

Abstract:

An attacker can gain system level privileges on a Windows 10 system using a Server Message Block (SMB) vulnerability. The vulnerability was notable due to a combination of leakage of information about the vulnerability by security companies before it was patched[1], the “triviality” of identifying the vulnerability (after reading a brief description[2] and its criticality [3].

Patches[4], detection rules[5] and Proof of concept (POC) scripts [6] are available.

This article details my experience in investigating this vulnerability. This mainly involves setting up vulnerable machines and trying the scripts against those VMs.

Before we start, a note to defenders

For network defenders, please make sure that:

- All Windows & Windows Servers ([version](#) 1903 and 1909) are patched with patch [KB4551762](#) and
- Block Ports 137,138,139,445 from [untrusted networks](#) (i.e. the internet) and
- Disable SMB Compression if you do not use it with the PowerShell command:
`Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" DisableCompression -Type DWORD -Value 1 -Force`
(This only protects SMB servers)

SMB – what’s this troublesome block?

Even before the current SMBv3 remote code execution vulnerability, there has been a few vulnerabilities in the Server Message Block network protocol that has allowed anything from privilege escalation to remote code execution. Examples of these are [CVE-2018-0749](#) and [CVE-2017-0144](#).

SMB is a network protocol that allows systems (mostly running on Microsoft Windows) to share files, printers and even serial ports. From the standpoint of the OSI layer conceptualization, one can visualize the protocol itself running on the application layer ([according to Microsoft](#)) and this running “on top” of Transmission Control Protocol (TCP) as the Transport Layer. Since SMB is a common feature of Windows that is enabled for file sharing, an attacker can expect that an insecurely configured computer would probably have it available (and even better, have it in an outdated and vulnerable state).

SMB has had several improvements from its initial implementation over the years. These include larger block sizes, SMB [Transparent Failover](#), AES based [signing](#), pre-authentication integrity [check](#) using SHA-512. While these improved performance and security, the vulnerability we are going to look is not mitigated by any of these (as the vulnerability affects the latest version SMB version 3.11).

Attack scenarios

Due to insufficient validation of two fields `OriginalCompressedSegmentSize` and `Offset/Length`, an attacker can trigger an integer overflow. With the right exploit, this leads to privileged code execution.

There are three scenarios of attack against a vulnerable Windows 10 computer that have been explained by [Sophos](#) and others. These scenarios are:

1. Scenario 1: An attacker attacks a Windows 10 machine that permits incoming SMB connections. In this scenario, the attacker gains control of the machine without any interaction (or awareness) from the victim. A situation of the attacker coming from the Internet to do this is unlikely (but not impossible, looking at the results from the [Shodan](#) search engine). Often, such an attack would come from an attacker that has already entered the network.

The screenshot shows the Shodan search engine interface. The search bar contains the query "os:\"windows 10\" port:445 country:sg". The results are categorized into "TOTAL RESULTS" (121), "TOP ORGANIZATIONS", and "TOP OPERATING SYSTEMS".

TOP ORGANIZATIONS

Organization	Count
Singapore	21
Telecommunications	18
Choopa, LLC	17
StarHub	17
SingTel Mobile	17
M1	11

TOP OPERATING SYSTEMS

Operating System	Count
Windows 10 Pro	18362
Windows 10 Home	18362
Windows 10 Pro	17134
Windows 10 Pro	18363
Windows 10 Enterprise	12
Evaluation	10240

Search Results Summary:

IP Address	Organization	Vulnerabilities	SMB Status
116.49	bbi.com.sg, net.com.sg, Singtel Fibre, Broadband	CVE-2020-0796	Authentication: enabled, SMB Version: 1, Capabilities: unicode,large-files,nt-smb,rpc-remote-api,nt-status,level2-op
227.210	M1 NET	CVE-2020-0796	Authentication: enabled, SMB Version: 1, Capabilities: unicode,large-files,nt-smb,rpc-remote-api,nt-status,level2-op
110.164	Singapore, Telecommunications	CVE-2020-0796	Authentication: enabled, SMB Version: 1, Capabilities: unicode,large-files,nt-smb,rpc-remote-api,nt-status,level2-oplocks,lock-and-read
206.36	Singapore, Telecommunications	CVE-2020-0796	Authentication: enabled, SMB Version: 1, Capabilities: unicode,large-files,nt-smb,rpc-remote-api,nt-status,level2-op

2. Scenario 2: Through social engineering, a victim is induced to click on a link to a malicious SMB server. Once connected, the SMB server sends the exploit packet to the victim. Control of the victim's computer follows thereafter.
3. Scenario 3: The attacker gains some privileges on the machine and uses the exploit to gain system level access. The initial privilege can come from a user with limited privileges (i.e. most corporate users in an environment that controls privileged access) clicking on a phishing email.

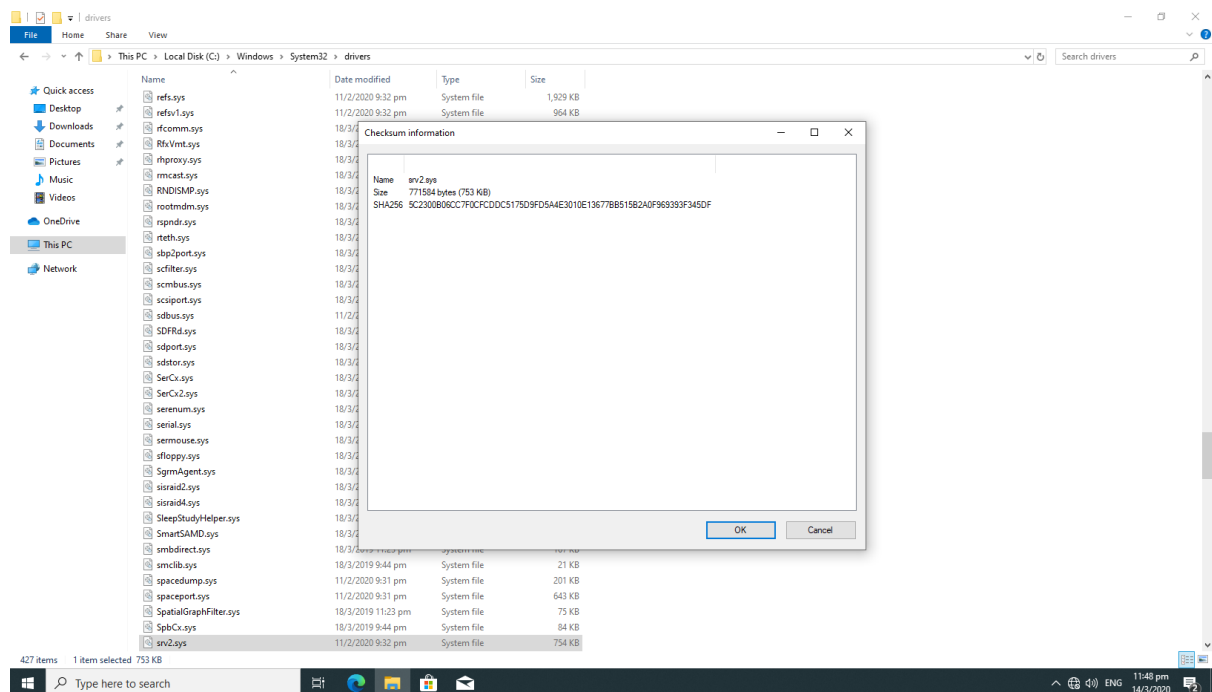
Systems that support SMB v3.11 and SMB Compression are affected. Unfortunately, these operating systems are known to run with SMB Compression enabled by default:

- Windows 10 Version 1903 for 32-bit Systems
- Windows 10 Version 1903 for ARM64-based Systems
- Windows 10 Version 1903 for x64-based Systems
- Windows 10 Version 1909 for 32-bit Systems
- Windows 10 Version 1909 for ARM64-based Systems
- Windows 10 Version 1909 for x64-based Systems
- Windows Server, version 1903 (Server Core installation)
- Windows Server, version 1909 (Server Core installation)

Setting up

Setting up a vulnerable virtual machine is not difficult. One only needs to install a fresh install of a vulnerable build. One of the ways that this can be done is via a Windows 10 Virtual Machine from Microsoft (<https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>).

Once the install is done, confirm that the file `srv2.sys` is not updated. This can be done by checking the “Date Modified” timestamp of the file. This file is the server component of what was patched in the Microsoft patch (see [McAfee analysis](#)).



After that, ensure that the attacking machine can access all ports on the victim machine. Your victim is now ready to be victimized.

Simulating an attacker

@josz5930

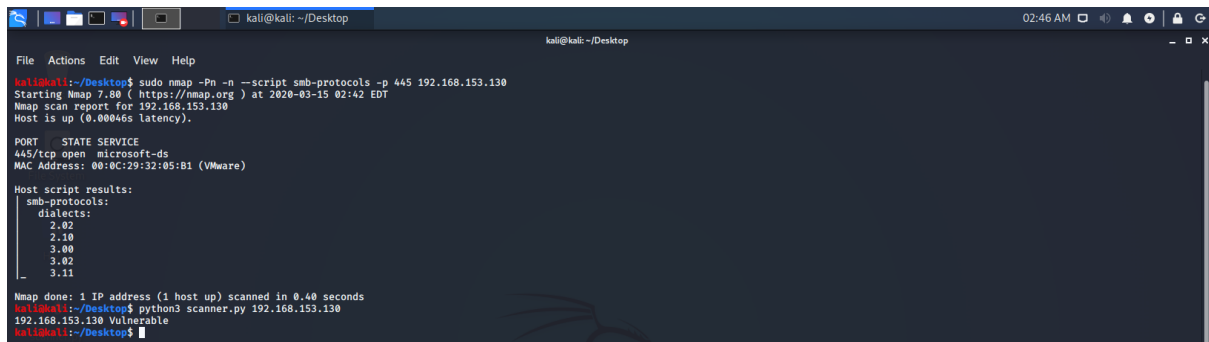
For the attack, we will only go through the steps of identifying a vulnerable system for later exploitation (reasons given later).

We will be using the following Python script to identify vulnerable systems:

[illegible][illegible]

@josz5930

The following screenshots shows a scan for a system that is running the vulnerable SMB version (3.11). Next, the python script is executed as a second level check.



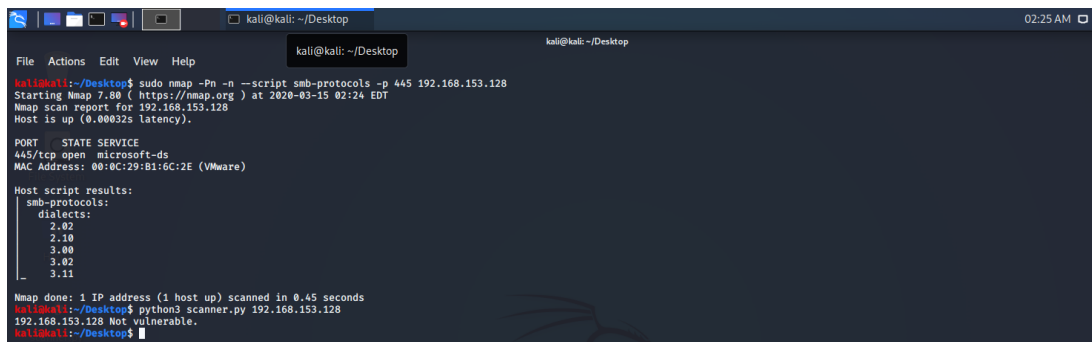
```
kali@kali: ~/Desktop
File Actions Edit View Help
kali@kali:~/Desktop$ sudo nmap -Pn -n --script smb-protocols -p 445 192.168.153.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-15 02:42 EDT
Nmap scan report for 192.168.153.130
Host is up (0.00046s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:32:05:B1 (VMware)

Host script results:
  smb-protocols:
    dialects:
      2.02
      2.10
      3.00
      3.02
      3.11
  _

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
kali@kali:~/Desktop$ python3 scanner.py 192.168.153.130
192.168.153.130 Vulnerable
kali@kali:~/Desktop$
```

In order to check that the script is actually working and not saying everything is vulnerable (false positive), you can opt to test it on a system that is and was not vulnerable.



```
kali@kali: ~/Desktop
File Actions Edit View Help
kali@kali:~/Desktop$ sudo nmap -Pn -n --script smb-protocols -p 445 192.168.153.128
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-15 02:24 EDT
Nmap scan report for 192.168.153.128
Host is up (0.00032s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:B1:6C:2E (VMware)

Host script results:
  smb-protocols:
    dialects:
      2.02
      2.10
      3.00
      3.02
      3.11
  _



Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
kali@kali:~/Desktop$ python3 scanner.py 192.168.153.128
192.168.153.128 Not vulnerable.
kali@kali:~/Desktop$
```

Links &References:


- [MS-SMB2]: Server Message Block (SMB) Protocol Versions 2 and 3 (2020, March 14). Retrieved from <https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-SMB2/%5bMS-SMB2%5d.pdf>
- Scanner for CVE-2020-0796 - SMBv3 RCE(2020, March 14). Retrieved from <https://github.com/ollypwn/SMBGhost>
- PoC for triggering buffer overflow via CVE-2020-0796 (2020, March 14). Retrieved from <https://github.com/eerykitty/CVE-2020-0796-PoC>
- Microsoft SMBv3.11 Vulnerability and Patch CVE-2020-0796 Explained. (2020, March 4). Retrieved from <https://www.sans.org/blog/microsoft-smbv3-11-vulnerability-and-patch-cve-2020-0796-explained/>
- SMBGhost (CVE-2020-0796): a new wormable Windows SMBv3 vulnerability (2020, March 15). Retrieved from <https://www.andreafortuna.org/2020/03/11/smbghost-cve-2020-0796-a-new-wormable-windows-smbv3-vulnerability/>
- 48K Windows Hosts Vulnerable to SMBGhost CVE-2020-0796 RCE Attacks (2020, March 4). Retrieved from <https://www.bleepingcomputer.com/news/security/48k-windows-hosts-vulnerable-to-smbghost-cve-2020-0796-rce-attacks/>
- SMBGhost – Analysis of CVE-2020-0796 (2020, March 14). Retrieved from <https://www.mcafee.com/blogs/other-blogs/mcafee-labs/smbghost-analysis-of-cve-2020-0796/>
- Technical analysis of Microsoft SMBv3 client / server remote code execution vulnerability (CVE-2020-0796) [Translated title] (2020, March 14). Retrieved from <http://blogs.360.cn/post/CVE-2020-0796.html>
- I'm SMBghost, Daba Dee Daba Da(2020, March 15). Retrieved from <https://www.synacktiv.com/posts/exploit/im-smbghost-daba-dee-daba-da.html>
- Wikipedia contributors. (2020, February 13). Server Message Block. In Wikipedia, The Free Encyclopedia. Retrieved 06:30, March 15, 2020, from https://en.wikipedia.org/w/index.php?title=Server_Message_Block&oldid=940583468
- CVE-2020-0796 - a wormable SMBv3 vulnerability. How to work. (2020, March 15). Retrieved from <https://github.com/cve-2020-0796/cve-2020-0796>
- Patch now! Microsoft releases fixes for the serious SMB bug CVE-2020-0796 (2020, March 16). Retrieved from <https://news.sophos.com/en-us/2020/03/12/patch-tuesday-for-march-2020-fixes-the-serious-smb-bug-cve-2020-0796/>

Note:

Windows Server 2019 (build 17763) is **not** impacted. It's the "equivalent" of Windows 10 LTS and is on the Long Term Servicing Channel ([LTSC](#)). I didn't realize the difference between LTSC and Semi-annual releases initially and thus observed unexpected (negative) results.

 Windows Server 2019
Evaluations | 180 days

In addition to your trial experience of Windows Server 2019, you can download a new feature on demand for Server Core, the App Compatibility FOD. This FOD contains additional features from the Desktop Experience to improve the compatibility of Server Core for apps and tools used for troubleshooting and debugging. Windows features on demand can be added to images prior to deployment or to actively running computers, using the DISM command. Learn more about the [Server Core App Compatibility FOD](#). Download this FOD. To learn more about FODs in general, and the DISM command, please visit [DISM Capabilities Package Servicing](#).

 **Start your evaluation**

Your download has started.

If the download did not start automatically, click the button below.

17763.737.190906-2324.rs5_release_svc_refresh_SERVER_EVAL_x64FRE_en-us_1.iso [Download](#)

