pentaho®
A Hitachi Group Company

Pentaho and Tomcat Security Best Practices

This page has intentionally been left blank.

Contents

This page has intentionally been left blank.

Overview

We have collected a set of best practice recommendations for you to leverage when using Pentaho with Tomcat as your web application server.

Some of the things discussed here include securing Tomcat by removing unused connectors and default applications, securing cookies, changing the default shutdown command and port, and setting up access logging.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version
Pentaho	6.x, 7.x
Apache Tomcat	7.0, 8.0

Guidelines Note

The guidelines presented in this document are generic in that they were not written with any particular solution in mind. They are considered guidelines and a starting point. These guidelines must be tested and validated within development or QA before being placed into production.

Pentaho and Apache Tomcat

The Pentaho platform is a collection of web applications deployed within Java application servers. For a complete list of servers supported by Pentaho, visit the Components Reference documentation. The tuning and configuration of the application server can impact the performance and security of solutions deployed on the Pentaho platform.

This guide presents best practices for configuring Apache Tomcat to host a Pentaho solution. The suggestions contained within are general concepts and techniques. They are guidelines and may not apply exactly to every solution.

Before implementing any suggestions within this document, make sure to create a backup of the Tomcat configuration files. These are typically found in `$tomcat/conf` directory.

This guide assumes the reader has a working understanding of Tomcat and its configuration files. In particular, an understanding of the `$tomcat/conf/server.xml` is required, with `$tomcat` representing the location of your Tomcat install. The Apache Tomcat documentation has more information on configuring Tomcat.

Securing Tomcat

Pentaho's Tomcat configuration is not optimized for production usage. There are ways to make Tomcat more secure for your production environment. The suggestions below are gathered from research and from working with customers over the years.

Additional information on securing Tomcat, including Secure Sockets Layer (SSL) information, can be found in the Security Considerations of the Tomcat documentation and in the article on enabling SSL in the Pentaho Server in the Pentaho Documentation.

Remove Unused Connectors

A `Connector` defines how client applications access content from Tomcat. As covered in a later section on performance guidelines, Tomcat supports different types of connectors. A single server instance can support multiple connectors.

Any unused connections should be removed from the `$tomcat/server.xml`.

Remove Unwanted and Default Applications

Some versions of Tomcat ship with sample and administrative applications deployed. Evaluate each of the default Tomcat applications as to their suitability for your deployment.

Tomcat suggests removing the applications that are not in use reduces risk from undiscovered vulnerabilities within those applications.

Pentaho does not rely on or require any of these applications, and they can be removed without impacting Pentaho.

Remove Server Banner and X-Powered-By

Tomcat communicates largely through the Hypertext Transfer Protocol (HTTP). Browser tools like FireBug, or slightly web-savvy developers, can analyze the HTTP communication to learn details about the server.

The server banner and `X-Powered-By` are HTTP headers that will identify the server product used and may lead to information leakage vulnerabilities.

The server banner can be set to an empty string or another text to obscure the server application used. The server banner is specified in the **Server** attribute of the connectors configured to provide access to Tomcat. Turning off the `X-Powered-By` header is done by setting the `xpoweredby` attribute to **false**. The `Connector` settings for a server are contained in the `$tomcat/conf/server.xml`. The following example sets the server banner to an empty string:

```
<Connector port="8080"
  protocol="HTTP/1.1" connectionTimeout="20000"
  server="" xpoweredby=”false” redirectPort="8443" />
```

Serve Content on the Desired Interfaces

By default, Tomcat will serve content on all IP addresses configured on the server.

To serve content only on a particular address or set of addresses, set the `address` attribute on the `Connector`.

Add Secure Flag for Cookies

By default, Tomcat manages user sessions through cookies.

It is possible for web user sessions to be stolen or manipulated.

Using secure cookies can add an additional layer of protection. The connectors are specified in the `$tomcat/conf/server.xml`.

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  Server="" secure="true" redirectPort="8443" />
```

Make Sure Cookies are HttpOnly

Some browsers allow client-side scripting to access cookies. Using the HttpOnly setting adds a layer of protection for cookies.

*Enable **HttpOnly** for Pentaho and other applications by adding a context for Pentaho in the `$tomcat/conf/context.xml` file. This setting defaults to **true** on Tomcat 7. However, setting it assures the added security layer is in place.*

```
<Context useHttpOnly="true">...</Context>
```

Set Up Access Logging

A common request is to understand who is accessing Pentaho. This can be done using the auditing functionality within Pentaho. However, it can also be done through Tomcat itself. Tomcat includes the ability to log access and track additional information about the user.

Setting up access logging can degrade performance on busy networks.

Examine the additional detail available through Tomcat's AccessLogValve and decide if the additional logging is useful.

Run Tomcat on a Non-Privileged Account

As a general practice, services should run with the least amount of permission necessary.

Use a separate non-privileged user for Tomcat to protect other services from running, in case of any security hole.

Change the Default SHUTDOWN Command and Port

Tomcat is configured to listen on a configured report for a specified command, as a method of shutting down the service. The default port and command will be widely known, allowing anyone who can access the server and port to shut down the service.

Change this port and the command to guard against such attacks.

The example below from an updated `$tomcat/conf/server.xml` sets the shutdown port to `8999` and the command to `SOMECOMMAND`:

```
<Server port="8999" shutdown="SOMECOMMAND">
```

Replace the Default Error Pages

Errors can always happen; even in the best solutions. Tomcat provides default error pages for errors such as attempting to access an unknown Uniform Resource Identifier (URI). These default error pages contain details about the Tomcat server and version.

Change these pages to protect the information about the server. Some web development is required to create the custom error pages.

The Tomcat Wiki has more information on configuring the custom error pages.

Remove Pentaho Demo and Sample Content

Pentaho provides sample content, a data source, and a sample repository for evaluation and testing purposes only.

Demo and sample content should be removed, before moving to development or production environments.

The Remove Sample Data from the Pentaho Server section in the Pentaho documentation has more information and instructions on removing the demo content and sample data.

Create Secure Passwords

Because Tomcat uses Extensible Markup Language (XML) for defining connections, the password for these connections could potentially be exposed as plain text.

Use the password utility located in `context.xml` or `server.xml` to create secure passwords.

Our best practice document on Securing Connection Passwords for the Pentaho Business Analytics Suite contains more information about securing your passwords.

Set Up a User Repository

Default Pentaho Security is designed for small and medium environments, and does not scale well to enterprise production environments.

Users and roles should be set up in a user repository that has policies regarding password length, strength, and longevity, such as Active Directory.

Linux Only: Use a SandBoxed Root for Tomcat User

Directory traverse attacks can occur if you are running Tomcat in a Linux environment and do not change `root`.

Change your Tomcat root user to create a chroot jail to prevent directory traverse attacks.

Reduce Session Timeouts in web.xml File

Longer timeout sessions can increase the risk of unauthorized access.

Reduce the session timeout for Tomcat in the web.xml file.

```
<session-config>
         <session-timeout>120</session-timeout>
  </session-config>
```

Remove Unused Endpoints from web.xml File

There are some endpoints in the `tomcat/webapps/pentaho/WEB-INF/web.xml` file that may pose a security risk. They should either be commented out or deleted. The following list contains recommendations on handling them on a case-by-case basis:

1. **ProxyTrustingFilter/ProxyTrustingServlet** – These are only needed during Migrator/Import/Export.

   ℹ *Comment out these endpoints until you need to do some kind of migration/import/export. After finishing your task, make sure to comment them out again.*

2. **ViewAction/ServiceAction** – Previously used to invoke action sequences (.xaction). These are no longer relevant, but could possibly present a vulnerability.

   ℹ *Delete the ViewAction and ServiceAction endpoints.*

3. **XMLA** – Used for XMLA (XML for Analysis).

   ℹ *If you are not using XMLA (XML for Analysis) on the server, you should comment out the endpoint.*

4. **GenericServlet** – This endpoint has been deprecated.

   ℹ *The GenericServlet endpoint should be commented out.*

5. **DebugHome** – Used for debugging Mantle.

   ℹ *Comment out the DebugHome endpoint on any production system.*

6. **Carte** – Used for Carte operations.

   ℹ *Comment out this endpoint if you are using only a Penatho Server.*

7. **AuditReport** / **AuditReportList** – These are deprecated entry points.

   ℹ *Delete the ViewAction and ServiceAction endpoints.*

8. **UserService** – Tells the caller how many active HTTP Sessions there are.

   ℹ *An administrator can make use of this, but they may want to add it to the `applicationContext-spring-security.xml` and lock it down.*

9. **Diagnostics** – Uses for server diagnostics.

   ℹ *This should be disabled unless someone is trying to get server diagnostics.*

10. **UploadService** / **PluggableUploadFileServlet** – Comment out these two endpoints.

    ℹ *Verify that Pentaho Metadata Editor (PME) and Pentaho Report Designer (PRD) can still publish after commenting these out.*

Related Information

Here are some links to information that you may find helpful while using this best practices document:

Apache Tomcat Documentation

- Access Log Valve
- Apache Tomcat
- Default Web Applications
- Security Considerations
- Tomcat Wiki FAQ / Connectors
- Tomcat Wiki FAQ / Miscellaneous

Arch Linux Documentation

- Change Root

OWASP Documentation

- HttpOnly

Pentaho Documentation

- Components Reference
- Enabling SSL in the Pentaho Server
- Remove Sample Data from the Pentaho Server

Best Practice Check List

This checklist is designed for you to use while you are thinking about how to secure and tune the performance of Tomcat. The Pentaho Enterprise Architecture Group is here to help you with any questions that arise during your implementation.

Name of the Project: ____

Date of the Review: ____

Name of the Reviewer: ____

Considerations/Think About:	Response	COMMENTS, WHY?
Have you removed unused connectors?	YES ___ NO ___	
Have you removed unwanted and default applications?	YES ___ NO ___	
Did you remove the server banner and X-Powered-By?	YES ___ NO ___	
Have you set up content to serve on the desired interfaces?	YES ___ NO ___	
Added a Secure flag for Cookies?	YES ___ NO ___	
Made sure cookies are HttpOnly?	YES ___ NO ___	
Setup Access Logging?	YES ___ NO ___	
Are you running Tomcat on a Non-Privileged Account?	YES ___ NO ___	
Did you change the default shutdown command and Port?	YES ___ NO ___	
Have you replaced the default error pages?	YES ___ NO ___	
Did you remove Pentaho sample content?	YES ___ NO ___	
Did you create secure passwords?	YES ___ NO ___	
Linux: set up a chroot for Tomcat user?	YES ___ NO ___	
Did you reduce Tomcat session timeouts?	YES ___ NO ___	
Have you removed unused endpoints?	YES ___ NO ___	