Background

Software architectures need to take into account a system’s non-functional requirements, such as performance and security. These become especially important when that system is used widely, both by humans and also other systems. A failure in one system would affect all of the users. An example of a widely used, large-scale system is Apache Tomcat, referred to as simply Tomcat henceforth. From Tomcat’s official website: “The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.”¹ Tomcat is used in a variety of commercial products.²

Tomcat has been around since 2000, and during this time, more than 9 major versions have been released. Each major version comes with a varying degree of architectural evolution and change. In this assignment, your tasks will be related to the evolution of the system from a security perspective.

Tomcat Components and Connectors³

Server

In the Tomcat world, a Server represents the whole container. Tomcat provides a default implementation of the Server interface that is rarely customized by users.

Service

A Service is an intermediate component that lives inside a Server and ties one or more Connectors to exactly one Engine. The Service element is rarely customized by users, as the default implementation, Service interface, is simple and sufficient.

Engine

An Engine represents a request processing pipeline for a specific Service. As a Service may have multiple Connectors, the Engine receives and processes all requests from these

¹ [https://tomcat.apache.org/](https://tomcat.apache.org/)
² [https://cwiki.apache.org/confluence/display/TOMCAT/PoweredBy](https://cwiki.apache.org/confluence/display/TOMCAT/PoweredBy)
³ Different major versions may have more, fewer, or different components.
connectors, handing each response back to the appropriate connector for transmission to the client. The Engine interface may be implemented to supply custom Engines, although this is uncommon.

Note that the Engine may be used for Tomcat server clustering via the jvmRoute parameter. Read the Clustering documentation\(^4\) for more information.

**Host**

A Host is an association of a network name, e.g. www.yourcompany.com, to the Tomcat server. An Engine may contain multiple hosts, and the Host element also supports network aliases such as yourcompany.com and abc.yourcompany.com. Users rarely create custom Hosts because the StandardHost implementation provides significant additional functionality.

**Connector**

A Connector handles communications with the client. There are multiple connectors available in Tomcat. These include the HTTP connector which is used for most HTTP traffic, especially when running Tomcat as a standalone server, and the AJP connector which implements the AJP protocol used when connecting Tomcat to a web server such as Apache HTTPD server. Creating a customized connector is a significant effort.

**Context**

A Context represents a web application. A Host may contain multiple contexts, each with a unique path. The Context interface may be implemented to create custom Contexts, but this is rarely the case because the StandardContext provides significant additional functionality.

**Your tasks**

- Pick one major version of Tomcat from 5.5 to 9. Check the appendix to understand how to get to the architectural documentation of a specific major version.
- Break down the selected version’s architecture by analyzing documentation or other credible resources. Please cite all the resources you use. Some possible resources are listed in Appendix.
- Provide the architecture’s visualization. You should use one of the online solutions that are available for this, such as https://structurizr.com/. You can Google them (e.g., by using a search string such as “software architecture visualization”). Note that several solutions you may come across are sold as commercial products, but many have free versions, which should be more than sufficient for your needs.
- Pick five security architectural vulnerabilities in that version. Instructions on how to find potential architectural vulnerabilities can be found in the Appendix as well as reported and fixed vulnerabilities in each major version of Tomcat.

---

\(^4\) Information about clustering can be found here: [https://tomcat.apache.org/tomcat-8.0-doc/cluster-howto.html](https://tomcat.apache.org/tomcat-8.0-doc/cluster-howto.html) Please make sure to use your own major version’s clustering documentation when trying to understand how this works.
In the links provided in the appendix, you will have access to commits that fixed a particular vulnerability. Analyze the fix in the source code and the commit message. Moreover, check the information the National Vulnerability Database (NVD)\(^5\) has about that particular vulnerability.

- What components in your diagram are involved in this vulnerability?
- Are they architectural because of a wrong architectural decision or a correct decision implemented incorrectly? Refer to the Tomcat documentation to get more information on this.
- Why or why not?
- What can be done to protect Tomcat from such a vulnerability?

- Pick a different version of Tomcat.
- Highlight the architectural changes between the two versions.
- Would the security issues from the five vulnerabilities you originally selected still be there? Why or why not?

\(^5\) https://nvd.nist.gov
Appendices

Appendix A
Tomcat Architecture Documentation For Each Major Version

- https://tomcat.apache.org/tomcat-5.5-doc/architecture/index.html

Appendix B
Tomcat Reports For Security Vulnerability Fixes For Each Major Version

- https://tomcat.apache.org/security-5.html

---

6 Vulnerabilities for 8 and 8.5 have been reported together
Appendix C

Instructions on Finding Architectural Vulnerabilities

Architectural vulnerabilities are those vulnerabilities that stem either from incorrect or missing architectural decisions, or incorrectly implemented architectural decisions.

When you click on one of the links in Appendix B, you will be redirected to a page where Tomcat vulnerabilities fixed in a particular version are presented. For instance, if you click on the link for vulnerabilities fixed in version 9 of Tomcat, one of the first subsections you will see is presented in Figure 1. There you see the ID the vulnerability has been given; in this case, that ID is CVE-2019-10072.

![Figure 1. Subsection of the page listing security vulnerabilities in Tomcat](image)

To see if this vulnerability is architectural, you can visit NVD’s website and search for that vulnerability ID. When you fetch that vulnerability’s page, as you can see in Figure 2, the Technical Details section has information about the category of the vulnerability, which in this case is CWE-400.

![Figure 2. Technical details section for a vulnerability page in NVD](image)

If you click on the hyperlink on the CWE ID, you will be redirected to the CWE page, where under Modes of Introduction, you can see if a particular CWE may stem from an architectural decision or not. In this case, you can see that Architecture is listed under Modes of Introduction, and as such, this vulnerability can be an architectural vulnerability. In cases like these, when a CWE has multiple modes of introduction, you will need to look

---

7 [https://nvd.nist.gov](https://nvd.nist.gov)
into the code and documentation a bit more to see if this is an architectural or implementation vulnerability.

```
Modes Of Introduction

The different Modes of Introduction provide which introduction may occur, while the Note column is a note on the introduction.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td></td>
</tr>
<tr>
<td>Architecture and Design</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 3. Modes of introduction for a particular CWE

In contrast, if we were to visit the page of CWE-66, we will see under its Mode of Introduction that only Implementation is listed. Cases like these you can discard immediately.
Appendix D

Resources