL server.

   options
   The following options are needed:

   -b supply
   Defines how the WebSEAL server passes the HTTP BA authentication information to the backend server.

   -c iv-creds
   Specify a value *client_identity_options*, such as `iv-creds` to instruct WebSEAL to insert the `iv-creds` HTTP header variable.

   -e utf8_uri
   Specifies the encoding to use when it generates HTTP headers for junctions. This encoding applies to headers that are generated with both the **-c** junction option and tag-value. The value `utf8_uri` specifies that WebSEAL sends the headers in UTF-8 but that URI also encodes them. This behavior is the default behavior.

   -h ***hostname***
   Specify the fully qualified host name of the IBM Security Identity Manager Server.

   -j Supplies junction identification in a cookie to handle script-generated server-relative URLs. This option is valid for all junctions except for the type of `local`.

   -s Specifies that the junction supports stateful applications. By default, junctions are not stateful. This option is valid for all junctions except for the type of `local`.

   -p ***port_number***
   Specify the port number for the IBM Security Identity Manager Server.

   -t tcp Defines the type of junction type.

   -x Creates a transparent path junction. This option is valid for all junctions except for the type of `local`.

   junction_name
   Specify a name for the junction point. Each junction point must have a unique name.

   For example, to define a TCP junction, type the following command on one line:

```
server task default-webseald-tam60-server create -b supply -t tcp -s -x
-e utf8_uri -c iv_creds -p 9080 -h ITIMServer.ondemandinc.com /itim/ui
```

5. Create two junctions, one for Identity Service Center and the other for IBM Security Identity Manager REST.

   See the example in Step 4.

- The junction name for Identity Service Center REST must be `/itim/ui` as shown in the example of Step 4 on page 39.
- The junction name for IBM Security Identity Manager REST must be `/itim/rest`.

Defining IBM Security Access Manager ACLs

Use the **pdadmin** utility to define IBM Security Access Manager access control lists (ACLs) to these IBM Security Identity Manager interfaces: Administrative console, the Self-service console, and the Identity Service Center.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Start the utility by typing `pdadmin` at a command prompt.
2. Create an ACL requiring authenticated access to associate with the WebSEAL junction.

   Use the `acl create` *`acl_name`* command, where *`acl_name`* is the name of the ACL being created.

   For example, for administrative console access, type the following command:

   ```
   pdadmin> acl create ITIM-ACL
   ```

   For self-service access, type the following command:

   ```
   pdadmin> acl create ITIM-Self-Help-ACL
   ```

   For Identity Service Center access, type the following command:

   ```
   pdadmin> acl create ITIM-ISC-ACL
   ```

Granting access to the IBM Security Access Manager ACLs

Grant IBM Security Access Manager groups access to their corresponding IBM Security Access Manager access control lists (ACLs).

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

For the administrator group (`ITIM-Group`), self-service group (`ITIM-Self-Service-Group`), and Identity Service Center group (`ITIM-ISC-Group`), complete these steps:

Procedure

1. Add groups to the ACL with the `acl modify` *`acl_name`* `set group` *`group_name permissions`* command. For example, add the administrator group to its corresponding ACL:

   ```
   pdadmin> acl modify ITIM-ACL set group ITIM-Group Trx
   ```

   where:

acl_name
Specifies the name of the ACL groups you want to add.

group_name
Specifies the name of the group you want to add.

permissions
Specifies one or more of the following permissions:

T Specifies traverse subdirectories.

r Specifies read.

x Specifies execute.

2. To allow unauthenticated users to only Traverse the directory, modify the ACL:
   `acl modify ITIM-ACL set any-other T`
3. To modify the ACL to allow users who are not authenticated to only Traverse the directory, type this command:
   `acl modify ITIM-ACL set unauthenticated T`
4. To modify the corresponding ACL to allow `ITIM-Self-Service-Group` the authority to Traverse directories and to also read and execute, type this command:
   `acl modify ITIM-Self-Help-ACL set group ITIM-Self-Service-Group Trx`
5. To modify`ITIM-Self-Help-ACL` to allow unauthenticated users to only Traverse the directory, type this command:
   `acl modify ITIM-Self-Help-ACL set any-other T`
6. To modify `ITIM-Self-Help-ACL` to allow users who are not authenticated to only Traverse the directory, type this command:
   `acl modify ITIM-Self-Help-ACL set unauthenticated T`
7. To modify the corresponding ACL to allow `ITIM-ISC-Group` the authority to Traverse directories and to also read and execute, type this command:
   `acl modify ITIM-ISC-ACL set group ITIM-ISC-Group Trx`
8. To modify`ITIM-ISC-ACL` to allow unauthenticated users to only Traverse the directory, type this command:
   `acl modify ITIM-ISC-ACL set any-other T`
9. To modify `ITIM-ISC-ACL` to allow users who are not authenticated to only Traverse the directory, type this command:
   `acl modify ITIM-ISC-ACL set unauthenticated T`

Associating the WebSEAL junction to the ACLs

Use the `pdadmin` utility to associate the WebSEAL junction with a URL path prefix to the corresponding IBM Security Access Manager access control list (ACL).

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

Associate the ACL with the `attach` *`junction_name acl_name`* command. The command syntax is:

`acl attach` *`prefix/webseal_junction/url_path_prefix acl_name`*

where:

prefix Specifies the IBM Security Access Manager Object Space prefix for your WebSEAL server.

Type the following command to see the prefix:

```
pdadmin> object list /WebSEAL
  /WebSEAL/tam60-server-default
```

In this example, the value of the prefix is `/WebSEAL/tam60-server-default`.

webseal_junction
Specifies the name of the WebSEAL junction that you created previously with the `server task create` command. In this example, the WebSEAL junction name is `/itimserver`. The fully qualified WebSEAL junction name is `/WebSEAL/tam60-server-default/itimserver`.

url_path_prefix
Specifies the name of the URL path prefix for the IBM Security Identity Manager administrative console (`itim/console`), or the self-service console (`itim/self`), or Identity Service Center (`itim/ui`).

acl_name
Specifies the name of the corresponding IBM Security Access Manager ACL (`ITIM-ACL` or `ITIM-Self-Help-ACL`).

For example, associate the fully qualified WebSEAL junction name `/WebSEAL/tam60-server-default/itimserver/itim/console` to:

- The IBM Security Identity Manager administrator console access to the IBM Security Access Manager ACL `ITIM-ACL`.

  ```
  acl attach /WebSEAL/tam60-server-default/itimserver/itim/console ITIM-ACL
  ```

- The IBM Security Identity Manager self-service console access to the IBM Security Access Manager ACL `ITIM-Self-Help-ACL`.

  ```
  acl attach /WebSEAL/tam60-server-default/itimserver/itim/self ITIM-Self-Help-ACL
  ```

- The IBM Security Identity Manager Identity Service Center access to the IBM Security Access Manager ACL `ITIM-ISC-ACL`.

  ```
  acl attach /WebSEAL/tam60-server-default/itimserver/itim/ui ITIM-ISC-ACL
  ```

Configuring WebSphere Application Server to point to IBM Security Access Manager

This task configures the WebSphere Application Server classpath variables to point to IBM Security Access Manager.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

IBM Security Access Manager must be installed.

About this task

Perform these steps on the WebSphere Application Server that hosts IBM Security Identity Manager.

Procedure

1. Open a command window.
2. Set up the WebSphere Application Server environment to modify the variables. Type one of the following commands.
   - Microsoft Windows operating systems:

     ```
     PROFILE_HOME\bin# .\setupCmdLine.bat
     ```

     PROFILE_HOME is the WebSphere Application Server profile directory where Security Identity Manager is deployed. For example, the *PROFILE_HOME* directory can be `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01`.
   - UNIX and Linux operating systems:

     ```
     PROFILE_HOME/bin# . ./setupCmdLine.sh
     ```

     PROFILE_HOME is the WebSphere Application Server profile directory where Security Identity Manager is deployed. For example, the *PROFILE_HOME* directory can be `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01`.
3. Add `PD.jar` to the class path. Type one of the following commands on one line.

   Note:

   If the IBM WebSphere Application Server that host Security Identity Manager is version 8, then copy the `PD.jar` file from *%WAS_HOME%*`/tivoli/tam` to the `%WAS_HOME%/java/jre/lib/ext` directory.
   - Microsoft Windows operating systems:

     ```
     set CLASSPATH=%WAS_HOME%/java/jre/lib/ext/PD.jar;%WAS_CLASSPATH%
     ```
   - UNIX and Linux operating systems:

     ```
     CLASSPATH=${WAS_HOME}/java/jre/lib/ext/PD.jar:${WAS_CLASSPATH}
     ```
4. Run the **com.tivoli.pd.jcfg.PDJrteCfg** utility to configure the Java Runtime Environment component. Type one of the following commands on one line.

   For IBM WebSphere Application Server, Version 7:
   - Microsoft Windows operating systems:

     ```
     java -cp "%CLASSPATH%" -Djava.ext.dirs
     -Dpd.home="%WAS_HOME%\java\jre\PolicyDirector"
     com.tivoli.pd.jcfg.PDJrteCfg -action config -was
     -java_home "%JAVA_HOME%/jre"
     -cfgfiles_path "%JAVA_HOME%/jre"
     -host "tam policy server host name"
     ```
   - UNIX and Linux operating systems:

     ```
     java -cp ${CLASSPATH} -Djava.ext.dirs
     -Dpd.home=${WAS_HOME}/java/jre/PolicyDirector
     com.tivoli.pd.jcfg.PDJrteCfg -action config -was
     -java_home ${JAVA_HOME}/jre
     -cfgfiles_path ${JAVA_HOME}/jre
     -host "tam policy server host name"
     ```

   For WebSphere Application Server, Version 8:

   WebSphere Application Server, version 8 does not permit modification of any files under the WebSphere Application Server JRE directory. So it hosts the IBM Security Access Manager specific files under *WAS_HOME*`/tivoli/tam` directory. In previous versions, these files were hosted under the *WAS_HOME*`/java/jre` hierarchy.
   - Microsoft Windows operating systems:

```
java -cp "%CLASSPATH%" -Djava.ext.dirs
-Dpd.home="%WAS_HOME%\tivoli\tam\PolicyDirector"
com.tivoli.pd.jcfg.PDJrteCfg  -action config -was
-java_home "%JAVA_HOME%/jre"
-alt_config  -cfgfiles_path "%WAS_HOME%/tivoli/tam"
-host "tam policy server host name"
```

- UNIX and Linux operating systems:

```
java -cp ${CLASSPATH} -Djava.ext.dirs
-Dpd.home=${WAS_HOME}/java/jre/PolicyDirector
com.tivoli.pd.jcfg.PDJrteCfg -action config -was
-java_home ${JAVA_HOME}/jre
-alt_config -cfgfiles_path ${WAS_HOME}/tivoli/tam
-host "tam policy server host name"
```

The software generates the following message:

```
Configuration of Access Manager Runtime for Java completed successfully.
```

Note: If this command fails with `NullPointerException` or `InvocationTargetException`, then use the **java** command that is not included in the WebSphere Application Server. For example, if you installed Java on your Security Directory Integrator directory, then use the **/opt/IBM/TDI/V7.1/jvm/jre/bin/java** command instead of the **java** command as follows:

For IBM WebSphere Application Server, Version 7:

```
/opt/IBM/TDI/V7.1/jvm/jre/bin/java -cp ${CLASSPATH} -Djava.ext.dirs
-Dpd.home=${WAS_HOME}/java/jre/PolicyDirector
com.tivoli.pd.jcfg.PDJrteCfg -action config -was
-java_home ${JAVA_HOME}/jre
-cfgfiles_path ${JAVA_HOME}/jre
-host "tam policy server host name"
```

For IBM WebSphere Application Server, Version 8:

```
/opt/IBM/TDI/V7.1/jvm/jre/bin/java -cp ${CLASSPATH} -Djava.ext.dirs
-Dpd.home=${WAS_HOME}/tivoli/tam/PolicyDirector
com.tivoli.pd.jcfg.PDJrteCfg -action config -was
-java_home ${JAVA_HOME}/jre
-alt_config -cfgfiles_path ${WAS_HOME}/tivoli/tam
-host "tam policy server host name"
```

5. Run the SSL configuration. Type one of the following commands on one line.

   For IBM WebSphere Application Server, Version 7:

   - Microsoft Windows operating systems:

```
 java -cp "%CLASSPATH%"
-Dpd.cfg.home="%WAS_HOME%\java\jre"
-Xnoargsconversion com.tivoli.pd.jcfg.SvrSslCfg -action config
-admin_id sec_master -admin_pwd ***** -appsvr_id sso
-policysvr tam:7135:1 -port 7135 -authzsvr tam:7136:1 -mode remote
-cfg_file "%WAS_HOME%\java\jre\PdPerm.properties"
-key_file "%WAS_HOME%\java\jre\lib\security\PdPerm.ks"
-cfg_action replace
```

   - UNIX and Linux operating systems:

```
java -cp ${CLASSPATH}
-Dpd.cfg.home=$WAS_HOME/java/jre
-Xnoargsconversion com.tivoli.pd.jcfg.SvrSslCfg -action config
-admin_id sec_master -admin_pwd ***** -appsvr_id sso
-policysvr tam:7135:1 -port 7135 -authzsvr tam:7136:1 -mode remote
-cfg_file $WAS_HOME/java/jre/PdPerm.properties
-key_file $WAS_HOME/java/jre/lib/security/PdPerm.ks
-cfg_action replace
```

   For IBM WebSphere Application Server, Version 8:

   - Microsoft Windows operating systems:

```
 java -cp "%CLASSPATH%"
-Dpd.cfg.home="%WAS_HOME%\tivoli\tam"
-Xnoargsconversion com.tivoli.pd.jcfg.SvrSslCfg -action config
-admin_id sec_master -admin_pwd ***** -appsvr_id sso
-policysvr tam:7135:1 -port 7135 -authzsvr tam:7136:1 -mode remote
-cfg_file "%WAS_HOME%\tivoli\tam\PdPerm.properties"
-key_file "%WAS_HOME%\tivoli\tam\lib\security\PdPerm.ks"
-cfg_action replace
```

- UNIX and Linux operating systems:

```
java -cp ${CLASSPATH}
-Dpd.cfg.home=$WAS_HOME/tivoli/tam
-Xnoargsconversion com.tivoli.pd.jcfg.SvrSslCfg -action config
-admin_id sec_master -admin_pwd ***** -appsvr_id sso
-policysvr tam:7135:1 -port 7135 -authzsvr tam:7136:1 -mode remote
-cfg_file $WAS_HOME/tivoli/tam/PdPerm.properties
-key_file $WAS_HOME/tivoli/tam/lib/security/PdPerm.ks
-cfg_action replace
```

-admin_pwd
Corresponds to the password for `sec_master` in the previous tasks.

-appsvr_id sso
Corresponds to the IBM Security Access Manager user that was created in the first task of the procedure.

The software generates the following message:

```
The configuration completed successfully.
```

What to do next

Configure the Trust Association Interceptor.

Configuring the Trust Association Interceptor

Configure the Trust Association Interceptor for single sign-on and to enable Lightweight Third-Party Authentication and the security domain.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

IBM Security Access Manager must be installed.

Procedure

1. Log on to WebSphere Application Server administrative console.
2. Click **Security** > **Global Security**.
   a. Under Authentication, select **LTPA**.
   b. Click **Apply** to save the changes.
3. In the navigation pane under Security, click **Security domains.**
   a. Click the IBM Security Identity Manager security domain.
   b. Expand the Trust Association attribute.
   c. Select **Customize this domain**.
   d. Click the check box to enable trust association.
4. Click **Interceptors**.
   a. Click **com.ibm.ws.security.web.TAMTrustAssociationInterceptorPlus**.

b. Add the following custom properties and click **New** to add each property.
   - `Name: com.ibm.websphere.security.webseal.id`
     `Value: iv-creds`
   - `Name: com.ibm.websphere.security.webseal.checkViaHeader`
     `Value: false`
   - `Name: com.ibm.websphere.security.webseal.loginId`
     `Value: sso`

     Note: `sso` is the IBM Security Access Manager user name that was created in the first task of this procedure.
   - For IBM WebSphere Application Server, Version 8:

     `Name: com.ibm.websphere.security.webseal.configURL`
     `Value: WAS_HOME/tivoli/tam/PdPerm.properties`

     Where *WAS_HOME* is the IBM WebSphere Application Server home directory.

5. Click **OK**.

What to do next

Configure IBM Security Identity Manager for single sign-on.

Configuring IBM Security Identity Manager to use single sign-on

Change the `ui.properties` file to configure IBM Security Identity Manager to use single sign-on.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Update the following attribute in the *ISIM_HOME*`/data/ui.properties` file so that the IBM Security Identity Manager logon page is not displayed:

   `enrole.ui.taiEnabled=true`
2. Stop and restart the WebSphere Application Server for the changes to take effect.

What to do next

Configure WebSEAL.

Configuring WebSEAL

Configure WebSEAL to use the password of the IBM Security Access Manager user that is created in the first task of this procedure.

Before you begin

IBM Security Access Manager must be installed.

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

This task configures WebSEAL to use form-based authentication. Form-based authentication ends the IBM Security Access Manager session when the user logs out. For more information about the topic, see the IBM Security Access Manager product documentation website at http://www.ibm.com/support/knowledgecenter/SS9JLE/welcome.

Procedure

1. On the WebSEAL server, locate the `webseald-default.conf` file. This file is in the WebSEAL installation directory.
2. Find the entry `basicauth-dummy-passwd`.
3. Replace `dummypasswd` with the password you specified for the user that is created in the first task of the procedure.
4. Configure WebSEAL to use form-base authentication. Ensure that these values are set in the WebSEAL configuration file.
   - `ba-auth=none`
   - `forms-auth=https`
5. Save your changes.
6. Restart the WebSEAL server.

What to do next

Log on to IBM Security Identity Manager Server with any defined IBM Security Access Manager user ID.

Note:

1. You are logging in through the WebSEAL junction URI and not directly in to IBM Security Identity Manager Server.
2. If you use basic authentication, you must close the browser when you log off to end the current session.

IBM Security Identity Manager web services in a single sign-on environment

The single sign-on (SSO) application in the IBM Security Identity Manager installed `example` directory, achieves single sign-on by using the IBM Security Identity Manager web services.

The SSO application fetches the Lightweight Third Party Authentication (LTPA) token from the Hypertext Transfer Protocol (HTTP) header. The LTPA token serves as an identity token for using and maintaining the authenticated user information. The token enables the user to access the resources without requiring to log in to the IBM WebSphere Application Server again. The SSO application inserts this token into the SOAP header and then makes a web service call.

IBM Security Identity Manager installation configures **`ISIMSecurityDomain`** as its security domain that scopes to the IBM WebSphere Application Server where the IBM Security Identity Manager is deployed. Any application that runs on the same WebSphere Application Server uses the **`ISIMSecurityDomain`**. Any application that runs on a separate WebSphere Application Server runs under a separate domain and the user security realm must be configured as trusted realm in **`ISIMSecurityDomain`**.

The SSO application uses form-based login when it is not accessed from a WebSEAL junction. IBM Security Identity Manager users can log in to the sample application by using the same credentials as the IBM Security Identity Manager account. The SSO application runs on the same WebSphere Application Server and uses the **ISIMSecurityDomain**. When deployed in a separate server than IBM Security Identity Manager, the SSO application must be configured to share the IBM Security Identity Manager user registry. Upon successful authentication, the SSO application receives an LTPA cookie in the response header from the WebSphere Application Server. The sample application extracts the **LtpaToken2** cookie from the Hypertext Transfer Protocol (HTTP) header and sets it in the session. The **WebServiceCall** servlet starts the **getPrincipalPerson** web service API.

When the SSO application is accessed from a WebSEAL junction, the TAI (Trust Association Interceptor) prevents WebSphere security from requiring multiple authentications. IBM Security Identity Manager users can log in to the sample application by using the credentials from the WebSEAL authentication server. Because the SSO application is deployed with the same **ISIMSecurityDomain** in the same WebSphere Application Server, the SSO application can log in to IBM Security Identity Manager seamlessly with the LTPA token from WebSEAL. When run on a separate WebSphere Application Server, the SSO application must run under a separate domain and the user security realm must be configured as a trusted realm in the **ISIMSecurityDomain**.

The SSO application demonstrates that you can achieve SSO authentication with the IBM Security Identity Manager web services in various deployment scenarios by using the **WS-Security** header. Modify the SOAP message to add the **WS-Security** header **BinarySecurityToken**. The **BinarySecurityToken** element has the LTPA identity token embedded. Provide the **WS-Security** header with the actor attribute, `http://services.itim.com/60/actor`, to enable the IBM Security Identity Manager web services for processing the security header. Modify the SOAP message with the outgoing request of the **ClientHandler.**

Installing on a system where the IBM Security Identity Manager is installed

You must install the single sign-on application by using the IBM WebSphere Application Server administrative console.

Before you begin

Familiarize yourself with the SSO application details and installation requirements before you install it.

You must install the IBM WebSphere Application Server fixes that are specified in the IBM Security Identity Manager Release Notes. Use the installation instructions in the Release Notes to install the fixes. Install the SSO application on the IBM WebSphere Application Server where the IBM Security Identity Manager is installed.

About this task

When you install the SSO application on the same system where IBM Security Identity Manager is installed, SSO authentication uses the IBM Security Identity Manager web services. The WebSphere Application Server returns an LTPA token when you authenticate with the WebSphere Application Server.

Procedure

1. Build the SSO application to create the `itim_ws.war` file. For information about building the application, see "Building the SSO application" on page 53.
2. Install the application by using the IBM WebSphere Application Server administrative console.
   a. Log on to the IBM WebSphere Application Server administrative console. For example, `http://localhost:9060/ibm/console`
   b. Click **Applications** > **New Applications** > **New Enterprise Application**.
   c. In the **Path to the new application** area, select **Local file system**.
   d. Click **Browse** to set **Full path** to the location of the `itim_ws.war` file.
   e. Click **Next**.
   f. In the **How do you want to install the application** area, select **Detailed - Show all installation options and parameters**.
   g. Click **Next**.
   h. At the Application Security Warnings window, click **Continue**.
   i. Click the **Map context roots for Web modules** step and specify the context root value as `/itim_ws`.
   j. Click **Map security roles to users or groups** step. Select the **ITIM_CLIENT** role
   k. Click **Map Special Subjects** > **All Authenticated in Trusted Realms**.
   l. Click **Next** repeatedly until the Summary window is displayed.
   m. Click **Finish**.
   n. Click **Save** to save your changes directly to the master configuration.
3. Update the class loader properties
   a. Click **Applications** > **Application Types** > **WebSphere enterprise applications**.
   b. Click **itim_ws.war**.
   c. Under **Detailed Properties**, click **Class loading and update detection**.
   d. Select **Classes loaded with local class loader first (parent last)** for the **Class loader order** and **Single class loader for application** for the **WAR class loader policy**.
   e. Click **OK**.
   f. Click **Save** to save your changes directly to the master configuration.

What to do next

The SSO application works only with its own authentication by using the IBM Security Identity Manager user registry. You must enable authentication with WebSEAL.

Enabling authentication with WebSEAL

Enabling authentication with WebSEAL eliminates the need for a separate password to access IBM Security Identity Manager.

Procedure

1. Follow the instructions in "Configuration of IBM Security Identity Manager for single sign-on with WebSphere Trust Association Interceptor and IBM Security Access Manager WebSEAL" on page 31 Use the following modifications for web services.

2. Create an ACL that requires authenticated access to associate with the WebSEAL junction. For example,

```
pdadmin> acl create SSOAPP-ACL
```

3. Grant access to the ACL. For example,

```
pdadmin> acl modify SSOAPP-ACL set group ITIM-Group Trx
         acl modify SSOAPP-ACL set any-other T
         acl modify SSOAPP-ACL set unauthenticated T
```

4. Create the junction between WebSEAL and the back-end WebSphere server If you are installing the SSO application on an IBM Security Identity Manager cluster, the LTPA token must be enabled at the WebSEAL junction. To enable the LTPA token at the junction to the SSO application, you must provide the following information.
   - The location of the key file that is used to encrypt the identity information.
   - The password to this key file.

   Web services configuration requirements are specified in three extra options to the **server task create** command that is used to create the junction.

   -A Enables the LTPA cookies.

   -F keyfile
   Specifies the full path name location on the WebSEAL server of the key file that is used to encrypt the identity information that is contained in the cookie. The shared key is originally created on the WebSphere server and copied securely to the WebSEAL server. See the appropriate WebSphere documentation for specific details about this task.

   -Z keyfile-password
   Specifies the password that is needed to open the key file. The password appears as encrypted text in the junction XML file.

   Use these options and the other junction options when you create the junction between WebSEAL and the back-end WebSphere server. For example,

```
server task default-webseald-tam60-server create -b supply -t tcp -s -j
-e utf8_uri -c iv-creds -A -F "/abc/xyz/key.file" -Z "abcdefg" -p 9080
-h ITIMServer.ondemandinc.com/isimserver
```

5. Associate the WebSEAL junction to the ACLs. For example,

```
acl attach /WebSEAL/tam60-server-default/itimserver/itim_ws SSOAPP-ACL
```

Installing on a separate system than where the IBM Security Identity Manager is installed

You must install the single sign-on application by using the IBM WebSphere Application Server administrative console.

Before you begin

Familiarize yourself with the SSO application details and installation requirements before you install it.

You must install the IBM WebSphere Application Server fixes that are specified in the IBM Security Identity Manager Release Notes. Use the installation instructions in the Release Notes to install the fixes. Install the SSO application on the IBM WebSphere Application Server where the IBM Security Identity Manager is installed.

About this task

When the SSO application is installed on a separate system, the IBM Security Access Manager is positioned as a single sign-on front. It returns an LTPA token from the WebSphere Application Server or the IBM Security Access Manager depending on if the junction has LTPA enabled.

Procedure

1. Prepare the WebSphere Application Server environment. See "Preparing the WebSphere Application Server" on page 55
2. Build the SSO application to create the `itim_ws.war` file. For information about building the application, see "Building the SSO application" on page 53.
3. Use File Transfer Protocol (FTP) to copy the `itim_ws.war` file to the location in the system where the SSO application is going to be deployed.
4. Install the application by using the IBM WebSphere Application Server administrative console.
   a. Log on to the IBM WebSphere Application Server administrative console. For example, `http://localhost:9060/ibm/console`
   b. Click **Applications** > **New Applications** > **New Enterprise Application**.
   c. In the **Path to the new application** area, select **Local file system**.
   d. Click **Browse** to set **Full path** to the location of the `itim_ws.war` file.
   e. Click **Next**.
   f. In the **How do you want to install the application** area, select **Detailed - Show all installation options and parameters**.
   g. Click **Next**.
   h. At the Application Security Warnings window, click **Continue**.
   i. Click the **Map context roots for Web modules** step and specify the context root value as `/itim_ws`.
   j. Click **Map security roles to users or groups** step. Select the **ITIM_CLIENT** role
   k. Click **Map Special Subjects** > **All Authenticated in Trusted Realms**.
   l. Click **Next** repeatedly until the Summary window is displayed.
   m. Click **Finish**.
   n. Click **Save** to save your changes directly to the master configuration.
5. Update the class loader properties
   a. Click **Applications** > **Application Types** > **WebSphere enterprise applications**.
   b. Click **itim_ws.war**.
   c. Under **Detailed Properties**, click **Class loading and update detection**.
   d. Select **Classes loaded with local class loader first (parent last)** for the **Class loader order** and **Single class loader for application** for the **WAR class loader policy**.
   e. Click **OK**.
   f. Click **Save** to save your changes directly to the master configuration.
6. Ensure that you properly export and import the LTPA keys for correct encryption and decryption of the identity tokens (LTPA). See the IBM WebSphere Application Server documentation for setting up SSO by using LTPA with multiple servers.

7. Make the security realm that the sample SSO application is deployed a trusted realm of the IBM Security Identity Manager server. Perform the following steps on the WebSphere Application Server where IBM Security Identity Manager is installed.
   a. Log on to the IBM WebSphere Application Server administrative console. For example, `http://localhost:9060/ibm/console`
   b. Click **Security** > **Security domains** > **ISIMSecurityDomain** > **User Realms:Customized - itimCustomRealm** > **Trusted authentication realms - inbound**.
   c. Click **Add External Realm** Type in the security realm of the SSO application. For example,
      `appCustomRealm`
   d. In the *ISIM_HOME*`/data` directory, modify theenRoleAuthentication.properties file. Change `enrole.authentication.idmapper` to `com.ibm.itim.authentication.mapping.SSOIDMapper`.
   e. Restart the IBM Security Identity Manager server.

What to do next

The SSO application works only with its own authentication by using the IBM Security Identity Manager user registry. You must enable authentication with WebSEAL.

Enabling authentication on a separate system with WebSEAL

Enabling authentication with WebSEAL eliminates the need for a separate password to access IBM Security Identity Manager.

About this task

Procedure

1. Follow the instructions in "Configuration of IBM Security Identity Manager for single sign-on with WebSphere Trust Association Interceptor and IBM Security Access Manager WebSEAL" on page 31 Use the following modifications for web services.
2. On the server where the SSO application is installed, configure a Trust Association Interceptor for the application security domain. For example,
   `APPSecurityDomain`

   Follow the steps in "Configuring the Trust Association Interceptor" on page 45.
3. Define a junction that points to the SSO application. For example,
   ```
   server task default-webseald-tam60-server create -b supply -t tcp
               -s -j  -e utf8_uri -c iv_creds -p 9080
               -h AppServer.ondemandinc.com /appserver
   ```
4. Create an ACL that requires authenticated access to associate with the WebSEAL junction. For example,
   ```
   pdadmin> acl create SSOAPP-ACL
   ```
5. Grant access to the ACL. For example,
   ```
   pdadmin> acl modify SSOAPP-ACL set group ITIM-Group Trx
            acl modify SSOAPP-ACL set any-other T
            acl modify SSOAPP-ACL set unauthenticated T
   ```
6. Associate the WebSEAL junction to the ACLs. For example,
   ```
   acl attach /WebSEAL/tam60-server-default/itimserver/itim_ws SSOAPP-ACL
   ```

Starting the SSO application

To use single sign-on, you must start the SSO application.

About this task

Perform these steps on the WebSphere Application Server where the SSO application is installed.

Procedure

1. Log on to the WebSphere Application Server administrative console. For example,

   `http://localhost:9060/ibm/console`
2. Click **Applications** > **Enterprise Applications**.
3. If **itim_ws_war** is not already started, then select the **itim_ws_war** check box.
4. Click **Start**.

Testing the SSO application

After you start the SSO application, you can access the Webservices Call page to test whether single sign-on is working.

Procedure

1. Use one of the following URLs to access the SSO application with a web browser.
   - Use the URL `http://`*`hostname`*`:9080/itim_ws/jsp/Home.jsp`
   - If you enabled SSL, use the URL `https://`*`hostname`*`:9443/itim_ws/jsp/Home.jsp`
   - If you configured WebSEAL to authenticate, use the URL `https://`*`WebSEAL Proxy/junction name`*`/itim_ws/jsp/Home.jsp`.

   The *hostname* is the name of the host where the SSO application is installed.
2. Display the Webservices Call page.
   a. Log in to IBM Security Identity Manager.
   b. Specify the server host and port on which the IBM Security Identity Manager web services are deployed.

      WS Host
      The host name or IP address of the server on which the IBM Security Identity Manager is deployed.

      WS Port
      The server port on which the IBM Security Identity Manager web services are available. For example, `9080`.
3. Click **Get Principal User** to invoke the web service to fetch information about the logged in user.

Building the SSO application

You can create the `itim_ws.war` file by using the build scripts in the `/examples` directory.

Before you begin

Before you compile the examples, the **WAS_HOME** environment variable must be set to the WebSphere Application Server home directory.

About this task

To build only the SSO application, specify `sso_sample` as the target when you build the examples. For example, when you issue the **`build sso_sample`** command on Windows operating systems, or the **`./build.sh sso_sample`** command on UNIX operating systems, it creates the `itim_ws.war` file.

You must re-create the `itim_ws.war` file if you change the source of the SSO application.

Procedure

1. Log in as a user in the IBM Security Identity Manager.
2. Open a command prompt.
3. Change directories to *ITIM_HOME*/extensions/*RELEASE_VERSION*/examples.
4. Compile the source code. Type one of the following commands
   - On Windows operating systems
     ```
     build
     ```
   - On UNIX operating systems
     ```
     build.sh
     ```

   The command compiles the Java classes into the following files.
   - *ITIM_HOME*/extensions/*RELEASE_VERSION*/lib/examples.jar
   - *ITIM_HOME*/extensions/*RELEASE_VERSION*/examples/self_care/itim_expi.war
   - *ITIM_HOME*/extensions/*RELEASE_VERSION*/examples/selfregistration/sr.war
   - *ITIM_HOME*/extensions/*RELEASE_VERSION*/examples/ws/SSOSample/itim_ws.war

What to do next

Some examples require that the `examples.jar` file is available on the classpath of the application server. See "Adding the `examples.jar` file to the class path"

Adding the `examples.jar` file to the class path

Some examples require that the example code is run within IBM Security Identity Manager. To run the code, the *ITIM_HOME*/extensions/*RELEASE_VERSION*/lib/examples.jar must be added to the class path of the application server.

About this task

Procedure

1. Stop the IBM Security Identity Manager application.
2. Add the `examples.jar` file to the **`ITIM_LIB`** WebSphere shared library.
   a. Log in to the WebSphere administrative console.
   b. Click **Environment** > **Shared Libraries**
   c. Click **ITIM_LIB**.
   d. Scroll down to the bottom of the list. Add the line $*ITIM_HOME*/extensions/*RELEASE_VERSION*/lib/examples.jar
   e. Click **OK** and then click **Save**.
3. Start the IBM Security Identity Manager application for the changes to take effect.

Preparing the WebSphere Application Server

To install the single sign-on application on a separate system than where IBM Security Identity Manager is installed, you must modify the WebSphere environment on that system.

About this task

Procedure

1. Make sure that administrative security is enabled for the profile on which the SSO application is to be installed.
2. Create a folder that is named `classes`, if it does not exist, in *WAS_HOME*`/profiles/profile_name` folder on which SSO application is deployed.

   Copy the `itim_server.jar`, `itim_common.jar`, and `jlog.jar` files from *ISIM_HOME*`/lib` to the *WAS_HOME*`/profiles/profile_name/classes` folder on the WebSphere Application Server client.
3. Copy the following property files from *ISIM_HOME*`/data` to the *WAS_HOME*`/profiles/profile_name /properties` folder on the WebSphere Application Server client.
   - `enRole.properties`
   - `enRoleAuthentication.properties`
   - `enRoleLDAPConnection.properties`

     Specify the IBM Security Identity Manager LDAP server in the **url java.naming.provider.url** field. Use the IP address or machine name, such as `java.naming.provider.url=ldap://10.88.36.209:389`.
   - `Properties.properties`
   - `tmsMessages.properties`
4. Create a data folder in the *WAS_HOME*`/profiles/profile_name` folder on the WebSphere Application Server client.
5. Copy the *ISIM_HOME*`/data/keystore` folder to the *WAS_HOME*`/profiles/profile_name/data` folder on the WebSphere Application Server client.
6. Restart the WebSphere Application Server client/server.
7. Log in to the WebSphere Application Server client. On WebSphere administrative console, click **Global security** > **Security Domains** > **Copy Global Security**.
8. Enter the information for IBM Security Identity Manager Security Domain.
9. Click **OK** and save the changes to the master configuration.
10. Configure the security domain.
    a. Go to **Security Domain** > **ISIMSecurityDomain**. Specify `server1` as the scope of the domain. Click **OK** and save the changes to the master configuration.
    b. Go to **Security Domain** > **ISIMSecurityDomain** > **Security Attributes** > **Application Security**. Select the option **Customize** for this domain and check the checkbox **Enable Application Security**. Click **OK** and save the changes to master configuration.
    c. Go to **Security Domain** > **ISIMSecurityDomain** > **User Realm**. Select **Standalone custom registry**.
    d. Click **Configure**. Enter the realm name and custom registry class name. Select **Ignore case for authorization**.
    e. Click **OK** and save the changes to master configuration.

11. Export and import the LTPA keys for the encryption and decryption of the identity tokens.
    a. Export the LTPA key from the WebSphere Application Server, where IBM Security Identity Manager is installed.
        1) Go to **Global Security** > **LTPA**.
        2) Specify a password in the **Password** and **Confirm password** fields.
        3) Specify the path and LTPA key file name in the **Fully qualified key file name** field.
        4) Click **Export keys**.
    b. Import the LTPA key on the WebSphere Application Server client, where SSO application is installed.
        1) Go to **Global Security** > **LTPA**.
        2) Specify the password that was used in exporting the LTPA key in the **Password** and **Confirm password** fields.
        3) Copy the LTPA key file from the WebSphere Application Server to the WebSphere Application Server client. Specify the path of the LTPA key file on the WebSphere Application Server client in the **Fully qualified key file name** field.
        4) Click **Import keys**.
        5) Save the changes to the master configuration.

Accessing IBM Security Identity Manager consoles

After a junction exists between the WebSEAL server and the host for IBM Security Identity Manager, the URLs must include the junction name to access the Administrative console, the Self-service console, and the Identity Service Center.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

About this task

Use the following syntax in each console address:

protocol`://`*host_name*`/`*junction_name*`/`*url_path_prefix*

protocol
: Specifies the type of protocol that you want to use. Your choices are http or https.

host_name
: Specifies the name of the computer on which the IBM Security Access Manager WebSEAL server is installed.

junction_name
: Specifies the name of the WebSEAL junction.

url_path_prefix
: - For the Administrative console, specify `itim/console`.

  For example, type:

  ```
  http://TAM60-Server/ITIMServer/itim/console
  ```

Or type:

`https://TAM60-Server/ITIMServer/itim/console`

- For the Self-service console, specify `itim/self`.

  For example, type:

  `http://TAM60-Server/ITIMServer/itim/self`

  Or type:

  `https://TAM60-Server/ITIMServer/itim/self`
- For the Identity Service Center, specify `itim/ui`.

  For example, type:

  `http://TAM60-Server/ITIMServer/itim/ui`

  Or type:

  `https://TAM60-Server/ITIMServer/itim/ui`

Frequently used commands to configure single sign-on

As an aid to configuring IBM Security Access Manager single sign-on for IBM Security Identity Manager, this section lists the most frequently used **pdadmin** commands. See IBM Security Access Manager documentation for the **pdadmin** command line.

Type each command in this example on one line.

Defining IBM Security Access Manager user accounts:

```
pdadmin -a sec_master -p password -m user delete -registry "itim manager"
pdadmin -a sec_master -p password -m user create "itim manager"
 "cn=itim manager,o=ibm,c=us" "itim manager" "itim manager" tivoli

pdadmin -a sec_master -p password -m user modify "itim manager"
 account-valid yes
pdadmin -a sec_master -p password -m user show "itim manager"

pdadmin -a sec_master -p password -m user delete -registry "myname"
pdadmin -a sec_master -p password -m user create "myname"
 "cn=My Name,o=ibm,c=us" "My Name" "Name" tivoli

pdadmin -a sec_master -p password -m user modify "myname"
 account-valid yes
pdadmin -a sec_master -p password -m user show "myname"

pdadmin -a sec_master -p password -m user delete -registry "teamleader"
pdadmin -a sec_master -p password -m user create "teamleader"
 "cn=Team Leader,o=ibm,c=us" "Team Leader" "Leader" tivoli

pdadmin -a sec_master -p tivoli -m user modify "teamleader" account-valid yes
pdadmin -a sec_master -p tivoli -m user show "teamleader"

pdadmin -a sec_master -p password -m server task default-webseald-tam60-server
 delete /itimserver
```

Defining a WebSEAL TCP or SSL Junction

```
pdadmin -a sec_master -p password -m server task default-webseald-tam60-server
 create -t tcp -s -j -e utf8_uri -c iv_user -p 9080 -h ITIMServer /itimserver

pdadmin -a sec_master -p password -m server task default-webseald-tam60-server
 create -t ssl -s -j -e utf8_uri -c iv_user -p 9443 -h ITIMServer /itimserver

pdadmin -a sec_master -p password -m server task
 default-webseald-tam60-server show /itimserver
```

```
pdadmin -a sec_master -p password -m acl detach
 /WebSEAL/tam60-server-default/itimserver/itim/console
```

Defining IBM Security Access Manager ACLs:

```
pdadmin -a sec_master -p password -m acl delete ITIM-ACL
pdadmin -a sec_master -p password -m acl create ITIM-ACL
pdadmin -a sec_master -p password -m acl detach
 /WebSEAL/tam60-server-default/itimserver/itim/self

pdadmin -a sec_master -p password -m acl delete ITIM-Self-Help-ACL
pdadmin -a sec_master -p password -m acl create ITIM-Self-Help-ACL
pdadmin -a sec_master -p password -m acl detach
 /WebSEAL/tam60-server-default/itimserver/itim/ui

pdadmin -a sec_master -p password -m acl delete ITIM-ISC-ACL
pdadmin -a sec_master -p password -m acl create ITIM-ISC-ACL
```

Defining IBM Security Access Manager groups:

```
pdadmin -a sec_master -p password -m group delete ITIM-Group -registry
pdadmin -a sec_master -p password -m group create
 ITIM-Group cn=ITIM-Group,o=ibm,c=us ITIM-Group

pdadmin -a sec_master -p password -m group modify ITIM-Group
 add "itim manager"
pdadmin -a sec_master -p password -m group show ITIM-Group

pdadmin -a sec_master -p password -m group
 delete ITIM-Self-Service-Group -registry
pdadmin -a sec_master -p password -m group create ITIM-Self-Service-Group
 cn=ITIM-Self-Service-Group,o=ibm,c=us ITIM-Self-Service-Group
pdadmin -a sec_master -p password -m group modify ITIM-Self-Service-Group
 add "itim manager"
pdadmin -a sec_master -p password -m group modify ITIM-Self-Service-Group
 add "myname"
pdadmin -a sec_master -p password -m group modify ITIM-Self-Service-Group
 add "teamleader"
pdadmin -a sec_master -p password -m group show ITIM-Self-Service-Group
pdadmin -a sec_master -p password -m group
 delete ITIM-ISC-Group -registry
pdadmin -a sec_master -p password -m group create ITIM-ISC-Group
 cn=ITIM-ISC-Group,o=ibm,c=us ITIM-ISC-Group
pdadmin -a sec_master -p password -m group modify ITIM-ISC-Group
 add "itim manager"
pdadmin -a sec_master -p password -m group modify ITIM-ISC-Group
 add "myname"
pdadmin -a sec_master -p password -m group modify ITIM-ISC-Group
 add "teamleader"
pdadmin -a sec_master -p password -m group show ITIM-ISC-Group
```

Associate the WebSEAL junction to the ACLs:

```
pdadmin -a sec_master -p password -m acl modify ITIM-ACL
 set group ITIM-Group Trx
pdadmin -a sec_master -p password -m acl modify ITIM-ACL
 set any-other T
pdadmin -a sec_master -p password -m acl modify ITIM-ACL
 set unauthenticated T
pdadmin -a sec_master -p password -m acl show ITIM-ACL

pdadmin -a sec_master -p password -m acl modify ITIM-Self-Help-ACL
 set group ITIM-Self-Service-Group Trx

pdadmin -a sec_master -p password -m acl modify ITIM-Self-Help-ACL
 set any-other T
pdadmin -a sec_master -p password -m acl modify ITIM-Self-Help-ACL
```

```
 set unauthenticated T
pdadmin -a sec_master -p password -m acl show ITIM-Self-Help-ACL
pdadmin -a sec_master -p password -m acl modify ITIM-ISC-ACL
 set group ITIM-ISC-Group Trx

pdadmin -a sec_master -p password -m acl modify ITIM-ISC-ACL
 set any-other T
pdadmin -a sec_master -p password -m acl modify ITIM-ISC-ACL
 set unauthenticated T
pdadmin -a sec_master -p password -m acl show ITIM-ISC-ACL
pdadmin -a sec_master -p password -m acl attach
  /WebSEAL/tam60-server-default/itimserver/itim/console ITIM-ACL

pdadmin -a sec_master -p password -m acl attach
  /WebSEAL/tam60-server-default/itimserver/itim/self ITIM-Self-Help-ACL
pdadmin -a sec_master -p password -m acl attach
  /WebSEAL/tam60-server-default/itimserver/itim/ui ITIM-ISC-ACL
```

Chapter 11. Security layer configuration around the data model and reports

An access to the data model and reports can be restricted to a set of authorization roles. The users can create the authorization roles and associate them with the reporting entities. Only entitled users can access the data model or reports.

Authentication and authorization for IBM Cognos reports

IBM Cognos Business Intelligence administrators can set up the folders that store the reports. They can then secure those folders so that only authorized users can view, change, or perform other tasks by using the reports in the folder. To set up access control on the reports, administrators can set up the user authentication and define the access control for the set of users.

User authentication setup by using LDAP

You can configure IBM Cognos 10.2.1 components to use an LDAP namespace for authentication when the users are in an LDAP user directory.

Configuring an LDAP Namespace for IBM Directory Server

If you configure a new LDAP namespace for use with the IBM Directory Server, you must modify the necessary settings and change the values for all properties of the IBM Directory objects.

Procedure

1. Open IBM Cognos Configuration.
2. In the Explorer window, under **Security**, right-click **Authentication**.
3. Click **New resource** > **Namespace**.
4. In the **Name** box, type a name for your authentication namespace.
5. In the **Type** list, click **LDAP-General default values**.
6. Click **OK**. The new authentication namespace resource appears in the Explorer window, under the **Authentication** component.
7. In the Properties window, for the **Namespace ID** property, specify a unique identifier for the namespace.

   Tip: Do not use colons (:) in the Namespace ID property.
   For **Host and Port**, specify `<Hostname>:<port>`. For example, `localhost:389`.
8. Specify the values for all other properties to ensure that IBM Cognos 10.2.1 can locate and use your existing authentication namespace.
   - For **Base Distinguished Name**, specify the entry for a user search.
   - For **User lookup**, specify `(uid=${userID})`.
   - For **Bind user DN and password**, specify `cn=root`. For example, `cn=root` as a user name and `secret` as a password.

     Note: Specify the values if you want an LDAP authentication provider to bind to the directory server by using a specific bind user DN and password. If no values are specified, an LDAP authentication namespace binds as anonymous.

9. If you do not use external identity mapping, use bind credentials to search an LDAP directory server. Complete the following items.
   - Set **Use external identity** to **False**.
   - Set **Use bind credentials for search** to **True**.
   - Specify the user ID and password for **Bind user DN and password**.
10. To configure an LDAP advanced mapping properties, see the values that are specified in the following table.

Table 6. LDAP advanced mapping values

Mappings	LDAP property	LDAP value
Folder	Object class	`organizationalunit`, `organization`, and `container`
	Description	`description`
	Name	`ou`, `o`, and `cn`
Group	Object class	`groupofnames`
	Description	`description`
	Member	`member`
	Name	`cn`
Account	Object class	`inetorgperson`
	Business phone	`telephonenumber`
	Content locale	`(leave blank)`
	Description	`description`
	Email	`mail`
	Fax/Phone	`facsimiletelephonenumber`
	Given name	`givenname`
	Home phone	`homephone`
	Mobile phone	`mobile`
	Name	`cn`
	Pager phone	`pager`
	Password	`userPassword`
	Postal address	`postaladdress`
	Product locale	`(leave blank)`
	Surname	`sn`
	Username	`uid`

   If the schema is modified, you must make extra mapping changes.
11. To prevent the anonymous access, complete the following steps:
    a. Go to **Security** > **Authentication** > **Cognos**.
    b. Set **Allow anonymous access?** to **False**.
12. From the **File** menu, click **Save**.

Results

A new LDAP namespace is configured with the appropriate values.

What to do next

Create the users in an LDAP. See “Creating users in an LDAP” on page 63.

Creating users in an LDAP

See the example in this procedure that uses an LDAP utility to create users in LDAP.

Procedure

1. Open an LDAP utility. For example, if you are using the IBM Directory Server, the LDAP utility is `idsldapadd`.
2. Import the sample file `LdapEntries.ldif` that lists all the users who are authorized to access the reports. See the following example.

Results

After the successful import operation, you can see the users that are created in `ou=users,ou=SWG`.

Example

A sample file: `LdapEntries.ldif`

In this example, `dc=com` is the root entry. Specify the entry according to the schema that you use.

```
dn: ou=SWG, dc=com
ou: SWG
objectClass: top
objectClass: organizationalUnit

dn: ou=users,ou=SWG, dc=com
ou: users
objectClass: top
objectClass: organizationalUnit

dn: uid=steves,ou=users,ou=SWG, dc=com
uid: steves
userPassword:: hello123
objectClass: inetOrgPerson
objectClass: top
objectClass: person
objectClass: organizationalPerson
sn: Wiley
cn: Steves

dn: uid=PortalAdmin,ou=users,ou=SWG, dc=com
userPassword:: hello123
uid: PortalAdmin
objectClass: inetOrgPerson
objectClass: top
objectClass: person
objectClass: organizationalPerson
sn: Poon
cn: Chuck

dn: uid=william,ou=users,ou=SWG, dc=com
userPassword:: hello123
uid: william
objectClass: inetOrgPerson
objectClass: top
objectClass: person
objectClass: organizationalPerson
sn: Hanes
cn: William
```

```
dn: uid=lucy,ou=users,ou=SWG, dc=com
userPassword:: hello123
uid: lucy
objectClass: inetOrgPerson
objectClass: top
objectClass: person
objectClass: organizationalPerson
sn: Haye
cn: Lucy
```

What to do next

Authenticate IBM Cognos® by using an LDAP user. Complete these steps:

1. Access the IBM Cognos Gateway URI. For example, `http://localhost:portnumber/ibmcognos/cgi-bin/cognos.cgi`. The *localhost* is the IP address or network host name where IBM Cognos gateway is configured. The *portnumber* is the port on which the IBM Cognos gateway is configured.
2. Select the configured **Namespace**, and click **OK**.
3. Enter your LDAP user ID and password.
4. Click **OK**.

Access control definition for the reports and reporting packages

You can define the access control for the LDAP users who are the members of a role that is defined in the IBM Cognos namespace. Access can be granted to those users who are the members of a defined role.

A user who has the system administrator privileges can grant the access.

Initially, all users are the members of the system administrator. Therefore, you can log in with your LDAP user authentication in IBM Cognos and access the administration section before you restrict the administration access.

Restricting administration access and adding an LDAP user to system administrator role

You can restrict the IBM Cognos administration access by using the system administrators role in IBM Cognos namespace. You can also add an LDAP user to the system administrator role for IBM Cognos report administration.

Procedure

1. Log in to IBM Cognos with an LDAP user whom you want to assign the system administrator role.
2. Go to **Launch**, and click **IBM Cognos Administration**.
3. Click the **Security** tab.
4. In the **Users, Groups, and Roles** section, click **Cognos**.
5. Navigate to **System Administrator** role.
6. Click the **More** link.
7. Under **Available actions**, click **Set properties**.
8. Click the **Members** tab.
9. Click the **Add** link.
10. Under **Available entries** section, click an LDAP namespace.
11. Select the **Show users in the list** check box.

12. Select the user whom you want to assign the system administrator role and make it into selected entries list.
13. Click **OK**.
14. Select **Everyone** from the members entry.
15. Click the **Remove** link to ensure that only the added users can have the system administration access.
16. Click **OK**.
17. Click the **Permissions** tab.
18. Verify that the system administrators are listed and they are provided all the permissions. If no permissions are provided, then select the system administrators and grant all the permissions. Complete this step:
    a. Select the **Override the access permissions acquired from the parent entry** check box to grant the permissions.
19. Click **OK**.

Results

An LDAP user is added with the system administrator role.

What to do next

Create a role and add LDAP users as the members to that role. See "Creating a role and adding LDAP users as members."

Creating a role and adding LDAP users as members

The topic describes the procedure to create a role in IBM Cognos and add the members from an LDAP namespace to it.

Procedure

1. Log in to IBM Cognos with an LDAP user who has the system administrator privileges. For example, `PortalAdmin`.
2. Go to **Launch**, and click **IBM Cognos Administration**.
3. Click the **Security** tab.
4. In the **Users, Groups, and Roles** section, click **Cognos**.
5. Click the **New Role** icon from the palette.
6. Specify the name for a role. For example, `ISIMAuditor`.
7. Add the description and the screen tip.
8. Click **Next**.
9. Under **Select the members**, click **Add**.
10. Under **Available Entries Directory**, click an LDAP namespace.
11. Select the **Show users in the list** check box.
12. Select the users whom you want to add as the members to the role and make it into selected entries list.
13. Click **OK**.
14. Click **Finish**.

Results

A new role is created and LDAP users are added as the members to the new role.

Defining an access to the report by using a role

You can define an access to the report by using a role. All the members of a role can access the report or reports.

Procedure

1. Log in to IBM Cognos with an LDAP user who has the system administrator privileges. For example, `PortalAdmin`.
2. Under **Public Folders**, click **SAReportingmodel_6.0**.
3. Click the **More** link on the **Actions** toolbar that is associated with the report for which you want to provide the access.
4. Under **Available actions**, click **Set properties**.
5. Click the **Permissions** tab.
6. Select the **Override the access permissions acquired from the parent entry** check box.
7. Click **Add** link at the bottom of the list of entries.
8. Click **Cognos**.
9. Select the role that you want to add and make it to the selected entries.
10. Click **OK**.
11. Select the role and grant the permissions.
12. Optional: Remove other roles for which you do not want to provide the access.
13. Click **OK**.

Results

An access is defined to the report by using a role and all the members of a role can access the reports.

Defining an access to the reporting package by using a role

You can define an access to the report package by using a role. All the members of a role can access the report package.

Procedure

1. Log in to IBM Cognos with an LDAP user who has the system administrator privileges. For example, `PortalAdmin`.
2. Under **Public Folders**, click the **More** link on the **Actions** toolbar that is associated with the report package **SAReportingmodel_6.0**.
3. Under **Available actions**, click **Set properties**.
4. Click the **Permissions** tab.
5. Select the **Override the access permissions acquired from the parent entry** check box.
6. Click the **Add** link at the bottom of the list of entries.
7. Click **Cognos**.
8. Select the role that you want to add and make it to the selected entries.
9. Click **OK**.
10. Select the role and grant the permissions.
11. Optional: Remove other roles for which you do not want to provide the access.
12. Click **OK**.

Results

An access is defined for the reporting package by using a role and the members of a role can access the reporting package.

References for IBM Cognos report security configuration

Use the following references that provide information about the topics that are related to the security configuration for the IBM Cognos reports.

Access the IBM Cognos Business Intelligence 10.2.1 documentation at http://pic.dhe.ibm.com/infocenter/cbi/v10r2m1/index.jsp and search for the following terms.

- **Security model**.
- **Authentication providers**.
- **Add or remove members of a cognos group or role**.
- **Create a cognos group or role**.
- **Authorization**.
- **Access permissions and credentials**.

Chapter 12. Setting the session timeout interval for IBM Security Identity Manager

You must set the session timeout value for IBM Security Identity Manager.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Log in to WebSphere Application Server administrative console.
2. Click **Applications** > **Application Types** > **WebSphere enterprise applications** > **ITIM**.
3. In Web Module Properties, click **Session management**.
4. Select the check box for **Override session management**.
5. Under Session timeout, click **Set timeout** and type a value in minutes. The default 30 minutes.
6. Click **OK**.
7. Click **Save**.
8. Restart the WebSphere Application Server for the changes to take effect.

Chapter 13. Setting the session timeout interval for the IBM Security Identity Manager Service Center

You must set the session timeout value for Identity Service Center.

Before you begin

Depending on how your system administrator customized your system, you might not have access to this task. To obtain access to this task or to have someone complete it for you, contact your system administrator.

Procedure

1. Log in to WebSphere Application Server administrative console.
2. Click **Applications** > **Application Types** > **Business-level applications** > **IdentityServiceCenterApplication**.
3. Click `com.ibm.isim_CU.eba`.
4. In Additional Properties, click **Session management**.
5. Select the check box for **Override session management**.
6. Under Session timeout, click **Set timeout** and type a value in minutes. The default 30 minutes.
7. Click **OK**.
8. Click **Save**.
9. Restart the WebSphere Application Server for the changes to take effect.

Index

A

access
  IBM Security Identity Manager consoles 56
access control
  reports 64
access reports, defining 66
account
  define 36
  mapping 32
ACL 41
  access 40
  define 40
adapter 15
administrator
  access restriction 64
authentication
  IBM Cognos reports 61
authorization
  IBM Cognos reports 61

B

backend server 35

C

certificate 13
class path
  adding the examples.jar file 54
clustered environment 11
command
  example 57
  single sign-on 57
communication 15, 17
  configuration 19
configure 31
  database server 22
  single sign-on 31
  SSL 22
custom application 17

D

directory server 22

E

examples.jar file 54
external user registry 3

F

file types 13
form-based authentication 46

G

group
  define 37

H

HTTP port, disabling 7
HTTP server 28

I

IBM HTTP server 28, 29
IBM Security Identity Manager server
  configure 25
iKeyman utility 20

L

LDAP
  creating users 63
LDAP namespace
  configuration 61
logoff page
  Security Identity Manager Console
    changing 33
  Self Service GUI
    changing 33

M

members, adding 65
middleware 19

P

pdadmin 38
pdadmin utility 40
protocol 56

R

references
  security configuration 67
report package
  defining access 66
reports
  access control 64
  define access 66
roles
  creating 65

S

secure environment
  practices 5
security 28
  disabling the non-SSL port 7
security configuration references 67
security layer configuration 61
self-signed certificate 20
separate systems
  single sign-on 55
session timeout interval
  set 69, 71
setup, user authentication 61
single sign-on 46
  enabling authentication with WebSEAL
    same system 49
    separate system 52
  installing
    same system 48
    separate system 50
  starting 53
  testing 53
  web services 47
  WebSphere Application Server preparation 55
SSL 11, 28, 29, 30
  configure 22
  SSL authentication, one-way, two-way 10
  SSL communication 9
  SSL configuration
    example 19
    preparation 20
  SSL implementation 12
  SSL terminology 9
SSL client 9
SSL communication 23
  test 26
SSL server 9

T

Trust Association Interceptor 31
  configure 45

U

ui.properties 46
user
  create 35
user account
  add 37
user authentication
  setup 61
user registry
  WebSphere 3

W

web services
  enabling authentication with WebSEAL
    same system 49
    separate system 52

web services *(continued)*
in a single sign-on environment 47
installing single sign-on
same system 48
separate system 50
single sign-on for separate systems 55
starting single sign-on 53
testing single sign-on 53
WebSEAL 31, 35, 46
WebSEAL junction 38
associate 41
WebSphere Application Server 28
configure 42
WebSphere Application Server plug-in 30
WebSphere security 1

Printed in USA