CIS Benchmarks

CIS Apache Tomcat 9 Benchmark

v1.1.0 - 12-18-2020

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

Table of Contents

Overview

This document, Security Configuration Benchmark for Apache Tomcat 9, provides prescriptive guidance for establishing a secure configuration posture for Apache Tomcat versions 9 running on Linux. This guide was tested against Apache Tomcat 9 as installed by tar packages provided by Apache. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Apache Tomcat on a Linux platform.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
Monospace font	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

  Items in this profile intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

- **Level 2**

  This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as defense in depth measure
  - may negatively inhibit the utility or performance of the technology

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Contributor
Joern Krueger
James Scott

Editor
Tim Harrison, Center for Internet Security

Recommendations

1 Remove Extraneous Resources

1.1 Remove extraneous files and directories (Manual)

Profile Applicability:

- Level 2

Description:

The installation may provide example applications, documentation, and other directories which may not serve a production use.

Rationale:

Removing sample resources is a defense in depth measure that reduces potential exposures introduced by these resources.

Audit:

Perform the following to determine the existence of extraneous resources:

```
$ ls -l $CATALINA_HOME/webapps/examples \
   $CATALINA_HOME/webapps/docs \
   $CATALINA_HOME/webapps/ROOT \
   $CATALINA_HOME/webapps/host-manager \
   $CATALINA_HOME/webapps/manager
```

No output implies no sample resources are present.

Remediation:

Perform the following to remove extraneous resources:

```
$ rm -rf $CATALINA_HOME/webapps/docs \
     $CATALINA_HOME/webapps/examples \
     $CATALINA_HOME/webapps/ROOT
```

If the Manager and HOST-Manager application are not utilized, also remove the following resources:

```
$ rm -rf $CATALINA_HOME/webapps/host-manager \
     $CATALINA_HOME/webapps/manager
```

Default Value:

`docs`, `examples`, `ROOT`, `manager` and `host-manager` are default web applications shipped with Tomcat.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html#Default_web_applications/General

1.2 Disable Unused Connectors (Manual)

Profile Applicability:

- Level 2

Description:

The default installation of Tomcat includes connectors with default settings. These are traditionally set up for convenience. It is best to remove these connectors and enable only what is needed.

Rationale:

Improperly configured or unnecessarily installed `Connectors` may lead to a security exposure.

Audit:

Execute the following command to find configured `Connectors`. Ensure only those required are present and not commented out:

```
$ grep "Connector" $CATALINA_HOME/conf/server.xml
```

Remediation:

Within the `$CATALINA_HOME/conf/server.xml`, remove, or comment out, each unused `Connector`. For example, to disable an instance of the `HTTPConnector`, remove the following:

```
<Connector className="org.apache.catalina.connector.http.HttpConnector"
...
connectionTimeout="60000"/>
```

Default Value:

`$CATALINA_HOME/conf/server.xml`, has the following connectors defined by default:

- A non-SSL HTTP Connector bound to port `8080`
- An AJP Connector bound to port `8009`

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html
2. https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html#Connectors

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2 Limit Server Platform Information Leaks

Limiting Server Platform Information Leaks make it harder for attackers to determine which vulnerabilities affect the server platform.

2.1 Alter the Advertised server.info String (Manual)

Profile Applicability:

- Level 2

Description:

The `server.info` attribute contains the name of the application service. This value is presented to Tomcat clients when clients connect to the tomcat server.

Rationale:

Altering the `server.info` attribute may increase the complexity for attackers to determine which vulnerabilities affect the server platform.

Audit:

Perform the following to determine if the server.info value has been changed:

Extract the `ServerInfo.properties` file and examine the `server.info` attribute.

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
$ grep server.info org/apache/catalina/util/ServerInfo.properties
```

Remediation:

Perform the following to alter the server platform string that gets displayed when clients connect to the tomcat server.

1. Extract the `ServerInfo.properties` file from the `catalina.jar` file:

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

2. Navigate to the `util` directory that was created

```
cd org/apache/catalina/util
```

3. Open `ServerInfo.properties` in an editor
4. Update the `server.info` attribute in the `ServerInfo.properties` file.

```
server.info=<SomeWebServer>
```

5. Update the `catalina.jar` with the modified `ServerInfo.properties` file.

```
$ jar uf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

Default Value:

The default value for the `server.info` attribute is `Apache Tomcat/<version>`. For example, `Apache Tomcat/9.0.0.M9`.

References:

1. https://www.owasp.org/index.php/Securing_tomcat

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.2 Alter the Advertised server.number String (Manual)

Profile Applicability:

- Level 2

Description:

The `server.number` attribute represents the specific version of Tomcat that is executing. This value is presented to Tomcat clients when connect.

Rationale:

Advertising a valid server version may provide attackers with information useful for locating vulnerabilities that affect the server platform. Altering the server version string may increase the complexity for attackers to determine which vulnerabilities affect the server platform.

Audit:

Perform the following to determine if the `server.number` value has been changed:

Extract the `ServerInfo.properties` file and examine the `server.number` attribute.

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
$ grep server.number org/apache/catalina/util/ServerInfo.properties
```

Remediation:

Perform the following to alter the server version string that gets displayed when clients connect to the server.

1. Extract the `ServerInfo.properties` file from the `catalina.jar` file:

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

2. Navigate to the `util` directory that was created

```
$ cd org/apache/Catalina/util
```

3. Open `ServerInfo.properties` in an editor
4. Update the `server.number` attribute

```
server.number=<someversion>
```

5. Update the `catalina.jar` with the modified `ServerInfo.properties` file.

```
$ jar uf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

Default Value:

The default value for the `server.number` attribute is a four part version number, such as `9.0.0.0`.

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.3 Alter the Advertised server.built Date (Manual)

Profile Applicability:

- Level 2

Description:

The `server.built` date represents the date which Tomcat was compiled and packaged. This value is presented to Tomcat clients when clients connect to the server.

Rationale:

Altering the `server.built` string may make it harder for attackers to fingerprint which vulnerabilities affect the server platform.

Audit:

Perform the following to determine if the `server.built` value has been changed:

Extract the `ServerInfo.properties` file and examine the `server.built` attribute.

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
$ grep server.built org/apache/catalina/util/ServerInfo.properties
```

Remediation:

Perform the following to alter the server version string that gets displayed when clients connect to the server.

1. Extract the `ServerInfo.properties` file from the `catalina.jar` file:

```
$ cd $CATALINA_HOME/lib
$ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

2. Navigate to the `util` directory that was created

```
$ cd org/apache/Catalina/util
```

3. Open `ServerInfo.properties` in an editor
4. Update the `server.built` attribute in the `ServerInfo.properties` file.

```
server.built=
```

5. Update the `catalina.jar` with the modified `ServerInfo.properties` file.

```
$ jar uf catalina.jar org/apache/catalina/util/ServerInfo.properties
```

Default Value:

The default value for the `server.built` attribute is build date and time. For example, `Jul 8 2008 11:40:35`.

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.4 Disable X-Powered-By HTTP Header and Rename the Server Value for all Connectors (Automated)

Profile Applicability:

- Level 2

Description:

The `xpoweredBy` setting determines if Apache Tomcat will advertise its presence via the `X-Powered-By` HTTP header. It is recommended that this value be set to `false`. The server attribute overrides the default value that is sent down in the HTTP header further masking Apache Tomcat.

Rationale:

Preventing Tomcat from advertising its presence in this manner may increase the complexity for attackers to determine which vulnerabilities affect the server platform.

Audit:

Perform the following to determine if the server platform, as advertised in the HTTP Server header, has been changed:

1. Locate all Connector elements in `$CATALINA_HOME/conf/server.xml`.
2. Ensure each Connector that has the `xpoweredBy` attribute does **NOT** have it set to `true`.

Remediation:

Perform the following to prevent Tomcat from advertising its presence via the `X-PoweredBy` HTTP header.

1. Add the `xpoweredBy` attribute to each Connector specified in `$CATALINA_HOME/conf/server.xml`. Set the `xpoweredBy` attributes value to `false`.

```
<Connector
...
xpoweredBy="false" />
```

   Alternatively, ensure the `xpoweredBy` attribute for each Connector specified in `$CATALINA_HOME/conf/server.xml` is absent.

2. Add the server attribute to each Connector specified in `$CATALINA_HOME/conf/server.xml`. Set the server attribute value to anything except a blank string.

Default Value:

The default value is `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.5 Disable client facing Stack Traces (Automated)

Profile Applicability:

- Level 1

Description:

When a runtime error occurs during request processing, Apache Tomcat will display debugging information to the requestor. It is recommended that such debug information be withheld from the requestor.

Rationale:

Debugging information, such as that found in call stacks, often contains sensitive information which may be useful to an attacker. By preventing Tomcat from providing this information, the risk of leaking sensitive information to a potential attacker is reduced.

Audit:

Perform the following to determine if Tomcat is configured to prevent sending debug information to the requestor

1. Ensure an `<error-page>` element is defined in `$CATALINA_HOME/conf/web.xml`.
2. Ensure the `<error-page>` element has an `<exception-type>` child element with a value of `java.lang.Throwable`.
3. Ensure the `<error-page>` element has a `<location>` child element.

Note: Perform the above for each application hosted within Tomcat. Per application instances of `web.xml` can be found at `$CATALINA_HOME/webapps/<app_name>/WEB-INF/web.xml`.

Remediation:

Perform the following to prevent Tomcat from providing debug information to the requestor during runtime errors:

1. Create a web page that contains the logic or message you wish to invoke when encountering a runtime error. For example purposes, assume this page is located at `/error.jsp`.
2. Add a child element, `<error-page>`, to the `<web-app>` element, in the `$CATALINA_HOME/conf/web.xml` file.
3. Add a child element, `<exception-type>`, to the `<error-page>` element. Set the value of the `<exception-type>` element to `java.lang.Throwable`.

4. Add a child element `<location>` to the `<error-page>` element. Set the value of the `<location>` element to the location of page created in step 1.

The resulting entry will look as follows:

```
<error-page>
 <exception-type>java.lang.Throwable</exception-type>
 <location>/error.jsp</location>
</error-page>
```

Default Value:

Tomcat's default configuration does not include an `<error-page>` element in `$CATALINA_HOME/conf/web.xml`. Therefore, Tomcat will provide debug information to the requestor by default.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/api/org/apache/tomcat/util/descriptor/web/ErrorPage.html

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.6 Turn off TRACE (Automated)

Profile Applicability:

- Level 1

Description:

The HTTP `TRACE` verb provides debugging and diagnostics information for a given request.

Rationale:

Diagnostic information, such as that found in the response to a `TRACE` request, often contains sensitive information which may useful to an attacker. By preventing Tomcat from providing this information, the risk of leaking sensitive information to a potential attacker is reduced.

Audit:

Perform the following to determine if the server platform, as advertised in the HTTP Server header, has been changed:

1. Locate all `Connector` elements in `$CATALINA_HOME/conf/server.xml`.
2. Ensure each `Connector` does not have an `allowTrace` attribute or, if present, the `allowTrace` attribute is **NOT** set `true`.

Perform the following for each application hosted within Tomcat with a `web-app` root element in the `web.xml`:

1. Locate each application instance of `web.xml` in `$CATALINA_HOME/webapps/<app_name>/WEB-INF/web.xml`.
2. Ensure a `security-constraint/web-resource-collection` exists the child value pairings:
    1. `web-resource-name` with a value of `restricted methods`.
    2. `url-pattern` with a value of `/*`.
    3. `http-method` with a value of `TRACE`.

Remediation:

Perform the following to prevent Tomcat from accepting a TRACE request:

1. Set the `allowTrace` attribute for each `Connector` specified in `$CATALINA_HOME/conf/server.xml` to `false`.

```
<Connector ... allowTrace="false" />
```

Alternatively, ensure the `allowTrace` attribute is absent from each `Connector` specified in `$CATALINA_HOME/conf/server.xml`.

2. Add the following as a child of the `web-app` root element, if present, in each web applications `web.xml`:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>restricted methods</web-resource-name>
        ...
        <http-method>TRACE</http-method>
        ...
    </web-resource-collection>
    ...
</security-constraint>
```

Default Value:

Tomcat does not allow the `TRACE` HTTP verb by default. Tomcat will only allow `TRACE` if the `allowTrace` attribute is present and set to `true`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

2.7 Ensure Sever Header is Modified To Prevent Information Disclosure (Automated)

Profile Applicability:

- Level 2

Description:

The server header is a vanity header developed to help identify the underlying technology in a server for troubleshooting and identification. This header is unnecessary and could be used to target your website for exploitation. Tomcat does not provide the ability to remove the server header, however, it does provide the ability to modify the header.

Rationale:

The server header may specify the underlying technology used by an application. Attackers are able to conduct reconnaissance on a website using these response headers. This header could be used to target attacks for specific known vulnerabilities associated with the underlying technology. Removing this header will prevent targeting of your application for specific exploits by non-determined attackers.

While this is not the only way to fingerprint a site through the response headers, it makes it harder and prevents some potential attackers from targeting your website.

Audit:

In `$CATALINA_HOME/conf/server.xml`, check for the server directive as shown below. If the directive is not present or the directive specifies something revealing on the underlying infrastructure then the server header should be changed.

```
<Connector port="8443" server=”Apache" redirectPort="8080" />
```

Remediation:

In `$CATALINA_HOME/conf/server.xml`, add the server directive to the connector as shown below replacing apache with the text of your choosing. This text should not help in identifying the server.

```
<Connector port="8443" server=”I am a teapot" redirectPort="8080" />
```

Scripted:

- If you do not have the header defined:

```
sed -ir 's/Connector/Connector server=”I am a teapot"/g'
$CATALINA_HOME/conf/server.xml
```

- If you already have a header but it is still revealing

```
sed -ir 's/server="[^"]*"/server="I Am A Teapot"/g'
$CATALINA_HOME/conf/server.xml
```

Default Value:

The default value is `Apache-Coyote/1.1`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html#server.xml
2. https://stackoverflow.com/questions/52637285/replacing-server-header-in-tomcat-with-sed

CIS Controls:

Version 7

13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization

Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.

3 Protect the Shutdown Port

Tomcat listens on TCP port `8005` to accept shutdown requests. By connecting to this port and sending the `SHUTDOWN` command, all applications within Tomcat are halted.

3.1 Set a nondeterministic Shutdown command value (Automated)

Profile Applicability:

- Level 1

Description:

Tomcat listens on TCP port `8005` to accept shutdown requests. By connecting to this port and sending the `SHUTDOWN` command, all applications within Tomcat are halted. The shutdown port is not exposed to the network as it is bound to the loopback interface. It is recommended that a nondeterministic value be set for the shutdown attribute in `$CATALINA_HOME/conf/server.xml`.

Rationale:

Setting the shutdown attribute to a nondeterministic value will prevent malicious local users from shutting down Tomcat.

Audit:

Perform the following to determine if the shutdown port is configured to use the default `shutdown` command:

Ensure the `shutdown` attribute in `$CATALINA_HOME/conf/server.xml` is not set to `SHUTDOWN`.

```
$ cd $CATALINA_HOME/conf
$ grep 'shutdown[[:space:]]*=[[:space:]]*"SHUTDOWN"' server.xml
```

Remediation:

Perform the following to set a nondeterministic value for the shutdown attribute.

Update the `shutdown` attribute in `$CATALINA_HOME/conf/server.xml` as follows:

```
<Server port="8005" shutdown="NONDETERMINISTICVALUE">
```

Note: `NONDETERMINISTICVALUE` should be replaced with a sequence of random characters.

Default Value:

The default value for the `shutdown` attribute is `SHUTDOWN`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/server.html

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

3.2 Disable the Shutdown port (Automated)

Profile Applicability:

- Level 2

Description:

Tomcat listens on TCP port `8005` to accept shutdown requests. By connecting to this port and sending the `SHUTDOWN` command, all applications within Tomcat are halted. The shutdown port is not exposed to the network as it is bound to the loopback interface. If this functionality is not used, it is recommended that the shutdown port be disabled.

Rationale:

Disabling the Shutdown port will eliminate the risk of malicious local entities using the shutdown command to disable the Tomcat server.

Audit:

Perform the following to determine if the shutdown port has been disabled:

Ensure the port attribute in `$CATALINA_HOME/conf/server.xml` is set to `-1`.

```
$ cd $CATALINA_HOME/conf/
$ grep '<Server[[:space:]]\+[^>]*port[[:space:]]*=[[:space:]]*"-1"'
server.xml
```

Remediation:

Set the port to `-1` in the `$CATALINA_HOME/conf/server.xml` to disable the shutdown port:

```
<Server port="-1" shutdown="SHUTDOWN">
```

Default Value:

The shutdown port is enabled on TCP port `8005`, bound to the loopback address.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/server.html

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

4 Protect Tomcat Configurations

The security of processes and data which traverse or depend on Tomcat may become compromised if the Tomcat configurations are not secured.

4.1 Restrict access to $CATALINA_HOME (Automated)

Profile Applicability:

- Level 1

Description:

`$CATALINA_HOME` is the environment variable which holds the path to the root Tomcat directory. It is important to protect access to this in order to protect the Tomcat binaries and libraries from unauthorized modification. It is recommended that the ownership of `$CATALINA_HOME` be `tomcat_admin:tomcat`. It is also recommended that the permission on `$CATALINA_HOME` block read, write, and execute for the world (`o-rwx`) and block write access to the group (`g-w`).

Rationale:

The security of processes and data which traverse or depend on Tomcat may become compromised if the `$CATALINA_HOME` is not secured.

Audit:

Perform the following to ensure the permission on the `$CATALINA_HOME` directory prevent unauthorized modification.

```
$ cd $CATALINA_HOME
$ find . -follow -maxdepth 0 \( -perm /o+rwx,g=w -o ! -user tomcat_admin -o !
-group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to establish the recommended state:

1. Set the ownership of the `$CATALINA_HOME` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin.tomcat $CATALINA_HOME
```

2. Remove write permissions for the group and read, write, and execute permissions for the world

```
# chmod g-w,o-rwx $CATALINA_HOME
```

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.2 Restrict access to $CATALINA_BASE (Automated)

Profile Applicability:

- Level 1

Description:

`$CATALINA_BASE` is the environment variable that specifies the base directory which most relative paths are resolved. `$CATALINA_BASE` is usually used when there are multiple instances of Tomcat running. It is important to protect access to this in order to protect the Tomcat-related binaries and libraries from unauthorized modification. It is recommended that the ownership of `$CATALINA_BASE` be `tomcat_admin:tomcat`. It is also recommended that the permission on `$CATALINA_BASE` block read, write, and execute for the world (`o-rwx`) and block write access to the group (`g-w`).

Rationale:

The security of processes and data which traverse or depend on Tomcat may become compromised if the `$CATALINA_BASE` is not secured.

Audit:

Perform the following to ensure the permission on the `$CATALINA_BASE` directory prevent unauthorized modification.

```
$ cd $CATALINA_BASE
$ find . -follow -maxdepth 0 \( -perm /o+rwx,g=w -o ! -user tomcat_admin -o !
-group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to establish the recommended state:

1. Set the ownership of the `$CATALINA_BASE` to tomcat_admin:tomcat.

```
# chown tomcat_admin.tomcat $CATALINA_BASE
```

2. Remove write permissions for the group and read, write, and execute permissions for the world

```
# chmod g-w,o-rwx $CATALINA_BASE
```

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.3 Restrict access to Tomcat configuration directory (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat `$CATALINA_HOME/conf` directory contains Tomcat configuration files. It is recommended that the ownership of this directory be `tomcat_admin:tomcat`. It is also recommended that the permissions on this directory deny read, write, and execute for the world (`o-rwx`) and deny write access to the group (`g-w`).

Rationale:

Restricting access to these directories will prevent local users from maliciously or inadvertently altering Tomcat's configuration.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf` are securely configured.

```
# cd $CATALINA_HOME/conf
# find . -maxdepth 0 \( -perm /o+rwx,g=w -o ! -user tomcat_admin -o ! -group
tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to Tomcat configuration files:

1. Set the ownership of the `$CATALINA_HOME/conf` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf
```

2. Remove write permissions for the group and read, write, and execute permissions for the world.

```
# chmod g-w,o-rwx $CATALINA_HOME/conf
```

Default Value:

The default permissions of the top-level directories are `770`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.4 Restrict access to Tomcat logs directory (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat `$CATALINA_HOME/logs` directory contains Tomcat logs. It is recommended that the ownership of this directory be `tomcat_admin:tomcat`. It is also recommended that the permissions on this directory deny read, write, and execute for the world (`o-rwx`).

Rationale:

Restricting access to these directories will prevent local users from maliciously or inadvertently altering Tomcat's logs.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/logs` are securely configured.

```
# cd $CATALINA_HOME
# find logs -follow -maxdepth 0 \( -perm /o+rwx -o ! -user tomcat_admin -o !
-group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to Tomcat log files:

1. Set the ownership of the `$CATALINA_HOME/logs` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/logs
```

2. Remove read, write, and execute permissions for the world

```
# chmod o-rwx $CATALINA_HOME/logs
```

Default Value:

The default permissions of the top-level directories are `770`.

CIS Controls:

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.5 Restrict access to Tomcat temp directory (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat `$CATALINA_HOME/temp` directory is used by Tomcat to persist temporary information to disk. It is recommended that the ownership of this directory be `tomcat_admin:tomcat`. It is also recommended that the permissions on this directory deny read, write, and execute for the world (`o-rwx`).

Rationale:

Restricting access to these directories will prevent local users from maliciously or inadvertently affecting the integrity of Tomcat processes.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/temp` are securely configured.

```
# cd $CATALINA_HOME
# find temp -follow -maxdepth 0 \( -perm /o+rwx -o ! -user tomcat_admin -o !
-group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to Tomcat temp directory:

1. Set the ownership of the `$CATALINA_HOME/temp` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/temp
```

2. Remove read, write, and execute permissions for the world

```
# chmod o-rwx $CATALINA_HOME/temp
```

Default Value:

The default permissions of the top-level directories are `770`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.6 Restrict access to Tomcat binaries directory (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat `$CATALINA_HOME/bin` directory contains executes that are part of the Tomcat run-time. It is recommended that the ownership of this directory be `tomcat_admin:tomcat`. It is also recommended that the permission on this directory deny read, write, and execute for the world (`o-rwx`) and deny write access to the group (`g-w`).

Rationale:

Restricting access to these directories will prevent local users from maliciously or inadvertently affecting the integrity of Tomcat processes.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/bin` are securely configured.

```
# cd $CATALINA_HOME
# find bin -follow -maxdepth 0 \( -perm /o+rwx,g=w -o ! -user tomcat_admin -o
! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to Tomcat bin directory:

1. Set the ownership of the `$CATALINA_HOME/bin` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/bin
```

2. Remove read, write, and execute permissions for the world

```
# chmod g-w,o-rwx $CATALINA_HOME/bin
```

Default Value:

The default permissions of the top-level directories are `770`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.7 Restrict access to Tomcat web application directory (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat `$CATALINA_HOME/webapps` directory contains web applications that are deployed through Tomcat. It is recommended that the ownership of this directory be `tomcat_admin:tomcat`. It is also recommended that the permission on this directory eny read, write, and execute for the world (`o-rwx`) and deny write access to the group (`g-w`).

Rationale:

Restricting access to these directories will prevent local users from maliciously or inadvertently affecting the integrity of web applications.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/webapps` are securely configured.

```
# cd $CATALINA_HOME
# find webapps -follow -maxdepth 0 \( -perm /o+rwx,g=w -o ! -user
tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to Tomcat webapps directory:

1. Set the ownership of the `$CATALINA_HOME/webapps` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/webapps
```

2. Remove read, write, and execute permissions for the world.

```
# chmod g-w,o-rwx $CATALINA_HOME/webapps
```

Default Value:

The default permissions of the top-level directories are `770`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.8 Restrict access to Tomcat catalina.properties (Automated)

Profile Applicability:

- Level 1

Description:

`catalina.properties` is a Java properties file which contains settings for Tomcat including class loader information, security package lists, and performance properties. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/catalina.properties` are securely configured.

```
# cd $CATALINA_HOME/conf/
# find catalina.properties -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o !
-user tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `catalina.properties`:

1. Set the ownership of the `$CATALINA_HOME/conf/catalina.properties` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/catalina.properties
```

2. Set the permissions of the `$CATALINA_HOME/conf/catalina.properties` to `600`

```
# chmod 600 $CATALINA_HOME/conf/catalina.properties
```

Default Value:

The default permissions of the top-level directories are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.9 Restrict access to Tomcat catalina.policy (Automated)

Profile Applicability:

- Level 1

Description:

The `catalina.policy` file is used to configure security policies for Tomcat. It is recommended that access to this file has the proper permissions to properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/catalina.policy` are securely configured.

```
# cd $CATALINA_HOME/conf/
# find catalina.policy -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o ! -
user tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `$CATALINA_HOME/conf/catalina.policy`.

1. Set the owner and group owner of the contents of `$CATALINA_HOME/conf/catalina.policy` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/catalina.policy
```

2. Set the permissions of the `$CATALINA_HOME/conf/catalina.policy` file to `600`.

```
# chmod 600 $CATALINA_HOME/conf/catalina.policy
```

Default Value:

The default permissions of `catalina.policy` are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.10 Restrict access to Tomcat context.xml (Automated)

Profile Applicability:

- Level 1

Description:

The `context.xml` file is loaded by all web applications and sets certain configuration options. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/context.xml` care securely configured.

```
# cd $CATALINA_HOME/conf
# find context.xml -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o ! -user
tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `context.xml`:

1. Set the ownership of the `$CATALINA_HOME/conf/context.xml` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/context.xml
```

2. Set the permissions for the `$CATALINA_HOME/conf/context.xml` to `600`.

```
# chmod 600 $CATALINA_HOME/conf/context.xml
```

Default Value:

The default permissions of `context.xml` are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.11 Restrict access to Tomcat logging.properties (Automated)

Profile Applicability:

- Level 1

Description:

`logging.properties` is a Tomcat files which specifies the logging configuration. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/logging.properties` care securely configured.

```
# cd $CATALINA_HOME/conf/
# find logging.properties -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o !
-user tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `logging.properties`:

1. Set the ownership of the `$CATALINA_HOME/conf/logging.properties` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/logging.properties
```

2. Set the permissions for the `$CATALINA_HOME/conf/logging.properties` file to `600`.

```
# chmod 600 $CATALINA_HOME/conf/logging.properties
```

Default Value:

The default permissions are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.12 Restrict access to Tomcat server.xml (Automated)

Profile Applicability:

- Level 1

Description:

`server.xml` contains Tomcat servlet definitions and configurations. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/server.xml` care securely configured.

```
# cd $CATALINA_HOME/conf/
# find server.xml -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o ! -user
tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `server.xml`:

1. Set the ownership of the `$CATALINA_HOME/conf/server.xml` to `tomcat_admin:tomcat`.

   ```
   # chown tomcat_admin:tomcat $CATALINA_HOME/conf/server.xml
   ```

2. Set the permissions of the `$CATALINA_HOME/conf/server.xml` to `600`

   ```
   # chmod 600 $CATALINA_HOME/conf/server.xml
   ```

Default Value:

The default permissions of the top-level directories are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.13 Restrict access to Tomcat tomcat-users.xml (Automated)

Profile Applicability:

- Level 1

Description:

tomcat-users.xml contains authentication information for Tomcat applications. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/tomcat-users.xml` care securely configured.

```
# cd $CATALINA_HOME/conf/
# find tomcat-users.xml -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o ! -
user tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `tomcat-users.xml`:

1. Set the ownership of the `$CATALINA_HOME/conf/tomcat-users.xml` to `tomcat_admin:tomcat`.

   ```
   # chown tomcat_admin:tomcat $CATALINA_HOME/conf/tomcat-users.xml
   ```

2. Set the permissions of the `$CATALINA_HOME/conf/tomcat-users.xml` to `600`

   ```
   # chmod 600 $CATALINA_HOME/conf/tomcat-users.xml
   ```

Default Value:

The default permissions of the top-level directories are `600`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.14 Restrict access to Tomcat web.xml (Automated)

Profile Applicability:

- Level 1

Description:

The Tomcat configuration file `web.xml` stores application configuration settings. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/web.xml` care securely configured.

```
# cd $CATALINA_HOME/conf/
# find web.xml -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+wx -o ! -user
tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `web.xml`:

1. Set the ownership of the `$CATALINA_HOME/conf/web.xml` to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/web.xml
```

2. Set the permissions for the `$CATALINA_HOME/conf/web.xml` file to `400`.

```
# chmod 400 $CATALINA_HOME/conf/web.xml
```

Default Value:

The default permissions of `web.xml` are `400`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4.15 Restrict access to jaspic-providers.xml (Automated)

Profile Applicability:

- Level 1

Description:

Tomcat implements JASPIC 1.1 Maintenance Release B (JSR 196). The implementation is primarily intended to enable the integration of 3rd party JASPIC authentication implementations with Tomcat.

JASPIC may be configured dynamically by an application or statically via the `$CATALINA_BASE/conf/jaspic-providers.xml` configuration file. It is recommended that access to this file properly protect from unauthorized changes.

Rationale:

Restricting access to this file will prevent local users from maliciously or inadvertently altering Tomcat's security policy.

Audit:

Perform the following to determine if the ownership and permissions on `$CATALINA_HOME/conf/jaspic-providers.xml` are securely configured.

```
# cd $CATALINA_HOME/conf/
# find jaspic-providers.xml -follow -maxdepth 0 \( -perm /o+rwx,g+rwx,u+x -o
! -user tomcat_admin -o ! -group tomcat \) -ls
```

The above command should not produce any output.

Remediation:

Perform the following to restrict access to `$CATALINA_HOME/conf/jaspic-providers.xml`.

1. Set the ownership of the `$CATALINA_HOME/conf/jaspic-providers.xml` file to `tomcat_admin:tomcat`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/jaspic-providers.xml
```

2. Set the permissions for the `$CATALINA_HOME/conf/jaspic-providers.xml` file to `600`.

```
# chmod 600 $CATALINA_HOME/conf/jaspic-providers.xml
```

Default Value:

The default permissions of `jaspic-providers.xml` are `600`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/jaspic.html

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5 Configure Realms

A Tomcat realm is a database of usernames and passwords used to identify valid users of web applications.

5.1 Use secure Realms (Automated)

Profile Applicability:

- Level 2

Description:

A realm is a database of usernames and passwords used to identify valid users of web applications. Review the Realms configuration to ensure Tomcat is not configured to use `MemoryRealm`, `JDBCRealm`, `UserDatabaseRealm`, or `JAASRealm`.

Rationale:

According to the Tomcat documentation: `MemoryRealm` and `JDBCRealm` are not designed for production usage and could result in reduced availability; the `UserDatabaseRealm` is not intended for large-scale installations; and the `JAASRealm` is not widely used and therefore the code is not as mature as the other realms.

Audit:

Perform the following to ensure an improper realm is not in use:

```
# grep "Realm className" $CATALINA_HOME/conf/server.xml | grep MemoryRealm
# grep "Realm className" $CATALINA_HOME/conf/server.xml | grep JDBCRealm
# grep "Realm className" $CATALINA_HOME/conf/server.xml | grep
UserDatabaseRealm
# grep "Realm className" $CATALINA_HOME/conf/server.xml | grep JAASRealm
```

The above commands should not emit any output.

Remediation:

Set the Realm `className` setting in `$CATALINA_HOME/conf/server.xml` to one of the appropriate realms.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html
2. https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2 Use LockOut Realms (Automated)

Profile Applicability:

- Level 2

Description:

A `LockOut` realm wraps around standard realms adding the ability to lock a user out after multiple failed logins.

Rationale:

Locking out a user after multiple failed logins slows down attackers from brute forcing logins.

Audit:

Perform the following to check to see if a `LockOut` realm is being used:

```
# grep "LockOutRealm" $CATALINA_HOME/conf/server.xml
```

Remediation:

Create a lockout realm wrapping the main realm similar to the example below:

```
<Realm className="org.apache.catalina.realm.LockOutRealm"
failureCount="3" lockOutTime="600" cacheSize="1000"
cacheRemovalWarningTime="3600">
    <Realm className="org.apache.catalina.realm.DataSourceRealm"
dataSourceName=... />
</Realm>
```

References:

1. http://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html
2. http://tomcat.apache.org/tomcat-9.0-doc/config/realm.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

6 Connector Security

Tomcat Connector Security will ensure applications built on Tomcat have an accurate depiction of the context and security guarantees provided to them.

6.1 Setup Client-cert Authentication (Automated)

Profile Applicability:

- Level 2

Description:

Client-cert authentication requires that each client connecting to the server have a certificate to authenticate. This is generally regarded as stronger authentication than a password as it requires the client to have the certificate and not just know a password.

Rationale:

Certificate based authentication is more secure than password based authentication.

Audit:

Review the Connector configuration in `server.xml` and ensure the `clientAuth` is set to `true` and `certificateVerification` is set to `required`.

Remediation:

In the Connector element, set the `clientAuth` parameter to `true` and the `certificateVerification` to `required`

```
<-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->

<Connector
     port="8443" minProcessors="5" maxProcessors="75"
     enableLookups="true" disableUploadTimeout="true"
     acceptCount="100" debug="0" scheme="https" secure="true";
     clientAuth="true" sslProtocol="TLS"/>
...
<Connector ...>
     <SSLHostConfig
      certificateVerification="required"
     />
```

Default Value:

Not configured

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html
2. https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html

CIS Controls:

Version 7

16.4 Encrypt or Hash all Authentication Credentials

Encrypt or hash with a salt all authentication credentials when stored.

6.2 Ensure SSLEnabled is set to True for Sensitive Connectors (Automated)

Profile Applicability:

- Level 1

Description:

The `SSLEnabled` setting determines if SSL is enabled for a specific Connector. It is recommended that SSL be utilized for any Connector that sends or receives sensitive information, such as authentication credentials or personal information.

Rationale:

The `SSLEnabled` setting ensures SSL is active, which will in-turn ensure the confidentiality and integrity of sensitive information while in transit.

Audit:

Review the `server.xml` and ensure all Connectors sending or receiving sensitive information have the `SSLEnabled` attribute set to `true`.

Remediation:

In `server.xml`, set the `SSLEnabled` attribute to `true` for each Connector that sends or receives sensitive information

```
<Connector
 …
 SSLEnabled="true"
 …
/>
```

Default Value:

`SSLEnabled` is set to `false`.

References:

1. http://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html
2. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

6.3 Ensure scheme is set accurately (Automated)

Profile Applicability:

- Level 1

Description:

The `scheme` attribute is used to indicate to callers of `request.getScheme()` which scheme is in use by the Connector. Ensure the `scheme` attribute is set to `http` for Connectors operating over HTTP. Ensure the scheme attribute is set to `https` for Connectors operating over HTTPS.

Rationale:

Maintaining parity between the scheme in use by the Connector and advertised by `request.getScheme()` will ensure applications built on Tomcat have an accurate depiction of the context and security guarantees provided to them.

Audit:

Review the `server.xml` to ensure the Connector's `scheme` attribute is set to `http` for Connectors operating over HTTP. Also ensure the Connector's `scheme` attribute is set to `https` for Connectors operating over HTTPS.

Remediation:

In `server.xml`, set the Connector's `scheme` attribute to `http` for Connectors operating over HTTP. Set the Connector's `scheme` attribute to `https` for Connectors operating over HTTPS.

```
<Connector
    …
    scheme="https"
    …
/>
```

Default Value:

The `scheme` attribute is set to `http`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html
2. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

6.4 Ensure secure is set to true only for SSL-enabled Connectors (Automated)

Profile Applicability:

- Level 1

Description:

The `secure` attribute is used to convey Connector security status to applications operating over the Connector. This is typically achieved by calling `request.isSecure()`. Ensure the `secure` attribute is only set to `true` for Connectors operating with the `SSLEnabled` attribute set to `true`.

Rationale:

Accurately reporting the security state of the Connector will help ensure that applications built on Tomcat are not unknowingly relying on security controls that are not in place.

Audit:

Review the `server.xml` and ensure the secure attribute is set to `true` for those Connectors having `SSLEnabled` set to `true`. Also, ensure the `secure` attribute is set to `false` for those Connectors having `SSLEnabled` set to `false`.

Remediation:

For each Connector defined in `server.xml`, set the `secure` attribute to `true` for those Connectors having `SSLEnabled` set to `true`. Set the `secure` attribute to `false` for those Connectors having `SSLEnabled` set to `false`.

```
<Connector SSLEnabled="true"
    …
    secure="true"
    …
/>
…
<Connector SSLEnabled="false"
    …
    secure="false"
    …
/>
```

Default Value:

The `secure` attribute is set to `false`.

References:

1. http://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html
2. http://tomcat.apache.org/tomcat-9.0-doc/config/http.html

6.5 Ensure 'sslProtocol' is Configured Correctly for Secure Connectors (Automated)

Profile Applicability:

- Level 1

Description:

The TLSv1.0 and TLSv1.1 protocols should be disabled via the sslProtocol directive. The TLSv1.0 protocol is vulnerable to information disclosure and both protocols lack support for modern cryptographic algorithms including authenticated encryption. The only SSL/TLS protocols which should be allowed are TLSv1.2 and the newer TLSv1.3 protocol.

Rationale:

The TLSv1.0 protocol is vulnerable to the BEAST attack when used in CBC mode (October 2011). Unfortunately, the TLSv1.0 uses CBC modes for all of the block mode ciphers, which only leaves the RC4 streaming cipher which is also weak and is not recommended. Therefore, it is recommended that the TLSv1.0 protocol be disabled. The TLSv1.1 protocol does not support Authenticated Encryption with Associated Data (AEAD) which is designed to simultaneously provide confidentiality, integrity, and authenticity. All major up-to-date browsers support TLSv1.2, and most recent versions of FireFox and Chrome support the newer TLSv1.3 protocol, since 2017.

The NIST SP 800-52r2 guidelines for TLS configuration require that TLS 1.2 is configured with FIPS-based cipher suites be supported by all government TLS servers and clients and requires support of TLS 1.3 by January 1, 2024. A September 2018 IETF draft also depreciates the usage of TLSv1.0 and TLSv1.1 as shown in the references.

As of March 2020 all major browsers will no longer support TLS 1.0 or TLS 1.1.

Audit:

Review `server.xml` to ensure the `sslProtocol` attribute is set to `TLSv1.2`, `TLSv1.3`, or `TLSv1.2+TLSv1.3` for all Connectors having `SSLEnabled` set to `true`.

Remediation:

In `server.xml`, set the `sslProtocol` attribute to `TLSv1.2+TLSv1.3` for Connectors having `SSLEnabled` set to `true`.

```
<Connector
…
sslProtocol="TLSv1.2+TLSv1.3"
…
/>
```

Default Value:

If not specified, the default value of `TLS` will be used.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html
2. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

Additional Information:

Using TLS v1.3 for JSSE is only supported when using a JVM which implements TLSv1.3.

CIS Controls:

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms
Use only standardized and extensively reviewed encryption algorithms.

7 Establish and Protect Logging Facilities

Enable logging and ensure logs are properly protected

7.1 Application specific logging (Automated)

Profile Applicability:

- Level 2

Description:

By default, `java.util.logging` does not provide the capabilities to configure per-web application settings, only per VM. In order to overcome this limitation Tomcat implements JULI as a wrapper for `java.util.logging`. JULI provides additional configuration functionality so you can set each web application with different logging specifications.

Rationale:

Establishing per application logging profiles will help ensure that each application's logging verbosity is set to an appropriate level in order to provide appropriate information when needed for security review.

Audit:

Ensure a `logging.properties` file is located at `$CATALINA_BASE/webapps/<app_name>/WEB-INF/classes`.

Remediation:

Create a `logging.properties` file and place that into your application `WEB-INF/classes` directory.

Note: By default, installing Tomcat places a `logging.properties` file in `$CATALINA_HOME/conf`. This file can be used as base for an application specific logging properties file

Default Value:

By default, per application logging is not configured.

CIS Controls:

Version 7

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

7.2 Specify file handler in logging.properties files (Automated)

Profile Applicability:

- Level 1

Description:

Handlers specify where log messages are sent. Console handlers send log messages to the Java console and File handlers specify logging to a file.

Rationale:

Utilizing file handlers will ensure that security event information is persisted to disk.

Impact:

Configuring logging to debug logging, i.e. `FINEST` or `ALL`, can generate large amounts of information which may impact server performance.

Audit:

Review each application's `logging.properties` file located in the applications `$CATALINA_BASE/webapps/<app_name>/WEB-INF/classes` directory and determine if the file handler properties are set.

```
$ grep handlers \
$ CATALINA_BASE/webapps/<app_name>/WEB-INF/classes/logging.properties
```

In the instance where an application specific logging has not been created, the `logging.properties` file will be located in `$CATALINA_BASE/conf`

```
$ grep handlers $CATALINA_BASE/conf/logging.properties
```

Remediation:

Add the following entries, replacing *<file_handler>* with either `FileHandler` or `AsyncFileHandler`, to your `logging.properties` file if they do not exist.

```
handlers=1catalina.org.apache.juli.<file_handler>,
2localhost.org.apache.juli.<file_handler>,
3manager.org.apache.juli.<file_handler>, 4host-
manager.org.apache.juli.<file_handler>, java.util.logging.ConsoleHandler
```

Ensure logging is not off and set the *<logging_level>* to the desired level (`SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, `FINEST` or `ALL`), for example:

```
org.apache.juli.FileHandler.level=<logging_level>
```

Default Value:

No value for new applications by default.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/logging.html

CIS Controls:

Version 7

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

7.3 Ensure className is set correctly in context.xml (Automated)

Profile Applicability:

- Level 2

Description:

Ensure the `className` attribute is set to `AccessLogValve`. The `className` attribute determines the access log valve to be used for logging.

Rationale:

Some log valves are not suited for production and should not be used. Apache recommends `org.apache.catalina.valves.AccessLogValve`

Audit:

Execute the following to ensure `className` is set properly:

```
# grep org.apache.catalina.valves.AccessLogValve
$CATALINA_BASE/webapps/<app_name>/META-INF/context.xml
```

Remediation:

Add the following statement into the `$CATALINA_BASE/webapps/<app_name>/META-INF/context.xml` file if it does not already exist.

```
<Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="$CATALINA_HOME/logs/"
prefix="access_log"
fileDateFormat="yyyy-MM-dd.HH"
suffix=".log"
pattern="%h %t %H cookie:%{SESSIONID}c request:%{SESSIONID}r %m %U %s %q %r"
/>
```

Default Value:

Does not exist by default.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html

CIS Controls:

Version 7

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

7.4 Ensure directory in context.xml is a secure location (Automated)

Profile Applicability:

- Level 1

Description:

The `directory` attribute tells Tomcat where to store logs. It is recommended that the location referenced by the `directory` attribute be secured.

Rationale:

Securing the log location will help ensure the integrity and confidentiality of web application activity.

Audit:

Review the permissions of the directory specified by the `directory` attribute to ensure the permissions are `o-rwx` and owned by `tomcat_admin:tomcat`:

```
# grep directory context.xml
# ls –ld <log_location>
```

Remediation:

Perform the following:

1. Add the following statement into the `$CATALINA_BASE/webapps/<app_name>/META-INF/context.xml` file if it does not already exist.

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="$CATALINA_HOME/logs/"
prefix="access_log" fileDateFormat="yyyy-MM-dd.HH" suffix=".log"
pattern="%h %t %H cookie:%{SESSIONID}c request:%{SESSIONID}r %m %U %s
%q %r"
/>
```

2. Set the location pointed to by the `directory` attribute to be owned by `tomcat_admin:tomcat` with permissions of `o-rwx`.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/logs
# chmod o-rwx $CATALINA_HOME/logs
```

Default Value:

Does not exist by default

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

7.5 Ensure pattern in context.xml is correct (Automated)

Profile Applicability:

- Level 1

Description:

The pattern setting informs Tomcat what information should be logged per application. At a minimum, enough information to uniquely identify a request, what was requested, where the requested originated from, and when the request occurred should be logged. The following will log the request date and time (`%t`), the requested URL (`%U`), the remote IP address (`%a`), the local IP address (`%A`), the request method (`%m`), the local port (`%p`), query string, if present, (`%q`), and the HTTP status code of the response (`%s`).

```
pattern="%t %U %a %A %m %p %q %s”
```

Rationale:

The level of logging detail prescribed will assist in identifying correlating security events or incidents.

Audit:

Review the pattern settings to ensure it contains all of the variables required by the installation.

```
# grep pattern $CATALINA_BASE/webapps/<app_name>/META-INF/context.xml
```

Remediation:

Add the following statement into the `$CATALINA_BASE/webapps/<app_name>/META-INF/context.xml` file if it does not already exist.

```
<Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="$CATALINA_HOME/logs/" prefix="access_log" fileDateFormat="yyyy-MM-
dd.HH" suffix=".log"
pattern="%h %t %H cookie:%{SESSIONID}c request:%{SESSIONID}r %m %U %s %q %r"
/>
```

Default Value:

Does not exist by default.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html

CIS Controls:

Version 7

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

7.6 Ensure directory in logging.properties is a secure location (Automated)

Profile Applicability:

- Level 1

Description:

The directory attribute tells Tomcat where to store logs. The directory value should be a secure location with restricted access.

Rationale:

Securing the log location will help ensure the integrity and confidentiality of web application activity records.

Audit:

Review the permissions of the directory specified by the directory setting to ensure the permissions are `o-rwx` and owned by `tomcat_admin:tomcat`:
Default:

```
# grep directory $CATALINA_BASE/conf/logging.properties
# ls -ld <log_location>
```

Application specific:

```
# grep directory $CATALINA_BASE/webapps/<app_name>/WEB-
INF/classes/logging.properties
# ls -ld <log_location>
```

Remediation:

Perform the following:

1. Add the following properties into your `logging.properties` file if they do not exist

```
<application_name>.org.apache.juli.AsyncFileHandler.directory=<log_loca
tion>
<application_name>.org.apache.juli.AsyncFileHandler.prefix=<application
_name>
```

2. Set the location pointed to by the directory attribute to be owned by `tomcat_admin:tomcat` with permissions of `o-rwx`.

```
# chown tomcat_admin:tomcat <log_location>
# chmod o-rwx <log_location>
```

Default Value:

The directory location is configured to store logs in `$CATALINA_BASE/logs`.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

8 Configure Catalina Policy

Configuring Catalina Policy prevents web applications from accessing restricted or unknown packages which may be malicious or dangerous to the application.

8.1 Restrict runtime access to sensitive packages (Automated)

Profile Applicability:

- Level 1

Description:

`package.access` grants or revokes access to listed packages during runtime. It is recommended that application access to certain packages be restricted.

Rationale:

Prevent web applications from accessing restricted or unknown packages which may be malicious or dangerous to the application.

Audit:

Review `package.access` list in `$CATALINA_BASE/conf/catalina.properties` to ensure only allowed packages are defined.

Remediation:

Edit `$CATALINA_BASE/conf/catalina.properties` by adding allowed packages to the `package.access` list:

```
package.access =
sun.,org.apache.catalina.,org.apache.coyote.,org.apache.tomcat.,
org.apache.jasper.
```

Default Value:

The default package.access value within `$CATALINA_BASE/conf/catalina.properties` is:

```
package.access =
sun.,org.apache.catalina.,org.apache.coyote.,org.apache.tomcat.,
org.apache.jasper.
```

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

9 Application Deployment

By running Tomcat with the Security Manager, applications are run in a sandbox which can prevent untrusted code from accessing files on the file system.

9.1 Starting Tomcat with Security Manager (Manual)

Profile Applicability:

- Level 1

Description:

Configure applications to run in a sandbox using the Security Manager. The Security Manager restricts what classes Tomcat can access thus protecting your server from mistakes, Trojans, and malicious code.

Rationale:

By running Tomcat with the Security Manager, applications are run in a sandbox which can prevent untrusted code from accessing files on the file system.

Audit:

Review the startup configuration in `/etc/init.d` for Tomcat to ascertain if Tomcat is started with the `-security` option.

Remediation:

The security policies implemented by the Java SecurityManager are configured in the `$CATALINA_HOME/conf/catalina.policy` file. Once you have configured the `catalina.policy` file for use with a SecurityManager, Tomcat can be started with a SecurityManager in place by using the `-security` option:

On Unix:

```
$ $CATALINA_HOME/bin/catalina.sh start -security
```

On Windows:

```
C:\> %CATALINA_HOME%\bin\catalina start -security
```

Default Value:

By default the `-security` option is not utilized.

References:

1. http://tomcat.apache.org/tomcat-9.0-doc/security-manager-howto.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

9.2 Disabling auto deployment of applications (Automated)

Profile Applicability:

- Level 2

Description:

Tomcat allows auto deployment of applications while Tomcat is running. It is recommended that this capability be disabled.

Rationale:

This could allow malicious or untested applications to be deployed and should be disabled.

Audit:

Perform the following to ensure `autoDeploy` is set to `false`.

```
# grep "autoDeploy" $CATALINA_HOME/conf/server.xml
```

Remediation:

In the `$CATALINA_HOME/conf/server.xml` file, change `autoDeploy` to `false`.

```
autoDeploy="false"
```

Default Value:

`autoDeploy` is set to `true`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/deployer-howto.html#Deploying_on_a_running_Tomcat_server
2. https://tomcat.apache.org/tomcat-9.0-doc/config/host.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

9.3 Disable deploy on startup of applications (Automated)

Profile Applicability:

- Level 2

Description:

Tomcat allows auto deployment of applications on startup. It is recommended that this capability be disabled.

Rationale:

This could allow malicious or untested applications to be deployed and should be disabled.

Audit:

Perform the following to ensure `deployOnStartup` is set to `false`.

```
# grep "deployOnStartup" $CATALINA_HOME/conf/server.xml
```

Remediation:

In the `$CATALINA_HOME/conf/server.xml` file, change `deployOnStartup` to `false`.

```
deployOnStartup="false"
```

Default Value:

`deployOnStartup` is set to `true`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/deployer-howto.html#Deployment_on_Tomcat_startup
2. https://tomcat.apache.org/tomcat-9.0-doc/config/host.html#Automatic_Application_Deployment

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10 Miscellaneous Configuration Settings

Store web content on a separate partition from Tomcat system files.

10.1 Ensure Web content directory is on a separate partition from the Tomcat system files (Manual)

Profile Applicability:

- Level 1

Description:

Store web content on a separate partition from Tomcat system files.

Rationale:

The web document directory is where the files which are served to the end user reside. In the past, directory traversal exploits have allowed malicious users to wreak havoc on a web server including executing code, uploading files, and reading sensitive data. Even if you do not have any directory traversal exploits in your server or code at this time, that doesn't mean they won't be introduced in the future. Moving your web document directory onto a different partition will prevent these kinds of attacks from doing more damage to other parts of the file system.

Audit:

Locate the Tomcat system files and web content directory. Review the system partitions and ensure the system files and web content directory are on separate partitions.

```
# df $CATALINA_HOME/webapps
# df $CATALINA_HOME
```

Note: Use the default value `webapps` which is defined by `appBase` attribute here.

Remediation:

Move the web content files to a separate partition from the tomcat system files and update your configuration.

Default Value:

Not Applicable

CIS Controls:

Version 7

2.10 Physically or Logically Segregate High Risk Applications

Physically or logically segregated systems should be used to isolate and run software that is required for business operations but incur higher risk for the organization.

10.2 Restrict access to the web administration application (Automated)

Profile Applicability:

- Level 1

Description:

Limit access to the web administration application to only those with a justified need.

Rationale:

Limiting access to the least privilege will ensure only those people with justified need will have access to a resource. The web administration application should be limited to only administrators.

Audit:

Review `$CATALINA_HOME/conf/server.xml` to determine whether the `RemoteAddrValve` option is uncommented and configured to only allow access to systems required to connect.

Remediation:

For the administration application, edit `$CATALINA_HOME/conf/server.xml` and uncomment the following:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.0\.0\.1"/>
```

Note: The `RemoteAddrValve` property expects a regular expression, therefore periods and other regular expression meta-characters must be escaped.

Default Value:

By default, this configuration is not present.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

10.3 Restrict manager application (Manual)

Profile Applicability:

- Level 2

Description:

Limit access to the manager application to only those with a justified need.

Rationale:

Limiting access to the least privilege will ensure only those people with a justified need will have access to a resource. The manager application should be limited to only administrators.

Audit:

Review `$CATALINA_BASE/conf/<enginename>/<hostname>/manager.xml` to determine if the `RemoteAddrValve` option is uncommented and configured to only allow access to systems required to connect.

Remediation:

For the manager application, edit `$CATALINA_BASE/conf/<enginename>/<hostname>/manager.xml`, and add the bolded line:

```
<Context path="/manager" docBase="${catalina.home}/webapps/manager" debug="0"
privileged="true"><Valve
className="org.apache.catalina.valves.RemoteAddrValve"
allow="127.0.0.1"/></Context>
```

Add hosts, comma separated, which are allowed to access the admin application.

Note: The `RemoteAddrValve` property expects a regular expression, therefore periods and other regular expression meta-characters must be escaped.

Default Value:

By default this setting is not present

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html
2. https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

10.4 Force SSL when accessing the manager application (Manual)

Profile Applicability:

- Level 1

Description:

Use the transport-guarantee attribute to ensure SSL protection when accessing the manager application.

Rationale:

By default when accessing the manager application, login information is sent over the wire in plain text. By setting the transport-guarantee within `web.xml`, SSL is enforced.

Note: This requires SSL to be configured.

Audit:

Ensure `$CATALINA_HOME/webapps/manager/WEB-INF/web.xml` has the `<transport-guarantee>` set to `CONFIDENTIAL`.

```
# grep transport-guarantee $CATALINA_HOME/webapps/manager/WEB-INF/web.xml
```

Remediation:

Set `<transport-guarantee>` to `CONFIDENTIAL` in `$CATALINA_HOME/webapps/manager/WEB-INF/web.xml`:

```
<security-constraint>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Default Value:

By default this configuration is not present.

References:

1. https://www.owasp.org/index.php/Securing_tomcat

CIS Controls:

Version 7

14.4 Encrypt All Sensitive Information in Transit

Encrypt all sensitive information in transit.

10.5 Rename the manager application (Manual)

Profile Applicability:

- Level 2

Description:

The manager application allows administrators to manage Tomcat remotely via a web interface. The manager application should be renamed to make it harder for attackers or automated scripts to locate.

Rationale:

By relocating the manager applications, an attacker will need to guess its location rather than simply navigate to the standard location in order to carry out an attack.

Audit:

Ensure `$CATALINA_HOME/conf/Catalina/localhost/manager.xml`, `$CATALINA_HOME/webapps/host-manager/manager.xml`, `$CATALINA_HOME/webapps/manager` and `$CATALINA_HOME/webapps/manager` do not exist.

Remediation:

Perform the following to rename the manager application:

1. Rename the manager application XML file:

```
# mv $CATALINA_HOME/webapps/host-manager/manager.xml \
$CATALINA_HOME/webapps/host-manager/<new-name>.xml
```

2. Update the docBase attribute within `$CATALINA_HOME/webapps/host-manager/<new-name>.xml` to `$CATALINA_HOME/webapps/<new-name>`
3. Move `$CATALINA_HOME/webapps/manager` to `$CATALINA_HOME/webapps/<new-name>`

```
# mv $CATALINA_HOME/webapps/manager $CATALINA_HOME/webapps/<new-name>
```

Default Value:

The default name of the manager application is `manager` and is located at:

```
$CATALINA_HOME/webapps/manager
```

References:

1. https://www.owasp.org/index.php/Securing_tomcat

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.6 Enable strict servlet Compliance (Manual)

Profile Applicability:

- Level 2

Description:

The `STRICT_SERVLET_COMPLIANCE` influences Tomcat's behavior in several subtle ways. See the References below for the complete list. It is recommended that `STRICT_SERVLET_COMPLIANCE` be set to `true`.

Rationale:

When `STRICT_SERVLET_COMPLIANCE` is set to `true`, Tomcat will always send an HTTP Content-type header when responding to requests. This is significant as the behavior of web browsers is inconsistent in the absence of the Content-type header. Some browsers will attempt to determine the appropriate content-type by sniffing

Impact:

Changing this to `true` will change a number of other default values which is likely to break the majority of systems as some browsers are unable to correctly handle the cookie headers that result from a strict adherence to the specifications. Please refer to the referenced documentation for a complete list of changed values. Defaults, regardless of whether or not they have been changed by setting `org.apache.catalina.STRICT_SERVLET_COMPLIANCE` can always be overridden by explicitly setting the appropriate system property or element attribute.

Audit:

Ensure the `-Dorg.apache.catalina.STRICT_SERVLET_COMPLIANCE=true` parameter is added to the startup script which by default is located at `$CATALINA_HOME/bin/catalina.sh`.

Remediation:

Start Tomcat with strict compliance enabled, add `-Dorg.apache.catalina.STRICT_SERVLET_COMPLIANCE=true` to your startup script.

Default Value:

The default value is `false`.

References:

1. http://tomcat.apache.org/tomcat-9.0-doc/config/systemprops.html

10.7 Turn off session facade recycling (Manual)

Profile Applicability:

- Level 1

Description:

The `RECYCLE_FACADES` can specify if a new facade will be created for each request. If a new facade is not created there is a potential for information leakage from other sessions.

Rationale:

When `RECYCLE_FACADES` is set to `false`, Tomcat will recycle the session facade between requests which may result in information leakage.

Audit:

Ensure `-Dorg.apache.catalina.connector.RECYCLE_FACADES=true` is added to the startup script which, by default, is located at `$CATALINA_HOME/bin/catalina.sh`.

Remediation:

Start Tomcat with `RECYCLE_FACADES` set to `true`. Add `-Dorg.apache.catalina.connector.RECYCLE_FACADES=true` to your startup script.

Default Value:

The default value is `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/systemprops.html
2. https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.8 Do not allow additional path delimiters (Manual)

Profile Applicability:

- Level 2

Description:

Being able to specify different path-delimiters on Tomcat creates the possibility that an attacker can access applications that were previously blocked by a proxy like `mod_proxy`.

Rationale:

Allowing additional path-delimiters allows for an attacker to get to an application or area which was not previously visible.

Audit:

Ensure the `-Dorg.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH=false` and `-Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=false` parameters are added to the startup script which, by default, is located at `$CATALINA_HOME/bin/catalina.sh`.

Remediation:

To start Tomcat with `ALLOW_BACKSLASH` and `ALLOW_ENCODED_SLASH` set to `false`, add `-Dorg.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH=false` and `-Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=false` to your startup script.

Default Value:

By default both parameters are set to `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/systemprops.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.9 Configure connectionTimeout (Automated)

Profile Applicability:

- Level 2

Description:

The `connectionTimeout` setting allows Tomcat to close idle sockets after a specific amount of time to save system resources.

Rationale:

Closing idle sockets reduces system resource usage which can provide better performance and help protect against Denial of Service attacks.

Impact:

This timeout will also apply when reading any request body when `disableUploadTimeout` is not set to `false`.

Audit:

Locate each `connectionTimeout` setting in `$CATALINA_HOME/conf/server.xml` and verify the setting is correct.

```
# grep connectionTimeout $CATALINA_HOME/conf/server.xml
```

Remediation:

Set the `connectionTimeout` for each connector in `$CATALINA_HOME/conf/server.xml` ensure to the optimal number of milliseconds based on hardware resources, load, and number of concurrent connections.

```
connectionTimeout="60000"
```

Default Value:

By default this is set to `60000` (i.e. 60 seconds).

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.10 Configure maxHttpHeaderSize (Automated)

Profile Applicability:

- Level 2

Description:

The `maxHttpHeaderSize` limits the size of the request and response headers defined in bytes.

Rationale:

Limiting the size of the header request can help protect against Denial of Service (DoS) requests.

Audit:

Locate each `maxHttpHeaderSize` setting in `$CATALINA_HOME/conf/server.xml` and verify that they are set to `8192`.

```
# grep maxHttpHeaderSize $CATALINA_HOME/conf/server.xml
```

Remediation:

Set `maxHttpHeaderSize` for each connector in `$CATALINA_HOME/conf/server.xml` to the appropriate setting.

```
maxHttpHeaderSize="8192"
```

Default Value:

The default is `8192` bytes.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.11 Force SSL for all applications (Automated)

Profile Applicability:

- Level 2

Description:

Use the `transport-guarantee` attribute to ensure SSL protection when accessing all applications. This can be overridden on a per application basis in the application configuration.

Rationale:

By default, when accessing applications SSL will be enforced to protect information sent over the network. By using the `transport-guarantee` attribute within `web.xml`, SSL is enforced.

Note: This requires SSL to be configured.

Impact:

If the data protection level is set to `INTEGRAL` or `CONFIDENTIAL`, and the client is not already using SSL, then the client is redirected to the same URI, but using port 443 or the port defined for the `redirectPort` attribute in the `<Connector>` element in `server.xml`.

Audit:

Ensure `$CATALINA_HOME/conf/web.xml` has the `transport-guarantee` attribute set to `CONFIDENTIAL`.

```
# grep transport-guarantee $CATALINA_HOME/conf/web.xml
```

Remediation:

Set `transport-guarantee` to `CONFIDENTIAL` in `$CATALINA_HOME/conf/web.xml`:

```
<user-data-constraint>
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

Default Value:

By default this configuration is not present.

References:

1. https://www.owasp.org/index.php/Securing_tomcat

CIS Controls:

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

10.12 Do not allow symbolic linking (Automated)

Profile Applicability:

- Level 1

Description:

Symbolic links permit one application to include the libraries from another. This allows for re-use of code but also allows for potential security issues when applications include libraries from other applications to which they should not have access.

Rationale:

Allowing symbolic links makes Tomcat susceptible to directory traversal vulnerability. Also, there is a potential that an application could link to another application to which it should not be linking. On case-insensitive operating systems there is also the threat of source code disclosure.

Audit:

Ensure the `allowLinking` attribute in all `context.xml` does not exist or is set to `false`.

```
# find . -name context.xml | xargs grep "allowLinking"
```

Remediation:

In all `context.xml`, set the `allowLinking` attribute to `false`:

```
<Context
...
    <Resources ... allowLinking=”false” />
...
</Context>
```

Default Value:

By default `allowLinking` has a value of `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/resources.html
2. https://tomcat.apache.org/tomcat-9.0-doc/config/context.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.13 Do not run applications as privileged (Automated)

Profile Applicability:

- Level 1

Description:

Setting the `privileged` attribute for an application changes the class loader to the Server class loader instead of the Shared class loader.

Rationale:

Running an application in privileged mode allows an application to load the manager libraries.

Audit:

Ensure the `privileged` attribute in each `context.xml` file does not exist or is set to `false`.

```
# find . -name context.xml | xargs grep "privileged"
```

Remediation:

Set the `privileged` attribute in all `context.xml` files to `false` unless it is required as for the manager application:

```
<Context ... privileged=”false” />
```

Default Value:

By default, `privileged` has a value of `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/context.html

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

10.14 Do not allow cross context requests (Automated)

Profile Applicability:

- Level 1

Description:

Setting `crossContext` to `true` allows for an application to call `ServletConext.getContext` to return a dispatcher for another application.

Rationale:

Allowing `crossContext` creates the possibility for a malicious application to make requests to a restricted application.

Audit:

Ensure the `crossContext` attribute in all `context.xml` does not exist or is set to `false`.

```
# find . -name context.xml | xargs grep "crossContext"
```

Remediation:

Set the `crossContext` attribute in all `context.xml` files to `false`:

```
<Context ... crossContext="false" />
```

Default Value:

By default `crossContext` has a value of `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/context.html

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

10.15 Do not resolve hosts on logging valves (Automated)

Profile Applicability:

- Level 2

Description:

Setting `enableLookups` to `true` on Connector will result in a DNS look-ups to obtain the host name of the remote client before logging any information. This uses additional resources when logging.

Rationale:

Allowing `enableLookups` adds additional overhead to resolve the host name of a remote client which is rarely needed.

Audit:

Ensure Connector elements have the `enableLookups` attribute does not exist or is set to `false`.

```
# grep enableLookups $CATALINA_HOME/conf/server.xml
```

Remediation:

In Connector elements, set the `enableLookups` attribute to `false` or remove it.

```
<Connector ... enableLookups="false" />
```

Default Value:

By default, DNS lookups are disabled.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html
2. https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.16 Enable memory leak listener (Automated)

Profile Applicability:

- Level 1

Description:

The JRE Memory Leak Prevention Listener provides work-arounds for known places where the Java Runtime environment uses the context class loader to load a singleton as this will cause a memory leak if a web application class loader happens to be the context class loader at the time. The work-around is to initialize these singletons when this listener starts as Tomcat's common class loader is the context class loader at that time. It also provides work-arounds for known issues that can result in locked JAR files.

Rationale:

Enabling the JRE Memory Leak Prevention Listener provides work-arounds for preventing memory leaks.

Audit:

Ensure this line is present and not commented out in the `$CATALINA_HOME/conf/server.xml`:

```
<Listener
className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
```

Remediation:

Uncomment the JRE Memory Leak Prevention Listener in `$CATALINA_HOME/conf/server.xml`

```
<Listener
className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
```

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/listeners.html#JRE_Memory_Leak_Prevention_Listener_-_org.apache.catalina.core.JreMemoryLeakPreventionListener

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.17 Setting Security Lifecycle Listener (Automated)

Profile Applicability:

- Level 1

Description:

The Security Lifecycle Listener performs a number of security checks when Tomcat starts and prevents Tomcat from starting if they fail.

Rationale:

When enabled, the **Security Lifecycle Listener** can

- Enforce a blacklist of OS users that must not be used to start Tomcat.
- Set the least restrictive `umask` before Tomcat start

Audit:

Review `server.xml` to ensure the Security Lifecycle Listener element is uncommented with the `checkedOsUsers` and `minimumUmask` attributes set with expected values.

Remediation:

Uncomment the listener in `$CATALINA_BASE/conf/server.xml`. If the operating system supports `umask` then the line in `$CATALINA_HOME/bin/catalina.sh` that obtains the `umask` also needs to be uncommented.
Within Server elements add:

- `checkedOsUsers`: A comma separated list of OS users that must not be used to start Tomcat. If not specified, the default value of **`root`** is used.
- `minimumUmask`: The least restrictive `umask` that must be configured before Tomcat will start. If not specified, the default value of **`0007`** is used.

```
<Listener className="org.apache.catalina.security.SecurityListener"
checkedOsUsers="alex,bob" minimumUmask="0007" />
```

Default Value:

The Security Lifecycle Listener is not enabled by default. For `checkedOsUsers`, the default value is `root`. For `minimumUmask`, the default value is `0007`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/listeners.html#Security_Lifecycle_Listener_-_org.apache.catalina.security.SecurityListener

CIS Controls:

Version 7

5.1 Establish Secure Configurations

Maintain documented, standard security configuration standards for all authorized operating systems and software.

10.18 Use the logEffectiveWebXml and metadata-complete settings for deploying applications in production (Automated)

Profile Applicability:

- Level 1

Description:

Both fragments and annotations give rise to security concerns. `web.xml` contains a `metadata-complete` attribute on the `web-app` element whose binary value defines whether other sources of metadata should be considered when deploying this web application, this includes annotations on class files (`@WebServlet`, but also `@WebListener`, `@WebFilter`, ...), `web-fragment.xml` as well as classes located in `WEB-INF/classes`. In addition, Tomcat could allow you to log the effective `web.xml`, when an application starts, and the effective `web.xml` is the result of taking the main `web.xml` for your application merging in all the fragments applying all the annotations. By logging that, you are able to review it, and see if that is in fact what you actually want.

Rationale:

Enable `logEffectiveWebXml` will allow you to log the effective `web.xml` and you are able to see if that is in fact what you actually want. Enable `metadata-complete` so that the `web.xml` is the only metadata considered.

Audit:

1. Review each application's `web.xml` file located in the applications `$CATALINA_HOME/webapps/<app_name>/WEB-INF/web.xml` and determine if the metadata-complete property is set.

```
<web-app
...
metadata-complete="true"
...
>
```

2. Review each application's `context.xml` file located in the applications `$CATALINA_HOME/webapps/<app_name>/META-INF/context.xml` and determine if the metadata-complete property is set.

```
<Context
...
logEffectiveWebXml="true"
```

```
...
>
```

Remediation:

- Set the `metadata-complete` value in the `web.xml` in each of the applications to `true`. **Note:** The `metadata-complete` option is not enough to disable all of annotation scanning. If there is a `ServletContainerInitializer` with a `@HandlesTypes` annotation, Tomcat has to scan your application for classes that use annotations or interfaces specified in that annotation.
- Set the `logEffectiveWebXml` value in the `context.xml` in each of the application to `true`.

Default Value:

If `logEffectiveWebXml` and/or `metadata-complete` is/are not specified, the default value is `false`.

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/config/context.html
2. https://alexismp.wordpress.com/2010/07/28/servlet-3-0-fragments-and-web-xml-to-rule-them-all/

CIS Controls:

Version 7

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

10.19 Ensure Manager Application Passwords are Encrypted (Manual)

Profile Applicability:

- Level 1

Description:

Apache Tomcat ships with a Manager Application which requires users with a role of `manager-gui`, `manager-status`, `manager-script`, and/or `manager-jmx` to authenticate. The usernames and passwords to log onto the Manager Application are stored in the `tomcat-users.xml` in plain text by default.

Rationale:

Storing passwords in plain text may allow users with access to read the `tomcat-users.xml` file to obtain the credentials of user who have been assigned roles for the Manager Application. This may allow for accounts to be compromised on Tomcat and elsewhere.

Audit:

Perform the following to determine if password digests are in use:

```
$ grep -i <login-config>[.\n]*<auth-method>DIGEST</auth-method>[.\n]*<realm-
name>UserDatabase</realm-name>[.\n]*</login-config>
$CATALINA_HOME/webapps/manager/WEB-INF/web.xml
```

If a `Realm` exists without a `digest` attribute or without a value for the `digest` attribute, this is a fail.

Remediation:

1. Generate the encrypted password:

   ```
   cd $CATALINA_HOME/bin
   digest.bat -a sha-256 YOURPASSWORD
   ```

   This will return the original password followed by encrypted password:

   ```
   YOURPASSWORD:240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74
   c720a9
   ```

2. Replace the plain text password with the above encrypted password generated above in `CATALINA_HOME/conf/tomcat-user.xml` file as follows.

```
<user username="admin"
password="240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c72
0a9"
roles="manager-gui"/>
```

3. Add the `digest` element as a child to the `login-config` element where the `realm-name` element has a value of `UserDatabase`. For example:

```
<login-config>
   <auth-method>DIGEST</auth-method>
   <realm-name>UserDatabase</realm-name>
</login-config>
```

References:

1. https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html#Configuring_Manager_Application_Access
2. https://tomcat.apache.org/tomcat-9.0-doc/config/realm.html#Memory_Based_Realm_-_org.apache.catalina.realm.MemoryRealm
3. https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html#Digested_Passwords

CIS Controls:

Version 7

14.8 Encrypt Sensitive Information at Rest

Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Remove Extraneous Resources**		
1.1	Remove extraneous files and directories (Manual)	☐	☐
1.2	Disable Unused Connectors (Manual)	☐	☐
2	**Limit Server Platform Information Leaks**		
2.1	Alter the Advertised server.info String (Manual)	☐	☐
2.2	Alter the Advertised server.number String (Manual)	☐	☐
2.3	Alter the Advertised server.built Date (Manual)	☐	☐
2.4	Disable X-Powered-By HTTP Header and Rename the Server Value for all Connectors (Automated)	☐	☐
2.5	Disable client facing Stack Traces (Automated)	☐	☐
2.6	Turn off TRACE (Automated)	☐	☐
2.7	Ensure Sever Header is Modified To Prevent Information Disclosure (Automated)	☐	☐
3	**Protect the Shutdown Port**		
3.1	Set a nondeterministic Shutdown command value (Automated)	☐	☐
3.2	Disable the Shutdown port (Automated)	☐	☐
4	**Protect Tomcat Configurations**		
4.1	Restrict access to $CATALINA_HOME (Automated)	☐	☐
4.2	Restrict access to $CATALINA_BASE (Automated)	☐	☐
4.3	Restrict access to Tomcat configuration directory (Automated)	☐	☐
4.4	Restrict access to Tomcat logs directory (Automated)	☐	☐
4.5	Restrict access to Tomcat temp directory (Automated)	☐	☐
4.6	Restrict access to Tomcat binaries directory (Automated)	☐	☐
4.7	Restrict access to Tomcat web application directory (Automated)	☐	☐
4.8	Restrict access to Tomcat catalina.properties (Automated)	☐	☐
4.9	Restrict access to Tomcat catalina.policy (Automated)	☐	☐
4.10	Restrict access to Tomcat context.xml (Automated)	☐	☐
4.11	Restrict access to Tomcat logging.properties (Automated)	☐	☐
4.12	Restrict access to Tomcat server.xml (Automated)	☐	☐
4.13	Restrict access to Tomcat tomcat-users.xml (Automated)	☐	☐
4.14	Restrict access to Tomcat web.xml (Automated)	☐	☐
4.15	Restrict access to jaspic-providers.xml (Automated)	☐	☐
5	**Configure Realms**		
5.1	Use secure Realms (Automated)	☐	☐

5.2	Use LockOut Realms (Automated)	☐	☐
6	**Connector Security**		
6.1	Setup Client-cert Authentication (Automated)	☐	☐
6.2	Ensure SSLEnabled is set to True for Sensitive Connectors (Automated)	☐	☐
6.3	Ensure scheme is set accurately (Automated)	☐	☐
6.4	Ensure secure is set to true only for SSL-enabled Connectors (Automated)	☐	☐
6.5	Ensure 'sslProtocol' is Configured Correctly for Secure Connectors (Automated)	☐	☐
7	**Establish and Protect Logging Facilities**		
7.1	Application specific logging (Automated)	☐	☐
7.2	Specify file handler in logging.properties files (Automated)	☐	☐
7.3	Ensure className is set correctly in context.xml (Automated)	☐	☐
7.4	Ensure directory in context.xml is a secure location (Automated)	☐	☐
7.5	Ensure pattern in context.xml is correct (Automated)	☐	☐
7.6	Ensure directory in logging.properties is a secure location (Automated)	☐	☐
8	**Configure Catalina Policy**		
8.1	Restrict runtime access to sensitive packages (Automated)	☐	☐
9	**Application Deployment**		
9.1	Starting Tomcat with Security Manager (Manual)	☐	☐
9.2	Disabling auto deployment of applications (Automated)	☐	☐
9.3	Disable deploy on startup of applications (Automated)	☐	☐
10	**Miscellaneous Configuration Settings**		
10.1	Ensure Web content directory is on a separate partition from the Tomcat system files (Manual)	☐	☐
10.2	Restrict access to the web administration application (Automated)	☐	☐
10.3	Restrict manager application (Manual)	☐	☐
10.4	Force SSL when accessing the manager application (Manual)	☐	☐
10.5	Rename the manager application (Manual)	☐	☐
10.6	Enable strict servlet Compliance (Manual)	☐	☐
10.7	Turn off session facade recycling (Manual)	☐	☐
10.8	Do not allow additional path delimiters (Manual)	☐	☐
10.9	Configure connectionTimeout (Automated)	☐	☐
10.10	Configure maxHttpHeaderSize (Automated)	☐	☐
10.11	Force SSL for all applications (Automated)	☐	☐
10.12	Do not allow symbolic linking (Automated)	☐	☐
10.13	Do not run applications as privileged (Automated)	☐	☐
10.14	Do not allow cross context requests (Automated)	☐	☐

10.15	Do not resolve hosts on logging valves (Automated)	☐	☐
10.16	Enable memory leak listener (Automated)	☐	☐
10.17	Setting Security Lifecycle Listener (Automated)	☐	☐
10.18	Use the logEffectiveWebXml and metadata-complete settings for deploying applications in production (Automated)	☐	☐
10.19	Ensure Manager Application Passwords are Encrypted (Manual)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
Sep 25, 2019	1.0.0	#109 10.1R Do not allow custom header st... (Ticket 2390)
Sep 25, 2019	1.0.0	#110 7.1R Configure log file size limit (Ticket 2391)
Sep 30, 2019	1.0.0	Permissions for Tomcat config files (Ticket 9226)
Sep 30, 2019	1.0.0	'Remediation' instructs one to change pe... (Ticket 9228)
Oct 1, 2019	1.0.0	Map Apache Tomcat 9 Benchmark to CIS Con... (Ticket 9232)
Oct 21, 2019	1.0.0	Initial Release
Sep 2, 2020	1.1.0	Review note on Audit Procedure (Ticket 9692)
Sep 2, 2020	1.1.0	Revise profile level due to impact (Ticket 11329)
Sep 2, 2020	1.1.0	Ensure only TLS 1.2 is enabled (Ticket 11152)
Nov 6, 2020	1.1.0	Encrypt admins password (Ticket 7609)
Dec 18, 2020	1.1.0	Planned Update