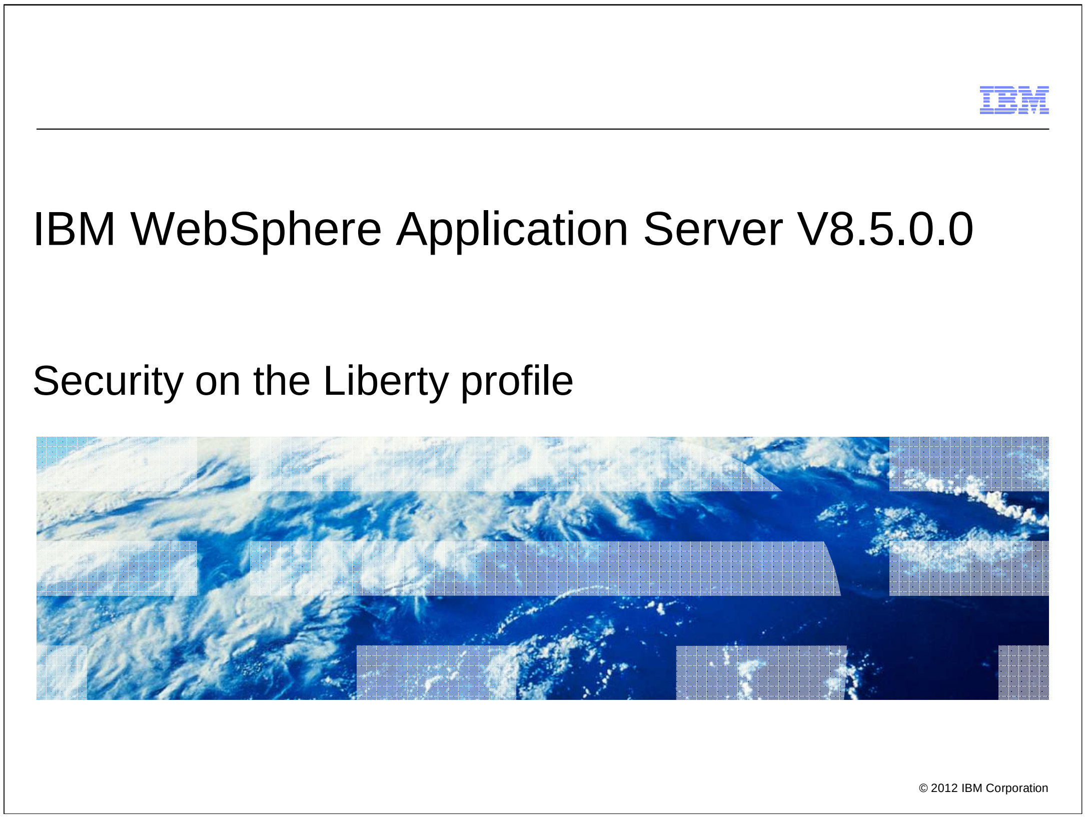
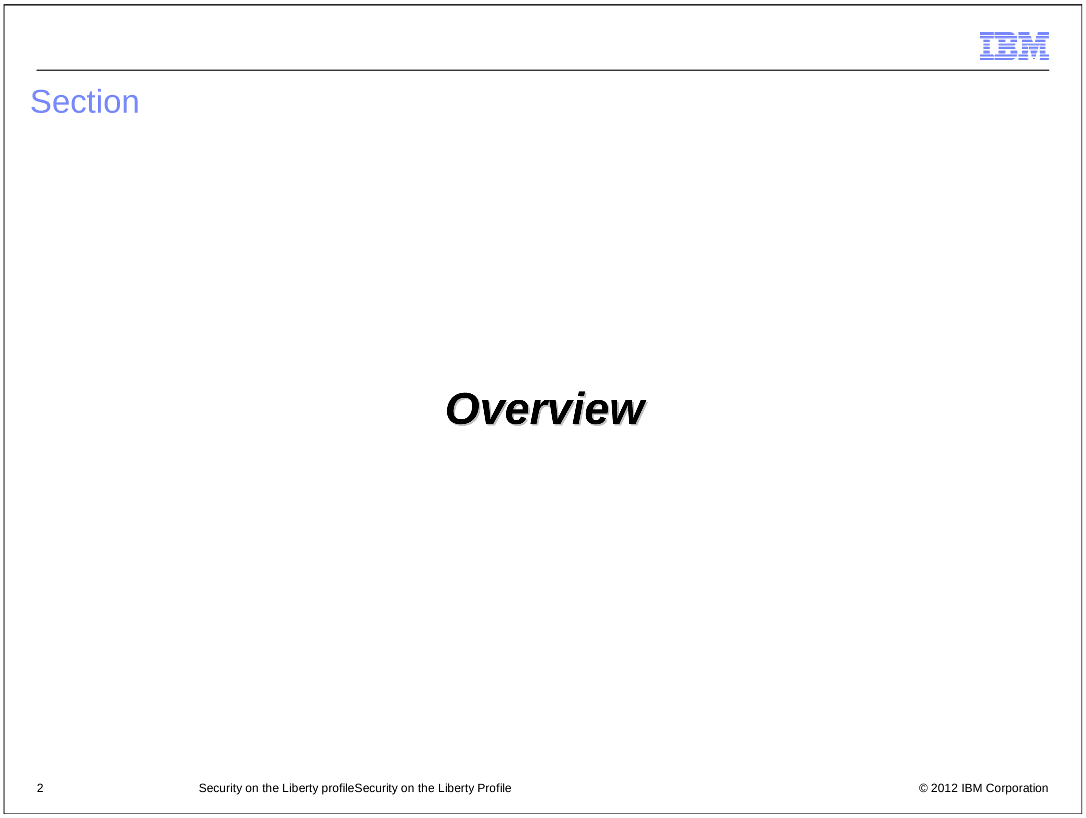


IBM WebSphere Application Server V8.5.0.0

Security on the Liberty profile

This presentation describes support for security in the Liberty profile included in IBM WebSphere Application Server V8.5

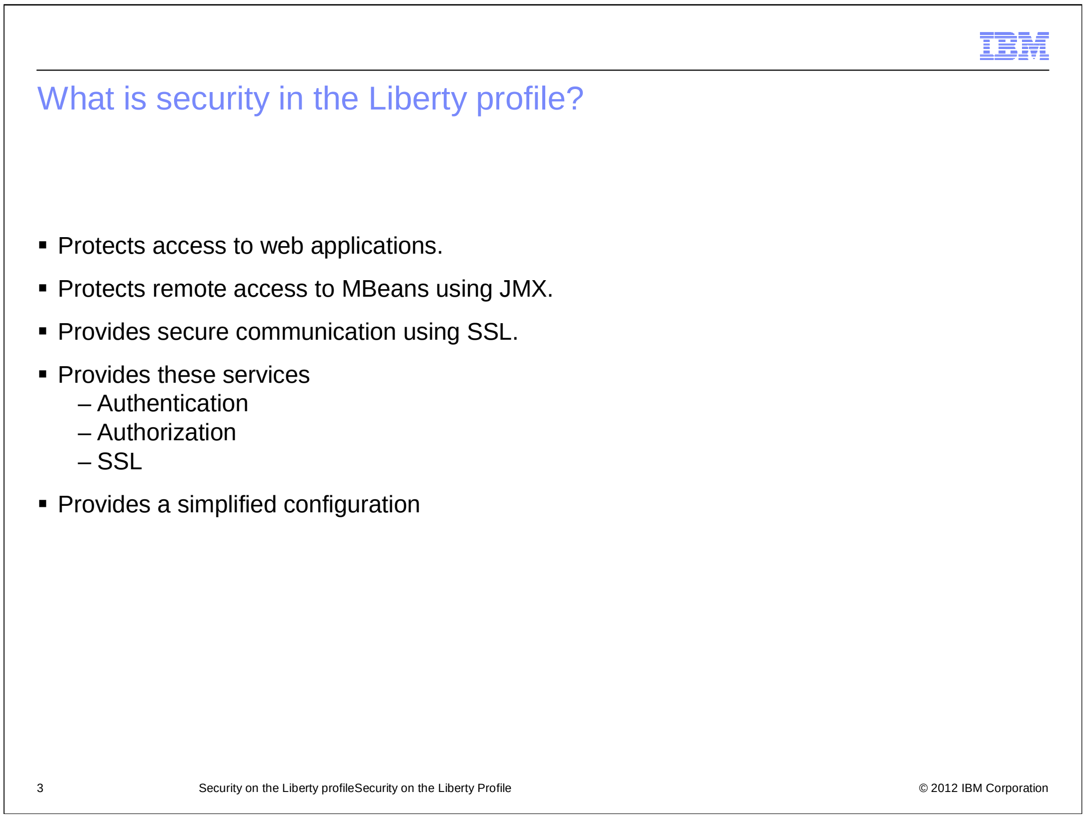


Section

Overview

2 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

The security feature will protect the web application resources against unauthorized access. It will also protect the remote access to MBeans using JMX.

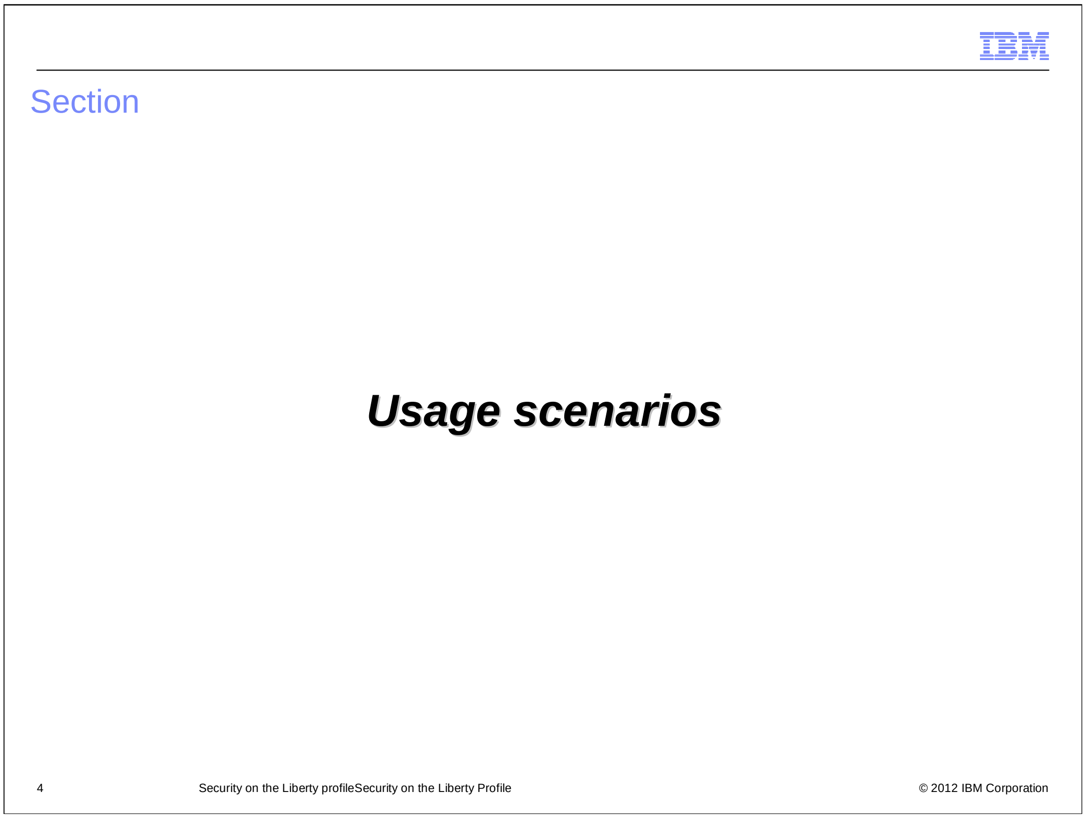


What is security in the Liberty profile?

- Protects access to web applications.
- Protects remote access to MBeans using JMX.
- Provides secure communication using SSL.
- Provides these services
  - Authentication
  - Authorization
  - SSL
- Provides a simplified configuration

3 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

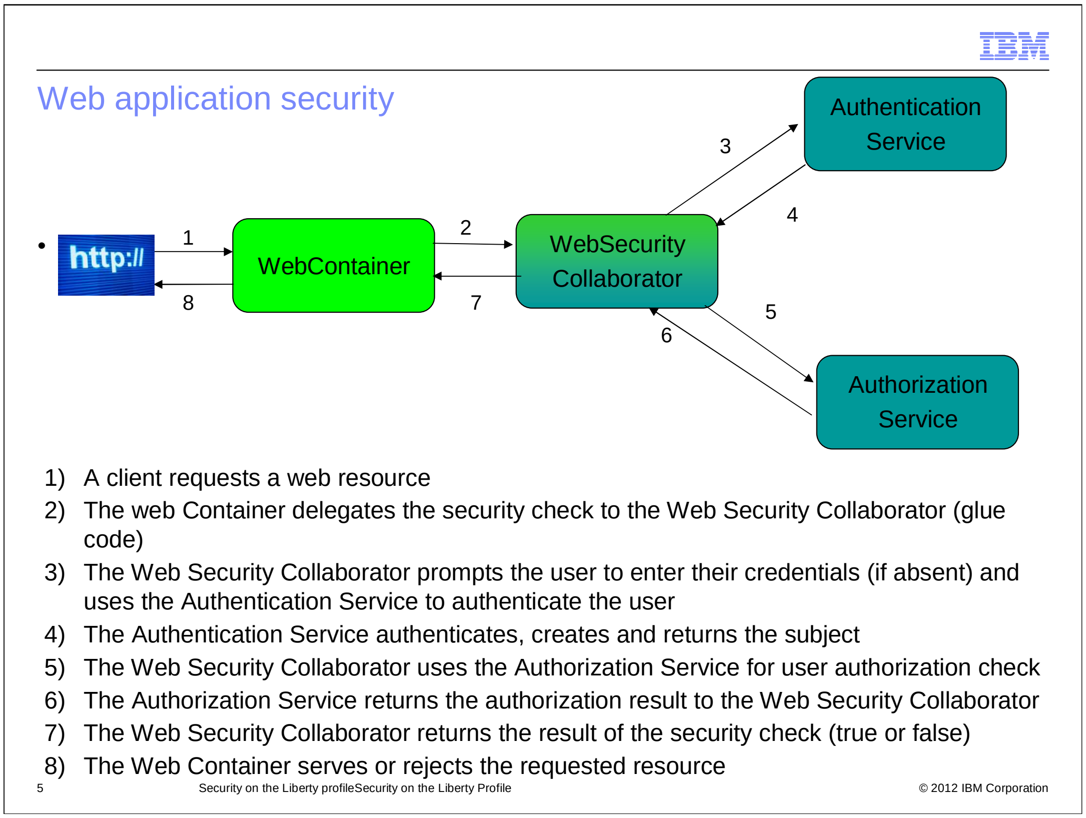
Security in the Liberty Profile provides services to protect your applications against unauthorized access. It supports the Servlet 3.0 security requirements. Provides capability to handle different user registries and default configuration to handle authentication.



Section

Usage scenarios

Security in Liberty is used in these scenarios.

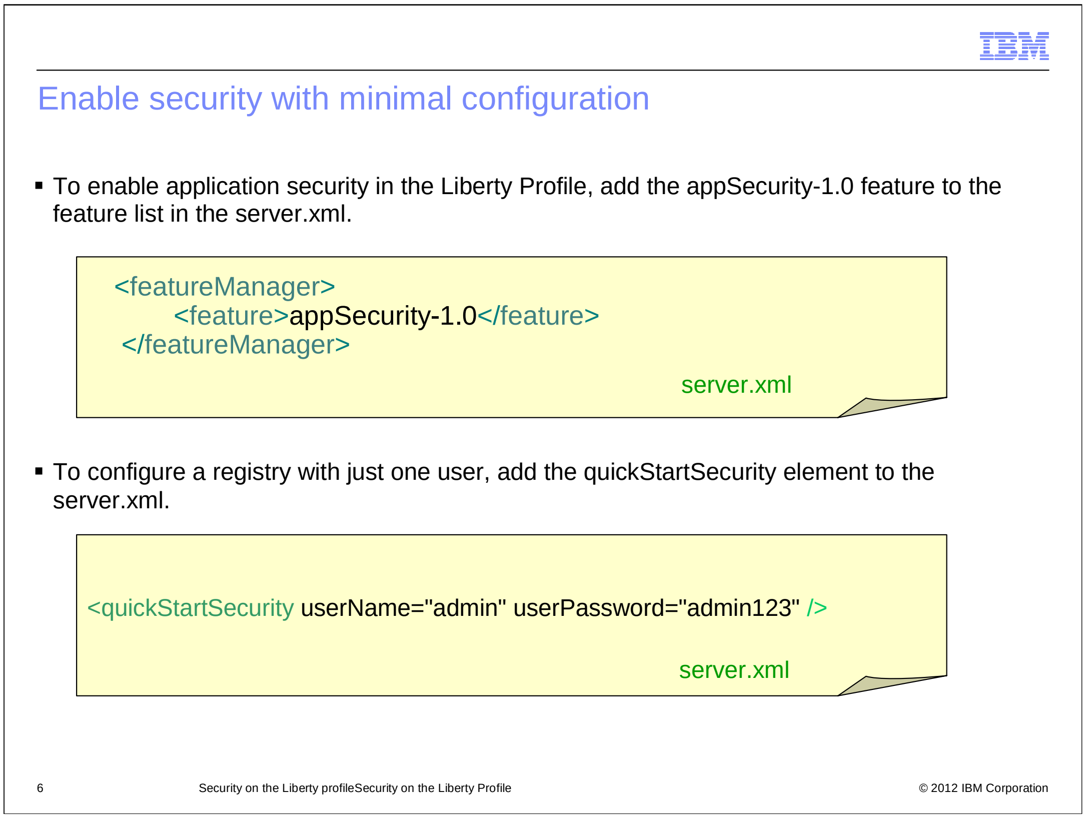


Web application security

1) A client requests a web resource
2) The web Container delegates the security check to the Web Security Collaborator (glue code)
3) The Web Security Collaborator prompts the user to enter their credentials (if absent) and uses the Authentication Service to authenticate the user
4) The Authentication Service authenticates, creates and returns the subject
5) The Web Security Collaborator uses the Authorization Service for user authorization check
6) The Authorization Service returns the authorization result to the Web Security Collaborator
7) The Web Security Collaborator returns the result of the security check (true or false)
8) The Web Container serves or rejects the requested resource

High level view of the application security process.

When security is configured, and a web resource is being accessed, the security runtime will perform the authentication and the authorization checks on a protected resource to enforce security.



Enable security with minimal configuration

- To enable application security in the Liberty Profile, add the appSecurity-1.0 feature to the feature list in the server.xml.

```
<featureManager>
        <feature>appSecurity-1.0</feature>
 </featureManager>
```

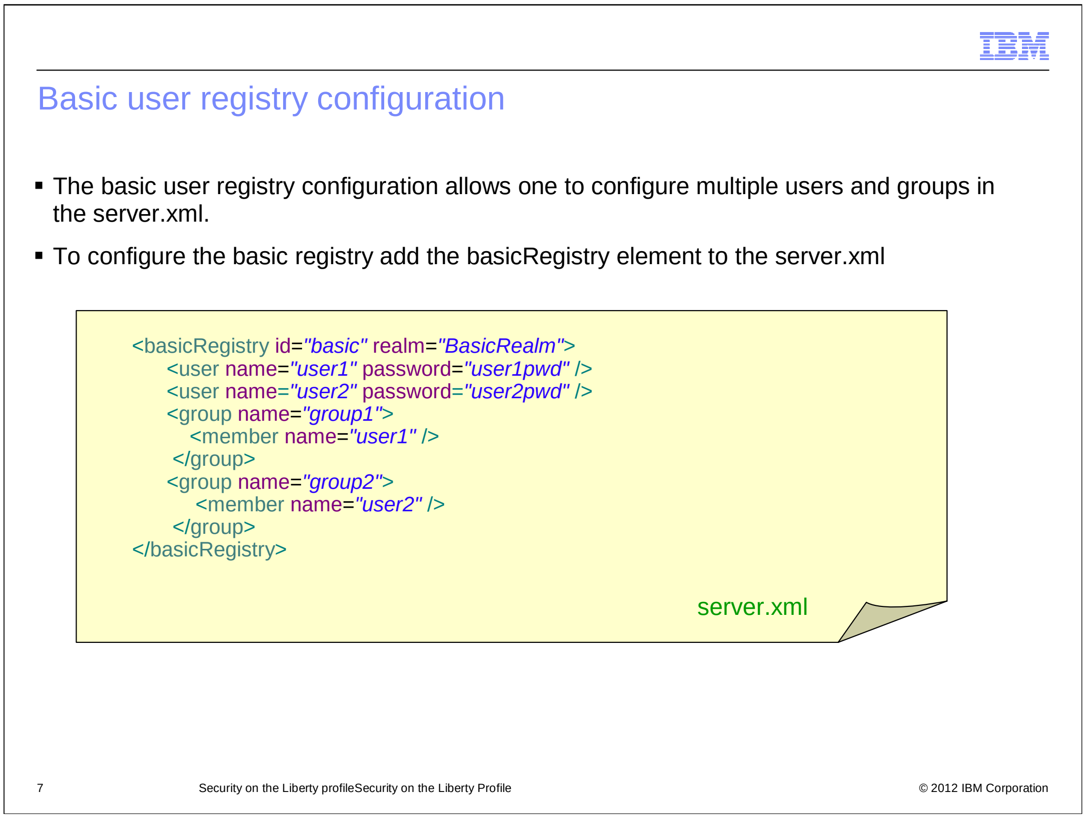
server.xml

- To configure a registry with just one user, add the quickStartSecurity element to the server.xml.

```
<quickStartSecurity userName="admin" userPassword="admin123" />
```

server.xml

This simple configuration will enable the security feature and creates a registry with single user called admin with password admin123. You should encode the password using the securityUtility encode utility. More information on this utility can be found in the references section. Once you configure this, you can protect your applications with a set of roles and associate the user “admin” to this role so that only that user can access them. If you have the restConnector-1.0 feature configured in the server.xml, the user “admin” is associated with the administrator role by default.



Basic user registry configuration

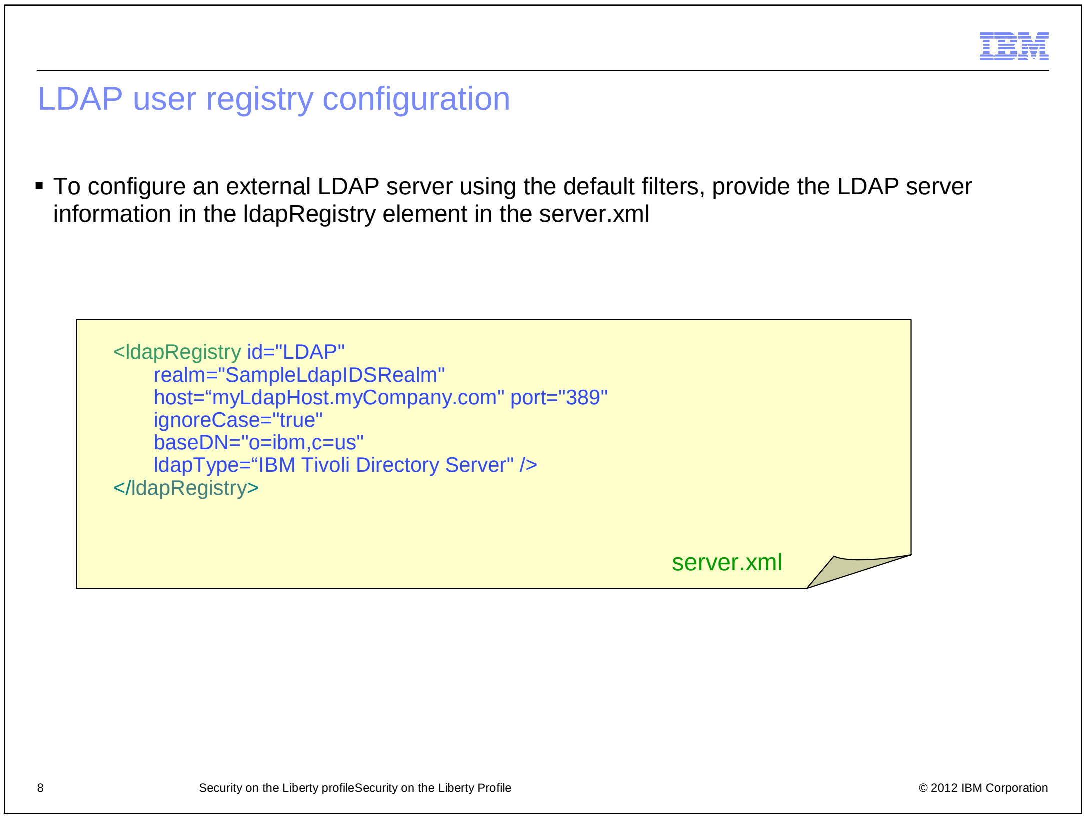
- The basic user registry configuration allows one to configure multiple users and groups in the server.xml.
- To configure the basic registry add the basicRegistry element to the server.xml

```
<basicRegistry id="basic" realm="BasicRealm">
     <user name="user1" password="user1pwd" />
     <user name="user2" password="user2pwd" />
     <group name="group1">
         <member name="user1" />
      </group>
     <group name="group2">
          <member name="user2" />
      </group>
</basicRegistry>
```

server.xml

This simple basic registry configuration will create two users (user1 and user2) and two groups (group1 and group2) and associate the users to the groups. Note encode the passwords shown here using the securityUtility encode utility. More information on this utility can be found in the information center.

Once you configure the basic registry, you can protect your applications with these users and groups.



LDAP user registry configuration

- To configure an external LDAP server using the default filters, provide the LDAP server information in the ldapRegistry element in the server.xml

```
<ldapRegistry id="LDAP"
      realm="SampleLdapIDSRealm"
      host=“myLdapHost.myCompany.com" port="389"
      ignoreCase="true"
      baseDN="o=ibm,c=us"
      ldapType=“IBM Tivoli Directory Server" />
</ldapRegistry>
```

server.xml

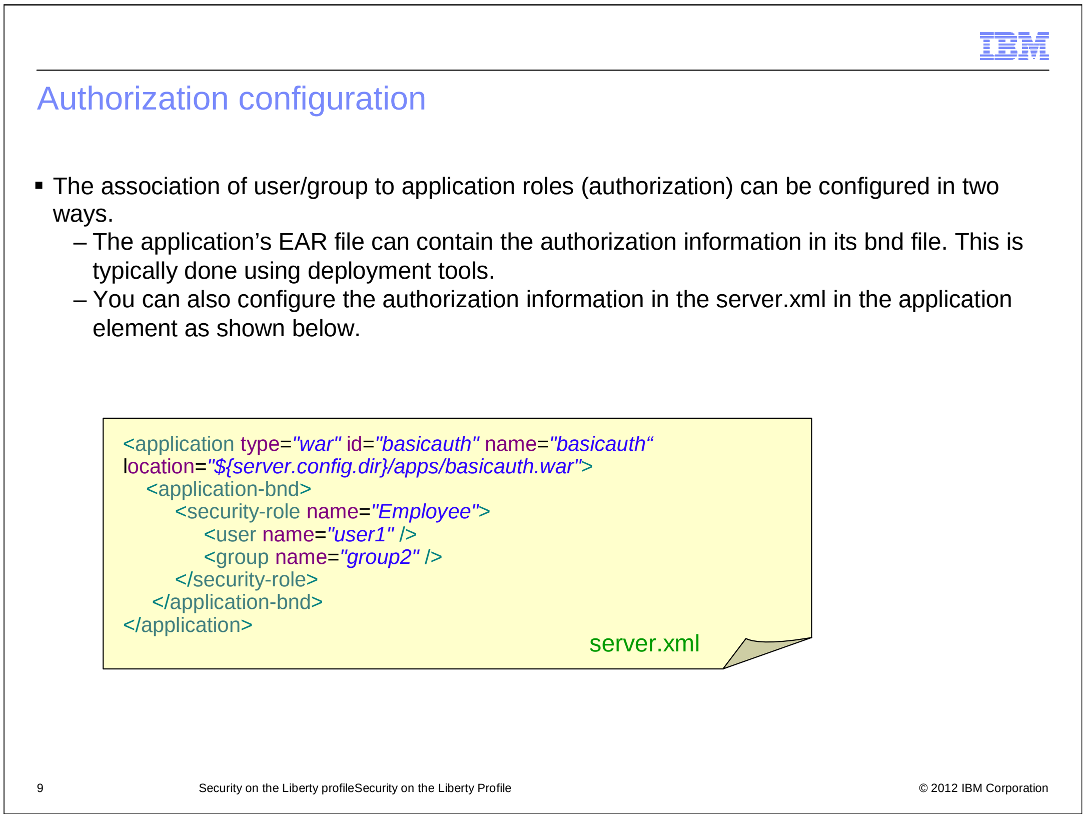
8 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

This LDAP registry configuration will let the security runtime use the user and group information in the this LDAP server. Different types of LDAP servers are supported.

This configuration assumes that you are using the default filters for user and group information. You can override the default if you choose to.

Optionally, you can also enable SSL connection to the LDAP server.

For more information on the various LDAP server configurations, see the information center.



Authorization configuration

- The association of user/group to application roles (authorization) can be configured in two ways.
  - The application’s EAR file can contain the authorization information in its bnd file. This is typically done using deployment tools.
  - You can also configure the authorization information in the server.xml in the application element as shown below.

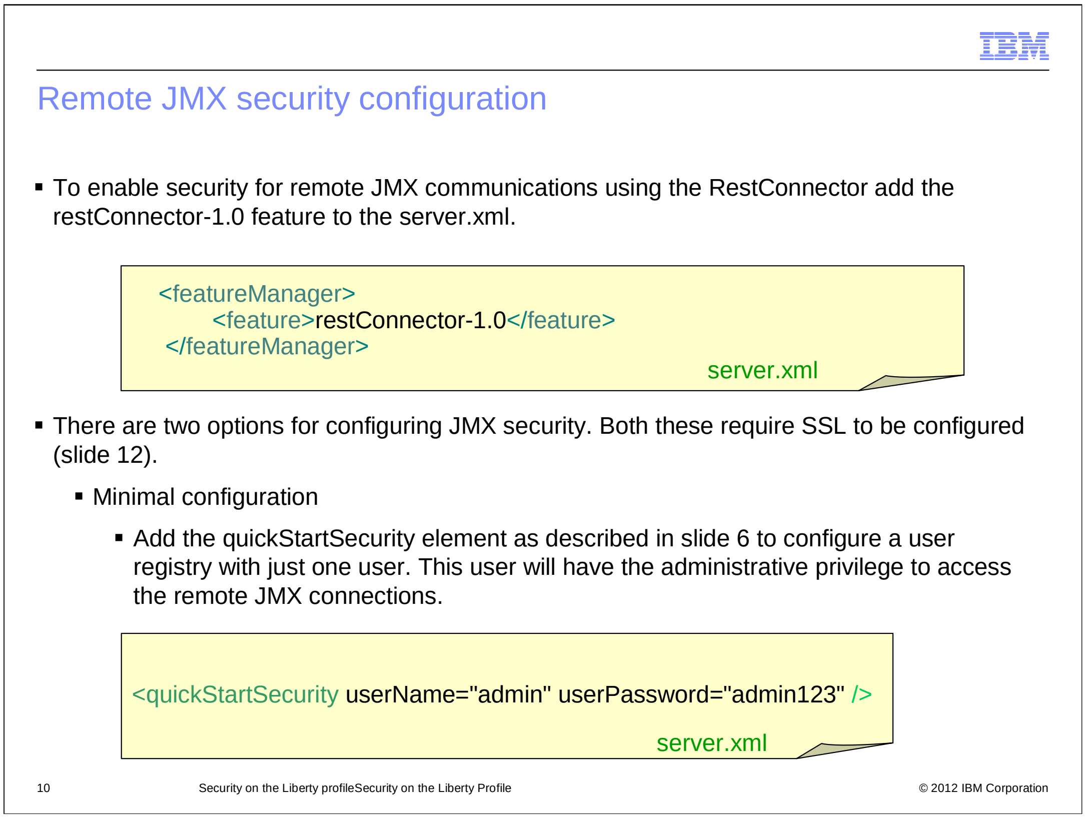
```
<application type="war" id="basicauth" name="basicauth“
location="${server.config.dir}/apps/basicauth.war">
   <application-bnd>
       <security-role name="Employee">
           <user name="user1" />
           <group name="group2" />
       </security-role>
    </application-bnd>
</application>
```

server.xml

In this example, user1 and the all the users in group2 are able access the resources protected by the Employee role

If the authorization information exists in both the bnd file and the server.xml it is merged with the server.xml taking precedence.

For more information on the authorization support see the information center.



Remote JMX security configuration

- To enable security for remote JMX communications using the RestConnector add the restConnector-1.0 feature to the server.xml.

```
<featureManager>
       <feature>restConnector-1.0</feature>
 </featureManager>
```

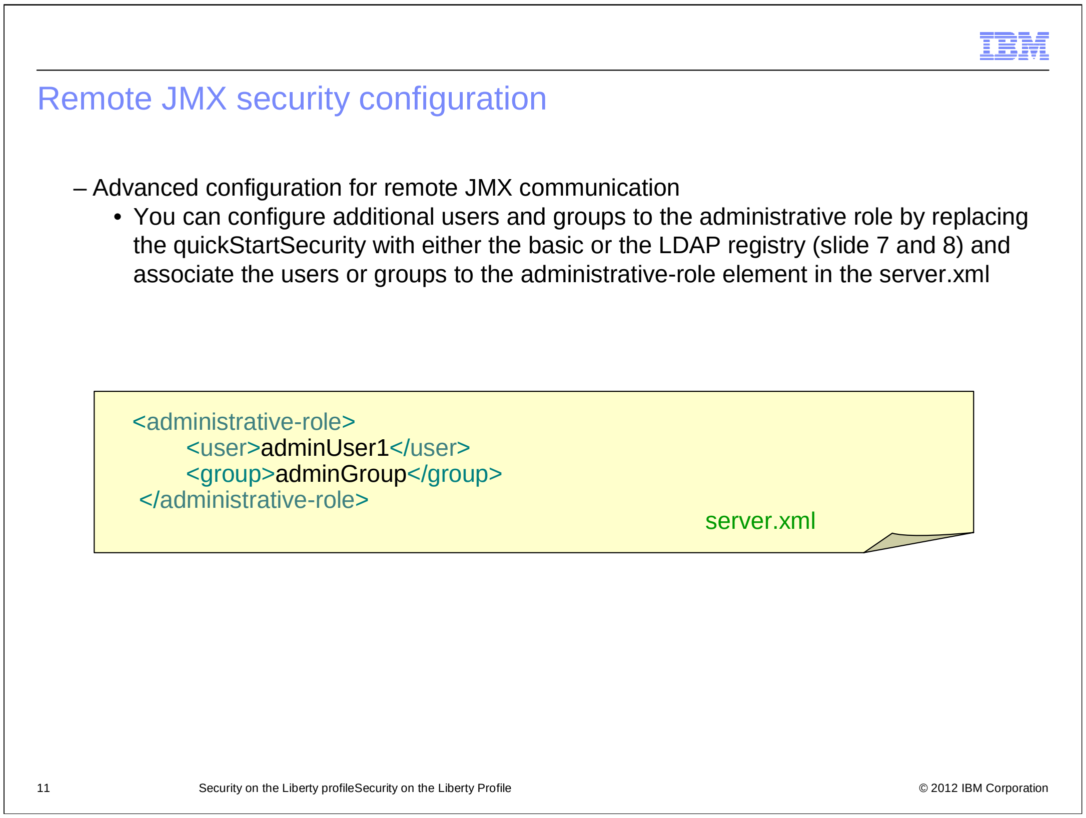
server.xml

- There are two options for configuring JMX security. Both these require SSL to be configured (slide 12).
  - Minimal configuration
    - Add the quickStartSecurity element as described in slide 6 to configure a user registry with just one user. This user will have the administrative privilege to access the remote JMX connections.

```
<quickStartSecurity userName="admin" userPassword="admin123" />
```

server.xml

The restConnector-1.0 feature provides the services to enforce security for remote communications to MBeans and is required to be configured for all remote JMX connections. If there is only administrative user, you can use the quickStartSecurity to configure it. The user configured in the quickStartSecurity element is automatically added to the administrative-role by the security runtime so no additional authorization configuration is needed. Remote JMX connections also requires SSL to be configured so restConnector-1.0 feature includes the ssl-1.0 feature. More information on the SSL configuration can be found in the next slides. If you are using only the localConnector for JMX (localConnector-1.0) you do not need to specify the security configuration.



Remote JMX security configuration

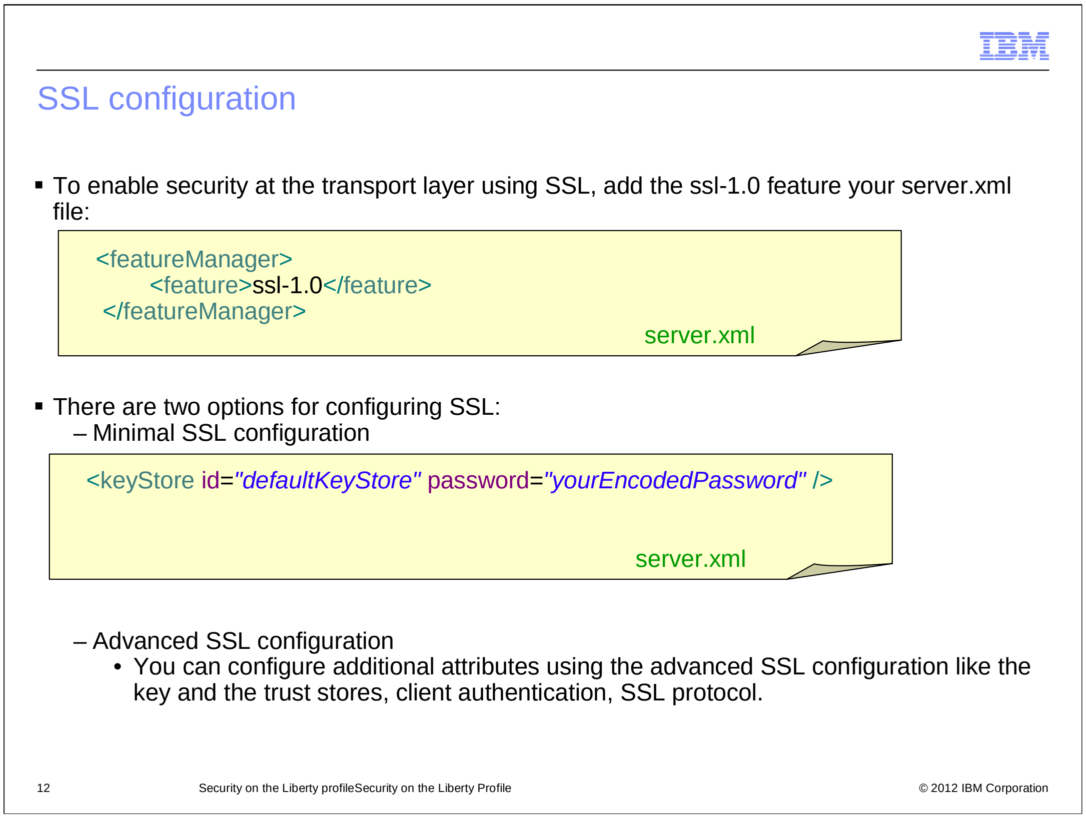
– Advanced configuration for remote JMX communication
  - You can configure additional users and groups to the administrative role by replacing the quickStartSecurity with either the basic or the LDAP registry (slide 7 and 8) and associate the users or groups to the administrative-role element in the server.xml

```
<administrative-role>
        <user>adminUser1</user>
        <group>adminGroup</group>
 </administrative-role>
```

server.xml

11 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

If you want to associate multiple users or groups to the administrative-role, you need to configure a registry (either a basic or LDAP) and associate the users to the administrative-role element in the server.xml.



SSL configuration

- To enable security at the transport layer using SSL, add the ssl-1.0 feature your server.xml file:

```
<featureManager>
        <feature>ssl-1.0</feature>
 </featureManager>
```

server.xml

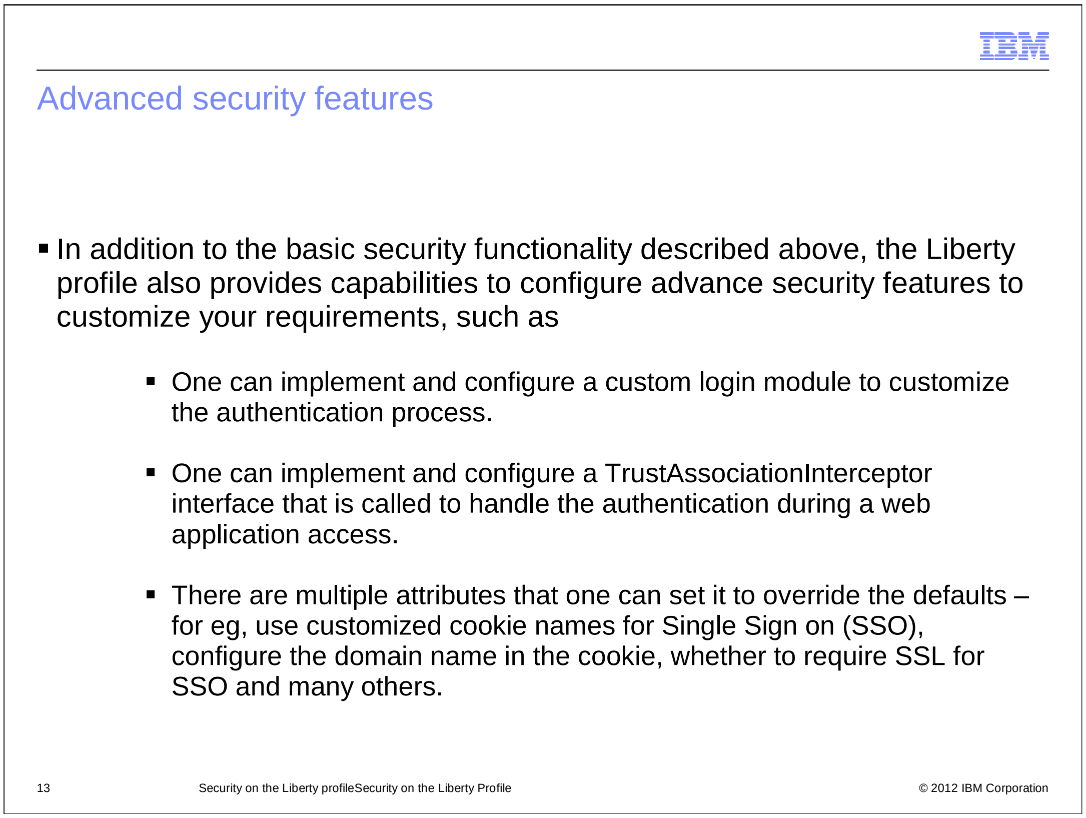
- There are two options for configuring SSL:
  – Minimal SSL configuration

```
<keyStore id="defaultKeyStore" password="yourEncodedPassword" />
```

server.xml

  – Advanced SSL configuration
    - You can configure additional attributes using the advanced SSL configuration like the key and the trust stores, client authentication, SSL protocol.

In the minimal SSL configuration, the password attribute is encoded. Use the securityUtility encode command to encode the password. The server will create a keystore called key.jks in the servers resources/security directory if it does not exist during SSL initialization. A self-signed certificate will get created and added to the keystore. The SSL protocol is set to SSL_TLS, the 128bit and higher cipher suites is used, and client authentication is disabled. For Advanced SSL configuration, see the information center.



Advanced security features

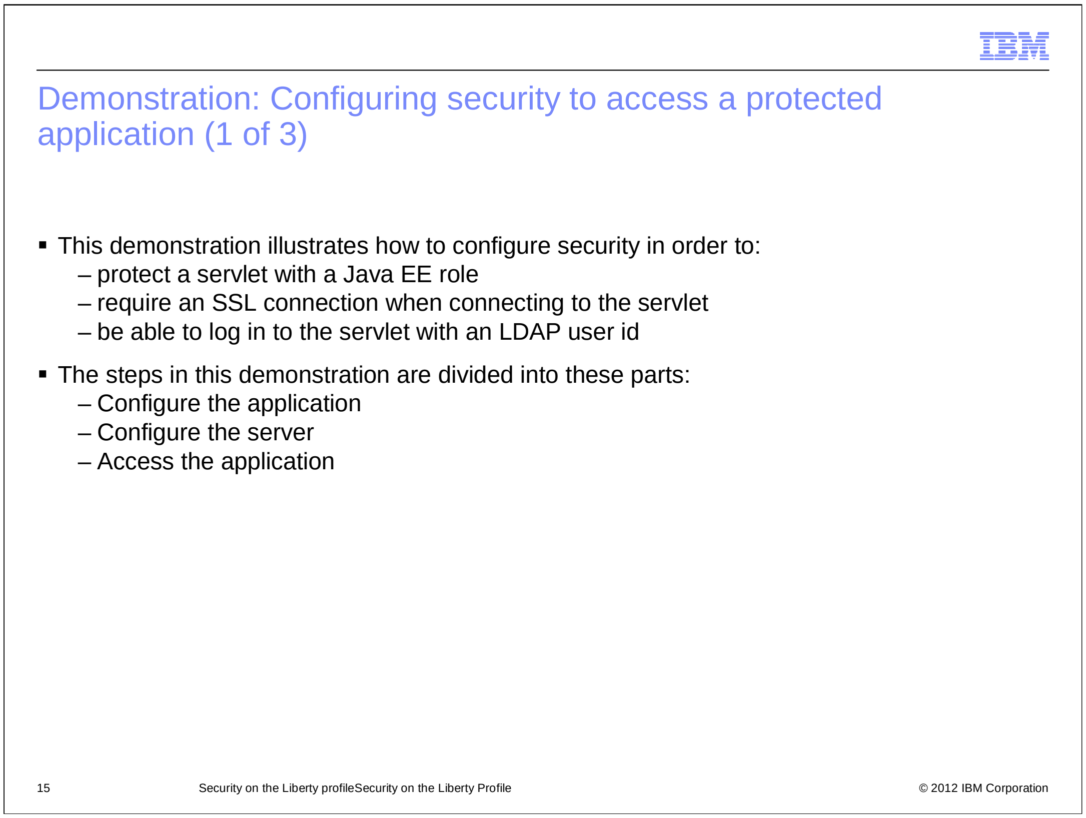
- In addition to the basic security functionality described above, the Liberty profile also provides capabilities to configure advance security features to customize your requirements, such as
  - One can implement and configure a custom login module to customize the authentication process.
  - One can implement and configure a TrustAssociationInterceptor interface that is called to handle the authentication during a web application access.
  - There are multiple attributes that one can set it to override the defaults – for eg, use customized cookie names for Single Sign on (SSO), configure the domain name in the cookie, whether to require SSL for SSO and many others.

To configure these advanced options, use the appropriate elements and attributes as described in the information center links in the reference section.

Section

Demonstration

This slide signals the start of the demonstration.

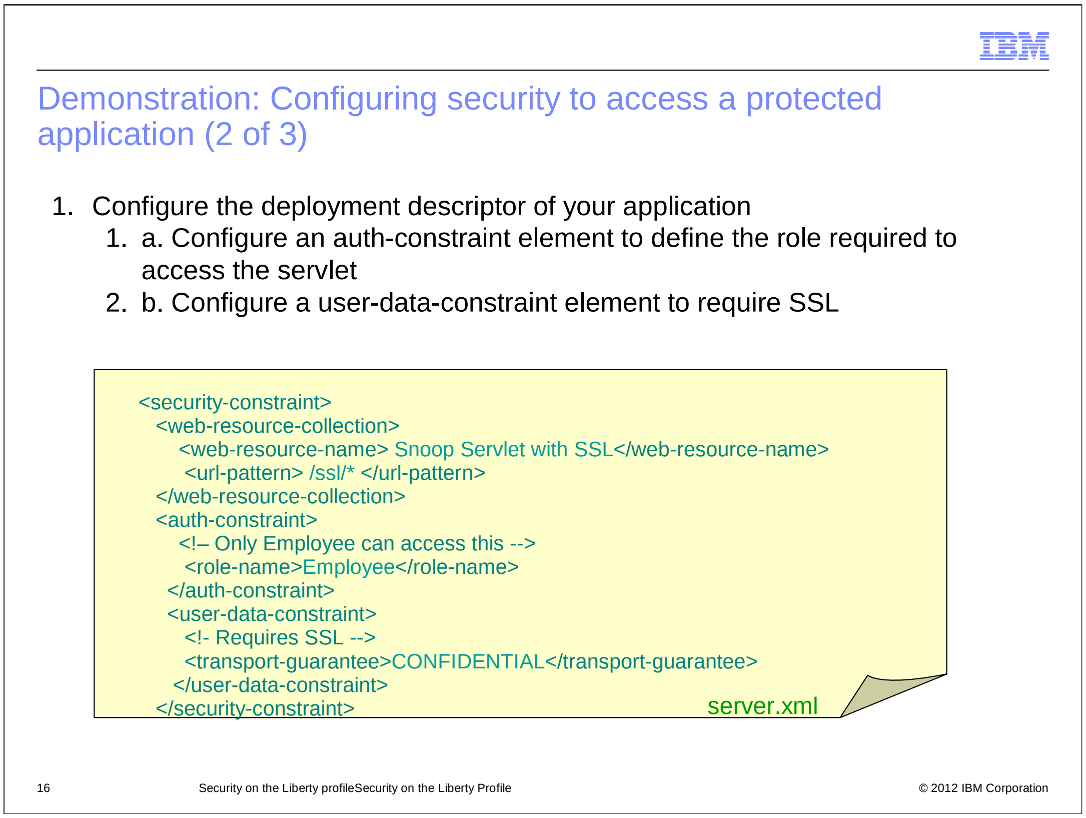


Demonstration: Configuring security to access a protected application (1 of 3)

- This demonstration illustrates how to configure security in order to:
  - protect a servlet with a Java EE role
  - require an SSL connection when connecting to the servlet
  - be able to log in to the servlet with an LDAP user id
- The steps in this demonstration are divided into these parts:
  - Configure the application
  - Configure the server
  - Access the application

15 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

The following demonstration illustrates an end-to-end scenario for accessing a protected servlet with security enabled on the Liberty profile. The servlet is configured to require SSL at the transport layer and for the user to be authenticated against an LDAP user registry, and have access to the role defined by the application. The demonstration walks you through the steps to configure your servlet, configure the server and finally test the configuration by accessing your servlet.



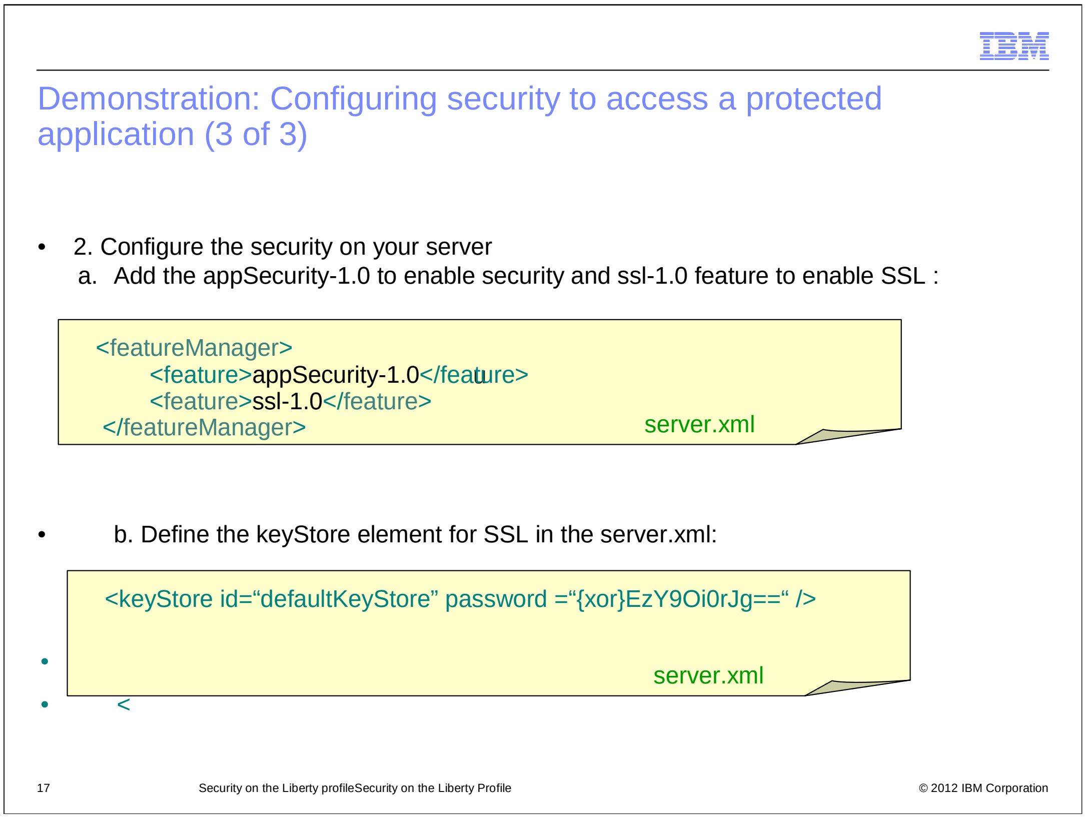
Demonstration: Configuring security to access a protected application (2 of 3)

1. Configure the deployment descriptor of your application
   1. a. Configure an auth-constraint element to define the role required to access the servlet
   2. b. Configure a user-data-constraint element to require SSL

```
<security-constraint>
   <web-resource-collection>
       <web-resource-name> Snoop Servlet with SSL</web-resource-name>
        <url-pattern> /ssl/* </url-pattern>
   </web-resource-collection>
   <auth-constraint>
       <!– Only Employee can access this -->
        <role-name>Employee</role-name>
     </auth-constraint>
     <user-data-constraint>
        <!- Requires SSL -->
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
      </user-data-constraint>
   </security-constraint>
```

server.xml

In Step 1, configure the application's deployment descriptor to define an auth-constraint element with the name of the role required to access the servlet. Then configure a user-data-constraint element to require SSL when accessing the servlet by specifying a value of CONFIDENTIAL. Follow the snippet of xml on this slide as an example.



Demonstration: Configuring security to access a protected application (3 of 3)

- 2. Configure the security on your server
  a. Add the appSecurity-1.0 to enable security and ssl-1.0 feature to enable SSL :

```
<featureManager>
       <feature>appSecurity-1.0</featuure>
       <feature>ssl-1.0</feature>
 </featureManager>
```

server.xml

- b. Define the keyStore element for SSL in the server.xml:

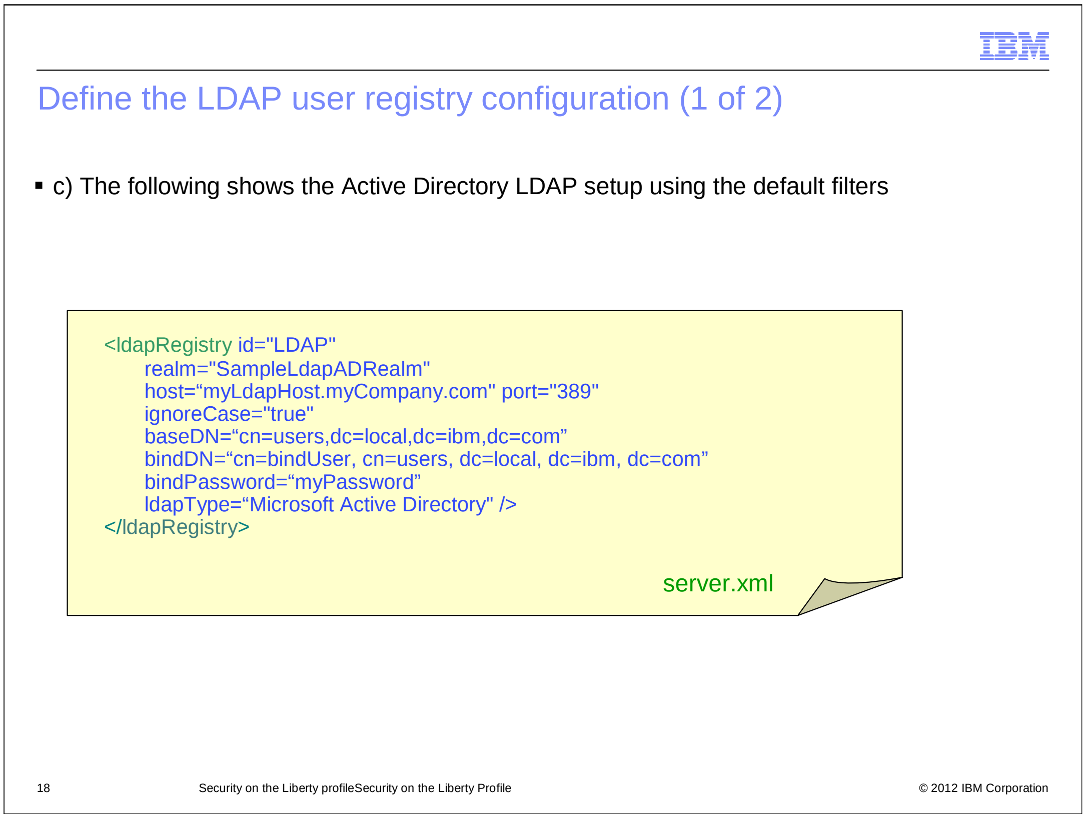
```
<keyStore id=“defaultKeyStore” password =“{xor}EzY9Oi0rJg==“ />
```

server.xml

-
- <

17 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

In Step 2, configure security on the server. Add the ssl-1.0 feature to enable SSL and add the appSecurity-1.0 feature to enable security. Add the keyStore element to the server.xml, specifying a password for the keystore. This keystore is created with the password you specified when the server is started for the first time. For security, the password should be longer than 6 characters and encoded using the securityUtility encode command. An encoded password is shown in the example.



Define the LDAP user registry configuration (1 of 2)

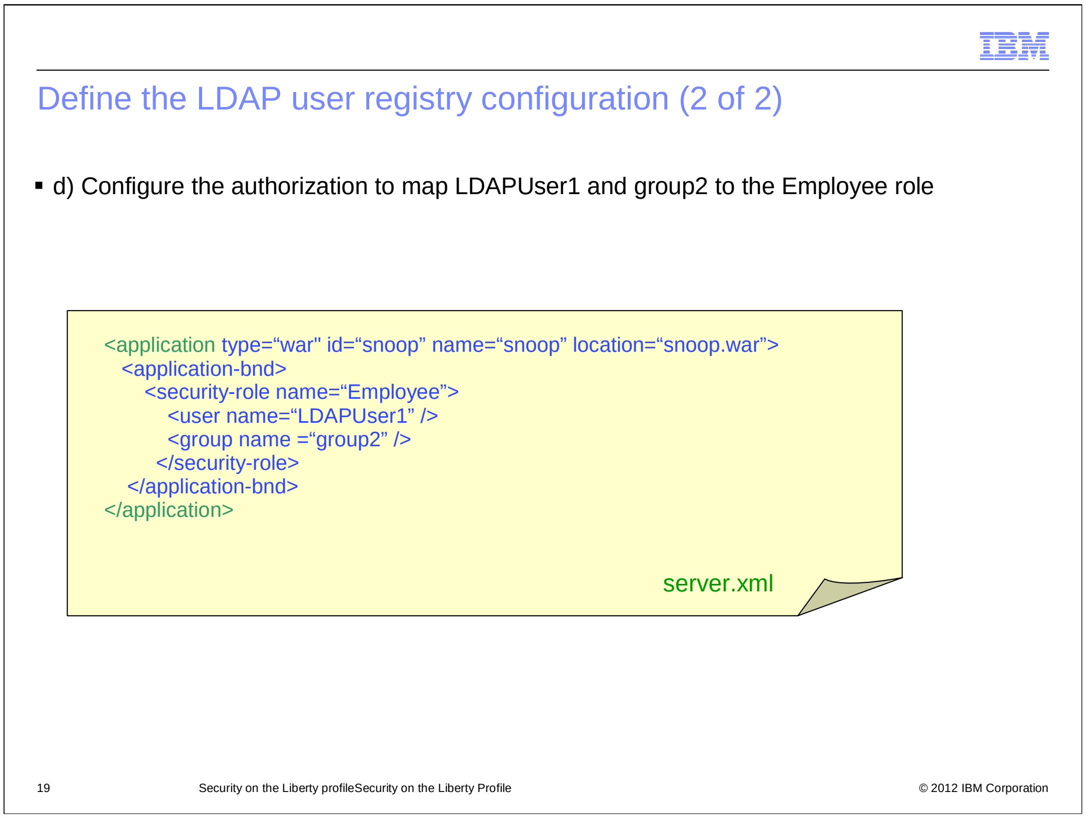
- c) The following shows the Active Directory LDAP setup using the default filters

```
<ldapRegistry id="LDAP"
      realm="SampleLdapADRealm"
      host=“myLdapHost.myCompany.com" port="389"
      ignoreCase="true"
      baseDN=“cn=users,dc=local,dc=ibm,dc=com”
      bindDN=“cn=bindUser, cn=users, dc=local, dc=ibm, dc=com”
      bindPassword=“myPassword”
      ldapType=“Microsoft Active Directory" />
</ldapRegistry>
```

server.xml

18 Security on the Liberty profileSecurity on the Liberty Profile © 2012 IBM Corporation

Configure the ldapRegistry element in the server.xml along with information about the LDAP server such as the host name and port. It is highly recommended to encode the password. This example uses an ActiveDirectory LDAP server.



Define the LDAP user registry configuration (2 of 2)

- d) Configure the authorization to map LDAPUser1 and group2 to the Employee role

```
<application type=“war” id=“snoop” name=“snoop” location=“snoop.war”>
  <application-bnd>
     <security-role name=“Employee”>
        <user name=“LDAPUser1” />
        <group name =“group2” />
      </security-role>
   </application-bnd>
</application>
```

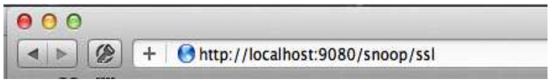
server.xml

Configure the authorization for the application by defining the application element in the server.xml. Under the application-bnd element, specify the roles using the security-role element and what users, groups and special subjects each role is mapped to.

IBM

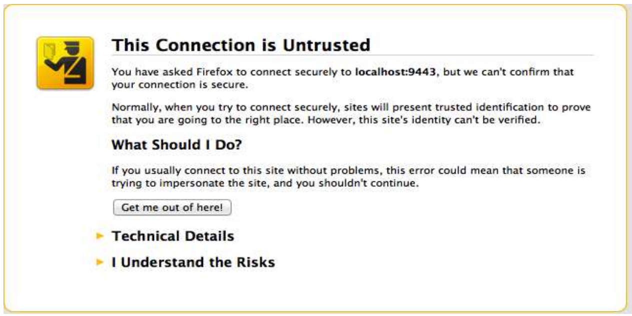
Demonstration: Configuring security to access a protected application (1 of 4)

- 3. Start the application server:
  - ./server start serverName
- 4. Access the protected servlet on the HTTP port:



http://localhost:9080/snoop/ssl

  - a. Note that the URL is redirected to the HTTPS port and you are prompted to trust the certificate:



This Connection is Untrusted

You have asked Firefox to connect securely to **localhost:9443**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

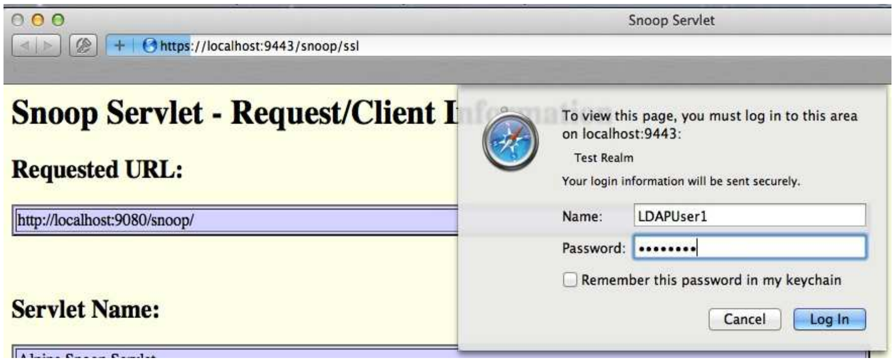
What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

In Step 3, start the Liberty profile server using the server start command. In Step 4, access the servlet on the HTTP port. Note that the request is redirected to the HTTPS port and that the browser prompts you to trust the certificate being presented by the server. The certificate is not trusted by default because it is not in your web browser's trust store.

Demonstration: Configuring security to access a protected application (2 of 4)

- 5. After trusting the certificate, you get prompted to enter a user and password to authenticate and be authorized to the required role
- 6. Enter the credentials for LDAPUser1


Snoop Servlet
https://localhost:9443/snoop/ssl
Snoop Servlet - Request/Client I
Requested URL:
http://localhost:9080/snoop/
Servlet Name:
Cancel
Log In

To view this page, you must log in to this area on localhost:9443:

Test Realm

Your login information will be sent securely.

Name: LDAPUser1

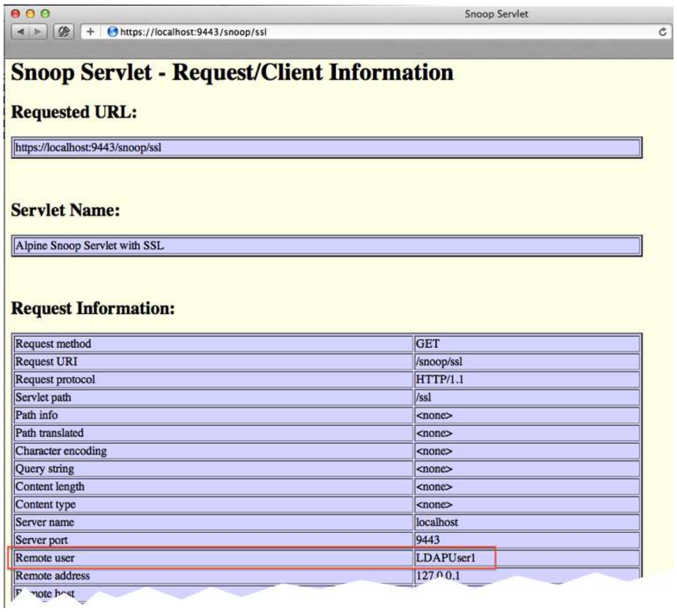
Password: ••••••••

□ Remember this password in my keychain

Once you trust the certificate, in Step 5 you will get prompted to enter a user and password. In Step 6, enter credentials for the LDAP user that is authorized to the role required by the servlet.

Demonstration: Configuring security to access a protected application (3 of 4)

- 7. The servlet is displayed.
- a. Note in the servlet output
- that the remote user is
- LDAPUser1



Snoop Servlet - Request/Client Information

Requested URL:

https://localhost:9443/snoop/ssl

Servlet Name:

Alpine Snoop Servlet with SSL

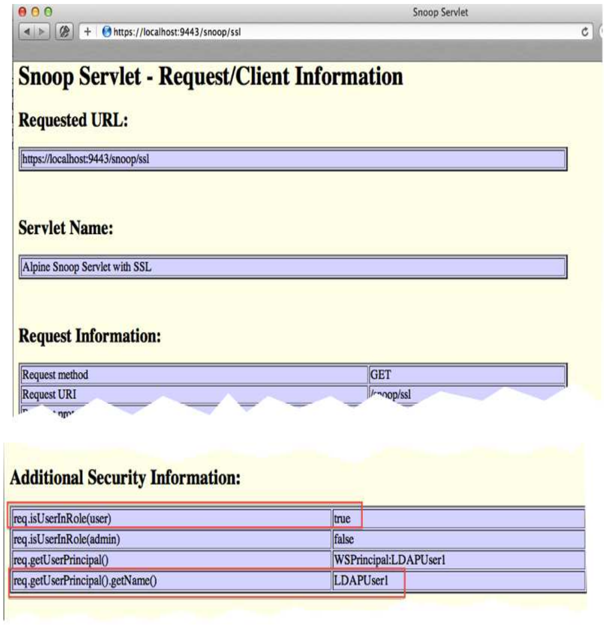
Request Information:

Request method	GET
Request URI	/snoop/ssl
Request protocol	HTTP/1.1
Servlet path	/ssl
Path info	<none>
Path translated	<none>
Character encoding	<none>
Query string	<none>
Content length	<none>
Content type	<none>
Server name	localhost
Server port	9443
Remote user	LDAPUser1
Remote address	127.0.0.1
Remote host	

In Step 7, the results of the servlet are displayed. Note that the remote user in the request information is set to the user that you logged in to the servlet with.

Demonstration: Configuring security to access a protected application (4 of 4)

- b. Note in the servlet output
- that the user principal is
- LDAPUser1 and that
- it is in the user role



Snoop Servlet - Request/Client Information

Requested URL:

https://localhost:9443/snoop/ssl

Servlet Name:

Alpine Snoop Servlet with SSL

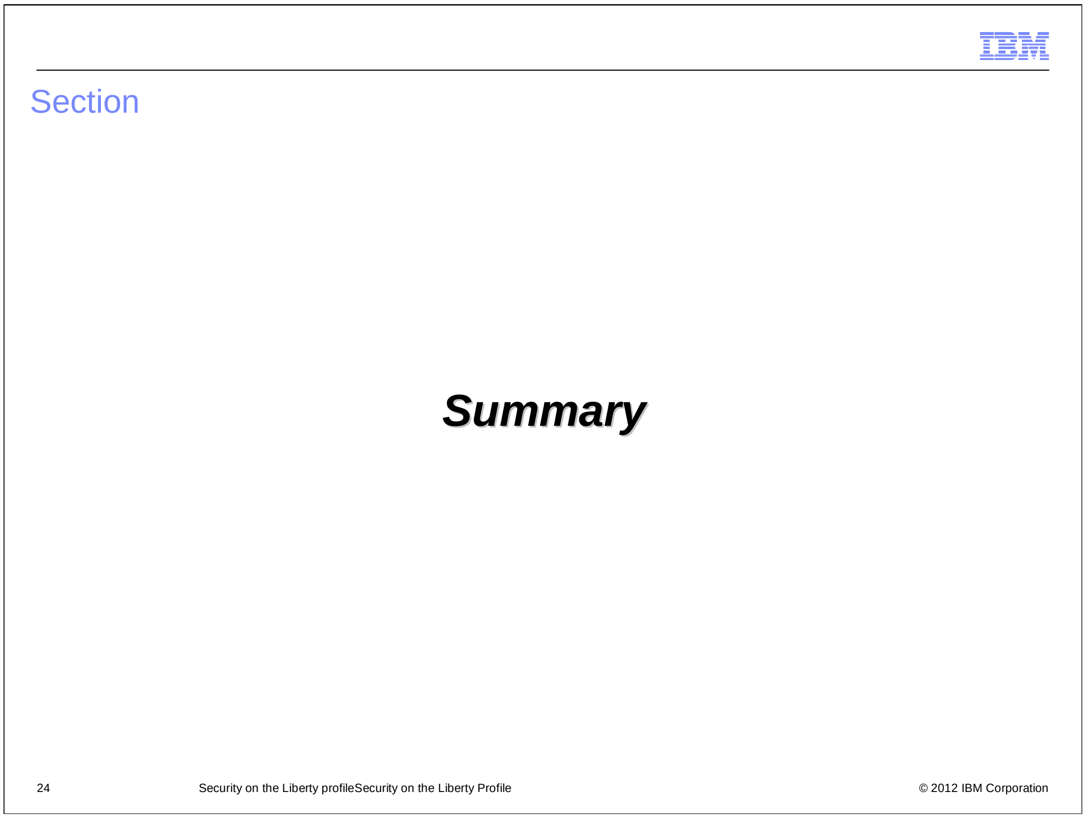
Request Information:

Request method	GET
Request URI	/snoop/ssl

Additional Security Information:

req.isUserInRole(user)	true
req.isUserInRole(admin)	false
req.getUserPrincipal()	WSPrincipal:LDAPUser1
req.getUserPrincipal().getName()	LDAPUser1

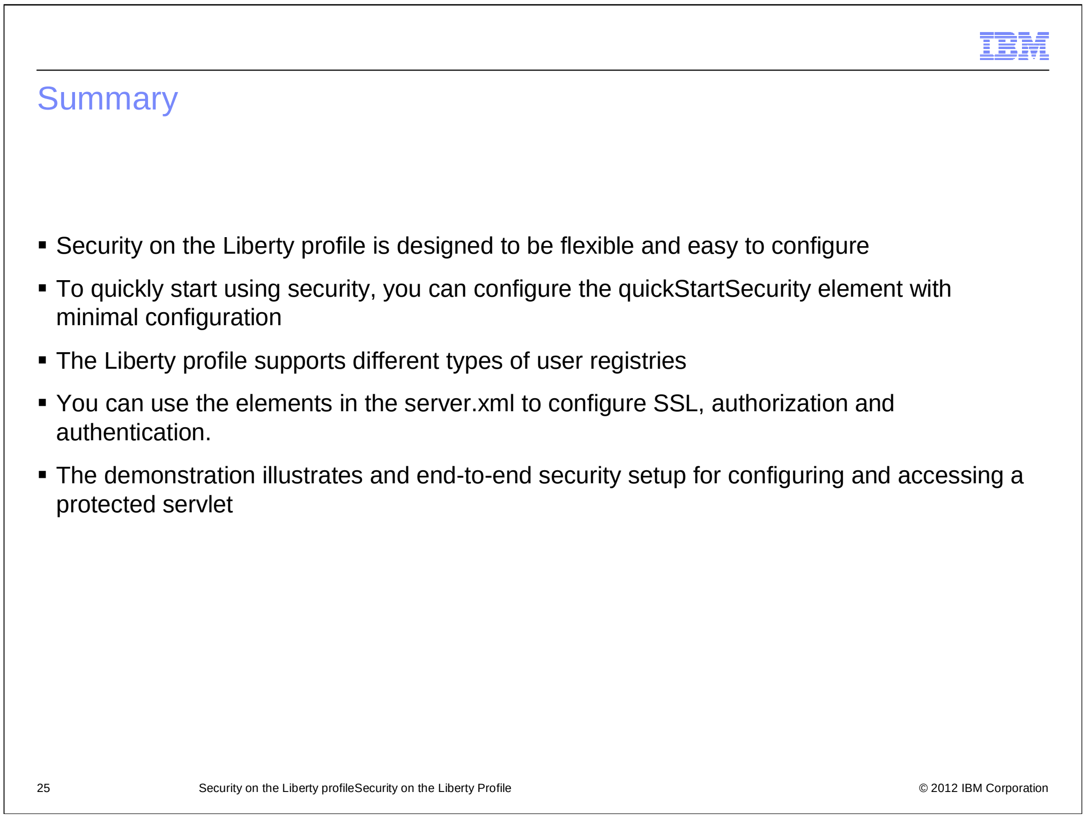
Further note in the output of the servlet that the API method isUserInRole() returns true for the role required by the application. The getUserPrincipal() API returns the principal of the user you logged in to the servlet with.



Section

Summary

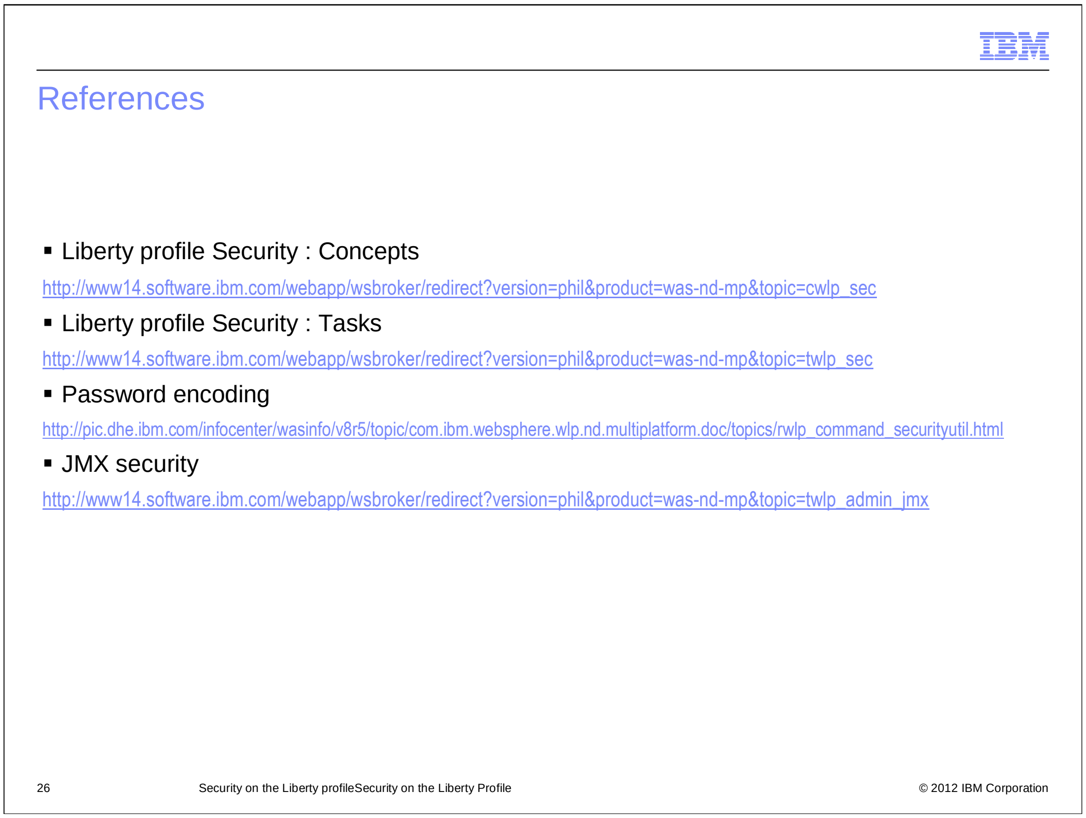
The following section presents a summary of this presentation.



Summary

- Security on the Liberty profile is designed to be flexible and easy to configure
- To quickly start using security, you can configure the quickStartSecurity element with minimal configuration
- The Liberty profile supports different types of user registries
- You can use the elements in the server.xml to configure SSL, authorization and authentication.
- The demonstration illustrates and end-to-end security setup for configuring and accessing a protected servlet

As you have seen in this presentation, the basic security setup requires minimal configuration. Only when you require advanced capabilities like custom login modules and TrustAssociationInterceptor one needs to configure additional data in the server.xml.



References

- Liberty profile Security : Concepts

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=cwlp_sec

- Liberty profile Security : Tasks

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=twlp_sec

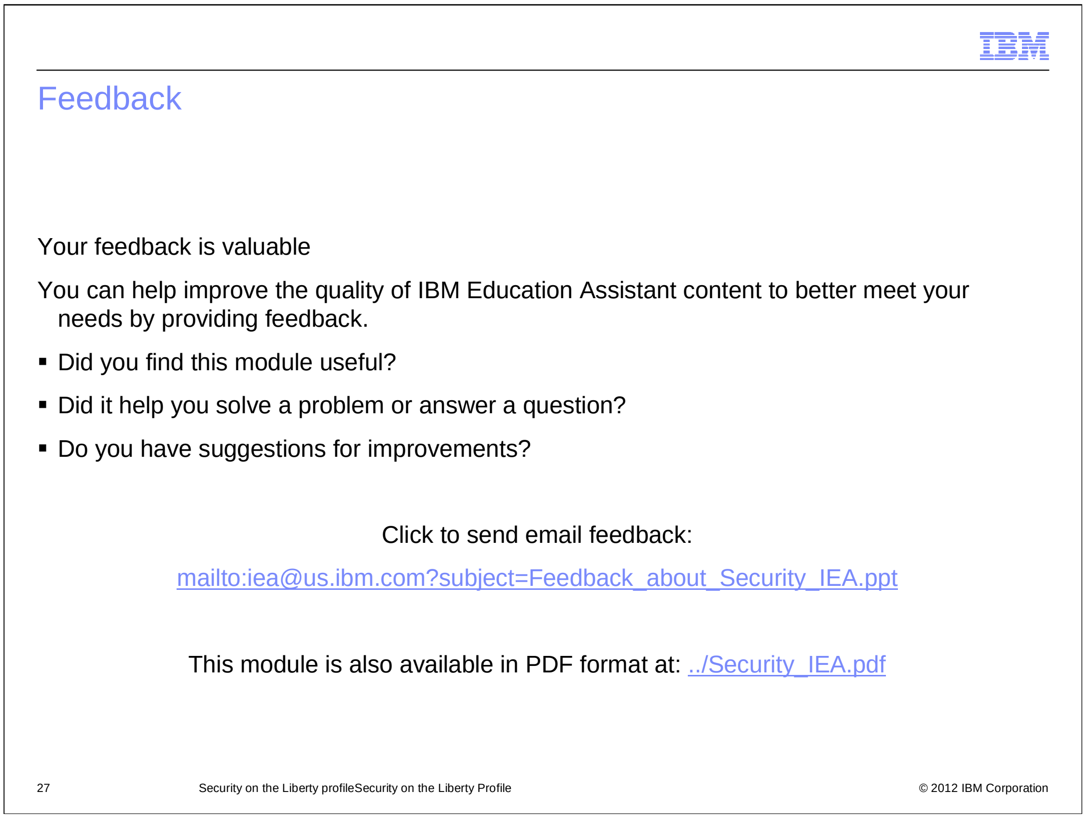
- Password encoding

http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.websphere.wlp.nd.multiplatform.doc/topics/rwlp_command_securityutil.html

- JMX security

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=twlp_admin_jmx

See these references for additional information about Security in the Liberty profile.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

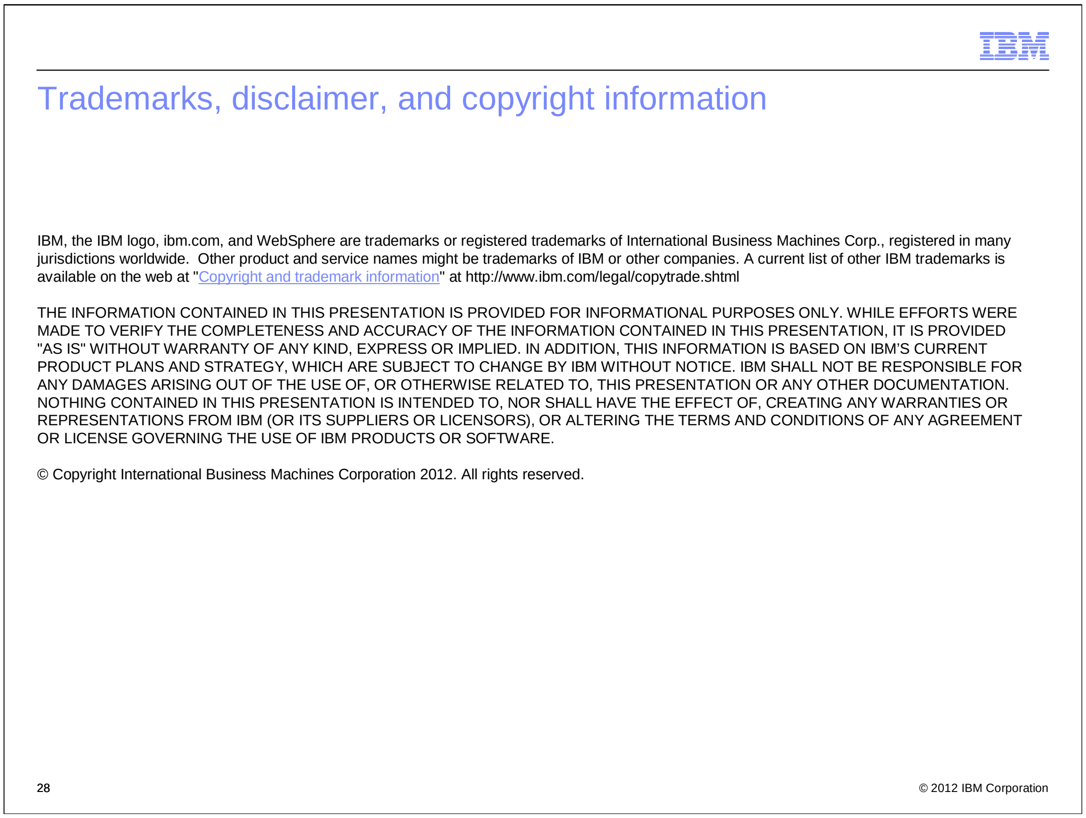
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Security_IEA.ppt

This module is also available in PDF format at: ../Security_IEA.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.