CVE-2017-12615

1. **BUG OVERVIEW**

When running Apache Tomcat 7.0.0 to 7.0.79 on Windows with HTTP PUTs enabled (e. g. via setting the read-only initialisation parameter of the Default to false) it was possible to upload a JSP file to the server via a specially crafted request. This JSP could then be requested and any code it contained would be executed by the server. That could lead to remote code execution on the server.

2. BUG EXPLAINATION

Apache Tomcat is used to deploy your Java Servlets and JSPs. So, in your Java project you can build your WAR (short for Web Archive) file, and just drop it in the deploy directory in Tomcat. So basically, Apache is an HTTP Server, serving HTTP. Tomcat is a Servlet and JSP Server serving Java technologies.

Jakarta Server Pages (JSP; formerly JavaServer Pages) is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, SOAP, or other document types.

A JSP file is a server-generated web page. It is similar to an . ASP or . PHP file, but contains Java code instead of ActiveX or PHP

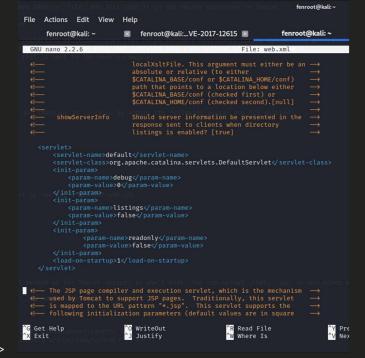
The code is parsed by the web server, which generates HTML that is sent to the user's computer.

Lab Setup:

- Install Apache Tomcat v7.0 in Docker Command: sudo docker run -p consol/tomcat-7.0
- 2. Note ContainerID and port Command: sudo docker ps
- Get IP Address of that docker image Command: sudo docker inspect -f format='{{.NetworkSettings.IPAddress}}' [ContainerID]

Checking Files:

PUT method is enabled through conf/web.xml
 (By default, PUT is enabled, so we are checking it to readonly =false by the following syntax)
 <init-param>
 <param-name>readonly</param-name>



<param-value>false</param-value>
</init-param>

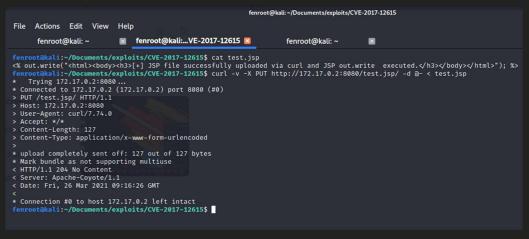
 No authentication enforced in the security-constraint set at the app's WEB-INF/web.xml (Removing authentication while uploading files)
 <auth-constraint></ri>
 </auth-constraint>

Request/Response from client:

By design, if you try to upload a JSP file via the HTTP PUT method on the Tomcat server, it won't work. You can upload .html, .jpg, or any other extensions except .jsp, .jspx and the variants.

Command:

curl -v -X PUT http://172.17.0.2:8080/test.jsp -d @- < test.jsp



However, we can bypass the extension check by appending a '/' behind the .jsp extension.

So now out file is uploaded successfully, with Response code 204.

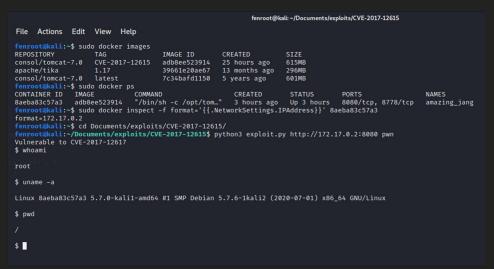
Now if we connect to http://172.17.0.2:8080/test.jsp , the java code runs and returns expected body.

```
| File Actions Edit View | Help | fenroot@kali:-/Documents/exploits/CVE-2017-12615 | fenroot@kali:-/Document
```

3. POC and EXPLOIT EXPLAINATION

Remote code Execution with limited functionalities can be attained with the following payload. **Payload**:

Exploit: command: python3 exploit.py http://172.17.0.2:8080 pwn



Details of the SERVER is as follows.

fenroot@kali: ~	
File Actions Edit View Help fenroot@kali:VE-2017-12615 fenroot@kali: ~	
fenroot@kali:~\$ sudo docker exec -it 8aeba83c57a3 /bin/bash root@8aeba83c57a3:/# whoami root root@aeba83c57a3:/# uname -a Linux 8aeba83c57a3 5.7.0-kali1-amd64 #1 SMP Debian 5.7.6-1kali2 (2020-07-01) x86_64 GNU/Linux root@8aeba83c57a3:/#	

Note 1: The details returned from exploit is same as that of the server. Note 2: The payload can be further crafted for further functionalities.

4. LINKS

POC Video Link: https://github.com/akverma00/exploits-ak/tree/main/CVE-2017-12615
Exploit Link: https://github.com/akverma00/exploits-ak/tree/main/CVE-2017-12615