

# ITIM – WebSphere Security Configuration

## 1. Introduction:

An ITIM Web application can run in a remote WebSphere application server (WAS) environment from the ITIM server. In this configuration, each WAS can have its own security arrangement, and can be placed under the same security realm (security domain) or different security realm. A security realm consists of all servers configured with the same user registry realm name. The realm can be the machine name of a Local OS user registry. In this case, all application servers must reside on the same physical machine. In other cases, the realm can be the machine name of a Lightweight Directory Access Protocol (LDAP) user registry. Multiple nodes can be configured to the same LDAP user registry, and under the management of a single realm.

The basic WAS 5.1.x security configuration provides two type of user registries, Local OS registry and LDAP user registry. Other than choosing the type of user registry, an authentication mechanism has to be configured. There are two authentication mechanisms provided and can be chosen from the WAS administration console. They are Simple WebSphere Authentication Mechanism (SWAM) and Lightweight Third-Party Authentication (LTPA). The SWAM is intended for simple, non-distributed, single application server environments. The LTPA, on the other hand, is intended for distributed, multiple application server and machine environment. For a distributed WAS environment where multiple WAS instances interacting with each other, may or may not be under the control of a WAS deployment manager (DM), the LTPA authentication is the recommended mechanism.

The LTPA protocol uses cryptographic keys (LTPA keys) to encrypt and decrypt user data that passes between the servers. These keys need to be shared for the resources in one WAS to access resources in other WAS', assuming that all WAS instances use the same LDAP registry. If the different WAS instances are not managed by a DM node (which will automatically distribute the LTPA key to all managed WAS instances), the LTPA export/import utility should be used to distribute the single LTPA key to all WAS instances.

With the general WAS security background, the best practice in running a remote ITIM Web application is to have both WAS instances (the WAS hosts the ITIM client and the WAS hosts the ITIM server) share the same LDAP user registry (under the same security realm), and use the LTPA authentication protocol. However, there might be cases that two WAS' need to be placed under different security realms (using different user registries) and apply different authentication mechanisms. This document discusses the possible security configurations between the ITIM Web application and the ITIM server when they run in different WAS instances. Required WAS configuration and a code example of using ITIM API to log in a remote ITIM server is provided in this document also.

## 2. The possible ITIM Web client and ITIM server security configuration:

Security configuration between an ITIM Web application and a remote ITIM server can be categorized in two groups. They can be placed in the same security realm or in different security realms. The possible configurations are:

1. Two ITIM components are in the same security realm. With this configuration, the LDAP user registry and the LTPA authentication mechanism have to be used. This is the recommended configuration since not only the user credential flows between two WAS instances can be encrypted and validated, but also the user identity is recognized by the authorization engine. There are two topologies in setting this configuration.
  - A. Both WAS instances are within the same WAS cell, and managed by a WAS deployment manager (DM).
  - B. The WAS instances are in the different cells.

Note that with this configuration, you cannot use SWAM on either side of WAS. The SWAM authentication is used for a security realm with a single server. The LTPA authentication has to be chosen when there are multiple servers with a security realm.

2. ITIM components are in different security realms, and same or different authentication mechanism is applied on each hosting WAS. There are four different combinations listed in the table below.

	WAS hosting ITIM Web Appl	WAS hosting ITIM Server
Authentication mechanism	LTPA	LTPA
	LTPA	SWAM
	SWAM	LTPA
	SWAM	SWAM

**Table 1: Possible combinations of authentication mechanism usage between two ITIM components**

When the ITIM server is running in a secure environment, the ITIM Web application must run in a security environment also. Attempting to log into a secured ITIM server from an unsecured ITIM Web application will fail with the WAS exception `com.ibm.websphere.csi.CSIException: SECJ0053E: Authorization failed for /UNAUTHENTICATED`, because the credentials provided by the ITIM client cannot be authenticated.

It is possible to operate an ITIM Web application from a secure environment to an unsecured ITIM server with a specific setup. However, it is not recommended to run the ITIM server in an unsecured environment.

### 3. ITIM Web application example

To secure ITIM access and resources, ITIM uses the Java Authentication and Authorization Service (JAAS) login modules provided by WAS as well as its own custom login module to perform programmatic authentication to the WAS security runtime and the ITIM server. The `InitialPlatformContext` is the class in ITIM API for an ITIM application to provide the platform (WAS) credential and platform data and to invoke the WAS JAAS login module. The following code snippet shows how to set up the platform context and invoke the class.

```
Hashtable env = new Hashtable();
env.put(InitialPlatformContext.CONTEXT_FACTORY,
    "com.ibm.itim.apps.impl.websphere.WebSpherePlatformContextFactory");
env.put(PlatformContext.PLATFORM_URL, "iiop://itimserver.ibm.com:2809");
env.put(PlatformContext.PLATFORM_PRINCIPAL, "itimadmin");
env.put(PlatformContext.PLATFORM_CREDENTIALS, "password");
PlatformContext platform = null;
try {
    platform = new InitialPlatformContext(env);
}
```

In this example, `PLATFORM_URL`, is consisted of the host name where the ITIM server running, and the WAS bootstrap port number, defaulted to 2809. `PLATFORM_PRINCIPAL` is the ITIM EJB user identifier which is mapped to the ITIM security role, and `PLATFORM_CREDENTIALS` is the password of the ITIM EJB user. `InitialPlatformContext` invokes the WAS `WSLoginLoginContext` class to authenticate the platform credential and return a `Subject` object that represents an authenticated entity. Once the platform context is established, the ITIM application can then use the `LoginContext` class to invoke the ITIM JAAS login module for further operation.

There are a couple of ITIM Web application examples, Self-Registration and Self-Care, packaged with ITIM 4.6. The source code and the instructions of running the example applications are located at `<ITIM_install_directory>/extensions/examples/selfregistration` and `<ITIM_install_directory>/extensions/examples/self-care`. More information about the implementation of these ITIM Web applications can be found there.

The Self-Registration application is used in this document as an example in setting up secured operation between two ITIM components in the same WAS security domain or in different WAS security domains. Follow the readme in the `<ITIM_install_directory>/extensions/examples/selfregistration` directory, a `sr.war` file can be easily built and deployed into a WAS server, assuming the host name of the WAS server is `itimclient.ibm.com`. The `context.properties` file of the application, located at `<WAS_install_home>/installedApps/<cell_name>/sr.war.ear/sr.war/WEB_INF/classes`, should

be updated with the correct platform URL, the ITIM EJB user ID, and the password after sr.war is deployed.

#### 4. Both ITIM client and ITIM server in the same security realm

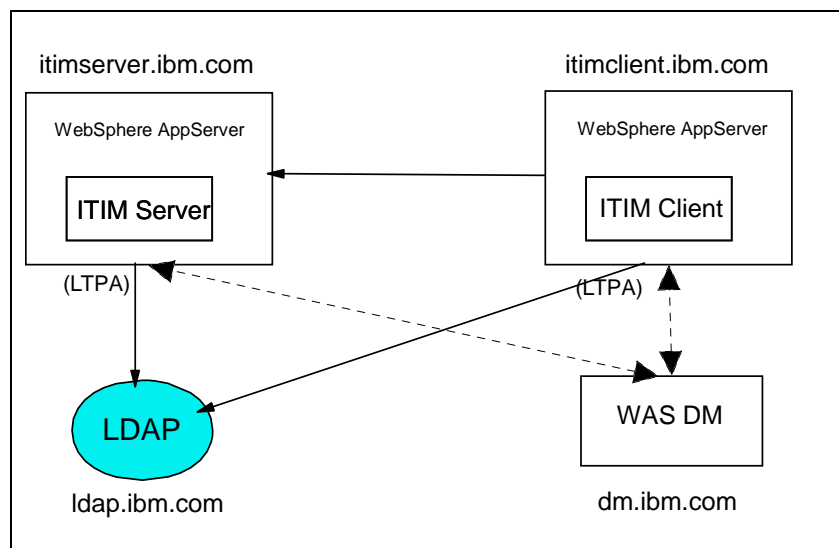
When multiple WAS servers are configured in the same security realm, a distributed user registry, such as the LDAP server, has to be used and shared among the servers. The LTPA protocol should be the authentication mechanism in this case. Prior to selecting the LTPA protocol, a LTPA key, which is used to encrypt and decrypt the user data transmitted between WAS servers, should be generated and shared as the common cryptographic key among all the WAS processes in the security realm.

Discussion here assumes the global security is already on and the LDAP has been configured as the user registry. It will only describe the setting associates to the LTPA configuration in WAS. For more information about configuring the LDAP user registry and activating the WAS global security, refer to *ITIM 4.6 Server Installation and Configuration Guide for WebSphere Environment*.

There are two topologies where the ITIM components reside when they are in the same WAS security realm. They are;

- Both WAS instances are managed by a WAS DM node
- The WAS instances are in different cells but share the same user registry

The Figure 1 illustrates the first topology that both WAS instances are managed by a WAS DM node and within the same security realm.



**Figure 1: WAS instances within a WAS cell**

On configuring the LTPA authentication protocol, a password should be set to encrypt the LTPA key. The LTPA key will be automatically generated when the password is initialized. To set the LTPA password, on the WAS DM administrative console:

- 1 Click Security -> Authentication Mechanisms -> LTPA
- 2 Enter the Password and Confirm Password which is used to encrypt the LTPA key
- 3 Apply and save the configuration

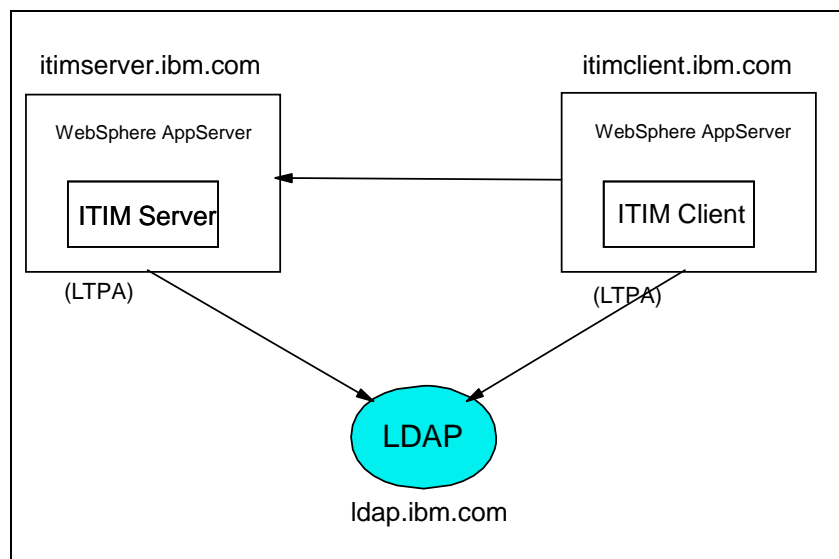
Once the LTPA key is generated, the key will be automatically distributed to all WAS instances with the cell by the DM node.

To select the LTPA as the authentication mechanism, on the WAS DM administrator console:

- 1 Click Security -> Global Security
- 2 Select LTPA under Active Authentication Mechanism
- 3 Apply and save the configuration

After restarting all WAS instances to activate the LTPA authentication mechanism and the LTPA key, the ITIM Self-registration application can be started by entering the URL <http://itimclient.ibm.com:9080/sr>, and submits the request to the ITIM server for creating a new ITIM identity.

The Figure 2 illustrates the second case which the two hosting WAS instances are running in different WAS cells (not managed by the WAS DM node) but under the same security realm.



**Figure 2: WAS instances in difference cells sharing the same user registry**

The same procedure as the previous case should be used to configure the user registry, to select the LTPA method and generate the LTPA key, and to set WAS global security. One extra step is required to export the LTPA key from one WAS instance and to import it into another WAS instance since the same LTPA key is needed to cipher the data transmitting in between. Assuming the master LTPA key is at the WAS hosting the ITIM server (itimserver.ibm.com). Follow the steps below to export and import the key.

Log on to the WAS administrator console at itimserver.ibm.com, and

- 1 Click Security -> Authentication Mechanisms -> LTPA
- 2 Enter the fully qualified file name on the field Key File Name
- 3 Click the Export Keys button on the top of the panel to export the key

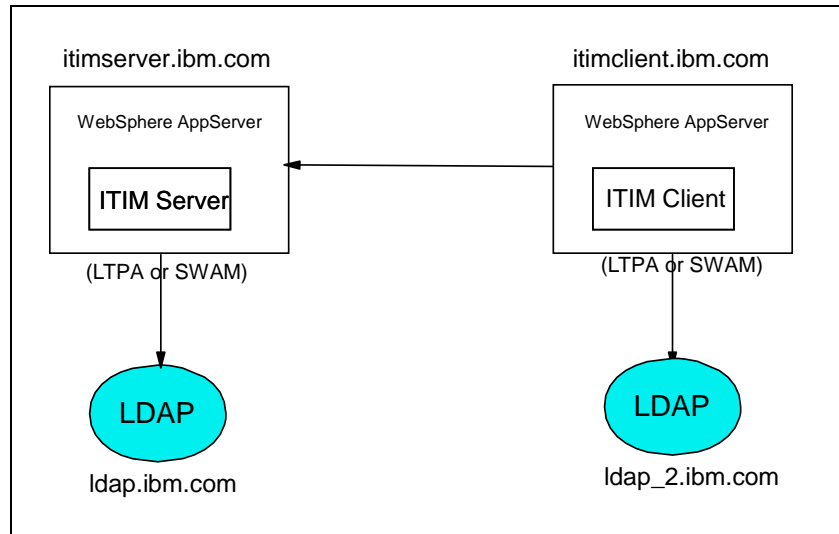
Transfer the key file to the WAS hosting the ITIM application client (itimclient.ibm.com), and log on to the WAS administrator console at itimclinet.ibm.com. Then,

- 1 Click Security -> Authentication Mechanisms -> LTPA
- 2 On the Key File Name field, enter the fully qualified file name which contains the master LTPA key
- 3 On the Password and Confirm Password fields, enter the password used to create the master LTPA key
- 4 Click the Import Keys button on the top of the panel to import the key
- 5 Apply and save the change
- 6 Restart the WAS to activate the new LTPA key

At this point, the ITIM Self-Registration application should be ready to submit the request to the remote ITIM server for creating new ITIM identities.

## **5. ITIM client and ITIM server in different security realms**

When the WebSphere application servers, hosting the ITIM Web Application and the ITIM server, use different user registries, they are under different security realms. In Figure 3 below, each WebSphere application server has its own LDAP user registry, and the authentication mechanism on each server can be either SWAM or LTPA. The user registry on each side can be replaced with the local OS if it desired.



**Figure 3: WAS instances in different cells and under different security realms**

By default, when WAS makes an outbound request from one server to another server in a different security realm, the request is rejected. This request is rejected to protect against a rogue server reading potentially sensitive information if successfully impersonating the home of the object. In order to enable the ITIM Self-Registration application to send outbound requests to the ITIM server in a different realm, the following changes should be made.

### Update the ITIM Self Registration platform context code

At the ITIM client side, in order for WAS JAAS WSLogin module, which is invoked by the `InitialPlatformContext` class, to authenticate the ITIM server EJB user ID and password at the target realm (the ITIM server side), the target realm name should be passed in the platform context on instantiating the `InitialPlatformContext` class.

ITIM Self-Registration has the `getPlatform()` method which reads the platform data from a properties file and passes the platform data to the `InitialPlatformContext` class. The following code snippet is extracted from the `getPlatform()` method.

```

...
try {
    ResourceBundle rb = ResourceBundle.getBundle("context");
    appServerURL = rb.getString("appServerURL");
    ejbUser = rb.getString("ejbUser");
    platformContextFactory =
        rb.getString("platformContextFactory");
    ejbPwd = rb.getString("ejbPwd");
    realm = rb.getString("securityRealm");
}
catch ( MissingResourceException e ) {
    .....
}
  
```

```

Hashtable env = new Hashtable();
env.put(InitialPlatformContext.CONTEXT_FACTORY,
        platformContextFactory);
env.put(PlatformContext.PLATFORM_URL, appServerURL);
env.put(PlatformContext.PLATFORM_PRINCIPAL, ejbUser);
env.put(PlatformContext.PLATFORM_CREDENTIALS, ejbPwd);
env.put("enrole.appServer.ejbuser.realm", realm);
PlatformContext platform = null;
try {
    platform = new InitialPlatformContext(env);
}

```

The lines of code in bold characters above are added to read the target security realm name, and pass the realm name to the InitialPlatformContext class.

### Update context.properties file of the ITIM Self Registration

A properties file, context.properties, containing the platform context is included in the package. On deploying the sr.war file onto WAS, the context.properties is placed under the <WAS\_install\_home>/installedApps/<cell\_name>/sr.war.ear/sr.war/WEB\_INF/classes directory. The properties in this file should be updated to reflect the system values after the application is deployed. In this case, a new property **securityRealm**, which does not currently exist, should be added. For the LDAP user registry, the realm name is the host\_name.domain\_name:port\_number. For the local OS registry, the realm name is the fully qualified host name. On the Windows platform, the realm name is domain name if a domain is in use for the local OS registry.

```

platformContextFactory=com.ibm.itim.apps.impl.websphere.WebSphere
PlatformContextFactory
appServerURL=iiop://itimserver.ibm.com:2809
loginContext=ITIM
ejbUser=itimadmin
ejbPwd=password
securityRealm=ldap.ibm.com:389

```

### Update WAS configuration

In order to allow the existing security information to flow to a target server residing in a different security realm and the WSLogin module to handle a foreign security realm, a couple of WAS configuration changes at the ITIM client side should be added. Log on to the WAS administrator console at itimclient.ibm.com, and

Enable WSLogin credentials to flow to a remote app server by adding the following custom properties to the WSLogin handler:



1. Click Security → JAAS Configuration → Application Logins → WSLLogin → JAAS Login Modules → com.ibm.ws.security...WSLoginModuleImpl → Custom Properties
2. Add the following couple properties
3. Apply and save the settings

Name	Value
use_appcontext_callback	True
use_realm_callback	True

Enable authentication for outbound RMI calls by configuring basic authentication on CSIv2 Outbound authentication protocol

1. Click Security → Authentication Protocol → CSIv2 Outbound Authentication
2. Ensure the “Basic Authentication” is selected, and set the field Trusted Target Realms as below.
3. Apply and save the settings

Name	Value
Basic Authentication	Supported
Trusted Target Realms	ldap.ibm.com:389

The WAS server should be restarted to activate the above changes.

Following the above instructions, rebuild and redeploy the sr.war file, update the context.properties file, restart the Self-Registration application from the WAS administrator console, the Self-Registration application should be able to operate with the remote ITIM server residing in a different security realm.

## 6. References

For more information concerning WAS security setup, refer to

- *IBM WebSphere V5.0 Security WebSphere Handbook (SG24-6753)*
- WebSphere Application Server 5.1.x InfoCenter  
(<http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/>)