ITIM 5.0 – WebSphere 6.1 Security Configuration

1. Introduction:

An ITIM J2EE application can run in a remote WebSphere application server (WAS) 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 realms. 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, or a federated repository name which is newly supported by WAS 6.1. Multiple nodes can be configured to an LDAP user registry or to a federated repository user registry, and under the management of a single realm.

The basic WAS 6.1 security configuration provides three types of user registries, Local OS registry, LDAP user registry, and federated repository. Other than choosing the type of user registry, an authentication mechanism has to be configured. The Lightweight third-Party Authentication (LTPA) is the default authentication mechanism which can be used in a single application server environment or a distributed environment. The Simple WebSphere Authentication Mechanism (SWAM), which is solely for a single application server environment, is being deprecated in WebSphere Application Server Version 6.1 and will be removed from the future release. However, SWAM is still available in WAS 6.1.

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 or the same federated repository. 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 hosting the ITIM client and the WAS hosting the ITIM server) share the same LDAP user registry or the same federated repository (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 6.1 configuration and a code example of using ITIM 5.0 API to log in a remote ITIM 5.0 server is provided in this document also.

2. The possible ITIM 5.0 J2EE client and ITIM 5.0 server security configuration:

Security configuration between an ITIM 5.0 J2EE application and a remote ITIM 5.0 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 LTPA authentication mechanism has to be used. The user registry can be either an LDAP user registry or a federated repository user registry. 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 within 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 J2EE 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 J2EE 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 5.0 J2EE 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 the 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`, consists of the host name where the ITIM server is running, and the WAS bootstrap port number, defaults 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 `WSLogin LoginContext` class to authenticate the platform credential and return a Subject object that represents an authenticated entity. Once the platform context is established, the ITIM J2EE application can then use the `LoginContext` class to invoke the ITIM JAAS login module for further operation.

There are a couple of ITIM J2EE application examples, Self-Registration and Self-Care, packaged with ITIM 5.0. 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 J2EE 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, where an 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, an 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 5.0 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.

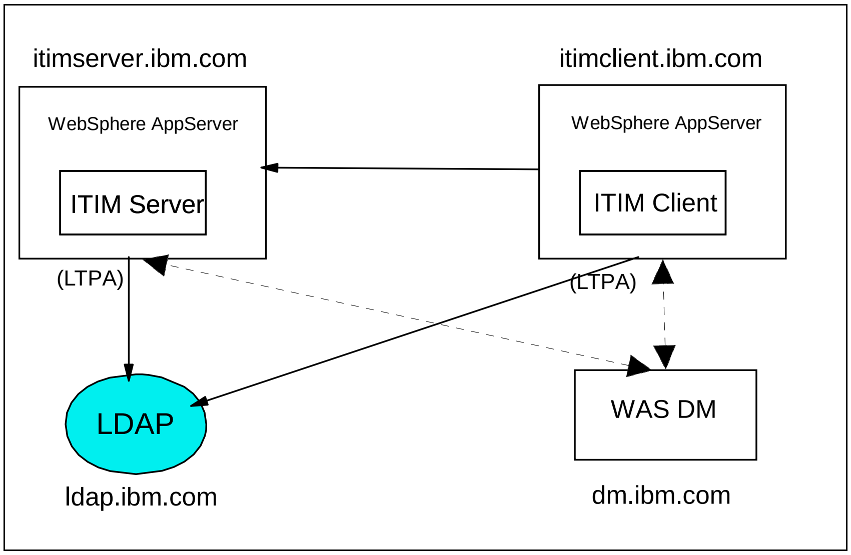

Figure 1: WAS instances within a WAS cell

When the WAS instances, where both ITIM components reside, are governed by a WAS DM node, the LTPA key, generated by the DM node on activating the WAS administrative security, will be automatically generated and distributed to the all WAS instances within the DM cell. No extra procedure is needed to make the ITIM client to connect to the ITIM server.

If the WAS security is already on, the ITIM Self-registration application can be accessed by entering the URL http://itimclient.ibm.com:9080/sr, and submits the request to the ITIM server for creating a new ITIM identity. If the WAS security is to be activated, restart all WAS instances before accessing the Self-registration application.

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.

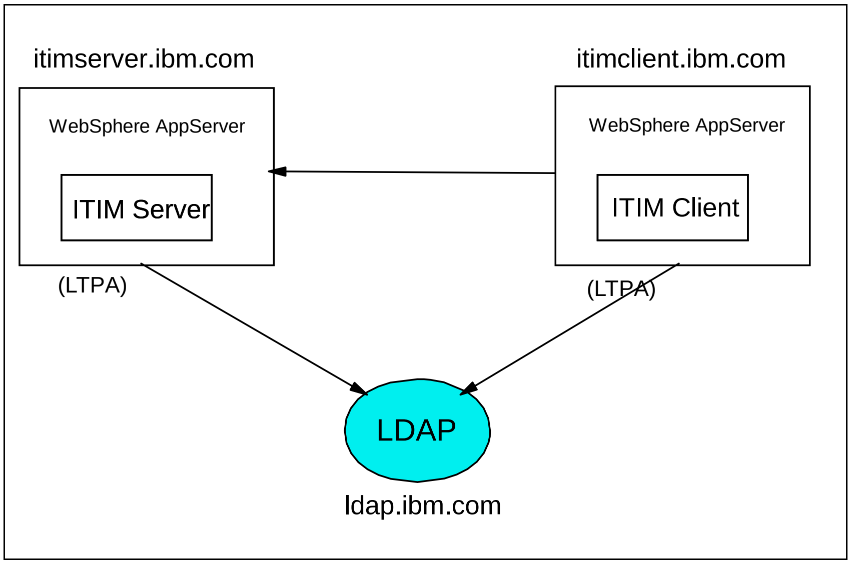

Figure 2: WAS instances in difference cells sharing the same user registry

In case two WAS instances are in the different cells but share the same user registry, the same LTPA key must be used at the both sides. The LTPA key has to be exported from one WAS instance and imported into another WAS instance. To export and import the LPTA key from the WAS administrative console follow the steps below.

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

1 Click Secure administration, applications, and infrastructure > Authentication mechanisms and expiration -> Cross-cell single sign-on
2 Enter the Password, the Confirm password, the Fully qualified key file name
3 Click Export keys

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 Secure administration, applications, and infrastructure > Authentication mechanisms and expiration -> Cross-cell single sign-on
2. Enter the Password, the Confirm password, the Fully qualified key file name pointing to the imported key file
3. Click Import keys
4. Restart the WAS receiving the new LPTA 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 J2EE 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 or a federated repository if it desired.

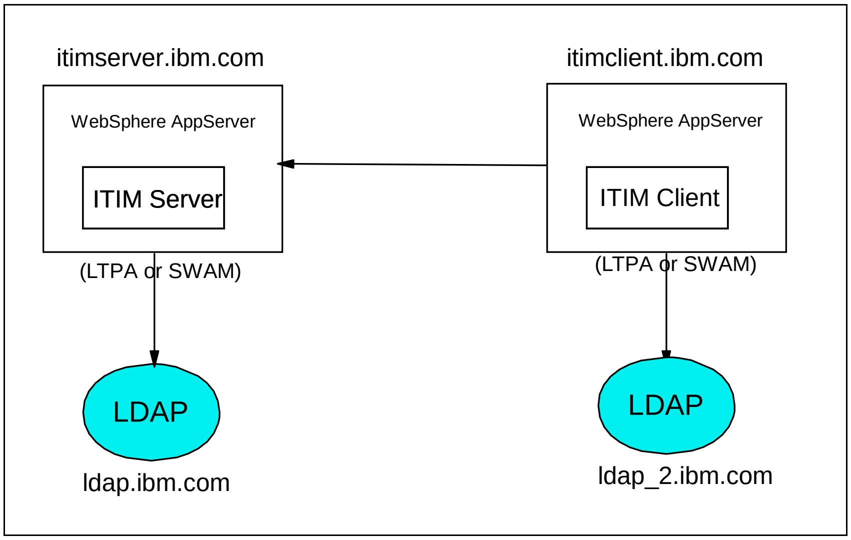

Figure 3: WAS instances in different cells and under different security realms

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, the `InitialPlatformContext` class is invoked, which runs WAS JAAS WSLogin module, to authenticate the provided credential. The credential should be the EJB user ID and password of the ITIM server at the target realm. Also, the target realm name should be passed in the platform context on instatiating 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(PlatformContext.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 the domain name if a domain is in use for the local OS registry. For the federated repository, the realm name is defined in WAS by the user or by default as defaultWIMFileBasedRealm. The realm can be found on the WAS Integrated Solution console following the Security -> Secure administration, application, and infrastructure -> Federated repositories (This selection is on the drop down field of "Available realm definitions". Once selected the definition, click the Configure button to advance to the next panel.) -> Realm name.

```
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, some 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 → Secure administration, applications, and infrastructure -> Java Authentication and Authorization Service → Application Logins → WSLogin → JAAS Login Modules → com.ibm.ws.security.common.auth.module.WSLoginModuleImpl→Custom Properties
2. Set the following two properties and set them to true (default is false)
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→ Secure administration, applications, and infrastructure -> RMI/IIOP security → 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

Then, the default signer certificate at the ITIM 5.0 server side should be imported to the ITIM J2EE client side. To accomplish this task, follow the steps below –

At the ITIM 5.0 server host:

- Run <WAS_Home>/bin/ikeyman.exe (or ikeyman.sh)
- On the ikeyman GUI, choose PKCS12 as the key database type
- Open the trust store trust.p12 at <WAS_Profile_Home>/config/cells/<cell_name>/nodes/<node_name>
- Default password for opening the trust.p12 is WebAS
- Select "Signer Certificates" under the "Key database content"
- Choose the default certificate
- Extract the certificate with the data type Based64-encoded ASCII data. A certificate file with the file type .arm will be genereated.
- Transfer the file to the ITIM client side

At the ITIM 5.0 client host:

- Run <WAS_Home>/bin/ikeyman.exe (or ikeyman.sh)
- On the ikeyman GUI, choose PKCS12 as the key database type
- Open the trust store trust.p12 at <WAS_Profile_Home>/config/cells/<cell_name>/nodes/<node_name>
- Default password for opening the trust.p12 is WebAS
- Select "Signer Certificates" under the "Key database content"
- Click "Add" to import the transported file containing the signer certificate of the server
- Enter a label for the imported certificate

Here :

<WAS_Home> is the installation directory of the WebSphere Application Server
<WAS_Profile_Home> is the WebSphere application server profile where either the ITIM server or the ITIM client is deployed
<cell_name> is the cell name where the WAS server hosting the ITIM server or the ITIM client is running
<node_name> is the node name where the WAS server hosting the ITIM server or the ITIM client is running

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 Application Server V6.1 Security WebSphere (SG24-6316)*
- WebSphere Application Server 6.1 InfoCenter (http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtop_overview.html)