SMBGhost

CVE 2020-0796

Safe Security

Table of Content

Introduction to CVE 2020-0796

CVE 2020-0796 was released in March 2020, with a CVSS:3.0 score of 10.0, which makes it a vulnerability to look out for. In this report, the readers will understand where this vulnerability resides in Windows 10, how it worked, and why it was so critically dangerous.

Let's start with where the vulnerability resides and on which version of Windows 10. So the affected versions are:

• Windows 10 v1903

• Windows 10 v1909

• Windows Server v1903

• Windows Server v1909

As stated by Microsoft's advisory, these versions are vulnerable to a critical remote code execution vulnerability in the SMBv3 protocol. The patches are available as of March 12, 2020, but the users who cannot patch their systems are advised to disable SMBv3 compression and block TCP port 445 on firewalls and client's computers as a workaround.

Understanding the criticality using basic score matrices

Taking a look at some base score metrics to understand the criticality:

1. Attack Vector: Network: This is a remotely exploitable vulnerability, and hence attack vector is Network.

2. Attack Complexity: Low: For this attack, no specialized access conditions are required. The complexity is hence Low. An attacker can expect success against this vulnerability easily.

3. Privileges Required: None: This vulnerability becomes more severe as there is no privilege required for this attack. An unauthorized attacker can get around this vulnerability without any access to settings or files.

4. User Interaction: None: This vulnerability can be exploited without any user interaction with the system.

5. Confidentiality: High: There can be a total loss of confidentiality since all the impacted systems the attacker can access data.

6. Integrity: High: There can be a total loss of integrity as the attacker can modify any or all the impacted system files.

7. Availability: High: The attacker can entirely deny access to the resources on the impacted system.

Now let's understand the vulnerability in SMBv3 that causes such critical damage to a user with the vulnerable Windows 10 versions.
The remote attack exists because the Microsoft Server Message Block 3.1.1 (SMBv3) handles specific requests.
Exploiting the flaw requires a specially crafted SMBv3 "Compression Transformation Header" request or response PDU. This flaw is labeled as "SMBGhost" or "CoronaBlue."
So as we know that an attacker does not require any authentication or special access, this attack becomes critical as the attacker can affect the client or the server end communication.

We will be looking at some publicly available command-line tools to check if a system is vulnerable to this attack and see the practical criticality of this vulnerability by remotely performing buffer overflow on the vulnerable Windows system and crashing them by just knowing the IP address of the victim machine.

Defining the goals and prerequisites

So we can define the goals of the exploit in this paper as:

    1. Identifying targets based on SMB protocol
    2. Triggering the buffer overflow by sending malformed PDU

Prerequisites for the practicals are:

    1. A Linux System
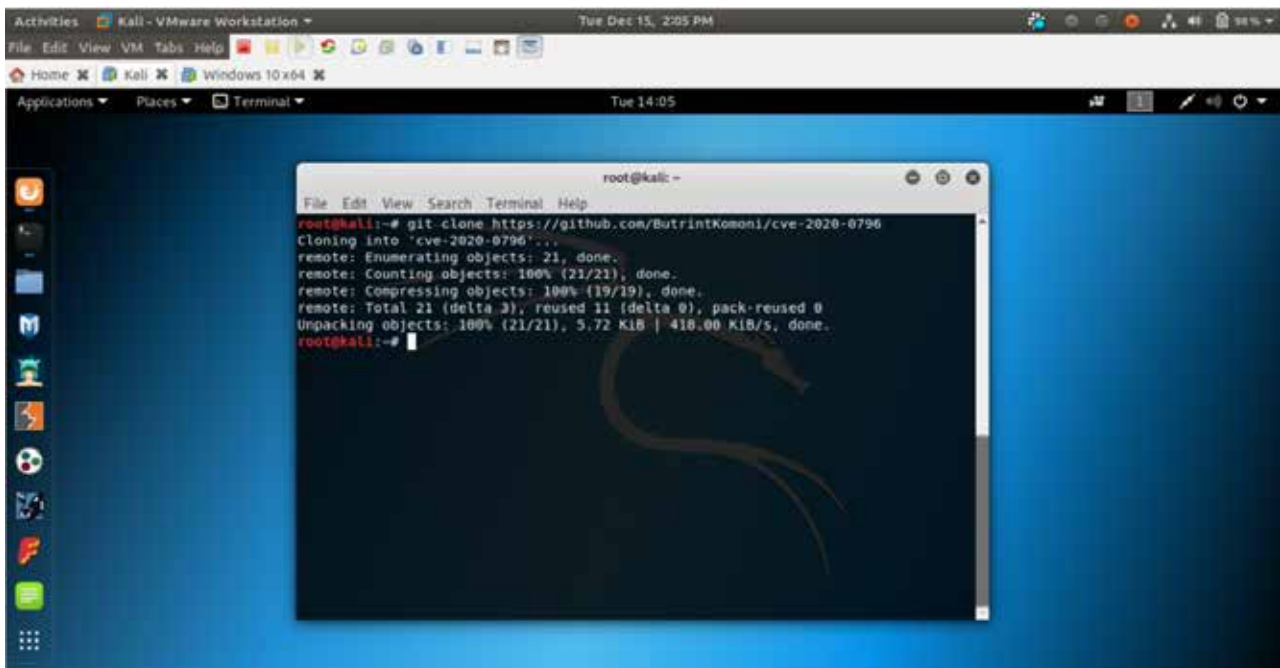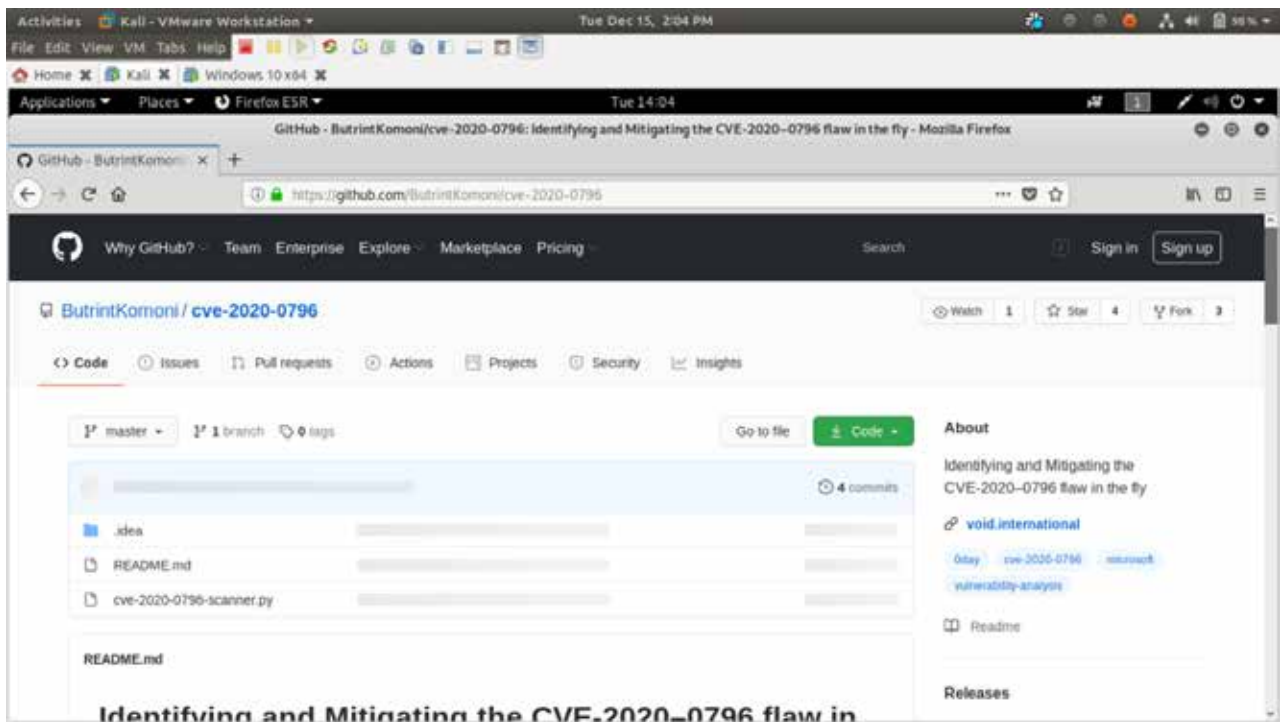   **2. A vulnerable version of Windows 10**

Here we are using Kali Linux to run a tool that helps us identify if the Target machine is vulnerable. Next, we will use another tool to start the attack and see our target machine crash due to buffer overflow. These two operations can be performed easily without any special privileges that make this attack so severe and gets a CVE score of 10.0

Let's start by installing the required tools. First, we are going to install a tool by ButrinkKomoni from Github. This is a straightforward tool to identify a vulnerable machine with SMBv3 running. The tool analyses the packet request negotiate on the SMB protocol and tells if the machine is vulnerable or not.

Installing required tools and exploiting the SMBGhost
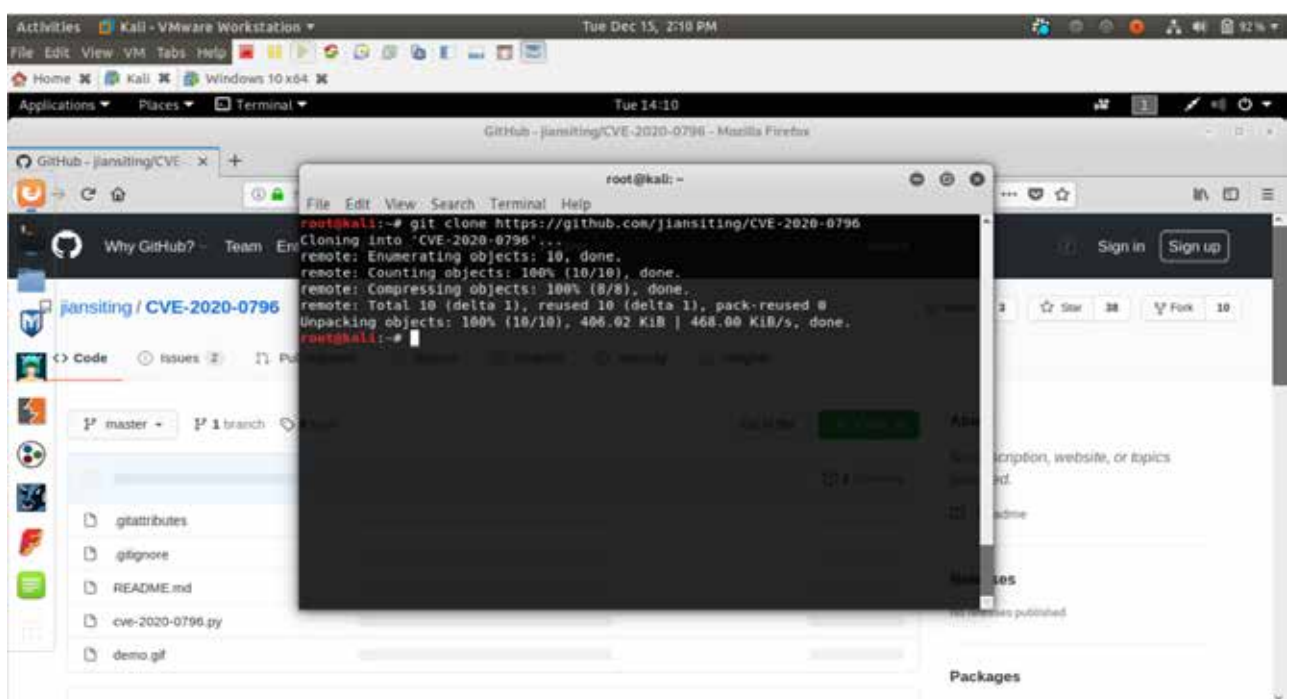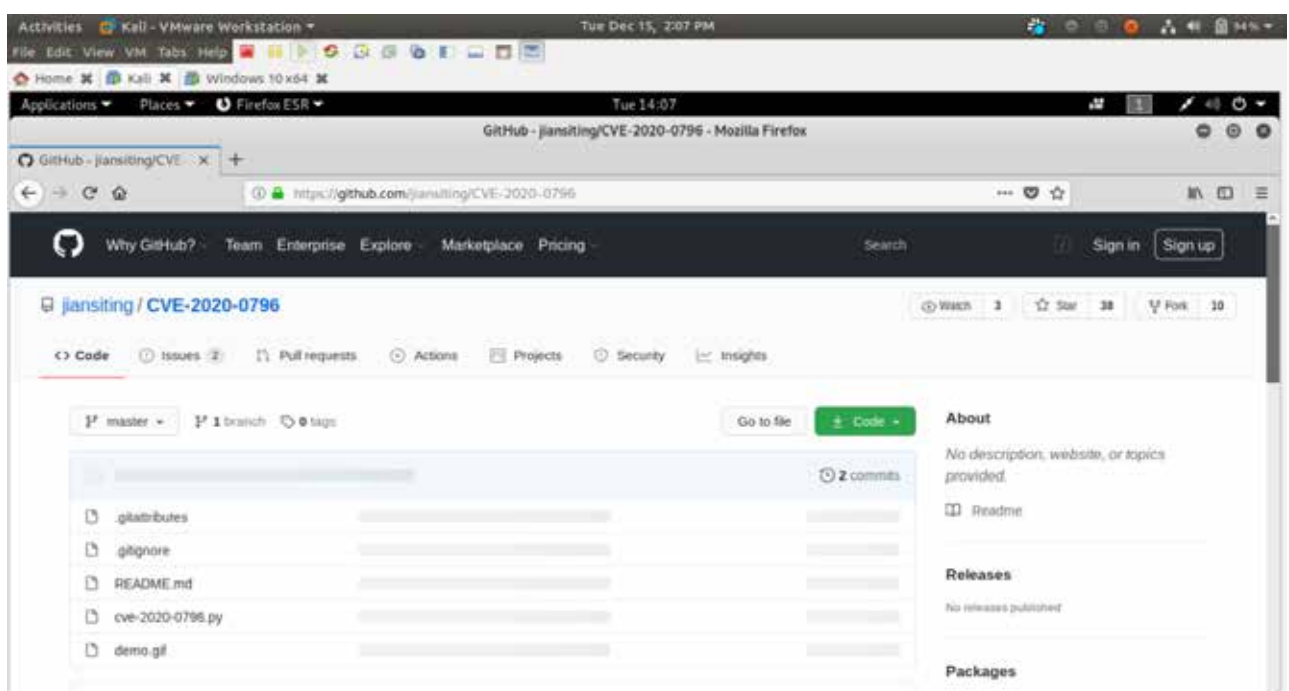
To install the tool,

1. Go to the Github repository: *https://github.com/ButrintKomoni/cve-2020-0796.*
2. Copy the link and open a terminal on your Linux system.
3. Enter the command: git clone following the URL

After the tool is installed, let's move on with the next tool to be installed from Github by jiansiting, which will help us crash the target system by just using the IP address of that system.
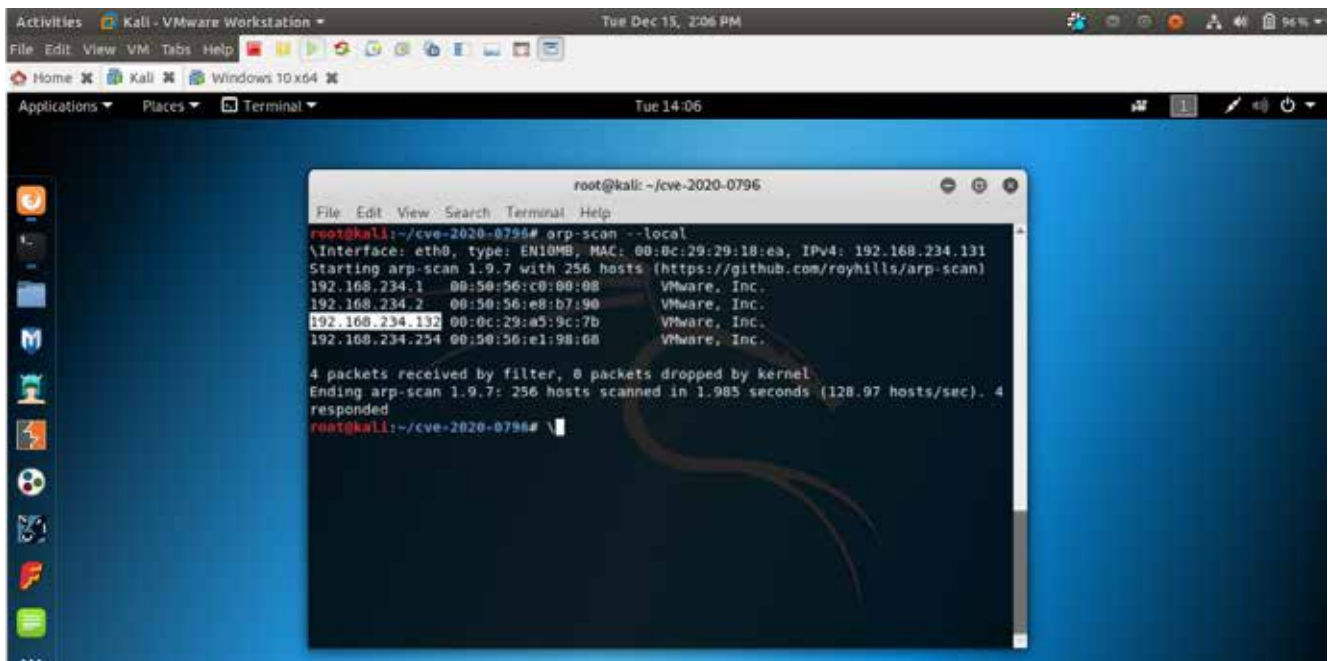
To install the tool,

1. Go to the Github repository: https://github.com/jiansiting/CVE-2020-0796.
2. Copy the link and open a terminal on your Linux system.
3. Enter the command: **git clone** following the URL

After we have installed these tools, we will start with the exploitation. The severity of this attack can be imagined as an attacker just needs the IP address of the target machine, and he will be able to crash or remotely access the victim machine resources.

We will first find the target machine's IP address from our host machine using the command: **arp-scan --local**, as the machines are on the same network.
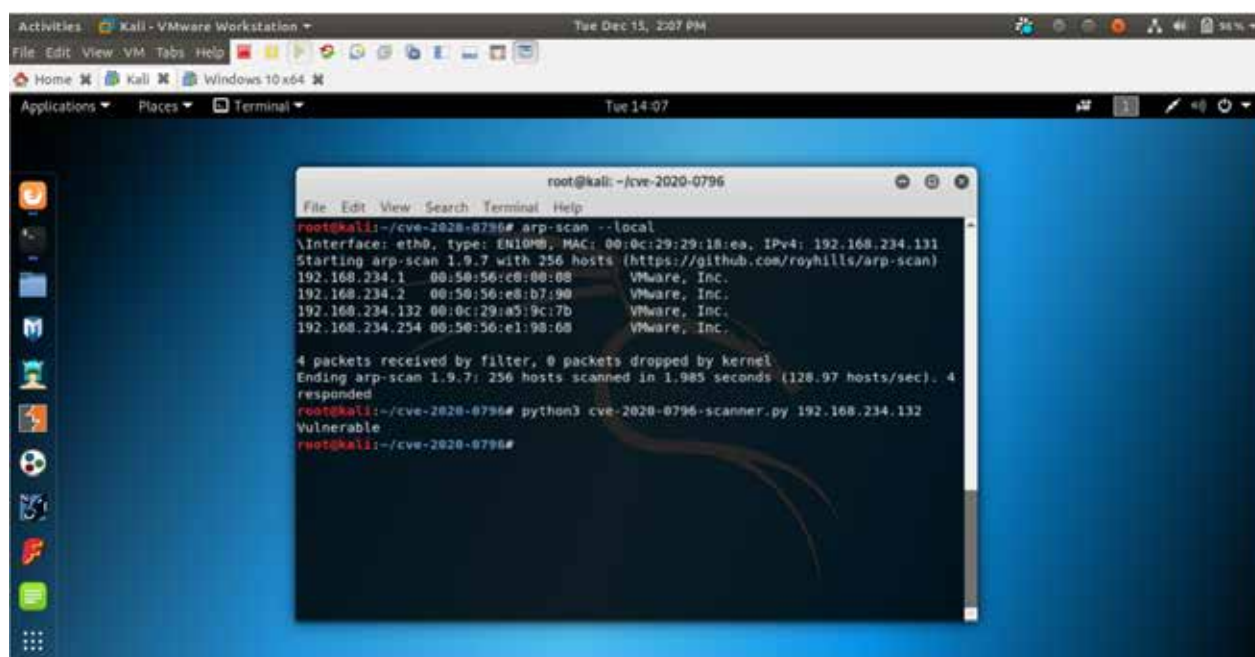
The target machine's IP address is, in our case: 192.168.234.132. So we will use this IP address to scan and crash the victim machine.

To scan the target, we need to go inside the directory of the scanner tool. There we will find a scanner python file.

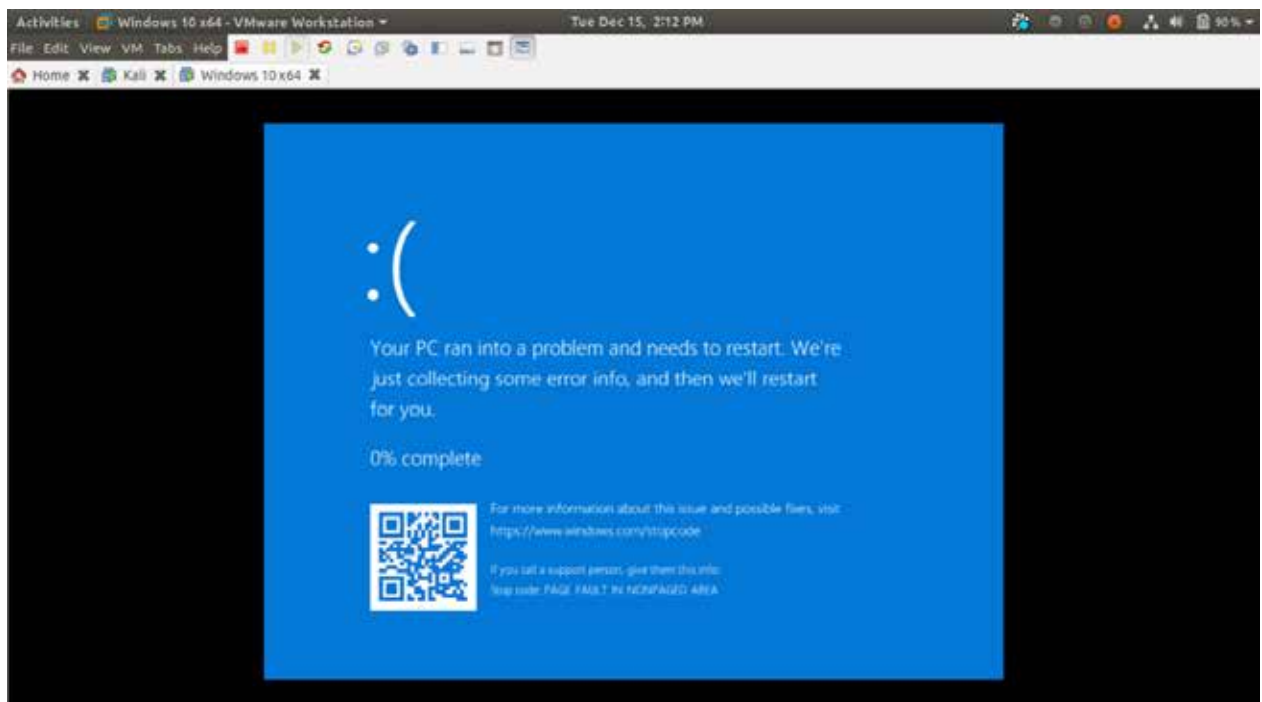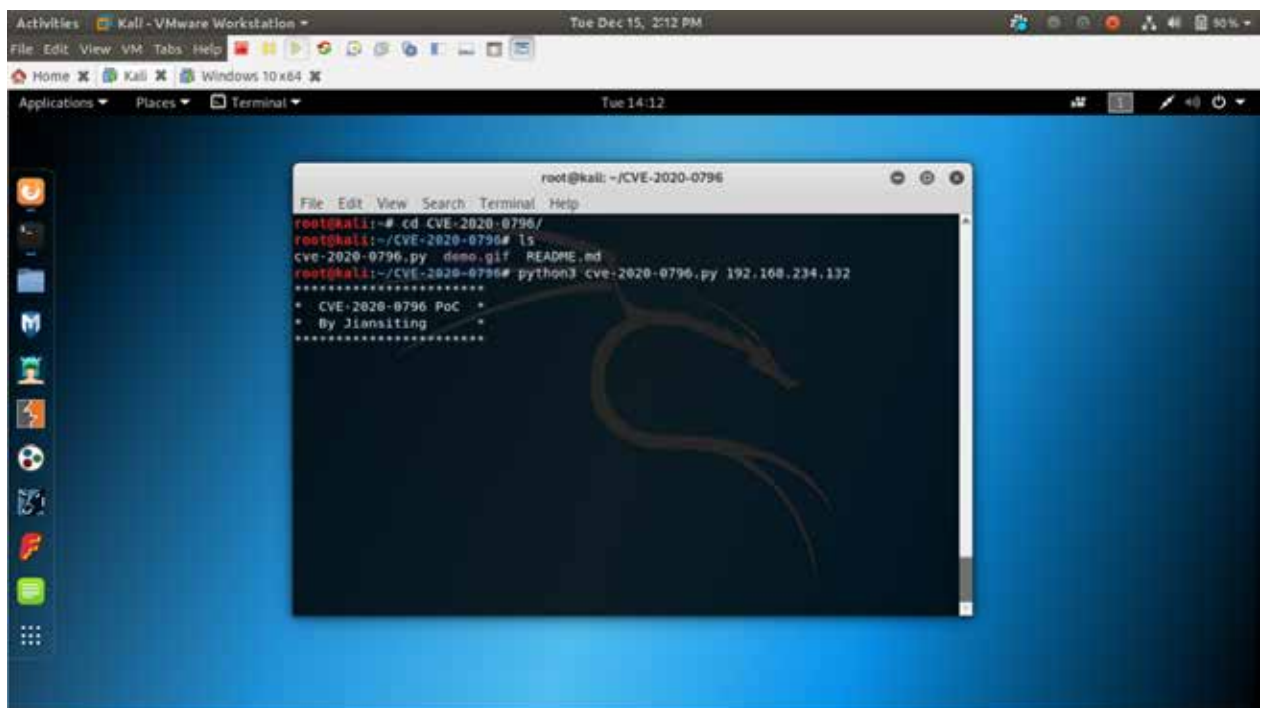To run the scan, enter the command: **python3 filename ip_address**

After running the command, you simply receive a message stating that the target is vulnerable. This means that we can move on with our second tool to crash the vulnerable target machine by just using a simple command and the IP address

Go to the directory of the second tool, and you will find a python file. Now enter the command: **python3 filename ip_address**

Let this command run and notice the changes in the target machine. In a few minutes, the target machine crashes, and a blue screen of death appears. target machine crashes and a blue screen of death appears.

So we have successfully exploited the vulnerability in the SMBv3 of Windows 10 and crashed a system remotely using two simple tools and the target IP address.

How to mitigate this vulnerability

The workaround should be applied to all servers and workstations that share this vulnerability. Also, make sure that firewall rules on the border firewall and endpoints prevent (block) inbound and outbound connections to the vulnerable service (445 TCP) if applicable.

1. Disable SMBv3 compression
Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters"
DisableCompression -Type DWORD -Value 1 -Force

2.Block inbound and outbound SMB
Consider blocking outbound SMB connections (TCP port 445 for SMBv3) from the local network to the WAN. Also, ensure that SMB connections from the internet cannot connect inbound to an enterprise LAN.

SAFE
SECURITY