

Dimitri Gielis

APEX Security

101 (mobile)



www.apexRnD.be



dgielis.blogspot.com



[@dgielis](https://twitter.com/dgielis)



dgielis@apexRnD.be

Dimitri Gielis

- ❖ Founder & CEO of APEX R&D
- ❖ 17+ years of Oracle Experience (OCP & APEX Certified)
- ❖ Oracle ACE Director
- ❖ “APEX Developer of the year 2009” by Oracle Magazine
- ❖ Presenter at Oracle Conferences (OOW, ODTUG, OGh, UKOUG, ...)



http://dgielis.blogspot.com

@dgielis

Dimitri Gielis Blog (Oracle Application Express - APEX)
I created this Blog to share my knowledge especially in Oracle Application Express (APEX) and my feelings ...

FRIDAY, SEPTEMBER 26, 2014

APEX 4.2.6 patch available

Oracle released [APEX 4.2.6](#) yesterday, it's also available from My Oracle Support as a patch set 19595896.

Here's the table of bug fixes.

Table 6 Bugs Fixed in the Oracle Application Express 4.2.6.00.03 Patch Set

Bug Number	Description
18393429	Importing pre-4.2 application with data loading results in runtime error
18609856	APEX_WEB_SERVICE.CLOBBASE642BLOB converts incorrectly
18658967	Corrupted HTML mail messages are created from APEX_MAIL.SEND
18719750	ORA-01858 during application import
18733989	Errors picking up session with cookie when TOP CALL is not F, SHOW, or ACCEPT
19021052	Oracle Application Express 4.2.4 throwing error while generating check boxes
19313700	Check box fails with ORA-06502 after upgrade from release 4.0 to 4.2.4
19482775	ORA-06502 when importing application containing translations

Applying the patch took less than 7 minutes.

```
Session altered.
...Validating Application Express
...Database user "SYS", database schema "APEX_040200", user# "106" 20:48:40
...Compiled 0 out of 3014 objects considered, 0 failed compilation 20:48:40
...271 packages
...263 package bodies
...452 tables
...11 functions
...16 procedures
...3 sequences
...457 triggers
...1320 indexes
...211 views
...0 libraries
...6 types
...0 type bodies
...0 operators
...0 index types
```

My next public appearances:

- [Oracle Open World: 27-SEP/2-OCT-2014](#)
- [ORCLAPEX-BE Meetup: 19-NOV-2014](#)
- [ODTUG KScope: 21/25-JUN-2015](#)

About Me

[Dimitri Gielis](#)

Passionate about Oracle and especially Oracle Application Express (APEX). Building applications with love @ APEX R&D.

[View my complete profile](#)

 **ORACLE**
ACE Director

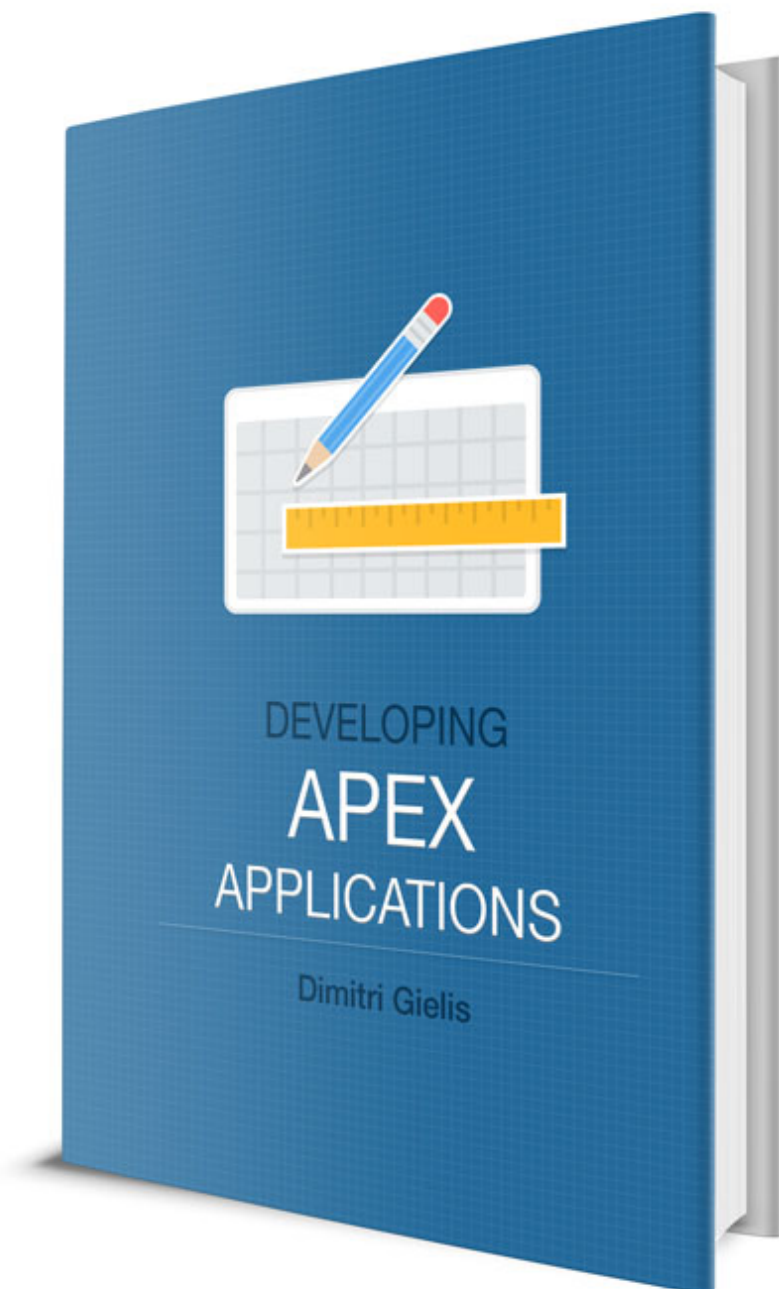
Need APEX consulting, training, development, then visit:



Twitter

 Follow @dgielis





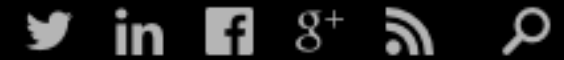
Security **still** an issue?

TRENDING:

Drones in the enterprise · See the Computerworld Digital Magazine · Resources/White Papers

≡ **COMPUTERWORLD**

Popular Now: ▼



Home > Security

OPINION

Evan Schuman: Starbucks caught storing mobile passwords in clear text

In a case of convenience for users trumping security, Starbucks has been storing the passwords for its mobile-payment app, along with geolocation data, in clear text

By Evan Schuman

FOLLOW

Computerworld | Jan 15, 2014 11:09 AM PT

RELATED TOPICS

Editorial note: Starbucks later responded to the security...

MORE LIKE THIS

Evan Schuman: Starbucks sat on its clear-text password problem for months

Evan Schuman: App testing and sins of omission

Starbucks vows to beef up security on its iPhone app

January 16, 2014 Company

Security of Starbucks Mobile App for iOS

Share



Related News

April 17, 2015

Starbucks Partners Around the World Offer Training and Advice to Young Adults

April 10, 2015

Starbucks Named One of the Top 10 Places to Work in Canada

[More related news >](#)

Security of Starbucks Mobile App for iOS

UPDATE (January 16, 2014 09:00 PM P.S.T.): As promised, we have released an updated version of Starbucks Mobile App for iOS which adds extra layers of protection. We encourage customers to download the update as an additional safeguard measure.

Read a letter from Curt Garner, Starbucks chief information officer, regarding customer information and Starbucks Mobile App for iOS

January 16, 2014

Dear Customer,

Your security is incredibly important to us. This week a research report identified theoretical vulnerabilities associated with the Starbucks Mobile App for iOS in the event a customer's iPhone were to be physically stolen and hacked.

Serious security issues in iOS Outlook app Microsoft to access user emails and credentials

February 3, 2015 By [Pierluigi Paganini](#)

HACKING



Java – New vulnerabilities affects million applications

Oracle warned that a dozen of new Java security vulnerabilities could be exploitable...

April 18, 2015 By [Pierluigi Paganini](#)



In-flight Wi-Fi can be an open door for hackers

A report published by the Government Accountability Office (GAO) warns...

April 17, 2015 By [Pierluigi Paganini](#)



How to move YouTube comments from any video to another

An Egyptian Researcher discovered a flaw that allowed him to duplicate/copy...

April 17, 2015 By [Pierluigi Paganini](#)



Wikileaks released documents from Sony hack, the company condemned it

WikiLeaks on Thursday published a searchable database of more than 30,000...

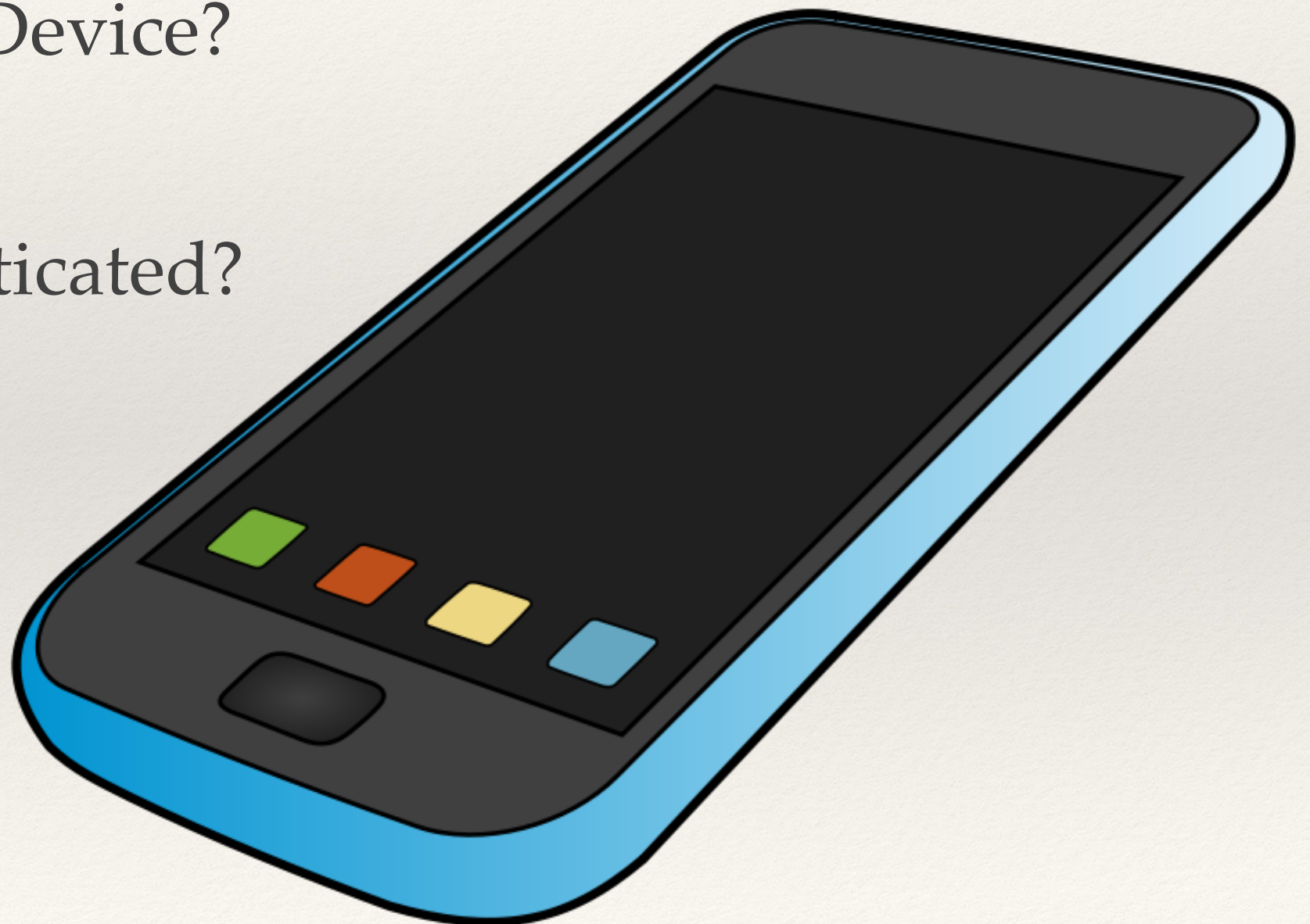
April 17, 2015 By [Pierluigi Paganini](#)

Smartphone stolen?

Connected to public network?

Data saved on Device?

Already authenticated?





Everybody needs a hacker

Now what?



OWASP

The Open Web Application Security Project



OWASP Mobile Security Project

[Home](#)[Top 10 Mobile Risks](#)[Mobile Tools](#)[Mobile Security Testing](#)[Mobile Cheat Sheet](#)[\[edit\]](#)[Secure Mobile Development](#)[Top 10 Mobile Controls](#)[OWASP Mobile Threat Model Project](#)[Mobile Device Management\(MDM\)](#)

LAB medium level projects

OWASP Mobile Security Project

The OWASP Mobile Security Project is a centralized resource intended to give developers and security teams the resources they need to build and maintain secure mobile applications. Through the project, our goal is to classify mobile security risks and provide developmental controls to reduce their impact or likelihood of exploitation.

Our primary focus is at the application layer. While we take into consideration the underlying mobile platform and carrier inherent risks when threat modeling and building controls, we are targeting the areas that the average developer can make a difference. Additionally, we focus not only on the mobile applications deployed to end user devices, but also on the broader server-side infrastructure which the mobile apps communicate with. We focus heavily on the integration between the mobile application, remote authentication services, and cloud platform-specific features.

We have a Google Doc where anyone who wants to be involved with the project can add their thoughts, suggestions, and take ownership of initiatives - [Click here](#). There are various tasks that people have started over the past 6 months with varying levels of quality and completeness.

This project is still a work in progress. We are small group doing this work and could use more help! If you are interested, please contact one of the project leads or feel free to visit [the mailing list](#) as well!

Email List

[Project Email List](#)

Project Leaders

[Mike Zusman](#)[Tony DeLaGrange](#)[Sarath Geethakumar](#)[Tom Eston](#)[Don Williams](#)[Jason Haddix](#)

Contributors

[Zach Lanier](#)[Jim Manico](#)[Ludovic Petit](#)[Swapnil Deshmukh](#)[Beau Woods](#)[Jonathan Carter](#)[David Martin Aaron](#)[Luca De Fulgentis](#)[Milan Singh Thakur](#)

[Home](#)
[About OWASP](#)
[Acknowledgements](#)
[Advertising](#)
[AppSec Events](#)
[Books](#)
[Brand Resources](#)
[Chapters](#)
[Donate to OWASP](#)
[Downloads](#)
[Funding](#)
[Governance](#)
[Initiatives](#)
[Mailing Lists](#)
[Membership](#)
[Merchandise](#)
[News](#)
[Community portal](#)
[Presentations](#)
[Press](#)
[Projects](#)
[Video](#)
[Volunteer](#)

▼ [Reference](#)
[Activities](#)
[Attacks](#)
[Code Snippets](#)
[Controls](#)
[Glossary](#)
[How To...](#)
[Java Project](#)
[.NET Project](#)
[Principles](#)
[Technologies](#)
[Threat Agents](#)
[Vulnerabilities](#)

► [Language](#)

► [Tools](#)

OWASP Mobile Top 10 Risks

M1 – Weak Server
Side Controls

M2 – Insecure
Data Storage

M3 - Insufficient
Transport Layer
Protection

M4 - Unintended
Data Leakage

M5 - Poor
Authorization and
Authentication

M6 - Broken
Cryptography

M7 - Client Side
Injection

M8 - Security
Decisions Via
Untrusted Inputs

M9 - Improper
Session Handling

M10 - Lack of
Binary Protections

A1 – Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2 – Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
A3 – Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4 – Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 – Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6 – Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7 – Missing Function Level Access Control	Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
A8 - Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A9 - Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
A10 – Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Security in APEX environment

Application Express Application Builder User's Guide



Table of Contents

- ▼ Oracle Application Express Application Builder User's Guide
 - ▶ Preface
 - ▶ Changes in This Release
 - ▶ Quick Start
 - ▶ Application Builder Concepts
 - ▶ Utilizing Packaged Applications
 - ▶ Understanding Application Builder
 - ▶ Managing the Application Life Cycle with Team Development
 - ▶ Creating Websheet Applications
 - ▶ Creating Database Applications
 - ▶ Managing Pages in Page Designer
 - ▶ Managing Pages in Component View
 - ▶ Developing Reports
 - ▶ Developing Forms
 - ▶ Managing Database Application Components
 - ▶ Using Themes and Theme Styles
 - ▶ Controlling Page Layout
 - ▶ Managing Database Application Controls

20 Managing Application Security

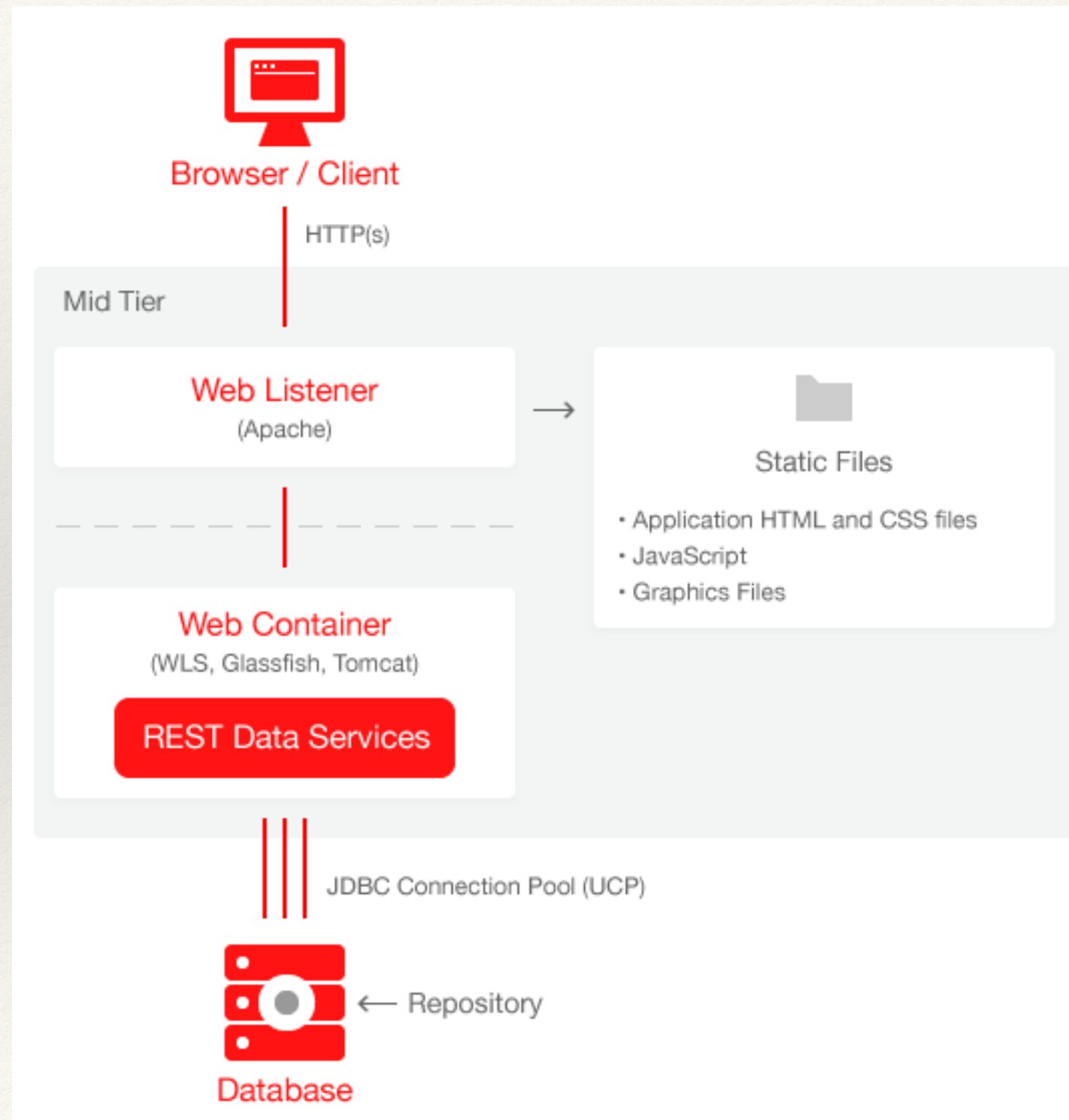
This section describes how to implement best practices for Oracle Application Express application security. While administrators are primarily responsible for ensuring the security of the Oracle Application Express installation, developers are responsible for building secure applications.

- [Understanding Administrator Security Best Practices](#)
- [Understanding Developer Security Best Practices](#)
- [Controlling Access to Applications, Pages, and Page Components](#)
- [Establishing User Identity Through Authentication](#)
- [Providing Security Through Authorization](#)

Architecture

VPN

Firewall(s)

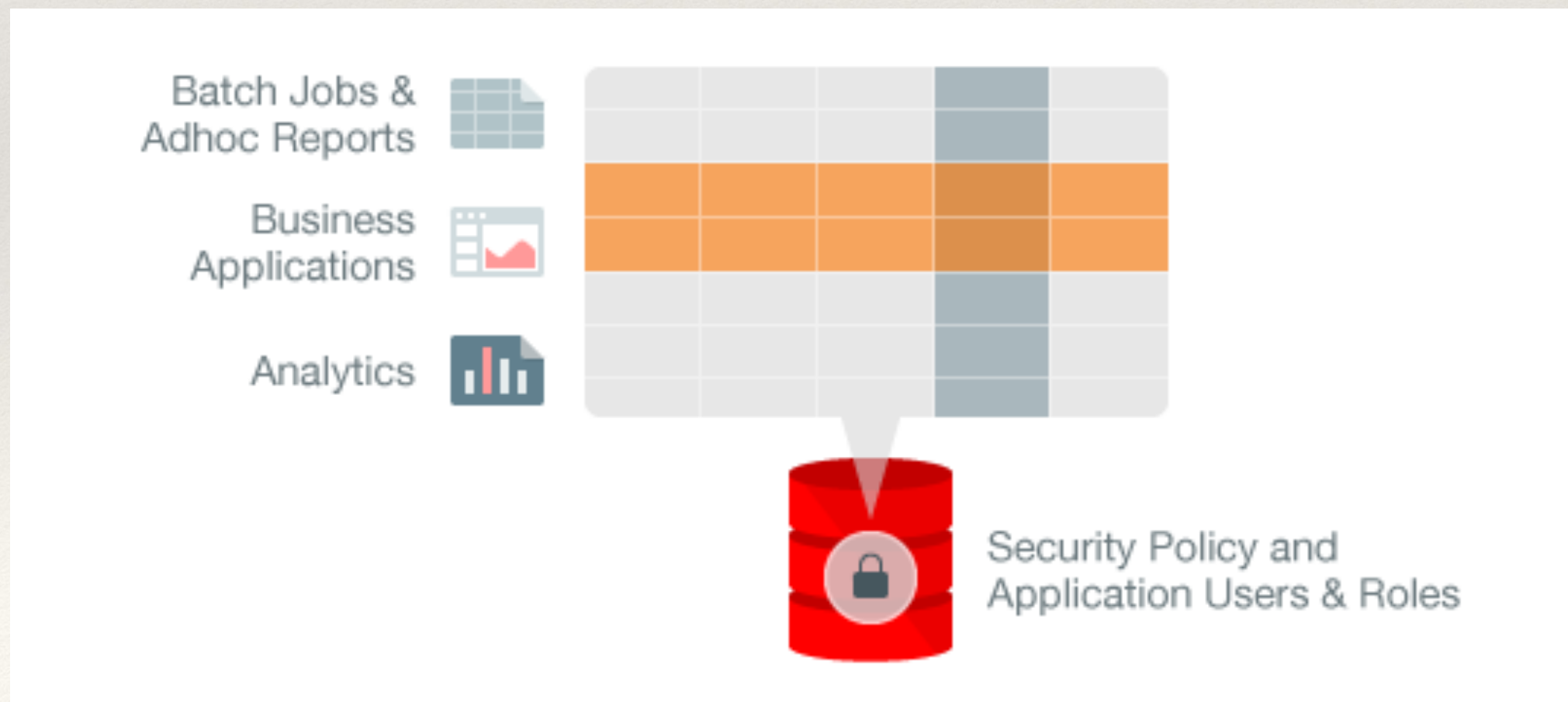


Server Side (global)

- ❖ Architecture (Tunnel, VPN, Firewall, Proxy, ...)
 - ❖ patching (all components)
- ❖ Configure ORDS
 - ❖ Set `security.requestValidationFunction`
- ❖ SSL
 - ❖ Instance settings: Require HTTPS
- ❖ APEX Runtime Environment

Data Protection (Server)

- ❖ Lowest level = in the database
- ❖ Real Application Security (RAS)
 - more secure, scalable, and cost effective than traditional Oracle VPD



Oracle RAS Benefits

- ❖ End-user session propagation to the database
- ❖ Data security based upon application users, role, privileges, and various relationships
- ❖ Audit of end-user activity
- ❖ Simplified administration with declarative security

Oracle RAS & APEX 5.0

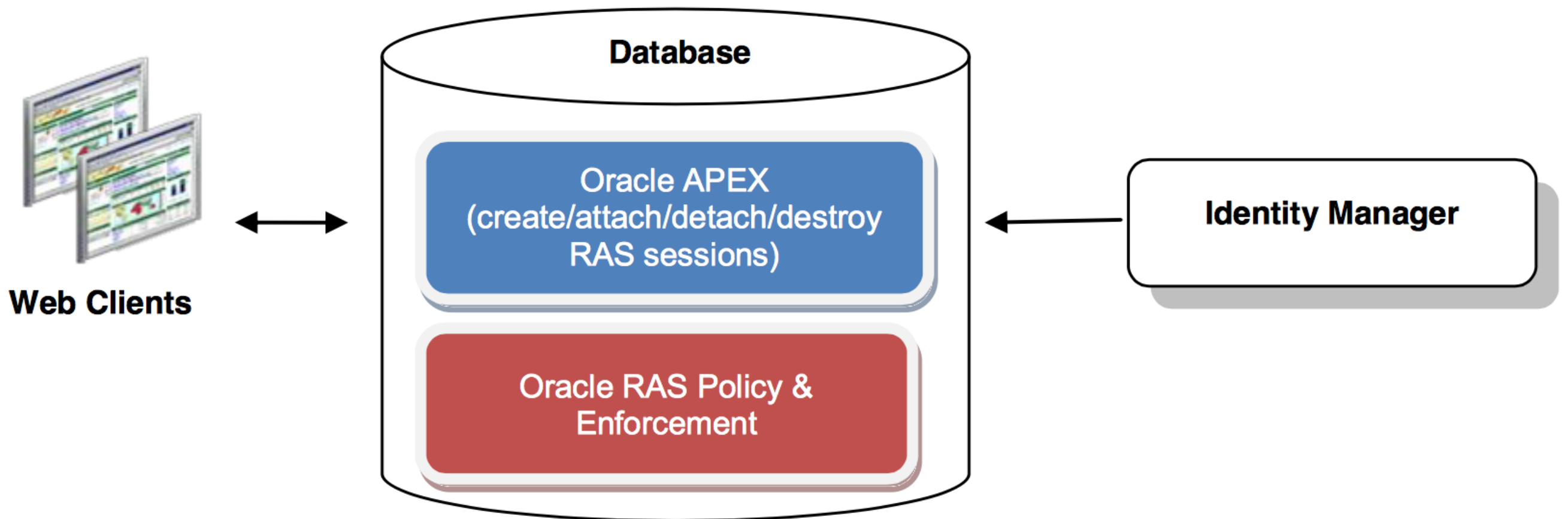


Figure14: APEX Applications with Oracle RAS

Oracle RAS & APEX 5.0

❖ Instance setting

Security Settings

Authorized URLs

Security

RESTful Access

Show All

Security

HTTP Protocol

RESTful Access

Real Application S...

Session Timeout

Workspace Isolation

Region and Web S...

Authentication Co...

Password Policy

Cancel

Apply Changes

Real Application Security

Allow Real Application Security

Yes

⌵

?

Session Timeout

Maximum Session Length in Seconds

36000

⌵

?

Maximum Session Idle Time in Seconds

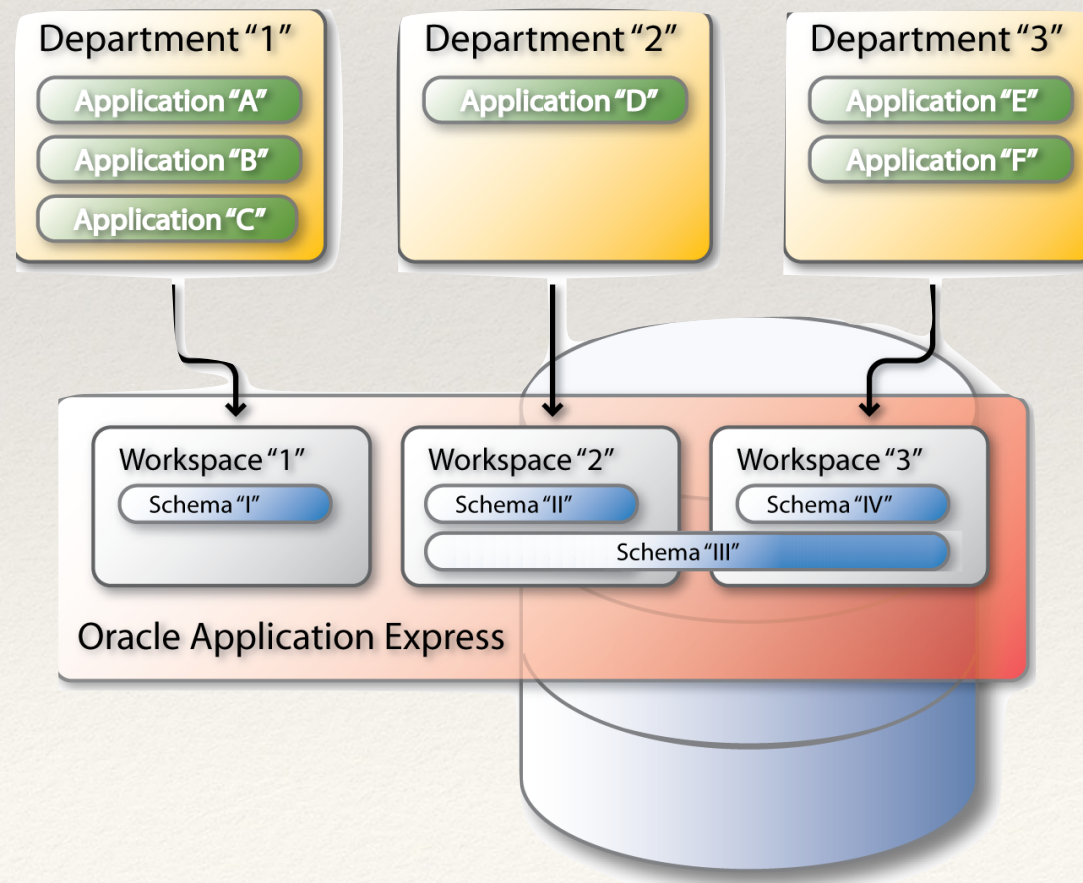
7200

⌵

?

Server Side (APEX)

- ❖ Isolating Workspaces
 - ❖ Allow Hostnames attribute
- ❖ Workspace to database schema assignments



Server Side (APEX)

- ❖ Session Timeout
- ❖ Password policies
- ❖ Disable Rejoin Sessions
- ❖ ...

Instance settings

Security

Cancel

Apply Changes

Security

Configure service level security settings typically used to lock down a production service.

Set Workspace Cookie

Yes

?

Disable Administrator Login

No

?

Disable Workspace Login

No

?

Allow Public File Upload

No

?

Restrict Access by IP Address

?

Instance Proxy

?

Checksum Hash Function

Most Secure

?

Rejoin Sessions

Enabled for Public Sessions

?

Unhandled Errors

Show Error Page

?

Instance settings

HTTP Protocol

Warning: *Require HTTPS* will make Application Express unreachable by the HTTP protocol. Before enabling this setting, ensure that the HTTPS protocol is enabled and configured correctly on your server.

Require HTTPS 

Require Outbound HTTPS 

HTTP Response Headers



RESTful Access

Allow RESTful Access 

Real Application Security

Allow Real Application Security 

Session Timeout

Maximum Session Length in Seconds 

Maximum Session Idle Time in Seconds 

Instance settings

Workspace Isolation

Configure isolation and resource limitation default values for all workspaces. Override these default values at workspace level.

Allow Hostnames



Resource Consumer Group



Maximum Concurrent Workspace Requests



Maximum Concurrent Session Requests



Concurrent Session Requests Kill Timeout



Maximum Size of Files in Workspace



Region and Web Service Excluded Domains

Domain Must Not Contain



Authentication Control

In APEX app

App level settings

Show All

Authentication

Authorization

Session Managem...

Session State Prot...

Browser Security

Database Session

Authentication

Define Authentication Schemes >

Authentication is the process of establishing each user's identify before they can access your application. You may define multiple authentication schemes for your application, however only one scheme can be current. The authentication logic of the current scheme is used when your application is run.

Application **196** ?

Public User ?

Authentication Scheme ?

Authorization

Define Authorization Schemes >

Application authorization schemes control access to all pages within an application. Unauthorized access to the application, regardless of which page is requested, will cause an error page to be displayed.

Authorization Scheme ?

Run on Public Pages ?

App level settings

Session Management

Rejoin Sessions ⓘ

Deep Linking ⓘ

Maximum Session Length in Seconds ⓘ

When Maximum Session Length is not set, the workspace level or instance level value will be used (currently, 36000 seconds).

Session Timeout URL ⓘ

Maximum Session Idle Time in Seconds ⓘ

When Maximum Session Idle Time is not set, the workspace level or instance level value will be used (currently, 7200 seconds).

Session Idle Timeout URL ⓘ

App level settings

Session State Protection

[Expire Bookmarks](#)[Manage Session State Protection >](#)

Enabling Session State Protection can prevent hackers from tampering with URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.

To enable Session State Protection for your application, select **Enabled** from the Session State Protection list. Enabling Session State Protection turns on session state protection controls defined at the page and item level. To configure Session State Protection, click **Manage Session State Protection**.

Session State Protection ?

Allow URLs Created After ?

Bookmark Hash Function ?

Browser Security

Cache ?

Embed in Frames ?

HTML Escaping Mode ?

HTTP Response Headers ?

App level settings

Database Session

* Parsing Schema

WEARABLES



Initialization PL/SQL Code



Cleanup PL/SQL Code



Runtime API Usage

- ☐ Modify This Application
- ☐ Modify Other Applications
- ☐ Modify Workspace Repository



Page level settings

▼ Security	
Authorization Scheme	- Select - ⌵ >
Authentication	Page Requires Authenticati ⌵
Rejoin Sessions	Application Default ⌵
Deep Linking	Application Default ⌵
Page Access Protection	Unrestricted ⌵
Form Auto Complete	On ⌵
Browser Cache	Application Default ⌵

Authentication

- ❖ Username / Password
- ❖ Single Sign-On
- ❖ 3rd party (Facebook/Google/Linkedin/...)
- ❖ Through device? (Touch ID)
- ❖ Plug-ins

Authentication (remember me)

```
--=====
-- create a cookie with the user token for that automatic login process
--=====
--
procedure send_remember_me_cookie (p_username in varchar2)
is
    l_cookie_name varchar2(20) := 'mtabornet';
    l_token        varchar2(100);
    l_person_id    number;
begin
    -- retrieve token
    select person_id, token
        into l_person_id, l_token
        from fbl_person
        where upper(email) = upper(p_username);

    if l_token is null
    then
        l_token := utl_raw.cast_to_raw (dbms_obfuscation_toolkit.md5(input_string => to_char(l_person_id) || dbms_random.string (

        update fbl_person
            set token = l_token
            where upper(email) = upper(p_username);
    end if;

    -- set cookie with token
    owa_cookie.send(name => l_cookie_name, value => l_token, expires => sysdate+365);

exception
    when no_data_found
    then
        -- remove cookie
        owa_cookie.remove(name => l_cookie_name, val => null);
end send_remember_me_cookie;
--
```

Password items

- ❖ do not save session state
 - ❖ or store the value encrypted
- ❖ APEX helps to find password items at risk:
 - ❖ Viewing the Security Profiles Report
 - ❖ Viewing the Password Items Report

Authorization

- ❖ Once in, limit what people can see and do

Session State Protection

https://www.apexrnd.be/ords/f?p=196:3:16560741994095::::P3_ID:43&cs=3QIOD3|7XRK0S

- ❖ Session
- ❖ URL Tempering
- ❖ Default enabled in APEX 5.0

SQL injection

- ❖ Incorrectly filtered user input used in an SQL operation leading to unintended side-effects

SQL injection

```
select *  
  from emp  
 where ename = '&P7_SEARCH1.'
```

SQL injection

Search1

KING KING' or 1=1--

Empno 	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10

1 - 1

Cross-site scripting (XSS)

- ❖ In a XSS attack, a web application is sent a script that activates when it is read by a user's browser. Once activated, these scripts can steal data, even session credentials, and return the information to the attacker.

Many Types of XSS

- ❖ Stored XSS
 - ❖ JavaScript in database
- ❖ Reflected XSS
 - ❖ Embedded JavaScript in URL request
- ❖ Stored XSS in uploaded files
 - ❖ HTML, Text file with .jpg extension, etc.

Escaping substitution strings

- ❖ `apex_escape.html()`
- ❖ Escape special characters attribute: YES

Protecting Regions

- ❖ **#COLUMN!HTML#** - Escapes reserved HTML characters.
- ❖ **#COLUMN!ATTR#** - Escapes reserved characters in a HTML attribute context.
- ❖ **#COLUMN!JS#** - Escapes reserved characters in a JavaScript context.
- ❖ **#COLUMN!RAW#** - Preserves the original item value and does not escape characters.
- ❖ **#COLUMN!STRIPHTML#** - Removes HTML tags from the output and escapes reserved HTML characters.

Data Protection (Client)

- ❖ Data encryption in Session State
- ❖ Encrypt locally stored data (on device)

Other tools

- ❖ Database Vault
- ❖ Audit Vault
- ❖ Database Firewall
- ❖ Label Security
- ❖ Virus Scanners (include in ORDS)
- ❖ ...

Q&A

APEX RD
research & development



www.apexRnD.be



dgielis.blogspot.com



@dgielis



dgielis@apexRnD.be

Consulting, Development, Training

- ❖ Looking for consulting, training and development in Oracle Application Express (APEX)?
- ❖ Contact : www.apexRnD.be
- ❖ Mail : info@apexRnD.be