Portal Security and SSO

IBM WebSphere Portal  8.5 Workshop
Agenda

- WebSphere Portal Security Overview
- Authentication
- Authorization
- Additional Security Features and Considerations
Portal Security Basics

- **Security is composed of many elements**
  - Authentication, Authorization, Auditability, Confidentiality, etc

- **Portal Security focal points**
  - Authentication (including Single Sign-On (SSO))
    - Who are you?
  - Authorization
    - What are you allowed to see and do?
Portal Security Basics

- Portal is a J2EE application
- Portal relies on Application Server authentication
  - Thus requires Application Server Security to be activated in production (LDAP) setups
- Portal does its own authorization (through Portal Access Control)
- Portal can leverage in-bound Application Server SSO
- Portal features a Credential Vault for Back-end SSO
  - From portlet to back-end applications
Authentication

- User identify themselves to gain access to Portal
- Most common method: user ID and password
- Once authenticated, portal determine if user is authorized to access portal resources
- Portal supports:
  - Form-based Authentication (Portal default)
  - SSL client certificate authentication
  - Third party authentication (e.g. IBM Security Access Manager)
  - HTTP Basic Authentication (for simple clients not capable of Form authentication)
    - via Portal provided TAI
Authentication

- **Portal authentication is the Application Server authentication**
  - You’re logging on to Application Server

- **Application Server authentication requires a user registry**
  - Federated Repository
  - LDAP
  - Database User Registry
  - Custom User Registry
    - Requires coding

- **Authentication process can be customized**
  - JAAS
  - Portal login actions/commands

- **Application Server authentication can be integrated with external providers (external security managers) using Trust Association Interceptors (TAI)**
  - Security Access Manager / WebSEAL, eTrust SiteMinder, custom
Planning for Single Sign-On (SSO)

- Authenticating a user once and allow subsequent access to applications, systems and networks

- In WebSphere Portal context, there are two SSO realms:
  1) **Client to the Portal** (and other applications):
     - SSO established using
       - LTPA token
       - Authentication proxy
  2) **Portal to backend** applications:
     - SSO established using
       - Credential vault
       - LTPA (only if backend accepts it through the Credential Vault / JCA)
Authentication – Client to the Portal

- **Portal uses Application Server Custom Form Login**
  - Relies on Application Server to authenticate
  - Relies on Application Server to provide the security context

- **Portal requires Application Server LTPA* token functionality**
  - LTPA token is carried to browser by an HTTP cookie
  - Cookies must be accepted and returned by the browser or a Proxy

- **Portal supports Application Server Trust Association Interceptors (TAI)**
  - Security Access Manager/WebSEAL TAI provided by IBM
  - Computer Associates eTrust SiteMinder TAI
  - RSA, Entrust, Oblix TAI provided by respective vendors
SSO – Client to the Portal

Alice

Another WebSphere Application

WebSphere Portal
SSO – Client to the Portal

Alice

Auth Proxy

Another WebSphere Application

WebSphere Portal

Some other Application
Simplified Sign-On

- Portal provides a plug-in (TAI) to handle communication between the Identity Provider (IdP) and WebSphere Portal as the service provider.
- User data and credentials are stored at Identity providers where Applications can retrieve authentication information from.
  - End users can create a userid which can be used across multiple applications thus creating a Simplified Sign-On using this “shared” credential.
- Portal trusts the identity provider and grants the user entrance.
- Portal supports OpenId and OAuth when working with Identity providers
  - Google and Yahoo use OpenId
  - Facebook uses OAuth
Integrating with OpenID Authentication

- **OpenID** describes a protocol for a user identity model.
- Enable Users to authenticate using an existing OpenID with Portal to
  - Authenticate
  - Register
  - Access shared user profile settings
- Provides SSO experience for your site with full control of allowed external services
- Simplify Enrollment and Profile update
  - Retrieve user data from external service

IBM WebSphere Portal
Authentication – Portal to the Backend

- Portal credentials do not necessarily match credentials of backend applications (i.e. SAP)
  - e.g. Portal userid is “fadams” but SAP userid is “adamsf”
- Need a way to store/map portal credentials to backend credentials
  - Credential Vault Service
- Typical usage scenario
  - Portlet prompts new user to enter userid/password to access backend application
  - Portlet stores these credentials using the Credential Vault service
  - In the future, portlet can obtain these credentials from the Credential Vault and would not need to prompt the user
SSO – Portal to Backend Applications

Alice

WebSphere Portal

Portlet

AjaxProxy

Credential Vault

Backend Application

Backend Application
Portal Credential Vault

- A **Portlet Service** for storing and retrieving SSO Credentials, including the user's JAAS Subject that was built during login.

- A **Portlet Service** for retrieving SSO Credentials other than userid and password.

- A Vault adapter **interface** to integrate vault implementations like the Tivoli Access Manager Global Sign-On Lockbox.
More on Credential Vault

- **Credential Vault Portlet Service**
  - Stores credentials on behalf of users, portlets or entire Portal
  - Supports *Passive* credential objects

- **Credential Vault Adapter Service Provider Interface**
  - Provides interface for credential storage mechanism (2 impl’s):
    - Database (default) and Tivoli Global Sign-On Lockbox
    - Default vault implementation Base64 encodes passive credentials
    - Obfuscation via a plugable exit

- **Portlets must be coded to use vault**
  - Only Portlets (or code running in Portal context) can access vault
  - No out-of-the-box function for end-user password management
    - Portlet responsibility, using sample code provided
Impersonation

- **Acting as a different user**
  - Portal tracks information about original user
  - Other systems will only recognize impersonated user (e.g. WebSphere)

- **Protected by Access Control**
  - Controls who can be impersonated by whom

- **Public APIs to identify Impersonator**
Impersonation Sample Scenario

Alice experience problem with their website and calls help center

Triggers impersonation and sees Alice's problem

Bob
Impersonation Sample Scenario

Bob

Triggers impersonation of Alice

Check accesscontrol

Logout current user(Bob)/invalidate session

Login Alice/Add Bob to Security Subject

Trigger Audit Event/redirect

Send redirect with new Security token

single request
Outbound HTTP Connections

- Central control for outbound HTTP connections
- Functions for authentication and cookie handling
- Administration via model APIs or configuration tasks
- Custom outbound service filter possible
- New in v8.5
Authenticating Outbound HTTP Connections

- Access to remote hosts needs to be protected
- **Outbound HTTP connection infrastructure provides internal authentication support for**
  - HTTP basic authentication
  - HTTP digest authentication
  - NTLM authentication (Windows NT LAN Manager)
  - SSO via SPNEGO token
  - SSO via LTPA tokens
  - Form-based authentication
Federated Repository

- **Federated repositories are:**
  - Federation capabilities provided by VMM
  - Integrated with WebSphere Application Server security
    - As an user registry option

- **Ability to use multiple repositories simultaneously for user registry**
Virtual Member Manager (VMM)

- Integrated in WebSphere Application Server V8.x
- VMM provides a model of the organizational entities and hides from them the disparate repositories
- VMM provides a common model, secure access to various brands and types of repositories, and the ability to use repositories with existing data.
- Out of the box VMM adapters:
  - File-based repository
  - LDAP repository
  - Database repository
VMM
Create Groups Dynamically

Build Dynamic Groups Inside of WebSphere Portal

- Rule-based groups
- Group definition outside of Repository
- Not managed by standard user repository
- Available since 8.0 or via Solution Catalog (7.0)
Community Integration

- IBM Connections Communities are available at WebSphere Portal via Puma and VMM
Multiple LDAP and Multiple Realm Support
Concept of Realms

- Allows you to aggregate users from one or more LDAP tree of user registries and expose them as a coherent user population to WebSphere Portal
- Referred to as horizontal partitioning
- Advantages:
  - Flexible user management with various configuration options
  - e.g. combine principals from one or more corporate LDAP tree
- Realms can overlap, which allows users to be members of more than one realm
  - Users can be members of multiple realms
  - Try to limit the usage of overlapping realms
- A realm can aggregate one or more nodes in a user registry
- A realm can combine multiple suffixes of one user registry
  - A suffix of a user repository can belong to one or more realms
  - The LDAP suffixes of the individual users must match the suffixes of the groups to which they belong
Multiple Realms For Virtual Portals

- Requires Federated Security
- VMM can dispatch calls to multiple user registries
- Realms can point to (subset of) a specific user registry or to (subsets of) multiple user registries
- User IDs need to be unique across all registries
Cleanup End User Artifacts

- **Two tools to use**
  - UserCleanup via XMLAccess
  - Memberfixer for WCM content

- **Helping you to keep end user customization after user backend changes**
  - DN changes
  - UniqueID changes
  - Cleanup or Data migration
What is Authorization (aka Access Control)?

Who is allowed to perform which action on which resource?

Authentication --> Unique User ID

Portal Resources examples: page, portlet

Examples:
view, edit, delete

Access Control in Portal is **role based** e.g. Alice is Editor@PageA and User@PortletB
Authorization

- **Portal does its own authorization (access control)**
- Portal is not integrated with Application Server Authorization
- Except in that it is based on the identity authenticated by Application Server

- **Portal provides Resource level authorization granularity**
  - Instance-based

- **Portal authorization can be integrated with external authorization (security) providers**
  - Security Access Manager, SiteMinder
  - Integration API is private
Access Control Model

Portal Resources, for example a Page

Permissions

Roles

Groups

Users

exposes

view edit manage delete ...

accessible

User Editor Manager ... Draft Creator Reviewer

Mapped to
Access Control Model

Inherits Access rights from Parent

Control inheritance via Propagation and Inheritance Role Blocks
Portal Access Control – Resource Permissions

- **Who** is allowed to perform which **action** on which **resource**?

<table>
<thead>
<tr>
<th>Roles</th>
<th>Allow Propagation</th>
<th>Allow Inheritance</th>
<th>Edit Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Security Administrator</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MARKUP_EDITOR</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Manager</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Editor</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Contributor</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Privileged User</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draft Creator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reviewer</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Resource** (Pages, Portlets, etc)

**Who?** (User, User Groups)

Roles
(View, Edit, Delete etc)
Role Concept

- Provide task permissions for users on resources
- **Denoted as** *Role@Resource*
  - Example: *Editor@Portal Page* – allows a user to view, modify and create resources
- **Hierarchically organized**
  - Roles higher in hierarchy inherit permissions of child roles
  - Each resource inherits the role assignments of its parent resource
- **Assigned to users and groups contained in the user registry**
  - Can have multiple roles on the same resource
Portal Roles

- **No Role Assigned** cannot interact with a resource
- **Users** are allowed to view portal content (resources)
- **Privileged Users** are allowed to view portal content, customize portlets and pages and create new private pages
- **Contributors** are allowed to view and create new content ONLY
- **Markup Editors** are allowed to change HTML source for static portal pages.
- **Editors** are allowed to create and edit shared resources
- **Managers** are allowed to create, edit, and delete shared resources
- **Can Run as User (user impersonation)** role allows to view resources and other portal components as another user.
- **Delegators** are allowed to grant access (roles) to other principals
- **Security Administrators** are allowed to create and delete role assignments on resources. Can act as a delegated administrator for a resource. No view/edit access to the resource.
- **Administrators** are allowed to do everything – unrestricted access
Resource Ownership

- Each resource can have an owner (a single user or a single group)
  - assigned when a resource is first created
  - can be changed later with the XMLaccess configuration tool or via the Resource Permissions portlet

- Non-private resources
  - resource ownership equivalent to the Manager role
  - same set of allowed actions

- Private resources
  - resource ownership equivalent to the Privileged User role
  - same set of allowed actions plus the ability to delete the resource
  - can be owned only by users
  - resource ownership cannot be inherited
  - no roles can be assigned
Creation of Shared (non-Private) Resources

- User creates a shared resource
- The user that created the resource becomes the owner of the resource
- The ownership grants specific permissions on the corresponding resource
  - equivalent to the Manager role
- Ownership can be changed
  - using XMLAccess tool or the Resource Permissions portlet
- Owner permissions are never subject to inheritance
Creation of Private Resources

- Users can be granted privileges to create private Pages
- The user that created the private page becomes the owner of the new page
- The ownership grants specific permissions on the corresponding resource
  - equivalent to the Privileged User role
- Private resources are visible only to the owner of the resource
- Private resources do not inherit any roles from their ancestor nodes
- Private Resources are deleted explicitly by the owner or automatically when the creator is removed from the portal
- Ownership can be changed
  - using XMLaccess tool or the Resource Permissions portlet
Attribute Based Security

- For WCM content
- Permission can be granted in addition to user/group based rights
- Public API plug point for custom code in Core Access Control Layer
- Custom code can decide based on attributes controlled by business users
- Available since v8.5
StepUp and RememberMe

**StepUp Framework**
- Predefined authentication levels: identified, authenticated, standard
- Allows to plug custom code for enforcing custom levels
  - e.g. enforce SSL or client-side certificate
- Authentication levels can be set for **portlets** and **pages**

**RememberMe Cookie**
- Persistent cookie to recognize user without login
  - if RememberMe support is activated, the portal login portlet shows a checkbox for setting the cookie
  - If cookie is present, portal treats the user as “identified” but not yet “authenticated”
  - After logout, a user is treated as “identified”
- Portlets can show personalized content
- For accessing protected resources, user has to authenticate

**Enabled via ConfigEngine task**
- The task is either “enable-stepup-authentication” or “enable-rememberme”.

**Administration of authentication levels**
- via UI (Portal Administration >...> Resource Permissions
- via XML Access
Reference materials

- IBM Knowledge Center (Community Articles, Product Documentation, Learning Center)
  - IBM WebSphere Portal & Web Content Manager 8.5:
    http://www-01.ibm.com/support/knowledgcenter/SSHRKX_8.5.0/welcome/wp_welcome.html

- DigitalExperience Developer on IBM developerWorks

- Portal Zone on developerWorks

- IBM Collaboration Solutions Catalog
  - https://greenhouse.lotus.com/catalog

- Upgrade Central

- IBM Software Events
Questions