DE MONTFORT UNIVERSITY LEICESTER

# HOST AND NETWORK SECURITY

Adil Monu Lali Prabhakar

# TABLE OF CONTENT

# EXECUTIVE SUMMARY

Deficiencies in an IT system may be exploited by hackers to carry out a successful attack, known as "vulnerabilities". Attackers will try to exploit any of these, generally combining one or more to achieve their purpose, which may be caused by bugs, features, or user mistake. We'll talk about a vulnerability discovered in 2020, and we'll get into a detailed investigation of that issue. The vulnerability we'll talk about in this study is called SMBGhost, and it's a Remote Code Execution Vulnerability in Windows SMBv3 Client/Server (CVE-2020-0796). Certain queries made to the Microsoft Server Message Block 3.1.1 (SMBv3) protocol may lead to remote code execution. An attacker might obtain access to the target server or client and execute code if the flaw was successfully exploited.

An unauthenticated attacker might transmit a specially crafted packet to a targeted SMBv3 server in order to exploit the vulnerability. An unauthenticated attacker would have to set up a rogue SMBv3 server and convince a user to connect to it in order to exploit the vulnerability against a client.

# TASK 1

## Windows SMBv3 Clients/Server Remote Code Execution Vulnerability (CVE-2020-0796)

SMBGhost(also knows as SMBleedingGhost or CoronaBlue) is a worm-like security vulnerability that affects Windows 10 PCs. It was first publicly reported on March 10,2020. The IT giant mistakenly disclosed information about a security upgrade for a vulnerability that is wormable in the Microsoft Server Message Block (SMB) protocol._The weakness in the Server Message Block 3.0(SMBv3) network communication protocol is a pre-remote code execution problem.

Windows 10 version 1903, Windows Server Version 1903 (Server Core installation), Windows 10 Version 1909, and Windows Server Version 1909 are all effected by the CVE-2020-0796 vulnerability (Server Core installation). Other Microsoft version, according to Fortinet, should be also affected with the vulnerability.

Lets take a look at the score metrics of this vulnerability.

1. **Network as an attack vector**: This is a vulnerability that may be exploited remotely, hence the attack path is via the network.

2. **Attack Complexity**: Low: No specialised access criteria are required for this attack.

As a result, the complexity is low. An attacker who exploits this flaw is likely to succeed.

3. **No privileges are required**: Because no privilege is necessary for this assault, it becomes even more serious. Without access to settings or files, an unauthorised attacker can exploit this vulnerability.

4. **Interaction with user**: None: This flaw can be exploited without the need for a user to interact with the system.

5. **Confidentiality**: High: Because the attacker has access to all of the impacted systems, there is a risk of total loss of confidentiality.

6. **Integrity**: High: System files may be altered by an attacker, resulting in a complete loss of integrity.

7. **High Availability**: The attacker has complete control over the resources on the impacted system.

| Metric | Value |
|---|---|
| Base score metrics (8) | |
| Attack Vector | Network |
| Attack Complexity | Low |
| Privileges Required | None |
| User Interaction | None |
| Scope | Changed |
| Confidentiality | High |
| Integrity | High |
| Availability | High |

# TASK 2

## What is Server Message Block?

Using SMB (Server Message Block protocol), users may share access to printers, files, serial ports, and other network resources between themselves and the server. As a means of interprocess communication, it may also convey transactional protocols. SMB has traditionally been used to link Windows PCs, however many other operating systems, such as macOS and Linux, have client components for connecting to SMB resources.

## How does the SMB protocol work?

Client applications use SMB to open, read, move, create, and update files on distant servers in a server in a secure and regulated manner. Server programs configured to receive SMB client requests can also communicate with the protocol.

The SMB protocol, often known as a response-request protocol, is one of the most widely used ways for network communication. In this model, the client initiates the connection by sending an SMB request to the server. When the server receives the request, it responds by sending an SMB response back to the client, which establishes the two-way communication channel.

The SMB protocol functions at the application layer, although transport is handled by lower network layers. SMB used to run on top of NetBIOS over TCP/IP (NetBIOS over TCP/IP, or NBT) or, to a lesser extent, archaic protocols like internetwork Packet Exchange or NetBIOS Extended User Interface. SMB relies on ports 137, 139, and 138 for transfer when using NBT. SMB now uses port 445 and runs natively over TCP/IP.

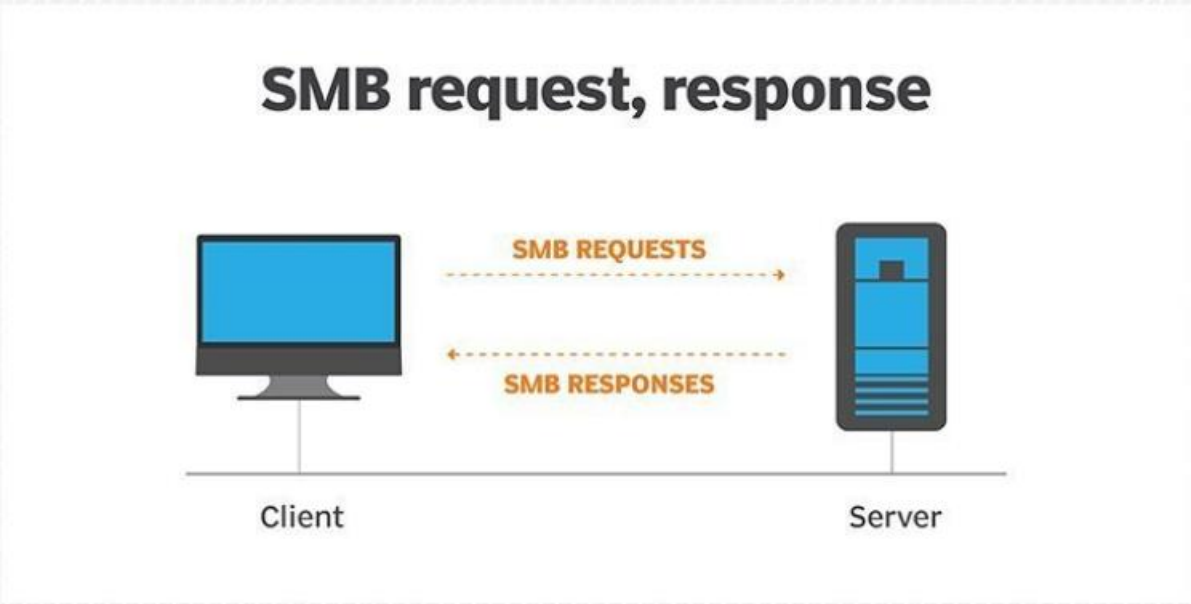



Figure 1: SMB request and response

The CVE-2020-0796 vulnerability has surfaced in SMB version 3 (SMBv3).

## Underlying flaws and technical explanation

SMB 3.1.1 protocol stack implementation on 2020 is vulnerable to a significant flaw, according to an official Microsoft advisory. Using this weakness, an attacker might run code in the SYSTEM user's context by exploiting the code's handling of certain request and response messages. The flaw, dubbed SMBGhost or CoronaBlue and given CVE-2020-0796, can only be exploited using a specifically crafted SMB v3 "compression Transform Header" Request or Response PDU. Only a small percentage of Windows 10 PCs running builds 1903 and 1909 are vulnerable to this issue. The affected versions are given in the following table:

| Product Name | Version | Platforms |
|---|---|---|
| Windows 10 | 1903(19H1),1909(19H2) | 32-bit,64-bit,ARM64 |
| Windows Server | 1903,1909 | - |

the attacker and victim have no way of verifying each other,so this flaw is considered critical and has a CVE score of 10. Furthermore, this flaw could affect both the client and server sides of a conversation.

The objective of publicly available scanners and exploit code appears to be:

- Using SMB protocol request and response behavior to identify targets (pre-crash phase).
- Sending faulty PDUs to cause the (integer) buffer to overflow.
- Remote code execution

**Identifying potential targets**

This is done by determining whether or not the destination host supports Data Compression. It is feasible to identify targets that are exploitable by this bug and those that are not based on the echoed back response data. When an SMB server receives a Negotiation request or its compression capabilities, the SMB server responds with a Negotiate Response Header containing the following information:

- Dialects that are supported (an SMB server version compatible with the targeted operating system)
- Context to Negotiate Count.

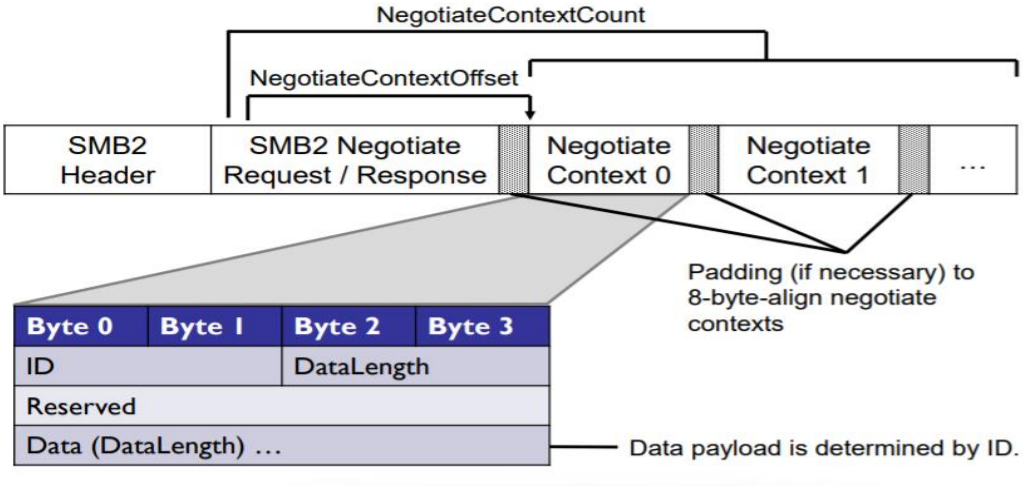



Figure 2: Negotiate contexts

Server contexts are only sent if the connection dialect is v3.1.1 (vulnerable dialect).

A remote code execution may be used to send a malicious package to the system, allowing an attacker to take over control of it.

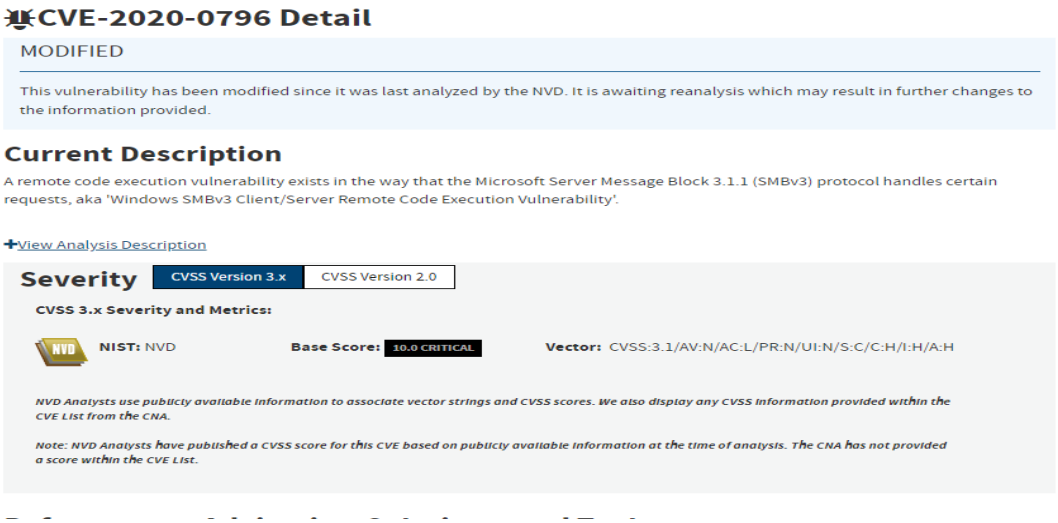



Figure 3: CVE database (https://nvd.nist.gov/vuln/detail/CVE-2020-0796)

However, not all programming languages guarantee that the memory addresses they offer are valid for the memory buffers they access. This might lead to memory accesses that are related to other variables, data structures, or internal programme data being accessed or written. Consequently, an attacker has the ability to execute any code, alter the intended control flow, access sensitive data or crash the system.

An effective buffer overflow attack might allow it to execute arbitrary code in the attacker's memory. If an attacker can overwrite enough memory to rewrite a function pointer, they may direct the reference to their own malicious code (typically 32 or 64 bits). Arbitrary code execution is possible regardless of how many bytes of information the attacker knows. This is sometimes because the same issue may be exploited again and the same outcome is obtained. Security-critical application data, such as a flag indicating whether or not the user is an administrator, may be overwritten by an attacker in certain cases.

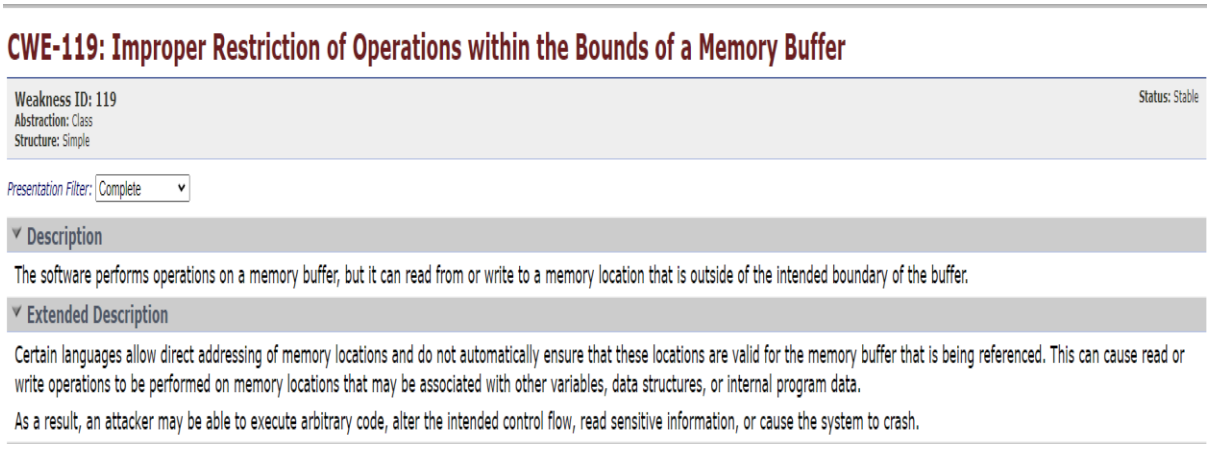



Figure 4: CWE database (http://cwe.mitre.org/data/definitions/119.html)

This vulnerability appears when a compressed message is faulty. The message's header is formatted like this:

The SMB2 COMPRESSION_TRANSFORM_HEADER is used by the client or server when sending compressed messages. This optional header is only valid for the SMB 3.1.1 dialect<69>.

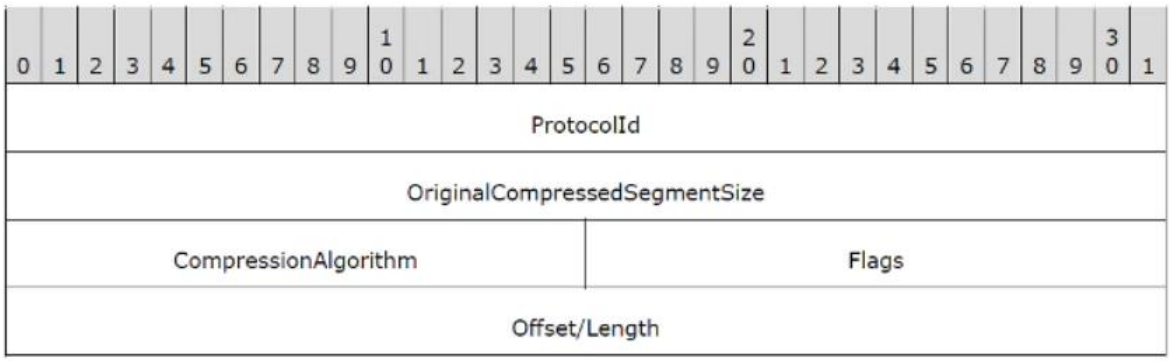



Figure 5: SMB compression transform header

The Srv2DecompressData function receives a compressed message from the client, allocates the necessary memory, and decompresses the contents. The data that is placed before the compressed data is then copied as is to the beginning of the allocated buffer if the Offset field is not zero.

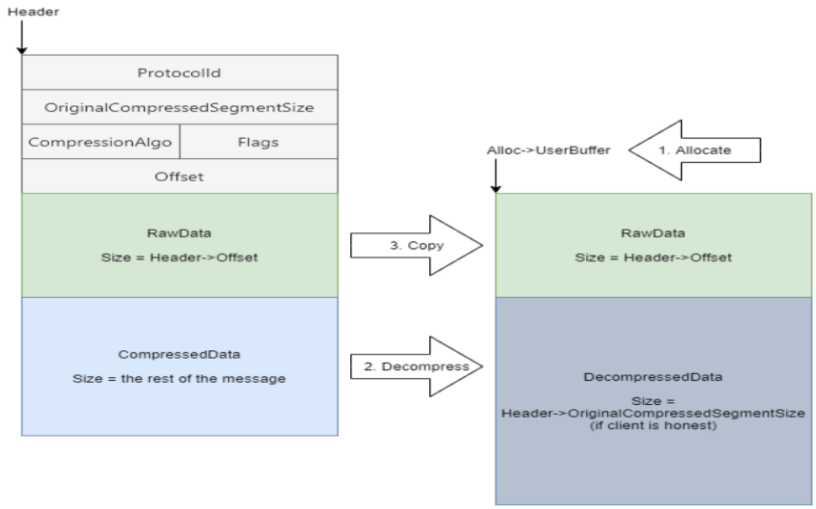



Figure 6:buffer allocation

Due to the fact that the addition is not tested for signedness, an attacker may create a larger buffer than intended. At buffer + offset, data is decompressed using data from packet+0x10.

SmbCompressionDecompression, a wrapper for RtlDecompressBufferEx2, accepts the OriginalCompressedSegmentSize as the UncompressedBufferSize argument.

This function converts a negative value to a big unsigned integer since the uncompressed buffer size is considered to be an unsigned long.

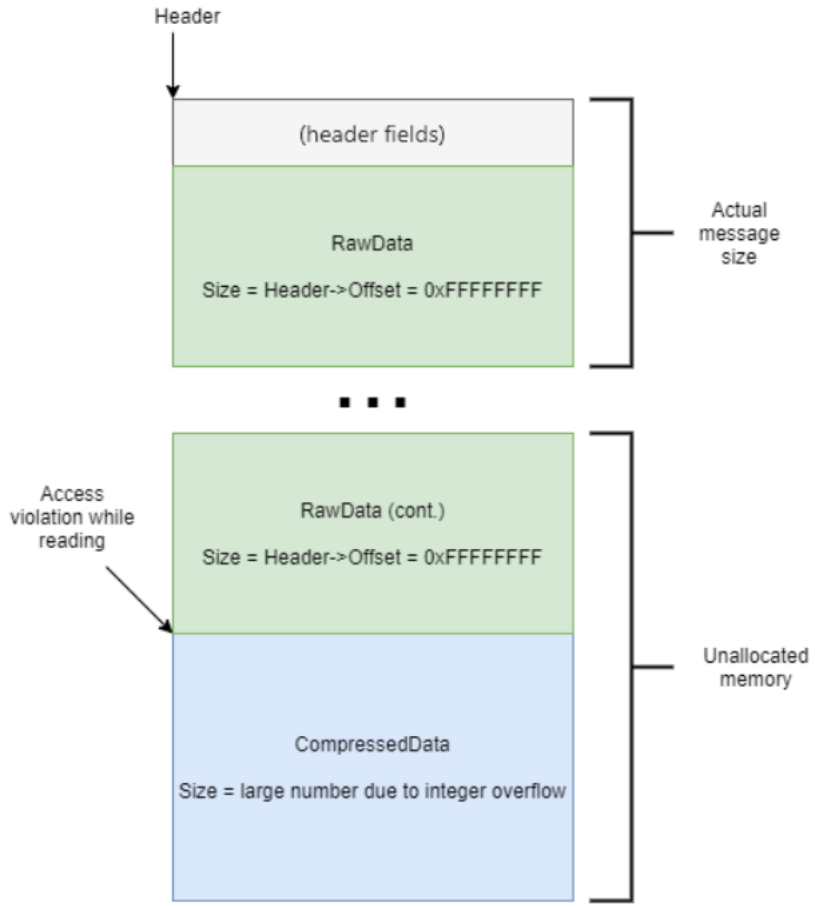



Figure 7: Buffer size

Thus, the decompression function may grow considerably bigger than the original size, provided it has a large buffer at its disposal. A compressed message following the Negotiate Protocol Responses potentially damage both the client and the server. There is a vulnerability in both the server and client code found in SmbCompressDecompress, which is located in the same file on both the server and client side (mrxsmb.sys).

If a computer enables incoming SMB3 communication by port 445, compression is enabled by default, and the client and server will negotiate the compression's "terms" before the client transfers a compressed payload.

| | | | | |
|---|---|---|---|---|
| 389 19.988011 | 192.168.204.1 | 192.168.204.135 | SMB | 127 Negotiate Protocol Request |
| 390 20.029671 | 192.168.204.135 | 192.168.204.1 | SMB2 | 506 Negotiate Protocol Response |
| 391 20.030023 | 192.168.204.1 | 192.168.204.135 | SMB2 | 296 Negotiate Protocol Request |
| 392 20.030855 | 192.168.204.135 | 192.168.204.1 | SMB2 | 588 Negotiate Protocol Response |

Figure 9:Negotiate Protocol Response

The initial size of attacker-controlled data in SMB3 will be very huge. An overflow of the smaller fixed buffer occurs as a result of the duplication. It's not only servers that may be hacked; the client-side, too, is vulnerable to cyberattacks. In order to trigger an overflow, both the initiator/client and the rogue SMB server must execute the same susceptible code, and a malicious server may respond to client requests in the same manner.

# TASK 3

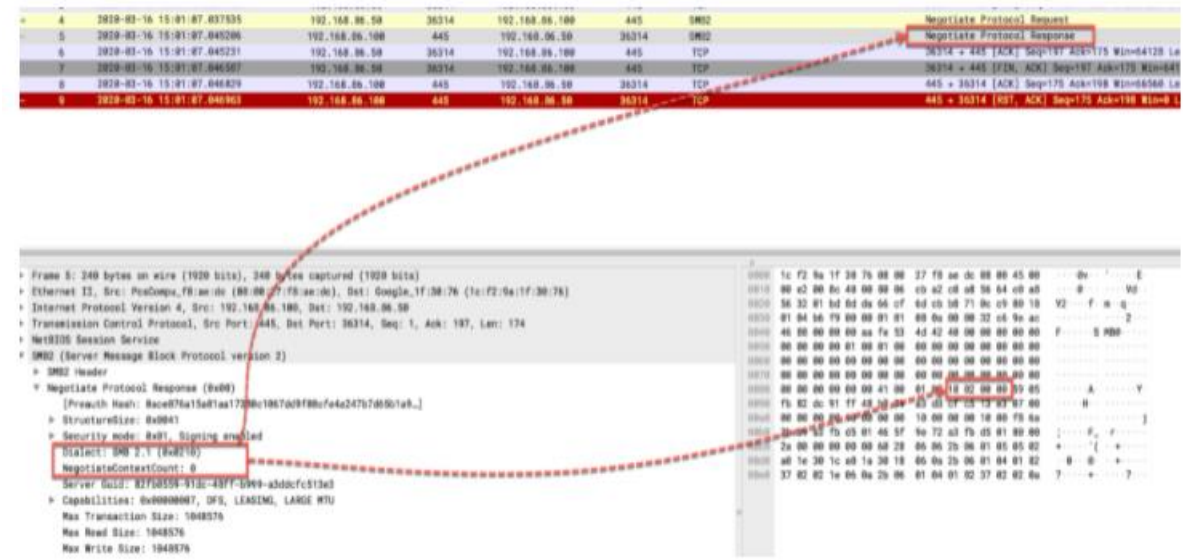
## Detection method

There are several ways to detect an attack or vulnerability, like checking and analyzing the PCAP file or using a tool to scan for the vulnerability etc.

We can detect the attack by analyzing the PCAP file and look for the Negotiate Protocol Response, the screenshot below will show the analysis of an windows 7 that is not vulnerable to this attack and a vulnerable windows 10 so that we can understand the difference between two.

Windows 7:

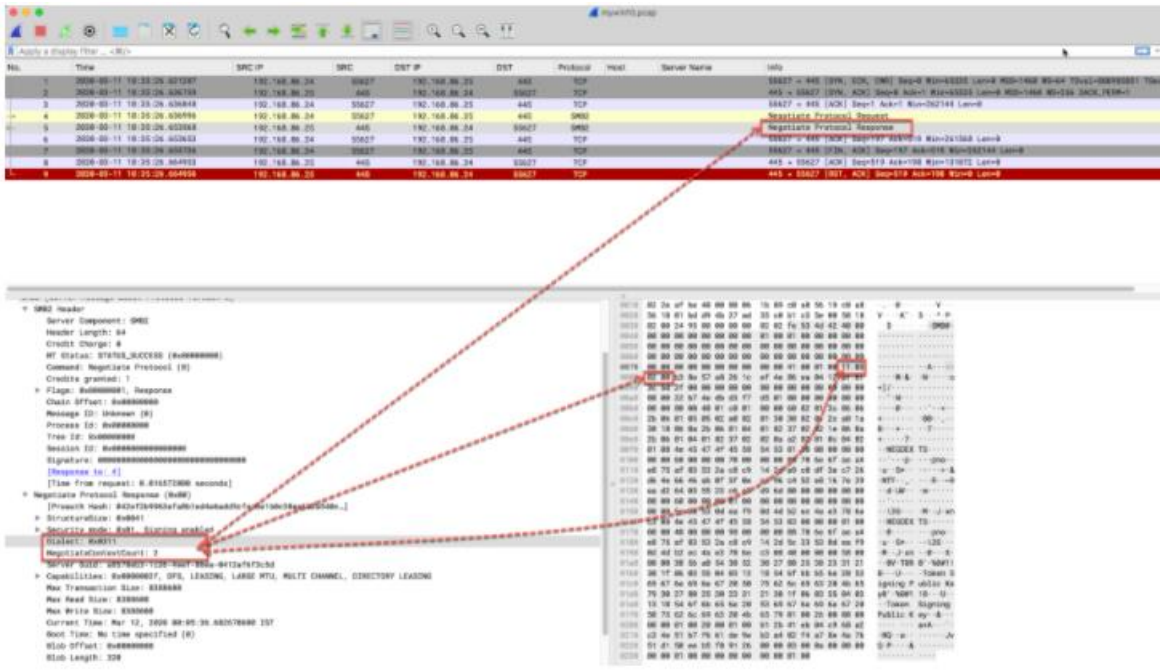
```
Dialect (2 Bytes): 0x0210 (SMB 2.1)
NegotiateContextCount (2 bytes): 0x0000
```



Windows 10:

```
Dialect (2 Bytes): 0x0311 (SMB 3.1.1)
NegotiateContextCount (2 bytes): 0x0200
```

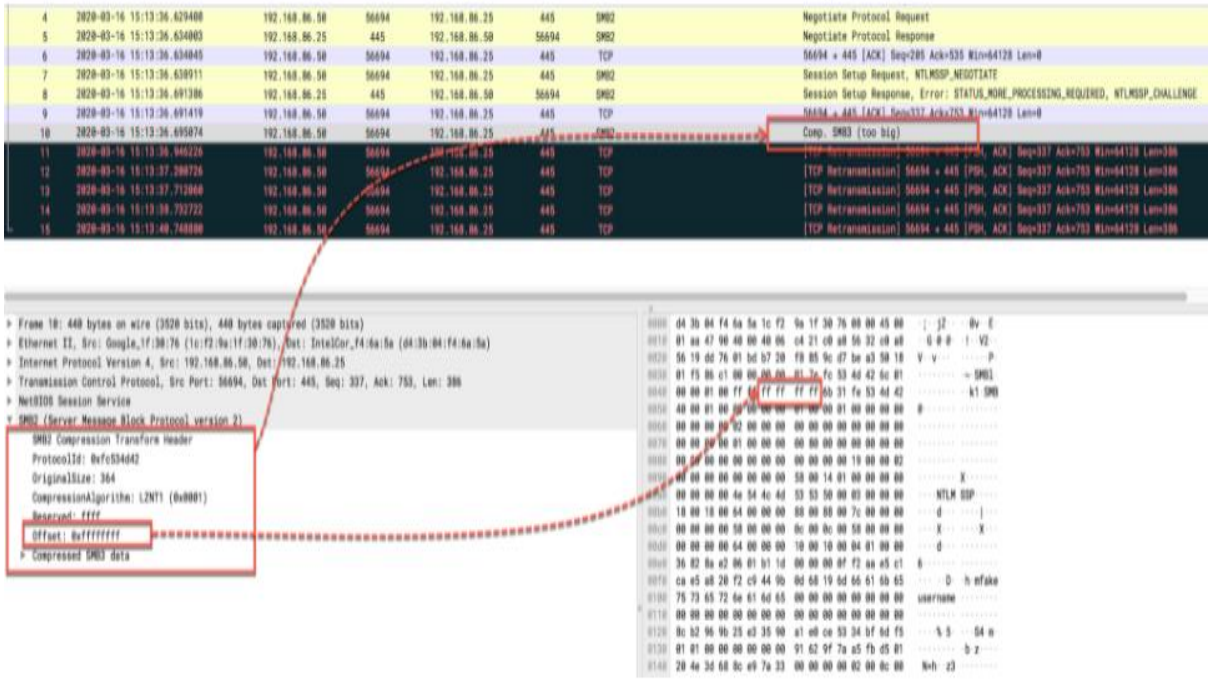
When the vulnerability host is found, the scanning behavior will look just like:



The main method a system admin can understand this vulnerability has been exploited really simple. The system will give a blue screen of death. The client must transmit an SMB2 Compression Transform Header PDU to the destination computer to cause the crash. The malicious compression request is broken down in full below:

```
ProtocolID (4 bytes): 0xFC534D42 (must)
OriginalSize (4 bytes): Length of compressed SMB3 Data (variable)
CompressionAlgorithm (2 bytes): LZNT1 (must be set to 0x0001)
Reserved (2 bytes): 0xFFFF
Offset (4 bytes): 0xFFFFFFFF (Higher value, -1 as signed long)
```

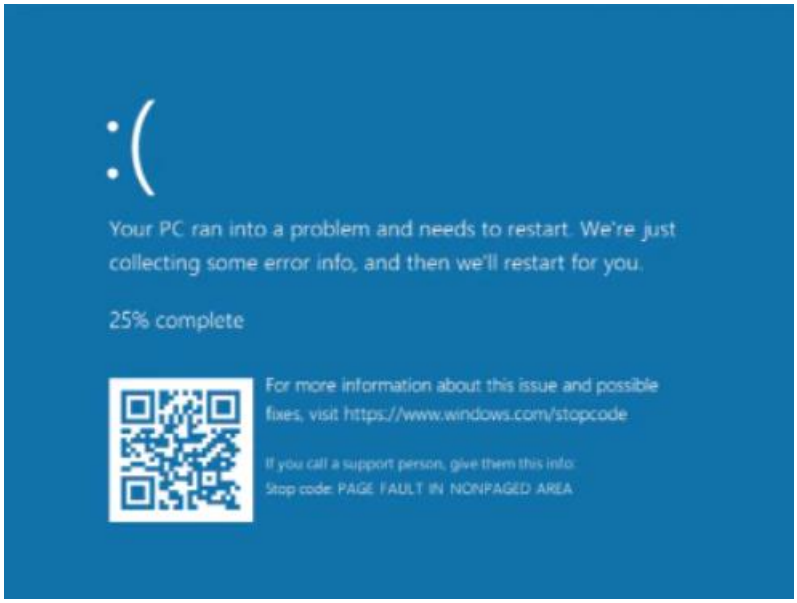
The attack data that triggers the crash is shown in the screenshot below:



When this vulnerability is been performed by any attacker, They focus mainly on two things:

1. Using the SMB protocol to identify targets
2. Causing a buffer overflow by delivering erroneous PDUs

A blue screen of death will appear on the target computer as a result.

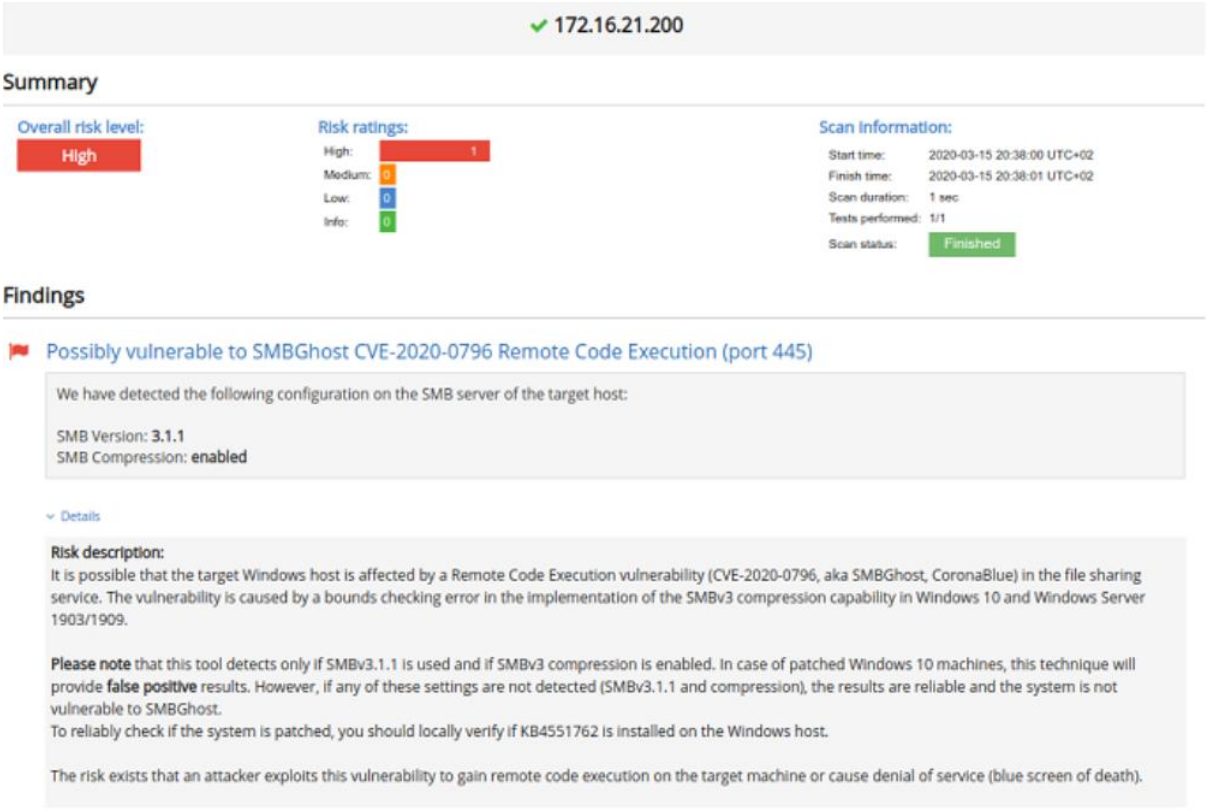


The SMBGhost is a vulnerability that a system admin can understand easily when it happens, but if this vulnerability is exploited, it will be very difficult to mitigate the consequences of the attack.

The one method a system admin can use to detect this vulnerability is by using a tool to scan for this vulnerability, here we will discuss about a tool known as Pentest-tool. They developed a SMBGhost scanner checks the target host's SMB version to see if compression is enabled in the SMB service. It begins by scanning the TCP 445 port, which is frequently used by Windows' file sharing service.

Additionally, Windows hosts that don't support SMBv3.1.1 or have SMBv3 compression disabled may be detected by the scanner.

An SMB negotiation is attempted with the target server, proposing SMB version 3.1.1 Dialect: 0x0311 with compression enabled, in order to verify these parameters. Count of NegotiateContexts: 2. If the recommended parameters are included in the SMB response packet, our scanner flags the target as potentially vulnerable to SMBGhost.




# TASK 4

## Countermeasures

Let us first discuss the fundamentals of security design principles that should be followed in order to create a safer environment and reduce the risk of being attacked.

The Cyber Security Principles provide the most universal recommendations. The Virtualization Design Principles apply to systems that rely on virtualization technology in a more particular way.

A set of guidelines, roughly related with the phases at which an attack could be reduced, divides it into five categories:

**Establish the context**

Determine all of the components that make up your system so that your defenses don't have any blind spots.

**Making compromise difficult**

Only the parts of a system to which an attacker has access may be attacked. Make it as difficult as possible to hack into your system.

**Making disruption difficult**

The goal is to build a system that can resist denial-of-service attacks and surges in traffic.

**Making compromise detection easier**

Systematize the detection of suspicious activity so that you can take immediate action when you see anything amiss.

**Reducing the impact of compromise**

Once they've established a foothold, an attacker will try to take advantage of your system's vulnerabilities. Increasingly complicate the situation for yourself.

Talking about the SMBGhost vulnerability, This issue, According to Microsoft, this vulnerability might lead to the execution of arbitrary remote code and be used as a wormable component by malware. According to the official Microsoft advisory, Awake recommends patching all devices right away. Also, firewall policies should not expose SMB service discovery to external networks. If patching isn't an option, the Microsoft advise recommends making registry modifications to disable SMB compression as a mitigation measure.

As a system admin we can discuss 2 methods which we can mitigate this attack:

Workarounds for this bug should be deployed on all servers and workstations that are impacted by this issue.

Make sure that firewall rules on the border and endpoints prohibit (block) connections to the vulnerable service, if applicable.. (445 TCP).

**The SMBv3 compression may be disabled in the first method.**

An endpoint can't begin an attack by connecting to another endpoint with bad intent. Incoming connections from other endpoints may still be attacked by malicious actors.

On the "server side," which is to say, the endpoints that listen for SMB rather than those that initiate it, disabling compression only works.

After making the adjustment, there is no need to reboot.

To disable the SMBv3, we use this command as shown below

```
• Set-ItemProperty -Path
  "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServ
  er\Parameters" DisableCompression -Type DWORD -
  Value 1 -Force
```

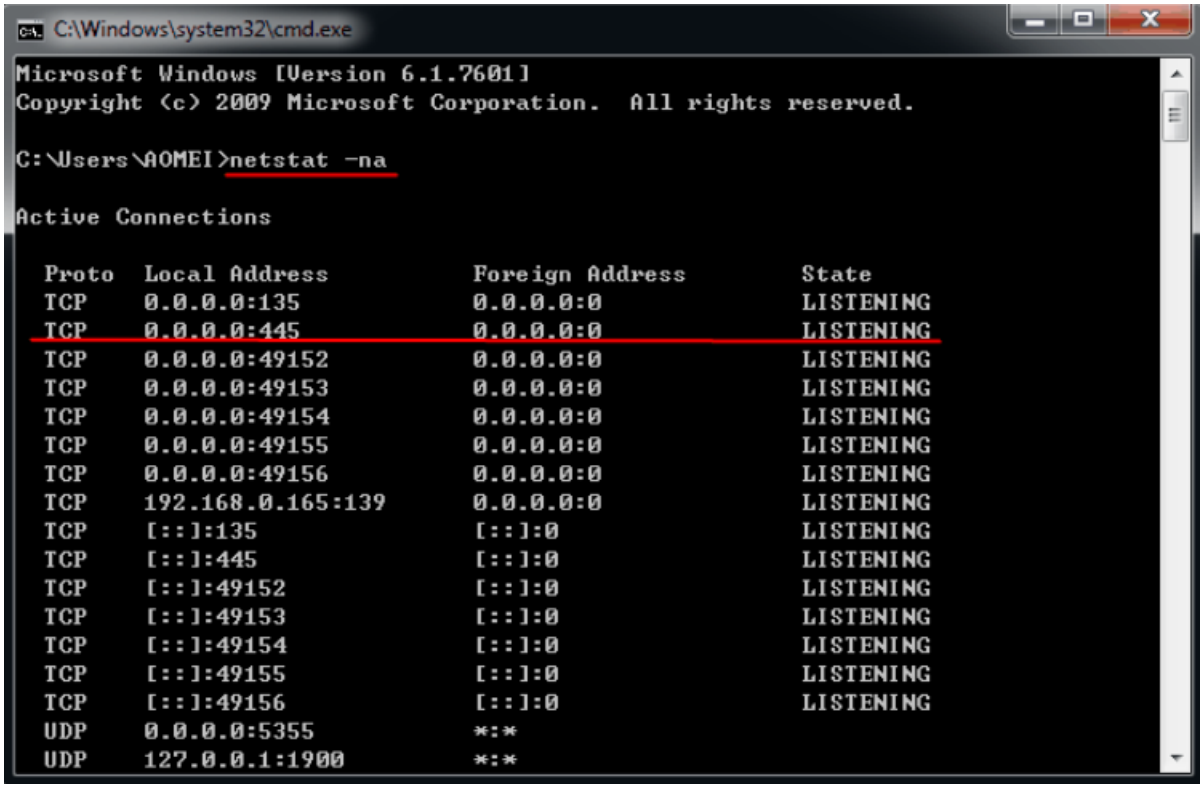
To re-enable:

```
• Set-ItemProperty -Path
  "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServ
  er\Parameters" DisableCompression -Type DWORD -
  Value 0 -Force
```

**Method 2: Block inbound and outbound SMB**

Imagine restricting Connections from an SMB server from the local network to the WAN (TCP port 445 for SMBv3). Also, make sure that internet-based SMB connections cannot connect inbound to an enterprise LAN.

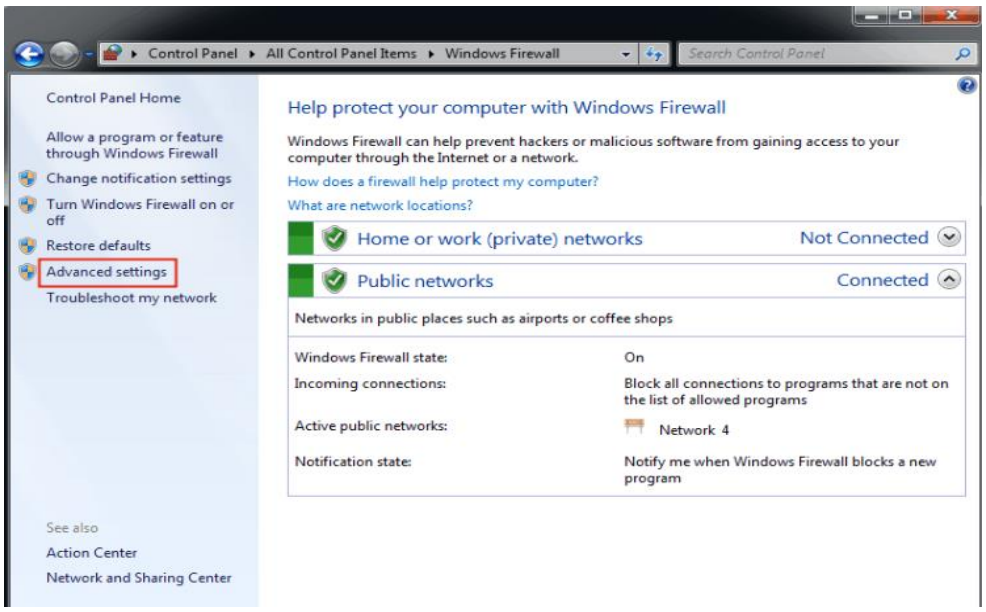
The port 445 is normally open by Windows' operating system, but to be safe, make sure it is on your host. To open the Run box, press the Windows + R key combination. To open Command Prompt, type "cmd." Then click Enter after typing "netstat –na." The command "netstat –na" scans all connected ports and displays the results in numbers.



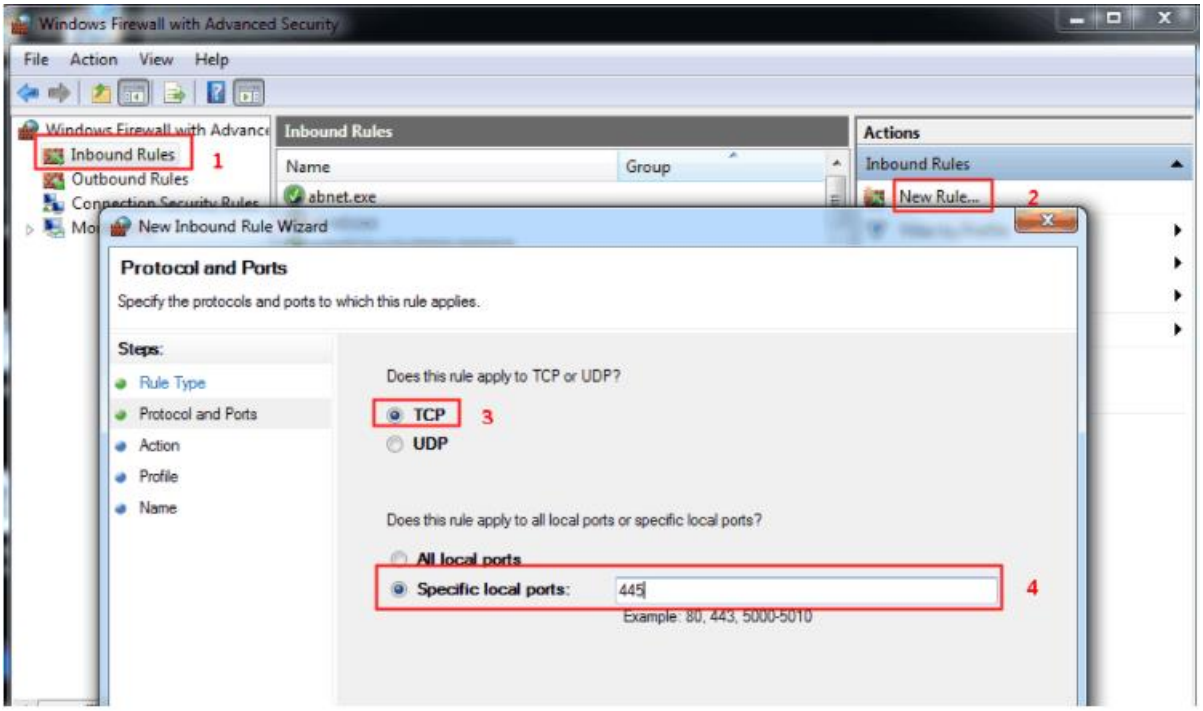

The image will appear in one or two seconds. If you move your cursor to the top of the screen, you'll see the IP address 445. The status in the last column reads "LISTENING." This indicates that TCP port 445 has been opened. Because it's one of the Internet's most hazardous ports, it was used in the WannaCry attack, and the SMBGhost vulnerability can be used in the same way. There are several ways to block port 445.

The first option is the simplest and is appropriate for practically all Windows users.

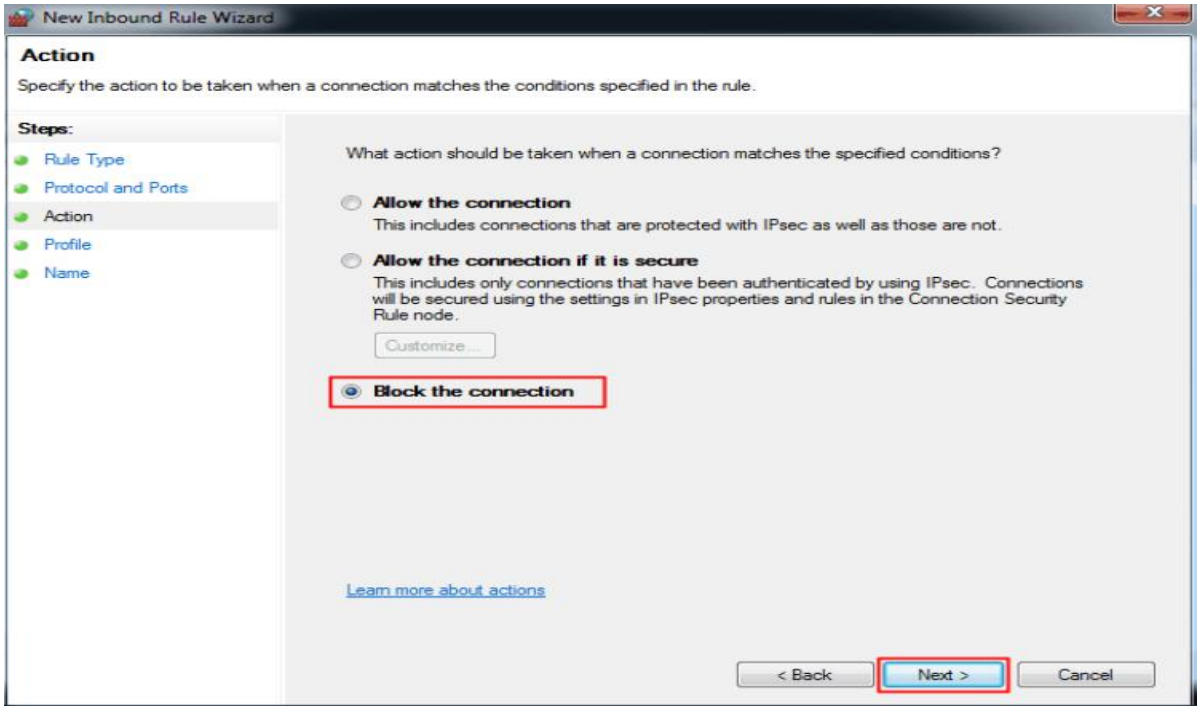
1. On the left side of the screen, go to Start > Control Panel > Windows Firewall and look for Advanced options.



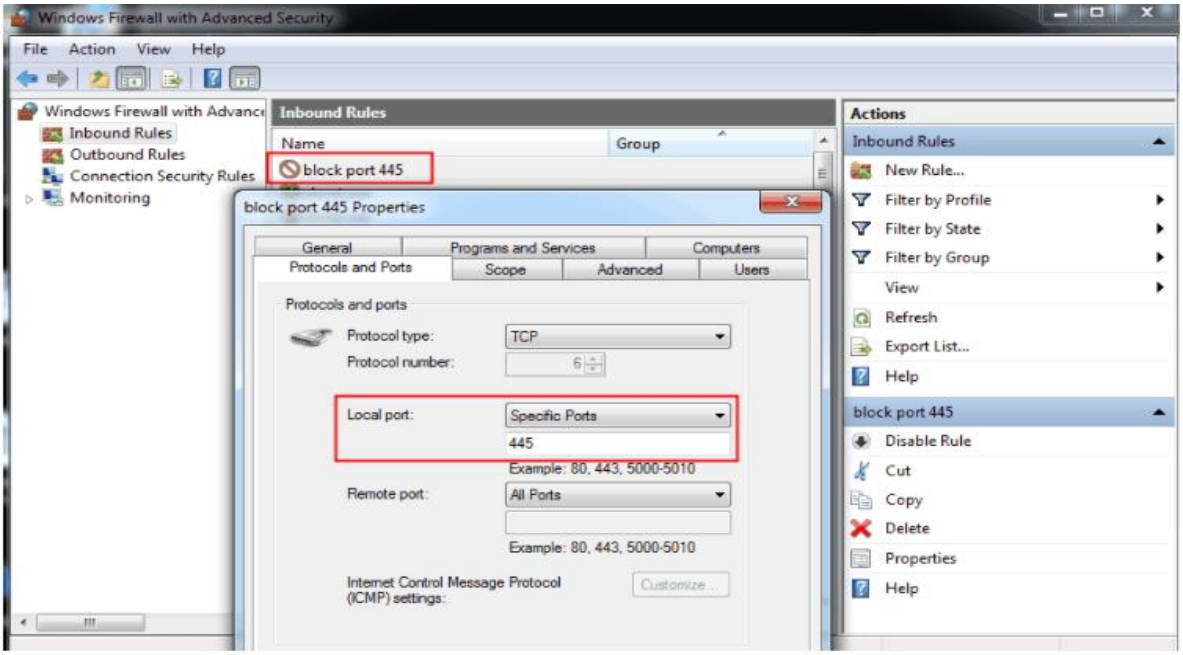
2. Select Inbound Rules > New Rule from the drop-down menu. Then select Port > Next >TCP > Specific local ports, type 445, then click Next in the pop-up window.




3. Select Block the connection > Continue. Click Next after checking the three checkboxes. Fill in the name and description as desired, then click Finish.



4. Go to Properties > Protocols and Ports > Local Port to see if the rule was created.

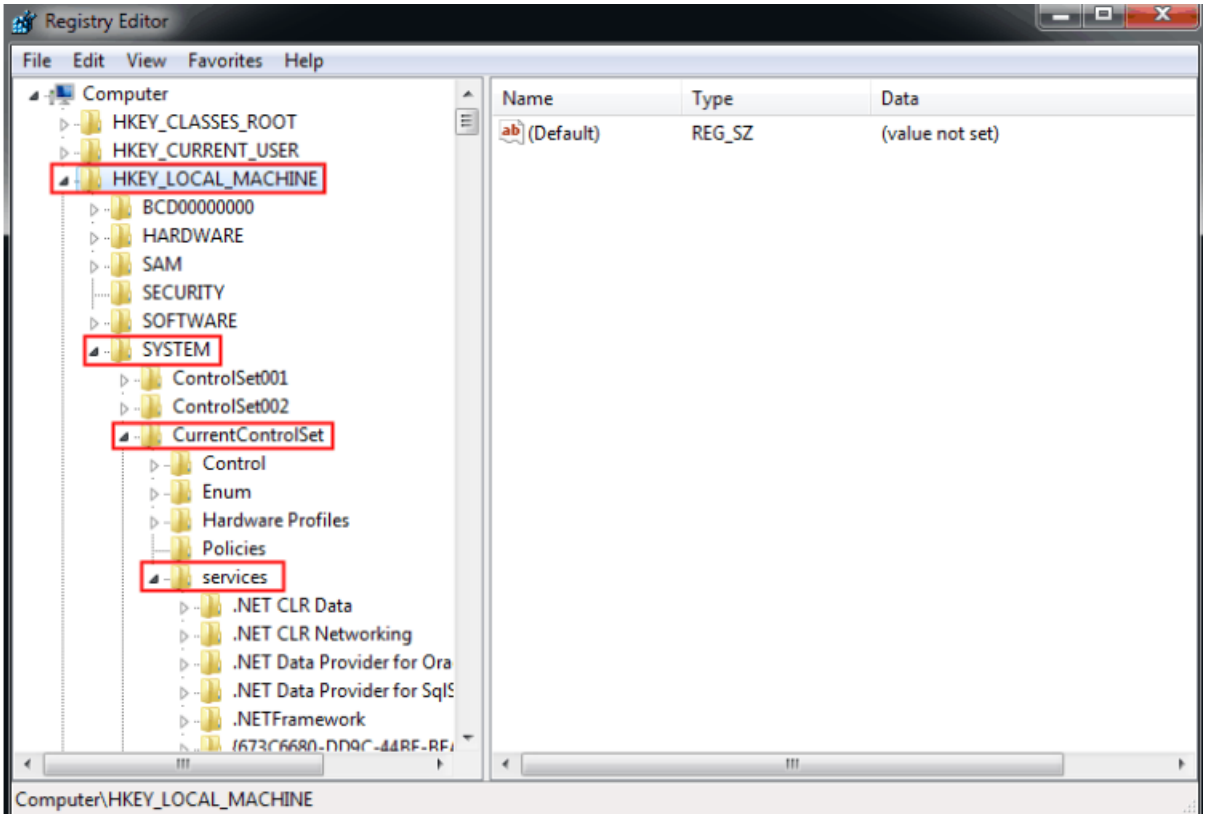



**Block TCP port 445 via RegEdit**

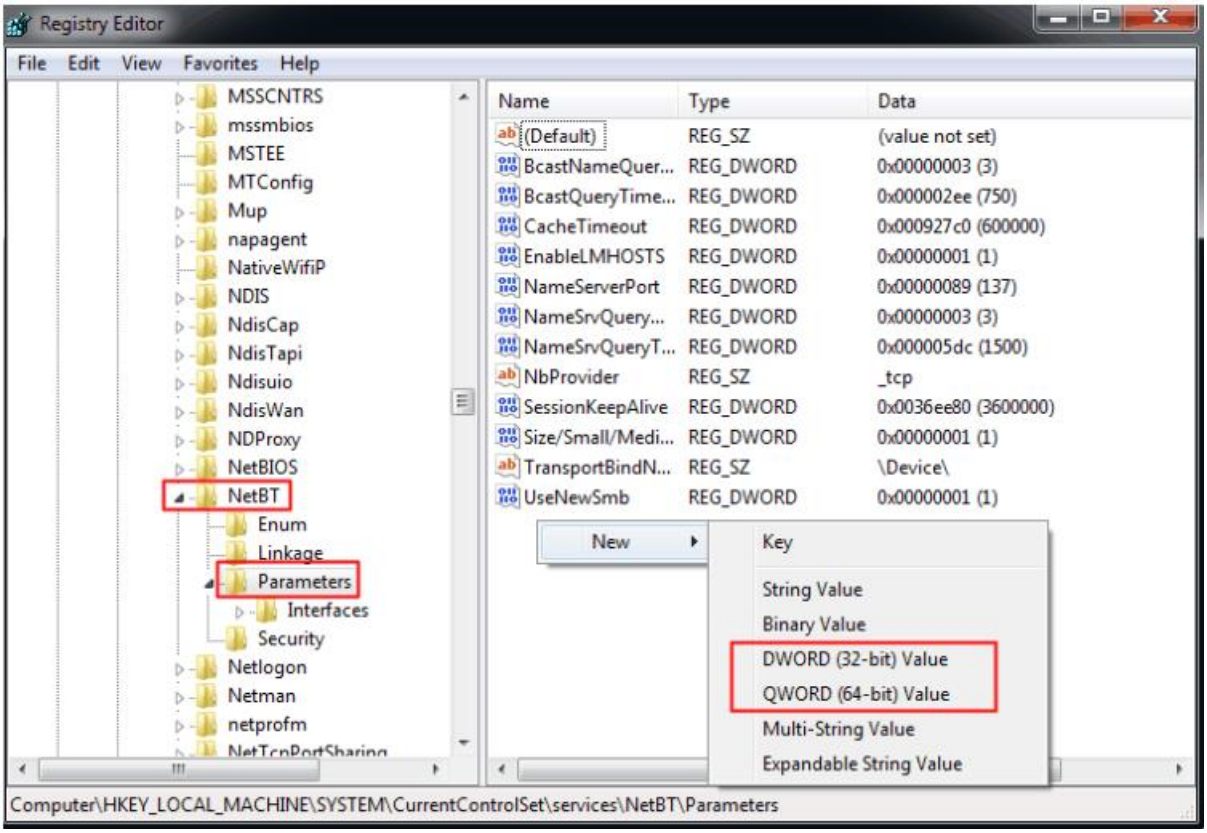
You can also defend yourself from this vulnerability by changing the system registry. You cannot, however, be more cautious while altering the register. It's a database for Windows system programmes and applications that have been installed. If you remove a critical file by accident, these apps may not work properly. Just in case, make a registry backup first.

1. In the same way, open the Run box. Enter "regedit" at the command prompt.
2. Find your way to the route:

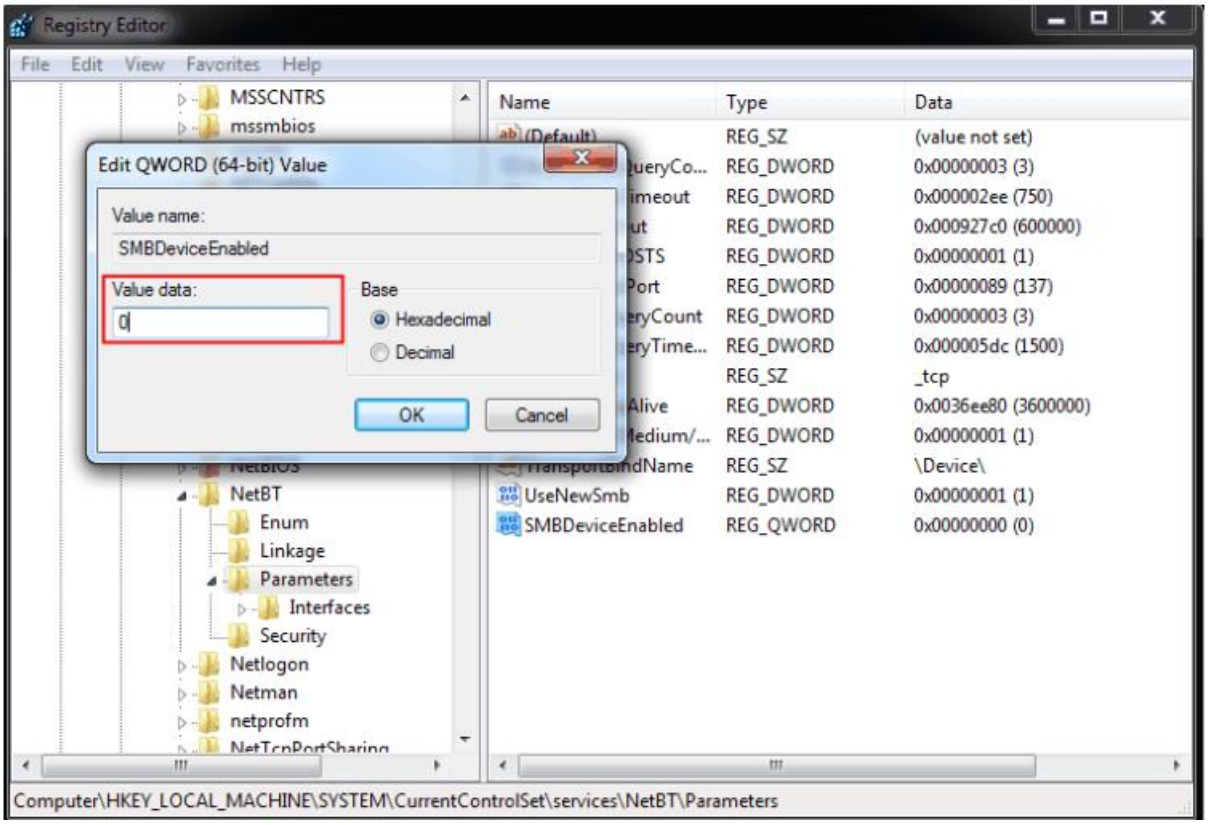
HKEY LOCAL MACHINE\SYSTEM\CurrentControlSet\services\NetBT\Parameters




Select New from the context menu when you right-click the blank area. Depending on your system type, choose between DWORD (32-bit) and QWORD (64-bit) values (32 bit or 64 bit).



SMBDeviceEnabled should be the new value's name. After that, right-click it and choose Modify. Change Value data from 1 to 0 in the pop-up window. To confirm, click OK.




A computer user may nearly universally benefit from this strategy.

## CONCLUSION

The SMBGhost weakness is simply the latest in a long line of mega-vulnerabilities, therefore it is clear that cybersecurity concerns cannot be ignored. The goalposts in the online world will continue to shift.

The question we should be asking is not how we can assure our security, but rather how we can continually enhance our security. Essential steps may vary, but the greatest potential results will always be based on the same technique.

We need to keep learning new things. Keeping an eye on our technological stock is essential. Risk profiling that is both flexible and context-aware must guide our network communication rules. A device's usual functions need disabling ports that aren't being utilised. Keeping an eye out for suspicious conduct in digital encounters is a must. We must, above all, we must remain vigilant.

# REFERENCE


- msrc.microsoft.com. (n.d.). *Security Update Guide - Microsoft Security Response Center*. [online] Available at: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-0796

- nvd.nist.gov. (n.d.). *NVD - CVE-2020-0796*. [online] Available at: https://nvd.nist.gov/vuln/detail/CVE-2020-0796.

- cwe.mitre.org. (n.d.). *CWE - CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer (4.6)*. [online] Available at: http://cwe.mitre.org/data/definitions/119.html.

- www.ncsc.gov.uk. (n.d.). *Cyber security design principles*. [online] Available at: https://www.ncsc.gov.uk/collection/cyber-security-design-principles/cyber-security-design-principles.

- www.ubackup.com. (n.d.). *Top Three Easy Methods to Block TCP Port 445 in Windows 10, 7, XP*. [online] Available at: https://www.ubackup.com/anti-ransomware/how-to-block-port-445-in-windows-3889.html

- Marquette, K. (n.d.). *Powershell: How to Disable SMBv3 Compression*. [online] powershellexplained.com. Available at: https://powershellexplained.com/2020-03-10-Powershell-disable-smb3-compression/

- SMBGhost. (n.d.). [online] Available at: https://www.safe.security/assets/img/research-paper/SMBGhost-CVE-2020-0796.pdf

- Rijnetu, I. (2020). *How to detect the Microsoft SMBGhost vulnerability with Pentest-Tools.com*. [online] Pentest-Tools.com Blog. Available at: https://pentest-tools.com/blog/how-to-detect-microsoft-smbghost-vulnerability

- ZecOps Blog. (2020). *Exploiting SMBGhost (CVE-2020-0796) for a Local Privilege Escalation: Writeup + POC*. [online] Available at: https://blog.zecops.com/research/exploiting-smbghost-cve-2020-0796-for-a-local-privilege-escalation-writeup-and-poc/

- says, L. (2020). *SMBGhost – Analysis of CVE-2020-0796*. [online] McAfee Blogs. Available at: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/smbghost-analysis-of-cve-2020-0796/

- Geffen, I. (n.d.). *The Windows 10 "SMBGhost" Vulnerability: What to Know & What to Do*. [online] blog.cybermdx.com. Available at: https://blog.cybermdx.com/the-smbghost-vulnerability-what-to-know-what-to-do

- Awake Security. (2020). *"SMBGhost" Wormable Vulnerability Analysis (CVE-2020-0796)*. [online] Available at: https://awakesecurity.com/blog/smbghost-wormable-vulnerability-analysis-cve-2020-0796/

- PDQ.com Corporation. (2020). *PDQ.com*. [online] Available at: https://www.pdq.com/blog/cve-2020-0796/#:~:text=How%20Does%20SMBGhost%20Work%3F