Report 06

URL: wiki.grab.com

Out-of-date Version (Tomcat)

HIGH

An out-of-date Tomcat vulnerability refers to a security weakness or flaw present in an older version of the Apache Tomcat server software. These vulnerabilities exist due to unpatched or outdated components within the Tomcat framework, making it susceptible to exploitation by attackers.

Running an out-of-date version of Tomcat exposes your web application to potential security risks. Attackers actively search for known vulnerabilities in older versions, which they can exploit to gain unauthorized access, compromise data integrity, or disrupt the normal functioning of the application.

URL	:	https://wiki.grab.com/login.action?os_destination=http://r87.com/n?%00.action&permissionViolation=true
Identified Version	:	9.0.65
Latest Version	:	9.0.75 (in this branch)
Overall Latest Version	:	10.1.9
Vulnerability Database	:	Result is based on 05/23/2023 20:30:00 vulnerability database content.
Attack Pattern	:	http%3a%2f%2fr87.com%2fn%3f%00.action

Method	**Parameter,**	**Parameter Type,**	**Value**
GET	os_destination	Querystring	http://r87.com/n?%00.action
GET	permissionViolation	Querystring	true

a. Apache Tomcat Inconsistent Interpretation of HTTP Requests ('HTTP Request/Response Smuggling') Vulnerability

If Apache Tomcat 8.5.0 to 8.5.82, 9.0.0-M1 to 9.0.67, 10.0.0-M1 to 10.0.26 or 10.1.0-M1 to 10.1.0 was configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false (the default for 8.5.x only), Tomcat did not reject a request containing an invalid Content-Length header making a request smuggling attack possible if Tomcat was located behind a reverse proxy that also failed to reject the request with the invalid header.

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N

CVE-2022-42252

b. Apache Tomcat Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection') Vulnerability

The JsonErrorReportValve in Apache Tomcat 8.5.83, 9.0.40 to 9.0.68 and 10.1.0-M1 to 10.1.1 did not escape the type, message or description values. In some circumstances these are constructed from user provided data and it was therefore possible for users to supply values that invalidated or manipulated the JSON output.

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N

CVE-2022-45143

Affected Components

- Tomcat Server: The vulnerable version of the Tomcat server itself is an affected component. This includes the core components, modules, libraries, and configurations of the Tomcat server.

- Web Applications: If the vulnerable Tomcat server hosts any web applications, those applications are also considered affected components. This includes both custom-developed web applications and third-party applications deployed on the Tomcat server.

- Network Infrastructure: The network infrastructure that facilitates communication with the Tomcat server, such as routers, switches, load balancers, and firewalls, can also be affected. The vulnerability may impact the network components involved in routing and forwarding requests to the vulnerable Tomcat server.

- Operating System: The underlying operating system on which the Tomcat server is running can be considered an affected component. The vulnerability may exploit weaknesses or security gaps in the operating system that could affect the Tomcat server's overall security.

- End Users: Depending on the nature of the vulnerability, end users accessing the web applications hosted on the vulnerable Tomcat server may also be affected.

Impact Assessment

- Unauthorized Access: The vulnerability may allow attackers to gain unauthorized access to the Tomcat server, web applications, or sensitive data stored within the

  server. This can lead to unauthorized viewing, modification, or deletion of data, as well as potential misuse of system resources.

- Data Breach: If the vulnerable Tomcat server stores sensitive data, such as user credentials, personal information, or financial data, a successful exploit can result in a data breach. This can lead to the exposure of sensitive information, potential identity theft, financial losses, and reputational damage.

- Service Disruption: Exploitation of the vulnerability can cause service disruptions or complete system compromise. Attackers may be able to disrupt the functioning of the Tomcat server, rendering web applications inaccessible to legitimate users. This can result in downtime, loss of productivity, and financial losses for the affected organization.

- Malware Injection: In some cases, an out-of-date Tomcat server vulnerability can be leveraged by attackers to inject and execute malicious code or malware. This can lead to further compromise of the server, unauthorized access to connected systems, or the distribution of malware to users accessing the compromised web applications.

Steps to Reproduce

- Set up the environment: Install and configure a vulnerable version of Apache Tomcat that is affected by the HTTP Request/Response Smuggling vulnerability. You can refer to the vulnerability advisory or security bulletin to identify the specific affected versions.

- Prepare the request: Craft a malicious HTTP request that exploits the vulnerability. This request should include specially crafted headers that can lead to inconsistent interpretation by the Apache Tomcat server.

- Send the malicious request: Use a tool like cURL, Postman, or a web browser plugin to send the malicious request to the targeted Apache Tomcat server.

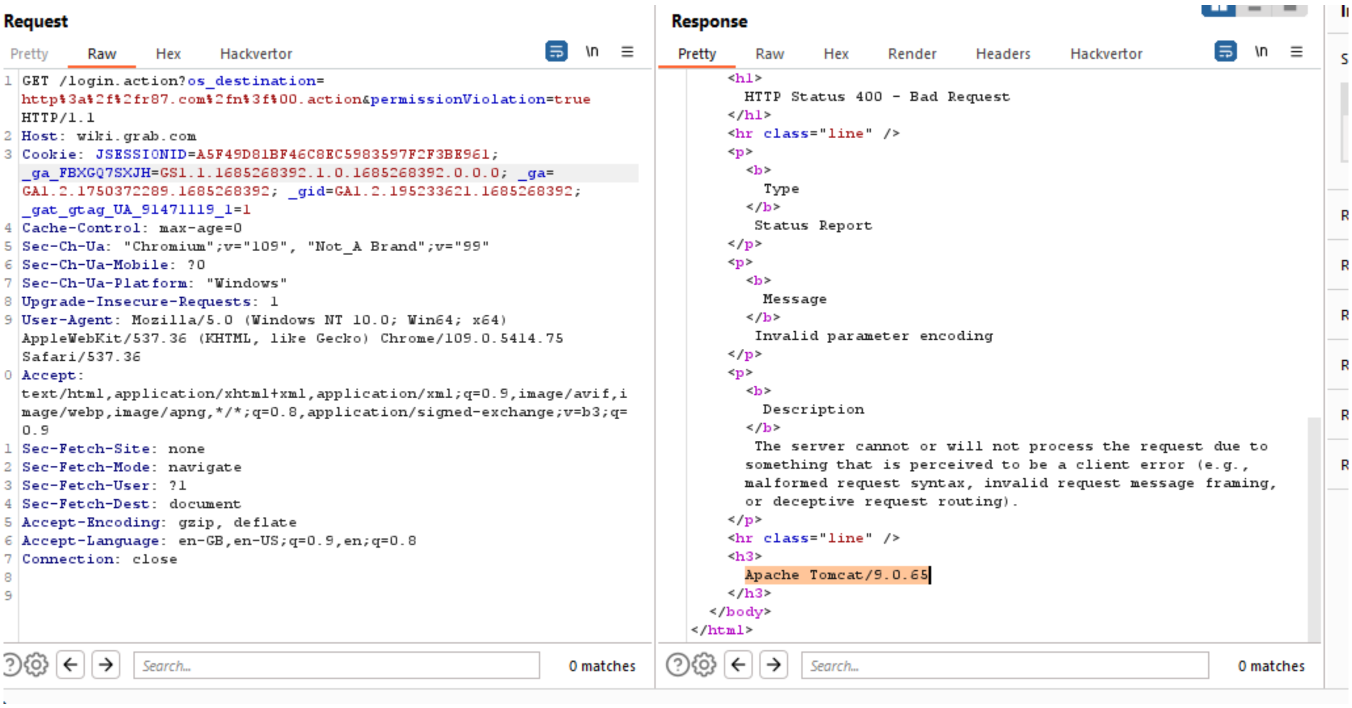
- Observe the server response: Monitor the server response for any unexpected behavior or inconsistencies in the interpretation of the HTTP request. Look for any signs that indicate that the server is vulnerable to HTTP Request/Response Smuggling.

- Analyze the response headers: Inspect the response headers received from the server. Look for any indications of inconsistent interpretation of the request, such as discrepancies in the Content-Length header or other headers related to request/response handling.

- Verify the vulnerability: Cross-reference the observed behavior and response headers with the known characteristics of the Apache Tomcat HTTP Request/Response Smuggling vulnerability. If the behavior aligns with the vulnerability description and known attack patterns, it confirms the presence of the vulnerability.

Proof of Concept

Identify Tomcat Version:



```
GET /login.action?os_destination=
http%3a%2f%2fr87.com%2fn%3f%00.action&permissionViolation=true
HTTP/1.1
Host: wiki.grab.com
Cookie: JSESSIONID=A5F49D81BF46C8EC5983597F2F3BE961;
_ga_FBXGQ7SXJH=GS1.1.1685268392.1.0.1685268392.0.0.0; _ga=
GA1.2.1750372289.1685268392; _gid=GA1.2.195233621.1685268392;
_gat_gtag_UA_91471119_1=1
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
```

```
    <h1>
      HTTP Status 400 - Bad Request
    </h1>
    <hr class="line" />
    <p>
      <b>
        Type
      </b>
       Status Report
    </p>
    <p>
      <b>
        Message
      </b>
       Invalid parameter encoding
    </p>
    <p>
      <b>
        Description
      </b>
       The server cannot or will not process the request due to
      something that is perceived to be a client error (e.g.,
      malformed request syntax, invalid request message framing,
      or deceptive request routing).
    </p>
    <hr class="line" />
    <h3>
      Apache Tomcat/9.0.65
    </h3>
  </body>
</html>
```

Proposed Mitigation or Fix

- ✓ Upgrade to the latest stable version: Upgrade your installation of Apache Tomcat to the latest stable version available. This ensures that you have the most recent security patches and fixes for known vulnerabilities.

- ✓ Patching known vulnerabilities: If you are unable to upgrade to the latest version immediately, apply patches or updates provided by the Apache Tomcat project specifically targeting the vulnerabilities identified (CVE-2022-42252 and CVE-2022-45143). These patches address the security issues and should be applied as soon as possible.

- ✓ Regular security updates: Keep your Apache Tomcat installation up to date by regularly checking for security updates and new releases. Stay informed about any security advisories or announcements from the Apache Tomcat project and promptly apply the recommended updates.

- ✓ Secure configuration: Ensure that your Apache Tomcat server is properly configured with secure settings. Review and adjust the server configuration files to adhere to recommended security best practices. This includes enabling secure protocols, disabling unnecessary features, and applying access controls.

- ✓ Web application security: Implement secure coding practices and perform regular security assessments and testing of your web applications deployed on Apache Tomcat. This helps identify and address any vulnerabilities specific to your applications.