Latest SMB Vulnerabilities

One recent significant vulnerability that affected the SMB protocol was called SMBGhost with the CVE-2020-0796. The vulnerability consisted of a compression mechanism of the version SMB v3.1.1 which made Windows 10 versions 1903 and 1909 vulnerable to attack by an unauthenticated attacker. The vulnerability allowed the attacker to gain remote code execution (RCE) and full access to the remote target system.

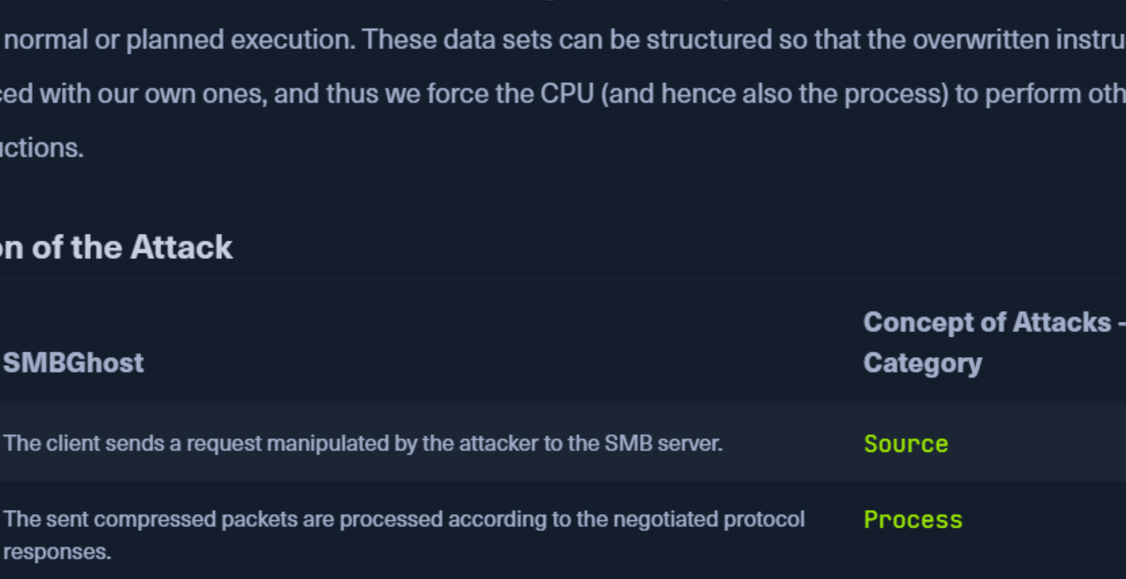
We will not discuss the vulnerability in detail in this section, as a very in-depth explanation requires some reverse engineering experience and advanced knowledge of CPU, kernel, and exploit development. Instead, we will only focus on the attack concept because even with more complicated exploits and vulnerabilities, the concept remains the same.

The Concept of the Attack

In simple terms, this is an integer overflow vulnerability in a function of an SMB driver that allows system commands to be overwritten while accessing memory. An integer overflow results from a CPU attempting to generate a number that is greater than the value required for the allocated memory space. Arithmetic operations can always return unexpected values, resulting in an error. An example of an integer overflow can occur when a programmer does not allow a negative number to occur. In this case, an integer overflow occurs when a variable performs an operation that results in a negative number, and the variable is returned as a positive integer. This vulnerability occurred because, at the time, the function lacked bounds checks to handle the size of the data sent in the process of SMB session negotiation.

To learn more about buffer overflow techniques and vulnerabilities, check out theStack-Based Buffer Overflows on Linux x86, and Stack-Based Buffer Overflows on Windows x86 module. These go into detail on the basics of how the buffer can be overwritten and handled by the attacker.

The Concept of Attacks



The vulnerability occurs while processing a malformed compressed message after the `Negotiate Protocol Responses`. If the SMB server allows requests (over TCP/445), compression is generally supported, where the server and client set the terms of communication before the client sends any more data. Suppose the data transmitted exceeds the integer variable limits due to the excessive amount of data. In that case, these parts are written into the buffer, which leads to the overwriting of the subsequent CPU instructions and interrupts the process's normal or planned execution. These data sets can be structured so that the overwritten instructions are replaced with our own ones, and thus we force the CPU (and hence also the process) to perform other tasks and instructions.

Initiation of the Attack

Step	SMBGhost	Concept of Attacks - Category
1.	The client sends a request manipulated by the attacker to the SMB server.	Source
2.	The sent compressed packets are processed according to the negotiated protocol responses.	Process
3.	This process is performed with the system's privileges or at least with the privileges of an administrator.	Privileges
4.	The local process is used as the destination, which should process these compressed packets.	Destination

This is when the cycle starts all over again, but this time to gain remote access to the target system.

Trigger Remote Code Execution

Step	SMBGhost	Concept of Attacks - Category
5.	The sources used in the second cycle are from the previous process.	Source
6.	In this process, the integer overflow occurs by replacing the overwritten buffer with the attacker's instructions and forcing the CPU to execute those instructions.	Process
7.	The same privileges of the SMB server are used.	Privileges
8.	The remote attacker system is used as the destination, in this case, granting access to the local system.	Destination

However, despite the vulnerability's complexity due to the buffer's manipulation, which we can see in the PoC, the concept of the attack nevertheless applies here.