Store Passwords as Digest

- The passwords are stored as clear text in the **${tomcat_home}/conf/tomcat-users.xml** file
- To protect the passwords, store the passwords in **digested form**
  - Generate the **digested form** of the password
  - Update the generated password in **tomcat-users.xml**
  - Include **<Realm>** element under the **<host>** element of the **server.xml** file to implement digested passwords

Password Stored as Clear Text

tomcat-users.xml

```
<role rolename="Configuser"/>
<role rolename="HRmanager"/>
<role rolename="admininistrator"/>
<user username="sysadmin" password="pass@$123"
roles="manager,admin"/>
<user username="cataloguser" password="qwerty@123"
roles="Configuser"/>
</tomcat-users>
```

Generating Digest Form of the Password

```
> cd ${tomcat_home}/bin
>./digest.sh -a SHA pass@123
> pass@123:aa04ew2afaae76c2a7dc5e25ex18c505e58f12d7
```

Password Stored in Digest Form

tomcat-users.xml

```
<role rolename="Configuser"/>
<role rolename="HRmanager"/>
<role rolename="admininistrator"/>
<user username="sysadmin" password="pass@$123"
roles="manager,admin"/>
<user username="cataloguser"
password="5413ee24723bba2c5a6ba2d0196c78b3ee4628d1"
roles="Configuser"/>
</tomcat-users>
```

Editing server.xml File

server.xml

```
unpackWARs="true" autoDeploy="true" xmlValidation="false"
xmlNamespaceAware="false">
<Realm className="org.apache.catalina.realm.MemoryRealm" digest="SHA" />
</Host>
```

Design Source

Do Not Run Tomcat as Root

- Do not run **Tomcat** as root. Create a user with the least required permissions and run Tomcat with the created user account
- Specify the following privileges to the created user
  - Read/Write permission to **${tomcat_home}** directory and subdirectories
  - Read permissions to **data directories**

Running Tomcat as User: Unix / Linux

- Create Tomcat User
  ```
  sudo adduser tomcatUser
  ```
- Download and unpack Tomcat
  ```
  sudo cp apache-tomcat-8.0.33.tar.gz /opt cd /opt
  sudo tar zxvf apache-tomcat-8.0.33.tar.gz
  ```
- Change ownership to created user
  ```
  sudo chown -R tomcat: tomcatUser /opt/apache-tomcat-8.0.33
  ```
- Run Tomcat as a created user
  ```
  sudo -u tomcat /opt/apache-tomcat-8.0.33/bin/catalina.sh run
  ```

Running Tomcat as User: Windows

- Create unprivileged user account using **Administrative Tools → Local Security Policy → Local Policies → User Rights Assignment → Log on as a service**
- Ensure to set **Apache Tomcat** service to run as the created user

Configure Restricted Datasets

1. While restricting access to datasets, create separate users with **restrictedDatasetUser** roles
2. Users with a **restrictedDatasetUser** role can use **access non-HTTP URL**, which can give rise to **session hijacking**
3. Users with a **restrictedDatasetUser** role should not be assigned roles with privileges to access the security features of Tomcat

Session Handling using App Mode in Tomcat

Session Timeout

- Session timeout for every web application must be set to **minimum required duration**
- To set the timeout edit the **CATALINA_HOME/conf/web.xml** file with following code

web.xml

```
585⊖    <session-config>
586       <session-timeout>20</session-timeout>
587     </session-config>
```

HttpOnly Flag

- This flag can be activated with the following configuration option that can be set in the CATALINA_HOME/conf/context.xml
- If HttpOnly flag is set to true then it may break the application functionality, if access to the session cookie via javascript is necessary

Syntax:

context.xml

```
37
38 <Context useHttpOnly="true">
39
```

CSRF Prevention Filter

- It helps prevent **Cross Site Request Forgery**
- CSRF filter generates a nonce and stores it in a session. It encodes the URLs with the generated nonce. On receiving another request for the URL the received nonce is matched with the sessions nonce. If the two do not match the request is considered forged.
- CSRF Filter can be configured in file CATALINA_BASE/conf/web.xml at Globally . The global setting can overwrite by writing in WEB-INF/web.xml

web.xml | context.xml

```
548⊖
549⊖    <filter-mapping>
550       <filter-name>httpHeaderSecurity</filter-name>
551       <url-pattern>/*</url-pattern>
552       <dispatcher>REQUEST</dispatcher>
553     </filter-mapping>
```

Session Handling using App Mode in Tomcat (Cont'd)

Disable Session URL rewriting

- This flag is used to disable the session id displayed in the URL with the following configuration
- CATALINA_HOME/conf/ web.xml with following code:

```
web.xml | context.xml
585  <session-config>
586      <session-timeout>20</session-timeout>
587      <tracking-mode>COOKIE</tracking-mode>
588  </session-config>
```

Secure Cookie Flag

- This flag will force the transmission of a cookie only by SSL, with the following configuration
- CATALINA_HOME/conf/ web.xml with following code

```
web.xml | context.xml
585  <session-config>
586      <cookie-config>
587      <secure>true</secure>
588      </cookie-config>
589  </session-config>
```

Replace the Default Session Name

- The default session name is JSESSIONID. The following configuration could be used to change the default name
- **CATALINA_HOME/conf/ web.xml** with following code:

```
web.xml | context.xml
585  <session-config>
586      <cookie-config>
587      <name>MySESSIONID</name>
588      </cookie-config>
589  </session-config>
```

Change the Session Id length

- The default session id length is 128bit / 16bytes. The following configuration could be used to change the default value
- **CATALINA_HOME/conf/context.xml** with following code

```
web.xml | context.xml
19
20  <Context >
21    <Manager pathname="" sessionLength="32" />
22
```

Role Based Security

Tomcat supports **role based authorization**

To implement role based authorization

1. Create users and roles in /conf/tomcat-user.xml
2. Define Security role in web.xml

Example : Setting Roles and Users in Tomcat-user.xml

```
tomcat-users.xml
67  <role rolename="tomcatUser"/>
68  <role rolename="roleA"/>
69  <role rolename="userMain"/>
    <user username="tomcatUser" password="tomcatuserpwd" roles="tomcatUser"/>
    <user username="roleA" password="tomcatuserpwd" roles="roleA"/>
    <user username="Mixed" password="tomcatuserpwd" roles="tomcatUser,roleA"/>
    <user username="Tom" password="tompwd" roles="userMain"/>
74 </tomcat-users>
Design | Source
```

Example : Defining Security Role in web.xml

```
web.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2    <web-app>
3    <welcome-file-list>
4    <welcome-file>index.html</welcome-file>
5    </welcome-file-list>
6    <security-role>
7  <description>The role use Manager Application </description>
8  <role-name>userMain</role-name>
9  </security-role>
10 <security-constraint>
11 <web-resource-collection>
   <url-pattern> /protected/* </url-pattern>
   </web-resource-collection>
   <auth-constraint>
   <role-name>userMain </role-name>
   </auth-constraint>
   </security-constraint>
   </web-app>
```

Securing Tomcat at Network Level

Prevent The Connectors:

- Prevent the application from listening on all interfaces/IP addresses available on the server system. Instead, the IP address must be specified
- Edit CATALINA_HOME/conf/server.xml file to restrict the listening interfaces
- Review every connecter and specify the correct IP addresses

Syntax:

```
<connector PORT="TCP_PORT"
address="Listen_IP_Address"…….
```

Tomcat Port Connection:

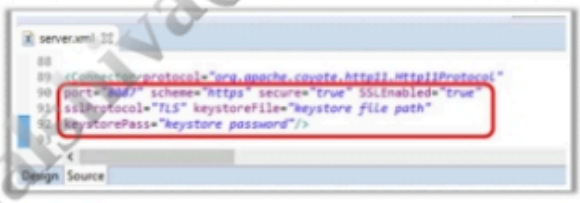
- Tomcat has many ports for network connection. To establish a connection a port works as an identifier. These ports need to be configured correctly
- For TCP default ports are 8080 and 8443. These ports are very important and need to be configured correctly
- By opening CATALINA_HOME/conf/server.xml file we can see every connector configuration for the correct/desired port assignment. Remove unnecessary or unused ports or connections

Securing Tomcat at Network Level (Cont'd)

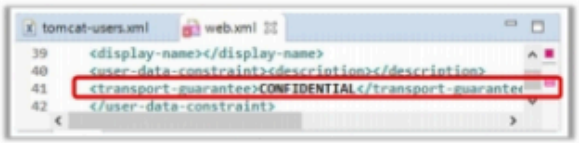
Trustworthy connection using encryption of network connection:

- **SSL configuration:** To secure an application, SSL configuration must be applied. SSL provides encryption and decryption for communication between two parties over the network. For this, the public key encryption method is used. The following steps are used for Tomcat configuration to use SSL
- **Two different implementations of SSL:**
  - **JSSE Implementation** → If Tomcat native library is installed
  - APR Implementation uses OpenSSL engine

- Edit the file CATALINA_HOME/conf/server.xml with following code



- Add the following in CATALINA_HOME/webapps/manager/WEB_INF/web.xml under <security-constraint> tags



Securing Tomcat at Network Level (Cont'd)

- **Use valve to filter by IP/hostname that allows only a subset of machines to connect**
  - Any one of the following codes can be added in the Context tag in CATALINA_HOME/conf/Catalina/localhost/context.xml

Example 1

```
<!-- allow only LAN IPs to connect to the manager webapp -->
<!-- contrary to the current Tomcat 7.0 documentation the value for allow is not
a regular expression -->
<!-- future versions may have to be specified as 192\.168\.1\.* -->
```

context.xml

```
41
42        <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="192.168.1.*" />
43 </Context>
```

Design | Source

Example 2

```
connect to the manager webapp -->
<!-- contrary to the current Tomcat 7.0 documation the value for allow is not a
regular expression -->
<!-- future versions may have to be specified as *\.localdomain\.com -->
```

context.xml

```
41
42        <Valve className="org.apache.catalina.valves.RemoteAddrValve"
43        allow="localdomain.com" />
```

Design | Source

Securing Tomcat at Network Level (Cont'd)

Mutual Authentication

- Certificate-based mutual authentication provides the mechanism to authenticate two parties by providing and verifying the certificate to each other. Web browsers authenticate themselves to a web server and the server also authenticates itself to the client by verifying the digital certificate

Securing Tomcat at Network Level (Cont'd)

- **Create Client Side KeyStore**
  command: `$JAVA_HOME/bin/keytool -genkey -alias client -keyalg RSA -keystore client.jks`
- **Export Client Side Certificate File**
  command: `$JAVA_HOME/bin/keytool -export -keystore client.jks -alias client -file client.cer`
- **Import client.cer to server trustStore**
  command: `$JAVA_HOME/bin/keytool -importcert -file client.cer -keystore truststore.jks -alias client`
- **Convert the client.jks to PKCS12 format**
  command: `$JAVA_HOME/bin/keytool -importkeystore -srckeystore client.jks -destkeystore client.pfx -srcstoretype JKS -deststoretype pkcs12 -deststorepass "XXX" -alias client -destalias client`
- **Import the client.pfx file to client side browser**
- **Configure server.xml**
- **Edit the file CATALINA_HOME/conf/server.xml with following code**

server.xml

```
<Connector protocol="org.apache.coyote.http11.Http11Protocol"
port="443" scheme="https" secure="true" SSLEnabled="true"
sslProtocol="TLSv1.2"
sslEnabledProtocols="TLSv1.1, TLSv1.2" clientAuth="true"
 keystoreFile="keystore file path" keystorePass="keystore password"
truststoreFile="truststore file path" truststorePass="truststore password"
<
```

Design | Source

Java Runtime Security Configurations

Java Security Manager

- Java security manager can be configured in file CATALINA_HOME/conf/Catalina.policy
- Once the file is configured Tomcat will start with SecurityManager in place of security option

Java Package Access

- Tomcat has a feature to restrict package access
- A user trying to access a package without access permission gets access exception
- To modify package.access alter the package.access list in CATALINA_BASE/conf/Catalina.properties

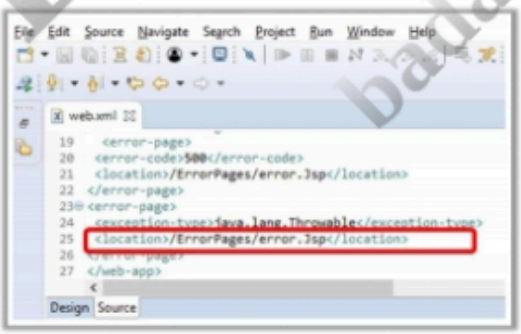
Tomcat General Security Setting

- Do not make unnecessary changes in the default settings of Tomcat configurations
- Remove **unwanted, default, servlet examples**, etc. from Tomcat **webapp directory**

Tomcat General Security Settings (Cont'd)

Custom Error Page

- It is the best practice to include a custom error page in the web application. The following code must be added in **CATALINA_HOME/webapps/webapp/WEB-INF/web.xml**



Automatic Deployment

- Automatic deployment must be **disabled** to prevent malicious applications from deploying
- This can be done by editing the file **CATALINA_HOME/conf/server.xml**
- Change the following line in server.xml file

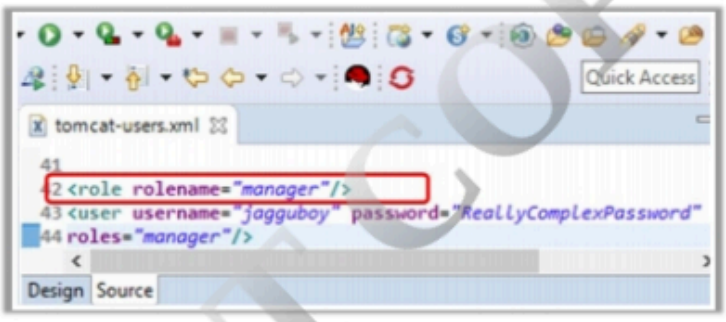
```
<.. ……..autoDeploy="false">

<.. ……..DeployOnStartup="false">
```

Tomcat General Security Settings (Cont'd)

Create Webapp

- A manager Webapp can be created by adding a new role and user into the CATALINA_HOME/conf/tomcat-users.xml file as follows:



Tomcat General Security Settings (Cont'd)

Rename the Manager Webapp as follows

- In the example below the new name is "sample"

```
1: Move CATALINA_HOME/conf/Catalina/localhost/manager.xml to
2: CATALINA_HOME/conf/Catalina/localhost/sample.xml
```

```
1: Update the docBase attribute within CATALINA_HOME/conf/Catalina/localhost/sample.xml to
2: ${catalina.home}/server/webapps/foobar
```

```
1: Move CATALINA_HOME/server/webapps/manager to CATALINA_HOME/server/webapps/sample
```

Run Tomcat with a Security Manager that controls the access to server resources

- It is a good practice to start Tomcat with "`-security`" parameter

Verify Trace Element Setting in server.xml

The tracing process is accountable for all activities accomplished by a web page on a web server and is enabled mainly to troubleshoot issues during the development of an application

However, leaving trace enabled after deployment of the application may expose critical information and allow attackers to alter and manipulate the application

Set **<Connector URIEncoding="UTF ..... allowTrace="false" />**. Disabling trace element prevent the attackers from gaining information from the trace

Vulnerable Configuration

server.xml

```
97  <Connector URIEncoding="UTF-8"
98  acceptCount="100"
99  sslProtocol="TLS"
100 allowTrace="true">
101 </Connector>
```

Design | Source

Secure Configuration

server.xml

```
97  <Connector URIEncoding="UTF-8"
98  acceptCount="100"
99  sslProtocol="TLS"
100 allowTrace="false">
101 </Connector>
```

Design | Source

Verify CustomError Settings in web.xml

Attackers can use this information to make an educated guess and craft attack techniques to compromise the security of the application

The attackers may use the exception details to **perform attacks** or **manipulation of information** in the application

The user will be redirected to any generic web page in the event of any error

The following code from web.xml file shows a secure way of setting custom error:

Secure Configuration

*web.xml

```
18
19
20  <error-page>
21      <error-code>404</error-code>
22      <location>**/error.html**</location>
23  </error-page>
24  <error-page>   <error-code>404</error-code>
25      <location>**/error.html**</location>
26  </error-page>
27  <error-page>
28      <exception-type>java.lang.Throwable</exception-type>
29      <location>**/error.html**</location>
30  </error-page>
```

Design | Source

Verify maxPostSize Setting

1. The **maxPostSize** attribute of httpRuntime element, restricts attackers from uploading files with large sizes to the server
2. Allowing users to upload files with large sizes may put the application under risk of **denial-of-service** attacks
3. Set **<connector... maxPostSize="6291456" />** the limit can be disabled by setting this attribute to less than zero, like **-1**

Vulnerable Configuration

server.xml

```
103
104 <Connector port="8080" protocol="HTTP/1.1"
105 connectionTimeout="20000"
106 redirectPort="8443"
107 maxPostSize="0" />
108
```

Secure Configuration

server.xml

```
102
103
104 <Connector port="8080" protocol="HTTP/1.1"
105 connectionTimeout="20000"
106 redirectPort="8443"
107 maxPostSize="6291456" />
```

Design | Source

Tomcat Security Checklist

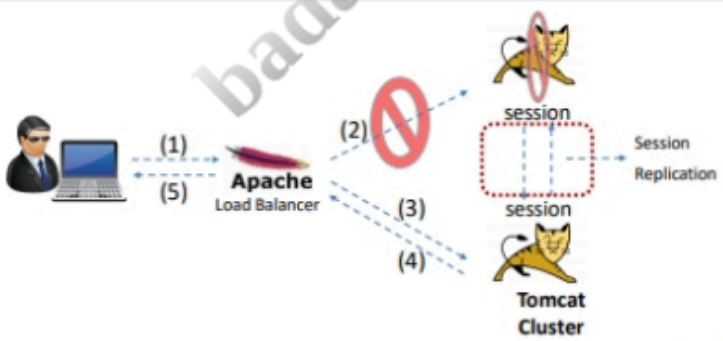
- The following default settings must not be changed in server.xml
  - **allowTrace="false"**
  - **Privileges="false"**
  - **crossContext="false"**
  - **allowLinking="false"**
- Make sure to change **DefaultServlet configuration** with **read only set to false**
- Make sure the following are set to false in Startup script
  - Dorg.apache.catalina.connector.RECYCLE_FACADES=false
  - Dorg.apache.catalina.connector. CoyoteAdapter.ALLOW_BACKSLASH=false
  - Dorg.apache.tomcat.util.buf. UDecoder.ALLOW_ENCODED_SLASH=false
  - Dorg.apache.coyote.USE_CUSTOM_STATUS_MSG_IN_HEADER=false

Checklist for Security Configuration in server.xml File in Apache Tomcat

- **Port Disabling**
  - Avoid malicious attacks by setting Port on the server element to (-1) disable
- **Security Lifecycle Listener**
  - Linux :By using Security Lifecycle Listener we can prevent Tomcat from running as root user
- **Specifying Interfaces for connectors**
  - We can specify address attribute of the connector tag or element thereby decrease the ways of attacks
- **Disable allowTrace Feature**
  - The AllowTrace feature enables the debugging and tracing feature which becomes a source of sensitive information discloser
- **Disable sslEnableProtocols**
  - By exploiting this (**sslEnableProtocols**) vulnerability a user can gain access to sensitive information like password, cookies etc.
  - It is recommended that **sslEnableProtocols** is **disabled**
- **Disabling Directory Listing**
  - When the directory listing feature is enabled the application displays the entire directory structure of the web application
  - Displaying directories containing a large no of files results in denial-of-service attack. To disable directory listing set DefaultServlet listings to false
- **Disable Automated Deployment(if not in use)**
  - Ensure that the host attributes (autoDeploy, deployOnStartup, and deployXML) are disabled if not in use
- **Avoid running Tomcat under root user**
  - Do not run Tomcat under a user account with administrative privileges
  - Ensure to create a user account with minimum required OS permissions for running the Tomcat process

Tomcat High Availability

- The system requires to be in **zero downtime** effectively while failure or disaster occurs
- An alternative system is the way to achieve immediate availability
- With **Session Replication**, clients don't have to reauthenticate while one of the server instance is down



Tomcat High Availability (Cont'd)

- **Tomcat Clustering Configuration:**
  - Edit the file **CATALINA_HOME/conf/server.xml,** add the following code in the <**engine**> section

server.xml

```
116    -->
117    <Engine defaultHost="localhost" name="Catalina">
118    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster" channelSendO
119    <Manager className="org.apache.catalina.ha.session.DeltaManager" expireSessio
120     notifyListenersOnReplication="true"/>
121    <Channel className="org.apache.catalina.tribes.group.GroupChannel">
122    <Membership className="org.apache.catalina.tribes.membership.McastService"
123    address="228.0.0.4"    port="45564"
124              frequency="500"  dropTime="3000"/>
125              <Receiver className="org.apache.catalina.tribes.transport.nio.Nio
126              address="auto" port="4000" autoBind="100"
127              selectorTimeout="5000" maxThreads="6"/>
128              <Sender className="org.apache.catalina.tribes.transport.Replicati
129              <Transport className="org.apache.catalina.tribes.transport.nio.Po
130              </Sender>
131              <Interceptor className="org.apache.catalina.tribes.group.intercep
132              <Interceptor className="org.apache.catalina.tribes.group.intercep
133              </Channel>
134              <Valve className="org.apache.catalina.ha.tcp.ReplicationValve" fi
135              <Valve className="org.apache.catalina.ha.session.JvmRouteBinderVa
136              <Deployer className="org.apache.catalina.ha.deploy.FarmWarDeploye
137              tempDir="/tmp/war-temp/"deployDir="/tmp/war-deploy/"
138              watchDir="/tmp/war-listen/"
139              watchEnabled="false"/>
140              <ClusterListener className="org.apache.catalina.ha.session.Cluste
141              </Cluster>
```


Tomcat High Availability (Cont'd)

- **Tomcat Clustering Configuration:**
  - Edit the file **APPLICATION/WEB_INF/web.xml** , add the code **<distributable />**

File Edit Source Navigate Search Project Run Window Help

Quick Access

web.xml

```
1  <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
2           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3           xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
4           http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5           version="3.1" id="data-fabric-web-tool" metadata-complete="true">
6  <display-name>Welcome</display-name>
7  <description>Welcome</description>
8  <distributable/>
9  </web-app>
```

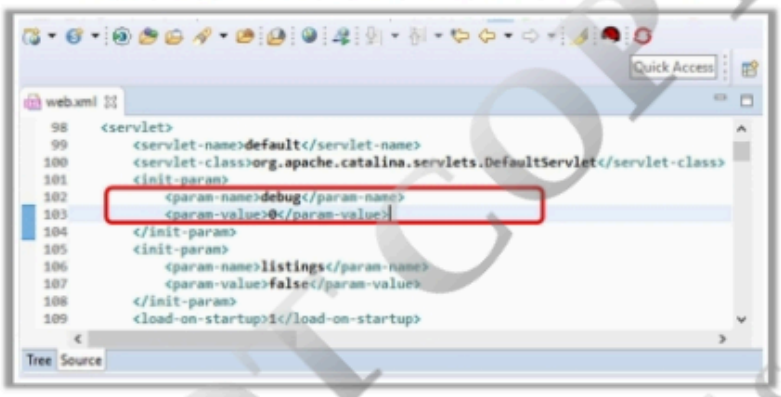
Design Source

- **Restart Tomcat Services**

Best Practices for Securing Tomcat

- Delete everything from the path **CATALINA_HOME/webapps**
- Delete everything from **CATALINA_HOME/server/webapps**
- Delete **CATALINA_HOME/conf/Catalina/localhost/host-manager.xml**
- Ensure that the default web servlet configuration does not serve the index pages in the absence of a welcome page

The following code should be implemented in CATALINA/conf/web.xml:

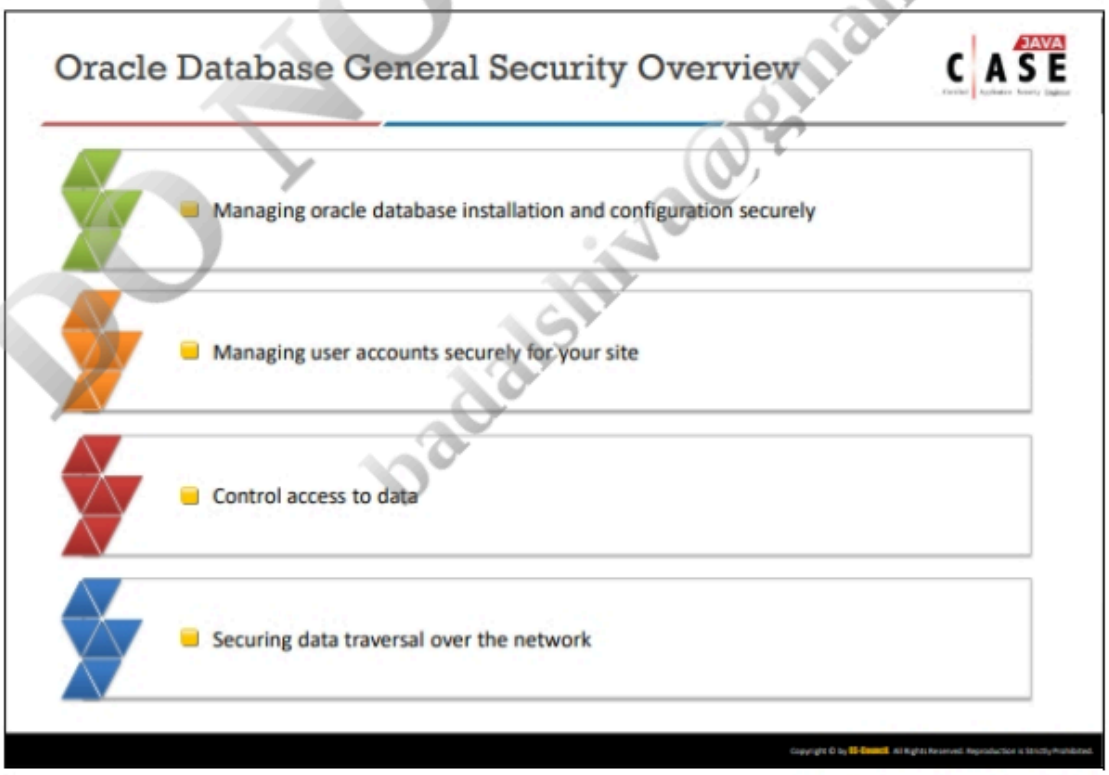


Best Practices for Securing Tomcat (Cont'd)

- Replace the version string of HTTP error messages with **CATALINA_HOME/server/lib/catalina.jar** and an updated **ServerInfo.properties** file
- Replace the default error page code in **CATALINA_HOME/conf/web.xml** with the following code

- Rename **CATALINA_HOME/conf/server.xml** to **CATALINA_HOME/conf/server-original.xml**
- Rename **CATALINA_HOME/conf/server-minimal.xml** to **CATALINA_HOME/conf/server.xml**
- Replace the server version string from HTTP headers in server responses, by adding the server keyword in your Connectors in **CATALINA_HOME/conf/server.xml**

- Clear all text passwords from **CATALINA_HOME/conf/server.xml**



Methods of Authentication in Oracle

- To verify users' identity Oracle provides these authentication methods. Users can use a combination of these methods:

Operating System	Authentication by **Operating System**
Network and LDAP Directories	Authentication by **Network**
Database	Authentication by **Oracle Database**
Multitier System	**Multitier** Authentication and Authorization
Secure Socket Layer Usage	Authentication of **Database Administrators**
Database Administrators	Authentication of **Database Administrators**

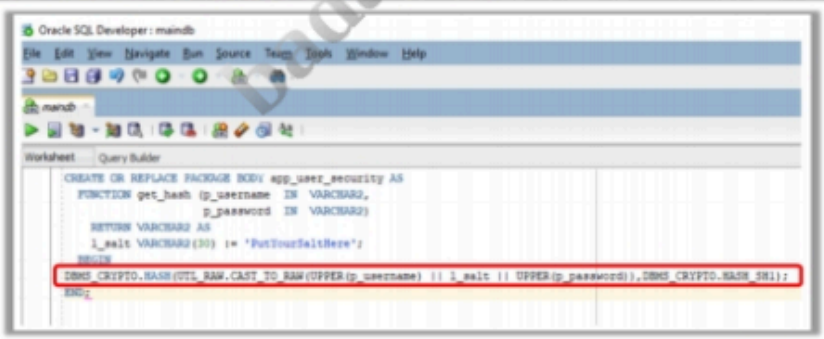
Authentication by Oracle Database

- To implement Oracle database authentication, DBA must create users and associate them with a password to establish a connection
- Set the **SQLNET.ALLOWED_LOGIN_VERSION** parameter in the server **sqlnet.ora** file to identify the authentication protocols allowed by the client or database

Oracle Database Authentication Supports:

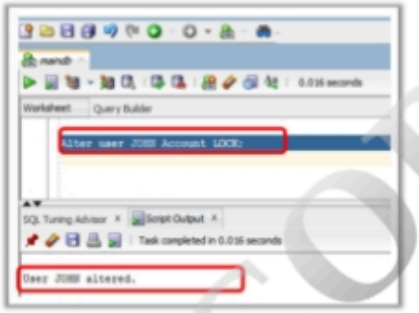
Password Encryption

- **DBMS_CRYPTO** Package is used for encryption and hashing



Authentication by Oracle Database (Cont'd)

- **Account Locking**
  - DBA can Lock accounts manually for a particular user command to lock a specific user



- **Password Lifetime and Expiration**
- **Password History**
- **Password Complexity Verification**

Oracle Security Features

Case Sensitive Passwords

- While creating Database via DBCA, a user is prompted to upgrade to **"New Security Standards"**. If chosen to do so, the created passwords will be case sensitive
- When you try to login, considering passwords are not case sensitive, an error will be prompted
- **For example,**

  SQL> conn scott/TIGER

  ERROR:

  ORA-01017: invalid username/password; logon denied

  Warning: You are no longer connected to ORACLE
- To temporarily disable the case sensitive

  set **sec_case_sensitive_logon = false**

  Example: SQL> alter system set sec_case_sensitive_logon = false;

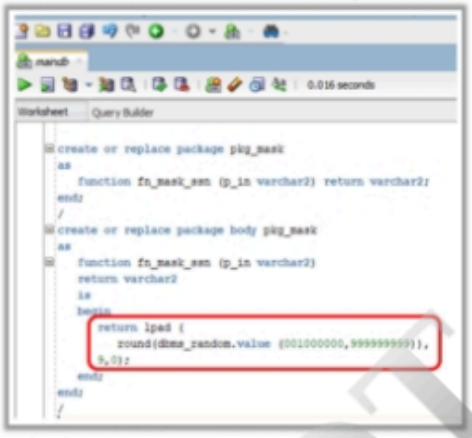
  System altered

Oracle Security Features (Cont'd)

Data Masking

- Converting sensitive data to some meaningless text in the non-production database, is a process known as data masking
- **Remap Function:** At the time of import process data will change

Following is a Function which is used for Remap Data



```
create or replace package pkg_mask
as
   function fn_mask_ssn (p_in varchar2) return varchar2;
end;
/
create or replace package body pkg_mask
as
   function fn_mask_ssn (p_in varchar2)
   return varchar2
   is
   begin
      return lpad (
         round(dbms_random.value (001000000,999999999)),
      9,0);
   end;
end;
/
```

```
    ACCNO ACCNAME                        ACCSSN_No
 ---------- ------------------------------
        1 Kelvin                        123456789
        2 Robert                        234567890

Here is table Accounts user wants to mask coloumn
ACC_SSN. A method Data Dump.At the time of
exporting have to use
Implement remap_data  parameter to mask the data.

$ expdp scott/tiger tables=scott.accountmaster
dumpfile=
   accounts.dmp
directory=tmp_dirremap_data=accounts.acc_ssn:pkg_m
ask.fn_mask_ssn
/
```

Default Database Installation and Configuration Security

- Oracle database contains **built-in protections** for managing different sections of DB. Some settings are while installing the database. Following some of the settings while installation will help protect the database and database dependent applications

Default Security Setting for New Database Creation

- Default **Auditing** setting enables investigating suspicious activity
- Creating **Strong Passwords**, using a combination of upper case, lower case and special characters and numbers in the password, password length should be between 12 and 30 characters
- Remove the **CREATE EXTERNAL JOB privilege** from the **PUBLIC** role
- While installation set **Security-related Initialization** and **Profile Parameter** settings

Default Security Settings for Initialization and Profile Parameters

Setting	10g Default	11g Default
AUDIT_TRAIL	NONE	DB
07_DICTIONARY_ACCESSIBILITY	FALSE	FALSE
PASSWORD_GRACE_TIME	UNLIMITED	7
PASSWORD_LOCK_TIME	UNLIMITED	1
FAILED_LOGIN_ATTEMPTS	10	10
PASSWORD_LIFE_TIME	UNLIMITED	180
PASSWORD_REUSE_MAX	UNLIMITED	UNLIMITED
PASSWORD_REUSE_TIME	UNLIMITED	UNLIMITED
REMOTE_OS_ROLES	FALSE	FALSE

Managing User Accounts Securely for the Site

Securing Oracle Database User Account

- On installation, Oracle provides some predefined accounts. The administrator should secure these accounts by the following methods
  - Safeguarding predefined **Database User Accounts**, setting secure password protection
  - Managing **Database Accounts** such as expired and or database account locks for users
  - Password management

- Oracle Database creates a set of predefined accounts at the time of the installation process. The following are some categories
  - Administrative accounts
  - Non-administrative accounts
  - Sample schema user accounts

Securing User Accounts

Lock and Expire the Default Created Accounts

- Make sure to lock and expire all pre-defined user accounts created while installation. The database configuration assistant by default locks and expires the created accounts
- Use **DBA_*, DBA_ROLES, DBA_SYS_PRIVS, DBA_ROLE_PRIVS, DBA_TAB_PRIVS, DBA_AUDIT_TRAIL** (if standard auditing is enabled), **DBA_FGA_AUDIT_TRAIL** (if fine-grained auditing is enabled) data dictionary views to get user database access details

- Revoke SYS.USER_HISTORY$ table access from all users except SYS and DBA accounts
- Revoke RESOURCE and CONNECT roles from typical application accounts
- Restrict proxy account privileges to CREATE SESSION only
- Maintain roles specific to each database installation
- Ensure that the default user passwords are modified
- Build a complex, long password which is easy to remember
- Make sure to enforce password policy
- Encrypt the column storing the user passwords
- Revoke Execute from **PUBLIC** for **UTL_SMTP, UTL_TCP, UTL_HTTP, UTL_FILE, DBMS_OBFUSCATION_TOOLKIT** packages

Example : Revoke on utl_file

```
SQL> REVOKE execute ON utl_file FROM PUBLIC;
```

Securing User Accounts (Cont'd)

- **Implement least privilege principle:** Grant minimum required privileges needed to execute the jobs
  - Restrict SYSTEM and OBJECT privileges granted to database users
  - Restrict users having SYS-privileged connections to the database
  - Restrict users having privileges, like DROP ANY TABLE
  - Restrict users having privileges to create, modify, or drop database objects

  Example : SQL Query to get all users with the DBA role

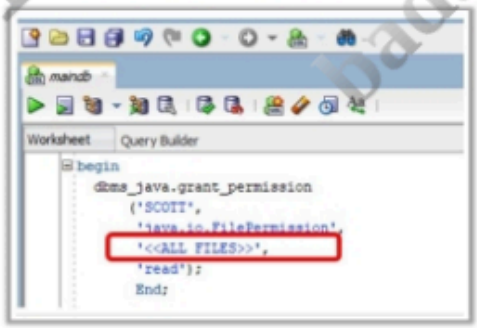
```
SELECT grantee FROM dba_role_privs WHERE granted_role = 'DBA';
```

- **Restrict giving CREATE ANY EDITION and DROP ANY EDITION privileges:** CREATE ANY EDITION and DROP ANY EDITION privileges should only be assigned to users who perform upgrades
- **Restrict security related privileges:** Privileges like CREATE ANY JOB, BECOME USER, EXP_FULL_DATABASE, and IMP_FULL_DATABASE should be restricted to a few users
- **Restrict non-administrative users from accessing SYS schema owners objects:** To protect data integrity only administrative users should be allowed to alter table rows or schema objects present in SYS schema, because doing so can compromise data integrity

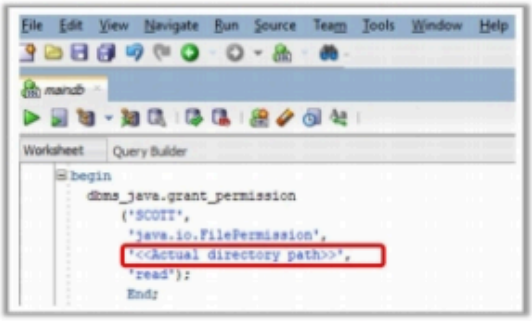
Securing User Accounts (Cont'd)

- Restrict giving **EXECUTE** privileges on the DBMS_RANDOM PL/SQL package
- Give limited **users permissions** on run-time facilities

Vulnerable Runtime Call



Secure Runtime Call-Specifying a Directory Path



Password Management

Password Locking

Parameter	Description
FAILED_LOGIN_ATTEMPTS	Number of failed login attempts before a user account is locked
PASSWORD_LOCK_TIME	Duration in days for locking the account once the failed login attempts exceed

Password Expiration and Aging

Parameter	Description
PASSWORD_LIFE_TIME	No. of days a password is valid before expiring
PASSWORD_GRACE_TIME	No of days after first successful login after the password has expired

Password History

Parameter	Description
PASSWORD_REUSE_TIME	Duration in days for password reuse
PASSWORD_REUSE_MAX	Number of password changes before re-using the current password

Password Verification

Parameter	Description
PASSWORD_VERIFY_FUNCTION	PL/SQL function for checking the complexity of password before assigning the password (Should be owned by `SYS User`)

Lock all Expired Accounts

- DBCA by default **expires and locks** all accounts other than the following Accounts
  - SYS
  - SYSTEM
  - SYSMAN
  - DBSNMP
- Ensure that you lock and expire all Manually created accounts which are not in use

Assign Users to Password Profile

- **Assign a password profile** to a user by editing the user

Disable Remote Operating System Authentication

- Since while implementing **Remote Authentication** the database user is authenticated externally on remote system, make sure to disable remote authentication unless you trust all clients using remote authentication
- To disable remote **Operating System Authentication**, ensure that the **REMOTE_OS_AUTHENT** instance initialization parameter is **FALSE**
- Remote Operating System Authentication is disabled by default

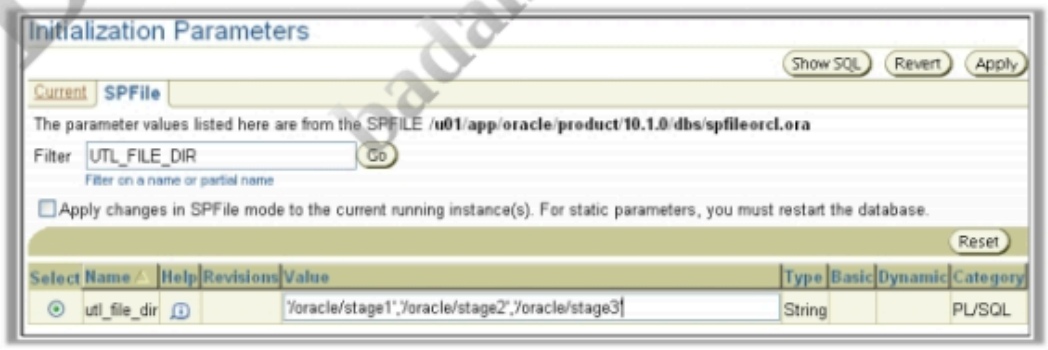
```
REMOTE_OS_AUTHENT = FALSE
```

Securing Data

- **Enable Data Dictionary Protection**
  - To prevent users having ANY TABLE system privileges from modifying the content present in the data dictionary, enable **data dictionary protection**
  - Enabling data dictionary protection enables SYS users to login as **SYSDBA** only and no other user
  - Data dictionary protection is enabled by default
  - To enable data dictionary protection ensure that O7_DICTIONARY_ACCESSIBILITY is FALSE
  - Set **O7_DICTIONARY_ACCESSIBILITY = FALSE** in initsid.ora control file or set O7_DICTIONARY_ACCESSIBILITY = FALSE in a server parameter file
- **Limit operating system access**
  - Make sure to restrict user access to data files, log files, trace files, external tables, BFILE data types
- **Encrypt sensitive data along with backup media containing database files**

Restrict Access to Operating System Directories

- Configure the **UTL_FILE_DIR** parameter to :
  - List the directories available for PL/SQL file I/O
  - List database server directories to which the user has read and write permissions



Securing Database Installation and Configuration

- To **prevent users from overwriting** installation files, installation should be done offline. If installation is done on the network make sure to disconnect logged in users before installation
- To **reduce the attack surface**, install only required components and remove components which will not be used in the production server
- Use the latest version and patches. Outdated and unpatched software increases security risks. Browse the Oracle website for the latest patches and versions
- Make sure to set the "unmask" value to 022 before beginning installation on a UNIX system. Improper "unmask" settings result in unrestricted permissions
- Check the permissions of important folders and files created while installation to prevent unauthorized access
- **Example :**
  - Set the permission to 0750 or less for the files present in **$ORACLE_HOME except $ORACLE_HOME/bin**
  - Set the permission to 0755 or less for the files present in **$ORACLE_HOME/bin**
- To prevent **sysdba login** to the database ensure that only the Oracle software owner belongs to dba
- Ensure that the "Tkprof" utility is either removed or disabled from production. "Tkprof" utility formats SQL trace output to human readable format. This becomes a source for attackers to identify issues in running the database

Securing Network

- Ensure that the **listener** is secured
  - Set the listener password
  - Turn on logging
  - Set ADMIN_RESTRICTIONS in listener.ora to disable all runtime modifications
  - Set a different listener name than LISTENER
  - Set directories for tracing
  - Use IP addresses rather than hostnames
  - Ensure that the listener password is not stored in listener.ora file
- Create separate listeners for clients and for administration
- Ensure the use of a **firewall** to protect **Oracle** (Windows)
- Accept connections from short list of **IP addresses**
- View the **sqlnet.log** files on the server and client machines
- Check with the port scanner for open default ports
- Secure the intelligent agent
- Ensure that the connection is configured to implement encryption between clients and the database

How to Configure Encryption on the Client and the Server

- Open **Net Manager**
- In **Windows**: Navigate **Start→ Programs→ Oracle - HOME_NAME→ Configuration and Migration Tools→Net Manager**
- Select **Profile** from **Oracle Net Configuration → Local→ Profile**
- Select **Network Security** from **Naming List** to open **Network Security** window
- Click the **Encryption** tab
- For the Encryption option select either **Client** or **Server**
- Select one of the following as **Encryption Type (REQUESTED, REQUIRED, ACCEPTED, REJECTED)**
- Enter **Encryption Seed** of random characters of 10 -70 length (Client encryption should not match with the server encryption seed)
- Select one of the listed algorithms
  - Navigate to **File→ Save Network Configuration to update the sqlnet.ora** file

Control Access Data

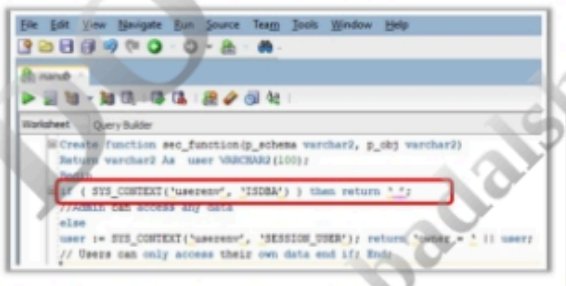
- Oracle has various methods to handle data securely. The following are some of the **methods** implemented to protect data:
  - **Oracle Virtual Private Database:** It creates a policy which enforces a **WHERE** clause for all SQL statements and restricts access to row and column level data
  - **Oracle Label Security:** This feature provides restriction for access to the database table at the **row level**
  - **Oracle Database Vault:** This feature restricts the privileged users and **DBA** of environment

Virtual Private Database

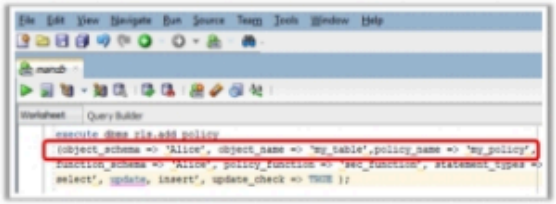
- **Virtual Private Database** helps DBA to define and implement row-level access control policies based on session attributes
- When a user accesses a table, view or synonym having a **VPD policy** attached
- Oracle server executes the attached policy function when ever a user executes a query
- Considering the **session attribute** or contents of the database, the policy function returns a predicate which is concatenated as a where clause
- The server executes the altered **SQL query**

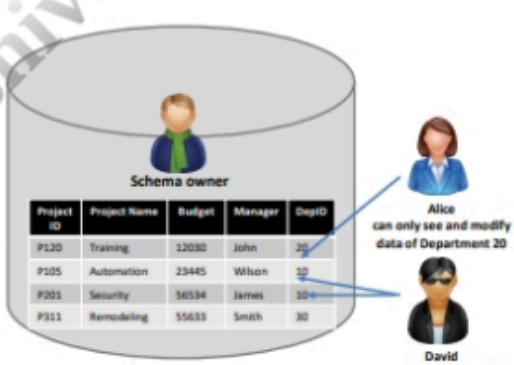
Virtual Private Database (Cont'd)

- **Implementation Virtual Private Database:**
  - **Step 1 : Create a policy function**



  - **Step 2 : Link created policy to a table**





Oracle Label Security

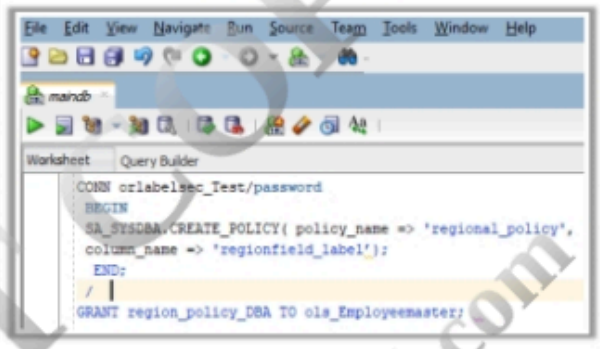
- **Oracle Virtual Private Database** technology implements **Oracle label security**. Each row having labels it compares with user label and privileges before displaying

Implementation of Oracle Label Security:

- To implement Oracle label security user has to checkout following steps:

Step1	Installation Oracle Label Security
Step2	Configure Instance
Step3	Create Policy to specify column
Step4	Define Component of label
Step5	Create Schema
Step6	Create Label Function
Step7	Apply Policy to table
Step8	Initialize Label
Step9	Test The Label Security Policy

Creating Policy for Test User

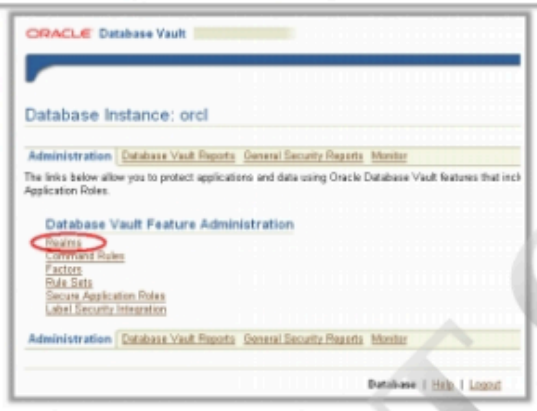


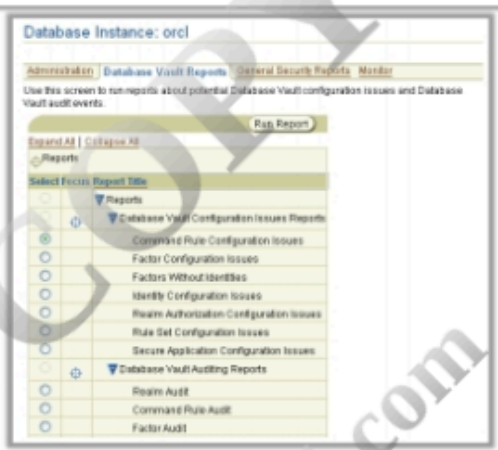
Database Vault

- Database vault protects the database against insider attacks by implementing **realms, factors, and command rules**
- Implements restriction on privileged users by:
  - Restricting privileged users from accessing application data
  - Enforcing separation of duty
- Implements Real time access controls by:
  - Restricting access based on IP address, authentication method, time of day etc.

Database Vault: Management and Reports

- **Database Vault Administrative Interface** contains web based Administrative Interface to manage **Realms, Rules, Factors, Reports, Dashboard**
- Database Vault Reporting
  - Generates various security reports for compliance
  - Generates Audit violation attempts
  - Generates Realm, Rule and Factor Reports
  - Generates System and Public Privileges reports

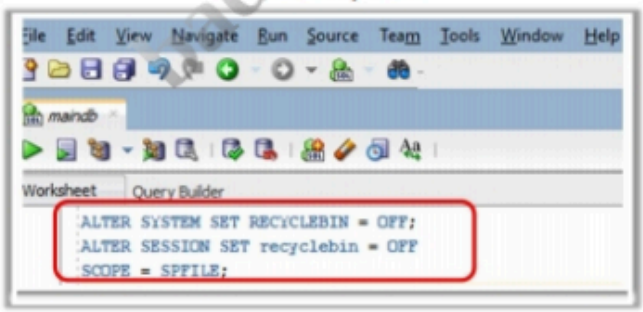




Database Vault: Disabling the Recycle Bin

- To prevent dropped database objects from exposure, disable the recycle bin
- SYS users having SYSDBA administrative privileges or users with **ALTER SYSTEM** system privilege can disable Recycle Bin

To Disable Recycle Bin



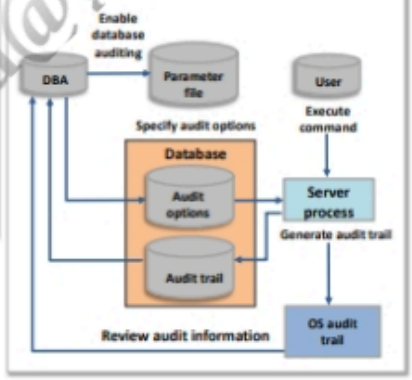
Audit Vault

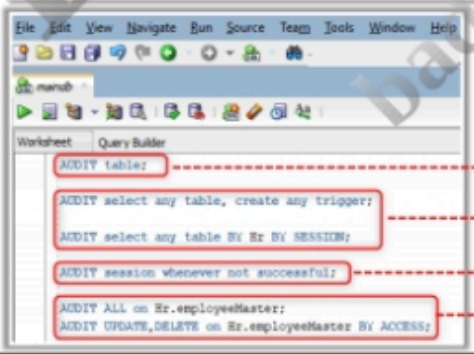
Audit Vault : Automated **Activity Management and Audit reporting**

1. Centralized audit policy management
2. Consolidates audit data into a secure repository
3. Identifies and alerts on detecting suspicious activities
4. Compliance reporting

Built-in Audit Tools: Standard Database Auditing

- Audits for privileged use including object access
- Can audit login events, use of system privileges, use of object privileges, SQL statements used
- Use **AUDIT_TRAIL** parameter to enable standard database auditing
  - NONE: Disable audit records
  - DB: Enable audit of records stored in the database
  - OS: Enable audit of records stored in the operating system audit trail





SQL statement auditing

System privilege auditing

Session auditing

Object privilege auditing

Built-in Audit Tools: Standard Database Auditing (Cont'd)

Viewing Auditing Options

Data Dictionary View	Description
ALL_DEF_AUDIT_OPTS	Default audit options
DBA_STMT_AUDIT_OPTS	Statement auditing options
DBA_PRIV_AUDIT_OPTS	Privileged auditing options
DBA_OBJ_AUDIT_OPTS	Schema object auditing options

Viewing Auditing Results

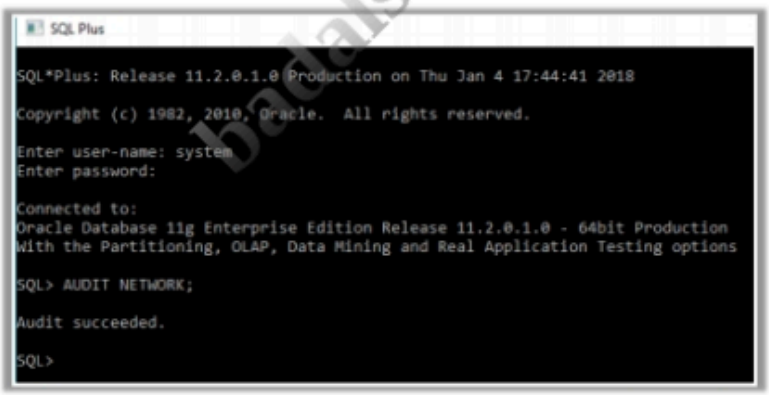
Audit Trail View	Description
DBA_AUDIT_TRAIL	All audit trail entries
DBA_AUDIT_EXISTS	Records for AUDIT EXIST/ DO NOT EXIST
DBA_AUDIT_OBJECT	Records concerning schema objects
DBA_AUDIT_SESSION	All connect and disconnect entries
DBA_AUDIT_STATEMENT	Statement auditing records

Standard Auditing Enable Network Auditing

Follow the given steps to Enable Network Auditing

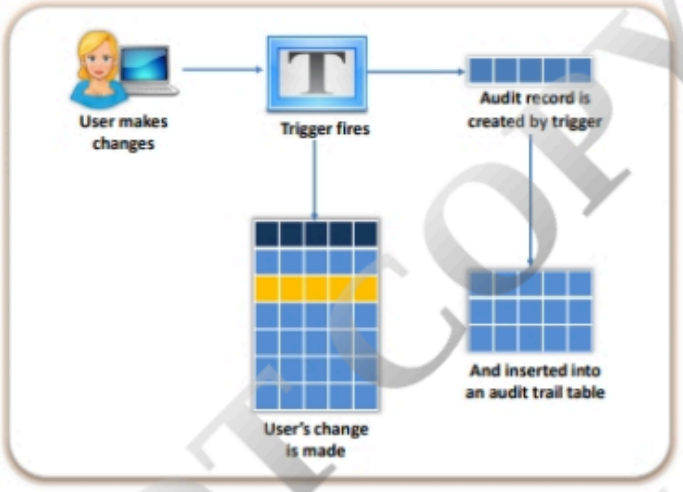
- Navigate to the **Start** menu in **Windows** and enter **SQLPLUS**
- Open **SQL*Plus Command Prompt** and use the command as shown here

```
SQL>Enter user-name: system
Enter password: SQL>AUDIT NETWORK;
```



Built-in Audit Tools: Value Based Auditing

- Value-based auditing: **Audits data** changed by **DML statements**



Built-in Audit Tools: Fine Grained Auditing (FGA)

- Audits SQL statements **(Insert, Update, Delete, and Select)**
- Audits data access based on content
- Can be implemented on a table or view
- Can execute a procedure
- Is administered with the **DBMS_FGA** package

DBMS_FGA package

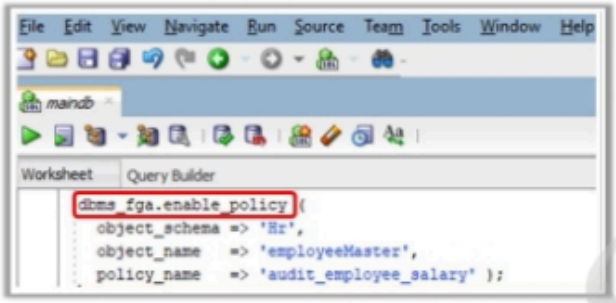
Subprogram	Description
ADD_POLICY	Creates an audit policy using the supplied predicate as the audit condition
DROP_POLICY	Drops an audit policy
ENABLE_POLICY	Enables an audit policy
DISABLE_POLICY	Disables an audit policy

Data Dictionary Views

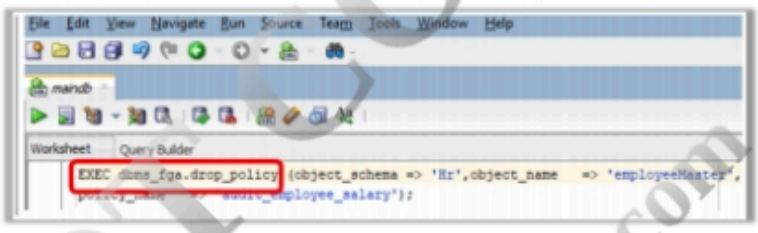
View Name	Description
DBA_FGA_AUDIT_TRAIL	All FGA events
ALL_AUDIT_POLICIES	All FGA policies for objects the current user can access
DBA_AUDIT_POLICIES	All FGA policies in the database
USER_AUDIT_POLICIES	All FGA policies for objects in the current user schema

Built-in Audit Tools: Fine Grained Auditing (FGA) (Cont'd)

Enabling a FGA Policy

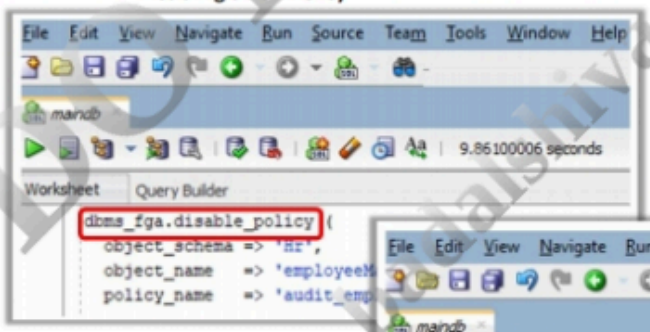


Dropping a FGA Policy

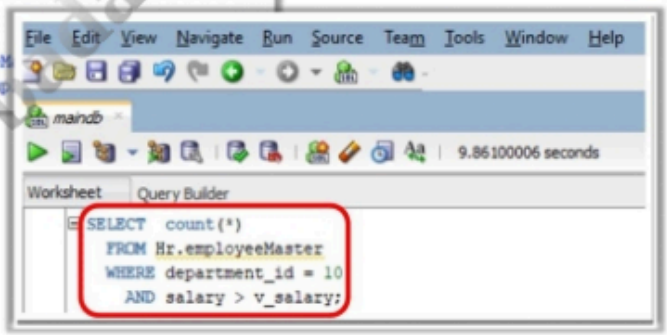


Built-in Audit Tools: Fine Grained Auditing (FGA) (Cont'd)

Disabling a FGA Policy



SQL Statements Resulting Audit



Recommended Audit Settings

- Ensure that **auditing** is **selective** and **effective**
- Auditing should focus on
  - Privileges, users
  - Selected tables with sensitive information
  - Secure configurations
- **Oracle provides following default audit configuration**

ALTER ANY PROCEDURE	AUDIT ROLE BY ACCESS
CREATE ANY JOB	CREATE EXTERNAL JOB
DROP ANY TABLE	GRANT ANY OBJECT PRIVILEGE
ALTER ANY TABLE	ALTER SYSTEM
CREATE ANY LIBRARY	CREATE PUBLIC DATABASE LINK
DROP PROFILE	GRANT ANY PRIVILEGE
ALTER DATABASE	ALTER USER
CREATE ANY PROCEDURE	CREATE SESSION
DROP USER	GRANT ANY ROLE
ALTER PROFILE	AUDIT SYSTEM
CREATE ANY TABLE	CREATE USER
EXEMPT ACCESS POLICY	AUDIT SYSTEM BY ACCESS
	DROP ANY PROCEDURE

Security Maintenance and Monitoring

Post Deployment Activities: Security Maintenance and Monitoring

1. Maintenance and monitoring is **iterative process** undertaken after initial deployment of the application
2. It includes set of activities that are being carried out to continuously maintain the security of web hosting environment. These activities allows to keep application up-to-date concerning the emerging vulnerabilities
3. The maintenance activities should be carried out at various level of web hosting environment
   - **OS level**
   - **Web server level**
   - **Application level**
4. The purpose of implementing the maintenance should be to modify the product without affecting its **integrity**

Security Maintenance Activities at OS Level

1. Maintain **Test server** and **Production server** separately
2. Look for the latest OS level security updates, patches, and hotfixes released continuously and apply them time to time
3. Monitor and analyze **system level logs**
4. Take **backup** of the data and OS regularly
5. Monitor the antivirus software to ensure updates are applied and functioning properly
6. Ensure **OS permissions** of all system folders are intact
7. Scan and update the system with the latest **antivirus** scan engine/virus definitions regularly
8. Ensure that spam/hacked services are not present by inspecting default start-up state of system services
9. Monitor **background processes** and startup items to prevent from malware infections continuously

Security Maintenance Activities at Web Container Level

I. Continuously look for the latest **versions, security updates, patches** and apply them time to time

II. Monitor and analyze Web container **logs**

III. Ensure that configured **web container security features** are intact

IV. Scan the **web server** periodically for identifying vulnerabilities and misconfigurations

V. Perform the penetration testing periodically to assess the effectiveness of existing security features

Security Maintenance Activities at Application Level

- Scan the application periodically for identifying **vulnerabilities** and **misconfigurations**
- Perform the **penetration testing** periodically to assess the effectiveness of existing security features
- **Patch** the application for vulnerabilities discovered
- Maintain **backup** of public websites

Module Summary

- Security should be the critical consideration while deploying any application
- JAVA web application secure deployment involves ensuring security at various levels from bottom to top
- Administrator should ensure the physical security of a host machine, its OS security, and security of the all other software installed on the machine
- A Web Application Firewall (WAF) provides a security layer that protects the web server from malicious traffic
- Administrator should ensure secure setting of the web server (Apache Tomcat, Jboss(WildFly))
- Administrator should configure and check the deployment security settings in both Server.xml and web.xml files carefully
- Maintenance and monitoring is an iterative process undertaken after the initial deployment of the application

This page is intentionally left blank.

References

Module 01: Understanding Application Security, Threats, and Attacks

1. J.D. Meier, Alex Mackman, Blaine Wastell, Cheat Sheet: Web Application Security Frame, last modified: 2005, http://msdn.microsoft.com/en-us/library/ff649461.aspx.
2. Ruby Qurashi, Eight steps for integrating security into application development, last modified: 2005, http://www.computerworld.com/s/article/106805/Eight_steps_for_integrating_security_into_application_development?taxonomyId=17&pageNumber=1.
3. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla, Anandha Murukan, Threat Modeling, last modified: 2003, http://msdn.microsoft.com/en-us/library/ff648644.aspx.
4. Russ McMahon, Security Development Lifecycle, accessed: December 22, 2017, https://www.owasp.org/images/4/42/Microsoft-SDL2.pdf.
5. Security Testing a .NET Application, accessed: December 22, 2017, http://etutorials.org/Programming/Programming+.net+security/Part+I+Fundamentals/Chapter+4.+The+Lifetime+of+a+Secure+Application/4.3+Security+Testing+a+.NET+Application/.
6. Ong Khai Wei, Securing Your Web Application against security vulnerabilities, accessed: December 22, 2017, http://www-07.ibm.com/sg/smarterbusiness/meettheexperts/includes/downloads/Securing_Your_Web_0910_eve.pdf.
7. Mark Sherman, Building Resilient Systems: The Secure Software Development Lifecycle, accessed: December 22,2017, http://higherlogicdownload.s3.amazonaws.com/AUVSI/c2a3ac12-b178-4f9c-a654-78576a33e081/UploadedImages/Proceedings/Breakouts/Cybersecurity/Mark%20Sherman-CMU_150722%20Automated%20Vehicles%20Symposium%20-%20CYBERSECURITY%20FOR%20AUTOMATED%20VEHICLES%20Breakout%20-%20Sherman%20v4%20-%20distribution(2).pdf.
8. Lalit Kale, Application Security-Understanding The Horizon, last modified: 2014, https://www.slideshare.net/lalitkale/application-securityi-understanding-the-horizon.
9. John Colley, Why Secure Coding is not Enough: Professionals' Perspective, accessed: December 22, 2017, https://link.springer.com/chapter/10.1007%2F978-3-8348-9363-5_30#page-1.
10. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla, Anandha Murukan, Web Application Security Methodology, accessed: December 22, 2017, http://www.guidanceshare.com/wiki/Web_Application_Security_Methodology.
11. File: Security in the SDLC Process.png, accessed: December 22, 2017, https://www.owasp.org/index.php/File:Security_in_the_SDLC_Process.png.
12. Software Security Maturity Models – BSIMM & OpenSAMM, last modified: 2016, https://jasonamorrow.com/bsimm-opensamm/.
13. State of Application Security Infographic, last modified: 2013, http://www.quotium.com/resources/state-of-application-security-infographic/.

Module 02: Security Requirements Gathering

14. Edward van Deursen, Jan Jaap Cannegieter, Eliciting security requirements needs a different process, accessed: January 2, 2018, https://re-magazine.ireb.org/issues/2015-2-bridging-the-impossible/eliciting-security-requirements/.

15. WHAT ARE SECURITY REQUIREMENTS?, accessed: January 2, 2018, https://www.securitycompass.com/media/pdf/article-what-are-security-requirements.pdf.
16. Application Security by Design, accessed: January 2, 2018, https://web.securityinnovation.com/hs-fs/hub/49125/file-14368892-pdf/whitepapers/application_security_by_design.pdf.
17. Nancy R. Mead, Eric D. Hough, Theodore R. Stehney, Security Quality Requirements Engineering (SQUARE) Methodology, last modified: 2005, https://resources.sei.cmu.edu/asset_files/technicalreport/2005_005_001_14594.pdf.
18. Jose Romero-Mariona, Hadar Ziv, Debra J. Richardson, Security Requirements Engineering: A Survey, last modified: 2008, https://isr.uci.edu/tech_reports/UCI-ISR-08-2.pdf.
19. P.Salinia, S.Kanmani, Security Requirements Engineering Process for Web Applications, accessed: January 2, 2018, https://www.sciencedirect.com/science/article/pii/S1877705812022412.
20. Hester Fox, Developing Secure Software, accessed: January 2, 2018, http://slideplayer.com/slide/10911394/.
21. Misuse case, accessed: January 2, 2018, https://en.wikipedia.org/wiki/Misuse_case.
22. Paco Hope, Peter White, Software Security Requirements, accessed: January 2, 2018, http://sqgne.org/presentations/2007-08/Hope-Sep-2007.pdf.
23. Donald G. Firesmith, Security Use Cases, accessed: January 2, 2018, http://www.jot.fm/issues/issue_2003_05/column6/.
24. Joshua J. Pauli, Refining Use/Misuse/Mitigation Use Cases for Security Requirements, accessed: January 2, 2018, http://file.scirp.org/pdf/JSEA_2014070811065395.pdf.
25. Steven Thomas, Abuser Story – User Stories to Prevent Hacking, last modified: 2014, http://itsadeliverything.com/abuser-story-user-stories-to-prevent-hacking.
26. Jim Bird, Adding Appsec to Agile: Security Stories, Evil User Stories and Abuse(r) Stories, accessed: January 2, 2018, https://dzone.com/articles/adding-appsec-agile-security.
27. Annette Tetmeyer, A POS Tagging Approach to Capture Security Requirements within an Agile Software Development Process, accessed: January 2, 2018, https://kuscholarworks.ku.edu/bitstream/handle/1808/11464/Tetmeyer_ku_0099M_12820_DATA_1.pdf;sequence=1.
28. Chandramohan Muniraman, Meledath Damodaran, A PRACTICAL APPROACH TO INCLUDE SECURITY IN SOFTWARE DEVELOPMENT, accessed: January 2, 2018, http://www.iacis.org/iis/2007/Muniraman_Damodaran.pdf.
29. SAMM - Security Requirements – 1, accessed: January 2, 2018, https://www.owasp.org/index.php/SAMM_-_Security_Requirements_-_1.
30. Jamie Boote, Software Integrity, last modified: 2016, https://www.synopsys.com/blogs/software-security/software-security-requirement/.
31. FRANK RIETTA, What Is an Abuser Story (Software), last modified: 2015, https://rietta.com/blog/2015/10/11/what-is-an-abuser-story-software/.
32. Agile Software Development: Don't Forget EVIL User Stories, accessed: January 2, 2018, https://www.owasp.org/index.php/Agile_Software_Development:_Don%27t_Forget_EVIL_User_Stories.
33. Adam Shostack, Threat Modeling: Designing for Security, accessed: January 2, 2018, https://books.google.co.in/books?id=YiHcAgAAQBAJ&pg=PT620&lpg=PT620&dq=threat+trees&source=bl&ots=eSUlHC7_Oq&sig=vUjg0WxKM9VUvwQigjAJYeln940&hl=en&sa=X&ved=0ahUKEwjghvD2wOHQAhXDo5QKHR1uAbM4FBDoAQghMAE#v=onepage&q=threat%20trees&f=false.
34. E. Gottesdlener, Abuse Case Guidelines, accessed: January 2, 2018, https://www.ebgconsulting.com/Pubs/Articles/Abuse%20Case%20Guidelines.pdf.

Module 03: Secure Application Design and Architecture

35. SECURE DESIGN AND ARCHITECTURE REVIEW, accessed: January 10, 2018, https://www.appsecurelabs.com/consulting/secure-design-and-architecture-review/#.
36. Application Security by Design, accessed: January 10, 2018, https://web.securityinnovation.com/hs-fs/hub/49125/file-14368892-pdf/whitepapers/application_security_by_design.pdf.
37. J.D. Meier, Alex Mackman, Blaine Wastell, How To: Create a Threat Model for a Web Application at Design Time, last modified: 2005, https://msdn.microsoft.com/en-us/library/ff647894.aspx.
38. Application Threat Modeling, accessed: January 10, 2018, https://www.owasp.org/index.php/Application_Threat_Modeling.
39. Attack Surface Analysis Cheat Sheet, accessed: January 10, 2018, https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet.
40. Umut IŞIK, Threat modelling with sample application, last modified: 2014, https://www.slideshare.net/UmutIIK/threat-modelling-withsampleapplication.
41. Sherif Koussa, Simplified Security Code Review Process, last modified: 2013, https://www.slideshare.net/skoussa/simplified-security-code-review-process.
42. James Jasper Fletcher, Software Security Assessment, accessed: January 10, 2018, http://slideplayer.com/slide/5776271/.
43. Building Real Software, last modified: 2013, http://swreflections.blogspot.in/2013/06/what-is-important-in-secure-software.html.

Module 04: Secure Coding Practices for Input Validation

44. Open Source Validation Frameworks, accessed: January 18, 2018, http://java-source.net/open-source/validation.
45. Java validation filter, accessed: January 18, 2018, http://zetcode.com/java/validationfilter/.
46. Category:OWASP Enterprise Security API, accessed: January 18, 2018, https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API.
47. Data validations in Struts2, accessed: January 18, 2018, http://www.techmyguru.com/struts2/index.php?section=13/Data_validations_in_Struts2.
48. Validation, Data Binding, and Type Conversion, accessed: January 18, 2018, https://docs.spring.io/spring/docs/4.1.x/spring-framework-reference/html/validation.html.
49. Arvind Rai, Validation Framework in Spring with Example, last modified: 2013, https://www.concretepage.com/spring/validation-framework-spring-example.
50. Baeldung, Java Bean Validation Basics, last modified: 2018, http://www.baeldung.com/javax-validation.
51. Spring MVC form validation with custom validator, accessed: January 18, 2018, http://javainsimpleway.com/spring-mvc-form-validation-with-custom-validator/.

Module 05: Secure Coding Practices for Authentication and Authorization

52. RAJNISH BHATIA, ALL CODES, last modified: 2008, http://rajnishbhatia19.blogspot.in/search/label/Kerberos.
53. Willie Wheeler, SECTION 1 Introduction, accessed: January 25, 2018, https://dzone.com/refcardz/expression-based-authorization?chapter=1.
54. Spring Book – Chapter 15 – Web Application Security with Spring, last modified: 2013, https://www.javacodebook.com/2013/08/08/spring-book-chapter-15-web-application-security-with-spring/6/.

55. Prasanth Gullapalli, Java Code Greeks, accessed: January 25, 2018, https://www.javacodegeeks.com/2013/11/spring-security-behind-the-scenes.html.
56. Alessandro, Spring Security Tutorial: Form Login, last modified: 2014, http://codehustler.org/blog/spring-security-tutorial-form-login/.
57. Jason Ferguson, Spring Security 3, last modified: 2011, https://www.slideshare.net/jasonferguson1/spring-security-3.
58. Willie Wheeler, SECTION 4 Web Authorization, accessed: January 25, 2018, https://dzone.com/refcardz/expression-based-authorization?chapter=4.
59. Willie Wheeler, SECTION 6 Web Authorization, Revisited, accessed: January 25, 2018, https://dzone.com/refcardz/expression-based-authorization?chapter=6.
60. Brad Rubin, Java security, Part 1: Crypto basics, last modified: 2002, https://www.ibm.com/developerworks/java/tutorials/j-sec1/j-sec1.html.
61. Eugen Paraschiv, Spring Security – Persistent Remember Me, last modified: 2016, http://www.baeldung.com/spring-security-persistent-remember-me.
62. Eugen Paraschiv, Spring Security Remember Me, last modified: 2016, http://www.baeldung.com/spring-security-remember-me.
63. Saeidzebardast, How to set Unlimited Session Timeout in Tomcat, last modified: 2016, https://coderwall.com/p/eocuyw/how-to-set-unlimited-session-timeout-in-tomcat.
64. Fusion Middleware Programming Security for Oracle WebLogic Server, accessed: January 25, 2018, https://docs.oracle.com/cd/E23943_01/web.1111/e13711/thin_client.htm#SCPRG133.

Module 06: Secure Coding Practices for Cryptography

65. Mkyong, Spring Security password hashing example, last modified: 2014, https://www.mkyong.com/spring-security/spring-security-password-hashing-example/.
66. Aaron Zauner, Javascript Object Signing & Encryption, last modified: 2015, https://www.slideshare.net/a_z_e_t/javascript-object-signing-encryption.
67. Dennis Detering, On the (in-)security of JavaScript Object Signing and Encryption, accessed: February 2, 2018, https://www.nds.rub.de/media/nds/arbeiten/2017/05/07/Masterthesis_Security-of-JOSE_Detering-FINAL.pdf.
68. Scott Arciszewski, Paseto is a Secure Alternative to the JOSE Standards (JWT, etc.), accessed: February 2, 2018, https://paragonie.com/blog/2017/03/jwt-json-web-tokens-is-bad-standard-that-everyone-should-avoid.
69. Brian Campbell, JOSE Can You See, last modified: 2014, https://www.slideshare.net/briandavidcampbell/jose-can-you-see-34360871.
70. Prabath Siriwardena, JWT, JWS and JWE for Not So Dummies! (Part I), last modified: 2016, https://medium.facilelogin.com/jwt-jws-and-jwe-for-not-so-dummies-b63310d201a3.
71. Javascript Object Signing and Encryption (JOSE), accessed: February 2, 2018, http://jose.readthedocs.io/en/latest/.

Module 07: Secure Coding Practices for Session Management

72. PANKAJ, Spring MVC Exception Handling – @ControllerAdvice, @ExceptionHandler, HandlerExceptionResolver, accessed: February 26, 2018, https://www.journaldev.com/2651/spring-mvc-exception-handling-controlleradvice-exceptionhandler-handlerexceptionresolver.

73. Spring MVC Exception Handling, last modified: 2016, https://memorynotfound.com/spring-mvc-exception-handling/.

74. Exception Handling in Spring MVC with Example, mast modified: 2016, https://www.dineshonjava.com/exception-handling-in-spring-mvc/.

75. How to handle exceptions in Struts2, last modified: 2015, http://www.codejava.net/frameworks/struts/how-to-handle-exceptions-in-struts2.

Module 08: Secure Coding Practices for Error Handling

1. What is HttpSession?, accessed: February 15, 2018, https://www.studytonight.com/servlet/httpsession.php.

2. Eugen Paraschiv, Java Session Timeout, accessed: February 15, 2018, http://www.baeldung.com/servlet-session-timeout.

3. Johnmelton, Session Cookie HttpOnly Flag Java, last modified: 2012, https://www.whitehatsec.com/blog/session-cookie-httponly-flag-java/.

4. Managing Session Attributes For Mvc Applications, accessed: February 15, 2018, https://wastemanagement.site/info/spring-mvc-session-management-tutorial/managing-session-attributes-for-mvc-applications/ZbPmeb3HYqk.html.

5. Biju Kunjummen, Using Http Session With Spring Based Web Applications, last modified: 2014, https://dzone.com/articles/using-http-session-spring.

6. How to implement a HTTPS login page in a web application?, accessed: February 15, 2018, https://stackoverflow.com/questions/1454021/how-to-implement-a-https-login-page-in-a-web-application.

7. Secure Session Management Tips, last modified: 2011, https://wblinks.com/notes/secure-session-management-tips/.

Module 09: Static and Dynamic Application Security Testing (SAST & DAST)

8. CRV2 FrameworkSpecIssuesASPTop10, accessed: March 12, 2018, https://www.owasp.org/index.php/CRV2_FrameworkSpecIssuesASPTop10.

9. Secure Code Review, accessed: March 12, 2018, https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/secure-code-review.

10. SECURE CODE REVIEW, accessed: March 12, 2018, https://www.appsecurelabs.com/consulting/secure-code-review/.

11. Naga Venkata Sunil Alamuri, Null meet Code Review, last modified: 2015, https://www.slideshare.net/sunilanv/null-meet-codereviewslideshare-5?qid=e80636f8-add5-4275-83a6-d3b03d46deee&v=&b=&from_search=3.

12. Secure Code Review: A Practical Approach, last modified: 2013, http://resources.infosecinstitute.com/secure-code-review-practical-approach/#gref.

13. secure_code_review_20141217.pptx, accessed: March 12, 2018, http://opensecuritytraining.info/SecureCodeReview_files/secure_code_review_20141217.pdf.

14. Security Code Review, accessed: March 12, 2018, https://www.torridnetworks.com/services/assessment/security-code-review.

15. Uncover Hidden Vulnerabilities With Security Code Review, accessed: March 12, 2018, https://www.paladion.net/source-code-review-services.

16. Security Code Review, last modified: 2013, https://www.youtube.com/watch?v=Q5ifAbg3ins&t=1712s.
17. Code Review, accessed: March 12, 2018, http://www.net-square.com/code_review.html.
18. J.D. Meier, Alex Mackman, Blaine Wastell, Prashant Bansode, Jason Taylor, Rudolph Araujo, How To: Perform a Security Code Review for Managed Code (.NET Framework 2.0), last modified: 2005, https://msdn.microsoft.com/en-us/library/ff649315.aspx.
19. Naga Venkata Sunil Alamuri, Null meet Code Review, last modified: 2015, https://www.slideshare.net/sunilanv/null-meet-codereviewslideshare-5?qid=e80636f8-add5-4275-83a6-d3b03d46deee&v=&b=&from_search=3.
20. Test, 1, 2, 3, test, test, last modified: 2011, https://www.64k-tec.de/2011/03/test-1-2-3-test-test/.
21. Source Code Analysis Tools, accessed: March 12, 2018, https://www.owasp.org/index.php/Source_Code_Analysis_Tools.
22. List of tools for static code analysis, accessed: March 12, 2018, https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis#.NET.
23. Eoin Keary, OWASP_AlphaRelease_CodeReviewGuide2.0.pdf, accessed: March 12, 2018, https://www.owasp.org/images/7/78/OWASP_AlphaRelease_CodeReviewGuide2.0.pdf.
24. OWASP CODE REVIEW GUIDE, accessed: March 12, 2018, https://www.owasp.org/images/2/2e/OWASP_Code_Review_Guide-V1_1.pdf.
25. Unvalidated Redirects and Forwards Cheat Sheet, accessed: March 12, 2018, https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.
26. Top 10 2010-A10-Unvalidated Redirects and Forwards, accessed: March 12, 2018, https://www.owasp.org/index.php/Top_10_2010-A10-Unvalidated_Redirects_and_Forwards.
27. Codereview-Authentication, accessed: March 12, 2018, https://www.owasp.org/index.php/Codereview-Authentication.
28. Philippe Arteau, Modern Static Analysis for .NET, last modified: 2016, http://gosecure.net/2016/11/23/modern-static-analysis-net/.
29. CWE-259: Use of Hard-coded Password, accessed: March 12, 2018, https://cwe.mitre.org/data/definitions/259.html.
30. CWE-321: Use of Hard-coded Cryptographic Key, accessed: March 12, 2018, https://cwe.mitre.org/data/definitions/321.html.
31. Use of hard-coded password, accessed: March 12, 2018, https://www.owasp.org/index.php/Use_of_hard-coded_password.
32. Use of hard-coded cryptographic key, accessed: March 12, 2018, https://www.owasp.org/index.php/Use_of_hard-coded_cryptographic_key.
33. Password Management: Hardcoded Password, accessed: March 12, 2018, https://www.owasp.org/index.php/Password_Management:_Hardcoded_Password.
34. Yamin Khakhu, Encrypt ConnectionString in Web.Config, last modified: 2014, https://www.codeproject.com/Tips/795135/Encrypt-ConnectionString-in-Web-Config.
35. How to: Secure Connection Strings When Using Data Source Controls, accessed: March 12, 2018, https://msdn.microsoft.com/en-IN/library/dx0f3cf2(v=vs.85).aspx.
36. Role-based authorization in ASP.NET Core, accessed: March 12, 2018, https://docs.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-2.1.
37. ASP.NET Authorization, accessed: March 12, 2018, https://msdn.microsoft.com/en-us/library/wce3kxhd.aspx.

38. Gsark, "Your Login Attempt was not Successful. Please Try Again." - ASP.NET Login Control, last modified: 2009, https://www.codeproject.com/Articles/27682/Your-Login-Attempt-was-not-Successful-Please-Try.

39. ASP.NET 2.0 Security Questions and Answers – Authentication, accessed: March 12, 2018, http://www.guidanceshare.com/wiki/ASP.NET_2.0_Security_Questions_and_Answers_-_Authentication.

40. Sample Secure Code Review Report, accessed: March 12, 2018, https://www.mitre.org/sites/default/files/publications/secure-code-review-report-sample.pdf.

41. Anita D'Amico, Dynamic Application Security Testing Tools: Searching the Black Box, last modified: 2015, https://codedx.com/2015/01/30/dynamic-application-security-testing-tools/.

42. HARI BALASUNDARAM, A Baseline Approach to Security Testing, last modified: 2014, https://blog.box.com/blog/a-baseline-approach-to-security-testing/.

43. Kevin Fealey, What Good is this Tool? A Guide to Choosing the Right Application Security Testing Tools, last modified: 2015, https://www.slideshare.net/kfealey/what-good-is-this-tool-final.

44. Apoorva Phadke, AST vs. DAST: What's the best method for application security testing?, last modified: 2016, https://www.synopsys.com/blogs/software-security/sast-vs-dast/.

45. Learn About WebInspect Automated Dynamic Scans (DAST), last modified: 2015, https://www.youtube.com/watch?v=j0idyQrkPGE.

46. Brian Shura , Web Application Security Scanner List, accessed: http://projects.webappsec.org/w/page/13246988/Web%20Application%20Security%20Scanner%20List.

47. Category:Vulnerability Scanning Tools, accessed: March 12, 2018, https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools.

48. Using Burp to Detect SQL Injection Flaws, accessed: March 12, 2018, https://support.portswigger.net/customer/portal/articles/1965677-Methodology_SQL_Injection_.html.

49. Using Burp to Exploit SQL Injection Vulnerabilities: The UNION Operator, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/2163777-Methodology_SQL%20Exploitation_Union.html.

50. Using Burp to Brute Force a Login Page, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1964020-Methodology_Attacking%20Authentication_Brute%20Force%20Login.html.

51. Using Burp to Test for Missing Function Level Access Control, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965720-Methodology_Missing%20Function%20Level%20Access%20Control.html#ForcedBrowsing.

52. Using Burp Scanner to Find Cross-Site Scripting (XSS) Issues, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965737-Methodology_Using%20Burp%20Scanner%20To%20Find%20XSS.html.

53. Using Burp to Test for Insecure Direct Object References, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965691-Methodology_Insecure%20Direct%20Object%20References.html.

54. Using Burp to Test for Security Misconfiguration Issues, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965728-Methodology_Security%20Misconfiguration.html.

55. Using Burp to Test for Sensitive Data Exposure Issues, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965730-Methodology_Sensitive%20Data%20Exposure.html.

56. Using Burp's Site Map to Test for Access Control Issues, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1969844-Methodology_Attacking%20Access%20Controls_User%20Accounts.html.

57. Using Burp's "Request in Browser" Function to Test for Access Control Issues, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1969842-Methodology_Attacking%20Access%20Controls_Request%20in%20Browser.html.

58. Using Burp to Test for Cross-Site Request Forgery (CSRF), accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965674-Methodology_Cross-Site%20Request%20Forgery%20(CSRF).html.

59. Using Burp to Test for Components with Known Vulnerabilities, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965732-Methodology_Testing%20for%20Components%20with%20Known%20Vulnerabilities.html/

60. Using Burp to Test for Open Redirections, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965733-Methodology_Testing%20for%20Open%20Redirections.html.

61. Using Burp to Bypass Hidden Form Fields, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1965741-Methodology_Bypassing%20Client-Side%20Controls_Hidden%20Form%20Fields.html.

62. Using Burp to Bypass Client Side JavaScript Validation, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1964212-Methodology_Bypassing%20Client-Side%20Controls_JavaScript%20Validation.html.

63. Using Burp to Hack Cookies and Manipulate Sessions, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1964073-Methodology_Attacking%20Session%20Management_Hacking%20Cookies.html.

64. Using Burp to Test Session Token Generation, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/1964169-using-burp-to-test-session-token-generation.

65. Using Burp Scanner to Test for DOM-Based, accessed: March 12, 2018, XSShttps://support.portswigger.net/customer/en/portal/articles/2325926-using-burp-scanner-to-test-for-dom-based-xss.

66. Using Burp to find Clickjacking Vulnerabilities, accessed: March 12, 2018, https://support.portswigger.net/customer/en/portal/articles/2363105-using-burp-to-find-clickjacking-vulnerabilities.

67. Gareth Heyes, Burp Clickbandit: A JavaScript based clickjacking PoC generator, last modified: 2015, https://portswigger.net/blog/burp-clickbandit-a-javascript-based-clickjacking-poc-generator.

68. Marcel Birkner, Automated Security Testing of web applications using OWASP Zed Attack Proxy, last modified: 2013, https://blog.codecentric.de/en/2013/10/automated-security-testing-web-applications-using-owasp-zed-attack-proxy/.

69. Geethu Alexander, ZAP Penetration Testing: A simple Tutorial to Detect Vulnerabilities, last modified: 2016, http://www.toobler.com/blog/zap-penetration-testing-simple-tutorial.

70. Jim Bird, Choosing Between a Penetration Test and a Secure Code Review, last modified: 2013, https://dzone.com/articles/choosing-between-penetration.

71. Sarah Vonnegut, 5 Best Practices for the Perfect Secure Code Review, last modified: 2016, https://www.checkmarx.com/2016/02/05/5-best-practices-perfect-secure-code-review/.

72. David Jacobs, Reviewing applications for security: Code review best practices, accessed: March 12, 2018, https://searchitchannel.techtarget.com/tip/Reviewing-applications-for-security-Code-review-best-practices.

73. Secure Code Review, accessed: March 12, 2018, https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/secure-code-review.

Module 10: Secure Deployment and Maintenance

74. Fusion Middleware Securing a Production Environment for Oracle WebLogic Server, accessed: March 27, 2018, https://docs.oracle.com/cd/E21764_01/web.1111/e13705/practices.htm#LOCKD124.

75. J.D. Meier, Alex Mackman, Blaine Wastell, Prashant Bansode, Andy Wigley, Kishore Gopalan, patterns & practices Security Deployment Review Index, last modified: 2005, https://msdn.microsoft.com/en-us/library/ff648510.aspx.

76. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, Chapter 22 :Deployment Review, accessed: March 27, 2018, https://msdn.microsoft.com/en-us/library/aa302438.aspx.

77. Tjylen Veselyj, Web Application Firewall (WAF) DAST/SAST combination, last modified: 2014, https://www.slideshare.net/TjylenVeselyj/web-application-firewall-waf-dastsast-combination-30531258.

78. Logan Kipp, Ask a Security Professional: Firewall Series – Part Two: Web Application Firewalls, last modified: 2016, https://wpdistrict.sitelock.com/blog/firewall-series-part2-waf/.

79. Web Application Firewall, accessed: March 27, 2018, http://tectonicsecurity.com/managed-security-services/web-application-firewall/.

80. Web Application Firewall, accessed: March 27, 2018, https://www.owasp.org/index.php/Web_Application_Firewall.

81. Web Application Firewall (WAF), accessed: March 27, 2018, https://f5.com/glossary/web-application-firewall.

82. Margaret Rouse, Web application firewall (WAF), accessed: March 27, 2018, https://searchsecurity.techtarget.com/definition/Web-application-firewall-WAF.

83. Category:OWASP Best Practices: Use of Web Application Firewalls, accessed: March 27, 2018, https://www.owasp.org/index.php/Category:OWASP_Best_Practices:_Use_of_Web_Application_Firewalls.

84. Brad Causey, Introduction to Web application firewalls in the enterprise, accessed: March 27, 2018, https://searchsecurity.techtarget.com/feature/Introduction-to-Web-application-firewalls-in-the-enterprise.

85. Why WAF Is Not Enough for Protecting Your Database, last modified: 2015, http://www.hexatier.com/why-web-application-firewalls-waf-only-cannot-protect-your-databases/.

86. Michael Thumann, The 5 Myths of Web Application Firewalls, last modified: 2012, https://insinuator.net/2012/04/the-5-myths-of-web-application-firewalls-2/.

87. Securing tomcat, accessed: March 27, 2018, https://www.owasp.org/index.php/Securing_tomcat.

88. 15 Ways To Secure Apache Tomcat 8, last modified: 2016, https://www.upguard.com/articles/15-ways-to-secure-apache-tomcat-8.

89. Mthomas, Best Practices for Securing Apache Tomcat 7, last modified: 2011, http://www.tomcatexpert.com/blog/2011/11/02/best-practices-securing-apache-tomcat-7.

90. Tomcat Security, accessed: March 27, 2018, https://www.unidata.ucar.edu/software/thredds/current/tds/reference/TomcatSecurity.html.

91. Jason Brittain, Ian F. Darwin,Top, Ten Tomcat Configuration Tips, last modified: 2003, http://www.onjava.com/pub/a/onjava/2003/06/25/tomcat_tips.html?page=1.

92. Apache Tomcat 7, accessed: March 27, 2018, https://tomcat.apache.org/tomcat-7.0-doc/config/filter.html#CSRF_Prevention_Filter.

93. Apache Tomcat 8, accessed: March 27, 2018, https://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html.

94. Oracle Database Hardening Guide, last modified: 2014, https://oracledock.wordpress.com/2014/10/15/oracle-database-hardening-guide/.

95. Hardening the Oracle 11g Database - Initial Steps, last modified: 2007, http://www.securedba.com/securedba/2007/12/hardening-the-o.html.

96. Authentication Methods, accessed: March 27, 2018, https://docs.oracle.com/cd/B19306_01/network.102/b14266/authmeth.htm#BABCGGEB.

97. Storing Passwords in an Oracle Database, accessed: March 27, 2018, https://oracle-base.com/articles/9i/storing-passwords-in-the-database-9i.

98. Oracle 11g Tutorial, accessed: March 27, 2018, http://eoracle11g.blogspot.in/p/alter-user-information.html.

99. Oracle Database 2 Day + Security Guide, accessed: March 27, 2018, https://docs.oracle.com/cd/E11882_01/server.112/e10575.pdf.

100. Configuring Oracle Database Network Encryption and Data Integrity, accessed: March 27, 2018, https://docs.oracle.com/database/121/DBSEG/asoconfg.htm#DBSEG020.

101. Securing the Network, accessed: March 27, 2018, https://docs.oracle.com/cd/B28359_01/server.111/b28337/tdpsg_network_secure.htm#TDPSG90000.

102. Oracle Label Security, accessed: March 27, 2018, https://oracle-base.com/articles/9i/oracle-label-security-9i#CreateTestSchema.

This page is intentionally left blank.

Printed by: badalshiva@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

EC-Council

EC-COUNCIL OFFICIAL CURRICULA