
ORACLE®

ADF, Security 1-0-1

Addressing OWASP Top-10 Security Vulnerabilities in Oracle ADF

Frank Nimphius
Senior Principal Product Manager
Oracle Mobility and Development Tools
June 04, 2014

Safe Harbor Statement

The presentation has been created for user conferences with presentation times limited to below 60 minutes. The session discusses security options and features available in Oracle ADF that help mitigating security risks published in the OWASP top-10 list of security vulnerabilities for the year 2013. No guarantee can be given that the set of recommendations expressed in this session is complete and that it does provide sufficient protection for the security threats listed in the OWASP top-10.

"The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted."

– OWASP, Top-10 Security Vulnerabilities 2013

OWASP Top-10 Security Vulnerabilities 2013

1. SQL Injection
2. Broken Authentication & Session
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Known Vulnerable Components
10. Unvalidated Redirects and Frowards

Where is the single button to protect ADF applications?

The Application Developer Defense Shield

Education: Awareness, Risk Analysis

Secure Code Guidelines: Security by Design and Default

Tools: Application, System, Network

Techniques: Design Patterns and Principles

Pattern – The Good Taste in Software Development

- Defense in depth
- Least privileged access
- Single access point
- Check point
- Roles
- Full view with errors
- Limited view
- Session

Creativity and Security are no Worlds apart

The Creative Security Developer View on ADF

The ADF Security Toolbox

- ADF Faces
  - Converters and Validators
  - Phase Listener
  - Servlet, Servlet Filter
  - WEB-INF
- ADF Controller
  - Bounded Task Flows
  - Task Flow Templates
  - Managed Beans

The Creative Security Developer View on ADF

The ADF Security Toolbox

- ADF
  - ADF Security
  - Error Handler
  - Logging
- ADF Business Components
  - Validators
  - Lifecycle Methods

The Creative Security Developer View on ADF

The ADF Security Toolbox

- RDBMS
  - Stored Procedures
  - PL/SQL Triggers
  - VPD, Proxy Users

If there is a top-10 list of vulnerabilities, is there also a top-100?

Let's Start with 10!

OWASP #1 – SQL Injection

The Vulnerability

- Ability for users to provide SQL strings as an application input
  - Potentially allows users to issue DML commands
- The Good Will
  - … where DEPARTMENT_ID = 60
- THE BAD Intent
- … where DEPARTMENT_ID = 999 OR 1=1; Drop table EMPLOYEES CASCADE CONSTRAINTS

OWASP #1 – SQL Injection

Oracle ADF Addressing

- ADF Business Component Queries
  - Use View Criteria to define the Where Clause
  - Use bind variables
- ADF Business Components entities
  - Validate attributes for known SQL strings
- Dynamic View Objects
  - Don't create dynamic view objects based on user provided queries!
- Client Methods
  - Validate input arguments

OWASP #1 – SQL Injection

Oracle ADF Addressing

- ADF Faces
  - Filter input text that contains SQL keywords
    - Declarative ADF Faces RegEx validator
    - Java validator configured on component
    - Converter
  - Be aware of immediate="true" in context of validation
- For parameter forms based on client methods
  - Define regular expression validation on attribute binding

OWASP #2 – Broken Authentication and Session Management

The Vulnerability

- The Web is Stateless
  - State is remembered through tokens, cookies and URL encoding
- Risk for 3rd party to access session and authentication information
  - Login interception, Password guessing
  - Trust evidence interception
- Risk of "Building your Own"
- Oracle ADF risk surface
  - Pillar Architecture Pattern
  - Man-in-the-Middle

OWASP #2 – Broken Authentication and Session Management

Oracle ADF Addressing

- ADF delegates session management and authentication to Oracle Platform Security Services (OPSS) in WLS
  - Authentication and session management dealt with by experts
- Use SSO
- Use SSL
- Ensure strong passwords
  - Usually the meaningless hard to remember passwords we all like
  - Expire passwords and use password history
- If using tokens use them securely

OWASP #3 – Cross-Site Scripting (XCSS)

The Vulnerability

- Benefits from unvalidated user input
- JavaScript or HTML code injected to website
  - Get access to user data like cookies or hidden field values
  - Show content good for phishing attacks
  - Invoke application actions on the authenticated user behalf
  - Automatically redirect users or launch popup

OWASP #3 – Cross-Site Scripting (XCSS)

Oracle ADF Addressing

- Use regular expressions to detect scripts and markup
  - ADF Business Components
    - Java Method validator
    - Declarative validator
  - Data Control & Binding Layer for non-ADF BC services
  - On ADF Faces input components
    - af:validateRegExp
  - Simple example finding opening "<" and closing ">" brackets
    - (\%3E)|(\%3e)|(\%3C)|(\%3c)|(<)|(>)
  - Be aware of immediate="true"

OWASP #3 – Cross-Site Scripting (XCSS)

Oracle ADF Addressing

- Encode all data output
  - ADF Faces output component does this by default
- Use JSF converter to remove characters that are not allowed in this context
  - Blacklist
- Think twice before disabling ADF Faces output escaping

<af:outputText>

UIComponent class: oracle.adf.view.rich.component.rich.output.RichOutputText
Component type: oracle.adf.RichOutputText

The outputText component supports styled text. The text can optionally be left unescaped, and supports conversion to and from

Code Example(s)

```
<af:outputText value="AFFieldText" styleClass="AFFieldText"/>
```

Events

Type	Phases
org.apache.myfaces.trinidad.event.AttributeChangeEvent	Invoke Applica Apply Request

Supported Facets

Name	Description
context	Location for contextual information. A

Attributes

Name	Type	Supports EL?	Description
attributeChangeListener	javax.el.MethodExpression	Only EL	a method reference to an attri request. An example of an att
binding	oracle.adf.view.rich.component.rich.output.RichOutputText	Only EL	an EL reference that will store
clientComponent	boolean	Yes	whether a client-side compone presence. Client component o there is a performance cost to
converter	javax.faces.convert.Converter	Yes	a converter object
customizationId	String	Yes	This attribute is deprecated. T

What about client side validation with JavaScript?

OWASP #4 – Insecure Direct Object References

The Vulnerability

- Exposing sensitive information by insecure direct references
  - Reports
  - Images, Photos
  - Slide share
- Example code that would either need protection or verification that information produced is public

```
<af:image source="/imageservlet?thumbnail=#{row.ProductId}"
                id="i7" shortDesc="#{row.ProductId}"/>
```

OWASP #4 – Insecure Direct Object References

Oracle ADF Addressing

- Use protected folders and resources access
  - ADF Security for Authorization
  - Java EE security constraints
  - WEB-INF directory
- Use indirect object references
  - "Virtualize" access to sensitive documents
    - Map random generated names to a document and objects
    - Expire document references
    - Use content management systems

OWASP #4 – Insecure Direct Object References

Oracle ADF Addressing

- Apply label security for objects queried from database
  - Use query predicate through VPD
  - Use Groovy on ADF BC bind variables to filter data queries based on the authenticated user
- Use random, non-sequential and not-so-easy-to-predict names for objects and object references

OWASP #5 – Security Misconfiguration

The Vulnerability

- Configuration that disables security on one of the following levels
  - Application
  - Transport Layer / Network
  - Server / System
- Oversights when moving from testing to production
- Disabling constraints for batch updates

Laziness is hard to control

OWASP #5 – Security Misconfiguration

Oracle ADF Addressing

- Use 4-Eyes principle
- Have security check list
- Implement Security by design as the default

OWASP #5 – Security Misconfiguration

Oracle ADF Addressing

- ADF Security enabling

```
<sec:JaasSecurityContext initialContextFactoryClass="oracle.adf.share.security.JAASInitialContextFactory"
                         jaasProviderClass="oracle.adf.share.security.providers.jps.JpsSecurityContext"
                         authorizationEnforce="true" authenticationRequire="true"/>
```

- Security on bounded task flows is configured on the assembling application
- Ensure all security test-role (usually granted to anonymous) are removed
- Avoid implicit defaults and instead explicitly set default values
  - Bounded task flow URL Invoke
  - Bounded task flow library internal

Btw. What is the default setting for URL invoke?

OWASP #6 – Sensitive Data Exposure

The Vulnerability

- Exposure of data that is confidential to the data owner, a job or business responsibility or the business as a whole
- Exposure of critical data that is hard to recover from damage, manipulation or loss
- Often caused by
  - Insecure data storage
  - Unencrypted user passwords
  - Unauthorized queries and query tampering
  - Unprotected data transport

OWASP #6 – Sensitive Data Exposure

Oracle ADF Addressing

- Use strong user password pattern that expire frequently and that are stored securely in a serious identity management system
- Ensure user sessions time out when left idle
  - Use pillar architecture to define different time out settings based on sensitivity of the application module
  - Use ADF Faces session timeout warning with short session timeout settings
  - Don't pin sessions using af:poll
    - Expire af:poll after some time so the session can expire
- Control error handling messages

OWASP #6 – Sensitive Data Exposure

Oracle ADF Addressing

- Filter data queries for authenticated usres
  - Use query predicates
    - VPD, Groovy
  - Use ViewCriteria
- User Interface
  - Use *rendered="false"* vs. *visible="false"*
  - Use ADF Security in Metadata Services (MDS) to ensure customization doesn't add page content for unauthorized users
    - MDS should never remove sensitive information but add at most

OWASP #7 – Missing Function Level Access Control

The Vulnerability

- URLs that trigger actions
  - http:// …. /somePage?action=delete
- Access to application views outside of the intended navigation context
- Java methods that bypass entity validation and protection
  - E.g. prepared JDBC statements
  - Service calls
- Javascript functions that issue Ajax calls
  - af:serverListener

OWASP #7 – Missing Function Level Access Control

Oracle ADF Addressing – Guard and Guide

- "Guard and guide" all application access
  - Single or reduced point(s) of entry
  - Checkpoints

OWASP #7 – Missing Function Level Access Control

Oracle ADF Addressing – ADF Security

- Protect Bounded Task Flow
- Protect views
  - ADF Security
  - Wrap views in bounded task flows
- ADF Security Expression Language
  - ADF Faces UI components, Task Flows
- Use ADF Security from Java
  - Managed Beans, Phase Listeners, MDS objects

OWASP #7 – Missing Function Level Access Control

Oracle ADF Addressing: Use Custom Resource Permissions

- Create custom Permissions based on the OPSS Resource Permission
  - Provide policy based access control beyond entity and view protection
- Custom Resource Permission
  - Permission Type
    - Defined declaratively using Name, ResourcePermission class, Actions
  - Resource Permissions
    - Name, reference to resource type, actions, grant to application role
- Evaluated in EL and Java

OWASP #7 – Missing Function Level Access Control

Oracle ADF Addressing: ADF Security Context accessible from EL and Java

- Security Expression Language
  - `#{securityContext.authenticated}`
  - `#{securityContext.userName}`
  - `#{securityContext.userInRole['roleList']}`
  - `#{securityContext.userInAllRoles['roleList']}`
  - `#{securityContext.taskflowViewable['target']}`
  - `#{securityContext.regionViewable['target']}`
  - `#{securityContext.userGrantedResource['permission']}`
  - `#{securityContext.userGrantedPermission['permission']}`
- Security Java API

OWASP #8 – Cross-Site Request Forgery (CSRF)

The Vulnerability

- The ability to invoke requests on behalf of an authenticated user
  - Phishing
  - XSS injection
- Exploits and manipulation of view states

OWASP #8 – Cross-Site Request Forgery (CSRF)

Oracle ADF Addressing

- Common Java EE counter measure
  - Add page token as hidden field and track token state in session
  - Change token with each request
  - Reduces attack window
- Page token: JavaServer Faces implementation
  - PhaseListener
    - RESTORE_VIEW to read and compare token in request
    - RENDER_VIEW to renew token

OWASP #8 – Cross-Site Request Forgery (CSRF)

Oracle ADF Addressing

- XSS prevention
  - XSRF and XSS are related
  - Closing the attack window for XSS narrows it for XSRF
- ADF Faces view state as token (default)
  - Token changes per-view
  - Trinidad StateManager uses strong cryptographic values to protect POST requests from CSRF attacks
- JSF 2.2 provides view state encryption functionality
  - On by default

OWASP #9 – Using Known Vulnerable Components

The Vulnerability

- The use of software versions that have known vulnerabilities
  - Software defects are reported by vendors or ethical hackers on the Internet
    - Usually to the time a fix is available
  - Example: OpenSSL heartbleed bug
- Using old software that no patches are provided for
  - Example: XP

OWASP #9 – Using Known Vulnerable Components

Oracle ADF Addressing

- Plan ahead and upgrade your software
  - From Jdeveloper & ADF 11g Rx to JDeveloper & ADF 12.1.3
- Join a community to keep up to date
  - Enterprise Methodology Group
  - German ADF community
  - JDeveloper and ADF forum on OTN
- Have a customer support contract
  - Information
  - One-off patches

OWASP #10 – Unvalidated Redirects and Forwards

The Vulnerability

- Applications redirect requests to other applications
  - ADF pillar architecture
- Applications called from non-ADF applications
  - Example: ADF – Oracle Forms , ADF – Apex interaction
- Redirects to resource servlets
  - Example: servlet to stream
    - Images
    - Documents (Reports)

OWASP #10 – Unvalidated Redirects and Forwards

Oracle ADF Addressing

- Ensure any request that is issued or received through a direct GET request is validated for the identity of the request origin or target
- If data needs to be passed to a Java EE application, ensure it is passed such that it cannot be tempered with and so the receiving application knows how to verify the validness of the data.
- Avoid any GET request or redirect for navigation in ADF applications. Use post-back navigation only.
- Limit the number of access points to the ADF application. This is easily achieved by using bounded task flows only in an ADF

OWASP #10 – Unvalidated Redirects and Forwards

Oracle ADF Addressing

- Bounded task flows that can be accessed from browsers directly should save input parameter values in managed beans in pageFlowScope and not directly in memory attributes
  - Allows validation and security verification on the provided input parameter
- Bounded task flows should verify request before executing task flow content
  - Router activity or method activity as the default activity
- ADF pages that are configured as bookmarkable should use a converter on all input parameters

Watch Your Back!

Hardware and Software
Engineered to Work Together


ORACLE®