

# Secure your Web resources

## Concrete scenarios offer Tivoli security tips

Skill Level: Introductory

[Peter Kovari \(peter.kovari@us.ibm.com\)](mailto:peter.kovari@us.ibm.com)  
WebSphere Specialist  
IBM

29 May 2003

This tutorial drives through three WebSphere Application Server/Tivoli Access Manager integration scenarios. You'll learn how to share the user registry, and to protect Web resources with WebSEAL via both LTAP and TAI. Setup and configuration details are provided for testing and configuring all the scenarios in the tutorial.

## Section 1. Before you start

### About this tutorial

The objective of this tutorial is to show different scenarios for integrating WebSphere Application Server and Tivoli Access Manager.

The tutorial includes a quick installation guide for a simple system on which you can run the scenarios introduced here. You will find details for the scenarios and detailed configuration samples that you can run on your system.

The tutorial should take you about 30 minutes to complete, not including the installation time.

This tutorial is a valuable resource for those who want to get quick hands-on experience with WebSphere Application Server and Tivoli Access Manager

integration.

This material is useful for architects who want detailed information about WebSphere and Tivoli integration in the enterprise. It can also help system administrators get a feel for the system administration tasks in a WebSphere/Tivoli domain.

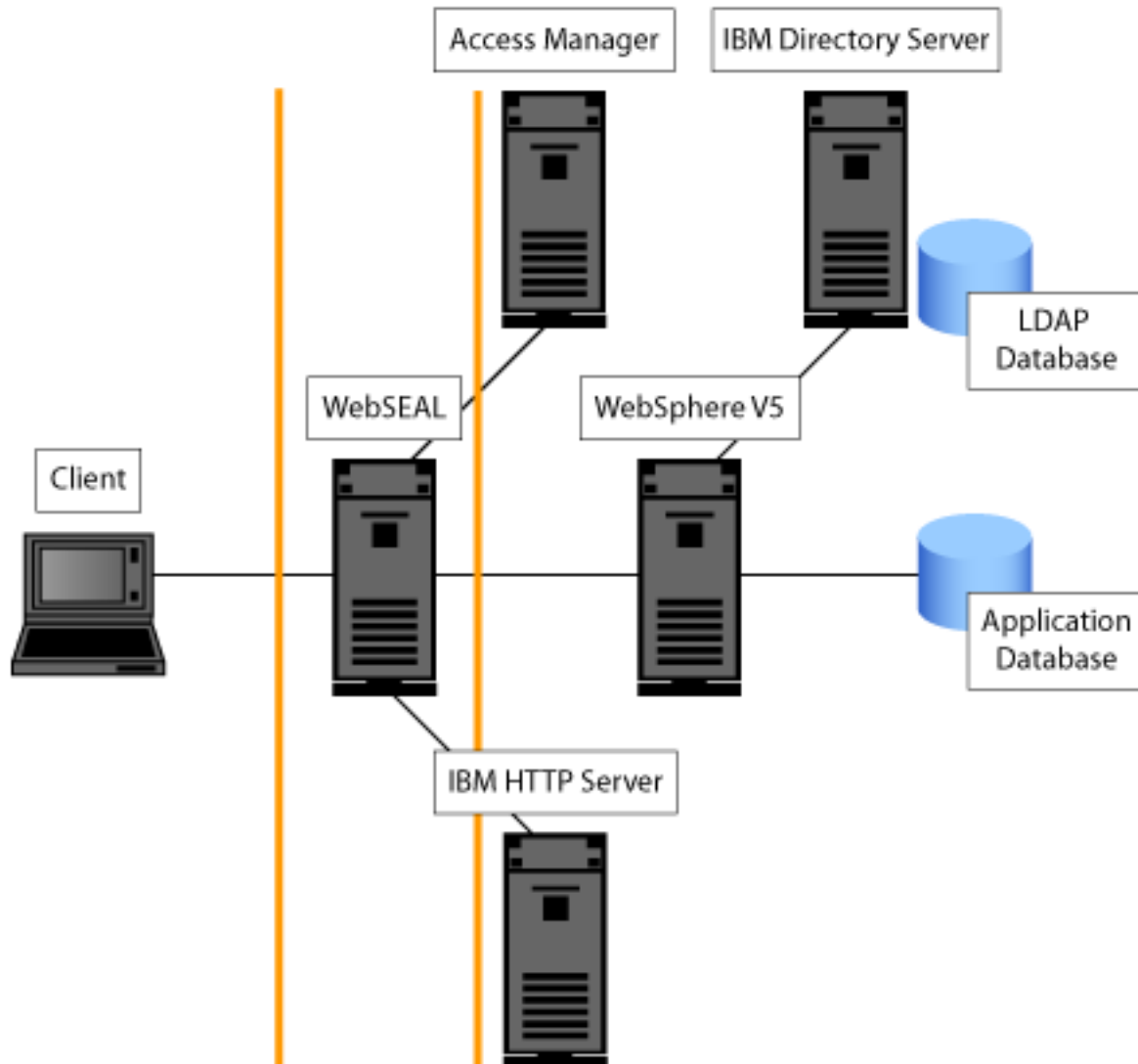
## Prerequisites

To complete the steps in this tutorial, you'll need the following software installed on your computer:

- WebSphere Application Server V5 for NT (download a [trial version](#))
- Tivoli Access Manager V4.1
- Tivoli Access Manager V4.1 Web Security (WebSEAL)

## Architecture

The diagram below depicts the enterprise architecture for WebSphere Application Server and Tivoli Access Manager.



The WebSEAL server usually sits in the demilitarized zone (DMZ). WebSEAL can be combined to run as a plug-in with the WebSphere Edge Server to provide load balancing for the servers behind the second firewall.

The Access Manager authorization server is positioned in the secure network zone, together with the directory server.

The WebSphere Application Server can use both the authorization server and the directory server for authorization and authentication purposes.

The IBM HTTP Server can move to the secure network zone; in this case, WebSEAL forwards and routes the requests to the static content on the Web server.

This architecture could be extended with the Tivoli Web Portal Manager, which is a graphical interface used to manage the Access Manager domain.

## Scenario system requirements

The scenarios in this tutorial can be run on one system if it meets the minimum requirements:

- Intel Pentium III or 4 processor or equivalent, 1 GHz minimum
- 1 GB memory minimum
- Windows 2000 Server (Professional also works, although it is not a supported platform)
- ServicePack 3 with the latest critical updates

This one system can run all the components introduced in the previous panel.

We'll use `dwserver` as the server name in this tutorial. Make sure that you replace this server name with your own hostname when appropriate.

---

## Section 2. Installation and configuration

### Installation steps

In this section, we will:

1. Install WebSphere Application Server V5.0 base
2. Install Tivoli Access Manager base, including:
  1. IBM Directory Server V4.1
  2. Tivoli Access Manager Runtime
  3. Tivoli Access Manager Authorization Server
3. Install Tivoli Access Manager WebSEAL

The installation process may take from a half an hour to a couple of hours, depending on the speed of your system. We'll take a look at each step in turn in the next few sections.

## Install WebSphere Application Server V5

The installation steps here are provided for the tutorial environment. If you already have an infrastructure set up, you can skip most of the upcoming installation steps; you only have to make sure that you have all the components installed.

The very first step is to install the application server. WebSphere Application Server provides the runtime environment for the J2EE applications.

1. Install WebSphere Application Server V5.0.
2. Select a custom installation; do not install the Application Server Samples, Embedded Messaging, IBM HTTP Server, Web Server Plugins, Performance And Analysis Tools, or Javadocs, as they are not required for this tutorial.
3. Set the application server directory to `C:\WebSphere\Appserver`.
4. Set the node name -- to `dwServerNode`, for example; then set the machine name.
5. Continue with the installation until it is done.

If you'd like, you can install the IBM HTTP server, which can be used for the directory server for management purposes. If you choose not to install the Web server, Tivoli Access Manager will install an earlier version of IBM HTTP Server for the directory server. Later in the tutorial (in the sections [Using LTPA to integrate WebSphere and WebSEAL](#) and [Using TAI to integrate WebSphere and WebSEAL](#)), to forward the requests to the application server, we use the WebSEAL security proxy rather than the IBM HTTP server. If you decide to keep the Web server for management purposes for the directory server, set the Web server port during installation to a nondefault port (i.e., to a port other than 80).

## Install Tivoli Access Manager directory server

The directory server stores user information for authentication purposes. It can also store any other data in a directory structured format. To install it, first run the install script `ezinstall_ldap_server.bat`. (You'll find this and other batch scripts mentioned in this section in the root directory of the appropriate install CD.)

Once you've run the script, answer the questions on the panels you're presented as follows (use `passw0rd`, with a numeric zero substituted for the letter O, for every password request):

**IBM DB2 configuration options:**

Option	Value
1. Administration ID .....	db2admin
2. Administration Password .....	***** (passwd)
3. Installation Directory .....	c:\sqllib

**IBM HTTP Server configuration options:**

Option	Value
1. Administration ID .....	Administrator
2. Administration Password .....	***** (passwd)
3. HTTP Port .....	80
4. Installation Directory .....	c:\ibmhttp

**IBM Global Security Toolkit configuration option:**

Option	Value
1. Installation Directory .....	c:\ibm\gsk

**IBM Directory Server configuration options:**

Option	Value
1. LDAP Administrator ID (DN) .....	cn=root
2. LDAP Administrator Password .....	***** (passwd)
3. LDAP Server Hostname .....	dwserver
4. LDAP DN for GSO Database .....	o=mycompany
5. LDAP Server Port .....	389
6. LDAP SSL Keyfile .....	D:\common\pd_ldapkey.kdb
7. LDAP SSL Key File Password .....	*****
8. SSL Client Certificate Label .....	PDLdap
9. Installation Directory .....	c:\ibm\ldap

Finally, install the patch for the LDAP server, using the script:  
`apply_ldap41_patch.bat`.

## Install Tivoli Access Manager runtime

You'll need to run the Tivoli Access Manager runtime environment on every node in the Access Manager domain that has a connection to Access Manager. Install the Runtime component using the `ezinstall_pdmgr.bat` script.

Once you've run the script, answer the questions on the panels you're presented as follows (leave those fields for which no value is indicated blank):

**IBM Tivoli Access Manager runtime configuration options:**

```

Option                                     Value
1. Configure Using This Registry Type ..... ldap
2. LDAP Server Hostname ..... dwserver
3. LDAP Server Port ..... 389
4. LDAP DN for GSO Database ..... o=mycompany
5. Enable SSL with LDAP Server ..... n
6. LDAP SSL Keyfile .....
7. LDAP SSL Keyfile DN .....
8. LDAP SSL Key File Password .....
9. LDAP Server SSL Port ..... 636
10. Installation Directory ..... c:\tivoli\policydirector

```

### IBM Tivoli Access Manager policy server configuration options:

```

Option                                     Value
1. LDAP Administrator ID (DN) ..... cn=root
2. LDAP Administrator Password ..... ***** (passwd)
3. Security Master Password ..... ***** (passwd)
4. SSL Server Port ..... 7135
5. Policy Server SSL Certificate Lifetime ... 365
6. Enable Download of Certificates ..... Y

```

## Install Tivoli Access Manager authorization server

The Tivoli Access Manager provides authorization services for different applications and servers. Install the authorization component using the `ezinstall_pdacl.d.bat` script.

Once you've run the script, answer the questions on the panel you're presented as follows:

### IBM Tivoli Access Manager authorization server configuration options:

```

Option                                     Value
1. LDAP Administrator ID (DN) ..... cn=root
2. LDAP Administrator Password ..... ***** (passwd)
3. Security Master Password ..... ***** (passwd)

```

## Install Tivoli Access Manager WebSEAL

Tivoli Access Manager WebSEAL is a security reverse proxy providing security services for the Web clients. Install the WebSEAL component using the `ezinstall_pdweb.bat` script.

Once you've run the script, answer the questions on the panel you're presented as follows:

## Access Manager WebSEAL Server (PDWEB) configuration options:

Option	Value
1. Security Master Password .....	***** (passw0rd)
2. Enable SSL with LDAP Server .....	n
3. LDAP SSL Keyfile .....	
4. LDAP SSL Keyfile DN .....	
5. LDAP SSL Key File Password .....	
6. LDAP Server SSL Port .....	

## Installation check

Open the Windows Services panel and make sure you have the following services up and running, and that the startup type is set to the right value for each:

1. Access Manager Authorization Server (Manual)
2. Access Manager Auto-Start Service (Automatic)
3. Access Manager Policy Server (Manual)
4. Access Manager WebSEAL (Manual)
5. DB2 - DB2 (Automatic)
6. DB2 - LDAPDB2 (Automatic)
7. IBM Directory Server (Automatic)
8. IBM HTTP Administration (Manual)
9. IBM HTTP Server (Manual)

---

## Section 3. Tivoli Access Manager scenarios with WebSphere

### Scenarios overview

You can integrate Tivoli Access Manager and WebSphere Application Server on different levels. This tutorial aims to introduce the different options for integration and

to give detailed information on how to do it. The following scenarios are described in this tutorial:

- **Sharing a user registry:** WebSphere and Access Manager share the same user registry.
- **Using WebSEAL to protect Web resources:** We'll see two different scenarios on this topic:
  - Integrating WebSEAL and WebSphere using lightweight third-party authentication (LTPA)
  - Integrating WebSEAL and WebSphere using trust association interceptor (TAI).

These scenarios can be used separately, in any combination, or all together in a solution. We'll cover these scenarios in detail in the following sections.

---

## Section 4. Sharing a user registry

### Configuring TAM

If you plan to share the user registry between the security server, Tivoli Access Manager, and the application server, WebSphere Application Server is essential in order to have the same source for authentication and authorization services. (If you decide not to share the user registry, you will need to maintain two separate user registries such that both have the same users and user information.)

The first step is to set WebSphere to use the same user registry as Tivoli Access Manager. Later, we will enable security for WebSphere, and it will use the user registry we configure. For an application, it would be nice to have the same user registry for security purposes, for authentication, in WebSphere, and in Tivoli Access Manager.

On this panel, you will start by creating a couple of users and groups in Access Manager for the tutorial. Keep in mind that creating users with Access Manager and creating users with the Directory Management Tool are separate processes. If you fail to realize the difference, you will end up failing to find users in the security domain.

To create users and groups in the users registry:

1. Start the Administration Command Prompt for Tivoli Access Manager; you can also use the Web Portal Manager for Tivoli Access Manager in your environment. At the command prompt, type in `login`, then provide the username `sec_master` and the password `passw0rd`.
2. Create the WebSphere server administrator user by issuing the following command:

```
user create -no-password-policy wsadmin cn=wsadmin,o=mycompany wsadmin  
"Administrator" passw0rd
```

3. Enable the administrator user:

```
user modify wsuser01 account-valid yes
```

4. Create a Web user:

```
user create -no-password-policy webuser01 cn=webuser01,o=mycompany webuser01  
"User01" passw0rd
```

5. Enable the Web user:

```
user modify webuser01 account-valid yes
```

6. Create a WebSEAL user:

```
user create -no-password-policy webseal01 cn=webseal01,o=mycompany webseal01  
"WebSEAL01" passw0rd
```

7. Enable the WebSEAL user:

```
user modify webseal01 account-valid yes
```

8. Create a group for the Web users:

```
group create webgroup cn=webgroup,o=mycompany webgroup
```

9. Add the Web user to the group:

```
group modify webgroup add webuser01
```

10. Type in `exit` to leave the application.

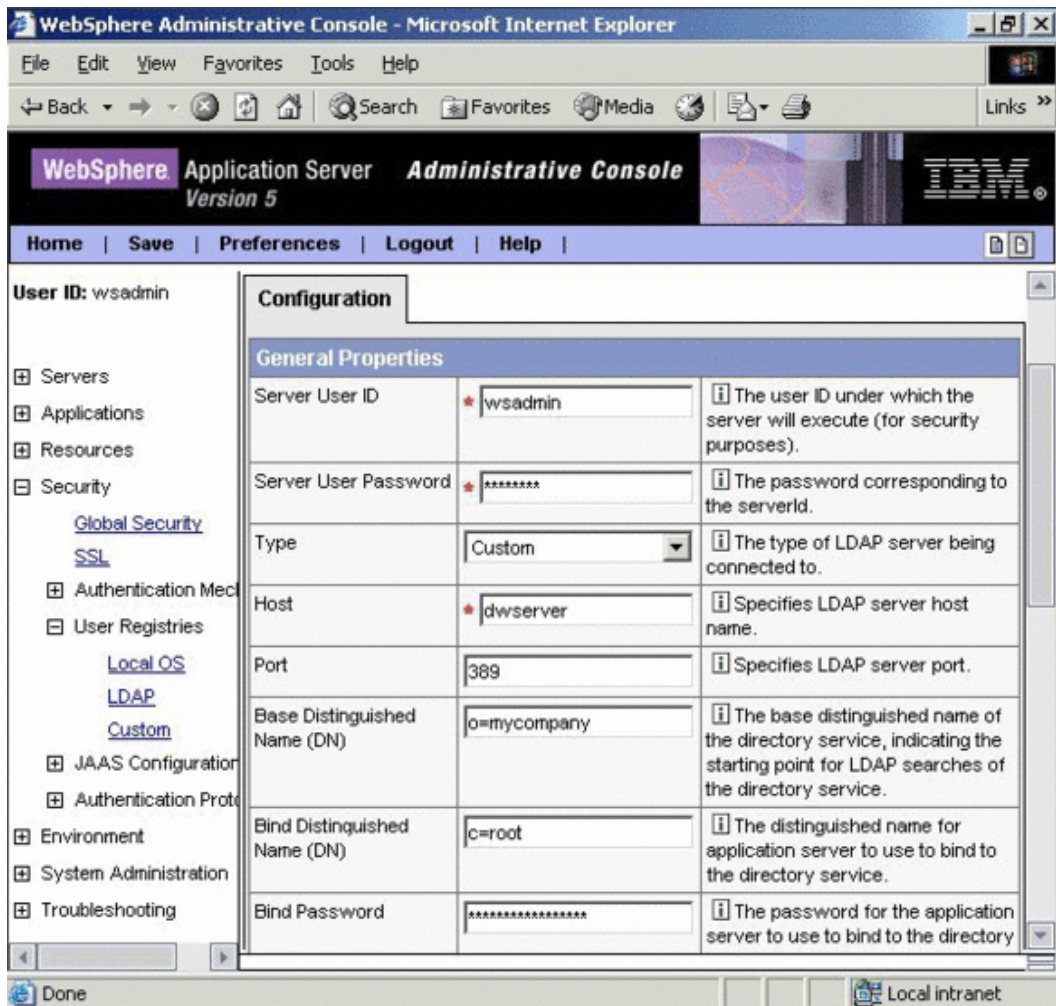
## Configuring WebSphere

Next, we will configure WebSphere to use the same directory server that Access Manager is using.

1. Start the WebSphere Administrative Console.
2. Log in with a name -- `wsadmin`, for example.
3. Navigate to **Security=>User registries=>LDAP**. Enter the following data:

```
Server User ID:          wsadmin
Server User Password:   passw0rd
Type:                   Custom
Host:                   Put your server's name here
Base Distinguished Name (DN): o=mycompany
Bind Distinguished Name (DN): cn=root
Bind Password:          passw0rd
```

The screen should look like this:



4. Click **Apply**.
5. Click **Advanced LDAP Settings**. Modify the User Filter field so that it contains the following expression:

```
(&(uid=%v)(objectclass=inetOrgPerson)(objectclass=ePerson))
```

Modify the Group Filter field so that it contains the following expression:

```
(&(cn=%v)(|(objectclass=groupOfNames)(objectclass=accessGroup)(objectclass=groupOfUniqueNames)))
```

Modify the Group ID Map field so that it contains the following expression:

```
groupOfNames:member;accessGroup:member; groupOfUniqueNames:uniqueMember
```

6. Click **Apply** and then save the WebSphere configuration. When you apply the changes under the Advanced LDAP Settings, the LDAP type will change to **Custom**.
7. At this stage, security is not yet enabled for the application server. Navigate to **Security=>Global Security**.
8. Enable security by checking the first checkbox. Disable the enforcement of Java 2 security; we do not need it for this tutorial. At the bottom of the screen, select **LDAP** as the active user registry
9. Click **OK**, then save the configuration for WebSphere.
10. Log out from the Administrative Console and close the browser.
11. Restart the application server with the following command:

```
stopserver server1  
startserver server1
```

After security is enabled, use the following command to stop the application server:

```
stopserver server1 -username wsadmin -password passw0rd
```

## Test your configuration

The easiest way to test the configuration is to launch the WebSphere Administrative Console and log in with the server ID `wsadmin`. You should be able to log in without any problem. Note that the browser starts a secure session (via HTTPS) for the administration application.

You can also turn on tracing for the application server and check the trace file and look for the authentication action to see where the user was found. The tracing expression for security is:

```
com.ibm.ws.security.*=all=enabled:SASRas=all=enabled:
```

---

## Section 5. Using WebSEAL to protect Web resources

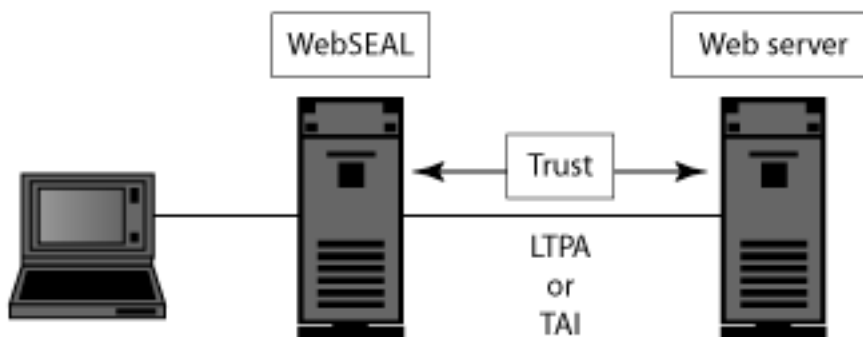
## Using WebSEAL to protect Web resources

Tivoli Access Manager WebSEAL is a high-performance Web server with a proxy plug-in and security services. Therefore, it acts as a security reverse proxy in the system. WebSEAL can be a proxy for any Web server, so it can serve as such for the WebSphere embedded Web server running in the Web container. WebSEAL captures all the incoming requests from the Web, authenticates them, and routes them to the appropriate Web server.

For more information about WebSEAL, refer to the Tivoli Access Manager documentation.

Simply using WebSEAL to protect Web resources will not help an application propagate user information to WebSphere. For example, if a user logs in to the Web application and starts browsing, the first time the Web module accesses a secured EJB method from the EJB module, it will fail because of the lack of user credentials.

To pass user credentials from WebSEAL, we can use either trust association interceptor (TAI) or lightweight third-party authentication (LTPA). The common element in both situation is the need to set up trust between the security reverse proxy (WebSEAL) and the application server (WebSphere Application Server). We'll do that in this section, and then consider the two individual scenarios in the subsequent sections.



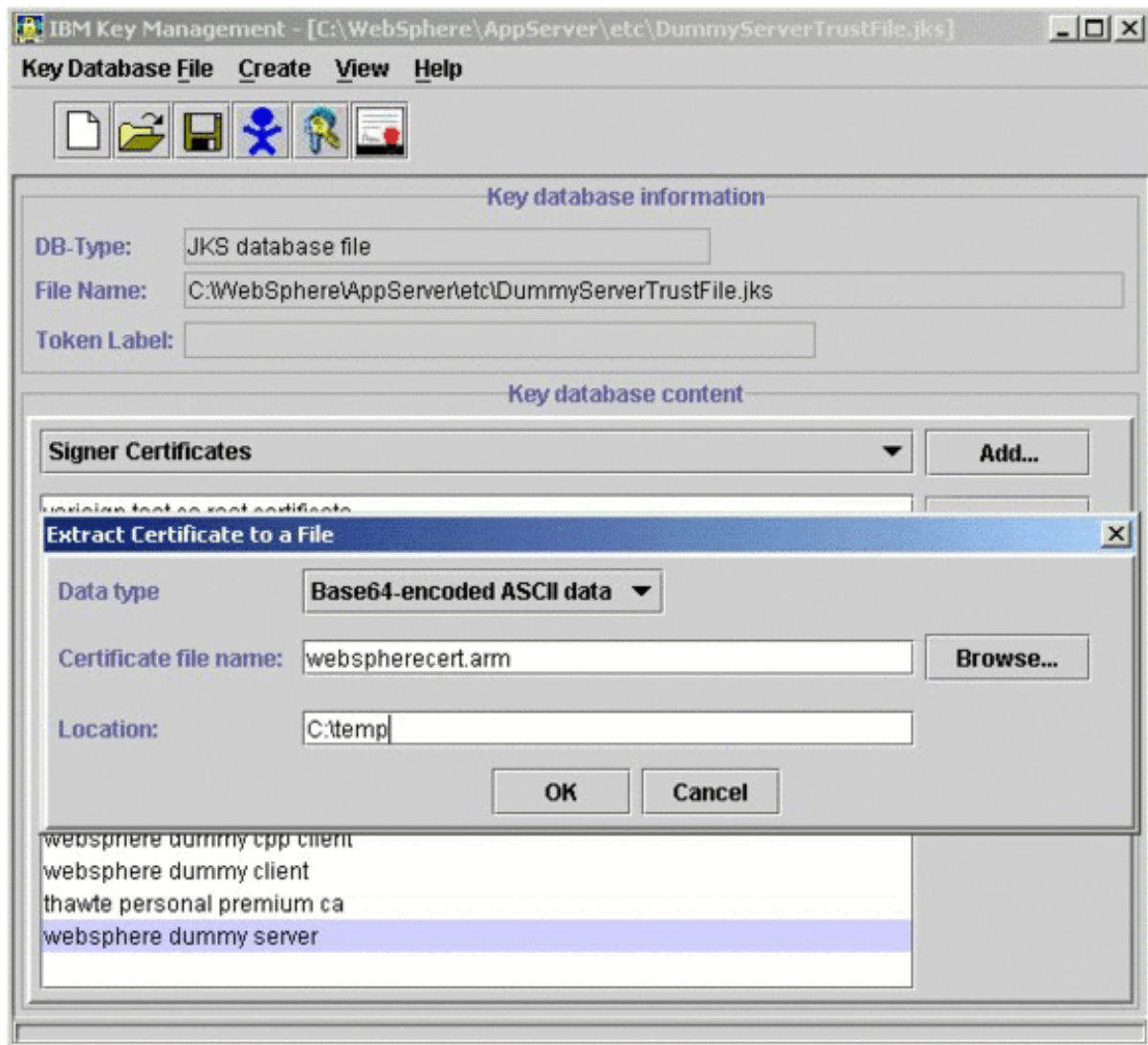
## Exporting an SSL certificate from WebSphere

In order to set up trust between WebSEAL and WebSphere using an SSL connection, we have to exchange the public certificates between the two servers. The first step is to export the WebSphere server certificate from WebSphere.

1. Start `ikeyman` from the WebSphere `bin` directory.
2. Open `DummyServerTrustFile.jks` from the WebSphere `etc`

directory, using the password `WebAS`.

3. Extract the certificate `websphere dummy server` into a file -- `c:\temp\webspherecert.arm`, for example.
4. Close the application.

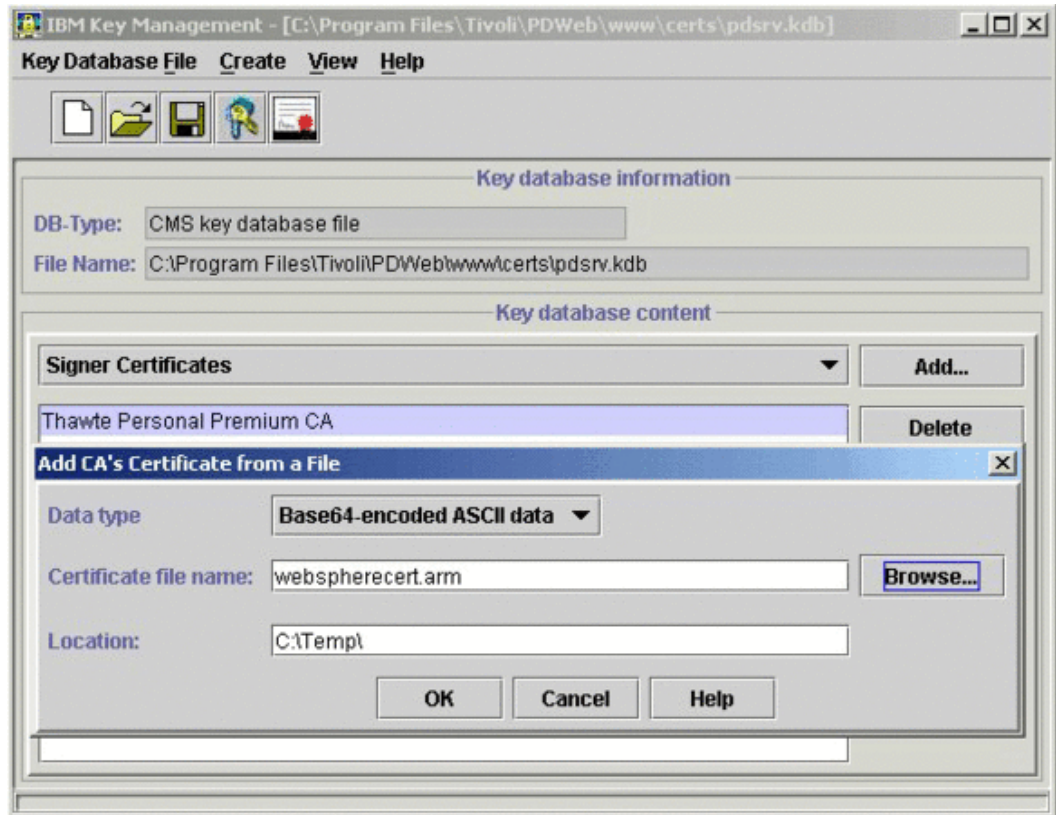


## Import SSL certificates into WebSEAL

Next, we'll import the WebSphere server certificate into WebSEAL.

1. Launch the key management utility for WebSEAL; it was installed together with the IBM HTTP Server.

2. Open the `webseald.kdb` file under the WebSEAL `certs` directory (`C:\Program Files\Tivoli\PDWeb\www\certs`); use the password `pdsrv`.
3. Add the certificate you have just exported from the WebSphere Server keystore as a signer certificate. Use the label name `WebSphereServer`.



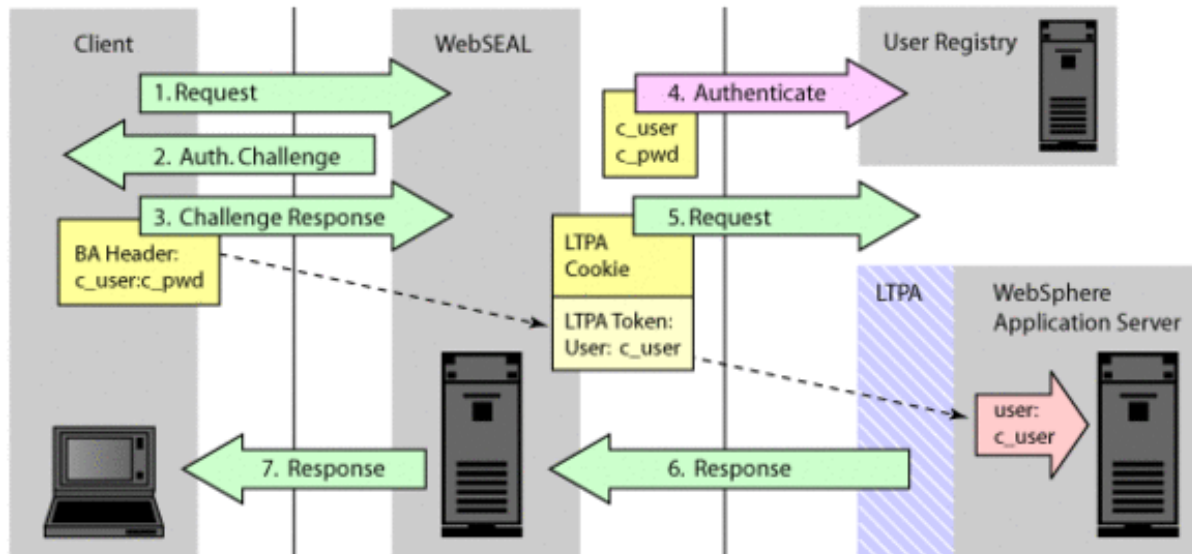
4. Close the application.
5. Restart WebSEAL. Open the Services window, then stop and start the Access Manager WebSEAL service.

---

## Section 6. Using LTPA to integrate WebSphere and WebSEAL

### What is LTPA?

LTPA stands for *lightweight third-party authentication*. It is an authentication mechanism that enables multiple or heterogeneous servers to share authentication information during a session; thus, the user does not have to reauthenticate when accessing another server.



LTPA uses an encrypted token to store user credentials. This token is passed between servers during the session. When a server receives the token, it decrypts the token first to get the user credentials, then uses the user information for authorization purposes. The process requires a trust setup between the servers that are passing the token back and forth, like the one we established in [Using WebSEAL to protect Web resources](#).

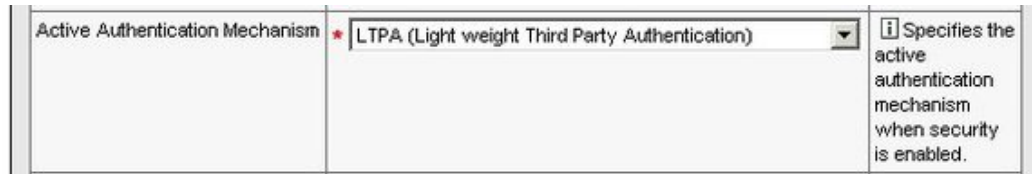
## WebSphere configuration

To configure LTPA for WebSphere and WebSEAL:

1. Launch the WebSphere Administrative Console. Navigate to **Security=>Authentication Mechanisms=>LTPA**.
2. Set the password and confirm password fields. We'll still use `passwd` for this example.
3. Click **Generate Keys**, then save the configuration for WebSphere.
4. Navigate to **Security=>Authentication Mechanisms=>LTPA** again.
5. Type in a file name and path in the Key File Name field -- `C:\WebSphere\Appserver\etc\ltpa.key`, for example. Click the

**Export Keys** button.

6. Navigate to **Security=>Global Security**.
7. Set the active authentication mechanism to **LTPA**.



8. Click **Apply**, then save the configuration for WebSphere.
9. Restart WebSphere.

## Configuring the WebSEAL junction

To configure a junction for WebSEAL with LTPA enabled:

1. Launch the Administration Command Prompt for Tivoli Access Manager. You can also use the Web Portal Manager for Tivoli Access Manager in your environment.
2. Type in `login`, then provide the username `sec_master` and password `passw0rd` at the command prompt.
3. Create the junction with the following command:

```
server task webseald-dwserver create -t ssl -b filter -A -F
"C:\WebSphere\Appserver\etc\ltpa.key" -Z "passw0rd" -h dwserver -p 9443 /ltpa
```

In this tutorial, we have the application server and WebSEAL running on the same system; therefore, in this example, the server name for the junction is the same name as that for the application server: `dwserver`. In a production environment, WebSEAL and the application server would probably run on different systems, or even on different networks. When creating a junction in a production environment, use the application server hostname with the `-h` option.

In addition, we've used `webseald-dwserver` as the name of the server task here. Replace that name with the real name of the task on your own server. You can get the name with the `server list` command.

## Testing LTPA

To determine if your system is properly using LTPA or not:

1. Open a Web browser and access the snoop servlet at `http://localhost:9080/snoop`. This link will go directly to the embedded Web server to access the servlet. Since the application is secured, and security is enabled for the application server, the browser will ask for a username and password. You can use the server ID `webuser01` and password `passw0rd` to log in.
2. Check the LTPA token, stored as a cookie on the client side. Type in `javascript:alert(document.cookie)` as the URL in your browser.
3. Close the browser, then open a new one and access the servlet at `https://localhost/ltpa/snoop`. Try the method from the previous step to get the LTPA token; you should not be able to see the cookie. The token is stored by WebSEAL this time, and WebSEAL maintains the session for the Web browser client and the application server.
4. You can also enable tracing for WebSphere, and look for the LTPA token in the trace file and see how it is used by WebSphere. Here's an example WebSphere trace; it's been modified in order to fit on the page, with some of the irrelevant details removed:

```
[4/14/03 12:01:36:446 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  getCookieValue parml=LtpaToken
[4/14/03 12:01:36:446 EDT] 20f1948c < UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  getCookieValue parml=9x8KFJ...jYA==
[4/14/03 12:01:36:446 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  A cookie was received. The name is LtpaToken and the value is 9x8...A==
[4/14/03 12:01:36:446 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  base64 ltpa token: parml=9x8KF...jYA==
[4/14/03 12:01:36:446 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  Validating the LTPA token that was retrieved from the cookie.
[4/14/03 12:01:36:446 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.web.WebAuthenticator
  validate
[4/14/03 12:01:36:446 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.util.Cache ...
  get
...
[4/14/03 12:01:36:476 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
  uid = null
[4/14/03 12:01:36:476 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
  realm = null
[4/14/03 12:01:36:476 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
  password = XXXXXXXX
[4/14/03 12:01:36:476 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.common.auth.util.Util ...
  toString(array)
```

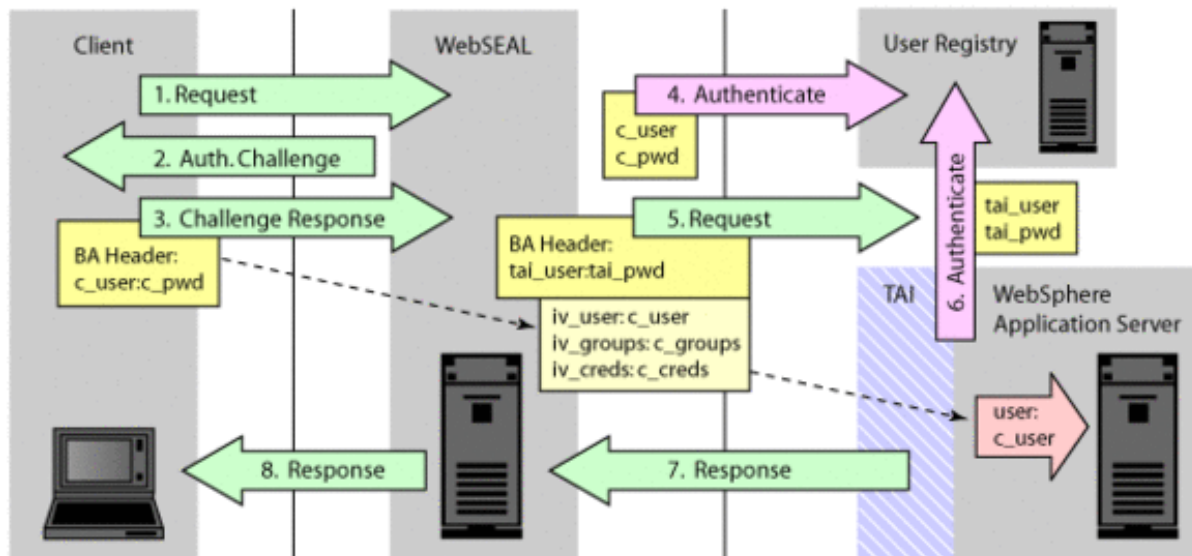
```
[4/14/03 12:01:36:476 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.common.auth.util.Util ...
array copied
[4/14/03 12:01:36:476 EDT] 20f1948c < UOW=
source=com.ibm.ws.security.common.auth.util.Util ...
toString(array)
[4/14/03 12:01:36:486 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
cred token = ...
[4/14/03 12:01:36:486 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
Successfully gathered authentication information
[4/14/03 12:01:36:486 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.server.lm.ltpaLoginModule ...
Using credential token for authentication
[4/14/03 12:01:36:486 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.ltpa.LTPAServerObject ...
validate
[4/14/03 12:01:36:496 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.ltpa.LTPAToken ...
LTPAToken 2
[4/14/03 12:01:36:496 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.ltpa.LTPAToken ...
user:dwserver:389/cn=webuser01,o=mycompany Expiration time: 03.04.14...
[4/14/03 12:01:36:496 EDT] 20f1948c < UOW=
source=com.ibm.ws.security.ltpa.LTPAToken ...
LTPAToken 2
[4/14/03 12:01:36:506 EDT] 20f1948c d UOW=
source=com.ibm.ws.security.ltpa.LTPAServerObject ...
validation successful - to create credential
[4/14/03 12:01:36:506 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.ltpa.LTPAServerObject ...
getSecurityName parml=user:dwserver:389/cn=webuser01,o=mycompany
[4/14/03 12:01:36:506 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.registry.UserRegistryImpl ...
getRealm
[4/14/03 12:01:36:506 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.registry ldap.LdapRegistryImpl ...
getRealm
[4/14/03 12:01:36:506 EDT] 20f1948c < UOW=
source=com.ibm.ws.security.registry ldap.LdapRegistryImpl ...
getRealm parml=dwserver:389
[4/14/03 12:01:36:506 EDT] 20f1948c < UOW=
source=com.ibm.ws.security.registry.UserRegistryImpl ...
getRealm parml=dwserver:389
[4/14/03 12:01:36:506 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.registry.UserRegistryImpl ...
getUserSecurityName parml=cn=webuser01,o=mycompany
[4/14/03 12:01:36:506 EDT] 20f1948c > UOW=
source=com.ibm.ws.security.registry ldap.LdapRegistryImpl ...
getUserSecurityName parml=cn=webuser01,o=mycompany
```

---

## Section 7. Using TAI to integrate WebSphere and WebSEAL

## What is TAI?

TAI stands for *trust association interceptor*. It is a mechanism that lets a server forward user credentials to the WebSphere Application Server.



Using TAI, WebSEAL puts the user credential information into the HTTP header and sends it to WebSphere together with the original request. The application server captures the request, and the interceptor module, specific to WebSEAL, has to extract the information from the HTTP header and insert it into the security context. In order to use TAI, you have to set up TAI for WebSphere and create a junction for WebSEAL.

## Configuring WebSphere

To configure TAI for WebSphere:

1. Launch the WebSphere Administrative Console. Navigate to **Security=>Authentication Mechanisms=>LTPA**.
2. Select the Trust Association link at the bottom of the page.
3. Check the enabled button for the Trust Association Enabled option, then click **Apply**.


[LTPA](#) >

### Trust Association

Enable Trust Association. Trust Association is used to connect reversed proxies to Websphere. 

**Configuration**

**General Properties**

Trust Association Enabled	<input checked="" type="checkbox"/>	 Enables Trust Association.
---------------------------	-------------------------------------	--

**Additional Properties**

<a href="#">Interceptors</a>	Specifies a list of Trust Association Interceptor implementations.
------------------------------	--

4. Select the Interceptors link, then select the `com.ibm.ws.security.web.WebSealTrustAssociationInterceptor` link.
5. Select the Custom Properties link, then set the following properties (keeping in mind that property names are case sensitive):

Key	Value
<code>com.ibm.websphere.security.webseal.hostnames</code>	<code>dwserver</code>
<code>com.ibm.websphere.security.webseal.ports</code>	<code>443</code>
<code>com.ibm.websphere.security.webseal.id</code>	<code>iv-user</code>
<code>com.ibm.websphere.security.webseal.loginId</code>	<code>webseal01</code>

Make sure you use your own hostname instead of `dwserver` for the first property.

6. Save the configuration for WebSphere.
7. Restart the application server to enable the interceptor.

## Configuring WebSEAL

To configure TAI for WebSEAL:

1. Launch the Administration Command Prompt for Tivoli Access Manager. You can also use the Web Portal Manager for Tivoli Access Manager in your environment.

2. Type in `login`, then provide the username `sec_master` and password `passw0rd` at the command prompt.
3. Create the junction with the following command:

```
server task webseald-dwserver create -t ssl -B -U "webseal01" -W  
"passw0rd" -c iv_user -h dwserver -p 9443 /tai
```

In this tutorial, we have the application server and WebSEAL running on the same machine; therefore, in this example, the server name for the junction is the same machine name as that for the application server: `dwserver`. In a production environment, WebSEAL and the application server would probably run on different machines, or even on different networks. When creating a junction in a production environment, use the application server hostname with the `-h` option.

In addition, we've used `webseald-dwserver` as the name of the server task here. You need to replace it with the real name of the task on your own server. Get the name using the `server list` command.

## Testing TAI

The following three steps will help you to test your TAI configuration and determine if your system is using TAI properly or not.

1. Test the TAI connection using the snoop servlet from the default application that comes with WebSphere. Launch a browser and type in `https://localhost/tai/snoop` as the URL.
2. The browser will ask for a valid username and password; use `webuser01` and `passw0rd`. Once the results of the snoop servlet come up within the browser, check the Remote User and the Principal fields. They should reflect the `webuser01` value.
3. You can also enable tracing for WebSphere and look for the WebSEAL TAI invocation in the trace file to see how it is used by WebSphere. Here's an example WebSphere trace; it's been modified in order to fit on the page, with some of the irrelevant details removed:

```
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=  
source=com.ibm.ws.security.web.WebAuthenticator ...  
TrustAssociation is enabled.  
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=  
source=com.ibm.ws.security.web.TrustAssociationManager ...  
getInterceptor  
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
```

```

source=com.ibm.ws.security.web.TrustAssociationManager ...
  Check if target interceptor ...
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  getCheckID
[4/14/03 18:19:11:860 EDT] 2d8c1d6 < UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  getCheckID
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=authorization
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=via
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: VIA=HTTP/1.1 dwserver:443
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  checkVia for dwserver:443
[4/14/03 18:19:11:860 EDT] 2d8c1d6 < UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  getCheckID: 0
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=user-agent
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=host
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=accept
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=connection
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=accept-language
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=iv-user
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  isTargetInteceptor: header name=accept-encoding
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  Yes, it is via WebSeal.
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  A TrustAssociation interceptor is available for this request.
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  Entering validateEstablishedTrust...
[4/14/03 18:19:11:860 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
  Going to authenticate webseal01.
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
  basicAuthenticate
[4/14/03 18:19:11:860 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.util.Cache ...
  get
[4/14/03 18:19:11:870 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.util.Cache ...
  Key = com.ibm.ws.security.util.CredentialCache$BasicAuthCacheData@f1c34512,
  key class is com.ibm.ws.security.util.CredentialCache$BasicAuthCacheData
[4/14/03 18:19:11:870 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.util.Cache ...
  com.ibm.ws.security.util.CredentialCache number of entries: 0

```

```
[4/14/03 18:19:11:870 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.util.CredentialCache ...
    update
[4/14/03 18:19:11:870 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.util.CredentialCache ...
    formCredential
[4/14/03 18:19:11:870 EDT] 2d8c1d6 d UOW=
source=SASRas ...
    [PrincipalAuthenticatorImpl.authenticate], [ServerID: server1]
    Beginning to authenticate principal: webseal01.
...
[4/14/03 18:19:11:940 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
    Successful authentication for validateEstablishedTrust.
[4/14/03 18:19:11:940 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
    TrustAssociation has been validated successfully.
[4/14/03 18:19:11:940 EDT] 2d8c1d6 > UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
    getAuthenticatedUsername
[4/14/03 18:19:11:940 EDT] 2d8c1d6 < UOW=
source=com.ibm.ws.security.web.WebSealTrustAssociationInterceptor ...
    Exiting getAuthenticatedUsername: webuser01
[4/14/03 18:19:11:940 EDT] 2d8c1d6 d UOW=
source=com.ibm.ws.security.web.WebAuthenticator ...
    Username retrieved is [webuser01]
```

---

## Section 8. Summary

In this tutorial, you have learned to use the following WebSphere Application Server and Tivoli Access Manager scenarios:

1. You've shared a user registry between Access Manager and WebSphere. Both components used the same LDAP directory for a user registry. The registry is maintained from Access Manager.
2. You've used security Web resources with WebSEAL using LTPA (lightweight third-party authentication). WebSEAL sits at the front of the application server and takes care of the authentication on behalf of the application server. This scenario illustrates how to use LTPA to pass the authentication information to the server.
3. In a solution similar to the previous one, you've had WebSEAL perform the authentication, with the system using TAI (trust association interceptor) to propagate the user information from the security reverse-proxy server to the application server.

You should now have a better idea of how to use Tivoli Access Manager to add

security features to your own WebSphere applications.

# Resources

## Learn

- IBM WebSphere Application Server V5.0 [product documentation and InfoCenter](#)
- Tivoli Access Manager V4.1 [product documentation](#)
- *IBM WebSphere V5.0 Security WebSphere Handbook Series* , a Redbook from IBM.
- *Securing Web portals with Tivoli Access Manager* , Paul Ashley and Sridhar Muppidi (developerWorks, June 2002).
- *Third-party security servers and WebSphere* , Nataraj Nagaratnam, Wilfred Jamison, and Gennaro Cuomo (developerWorks, July 2001).
- Stay current with [developerWorks technical events and Webcasts](#).

## Get products and technologies

- Build your next development project with [IBM trial software](#), available for download directly from developerWorks.

## Discuss

- [Participate in the discussion forum for this content.](#)

# About the author

## Peter Kovari

Peter Kovari is a WebSphere specialist at the International Technical Support Organization, Raleigh Center in Research Triangle Park, NC. He writes extensively about all areas of WebSphere. His areas of expertise include WebSphere Application Server, WebSphere Application Server Enterprise, enterprise application design and development, security, enterprise messaging, and pervasive solutions.