the CENTER for INTERNET SECURITY

Security Configuration Benchmark For

Apache Tomcat 5.5/6.0

Version 1.0.0
December 12th, 2009

http://cisecurity.org
feedback@cisecurity.org

Terms of Use Agreement

Background.

CIS provides benchmarks, scoring tools, software, data, information, suggestions, ideas, and other services and materials from the CIS website or elsewhere ("**Products**") as a public service to Internet users worldwide. Recommendations contained in the Products ("**Recommendations**") result from a consensus-building process that involves many security experts and are generally generic in nature. The Recommendations are intended to provide helpful information to organizations attempting to evaluate or improve the security of their networks, systems and devices. Proper use of the Recommendations requires careful analysis and adaptation to specific user requirements. The Recommendations are not in any way intended to be a "quick fix" for anyone's information security needs.

No representations, warranties and covenants.

CIS makes no representations, warranties or covenants whatsoever as to (i) the positive or negative effect of the Products or the Recommendations on the operation or the security of any particular network, computer system, network device, software, hardware, or any component of any of the foregoing or (ii) the accuracy, reliability, timeliness or completeness of any Product or Recommendation. CIS is providing the Products and the Recommendations "as is" and "as available" without representations, warranties or covenants of any kind.

User agreements.

By using the Products and/or the Recommendations, I and/or my organization ("**we**") agree and acknowledge that:

No network, system, device, hardware, software or component can be made fully secure;
We are using the Products and the Recommendations solely at our own risk;

We are not compensating CIS to assume any liabilities associated with our use of the Products or the Recommendations, even risks that result from CIS's negligence or failure to perform;

We have the sole responsibility to evaluate the risks and benefits of the Products and Recommendations to us and to adapt the Products and the Recommendations to our particular circumstances and requirements;

Neither CIS, nor any CIS Party (defined below) has any responsibility to make any corrections, updates, upgrades or bug fixes or to notify us if it chooses at it sole option to do so; and

Neither CIS nor any CIS Party has or will have any liability to us whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages (including without limitation loss of profits, loss of sales, loss of or damage to reputation, loss of customers, loss of software, data, information or emails, loss of privacy, loss of use of any computer or other equipment, business interruption, wasted management or other staff resources or claims of any kind against us from third parties) arising out of or in any way connected with our use of or our inability to use any of the Products or Recommendations (even if CIS has been advised of the possibility of such damages), including without limitation any liability associated with infringement of intellectual property, defects, bugs, errors, omissions, viruses, worms, backdoors, Trojan horses or other harmful items.

Grant of limited rights.

CIS hereby grants each user the following rights, but only so long as the user complies with all of the terms of these Agreed Terms of Use:

Except to the extent that we may have received additional authorization pursuant to a written agreement with CIS, each user may download, install and use each of the Products on a single computer;

Each user may print one or more copies of any Product or any component of a Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, provided that all such copies are printed in full and are kept intact, including without limitation the text

of this Agreed Terms of Use in its entirety.

Retention of intellectual property rights; limitations on distribution.

The Products are protected by copyright and other intellectual property laws and by international treaties. We acknowledge and agree that we are not acquiring title to any intellectual property rights in the Products and that full title and all ownership rights to the Products will remain the exclusive property of CIS or CIS Parties. CIS reserves all rights not expressly granted to users in the preceding section entitled "Grant of limited rights." Subject to the paragraph entitled "Special Rules" (which includes a waiver, granted to some classes of CIS Members, of certain limitations in this paragraph), and except as we may have otherwise agreed in a written agreement with CIS, we agree that we will not (i) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any software Product that is not already in the form of source code; (ii) distribute, redistribute, encumber, sell, rent, lease, lend, sublicense, or otherwise transfer or exploit rights to any Product or any component of a Product; (iii) post any Product or any component of a Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device, without regard to whether such mechanism or device is internal or external, (iv) remove or alter trademark, logo, copyright or other proprietary notices, legends, symbols or labels in any Product or any component of a Product; (v) remove these Agreed Terms of Use from, or alter these Agreed Terms of Use as they appear in, any Product or any component of a Product; (vi) use any Product or any component of a Product with any derivative works based directly on a Product or any component of a Product; (vii) use any Product or any component of a Product with other products or applications that are directly and specifically dependent on such Product or any component for any part of their functionality, or (viii) represent or claim a particular level of compliance with a CIS Benchmark, scoring tool or other Product. We will not facilitate or otherwise aid other individuals or entities in any of the activities listed in this paragraph.

We hereby agree to indemnify, defend and hold CIS and all of its officers, directors, members, contributors, employees, authors, developers, agents, affiliates, licensors, information and service providers, software suppliers, hardware suppliers, and all other persons who aided CIS in the creation, development or maintenance of the Products or Recommendations ("**CIS Parties**") harmless from and against any and all liability, losses, costs and expenses (including attorneys' fees and court costs) incurred by CIS or any CIS Party in connection with any claim arising out of any violation by us of the preceding paragraph, including without limitation CIS's right, at our expense, to assume the exclusive defense and control of any matter subject to this indemnification, and in such case, we agree to cooperate with CIS in its defense of such claim. We further agree that all CIS Parties are third-party beneficiaries of our undertakings in these Agreed Terms of Use.

Special rules.

CIS has created and will from time to time create special rules for its members and for other persons and organizations with which CIS has a written contractual relationship. Those special rules will override and supersede these Agreed Terms of Use with respect to the users who are covered by the special rules. CIS hereby grants each CIS Security Consulting or Software Vendor Member and each CIS Organizational User Member, but only so long as such Member remains in good standing with CIS and complies with all of the terms of these Agreed Terms of Use, the right to distribute the Products and Recommendations within such Member's own organization, whether by manual or electronic means. Each such Member acknowledges and agrees that the foregoing grant is subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Choice of law; jurisdiction; venue.

We acknowledge and agree that these Agreed Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland, that any action at law or in equity arising out of or relating to these Agreed Terms of Use shall be filed only in the courts located in the State of Maryland, that we hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action. If any of these Agreed Terms of Use shall be determined to be unlawful, void, or for any reason unenforceable, then such terms shall be deemed severable and shall not affect the validity and enforceability of any remaining provisions. We acknowledge and agree that we have read these Agreed Terms of Use in their entirety, understand them and agree to be bound by them in all respects.

Table of Contents

Overview

This document, *Security Configuration Benchmark for Apache Tomcat 5.5/6.*0, provides prescriptive guidance for establishing a secure configuration posture for Apache Tomcat versions 5.5 – 6.0.20 running on Linux. This guide was tested against Apache Tomcat 5.5 and 6.0.20 as installed by tar packages provided by Apache. To obtain the latest version of this guide, please visit http://cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Consensus Guidance

This guide was created using a consensus review process comprised of volunteer and contract 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 released to the public Internet. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the CIS benchmark. If you are interested in participating in the consensus review process, please send us a note to 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.

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:

Authors
Adam Ely, *TiVo, Inc.*
Raymond Forbes, *Disney Interactive Media Group*

Contributors and Reviews
Harold Cochran, *Talbots, Inc.*
Blake Frantz, *Center for Internet Security*
Mike de Libero*, MDE Development, Inc.*
Jasaun Neff, *NASA*
Alan Tetrault, *OppTek, Inc.*
Mark Thomas
Bedirhan Urgun

Typographic 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

Configuration Levels

This section defines the configuration levels that are associated with each benchmark recommendation. Configuration levels represent increasing levels of security assurance.

Level-I Benchmark settings/actions

Level-I Benchmark recommendations are intended to:

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

Level-II Benchmark settings/actions

Level-II Benchmark recommendations exhibit one or more of the following characteristics:

- may negatively inhibit the utility or performance of the technology
- acts as defense in depth measure

Scoring Status

This section defines the scoring statuses used within this document. The scoring status indicates whether compliance with the given recommendation is discernable in an automated manner.

Scorable

The platform's compliance with the given recommendation can be determined via automated means.

Not Scorable

The platform's compliance with the given recommendation cannot be determined via automated means.

1. Recommendations

1.1 Pre-Installation

In this section we present per-installation considerations for deploying Tomcat securely.

1.1.1 Do not install Tomcat on a multi-use system (Level 2, Not Scorable)

Description:
Servers often expose a variety of services or daemons. It is recommended that the number of services and daemons executing on the Tomcat server be limited to those necessary.

Rationale:
Maintaining a server for a single purpose increases the security of your application and system. The more services which are exposed to an attacker, the more potential vectors an attacker has to exploit the system. Tomcat services should function as application servers only and should not be mixed with other functions.

Remediation:
Leverage the package or services manager for your OS to uninstall or disable unneeded services. On Red Hat systems, the following will disable a given daemon:

```
chkconfig <servicename> off
```

Audit:
Leverage the package or services manager for your OS to uninstall or disable unneeded services. On Redhat systems, the following will produce the current service/daemon list:

```
chkconfig --list
```

Default Value: NA

1.2 Installation

In this section we present various methods of installing Tomcat. We will briefly cover both Windows and UNIX installations. For full installation details and options you should refer to the Tomcat documentation. When deciding upon which Tomcat version to install, be sure to install the latest stable version of the Java Runtime Environment (JRE), Tomcat, and any third party libraries.

1.3 Remove Extraneous Resources

1.3.1 Remove extraneous files and directories (Level 2, Scorable)

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.

Remediation:
Perform the following to remove extraneous resources:

1. The following should yield no output:

```
$ rm -rf $CATALINA_HOME/webapps/js-examples \
    $CATALINA_HOME/webapps/servlet-example \
    $CATALINA_HOME/webapps/webdav \
    $CATALINA_HOME/webapps/tomcat-docs \
    $CATALINA_HOME/webapps/balancer \
    $CATALINA_HOME/webapps/ROOT/admin \
    $CATALINA_HOME/webapps/examples
```

If the Manager application is not utilized, also remove the following resources:

```
$ rm –rf $CATALINA_HOME/server/webapps/host-manager \
     $CATALINA_HOME/server/webapps/manager \
     $CATALINA_HOME/conf/Catalina/localhost/host-manager.xml \
     $CATALINA_HOME/conf/Catalina/localhost/manager.xml
```

Audit:
Perform the following to determine the existence of extraneous resources:

1. List all files extraneous files. The following should yield no output:

```
$ ls -l $CATALINA_HOME/webapps/js-examples \
    $CATALINA_HOME/webapps/servlet-example \
    $CATALINA_HOME/webapps/webdav \
    $CATALINA_HOME/webapps/tomcat-docs \
    $CATALINA_HOME/webapps/balancer \
    $CATALINA_HOME/webapps/ROOT/admin \
    $CATALINA_HOME/webapps/examples
    \
    $CATALINA_HOME/server/webapps/manager \
    $CATALINA_HOME/conf/Catalina/localhost/host-manager.xml \
    $CATALINA_HOME/conf/Catalina/localhost/manager.xml
```

Default Value: Depending on your install method, default extraneous resources will vary.

1.3.2 Disable Unused Connectors (Level 1, Not Scorable)

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 develop connectors from scratch, only enabling exactly what is needed.

Rationale:

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

Remediation:
Perform the following to disable unused Connectors:

1. Within `$CATALINA_HOME/conf/server.xml`, remove or comment 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"/>
```

Audit:
Perform the following to identify configured `Connectors`:

1. 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
```

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

- A non-SSL `Connector` bound to port 8080
- An AJP 1.3 `Connector` bound to port 8009

References:

1. http://tomcat.apache.org/tomcat-6.0-doc/connectors.html

1.4 Limit Server Platform Information Leaks

1.4.1 Alter the Advertised server.info String (Level 2, Scorable)

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 make it harder for attackers to determine which vulnerabilities affect the server platform.

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:

For Tomcat 5.5

```
$ cd $CATALINA_HOME/server/lib
```

For Tomcat 6.0

```
$ 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
```

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

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

For Tomcat 5.5

```
$ cd $CATALINA_HOME/server/lib
```

For Tomcat 6.X

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

Default Value:
The default value for the `server.info` attribute is `Apache Tomcat/<MajorVer>.<MinorVer>`. For example, `Apache Tomcat/5.5`.

References:
1. http://www.owasp.org/index.php/Securing_tomcat

1.4.2 Alter the Advertised server.number String (Level 2, Scorable)

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 make it harder for attackers to determine which vulnerabilities affect the server platform.

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:

   For Tom 5.5:
   ```
   $ cd $CATALINA_HOME/server/lib
   ```

   For Tomcat 6.X:
   ```
   $ 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
   ```

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

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

   For Tomcat 5.5
   ```
   $ cd $CATALINA_HOME/server/lib
   ```

   For Tomcat 6.X
   ```
   $ cd $CATALINA_HOME/lib
   $ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
   $ grep server.number org/apache/catalina/util/ServerInfo.properties
   ```

Default Value:
The default value for the `server.number` attribute is a four part version number, such as `5.5.20.0`.

References:

1. http://techgurulive.com/2009/05/27/how-to-hide-tomcat-version/

1.4.3 Alter the Advertised server.built Date (Level 2, Scorable)

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.

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:

   For Tomcat 5.5
   ```
   $ cd $CATALINA_HOME/server/lib
   ```

   For Tomcat 6.X
   ```
   $ 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=<BuildDate>
   ```

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

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

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

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

   For Tomcat 5.5
   ```
   $ cd $CATALINA_HOME/server/lib
   ```

   For Tomcat 6.X
   ```
   $ cd $CATALINA_HOME/lib
   $ jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties
   ```

```
$ grep server.built 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`.

1.4.4 Disable X-Powered-By HTTP Header and Rename the Server Value for all Connectors (Level 2, Scorable)

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 make it harder for attackers to determine which vulnerabilities affect the server platform.

Remediation:
Perform the following to prevent Tomcat from advertising its presence via the `X-Powered-By` 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.

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` has a `server` attribute and that the `server` attribute does not reflect `Apache Tomcat`. Also, make sure that the `xpoweredBy` attribute is NOT set to `true`.

Default Value:
Tomcat does not advertise the `X-Powered-By` HTTP header by default. Tomcat will only advertise in this manner if the `xpoweredBy` attribute is present and set to `true`.

References:

1. http://tomcat.apache.org/tomcat-5.5-doc/config/http.html

1.4.5 Disable client facing Stack Traces (Level 1, Scorable)

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 that may useful to an attacker. By preventing Tomcat from providing this information, the risk of leaking sensitive information to a potential attacker is reduced.

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 #1.

The resulting entry will look as follows:

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

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`

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. http://tomcat.apache.org/tomcat-5.5-doc/catalina/docs/api/org/apache/catalina/deploy/ErrorPage.html

1.4.6 Turn off TRACE (Level 1, Scorable)

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 that may useful to an attacker. By preventing Tomcat from providing this information, the risk of leaking sensitive information to a potential attacker is reduced.

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

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

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

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

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 a `getTrace` attribute or if the `getTrace` attribute is not set `true`.

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`

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. http://tomcat.apache.org/tomcat-5.5-doc/config/http.html

1.5 Protect the Shutdown Port

1.5.1 Set a nondeterministic Shutdown command value (Level 1, Scorable)

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.

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

1. 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.

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

1. 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
```

Default Value:
The default value for the shutdown attribute is `SHUTDOWN`.

References:

1. http://tomcat.apache.org/tomcat-5.5-doc/config/server.html

1.5.2 Disable the Shutdown port (Level 2, Not Scorable)

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.

Remediation:
Perform the following to disable the Shutdown port.

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

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

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

1. 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
```

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

References:
1. http://tomcat.apache.org/tomcat-5.5-doc/config/server.html

1.6 Protect Tomcat Configurations

1.6.1 Restrict access to $CATALINA_HOME (Level 1, Scorable)

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` prevent read, write, and execute for the world (`o-rwx`) and prevent write access to the group (`g-w`).

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

Remediation:
Perform the following to establish the recommended state:

1. Set the ownership of the $CATALINA_HOME to tomcat_admin.tomcat.
2. Remove read, write, and execute permissions for the world
3. Remove write permissions for the group.

```
# chown tomcat_admin.tomcat $CATALINA_HOME
# chmod g-w,o-rwx $CATALINA_HOME
```

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

```
$ cd $CATALINA_HOME
$ find . -follow -maxdepth 0 \( -perm -o-rwx -o -perm -g-w ! -user
tomcat_admin \) -ls
```

The above command should not emit any output.

1.6.2 Restrict access to $CATALINA_BASE (Level 1, Scorable)

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 is 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 prevent read, write, and execute for the world (o-rwx) and prevent write access to the group (g-w).

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

Remediation:
Perform the following to establish the recommended state:

1. Set the ownership of the $CATALINA_BASE to tomcat_admin.tomcat.
2. Remove read, write, and execute permissions for the world
3. Remove write permissions for the group.

```
# chown tomcat_admin.tomcat $CATALINA_BASE
# chmod g-w,o-rwx $CATALINA_BASE
```

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

```
$ cd $CATALINA_BASE
```

```
$ find . -follow -maxdepth 0 \( -perm –o-rwx -o -perm –g-w ! -user
tomcat_admin \) -ls
```

The above command should not emit any output.

1.6.3 Restrict access to Tomcat configuration directory (Level 1, Scorable)

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 prevent read, write, and execute for the world (`o-rwx`) and prevent 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.

Remediation:
Perform the following to restrict access to Tomcat configuration files:

1. Set the ownership of the `$CATALINA_HOME/conf` to `tomcat_admin:tomcat`.
2. Remove read, write, and execute permissions for the world
3. Remove write permissions for the group.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf
# chmod g-w,o-rwx $CATALINA_HOME/conf
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA_HOME/conf
# find catalina.policy -follow -maxdepth 0 \( -perm –o-rwx -o -perm –g-
w ! -user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 770.

1.6.4 Restrict access to Tomcat logs directory (Level 1, Scorable)

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 prevent 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.

Remediation:
Perform the following to restrict access to Tomcat log files:

1. Set the ownership of the `$CATALINA_HOME/logs` to `tomcat_admin:tomcat`.
2. Remove read, write, and execute permissions for the world

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

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

1. Change to the location of the `$CATALINA_HOME/logs` and execute the following:

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

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 770.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/security-manager-howto.html

1.6.5 Restrict access to Tomcat temp directory (Level 1, Scorable)

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 prevent 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.

Remediation:

Perform the following to restrict access to Tomcat log files:

1. Set the ownership of the `$CATALINA_HOME/logs` to `tomcat_admin:tomcat`.
2. Remove read, write, and execute permissions for the world

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

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

1. Change to the location of the `$CATALINA_HOME/temp` and execute the following:

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

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 770.

1.6.6 Restrict access to Tomcat binaries directory (Level 1, Scorable)

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

Remediation:
Perform the following to restrict access to Tomcat log files:

1. Set the ownership of the `$CATALINA_HOME/logs` to `tomcat_admin:tomcat`.
2. Remove read, write, and execute permissions for the world

```
# chown tomcat_admin:tomcat $CATALINA_HOME/bin
# chmod g-w,o-rwx $CATALINA_HOME/bin
```

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

1. Change to the location of the `$CATALINA_HOME/bin` and execute the following:

```
# cd $CATALINA_HOME
# find bin -follow -maxdepth 0 \( -perm -o-rwx -o -perm -g-w ! -
user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 770.

1.6.7 Restrict access to Tomcat web application directory (Level 1, Scorable)

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

Remediation:
Perform the following to restrict access to Tomcat log files:

1. Set the ownership of the `$CATALINA_HOME/webapps` to `tomcat_admin:tomcat`.
2. Remove read, write, and execute permissions for the world.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/webapps
# chmod g-w,o-rwx $CATALINA_HOME/webapps
```

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

1. Change to the location of the `$CATALINA_HOME/webapps` and execute the following:

```
# cd $CATALINA_HOME
# find webapps -follow -maxdepth 0 \( -perm -o-rwx -o -perm -g-w
! -user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 770.

1.6.8 Restrict access to Tomcat catalina.policy (Level 1, Scorable)

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.

Note: Some Tomcat Linux packages, such as the `tomcat55` Debian package, cause `catalina.policy` to be automatically regenerated, at service restart, from individual policy files located in `/etc/tomcat55/policy.d`. In such scenarios, it is highly recommended that access to the `/etc/tomcat55/policy.d` directory and its contents be similarly restricted.

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

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/` to `tomcat_admin` and `tomcat`, respectively.

```
# chmod 770 $CATALINA_HOME/conf/catalina.policy
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/catalina.policy
```

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

1. Change to the location of the `$CATALINA_HOME/` and execute the following:

```
# cd $CATALINA_HOME/conf/
# find catalina.policy !-follow -maxdepth 0 \( -perm -o+rwx -o -perm -
g+rwx ! -user tomcat_admin -group tomcat -perm /770 \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of `catalina.policy` is 600.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/security-manager-howto.html

1.6.9 Restrict access to Tomcat catalina.properties (Level 1, Scorable)

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

Note: Some Tomcat Linux packages, such as the `tomcat55` Debian package, cause `catalina.policy` to be automatically regenerated, at service restart, from individual policy files located in `/etc/tomcat55/policy.d`. In such scenarios, it is highly recommended that access to the `/etc/tomcat55/policy.d` directory and its contents be similarly restricted.

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

Remediation:
Perform the following to restrict access to `catalina.policy`:

1. Set the ownership of the `$CATALINA_HOME/conf/catalina.policy` to `tomcat_admin:tomcat.`
2. Remove read, write, and execute permissions for the world.
3. Remove write permissions for the group.

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/catalina.properties
# chmod g-w,o-rwx $CATALINA_HOME/conf/catalina.properties
```

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

1. Change to the location of the `$CATALINA_HOME/` and execute the following:

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

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 600.

1.6.10 Restrict access to Tomcat context.xml (Level 1, Scorable)

Description:

The `context.xml` file is loaded by all web applications and sets certain configuration options. 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.

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

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/context.xml
# chmod g-w,o-rwx $CATALINA_HOME/conf/context.xml
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA_HOME/conf
# find context.xml -follow -maxdepth 0 \( -perm -o-rwx -o -perm -g-w ! 
-user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of context.xml are 600.

1.6.11 Restrict access to Tomcat logging.properties (Level 1, Scorable)

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

Note: Some Tomcat Linux packages, such as the `tomcat55` Debian package, cause `catalina.policy` to be automatically regenerated, at service restart, from individual policy files located in `/etc/tomcat55/policy.d`. In such scenarios, it is highly recommended that access to the `/etc/tomcat55/policy.d` directory and its contents be similarly restricted.

Rationale:

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

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

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/logging.properties
# chmod g-w,o-rwx $CATALINA_HOME/conf/logging.properties
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA HOME/conf/
# find logging.properties -follow -maxdepth 0 \( -perm –o-rwx -o
-perm –g-w ! -user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions are 600.

1.6.12 Restrict access to Tomcat server.xml (Level 1, Scorable)

Description:
`server.xml` contains Tomcat servlet definitions and configurations. It is recommended that access to this file has the proper permissions to properly protect from unauthorized changes.

Note: Some Tomcat Linux packages, such as the `tomcat55` Debian package, cause `catalina.policy` to be automatically regenerated, at service restart, from individual policy files located in `/etc/tomcat55/policy.d`. In such scenarios, it is highly recommended that access to the `/etc/tomcat55/policy.d` directory and its contents be similarly restricted.

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

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

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/server.xml
# chmod g-w,o-rwx $CATALINA_HOME/conf/server.xml
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA HOME/conf/
# find server.xml -follow -maxdepth 0 \( -perm –o-rwx -o -perm –
g-w ! -user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 600.

1.6.13 Restrict access to Tomcat tomcat-users.xml (Level 1, Scorable)

Description:
`tomcat-users.xml` contains authentication information for Tomcat applications. It is recommended that access to this file has the proper permissions to properly protect from unauthorized changes.

Note: Some Tomcat Linux packages, such as the `tomcat55` Debian package, cause `catalina.policy` to be automatically regenerated, at service restart, from individual policy files located in `/etc/tomcat55/policy.d`. In such scenarios, it is highly recommended that access to the `/etc/tomcat55/policy.d` directory and its contents be similarly restricted.

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

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

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/tomcat-users.xml
# chmod g-w,o-rwx $CATALINA_HOME/conf/tomcat-users.xml
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA_HOME/conf/
# find tomcat-users.xml -follow -maxdepth 0 \( -perm –o-rwx -o -
perm –g-w ! -user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of the top-level directories is 600.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/security-manager-howto.html

1.6.14 Restrict access to Tomcat web.xml (Level 1, Scorable)

Description:
`web.xml` is a Tomcat configuration file that stores application configuration settings. 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.

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

```
# chown tomcat_admin:tomcat $CATALINA_HOME/conf/web.xml
# chmod g-w,o-rwx $CATALINA_HOME/conf/web.xml
```

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

1. Change to the location of the `$CATALINA_HOME/conf` and execute the following:

```
# cd $CATALINA_HOME/conf/
# find web.xml -follow -maxdepth 0 \( -perm -o-rwx -o -perm -g-w ! -
user tomcat_admin \) -ls
```

Note: If the ownership and permission are set correctly, no output should be displayed when executing the above command.

Default Value:
The default permissions of `web.xml` is 400.

1.7 Configure Realms

1.7.1 Use secure Realms (Level 2, Scorable)

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 configured to use `JDBCRealm`, `DataSourceRealm`, `JNDIRealm`, or `JAASRealm`. Specifically, Tomcat should not utilize `MemoryRealm`.

Rationale:
According to the Tomcat documentation, `MemoryRealm` is not designed for production usage and could result in reduced availability.

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

Audit:
Perform the following to ensure the `MemoryRealm` is not in use:

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

The above command should not emit any output.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html

1.7.2 Use LockOut Realms (Level 2, Scorable)

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.

Remediation:
Create a lockout realm wrapping the main realm like 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>
```

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

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

References:

1. http://eu.apachecon.com/presentation/materials/78/2009-03-26-SecuringApacheTomcat.pdf
2. http://tomcat.apache.org/tomcat-6.0-doc/config/realm.html

1.8 Connector Security

1.8.1 Setup Client-cert Authentication (Level 2, Scorable)

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

Rationale:
Certificate based authentication is more secure than password based authentication.

Remediation:
In the `Connector` element, set the `clientAuth` parameter to `true`.

```
<-- 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"/>
```

Audit:

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

Default Value:
Not configured

References:
1. http://wiki.apache.org/tomcat/SSLWithFORMFallback
2. http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html

1.8.2 Ensure SSLEnabled is set to True for Sensitive Connectors (Level 1, Not Scorable)

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.

Remediation:
Set the `SSLEngine` attribute is set to `on` in the `Listener` node within `server.xml`. Also in server.xml, set the `SSLEnabled` attribute to `true` for each `Connector` that sends or receives sensitive information.

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
```

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

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

Default Value:
`SSLEnabled` is set to `false`.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
2. http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html
3. http://tomcat.apache.org/tomcat-6.0-doc/config/http.html

1.8.3 Ensure scheme is set accurately (Level 1, Scorable)

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 of 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.

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 of HTTPS.

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

Audit:
Review `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.

Default Value:
The `scheme` attribute is set to `http`.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
2. http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html
3. http://tomcat.apache.org/tomcat-6.0-doc/config/http.html

1.8.4 Ensure secure is set to true only for SSL-enabled Connectors (Level 1, Scorable)

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.

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 set to `false` for those `Connectors` having `SSLEnabled` set to `false`.

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

Default Value:
The `secure` attribute is set to `false`.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
2. http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html
3. http://tomcat.apache.org/tomcat-6.0-doc/config/http.html

1.8.5 Ensure sslProtocol is set to TLS for Secure Connectors (Level 1, Scorable)

Description:
The `sslProtocol` setting determines which protocol Tomcat will use to protect traffic. It is recommended that **`sslProtocol`** attribute be set to **`TLS`**.

Rationale:
The TLS protocol does not contain weaknesses that affect other secure transport protocols, such as SSLv1 or SSLv2. Therefore, TLS is leveraged to protect the confidentiality and integrity of data while in transit.

Remediation:
In `server.xml`, set the `sslProtocol` attribute to `TLS` for all `Connectors` having `SSLEngine` set to `on`.

```
<Connector
…
sslProtocol="TLS"
…
/>
```

Audit:
Review `server.xml` to ensure the `sslProtocol` attribute is set to `TLS` for all `Connectors` having `SSLEngine` set to `on`.

Default Value:

If the `sslProtocol` attribute is not set, Tomcat will utilize `TLS`.

References:

1. http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html
2. http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html
3. http://tomcat.apache.org/tomcat-6.0-doc/config/http.html

1.9 Establish and Protect Logging Facilities

Enable logging and ensure logs are properly protected.

1.9.1 Application specific logging (Level 2, Scorable)

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.

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.

Audit:
Ensure a `logging.properties` file is locate at `$CATALINA_BASE\<app_name>\WEB-INF\classes`.

Default Value:
By default, per application logging is not configured.

1.9.2 Specify file handler in logging.properties files (Level 1, Scorable)

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.

Remediation:

Add the following entries to your `logging.properties` file if they do not exist.

```
handlers=org.apache.juli.FileHandler, java.util.logging.ConsoleHandler
```

Ensure logging is not off and set the logging level to the desired level such as:

```
org.apache.juli.FileHandler.level=FINEST
```

Audit:
Review each application's `logging.properties` file located in the applications `$CATALINA_BASE\<app name>\WEB-INF\classes` directory and determine if the file handler properties are set.

```
$ grep handlers \
$CATALINA_BASE\<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
```

Default Value:
No value for new applications by default.

1.9.3 Ensure `className` is set correctly in context.xml (Level 2, Scorable)

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

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

Remediation:
Add the following statement into the `$CATALINA_BASE\<app name>\META-INF\context.xml` file if it does not already exist.

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

Audit:

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

```
# grep org.apache.catalina.valves.FastCommonAccessLogValve context.xml
```

Default Value:
Does not exist by default.

1.9.4 Ensure `directory` in context.xml is a secure location (Level 1, Scorable)

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

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

Remediation:
Perform the following:

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

```
<Valve
    className="org.apache.catalina.valves.FastCommonAccessLogValve"
    directory="$CATALINA_HOME/logs/"
    prefix="access_log"
    fileDateFormat="yyyy-MM-dd.HH"
    suffix=".log"
    pattern="%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
```

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`:

```
# grep directory context.xml
# ls -l <log location>
```

Default Value:
Does not exist by default.

1.9.5 Ensure `pattern` in context.xml is correct (Level 1, Scorable)

Description:
The `pattern` setting informs Tomcat what information should be logged. 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.

Remediation:
Add the following statement into the `$CATALINA_BASE\<app name>\META-INF\context.xml` file if it does not already exist.

```
<Valve
    className="org.apache.catalina.valves.FastCommonAccessLogValve"
    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"
/>
```

Audit:
Review the `pattern` settings to ensure it contains all the variables required by the installation.

```
# grep pattern context.xml
```

Default Value:
Does not exist by default.

Reference:
1. http://tomcat.apache.org/tomcat-5.5-doc/config/valve.html

1.9.6 Ensure `directory` in logging.properties is a secure location (Level 1, Scorable)

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.

Remediation:
Perform the following:

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

```
<application_name>.org.apache.juli.FileHandler.directory=<log_location>
<application_name>.org.apache.juli.FileHandler.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>
```

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`:

```
# grep directory logging.properties
# ls -l <log_location>
```

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

1.9.7 Configure log file size limit (Level 2, Scorable)

Description:
By default, the `logging.properties` file will have no defined limit for the log file size. This is a potential denial of service attack as it would be possible to fill a drive or partition containing the log files.

Rationale:
Establishing a maximum log size that is smaller than the partition size will help mitigate the risk of an attacker maliciously exhausting disk space.

Remediation:
Create the following entry in your `logging.properties` file. This field is specified in bytes.

```
java.util.logging.FileHandler.limit=10000
```

Audit:
Validate the max file limit is not greater than the size of the partition where the log files are stored.

Default Value:
No limit by default.

1.10 Configure Catalina Policy

1.10.1 Restrict runtime access to sensitive packages (Level 1, Scorable)

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.

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
```

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

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
```

1.11 Application Deployment

1.11.1 Starting Tomcat with Security Manager (Level 1, Scorable)

Description:
Configure application 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.

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:

```
$ $CATALINA_HOME/bin/catalina.sh start -security       (Unix)
C:\> %CATALINA_HOME%\bin\catalina start -security      (Windows)
```

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

Default Value:
By default the `-security` option is not utilized.

References:
1. http://tomcat.apache.org/tomcat-5.5-doc/security-manager-howto.html

1.11.2 Disabling auto deployment of applications (Level 2, Scorable)

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.

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

```
autoDeploy="false"
```

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

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

Default Value:
`autoDeploy` is set to `true`

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/deployer-howto.html#Deployment%20on%20Tomcat%20startup

1.11.3 Disable deploy on startup of applications (Level 2, Scorable)

Description:

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

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

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

```
deployOnStartup="false"
```

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

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

Default Value:
`deployOnStartup` is set to `true`

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/deployer-howto.html#Deployment%20on%20Tomcat%20startup

1.12 Miscellaneous Configuration Settings

1.12.1 Ensure Web content directory is on a separate partition from the Tomcat system files (Level 1, Not Scorable)

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

Rationale:
The web document directory is where the files which are severed to the end user reside. In the past, directory traversal exploits have allowed malicious users to play 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 part of the file system.

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

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.

```
# more /etc/init.d/tomcat* | grep $CATALINA_HOME
# more /etc/fstab
```

Default Value:
Not Applicable

1.12.2 Restrict access to the web administration (Level 2, Not Scorable)

Description:
Limit access to the web administration application to only those with a required needed.

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

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.

Audit:
Review `$CATALINA_HOME/conf/server.xml` to ascertain that the `RemoteAddrValve` option is uncommented and configured to only allow access to systems required to connect.

Default Value:
By default, this configuration is not present.

References:
1. http://www.unidata.ucar.edu/Projects/THREDDS/tech/reference/TomcatSecurity.html

1.12.3 Restrict manager application (Level 2, Not Scorable)

Description:
Limit access to the manager application to only those with a required needed.

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

Remediation:

For the manager application, edit `$CATALINA_HOME/conf/Catalina/localhost/webapps/manager.xml` in Tomcat 5.5 and `$CATALINA_HOME/webapps/host-manager/manager.xml` in Tomcat 6.X, and add the bolded line:

```
<Context path="/manager"
docBase="${catalina.home}/server/webapps/manager" debug="0"
privileged="true">

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

    <!-- Link to the user database we will get roles from -->
    <ResourceLink name="users" global="UserDatabase"
    type="org.apache.catalina.UserDatabase"/>
</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.

Audit:
Review `$CATALINA_HOME/conf/Catalina/localhost/webapps/manager.xml` on Tomcat 5.5 and `$CATALINA_HOME/webapps/host-manager/manager.xml` on Tomcat 6.X to ascertain that the `RemoteAddrValve` option is uncommented and configured to only allow access to systems required to connect.

Default Value:
By default this setting is not present.

References:
1. http://www.unidata.ucar.edu/Projects/THREDDS/tech/reference/TomcatSecurity.html

1.12.4 Force SSL when accessing the manager application (Level 1, Scorable)

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 using the `transport-guarantee` attribute within `web.xml`, SSL is enforced.

NOTE: This requires SSL to be configured.

Remediation:
Set `$CATALINA_HOME/webapps/manager/WEB-INF/web.xml` in Tomcat 6.X and in Tomcat 5.5 set `$CATALINA_HOME/server/webapps/manager/WEB-INF/web.xml`:

```
<security-constraint>
```

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

Audit:
Ensure `$CATALINA_HOME/webapps/manager/WEB-INF/web.xml` in Tomcat 6.X and `$CATALINA_HOME/server/webapps/manager/WEB-INF/web.xml` in Tomcat 5.5 has the `<transport-guarantee>` attribute set to `CONFIDENTIAL`.

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

Default Value:
By default this configuration is not present.

References:
1. http://www.owasp.org/index.php/Securing_tomcat

1.12.5 Rename the manager application (Level 2, Scorable)

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:
Obscurity can be helpful when used with other security measures. 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.

Remediation:
Perform the following to rename the manager application:

1. Rename the manager application XML file:

   For Tomcat 5.5
   ```
   # mv $CATALINA_HOME/conf/Catalina/localhost/manager.xml \
   $CATALINA_HOME/conf/Catalina/localhost/new-name.xml
   ```

   For Tomcat 6.X
   ```
   # mv $CATALINA_HOME/webapps/host-manager/manager.xml \
   $CATALINA_HOME/webapps/host-manager/new-name.xml
   ```

2. Update the `docBase` attribute within `$CATALINA_HOME/conf/Catalina/localhost/new-name.xml` (Tomcat 5.5), `$CATALINA_HOME/webapps/host-manager/new-name.xml` (Tomcat 6.X) to `${catalina.home}/server/webapps/new-name`

3. In Tomcat 5.5 move `$CATALINA_HOME/server/webapps/manager` to `$CATALINA_HOME/server/webapps/new-name`
   In Tomcat 6.X move `$CATALINA_HOME/webapps/manager` to `$CATALINA_HOME/webapps/new-name`

   For Tomcat 5.5

```
# mv $CATALINA_HOME/server/webapps/manager \
$CATALINA_HOME/server/webapps/new-name
```

   For Tomcat 6.X

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

Audit:
Ensure `$CATALINA_HOME/conf/Catalina/localhost/manager.xml`, `$CATALINA_HOME/webapps/host-manager/manager.xml`, `$CATALINA_HOME/server/webapps/manager and $CATALINA_HOME/webapps/manager` do not exsist.

Default Value:
The default name of the manager application is "manager" and is located at:

For Tomcat 5.5:

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

For Tomcat 6.X:

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

References:
1. http://www.owasp.org/index.php/Securing_tomcat

1.12.6 Enable strict servlet Compliance (Level 1, Scorable)

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

Remediation:
Start Tomcat with strict compliance enabled. Add the following to your startup script.

```
-Dorg.apache.catalina.STRICT_SERVLET_COMPLIANCE=true
```

Audit:
Ensure the above parameter is added to the start up script which by default is located at `$CATALINA_HOME\bin\catalina.sh`.

Default Value:
By default this configuration parameter is not present.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/config/systemprops.html
2. http://mail-archives.apache.org/mod_mbox/tomcat-dev/200906.mbox/%3C4A315DD0.8070706@apache.org%3E

1.12.7 Turn off session façade recycling (Level 1, Scorable)

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

Rationale:
When `RECYCLE_FACADES` is set to `true`, Tomcat will recycle the session façade between requests. This will allow for information leakage between requests.

Remediation:
Start Tomcat with `RECYCLE_FACADES` set to `false`. Add the following to your startup script.

```
-Dorg.apache.catalina.connector.RECYCLE_FACADES=false
```

Audit:
Ensure the above parameter is added to the start up script which by default is located at `$CATALINA_HOME\bin\catalina.sh`.

Default Value:
By default recycling of facades is set to false.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/config/systemprops.html
2. http://www.jeremythomerson.com/blog/2008/11/apachecon-securing-apache-tomcat-for-your-environment/

1.12.8 Do not allow additional path delimiters (Level 2, Scorable)

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

Rationale:

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

Remediation:
Start Tomcat with `ALLOW_BACKSLASH` set to `false` and `ALLOW_ENCODED_SLASH` set to `false`. Add the following to your startup script.

```
-Dorg.apache.catalina.connector. CoyoteAdapter.ALLOW_BACKSLASH=false
-Dorg.apache.tomcat.util.buf. UDecoder.ALLOW_ENCODED_SLASH=false
```

Audit:
Ensure the above parameters are added to the start up script which by default is located at `$CATALINA_HOME\bin\catalina.sh`.

Default Value:
By default allowing additional parameters is set to false.

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/config/systemprops.html
2. http://www.jeremythomerson.com/blog/2008/11/apachecon-securing-apache-tomcat-for-your-environment/
3. https://www.covalent.net/download/patch2.0/README-ers-3.1.0-patch-tomcat-20070315.txt

1.12.9 Do not allow custom header status messages (Level 2, Scorable)

Description:
Being able to specify custom status messages opens up the possibility for additional headers to be injected. If custom header status messages are required make sure it is only in US-ASCII and does not include any user-supplied data.

Rationale:
Allowing user-supplied data into a header allows the possibility of XSS.

Remediation:
Start Tomcat with `USE_CUSTOM_STATUS_MSG_IN_HEADER` set to `false`. Add the following to your startup script.

```
-Dorg.apache.coyote.USE_CUSTOM_STATUS_MSG_IN_HEADER=false
```

Audit:
Ensure the above parameter is added to the start up script which by default is located at `$CATALINA_HOME\bin\catalina.sh`.

Default Value:
By default allowing custom header status messages is set to false.

References:

1. http://tomcat.apache.org/tomcat-6.0-doc/config/systemprops.html
2. http://www.jeremythomerson.com/blog/2008/11/apachecon-securing-apache-tomcat-for-your-environment/

1.12.10 Configure connectionTimeout (Level 2, Scorable)

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 thus can provide better performance and help protect against Denial of Service attacks.

Remediation:
Within `$CATALINA_HOME/conf/server.xml` ensure each connector is configured to the `connectionTimeout` setting that is optimal based on hardware resources, load, and number of concurrent connections.

```
connectionTimeout="60000"
```

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

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

Default Value:
`connectionTimeout` is set to `60000`

References:

1. http://tomcat.apache.org/connectors-doc/generic_howto/timeouts.html

1.12.11 Configure maxHttpHeaderSize (Level 2, Scorable)

Description:
The `maxHttpHeaderSize` limits the size of the request and response headers defined in bytes. If not specified, the default is 8192 bytes.

Rationale:
Limiting the size of the header request can help protect against Denial of Service requests.

Remediation:
Within `$CATALINA_HOME/conf/server.xml` ensure each connector is configured to the appropriate `maxHttpHeaderSize` setting.

```
maxHttpHeaderSize=”8192”
```

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
```

Default Value:
`maxHttpHeaderSize` is set to `8192`

References:
1. http://tomcat.apache.org/tomcat-6.0-doc/config/http.html

1.12.12 Force SSL for all applications (Level 2, Scorable)

Description:
Use the `transport-guarantee` attribute to ensure SSL protection when accessing all applications. This can be overridden to be disabled 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.

Remediation:
In `$CATALINA_HOME/conf/web.xml`, set the following:

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

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

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

Default Value:
By default this configuration is not present.

References:
1. http://www.owasp.org/index.php/Securing_tomcat

1.12.13 Increase the entropy in session identifiers (Level 2, Scorable)

Description:

Having a server that has deterministic session identifiers can lead to session hi-jacking. Specifying a **`randomClass`** attribute allows for truly random session identifiers.

Rationale:
By default the entropy attribute on session managers uses the string representation of the Manager class name. Leading to a deterministic session identifier.

Remediation:
In `$CATALINA_HOME/conf/context.xml`, set the following:

```
<Manager ... randomClass="java.security.SecureRandom" />
```

Audit:
Ensure `$CATALINA_HOME/conf/context.xml` has the `randomClass` attribute set to `java.security.SecureRandom`.

```
# grep randomClass $CATALINA_HOME/conf/context.xml
```

Default Value:
By default the string representation of the Manager class is used for entropy.

Reference:
1. http://www.jeremythomerson.com/blog/2008/11/apachecon-securing-apache-tomcat-for-your-environment/

1.12.14 Do not allow symbolic linking (Level 1, Scorable)

Description:
Symbolic links allows 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 they should not have access to.

Rationale:
Allowing symbolic links opens up all Tomcat versions prior to 6.0.18 to directory traversal vulnerability. Also there is a potential that an application could link to another application it should not be linking too. On case-insensitive operating systems there is also the threat of source code disclosure.

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

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

Audit:
Ensure all `context.xml` have the `allowLinking` attribute set to `false` or `allowLinking` does not exist.

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

Default Value:
By default **`allowLinking`** has a value of false.

Reference:
1. http://eu.apachecon.com/presentation/materials/78/2009-03-26-SecuringApacheTomcat.pdf
2. http://tomcat.apache.org/tomcat-6.0-doc/config/context.html

1.12.15 Do not run applications as privileged (Level 1, Scorable)

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.

Remediation:
In all `context.xml`, set the `privileged` attribute to false unless it is required like the manager application:

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

Audit:
Ensure all `context.xml` have the `privileged` attribute set to `false` or `privileged` does not exist.

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

Default Value:
By default `privileged` has a value of false.

Reference:
1. http://eu.apachecon.com/presentation/materials/78/2009-03-26-SecuringApacheTomcat.pdf
2. http://tomcat.apache.org/tomcat-6.0-doc/config/context.html

1.12.16 Do not allow cross context requests (Level 1, Scorable)

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.

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

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

Audit:
Ensure all `context.xml` have the `crossContext` attribute set to `false` or `crossContext` does not exist.

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

Default Value:
By default `crossContext` has a value of false.

Reference:

1. http://eu.apachecon.com/presentation/materials/78/2009-03-26-SecuringApacheTomcat.pdf
2. http://tomcat.apache.org/tomcat-6.0-doc/config/context.html

1.12.17 Do not resolve hosts on logging valves (Level 2, Scorable)

Description:
Setting `resolveHosts` to true on logging valves requires a DNS look-up before logging the information. This adds additional resources when logging.

Rationale:
Allowing `resolveHosts` adds additional overhead that is rarely needed.

Remediation:
In all `context.xml` and `server.xml` that have `Valve` nodes, set the `resolveHosts` attribute to false:

```
<Valve ... resolveHosts="false" />
```

Audit:
Ensure all `Valve` nodes have the `resolveHosts` attribute set to `false` or `resolveHosts` does not exist.

```
# find . -name *.xml | xargs grep "resolveHosts"
```

Default Value:
By default `resolveHosts` has a value of false.

Reference:

1. http://eu.apachecon.com/presentation/materials/78/2009-03-26-SecuringApacheTomcat.pdf
2. http://tomcat.apache.org/tomcat-6.0-doc/config/valve.html

Appendix A: Change History

Date	Version	Changes for this version
December 12th , 2009	1.0	Public Release