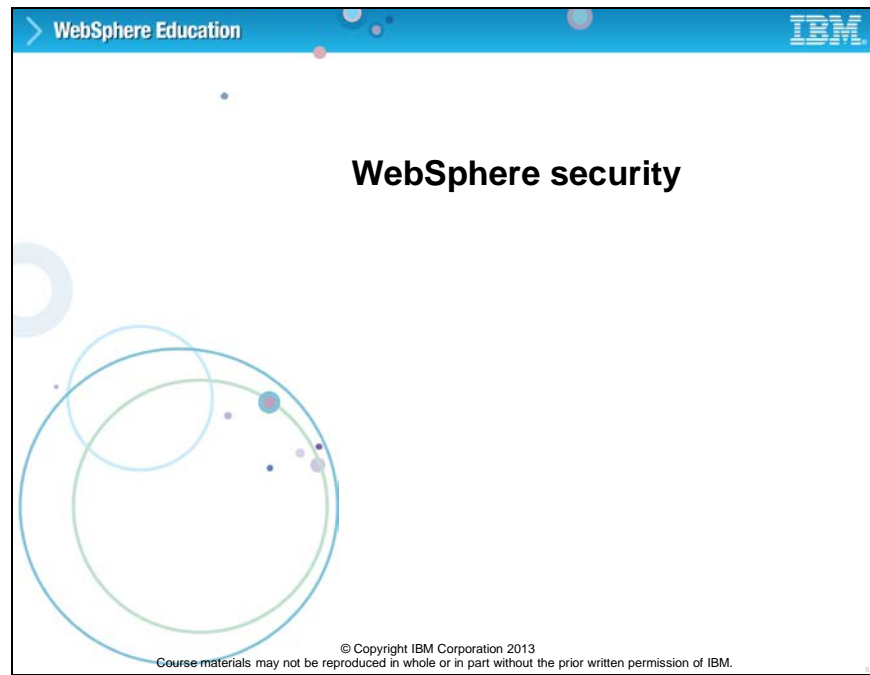



Slide 1



This unit describes the security services that run in WebSphere Application Server.

Slide 2



The slide is titled 'Unit objectives' and is part of a 'WebSphere Education' presentation. It lists ten objectives for the unit. The IBM logo is in the top right corner, and a copyright notice is at the bottom right.

WebSphere Education

Unit objectives

After completing this unit, you should be able to:

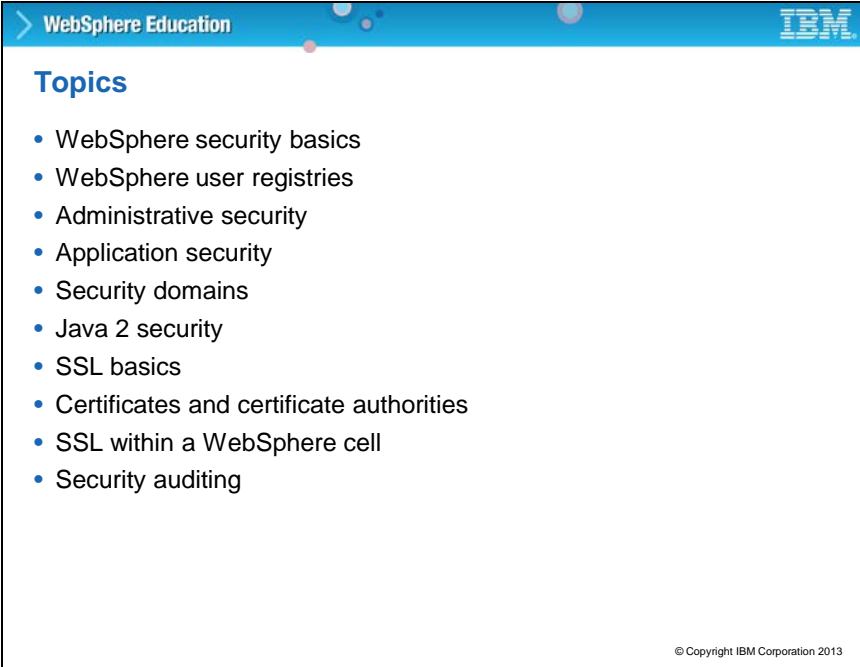
- Explain basic security concepts
- Describe the WebSphere Application Security architecture
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe SSL concepts and configuration
- Describe support for multiple security domains
- Describe auditing features and functions
- Describe support for Java Platform, Enterprise Edition 6 (Java EE 6) security annotations

© Copyright IBM Corporation 2013

After completing this unit, you should be able to:

- Explain basic security concepts
- Describe the WebSphere security architecture
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe SSL concepts and configuration
- Describe support for multiple security domains
- Describe auditing features and functions
- Describe support for Java Platform, Enterprise Edition 6 (Java EE6) security annotations

Slide 3



The slide is titled 'WebSphere Education' in the top left corner and features the IBM logo in the top right corner. The main heading is 'Topics' in blue. Below it is a bulleted list of ten topics related to WebSphere security. The slide has a blue header bar and a white main content area.

Topics

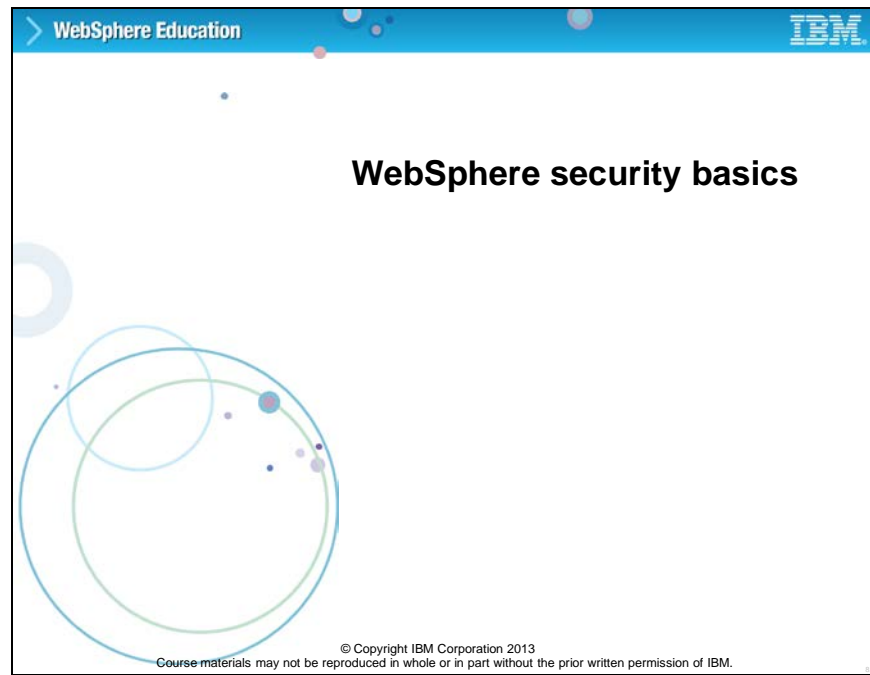
- WebSphere security basics
- WebSphere user registries
- Administrative security
- Application security
- Security domains
- Java 2 security
- SSL basics
- Certificates and certificate authorities
- SSL within a WebSphere cell
- Security auditing

© Copyright IBM Corporation 2013

This unit includes the following topics:

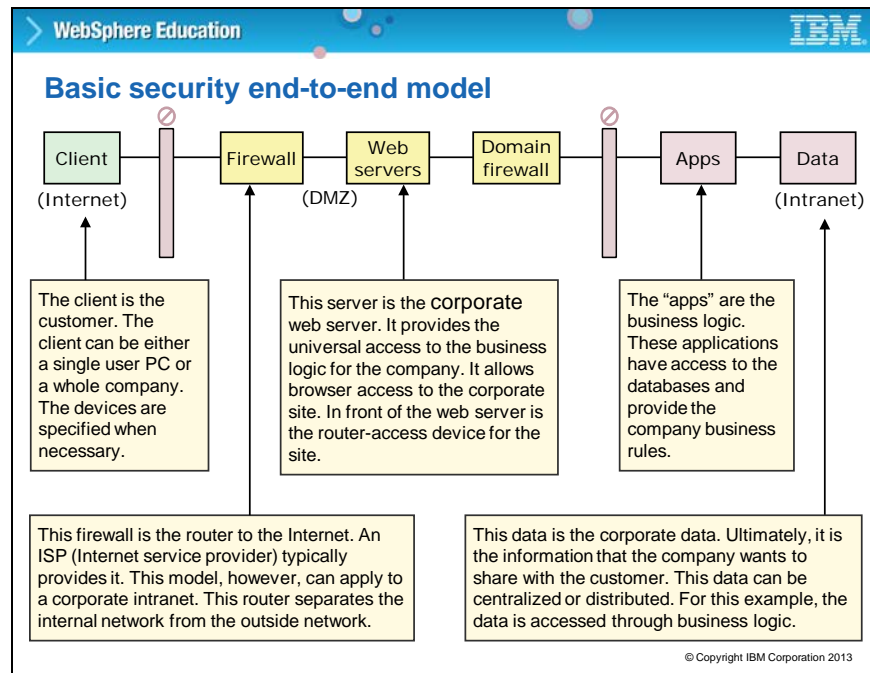
- WebSphere security basics
- WebSphere user registries
- Administrative security
- Application security
- Security domains
- Java 2 security
- SSL basics
- Certificates and certificate authorities
- SSL within a WebSphere cell
- Security auditing

Slide 4



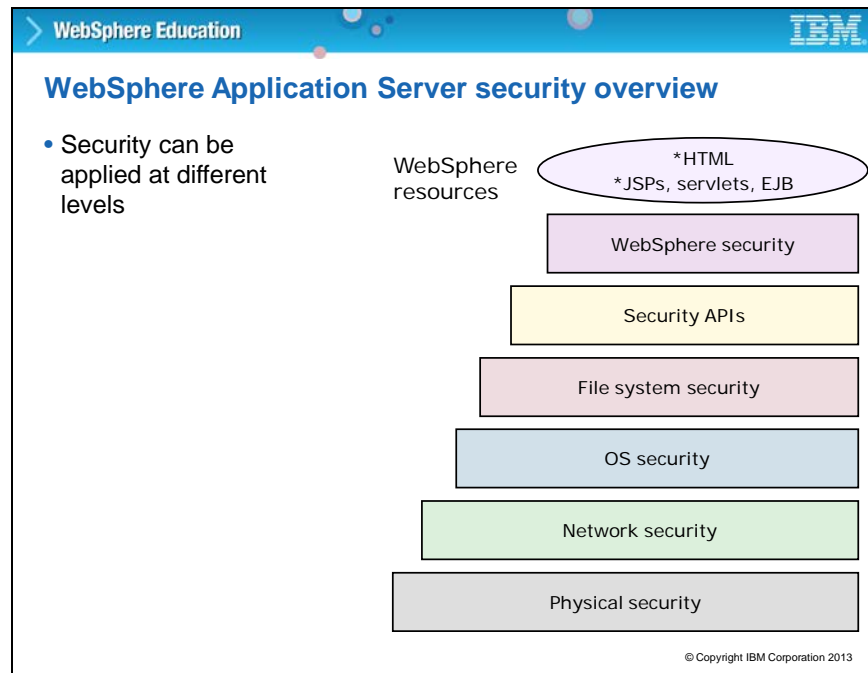
In this topic, you get an introduction to basic security concepts for WebSphere.

Slide 5



With an e-business application, a general topology must be secured from end- to-end. The end-to-end security model involves securing many different points. In the diagram, the client represents a single user or a whole organization. Requests from the client might go through a firewall or router on the Internet before they hit your web server in the DMZ. The web server allows browser access to your applications. There is another firewall between the DMZ and your intranet. The client request might then go to the application server layer, which has access to your data.

Slide 6



This diagram illustrates another way to look at security. There are many layers, starting with physical security, for example, building or server room access. Next, there is network security, operating system security, and file system security.

The security infrastructure of the underlying operating system provides certain security services to the WebSphere security application. The security services include the file system security support to secure sensitive files in WebSphere product installation.

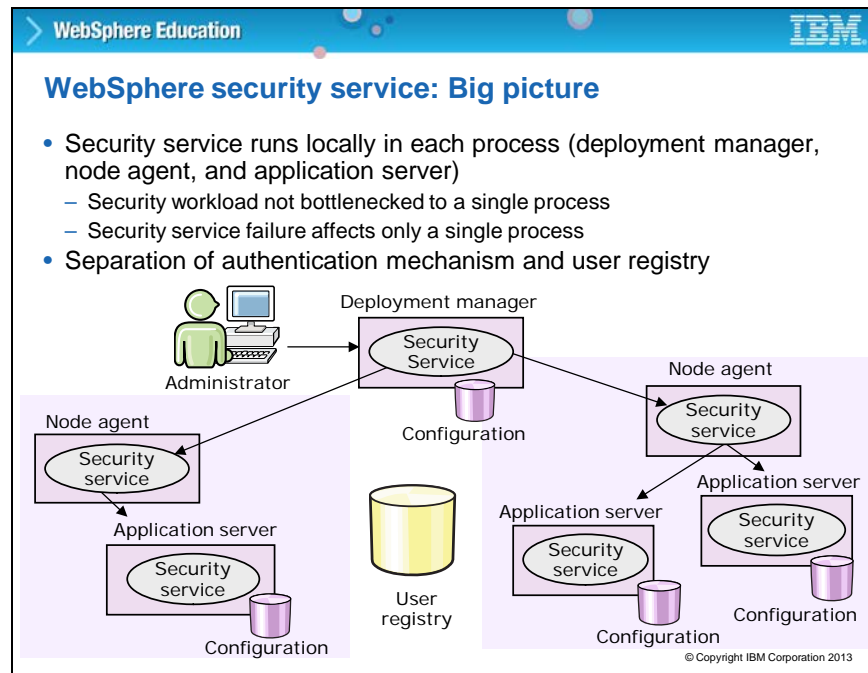
The WebSphere system administrator can configure the product to obtain authentication information directly from the operating system user registry. The JVM security model provides a layer of security above the operating system layer.

And then there is CORBA security. Any calls that are made among secure ORBs are started over a Secure Association Service (SAS) or CSCIv2 layer that sets up the security context and the necessary quality of protection. After the session is established, the call is passed up to the enterprise bean layer. This layer is for a distributed platform only. Next, is Java EE security. The security collaborator enforces Java based security policies and supports Java EE security APIs.

WebSphere security enforces security policies and services in a unified manner on access to web resources and enterprise beans. It consists of WebSphere security technologies and features to support the needs of a secure enterprise environment.

WebSphere security provides only part of the total security that must be applied. File system security still must be taken into account to protect your configuration files and key rings.

Slide 7




The security service runs within each of the managed processes so there is no single point of failure. The deployment manager, node agents, and application servers in a cell all have a security service, which means there is not a bottleneck for security workload.

Information about users and groups are stored in a user registry. A user registry authenticates a user and retrieves information about users and groups to handle security-related functions. One registry is configured for a cell.

Security can be administered from the administrative console or by using wsadmin from the deployment manager.

Slide 8

WebSphere Education 

Types of security

- Administrative security
 - Protects things such as administrative console, wsadmin, scripts
- Application security
 - Protects access to the applications
- Java 2 security
 - Protects the local systems



© Copyright IBM Corporation 2013

Within WebSphere, there are a number of different types of security that can be configured. The types of security are described in more detail during this unit. The types include administrative security, application security, and Java 2 security. The slide displays a screen capture from the administrative console. The slide shows where the security types are configured.

Administrative security is configured by default and application security is disabled. Before you can configure application security, you must verify that administrative security is configured. Application security is in effect only when administrative security is configured. Java 2 security is disabled by default.

Slide 9

WebSphere Education **IBM**

Administrative security

- Protects administrative console, scripts, wsadmin, and others
- Access can be restricted through:
 - Administrative roles
 - Fine-grained access

© Copyright IBM Corporation 2013

Administrative security allows the administrator to restrict access to the administrative interfaces, including the administrative console, administrative scripts, and wsadmin. You can define various administrative roles that give a particular user who is mapped to the role, a granular level of access to administrative resources in the environment. WebSphere Application Server is also more fine-grained, meaning that access can be granted to each user per resource.

Notice that if security is enabled, you are required to enter a user name and password when issuing certain functions and commands. The slide displays the various ways that you can provide the user name and password in the administrative tools.

Slide 10

WebSphere Education

IBM

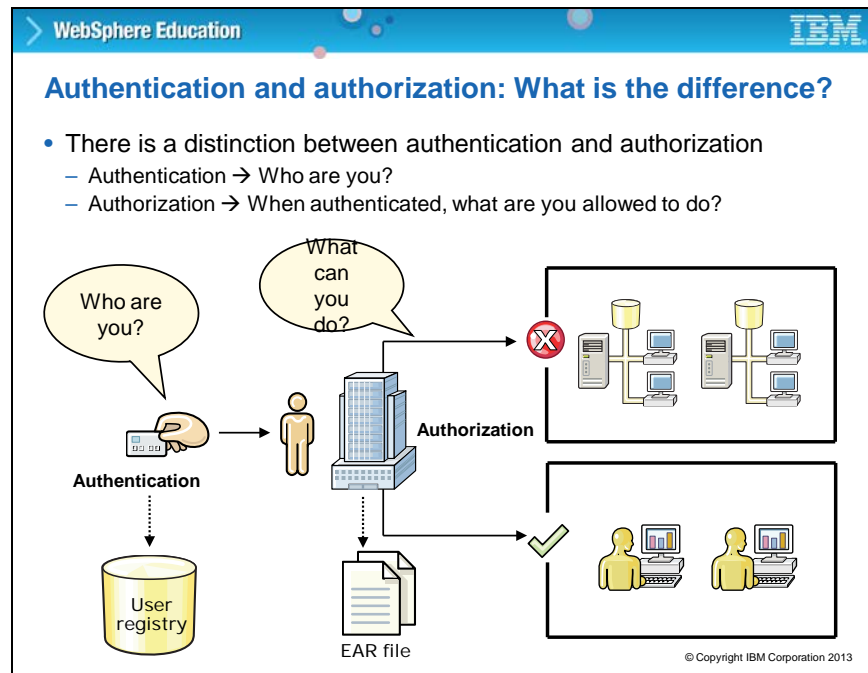
Application security

- Enables security for the applications in your environment
- Provides application isolation and requirements for authenticating application users
 - Security constraints protect servlets
 - Method permissions protect EJBs

The screenshot shows a web application titled 'PLANTS BY WEBSHERE'. It has a navigation bar with links: 'Flowers', 'Fruits & Vegetables', 'Trees', 'Accessories', 'HOME', 'SHOPPING CART', 'LOGIN', and 'HELP'. A message at the top right says 'Your shopping cart is currently empty'. An 'Authentication Required' dialog box is overlaid on the page. The dialog box contains a question mark icon and the text: 'A username and password are being requested by http://was8host01:9080. The site says: "Default"'. It has input fields for 'User Name:' (containing 'PlantsUser') and 'Password:' (masked with dots). There are 'OK' and 'Cancel' buttons at the bottom. Below the dialog box, there are sections for 'Tips' and 'Specials'. The 'Specials' section lists items: 'Bonsai Tree \$30.00 each', 'Red Delicious Strawberries \$3.50 (50 seeds)', and 'Tulips \$17.00 (10 bulbs)'. The copyright notice at the bottom right is '© Copyright IBM Corporation 2013'.

Application security allows the administrator to restrict access to the enterprise applications. Restricting access is done by defining security roles, security constraints, and method permissions, which protect the servlets, JSPs, and EJBs. Then, the security roles are mapped to the users and groups in the environment.

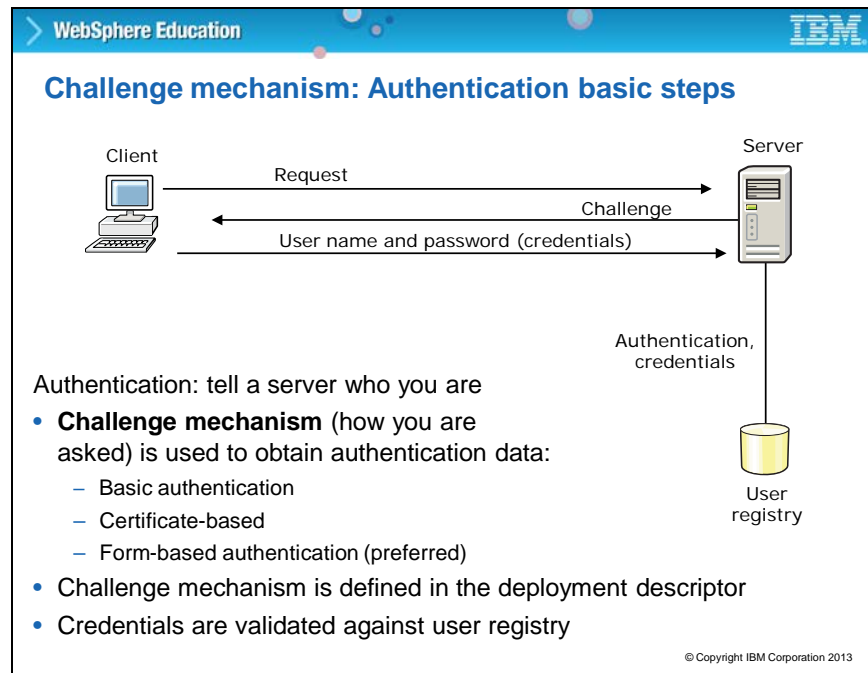
You configure application security for the PlantsByWebSphere application in the upcoming lab exercise.



Authentication information can be found in a user registry. Authorization information can be found within the EAR file. The WebSphere security service is responsible for making sure that protected resources are only accessible by authenticated and correctly authorized users.

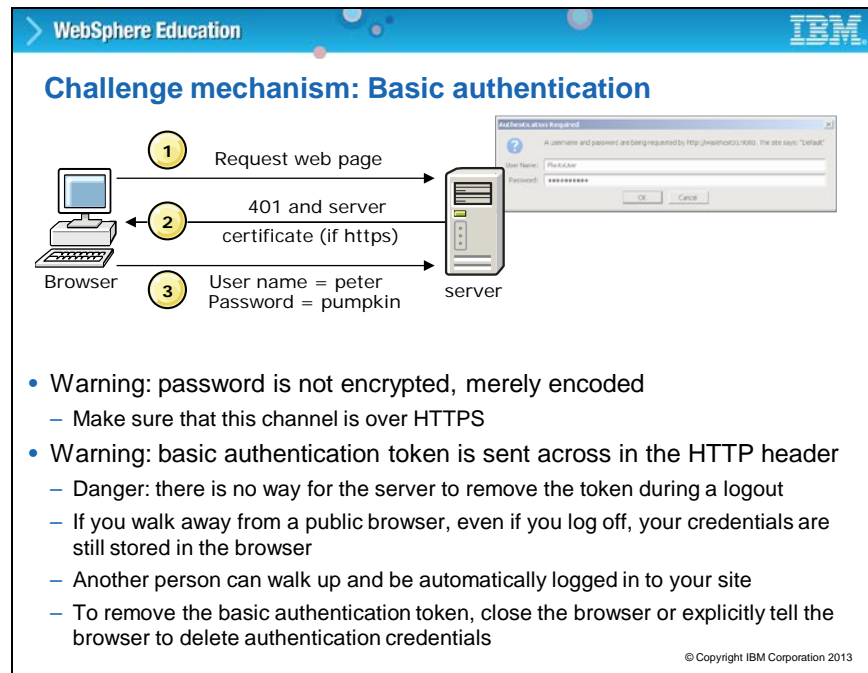
Authentication in its simplest form is rather straightforward. A request is made. A challenge is returned. A user ID and password are sent. The server checks a user registry to verify that the information is valid.

Authorization information determines whether a user or group has the necessary privileges to access resources.



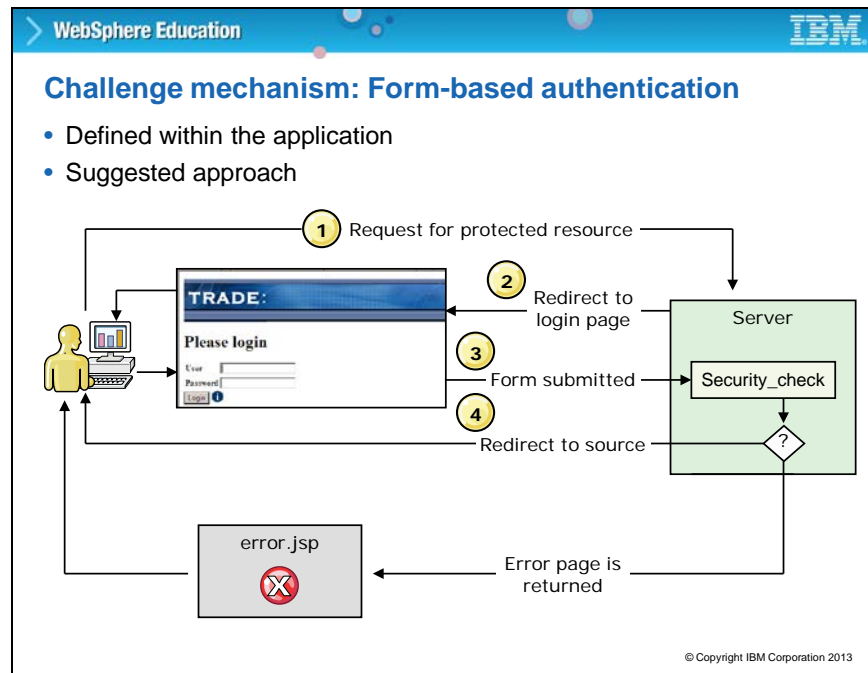
Authentication is rather straightforward. A request is made. A challenge is returned. A user ID and password are sent. The server checks a user registry to verify that the information is valid.

There are three options for the challenge mechanism that are used in an environment. The challenge mechanism that is selected is defined in the deployment descriptor for the application.



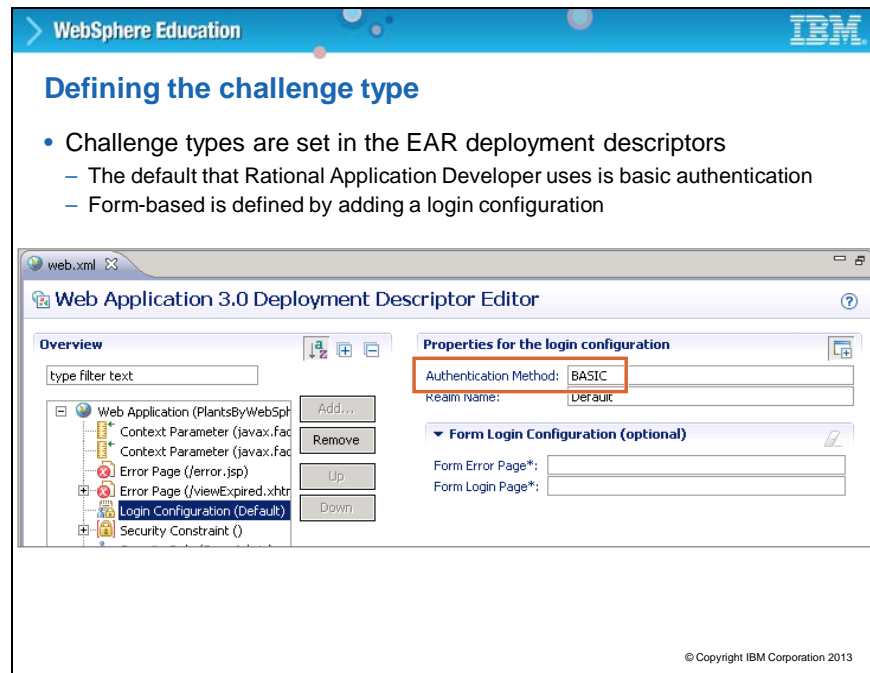
Basic authentication through the browser allows the server to initiate a user ID and password challenge by responding with a 401. The user ID and password are then encoded and sent to the server.

The basic authentication mechanism is not terribly secure since the encoding is not powerful. It is generally suggested that basic authentication is done after an HTTPS session is established. Authentication is stored at the server. The web browser validates the server certificate.



Form-based authentication provides a user with an editable HTML form to input and submit information to log in. When a user requests access to protected resources, they are redirected to a login page and presented with the form, where they can provide credential information. The user ID and password are entered and the form is submitted. If authentication is successful, the user is given access to the requested resource. If authentication fails, the user is redirected to an error page.

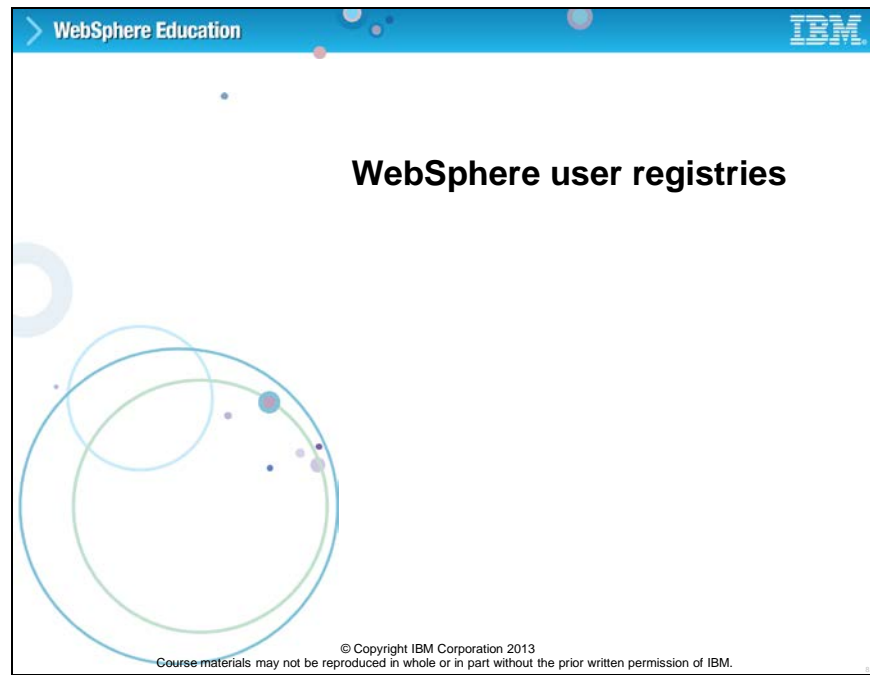
Slide 15



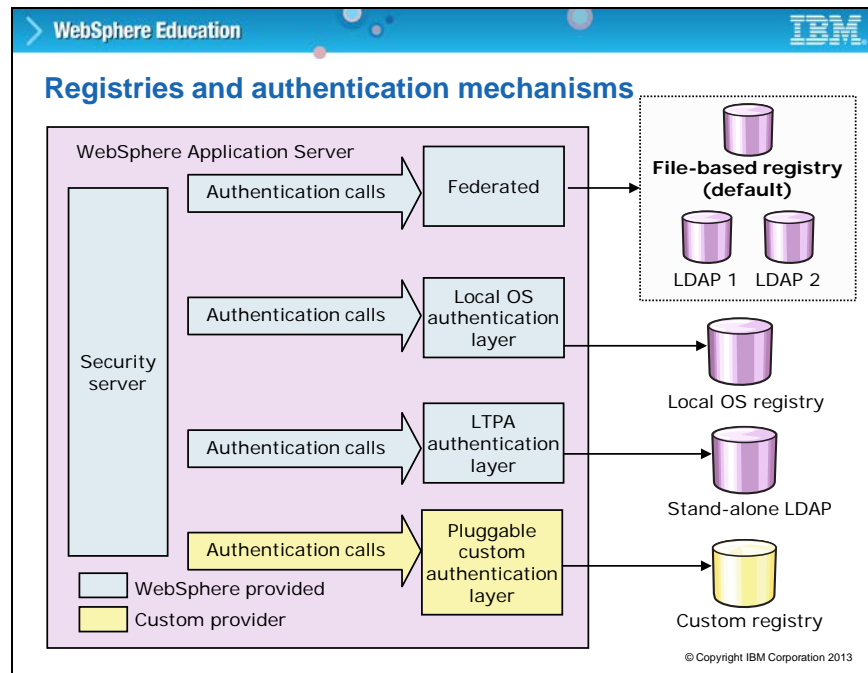
The challenge type is defined within the EAR file. By using Rational Application Developer assembly and deployment, the challenge type is defined within the deployment descriptor. The default type is basic authentication. You can add form-based authentication by adding a login configuration.

Challenge types are set in the deployment descriptors of the EAR file. Basic authentication is the default.

Slide 16



This topic examines the support for WebSphere user registries.

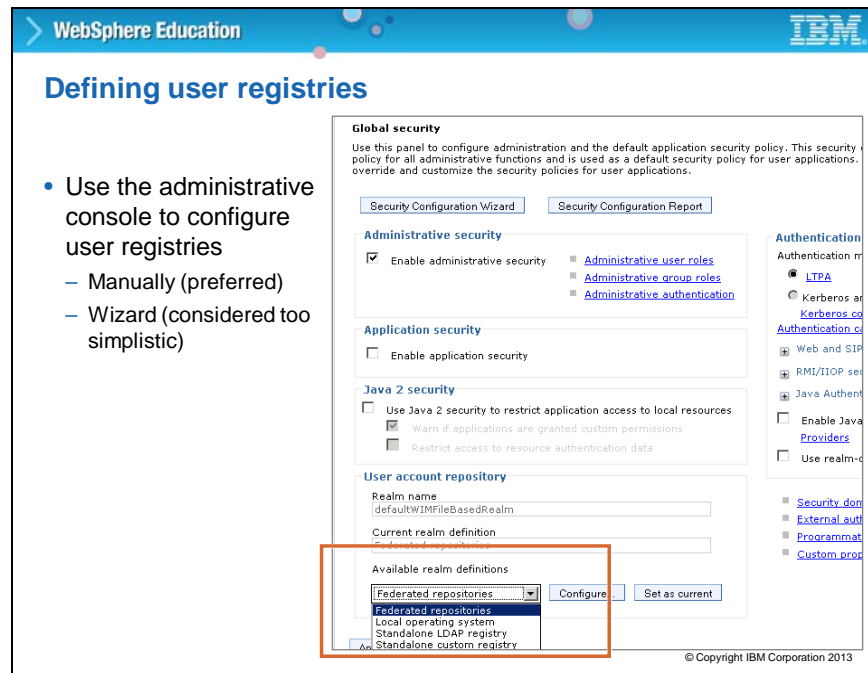


In WebSphere Application Server, a registry, or repository, is used for authentication and authorization. WebSphere provides implementations that support a number of user registries, which include local operating system, stand-alone LDAP, custom user registry, and federated registry.

A commonly used registry is the federated registry. The federated registry effectively combines multiple repositories into a single view. It can support multiple LDAP servers, file-based repository, database repository, and custom repositories.

WebSphere also provides a plug-in to support any registry by using the custom registry support. The custom registry allows you to configure any user registry that is not made available through the security configuration panels for WebSphere Application Server.

Slide 18



This screen capture on the slide shows where you can specify the user registry for your environment and configure other security settings by using the administrative console.

Configuring a registry can be done through the administrative console, wsadmin, or through a wizard. Generally speaking, the wizard is considered too simplistic to be used for configuring anything but the simplest configurations. The manual configuration is the preferred option for configuration. Manual configuration is done by using wsadmin or the administrative console Global security page.

Slide 19

Manual security configuration

User account repository

Realm name
defaultWIMFileBasedRealm

Current realm definition
Federated repositories

Available realm definitions

- Federated repositories (selected)
- Local operating system
- Standalone LDAP registry
- Standalone custom registry

Configure...

Global security > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

General Properties

★ Realm name
defaultWIMFileBasedRealm

★ Primary administrative user name
wasadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node
Password

☒ Ignore case for authorization

☐ Allow operations if some of the repositories are down

Repositories in the realm:

Add Base entry to Realm... Use built-in repository Remove

Select	Base Entry	Repository Identifier	Repository Type
<input type="checkbox"/>	o=defaultWIMFileBasedRealm	InternalFileRepository	File

You can administer the following resources:

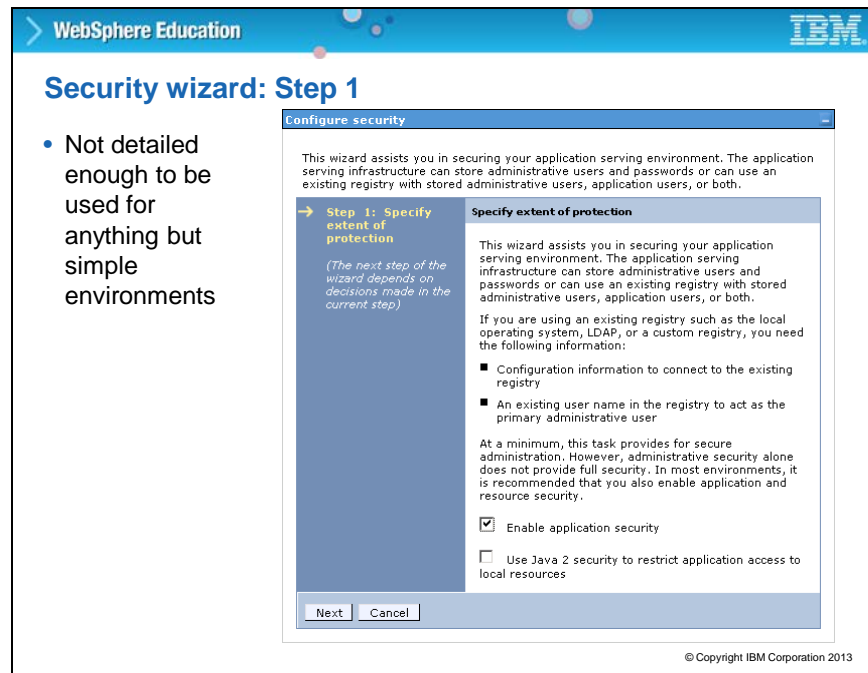
© Copyright IBM Corporation 2013

- The steps depend on the specific environment and which repository is being configured
- Much more detailed than the security wizard

The graphics on the slide show some of the screens that are used to manually configure federated repositories. The steps that you see depend on the specific environment and which repository is being used. The manual configuration provides more options and detail than going through the security wizard.

To access the configuration area in the administrative console, go to **Global security > Federated repositories**.

Slide 20



The wizard provides a way to interactively configure a registry and is only appropriate for basic configurations. The slide displays the first screen in security wizard. You can check the boxes to enable application security and Java 2 security here. Keep in mind, Java 2 security restricts application access to local resources.



The slide has a screen capture that shows step 2 of the security wizard. The type of user repository is selected on step 2.

The screenshot shows the 'Configure security' wizard window. The title bar says 'WebSphere Education' and 'IBM'. The main title is 'Security wizard: Step 3'. The window is titled 'Configure security'. It has a sidebar on the left with four steps: 'Step 1: Specify extent of protection', 'Step 2: Select user repository', 'Step 3: Configure federated repository' (which is highlighted with a yellow arrow), and 'Step 4: Summary'. The main content area is titled 'Configure federated repository'. It contains a paragraph explaining that a secure, file-based user repository is built into the system for storing administrative users or environments with a small number of users. It also mentions that the repository can be federated with one or more external LDAP repositories. A note states: 'Note: Use this panel to configure a federated repository with a built-in, file-based repository in the realm. To configure a federated repository with a non file-based repository in the realm, you must use the User accounts repository section on the Global security panel.' Below the note are three input fields: 'Primary administrative user name' with the value 'wasadmin', 'Password' with masked characters, and 'Confirm password' with masked characters. At the bottom are three buttons: 'Previous', 'Next', and 'Cancel'. A copyright notice '© Copyright IBM Corporation 2013' is at the bottom right.

WebSphere Education IBM

Security wizard: Step 3

Configure security

Secure the application serving environment

Step 1: Specify extent of protection
Step 2: Select user repository
→ **Step 3: Configure federated repository**
Step 4: Summary

Configure federated repository

A secure, file-based user repository is built into the system for storing administrative users or environments with a small number of users. The file-based user repository can be federated with one or more external LDAP repositories. If this is the first time security has been enabled using this repository, provide a new user name and password to act as an administrator. If security was previously enabled using this repository, provide the name of a user with administrator privileges that is in the built-in repository.

Note: Use this panel to configure a federated repository with a built-in, file-based repository in the realm. To configure a federated repository with a non file-based repository in the realm, you must use the User accounts repository section on the Global security panel.

Primary administrative user name
wasadmin

Password

Confirm password

Previous Next Cancel


© Copyright IBM Corporation 2013

In this case, in step 3 of the wizard, you specify the administrative user name and password.



After all the steps are completed, the summary screen displays. When you select **Finish**, your selected user registry is configured.

Slide 24

WebSphere Education 

User registry support

- WebSphere Application Server supports some user registries

Local OS	LDAP
NT Domain, NT WorkGroup, Windows	IBM Tivoli Directory Server
AIX	IBM SecureWay Directory Server
Solaris	Sun Java System Directory Server
HP-UX	IBM Lotus Domino
Linux	Microsoft Active Directory
OS/400	Novell eDirectory
	Custom (requires addition configuration)

© Copyright IBM Corporation 2013

The tables on the slide provide a list of the supported operating systems and LDAP servers. Besides, those listed, other LDAP servers can also be used by defining the appropriate schema mappings through the advanced LDAP properties.

Generally, local OS registries are avoided, particularly in distributed, non-domain environments.

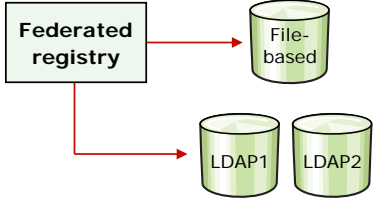
For complete details on supported user registries, visit the IBM WebSphere Application Server system requirements page.

WebSphere Education
IBM

Federated repositories

- The installation wizard and profile management tools have a default of enabling administrative security
 - The default repository type is a file-based federated repository
- Federated repositories provide for the use of multiple repositories with WebSphere Application Server
- Can be:

<ul style="list-style-type: none"> – File-based – Single LDAP – Custom registry 	<ul style="list-style-type: none"> – Database – Multiple LDAPs – Subtree of an LDAP
--	--
- Defined and theoretically combined under a single realm
- All of the user repositories that are configured under the federated repository are invisible to WebSphere Application Server



```

graph LR
    FR[Federated registry] --> FB[(File-based)]
    FR --> L1[(LDAP1)]
    FR --> L2[(LDAP2)]
  
```

© Copyright IBM Corporation 2013

Federated repositories enable you to use multiple repositories with WebSphere Application Server. The multiple different repositories are combined under a single realm.

You can configure various registries such as file-based, single LDAP, multiple LDAPs, subtrees of an LDAP, database, and a custom registry. When you use the federated repositories functions, all of the configured repositories that you specified during configuration, become active. To WebSphere, it appears as a single user repository.

A file-based federated user repository is configured as the active user registry by default when creating profiles.

WebSphere Education IBM

Custom registry: Configuration

[Global security](#) > [Standalone custom registry](#)

Specifies a custom registry that implements the UserRegistry interface in the com.ibm.websphere.security package. For backward compatibility, the application server also supports a custom registry that implements the CustomRegistry interface in the com.ibm.websphere.security package. When security is enabled and any of the properties on this panel are changed, go to the Security > Global security panel. Click Apply or OK to validate the changes.

General Properties

* Primary administrative user name

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

* Custom registry class name

☐ Ignore case for authorization

Custom properties

Select	Name	Value	New	Delete
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="New"/>	<input type="button" value="Delete"/>

Related Items

- Trusted authentication realms - inbound

Configured from administrative console:

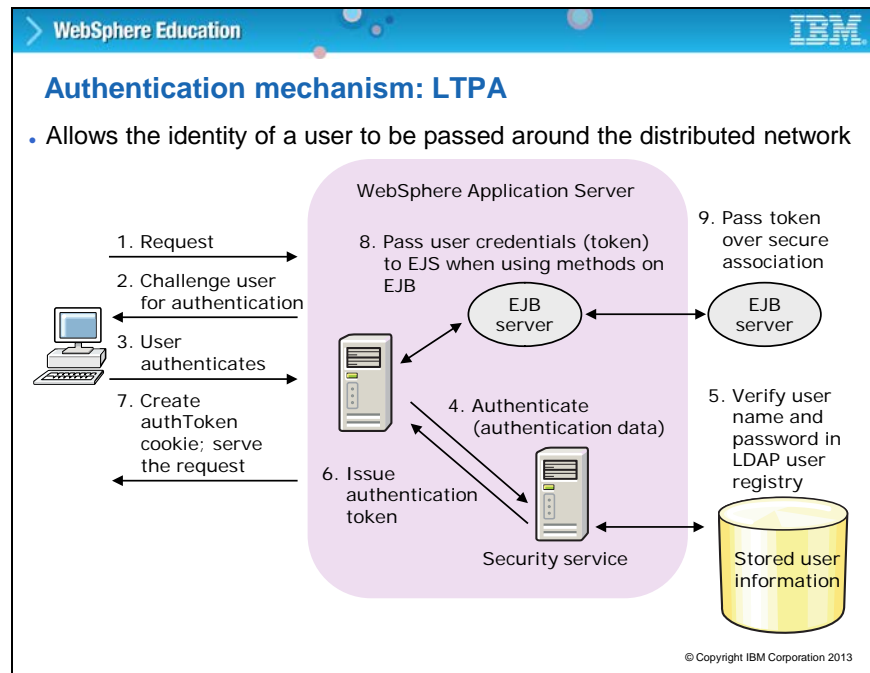
- **Security > Global security**
- Select **Stand-alone custom registry** from the **Available realm definitions**
- Click **Configure**
 - User name and password must exist
 - Class name must be implemented and in class path

© Copyright IBM Corporation 2013

The screen capture shows where you can configure a custom registry in the administrative console. Select **Global security > Standalone custom registry**. Specify the custom registry class name as shown. The custom registry implementation classes must be coded and available on the class path for the server. Also, the user and password must be defined and available to the implementing class because WebSphere Application Server checks to make sure that the defined user can be authenticated. If it cannot be authenticated, you are not able to enable security, which is a safety feature. The feature prevents you from enabling security and not being able to log back in the next time the server is restarted.

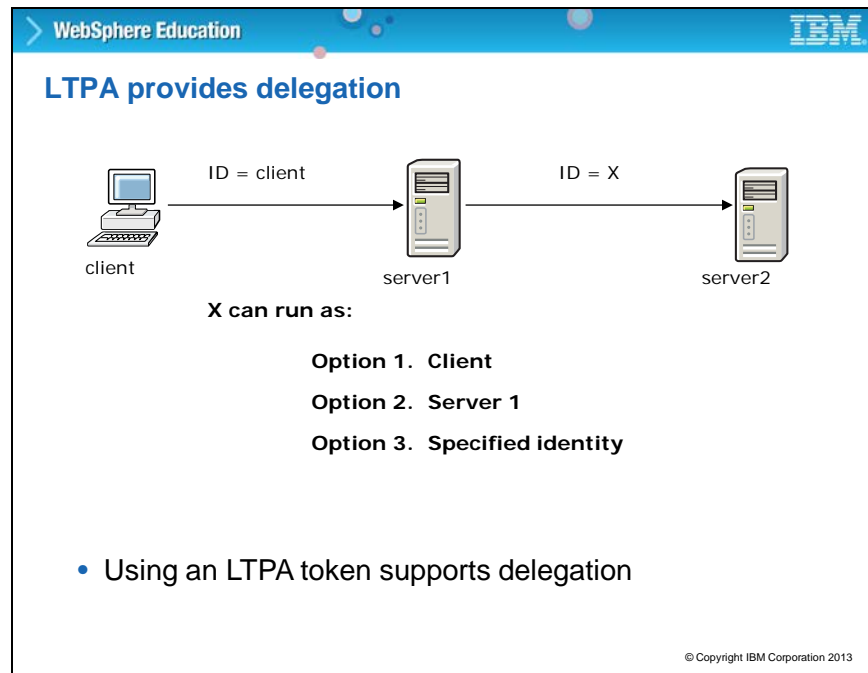
Custom registries can be used to handle a number of possibilities, including a database, flat file, OS with more custom logic, or other not directly supported registries.

There is a sample custom registry available in the information center that uses a flat file.



The authentication mechanism that is used in WebSphere Application Server is called Lightweight Third-Party Authentication, or LTPA. It allows a user ID to be passed around a distributed network. The diagram shows how LTPA is involved in authentication.

The user receives credentials in a token, which can be passed around the network. Regardless of what registry type you use, an LTPA token is generated.



LTPA supports delegation, meaning that a request can pass its security context that contains its credentials. As the call proceeds through the servers to its final destination, credentials can be changed. The options are to keep the credentials of the client, switch to the credentials of the server, or some other specified identity.

WebSphere Education **IBM**

LTPA provides single sign-on (SSO)

Global security > Single sign-on (SSO)
Specifies the configuration values for single sign-on.

General Properties

☒ Enabled

☐ Requires SSL

Domain name

☐ Interoperability mode

LTPA V1 cookie name

LTPA V2 cookie name

☒ Web inbound security attribute propagation

☒ Set security cookies to HTTPOnly to help prevent cross-site scripting attacks

- As soon as clients have a valid LTPA token, they do not need to reauthenticate within a cell (until the LTPA token expires)
- SSO is on by default
- Issues cookies to web browser to track user authentication information
- Provides for SSO within or even between WebSphere cells

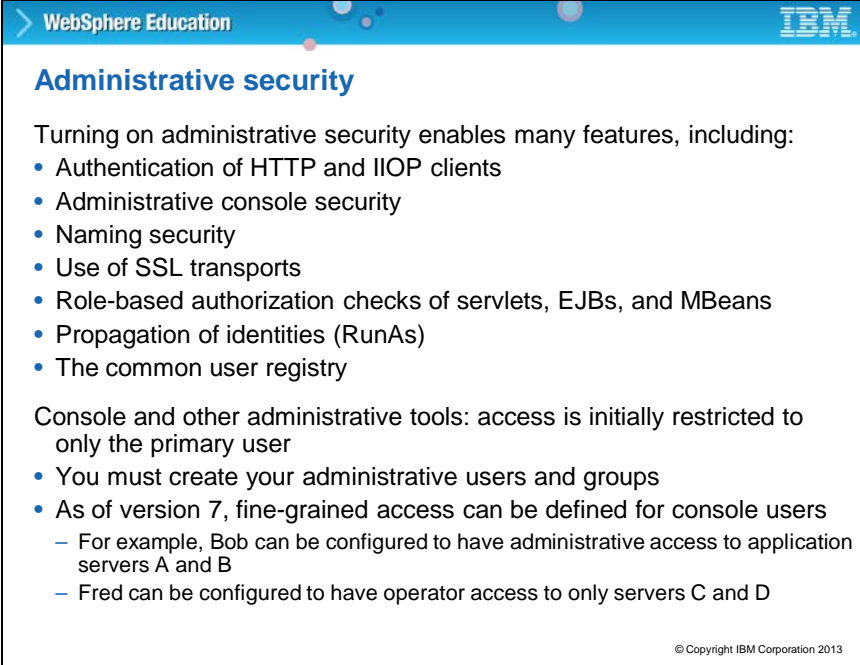
© Copyright IBM Corporation 2013


LTPA also supports single sign-on (SSO). The slide shows where you can enable SSO in the administrative console. A user authenticates only one time in a DNS domain and can access resources in other WebSphere Application Server cells without getting prompted again. SSO requires LTPA across the cells within the domain that is participating in SSO. The same realm names are on each system in the SSO domain. For local OS, on the Windows platform, the realm name is the domain name, if a domain is in use, or the computer name. On the UNIX platform, the realm name is the same as the host name. For LDAP, the realm name is the host:port of the LDAP server.

Slide 30



In this topic, concepts of administrative security are provided.



WebSphere Education 

Administrative security

Turning on administrative security enables many features, including:

- Authentication of HTTP and IIOP clients
- Administrative console security
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, EJBs, and MBeans
- Propagation of identities (RunAs)
- The common user registry

Console and other administrative tools: access is initially restricted to only the primary user

- You must create your administrative users and groups
- As of version 7, fine-grained access can be defined for console users
 - For example, Bob can be configured to have administrative access to application servers A and B
 - Fred can be configured to have operator access to only servers C and D

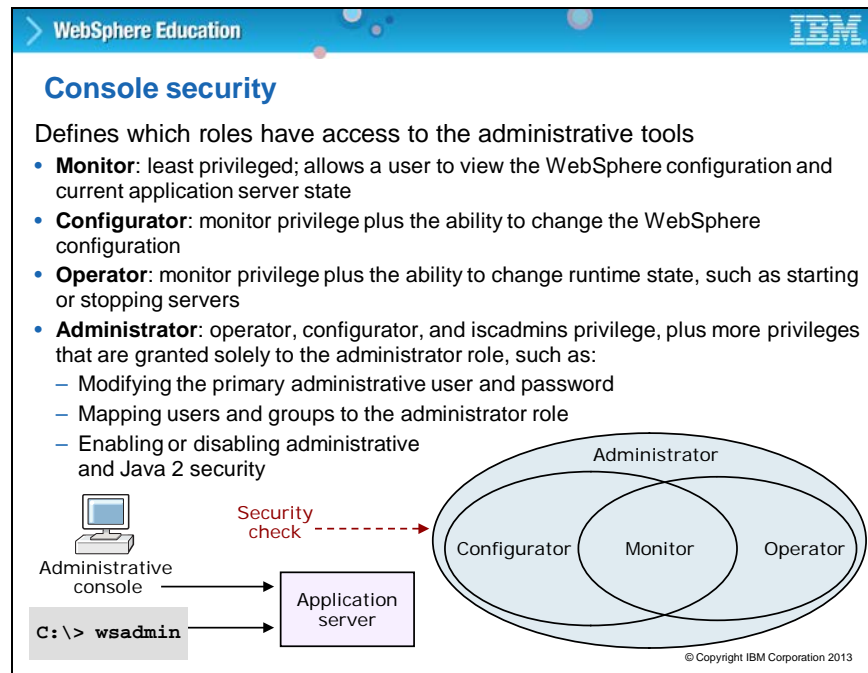
© Copyright IBM Corporation 2013

Administrative security protects the administrative tools, but it also configures a number of other security features. These features include:

- Authentication of HTTP and IIOP clients
- Naming security
- Use of SSL transports
- Role-based authorization checks of servlets, EJBs, and MBeans
- Propagation of identities (RunAs)
- The common user registry

Initially, administrative console access is restricted to the primary user, but you can configure other administrative users to access the console.

The first step in providing administrative access to users, is to create your administrative users and groups. You can also provide fine-grained administrative access to particular users.



After global security is enabled, there is a security check when the administrative console application is accessed. The security check makes sure that the accessing user is authenticated and the user is mapped to one of the console security roles, such as monitor, configurator, operator, or administrator. Depending on the console role to which the user is mapped, different functions are available. The user ID that is used to run the application server process has implicit access as a console administrator user.

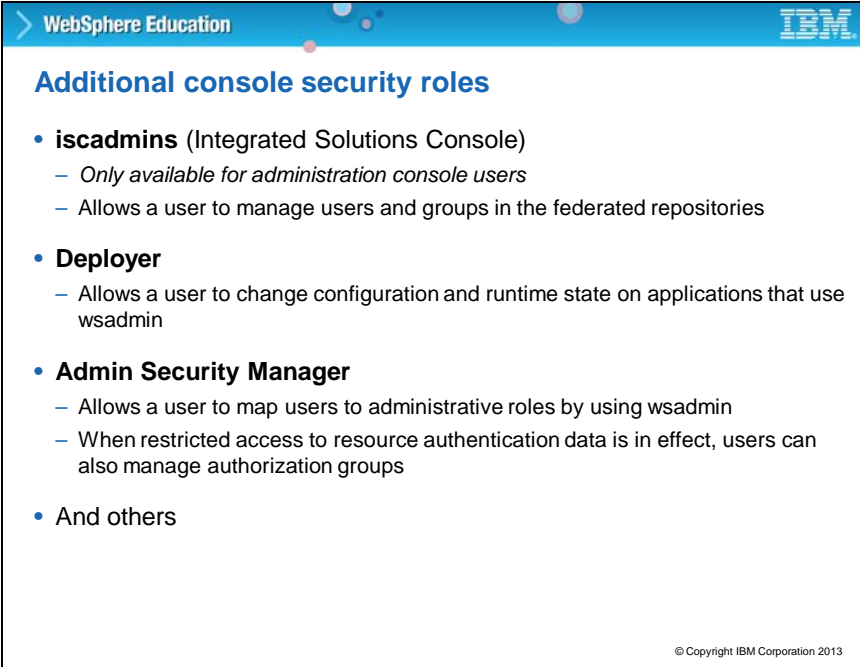
The monitor is the least privileged user and allows one to view the WebSphere configuration and current application server state.

The configurator has the monitor privileges plus the ability to make needed changes to the configuration.

The operator has the monitor privileges and can start and stop servers.

The administrator has all the privileges of the former roles, plus some additional privileges to modify the administrative user and password, map users, and groups to the administrator role, and enable or disable administrative security.

Slide 33



The slide is titled "Additional console security roles" and is part of a WebSphere Education presentation. It lists four bullet points: "iscadmins (Integrated Solutions Console)", "Deployer", "Admin Security Manager", and "And others". Each role has a brief description of its capabilities. The slide also includes the IBM logo and a copyright notice for 2013.

- **iscadmins** (Integrated Solutions Console)
 - Only available for administration console users
 - Allows a user to manage users and groups in the federated repositories
- **Deployer**
 - Allows a user to change configuration and runtime state on applications that use wsadmin
- **Admin Security Manager**
 - Allows a user to map users to administrative roles by using wsadmin
 - When restricted access to resource authentication data is in effect, users can also manage authorization groups
- And others

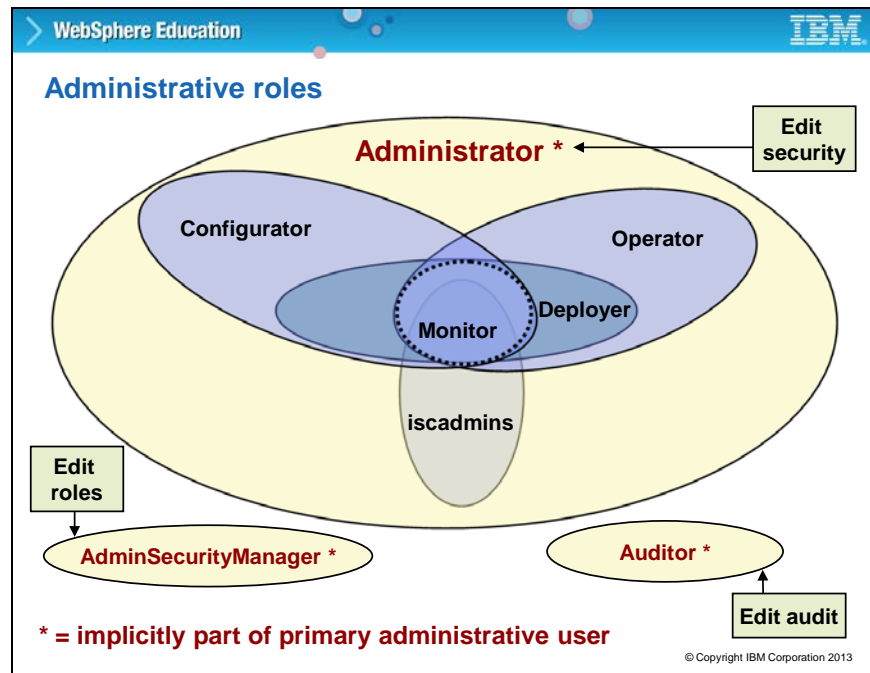
© Copyright IBM Corporation 2013

Some additional roles are listed on the slide. The iscadmins role is only available to administrative console users and not wsadmin users. The iscadmins role allows the user to manage users and groups in the federated repositories in the administrative console.

The Deployer role allows a user to change configuration and runtime state on applications.

The Admin Security Manager role separates administrative security administration from other application administration. The role allows a user to map other users to various roles.

Slide 34



The graphic on the slide displays a representation of the various administrative security roles that are available. The graphic shows how the roles overlap.

Slide 35

WebSphere Education IBM

Console security: Creating users and groups

- To set up console security
 - Turn on administrative security
 - Create console users and groups**
 - Done in active user registry

The screenshot displays the WebSphere console interface. On the left is a navigation tree with the following items: Environment, System administration, Users and Groups (expanded), Monitoring and Tuning, Troubleshooting, Service integration, and UDDI. Under 'Users and Groups', the sub-items are Administrative user roles, Administrative group roles, Manage Users (selected), and Manage Groups (selected). On the right, there are two panels. The top panel is 'Manage Users', which includes a search section with 'Search by' (User ID), 'Search for' (a text input field), and 'Maximum results' (100). Below the search section, it states '0 users matched the search criteria.' and has 'Create...' and 'Delete' buttons. The bottom panel is 'Manage Groups', which includes a search section with 'Search by' (Group name), 'Search for' (a text input field), and 'Maximum results' (100). Below the search section, it states '0 groups matched the search criteria.' and has 'Create...' and 'Delete' buttons. Arrows point from the 'Manage Users' and 'Manage Groups' sub-items in the navigation tree to their respective panels on the right. The footer of the console shows 'Page 1 of 1' and 'Total: 0' for both panels, and a copyright notice '© Copyright IBM Corporation 2013'.

To set up console security, first turn on administrative security, as shown previously, then create or use console users and groups. It is easier to manage users by assigning them to groups and then assign privileges to those groups, rather than assigning privileges to individual users. In the navigation tree, select **Users and groups > Manage users** (or **Manage groups**) to create new users or groups. They are created in the repository that you defined.

From these panels, you can also search for existing users or groups, and delete them.

WebSphere Education **IBM**

Console security: Mapping users and groups

- To set up console security
 - Turn on administrative security
 - Create console users and groups
 - Map users and groups to administrative roles**

Environment

- System administration
- Users and Groups**
 - Administrative user roles**
 - Administrative group roles
 - Manage Users
 - Manage Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Administrative user roles

Administrative user roles > User

Use this page to add, update or to remove administrative roles to users enables them to administer application through administrative console or through wsadmin scripting.

Role(s)

- Admin Security Manager
- Administrator
- Auditor
- Configurator

Search and Select Users

Decide how many results to display, enter a search string (e.g. *). Search. Select users from the Available list and add them to the Mapped to role list. Users which have already been mapped to a role will not be returned.

Search string:

Maximum results to display:

Available

-

Mapped to role

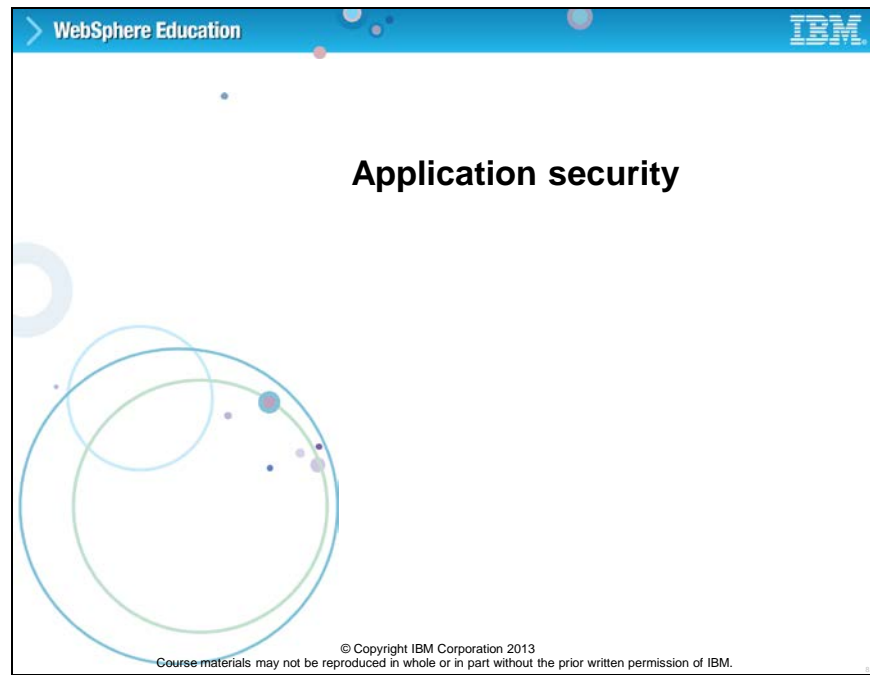
-

© Copyright IBM Corporation 2013

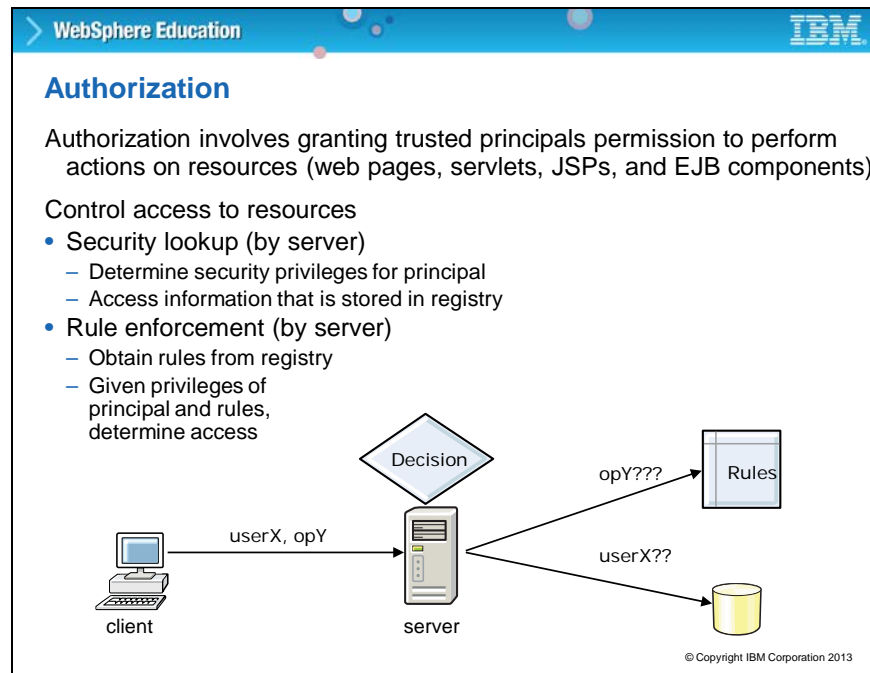
After users and groups are created, you can map them to the administrative roles. In the console, select **Administrative user roles (Administrative group roles) > Add** and then select the role from the list. On the panel for mapping groups, you have some special options that you can select for mapping a role to a group. All authenticated includes all authenticated users that are in the realm.

When you map all authenticated users to a specified role, all of the valid users in the current registry who authenticated can access resources that this role protects. Everyone maps everyone to the selected role. When you map everyone to a role, anyone can access the resources that this role protects and, essentially, there is no security. These special groups are probably not useful for mapping administrative users. However, they might come in handy for configuring application security later on. Find and select the appropriate users or groups and click the arrow to move them to the Mapped to role list.

Slide 37

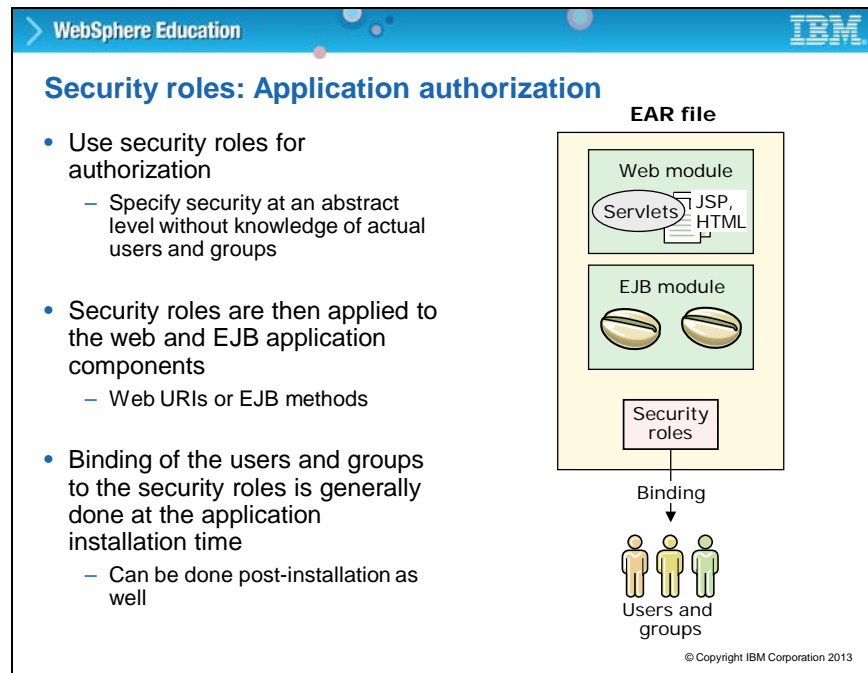


In this topic, concepts on application security are provided.

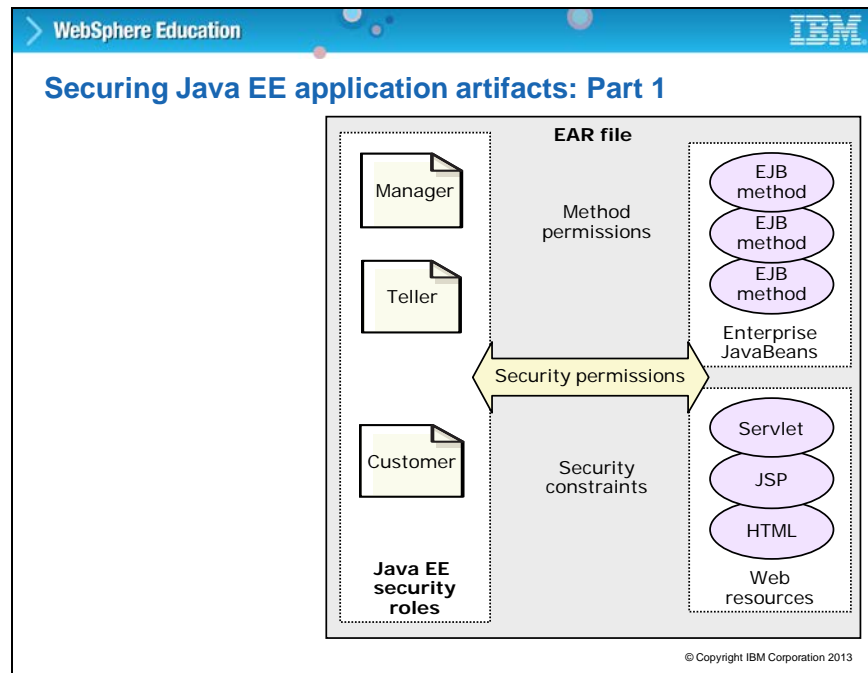


The topics that were covered so far in the unit are related to the concept of authentication. Now, the concept of authorization is covered. After a user is authenticated, you can control which resources that a user has access to in your domain through authorization. Authorization is provided through application security.

You can define access at a fine-grained level, such as which actions a user can do on a specific web page, servlet, or EJB. The rules that define access to resources are stored in the security registry. When a user requests access to a resource, the security server consults the registry and then grants permission as appropriate. Java EE security protects the rules engine through the WebSphere Application Server. Alternatively, a third-party authorization service, such as Tivoli Access Manager, can be used.



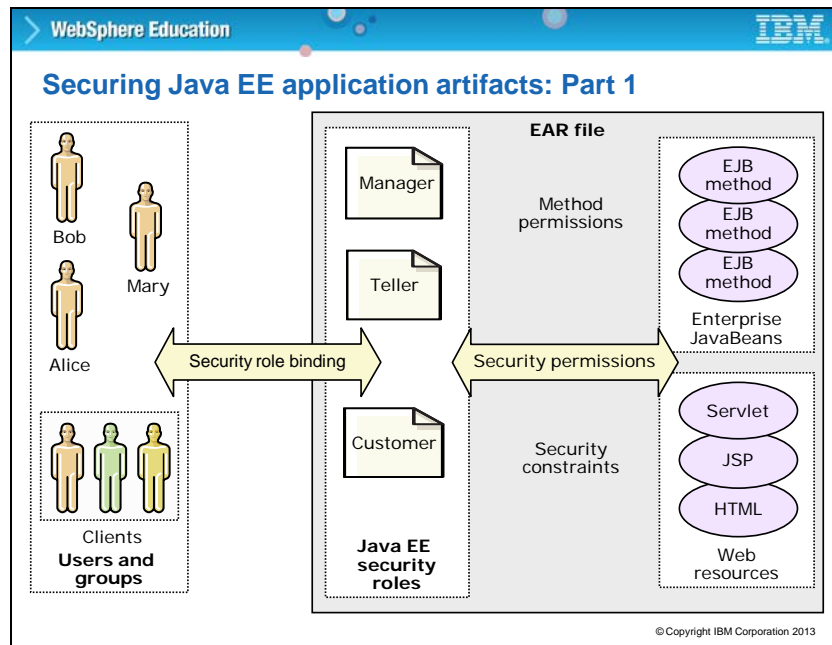
Roles are also used for authorization. When a developer creates an application, a security role is applied to the various application components, such as web and EJB components. These roles must be mapped to users and groups in your registry, which is called binding. Binding of the users and groups can be done at application installation time, or can be configured after the application is installed.



This diagram shows how security roles can be associated with specific components in an application. A developer can define access to certain actions that can be done on web components, such as GET and PUT, which are called security constraints.

For EJBs, you define method permissions to control access to specific methods on an EJB. The component provider or application assembler usually creates the security permission.

Slide 41



During deployment, the administrator can map the security roles to user and groups within the environment. The WebSphere runtime environment is then responsible for enforcing the restrictions. The diagram shows the association between permissions and security roles that are defined in the application EAR file, and actual users and groups in the domain.

A system administrator, or deployer, can map users and groups to the security roles during application installation or at some point after installation. In this example, some roles for a banking application are defined in the EAR file, for example Customer. At deployment time, you can assign individual users, such as Bob or Mary, or a group, such as Clients to the Customer role.

Slide 42

WebSphere Education
IBM

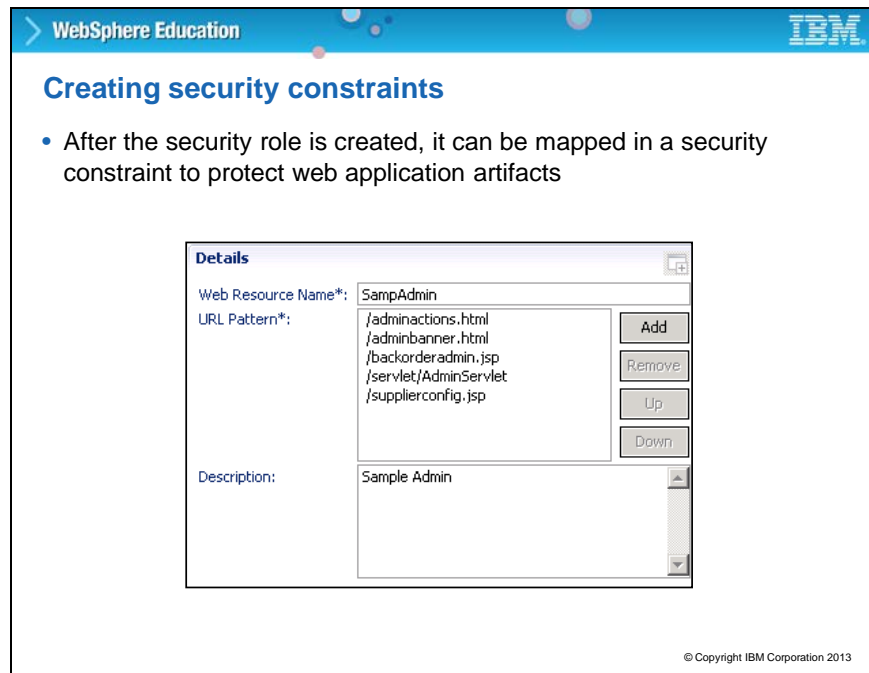
Applying application security

- Application security can be applied to resources within an EAR
 - Security roles are defined in the application deployment descriptor
 - Servlets and JSPs are protected with security constraints, which are mapped to the security roles
 - EJBs are protected with method permissions, which are mapped to the security roles
- The security roles are then mapped to actual users and groups during installation of the application

The screenshot shows the 'Web Application 3.0 Deployment Descriptor Editor' interface. The 'Overview' tab on the left displays a tree view of the application structure, including 'Context Parameter (javax.fac)', 'Error Page (/error.jsp)', 'Error Page (/viewExpired.xhtml)', 'Login Configuration (Default)', 'Security Constraint ()', and 'Security Role (SampAdmin)'. The 'Details' tab on the right shows the 'Authorization Constraint (optional)' section, where the 'Role Name' is set to 'SampAdmin'. The interface includes buttons for 'Add...', 'Remove', 'Up', and 'Down' to manage the constraints and roles.

The security roles can be defined in the EAR file by using a tool such as Rational Application Developer. Security roles can be defined in the deployment descriptor. The screen capture shows the SampAdmin security role within the PlantsByWebSphere application.

Slide 43



The screen capture show where you can define security constraints for web resources in Rational Application Developer. The constraints that are defined are for the PlantsByWebSphere application.

The details panel specifies a role named SampAdmin, and the URL patterns of the resources that SampleAdmin is allowed to access.

Slide 44

WebSphere Education
IBM

Using the console to map security roles

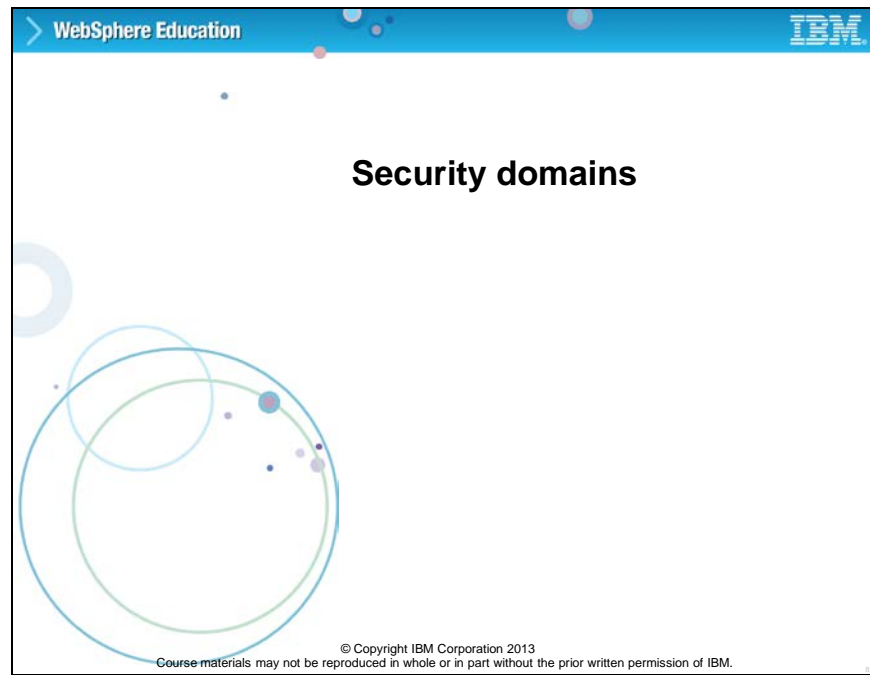
- The mapping of users and groups to security roles can take place during or after application installation
 - After installation, use the administrative console to go to the application and under Detailed Properties, select **Security role to user/group mapping**

The screenshot shows the 'Enterprise Applications' console window. The breadcrumb trail is 'Enterprise Applications > EnhancedPlantsByWebSphere > Security role to user/group mapping'. The page title is 'Security role to user/group mapping'. It contains a detailed explanation of role mapping and a table for mapping roles to users or groups. The table has columns for 'Select', 'Role', 'Special subjects', 'Mapped users', and 'Mapped groups'. One row is visible with 'SampAdmin' as the role and 'None' as special subjects. Buttons for 'Map Users...', 'Map Groups...', and 'Map Special Subjects' are at the top. 'OK' and 'Cancel' buttons are at the bottom.

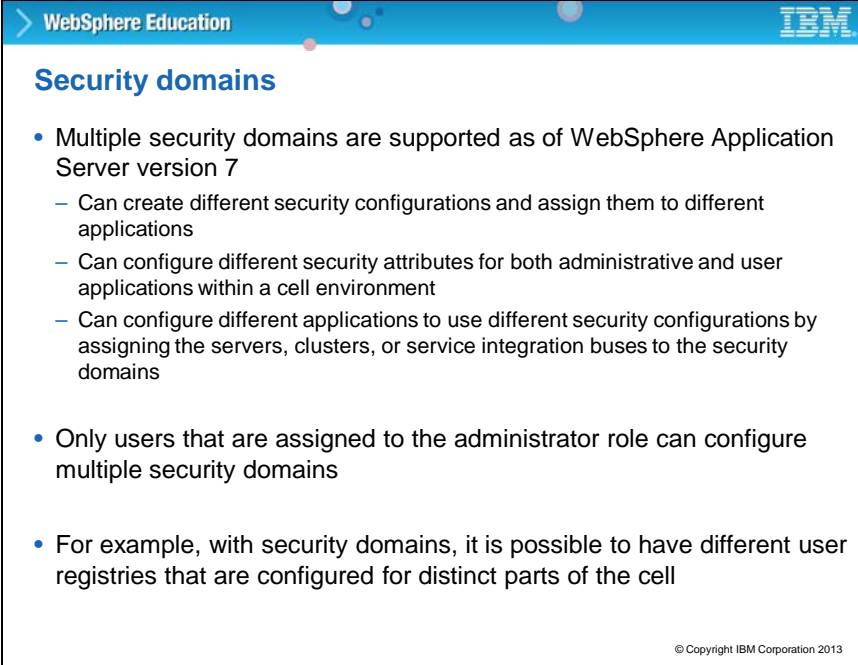
After application installation, the administrative console can be used to map the security roles to the actual user and groups that exist within that environment. Go to the detail page for the application, and under detailed properties, select **Security role to user/group mapping**.

You can then select a role, look up users or groups, and map the role to users or groups, or you can select one of the special options of all authenticated, or everyone.

Slide 45



In this topic, you explore the concept of security domains.



The slide is titled "Security domains" and is part of a "WebSphere Education" presentation, as indicated by the header. It features a blue header bar with the "WebSphere Education" text and the IBM logo. The main content is a bulleted list describing the capabilities of security domains in WebSphere Application Server version 7. The list includes: multiple security domains supported, the ability to create and assign different security configurations to applications, configuring security attributes for administrative and user applications, and assigning servers, clusters, or service integration buses to different security domains. It also notes that only users with the administrator role can configure multiple security domains and that different user registries can be configured for different parts of the cell. A small copyright notice "© Copyright IBM Corporation 2013" is located at the bottom right of the slide.

WebSphere Education

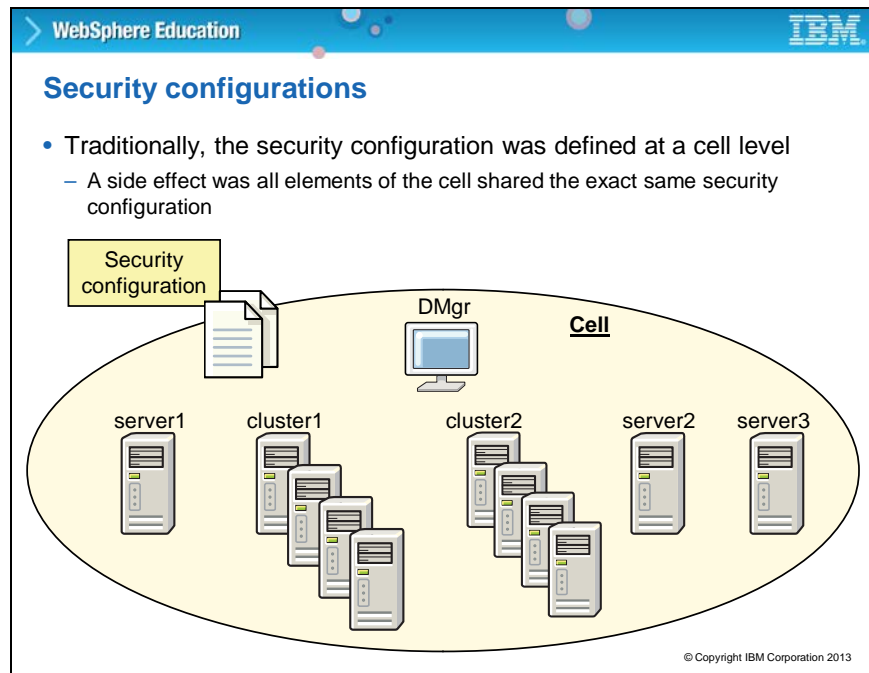
Security domains

- Multiple security domains are supported as of WebSphere Application Server version 7
 - Can create different security configurations and assign them to different applications
 - Can configure different security attributes for both administrative and user applications within a cell environment
 - Can configure different applications to use different security configurations by assigning the servers, clusters, or service integration buses to the security domains
- Only users that are assigned to the administrator role can configure multiple security domains
- For example, with security domains, it is possible to have different user registries that are configured for distinct parts of the cell

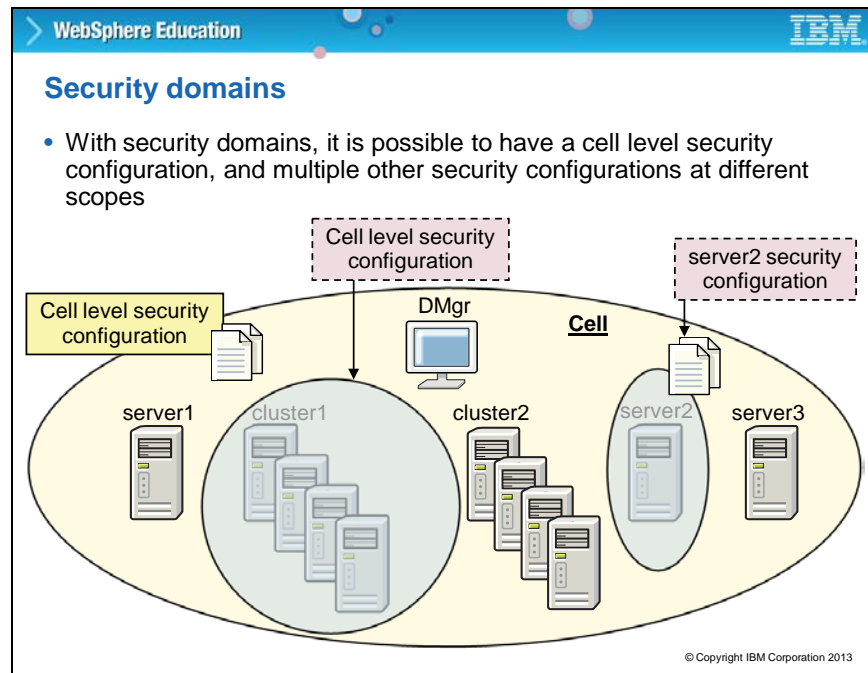
© Copyright IBM Corporation 2013

Security domains were a new feature in WebSphere Application Server V7. Security domains allow security to be defined at multiple levels, not just at the cell level. Previously, the security model for the cell was all that could be defined. Now, with security domains, it is possible to define one set of security settings for one application server and another security configuration for a second application server. You can configure different security settings for either administrative or user applications, and you can assign servers, clusters, or service integration buses to different security domains. You can also have different security registries that are configured for different domains. Only the system administrator role can configure security domains.

Slide 47

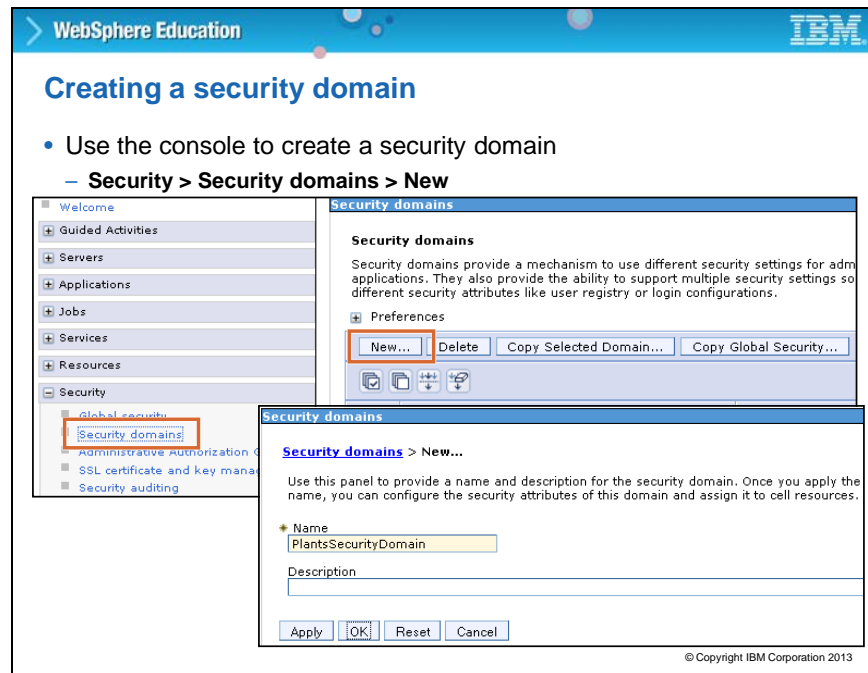


The diagram on the slide shows a security configuration that is defined at a cell level. A side effect of a cell level configuration is all of the elements in the cell share a security configuration.



The diagram on the slide shows not only the security configuration that is defined at the cell level, but also various security domains. Security domains provide the flexibility to use different security configurations within a cell. Each configuration is different from the cell level configuration and each is defined to a particular scope level.

Slide 49



You can create a security domain in the administrative console, by selecting **Security > Security domains > New**. Specify a name for the domain.

Slide 50

WebSphere Education

IBM

Configuring a security domain

- Define a scope and configure the attributes
 - It is possible to enable application security for only the PlantsByWebSphere

Assigned Scopes

Assign the security domain to the entire cell or select the specific servers, clusters, and service integration buses to include in this security domain.

Show: All scopes

- ☐ Cell
 - ☐ Clusters
 - ☐ Service integration buses
 - ☐ Nodes
 - ☐ was8host01CellManager01
 - ☐ was8host01Node01
 - ☐ Servers
 - ☒ server1
 - ☐ was8host01Node02

Security Attributes

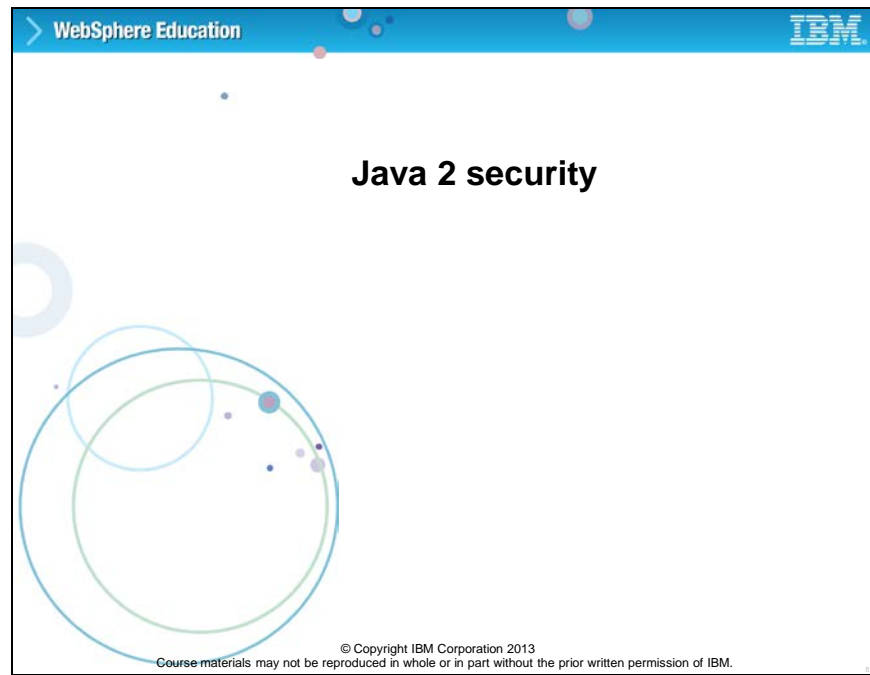
- Application Security:** Disabled
 - ☒ Use global security settings
Do not enable application security
 - ☐ Customize for this domain
 - ☐ Enable application security
- Java 2 Security:** Disabled
- User Realm:** Administrative realm
- Trust Association:** Disabled

© Copyright IBM Corporation 2013

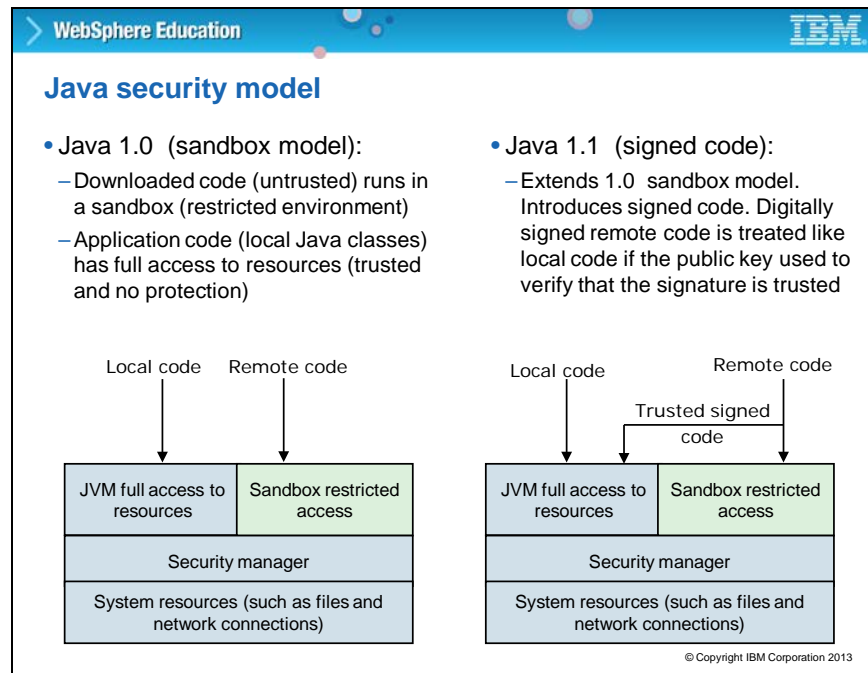
The next step is to define the scope of the security domain. For example, you might enable security for the cluster or a specific node. The Assigned scopes screen allows you to choose specific servers, clusters, or service integration buses.

You can also select the application security settings and Java 2 security settings on the second screen capture as shown on the slide.

Slide 51



In this topic, Java 2 security is introduced.

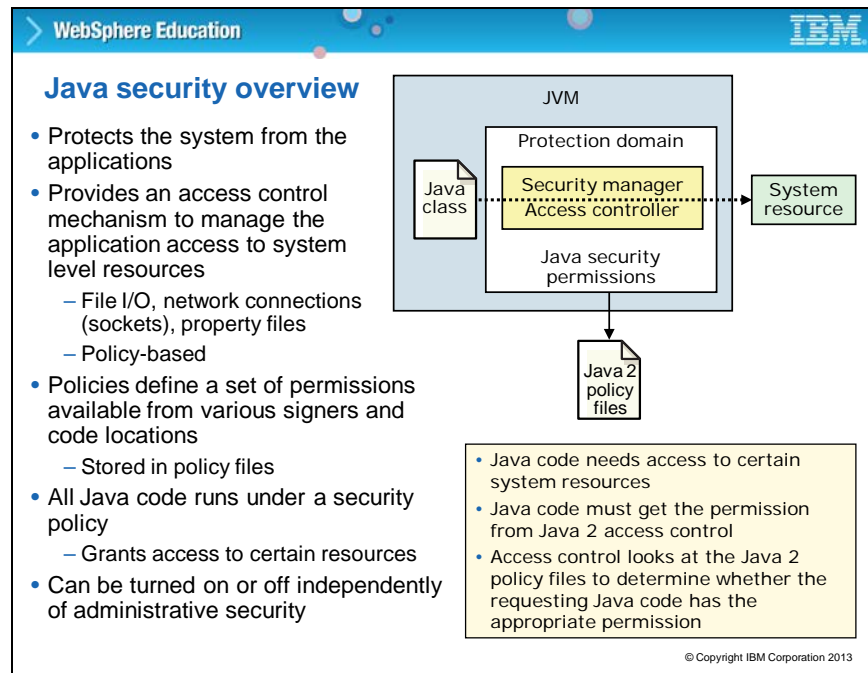


These diagrams illustrate the difference between the Java 1.0 and Java 1.1 security models. Java 1.0 is called a sandbox because components that are running within the sandbox cannot access anything on the local system.

Java security was designed to protect the local system from the code that is running on it. This method is different from application security, which protects the running code from clients that are attempting to use it. With Java 1.1, it was possible to sign code and therefore trust it.

Java 1.1 allowed the administrator to have code that was running access certain parts of the local system.

Slide 53



Java 2 security is different from application security, and is about protecting system resources. It is policy-based. Several policy files control it, and can provide fine-grained access control to system resources, such as file I/O, sockets, and properties.

Applications must be prepared to work with Java 2 security, or problems can occur. Applications that need access to certain system resources must get permission for Java 2 access control. Therefore, the Java 2 policy files must be defined appropriately.

You can turn on or turn off Java 2 security by using the administrative console. Java 2 can be turned on or off independent of administrative security.

Slide 54

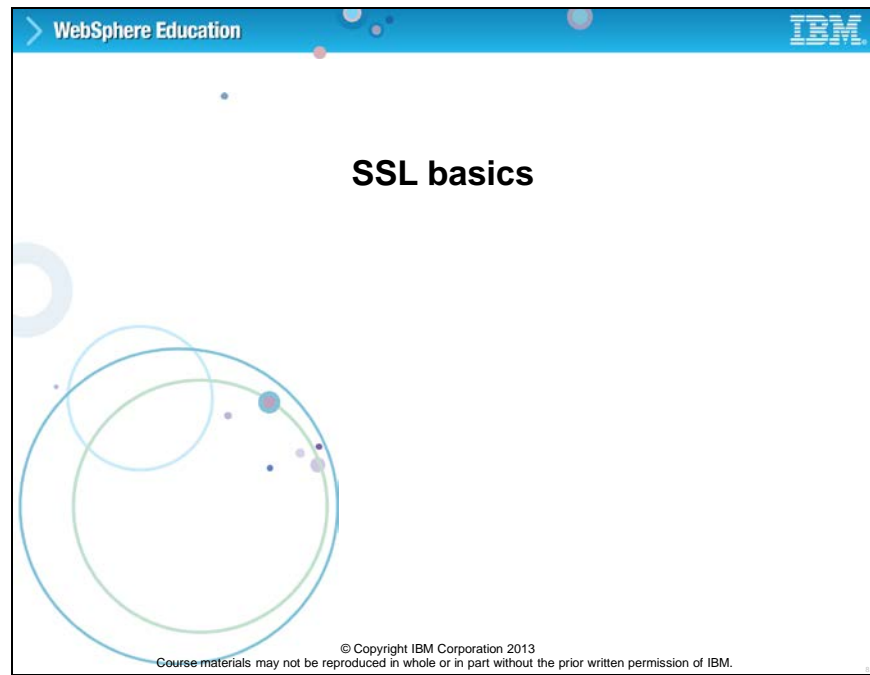


This screen in the console shows where Java 2 security can be configured. When Java 2 security is configured, the security manager component by default throws a `java.security.AccessControl` exception when a permission violation occurs. The exception, if not handled, often causes a runtime failure. The exception is also logged in the `SystemOut.log` file.

However, when the JVM `norethrow` property is set and has a value of `true`, the security manager does not throw the `AccessControl` exception, it just logs it in the log file. The property is intended for a sandbox or debug environment only, since it instructs the security manager not to throw the `AccessControl` exception.


By not rethrowing the exception, Java 2 security is not enforced. The property must not be used in a production environment where a relaxed Java 2 security environment weakens the integrity Java 2 security is intended to produce.

Slide 55



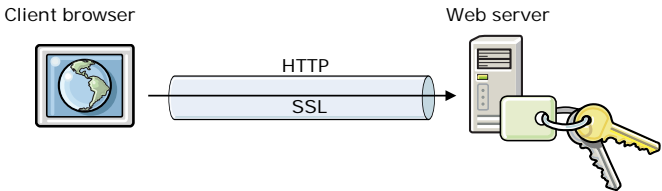
This topic introduces basic concepts of SSL.

Slide 56

WebSphere Education 

What is SSL?


- SSL stands for Secure Sockets Layer
- SSL provides connection security through:
 - Communication privacy: the data on the connection can be encrypted
 - Communication integrity: the protocol includes a built-in integrity check
 - Authentication: the client knows who the server is
- Creates a VPN
 - Uses both symmetric and asymmetric key encryption



The diagram illustrates a secure connection between a Client browser and a Web server. On the left, a 'Client browser' is represented by a computer icon with a globe. On the right, a 'Web server' is represented by a server rack icon. A horizontal cylinder connects them, with 'HTTP' written above it and 'SSL' written below it. To the right of the cylinder, a green padlock is shown with a yellow key inserted into it, symbolizing encryption and security.

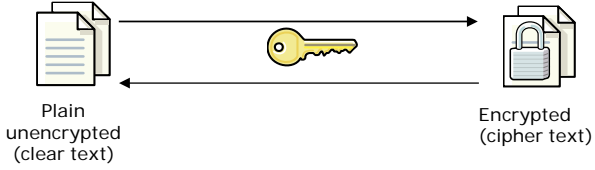
© Copyright IBM Corporation 2013

The following slides describe SSL and how it is used in WebSphere Application Server. SSL, or Secure Sockets Layer, provides transport level security between two points, much like a VPN. It is often used to secure communications between a browser and a web server.

WebSphere Education 

Symmetric key encryption


- Symmetric or secret key technology is a model in which two parties have a shared secret
- The same key is used for both encryption and decryption



The diagram illustrates the symmetric key encryption process. On the left, a document icon is labeled 'Plain unencrypted (clear text)'. An arrow points from this document to a document icon on the right labeled 'Encrypted (cipher text)'. A yellow key icon is positioned above the arrow, indicating that the same key is used for both encryption and decryption.

© Copyright IBM Corporation 2013

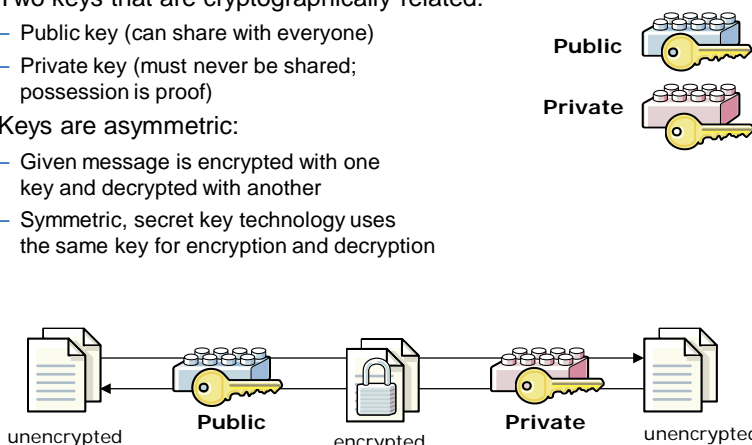
SSL uses two different kinds of encryption, asymmetric and symmetric. Symmetric key encryption allows two parties to send each other secured messages, but requires a shared secret, or key. The trouble with this approach is that at some point, that secret must be shared. In an e-business application, a shared secret is rather difficult.

WebSphere Education 

Asymmetric key encryption

Public key cryptography

- Two keys that are cryptographically related:
 - Public key (can share with everyone)
 - Private key (must never be shared; possession is proof)
- Keys are asymmetric:
 - Given message is encrypted with one key and decrypted with another
 - Symmetric, secret key technology uses the same key for encryption and decryption

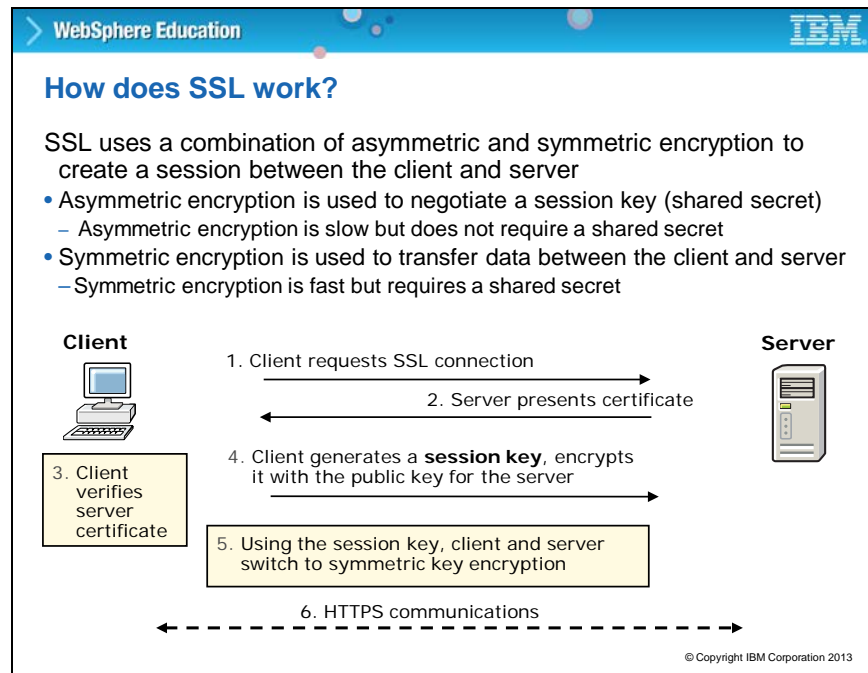


The diagram illustrates the asymmetric key encryption process. It shows a sequence of steps: an 'unencrypted' message (represented by a document icon) is processed using a 'Public' key (represented by a blue key icon) to become an 'encrypted' message (represented by a document icon with a lock). This encrypted message is then processed using a 'Private' key (represented by a red key icon) to become an 'unencrypted' message (represented by a document icon). Above the keys, the text 'Public' and 'Private' are written next to their respective key icons. The diagram also includes a copyright notice: '© Copyright IBM Corporation 2013'.

With asymmetric key encryption, there are two keys. A public key that can be shared, and a private key that is never shared. The keys are asymmetric, meaning that a message can be encrypted with one, but must be decrypted with the other.

If a server has a public-private key set, it can send out its public key through a signing certificate, also known as a certificate, to clients. Those clients can then use that public key to encrypt messages that are destined for the server, which then only the server can decrypt. Unlike symmetric key encryption, this process does not require the client and server to have a shared secret. Since the client can validate the certificate of the server, there is one-way authentication.

Currently, the server has no way to authenticate the client, nor can the server send the client secured messages.



SSL combines both methods of encryption to use the advantages and benefits of each. Symmetric encryption is faster, but requires a shared secret. Asymmetric is slow, but does not require a shared secret.

SSL uses asymmetric encryption to negotiate a session key and establish a session between a client and a server. At that point, symmetric encryption is used to transfer data between the client and server because it is faster. Because the client chooses its own session key, nobody else knows it. It can securely send that session key to the server by using the public key of the server. Now, nobody but the client and server know the session key. The session key is then used as a “shared secret” to switch to the much more efficient symmetric key encryption. This process is typically called the SSL handshake.

Slide 60



This topic introduces basic concepts of certificates and certificate authorities.

WebSphere Education

IBM


What is a certificate?

Simple answer:

- It is an electronic document that identifies you, and a third-party vouches for both you and the certificate itself
- Examples:
 - Employee badge (vouched for by your employer)
 - Drivers license (vouched for by your state)
 - Passport (vouched for by your country)

More information:

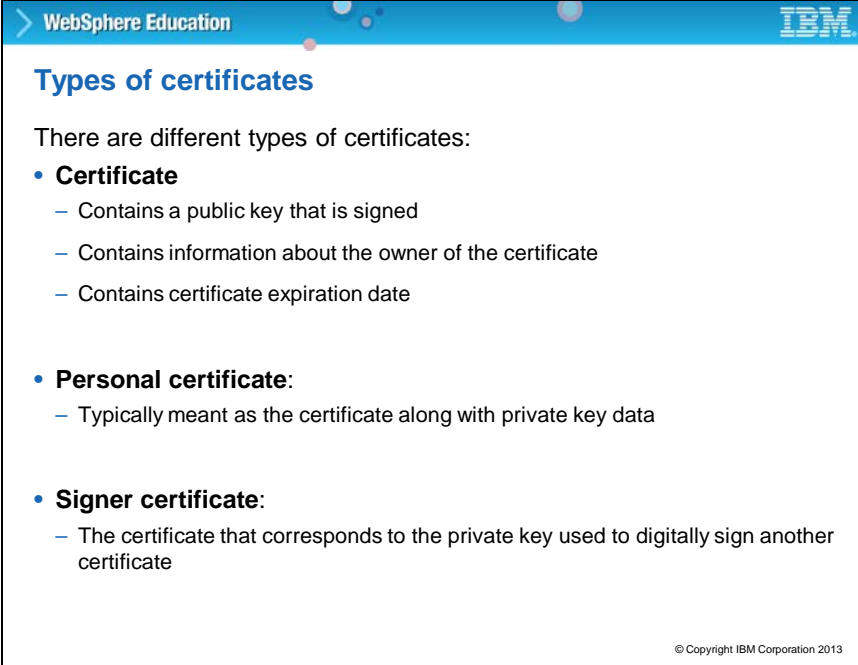
- Includes information about you
- Includes public key
- A certificate authority digitally signs it




© Copyright IBM Corporation 2013

A digital certificate is an electronic document that identifies you; a certificate authority signs it. A certificate contains information about the server, including the public key of the server, and the certificate that a certificate authority digitally signs.

Slide 62



The slide is titled "Types of certificates" and is part of a WebSphere Education presentation. It lists three types of certificates: Certificate, Personal certificate, and Signer certificate, each with a brief description of its contents.

WebSphere Education 

Types of certificates

There are different types of certificates:

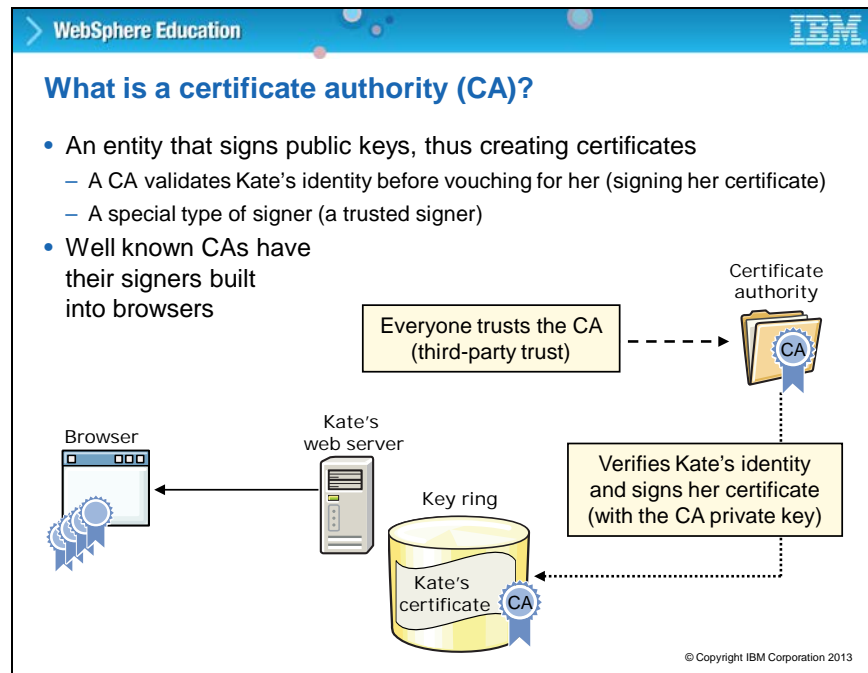
- **Certificate**
 - Contains a public key that is signed
 - Contains information about the owner of the certificate
 - Contains certificate expiration date
- **Personal certificate:**
 - Typically meant as the certificate along with private key data
- **Signer certificate:**
 - The certificate that corresponds to the private key used to digitally sign another certificate

© Copyright IBM Corporation 2013

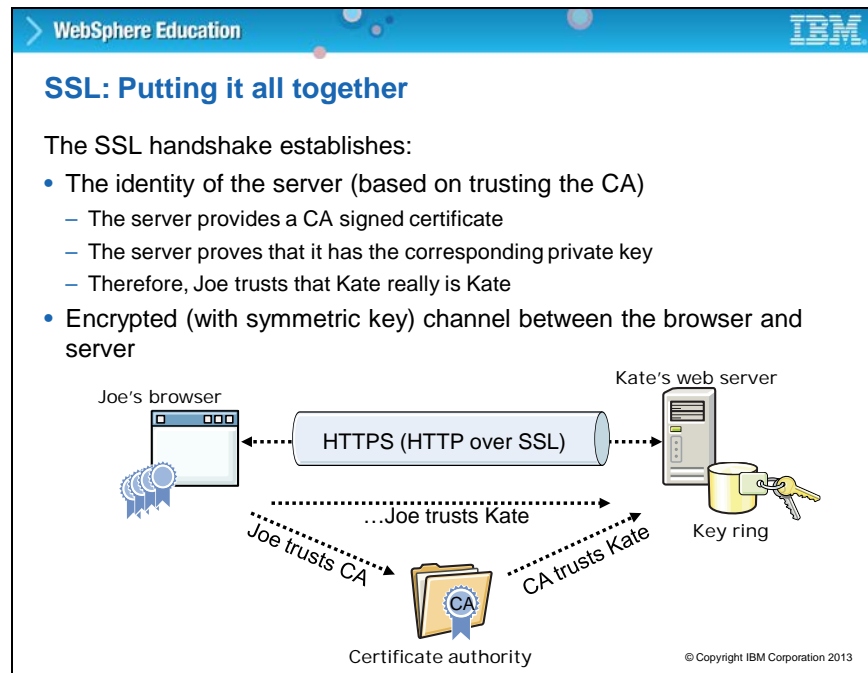
A certificate contains a public key that a certificate authority verifies and signs. It typically contains information about you that a certificate authority verifies.

A personal certificate is a certificate that contains the private key, which must be kept private.

A signer certificate is a trusted certificate, which contains the public key and information about the owner. The private key is used to sign the certificate.

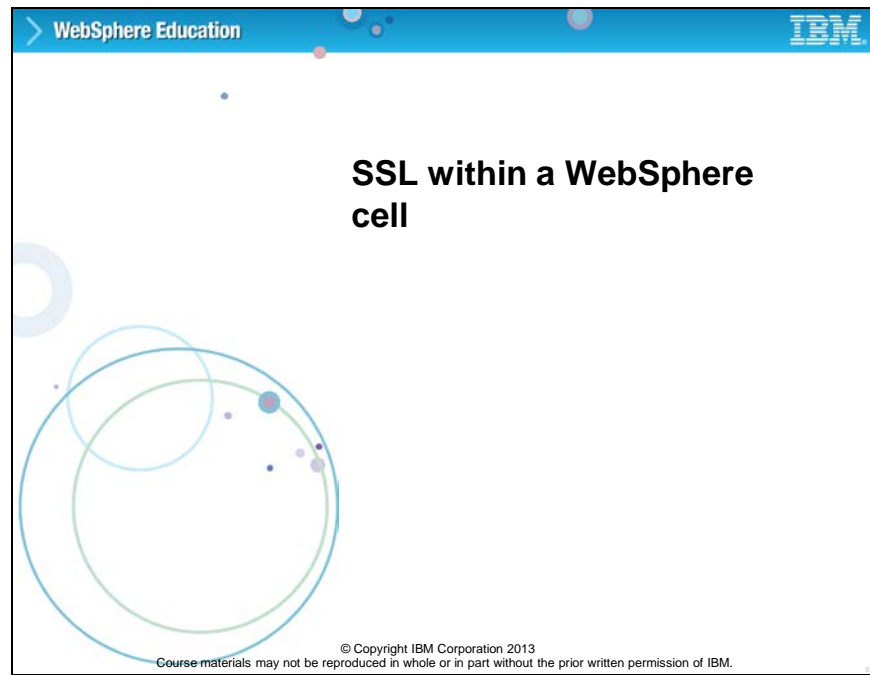


A certificate authority, or CA, is a trusted third party that vouches for your identity. If a client is going to trust a certificate from a server, the client must be able to validate the certificate. This validation is possible because a certificate authority generates the certificates and signs them with the private key of the certificate authority. That means that the web browser can verify the digital signature by using the public key of the certificate authority. The public keys from the standard certificate authorities are built into web browsers. If the web browser does not have a copy of the public key of the certificate authority, the user is prompted.



This diagram puts all of the previous concepts together. Joe is using a web browser and the browser is willing to trust the web server for Kate. Kate presented the certificate to Joe. A certificate authority signed the certificate for Kate and Joe trusts the certificate since a certificate authority signed it. Because of the certificate, Joe trusts Kate.

Slide 65



This topic describes the use of SSL within a WebSphere cell.

WebSphere Education
IBM

SSL within WebSphere Application Server

- SSL can be used to secure network traffic for a number of links
 - From the client to the web server
 - From the plug-in to the application server
 - Other network links can also be secured (LDAP and others)
- The **administrative console** (or iKeyman) can be used to create and manage the necessary keys and keystores
 - Keystores contain digital certificates that are needed for SSL to establish secure communication between two points

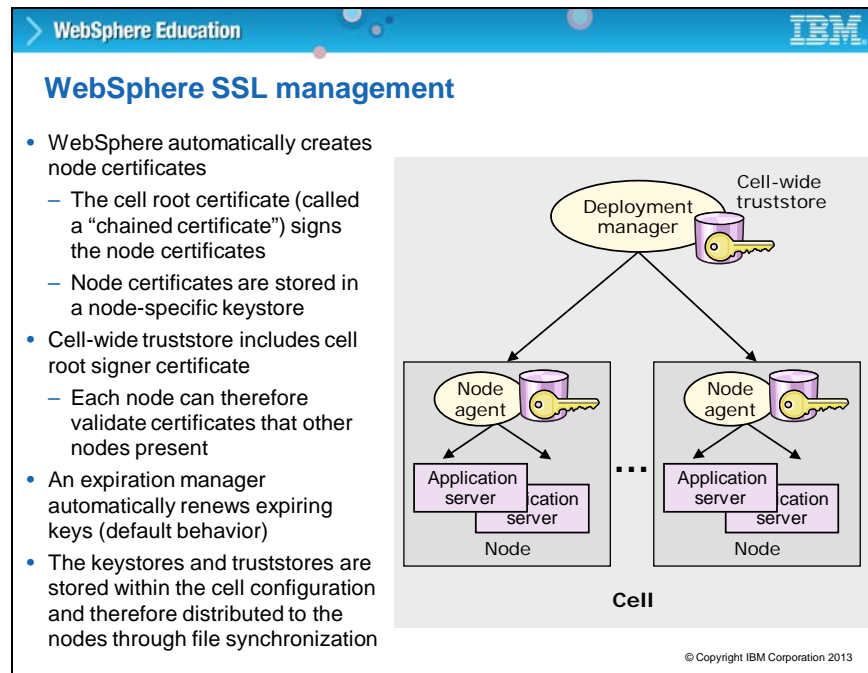
1. Client to HTTPd

2. Plug-in to application server

© Copyright IBM Corporation 2013

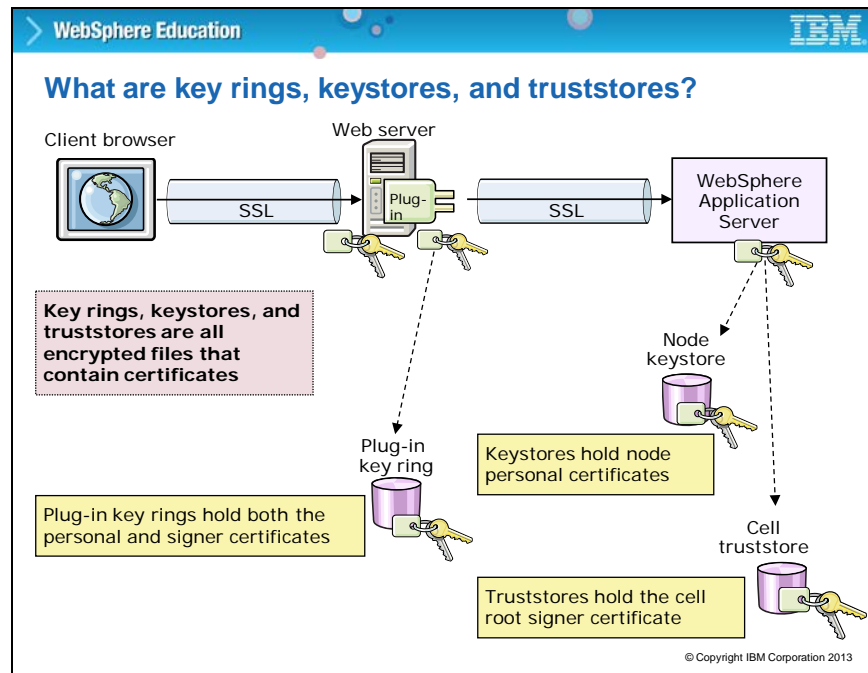
SSL can be used to secure communication between clients and web servers, between the WebSphere plug-in and the application server, and between other components in an intranet. You can use the administrative console, or another utility that is called iKeyman, to create and manage the necessary certificates and keystores. A keystore contains the certificates that are needed for SSL to establish sessions between two points.

Slide 67



You can also configure cell-wide SSL by using a cell-wide truststore. Each node within a cell gets two certificates, a node personal certificate with a one-year lifespan, and a node signer certificate with a 15-year lifespan. The node signer certificates sign the node personal certificates. Therefore, all of the nodes have each others' node signer certificates. They can all validate each other because WebSphere adds each of the node signer certificates into the cellDefaultTrustStore, and that is propagated to all of the nodes through node file synchronization. There are several desirable side effects.

First, is that all of the nodes can securely communicate with each other after validating themselves. Second, when a personal certificate is replaced when it expires, the other nodes still accept the certificate since a known signer has signed it. A new signer does not need to be distributed again as it would, had the certificate been self-signed.

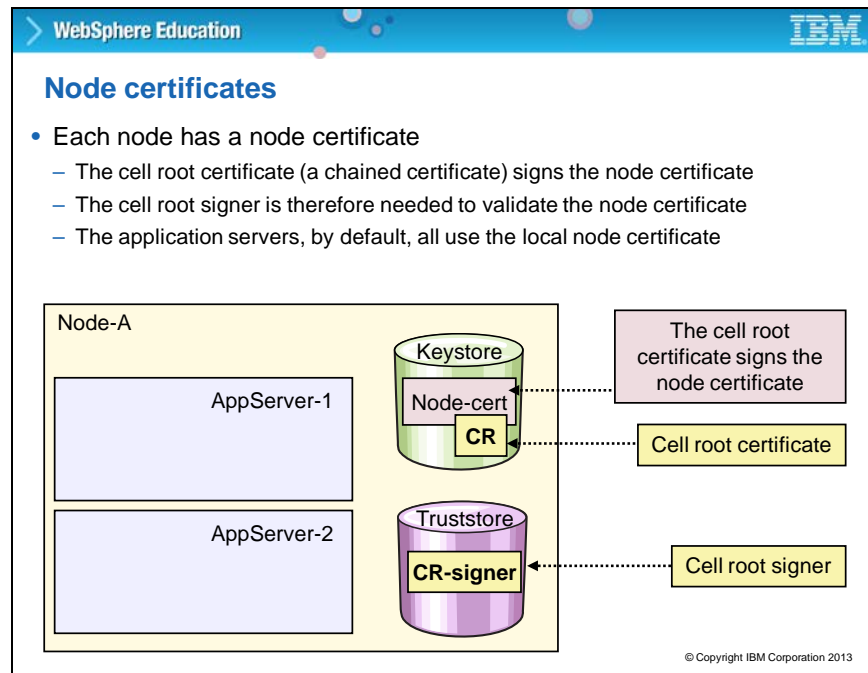


The diagram on the slide shows all the keys and keystores you must create to configure SSL from the client to the application server. The SSL implementation that WebSphere Application Server uses, stores personal certificates in an SSL key file and signer certificates in a trust file. A key file contains a collection of certificates, each one of which is presented during an SSL connection initiation to prove identity. A trust file is a file that contains a collection of certificates that are considered trustworthy and against which the presented certificates are matched during an SSL connection initiation to assure identity. Self-signed keys can actually be more secure than certificate authority certificates for internal communications like the ones between the plug-in and the application server. WebSphere Application Server comes with a dummy keyring, but the dummy key ring is not secure since it is readily available to anyone who has a copy of WebSphere.

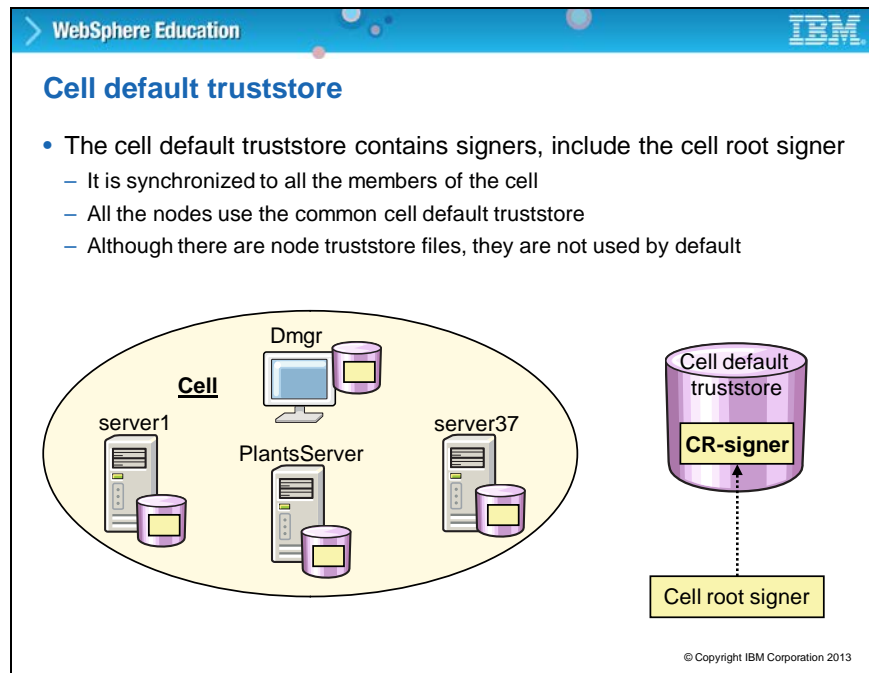
Configure SSL between the plug-in and the application server as follows:

- Create a new key ring for the plug-in, and a self-signed key in the key ring for the plug-in.
- Export the public certificate for the plug-in.
- Create a new key ring and a self-signed key for the application server.
- Export the signing public certificate for the application server.
- Create a trust file for the application server and import the public certificate of the application server into the key ring for the plug-in.

- Import the public certificate for the plug-in into the trust file for the application server.
- Configure the plugin-cfg.xml to point at the new plug-in key ring and stash file.
- Configure the application server to use the new application server key ring and trust file.



A keystore contains the nodes personal certificate, which the cell root certificate signs. The truststore holds the signer certificates that a node chooses to trust. By default, that means the cell root signer certificate.



The cell default truststore contains the cell root signer certificate. It is important since each node needs the cell root signer to validate potential SSL connections. The file is made available to each of the nodes through standard file synchronization.

Slide 71

WebSphere Education
IBM

Managing WebSphere keystores

- Keystores and certificates for the cell, nodes, and plug-ins can be managed directly from the console
- Expiration management
- Keystores
- Trust files
- Certificates

SSL certificate and key management

SSL configurations

The Secure Sockets Layer (SSL) protocol provides secure communications between remote server processes or endpoints. SSL security can be used for establishing communications inbound to and outbound from an endpoint. To establish secure communications, a certificate and an SSL configuration must be specified for the endpoint.

In previous versions of this product, it was necessary to manually configure each endpoint for Secure Sockets Layer (SSL). In this version, you can define a single configuration for the entire application-serving environment. This capability enables you to centrally manage secure communications. In addition, trust zones can be established in multiple node environments by overriding the default, cell-level SSL configuration.

If you have migrated a secured environment to this version using the migration utilities, the old Secure Sockets Layer (SSL) configurations are restored for the various endpoints. However, it is necessary for you to re-configure SSL to take advantage of the centralized management capability.

Configuration settings

[Manage endpoint security configurations](#)

[Manage certificate expiration](#)

☐ Use the United States Federal Information Processing Standard (FIPS) algorithms. Note: This option requires the TLS handshake protocol, which some browsers do not enable by default.

☒ Dynamically update the run time when SSL configuration changes occur

Apply Reset

© Copyright IBM Corporation 2013

Keystores and trust files within a cell can be managed through the administrative console. The screen capture on the slide shows the SSL certificate and key management page where you can manage expiration and other attributes for SSL.

Slide 72

WebSphere Education
IBM

Creating keystores and certificates

Create
Delete
Receive from a certificate authority...
Replace...
Extract...
Import...
Export...
Revoke...
Renew

Select	Alias	Issued To	Issued By	Serial Number	Expiration
You can administer the following resources:					
<input type="checkbox"/>	default	CN=192.168.136.128, OU=was8host01Node02Cell, OU=was8host01Node03, O=IBM, C=US	CN=was8host01, OU=Root Certificate, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US	32492542466710	Valid from Oct 20, 2011 to Oct 19, 2012.
<input type="checkbox"/>		CN=was8host01, OU=Root Certificate, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US	CN=was8host01, OU=Root Certificate, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US	577457852324	Valid from Aug 14, 2011 to Aug 10, 2026.

General Properties

Alias
default

Version
X509 V3

Key size
2048 bits

Serial number
32492542466710

Validity period
Valid from Oct 20, 2011 to Oct 19, 2012.

Issued to
CN=192.168.136.128, OU=was8host01Node02Cell, OU=was8host01Node03, O=IBM, C=US

Issued by
CN=was8host01, OU=Root Certificate, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US

Fingerprint (SHA digest)
97:A6:3A:19:68:F8:13:13:7E:3E:C3:8D:05:83:1D:A

- Keystores for WebSphere are managed through the administrative console
 - Creating keystores
 - Requests and imports CA certificates
 - iKeyman can also be used

© Copyright IBM Corporation 2013

Certificate management that is comparable to iKeyman functions is now integrated into the keystore management panels of the administrative console. Using the administrative console, you can manage personal certificates, certificate requests, and signer certificates. You can still use iKeyman for the various tasks.

The slide shows the administrative console view of a node certificate. It shows the initial information about the node certificate and the signer certificate.

WebSphere Education IBM

What is a chained certificate?

- A chained certificate is merely a certificate that another certificate signs
- The cell root certificate (sometimes called a mini-CA) signs the node certificate

Chained certificate

- The cell root certificate signs it

Select	Alias	Issued To	Issued By	Serial Number	Expiration
<input type="checkbox"/>	default	CN=was8host01.localdomain, OU=was8host01Node01Cell, O=IBM, C=US	CN=was8host01.localdomain, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US	4100991604240	Valid from Nov 16, 2011 to Nov 13, 2021.
<input type="checkbox"/>		CN=was8host01.localdomain, OU=Root Certificate, OU=was8host01Cell01, O=IBM, C=US	CN=was8host01.localdomain, OU=Root Certificate, OU=was8host01Cell01, OU=was8host01CellManager01, O=IBM, C=US	3264426922974	Valid from Nov 16, 2011 to Nov 12, 2026.

Notice: these two are the same

© Copyright IBM Corporation 2013

A chained certificate is a personal certificate that is created by using another personal certificate to sign it. A chained certificate means that the cell root signer signs the node certificate. The slide provides a screen capture with an example of chained certificates.

Slide 74

WebSphere Education

Expiration manager scheduling

Start now

Manual start

General Properties

* Expiration notification threshold: 60 days

☒ Enable checking

Expiration checking

Scheduled time of day to check for expired certificates

21 : 30 ☐ A.M. ☐ P.M. ☒ 24-hour

☒ Check by calendar

Weekday: Sunday Repeat interval: 4 weeks

☐ Check by number of days

* Repeat interval: 7 days

Next start date: Sunday, March 13, 2011 9:30 PM

Expiration check notification

MessageLog

☒ Automatically replace expiring self-signed and chained certificates

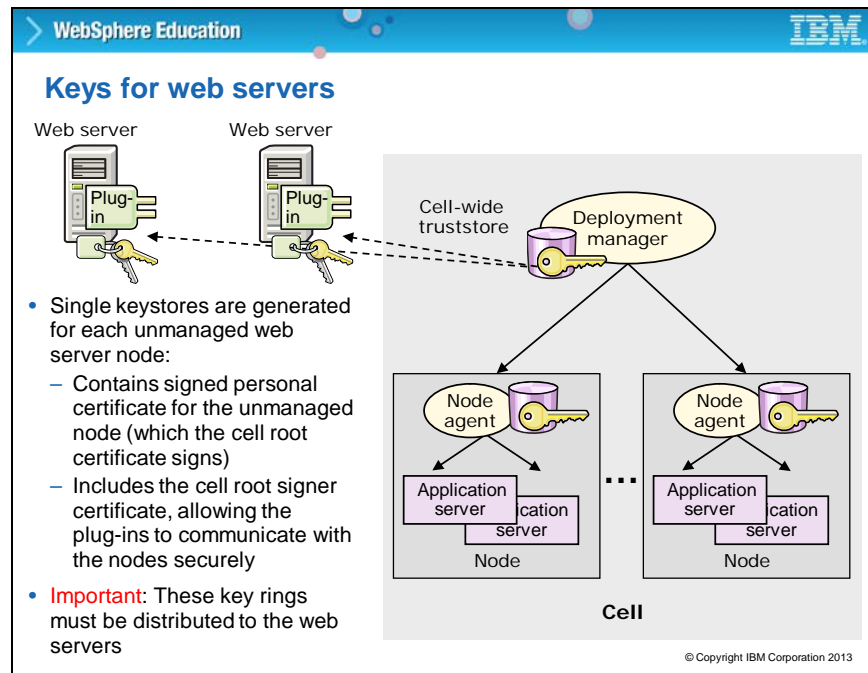
☒ Delete expiring certificates and signers after replacement

© Copyright IBM Corporation 2013

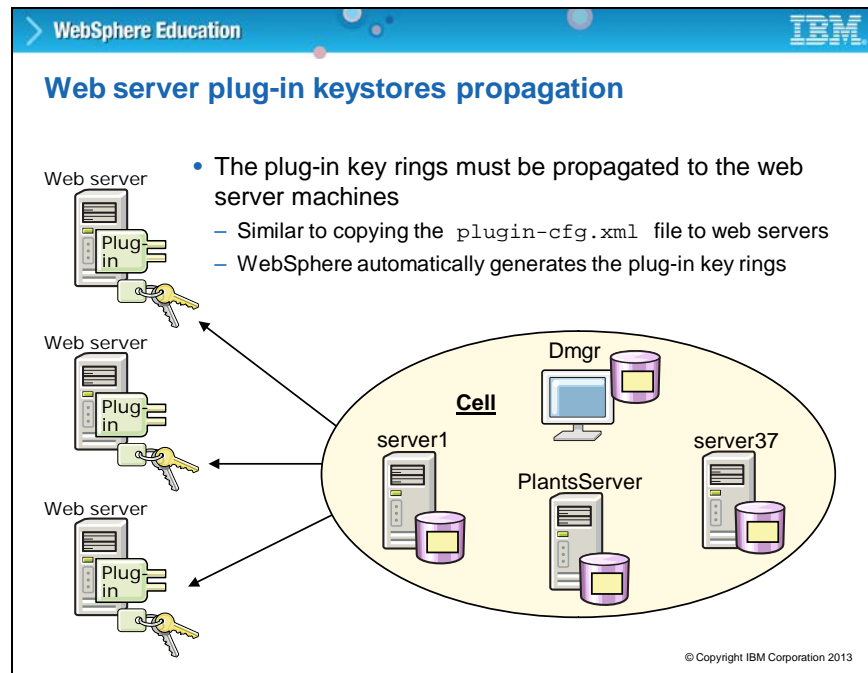
- The expiration manager can be run manually or through a schedule
- Running manually can be useful since you actively monitor the log file and thus generate a list of certificates that are going to expire soon

WebSphere automatically manages certification expiration in the environment. The certificate expiration monitor administrative task is a scheduled task that cycles through all the keystores in the security configuration and reports on any certificates that expire. It also looks for certificates that fall within an expiration threshold and certificates that fall within the pre-notification period that is set.

You can configure the expiration monitor task to run according to a particular schedule.



The key management concept can be extended out to the managed web servers in your configuration. Initially, when a new web server and plug-in are defined within a WebSphere environment, WebSphere generates a keystore for it. This keystore includes a personal certificate for the plug-in that the cell signer certificate signs. The keystore also includes all of the current node signer certificates. This configuration means that the new plug-in can communicate with all the existing nodes. One thing to keep in mind is that the plug-in keystore (just like the plugin-cfg.xml file) must be propagated to the plug-in. Key generations must be done whenever new nodes are added to the cell.



When a new node is added, a new signer certificate is created and added to servers and plug-ins. The administrator must propagate the new plug-in keystores out to the web servers. The keystore propagation is not that serious of an issue, since a new `plugin-cfg.xml` must be propagated as well. If the web server personal certificate expires, a new certificate is generated for the plug-in. However, the administrator must be aware that the expiration is happening and remember to manually propagate the new keystore. The administrator must remember to repropagate new keystores to all of the plug-ins. All of the propagation can be done by using the administrative console.

WebSphere Education
IBM

IBM HTTP Server key ring propagation

Web servers

Web servers > webserver1 > Plug-in properties

Use this page to configure a web server plug-in. The plug-in passes HTTP requests from servers.

Runtime
Configuration

Plug-in properties

☐ Ignore DNS failures during Web server startup

Refresh configuration interval
60 seconds

Repository copy of Web server plug-in files:

Plug-in configuration file name
plugin-cfg.xml
View

☒ Automatically generate the plug-in configuration file

☒ Automatically propagate plug-in configuration file

Plug-in key store file name
plugin-key.kdb

Manage keys and certificates

Copy to Web server key store directory

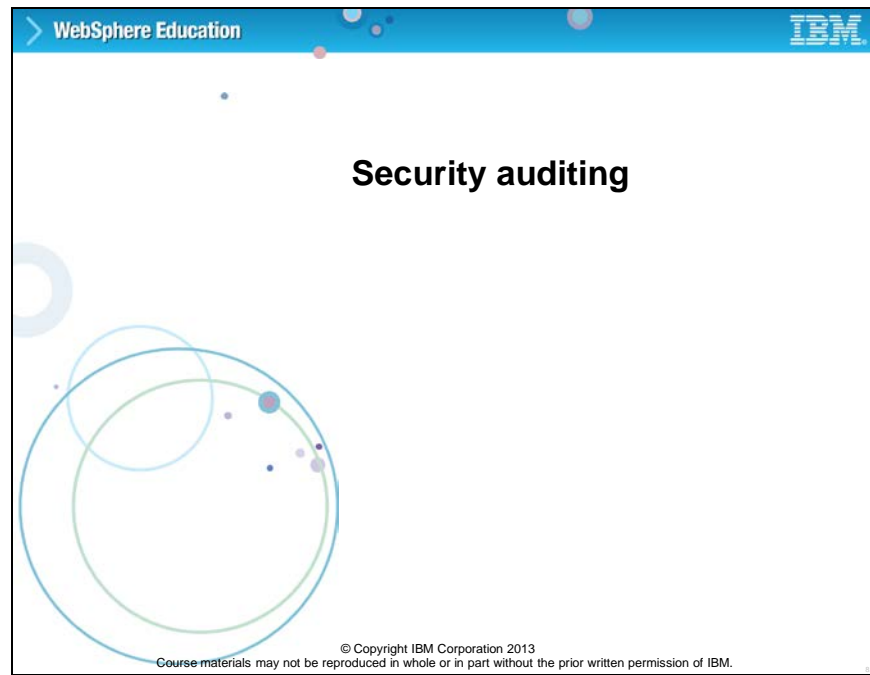
- Web server keystores are automatically generated
 - Can be managed from the administrative console
- The keystore for IBM HTTP Server servers can be remotely propagated

© Copyright IBM Corporation 2013

The screen capture shows the administrative console page where you can configure keystores for the web server plug-in. You can specify the keystore file name and some other attributes. WebSphere automatically generates keystores for defined web server plug-ins, and these keystores then must be copied to the web server.

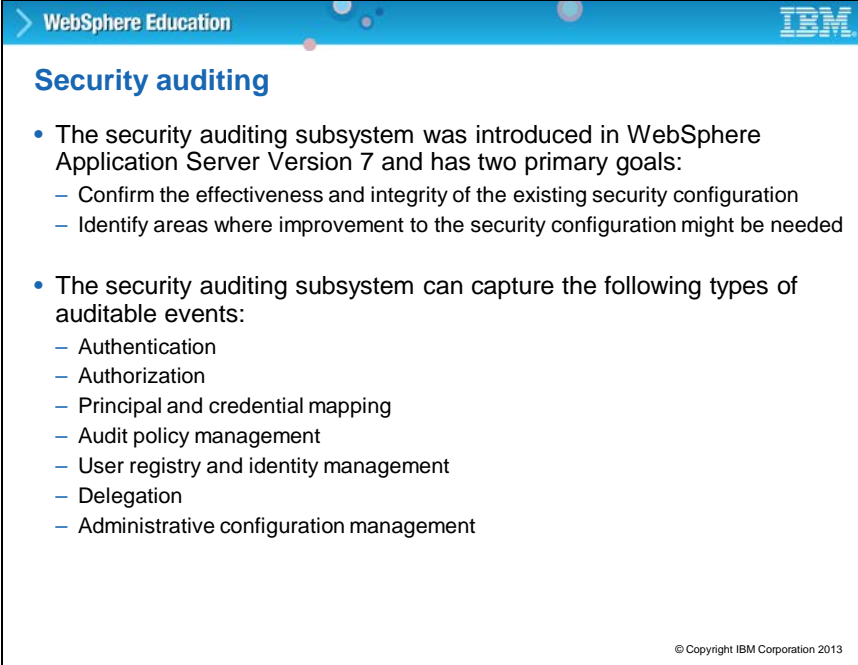
It is also possible to configure WebSphere to propagate the keystore that WebSphere generates to the IBM HTTP Server web server plug-in.

Slide 78



This topic introduces the concept of security auditing.

Slide 79



The slide is titled "Security auditing" and is part of a "WebSphere Education" presentation. It contains two main bullet points. The first bullet point states that the security auditing subsystem was introduced in WebSphere Application Server Version 7 and has two primary goals: to confirm the effectiveness and integrity of the existing security configuration and to identify areas where improvement to the security configuration might be needed. The second bullet point states that the security auditing subsystem can capture the following types of auditable events: Authentication, Authorization, Principal and credential mapping, Audit policy management, User registry and identity management, Delegation, and Administrative configuration management. The IBM logo is in the top right corner, and the copyright notice "© Copyright IBM Corporation 2013" is in the bottom right corner.

- The security auditing subsystem was introduced in WebSphere Application Server Version 7 and has two primary goals:
 - Confirm the effectiveness and integrity of the existing security configuration
 - Identify areas where improvement to the security configuration might be needed
- The security auditing subsystem can capture the following types of auditable events:
 - Authentication
 - Authorization
 - Principal and credential mapping
 - Audit policy management
 - User registry and identity management
 - Delegation
 - Administrative configuration management

© Copyright IBM Corporation 2013

Security auditing was introduced in WebSphere Application Server V7. Auditing allows you to report and track auditable events to ensure the integrity of your system.

Security auditing gives you the ability to log various events. Events include authentication, authorization, principal-credential mapping, audit policy management, user registry and identity management, delegation, and administrative configuration management. The purpose of auditing is to verify the effectiveness and integrity of the security configuration and identify areas for improvement.

All of these auditable events can be recorded into audit log files, which can be analyzed to discover breaches in the infrastructure or potential weaknesses in the security infrastructure.

WebSphere Education
IBM

Enabling security auditing

- Configuration is necessary before auditing can be enabled
 - Create an audit-specific set of console users or groups and map to Auditor role
 - Define notification mechanism (log file, email)
 - Enable monitoring
- Enabling auditing

Security auditing
 Security auditing provides a means to gather and store auditable event records to help assure the integrity of the business computing environment.

General Properties

☒ Enable security auditing

Audit subsystem failure action
 No warning

Primary auditor user name
 wasadmin

☐ Enable verbose auditing

Apply Reset

Related Items

- Event type filters
- Audit service provider
- Audit event factory configuration
- Audit encryption key stores and certificates
- Audit record encryption configuration
- Audit record signing configuration
- Audit monitor

© Copyright IBM Corporation 2013

You can enable auditing by using the administrative console. However, some configuration is necessary for auditing. For example, you can map the Auditor role to a set of console users or a group, as shown earlier in this unit. You can also specify the notification mechanism, which can be a log file, or email. You can enable or disable auditing by selecting the box.

Before you can enable security auditing, global security must be configured in your environment.

WebSphere Education

Viewing audit data

- Audit data can be viewed as:
 - Text
 - An HTML report (through wsadmin)

Start Display Current Environment

```

WebSphere Platform 7.0.0.0 (ND 7.0.0.0 R0815.G1) running with process name was7host01Cell1
Host Operating System is Windows XP, version 5.1 build 2600 Service Pack 3
Java Version = J2RE 1.6.0 IBM J9 2.4 Windows XP x86-32 JVMW13260-20080816_22093 (JIT enab
JVMW - 20080816_021093 IBMHC
JIT - v9_20080721_13301fx2
GC - 20080724_AA, Java Compiler = 39j1c24, Java VM name = IBM J9 VM
was.install.root = C:\Program Files\IBM\WebSphere\AppServer
user.install.root = C:\Program Files\IBM\WebSphere\AppServer\profiles\bmgrProfile
Java Home = C:\Program Files\IBM\WebSphere\AppServer\java\jre
was.ext.dirs = C:\Program Files\IBM\WebSphere\AppServer\java\lib;C:\Program Files\IBM\WebS
Classpath = C:\Program Files\IBM\WebSphere\AppServer\profiles\bmgrProfile\properties;C:\P
Java Library path = C:\Program Files\IBM\WebSphere\AppServer\java\jre\bin;.C:\Program Fi
Current trace specification = ""info
***** End Display Current Environment *****
Seq = 0 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 1 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 2 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 3 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 4 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 5 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
Seq = 6 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = 
  
```

Audit Records - Windows Internet Explorer

ResourceName=placeReport	ResourceType=SM_MBEAN	ResourceUniqueld=0
243	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Wed Jan 28 15:21:21 EST 2009	Action=preinvoke MBean	ProgName=StatusCache
RemoteAddr=192.168.58.135	RemotePort=3570	RemoteHost=192.168.58.135
ResourceName=placeReport	ResourceType=SM_MBEAN	ResourceUniqueld=0
244	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Wed Jan 28 15:22:01 EST 2009	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=192.168.58.135	RemotePort=3570	RemoteHost=192.168.58.135
ResourceName=getProcessType	ResourceType=SM_MBEAN	ResourceUniqueld=0
245	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Wed Jan 28 15:22:14 EST 2009	Action=preinvoke MBean	ProgName=StatusCache
RemoteAddr=192.168.58.135	RemotePort=3570	RemoteHost=192.168.58.135
ResourceName=placeReport	ResourceType=SM_MBEAN	ResourceUniqueld=0

© Copyright IBM Corporation 2013

After auditing is enabled, you can view the security logs as either text files or as HTML reports that are generated through wsadmin.

WebSphere Education
IBM

Securing audit records

- Access to audit configurations is restricted
 - To change audit settings, Auditor access is required (Administrator access is not sufficient)
- Audit data can be digitally protected
 - Can be digitally signed
 - Can be encrypted with a separate audit certificate

Security auditing

[Security auditing](#) > **Audit record signing configuration**

Signing audit records provides a means of tamper-proofing audit records.

General Properties

☒ Enable signing

Managed keystore containing the signing certificate:

CellDefaultKeyStore ((cell):was8host01Cell01)

☐ Certificate in keystore

Certificate alias: default

Security auditing

[Security auditing](#) > **Audit record encryption configuration**

By encrypting the audit records, only a user given the Auditor role can view the audit records.

General Properties

☒ Enable encryption

The Audit keystore containing the encryption certificate:

AuditKeyStore New...

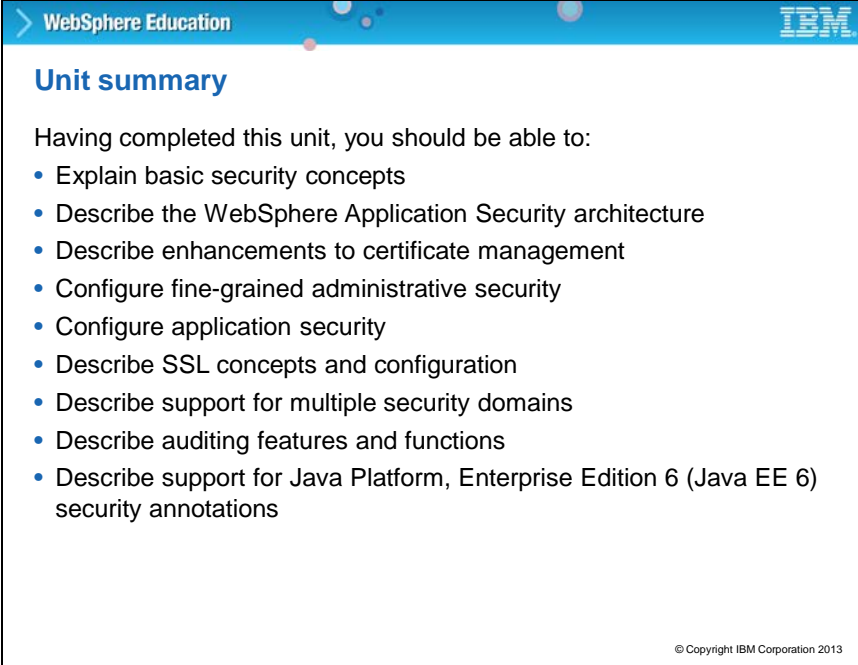
☐ Certificate in keystore

Certificate alias: auditcert

© Copyright IBM Corporation 2013

Since audit records contain important and possibly sensitive information, they can be protected through either digitally signing them or encrypting them. Access to audit configurations is restricted. To change audit settings, you must have the Auditor role. Administrator is not sufficient. As auditor, you can configure signing and encryption settings through the administrative console, as shown on the slide.

Slide 83



The slide is titled 'Unit summary' and is part of the 'WebSphere Education' series, as indicated by the header. It lists the learning objectives for the unit. The IBM logo is in the top right corner. The copyright notice '© Copyright IBM Corporation 2013' is at the bottom right.

Unit summary

Having completed this unit, you should be able to:

- Explain basic security concepts
- Describe the WebSphere Application Security architecture
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe SSL concepts and configuration
- Describe support for multiple security domains
- Describe auditing features and functions
- Describe support for Java Platform, Enterprise Edition 6 (Java EE 6) security annotations

© Copyright IBM Corporation 2013

Having completed this unit, you should be able to:

- Explain basic security concepts
- Describe the WebSphere Application Security architecture
- Describe enhancements to certificate management
- Configure fine-grained administrative security
- Configure application security
- Describe SSL concepts and configuration
- Describe support for multiple security domains
- Describe auditing features and functions
- Describe support for Java Platform, Enterprise Edition 6 (Java EE 6) security annotations