**PENETRATION TESTING**

Penetration testing is the practice of testing a computer machine, network, or web application to discover security vulnerabilities/weakness that an attacker could exploit and do unauthorised intrusion. It is also termed as pen testing or ethical hacking.

**METASPLOIT**Metasploit is one of the commonly used penetration testing frameworks. It has made hacking way simpler than it used to be. It is a very important tool for both defenders and attackers.

It is a collaboration between the open source community and Rapif7. It helps security teams to do more than just verifying vulnerabilities, improving security awareness, and managing security assessments. It basically empowers and arms defenders to always stay a step or two ahead in the game.

In our project we will be using this framework to learn to scan the vulnerabilities of the target machine (i.e. Metasploitable 2) and exploit the same.

**METASPLOITABLE 2**

Metasploitable 2 is a vulnerable Ubuntu Linux virtual machine that can be used as a target for attacks, security testing and for practicing common penetration testing techniques.

In our project we will be using this as the target machine which will be attacked upon by the Metasploit framework.

**Some Important key terms:**

**Exploits**: The module that will take advantage of the system vulnerabilities and it will install a payload on the system to gain access.

**Payloads:** The files left on the exploited system which give the attacker the control over the systems. The attacker basically gets to own the target system.

**Auxiliary:** This provides you with unique type of attacks e.g.: dos functionality, robust tools, scanner, etc

**Nops:** It stands for “no operation”. It causes a systems processor to stop doing anything for an entire clock-cycle: good for (attacking)system to run a specific file after the buffer exploitation.

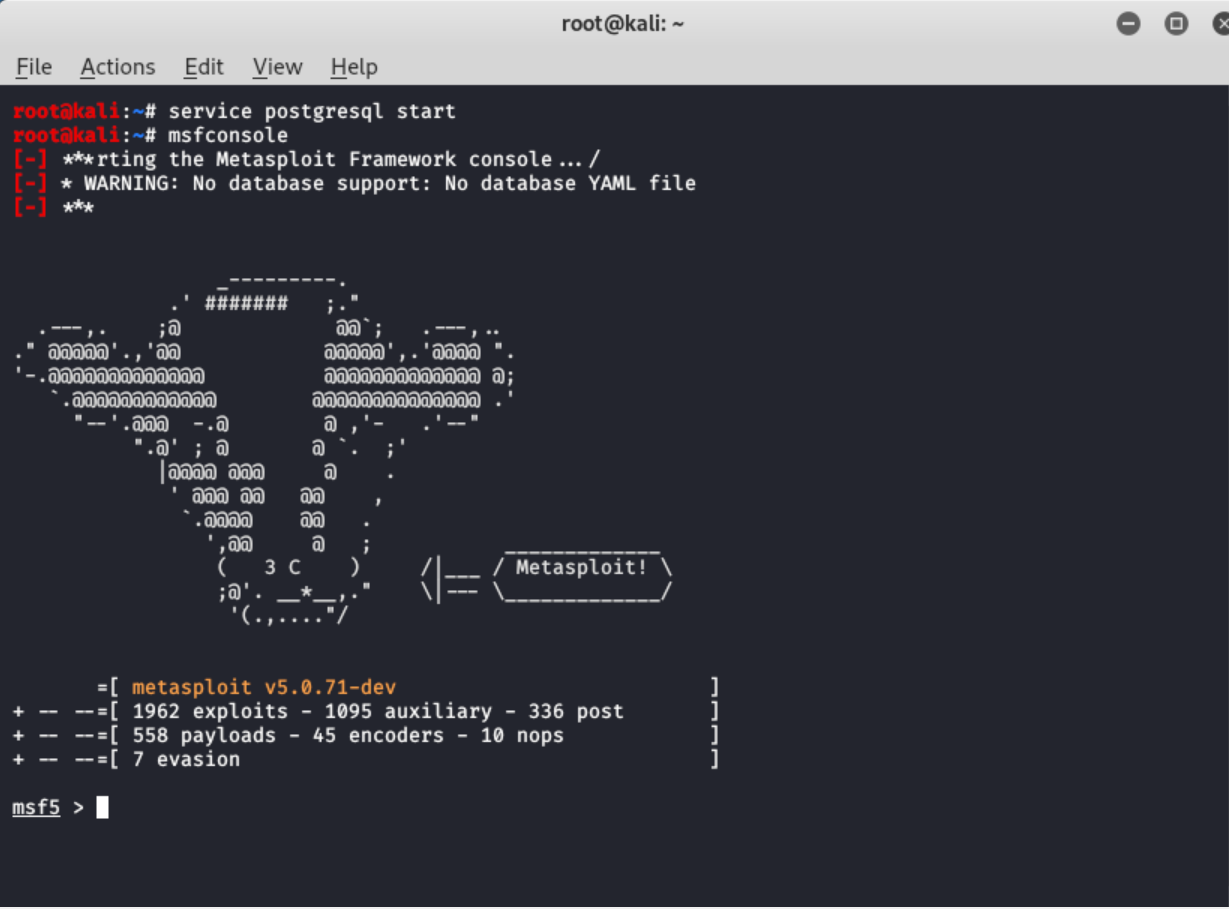
**Post:** It is used after the system has been exploited, allows you to perform attacks after the target system has been owned.

**Encoders:** It means to re-encode payloads which help getting past security systems like antivirus.

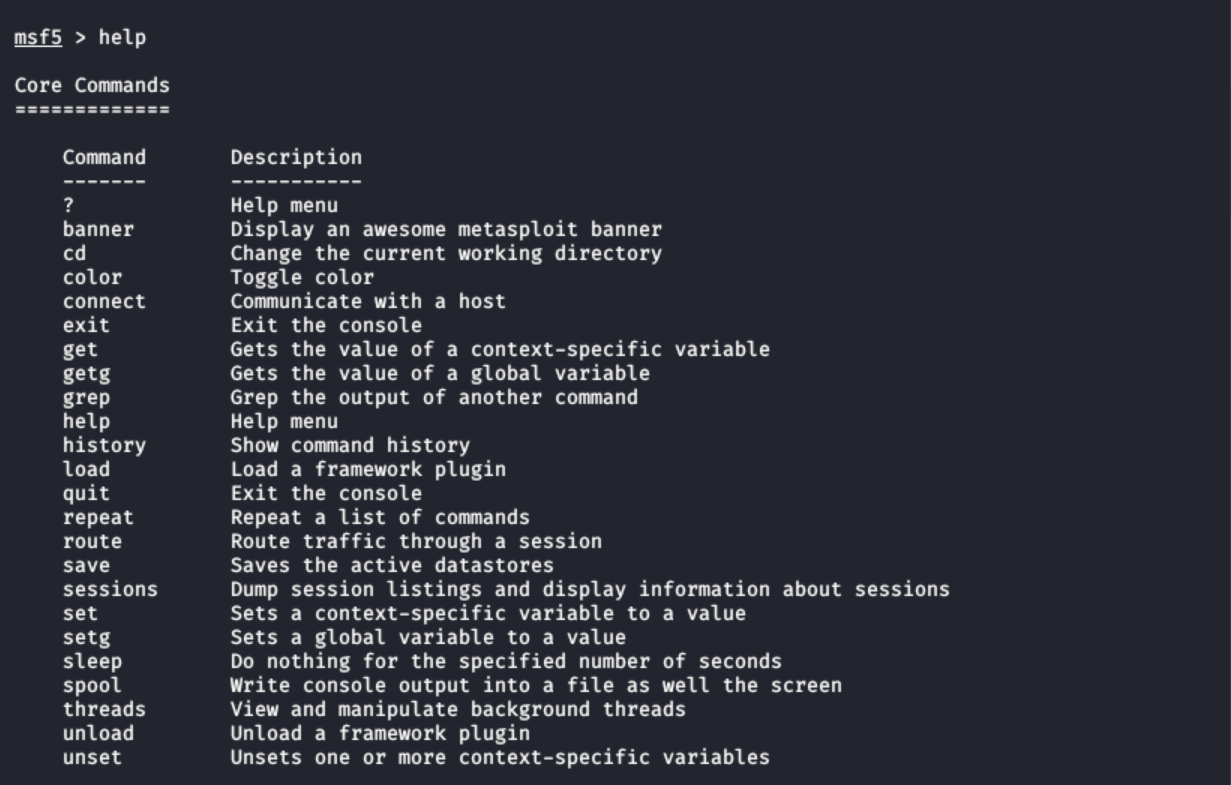
**EXPLORING THE METASPLOIT FRAMEWORK**

**We tried to run few basic commands like help, use, show all, etc**

**Starting MSF console:**

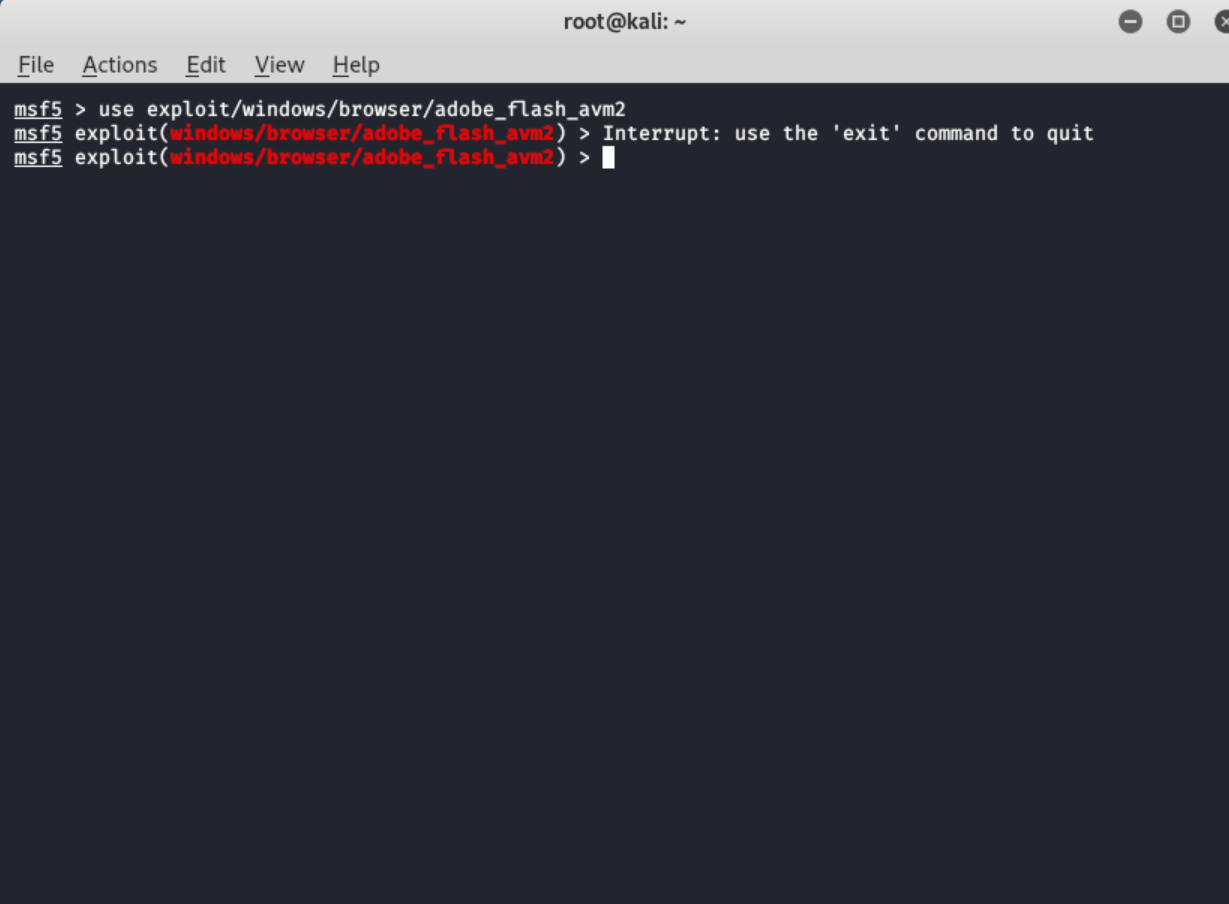


**COMMAND: help**

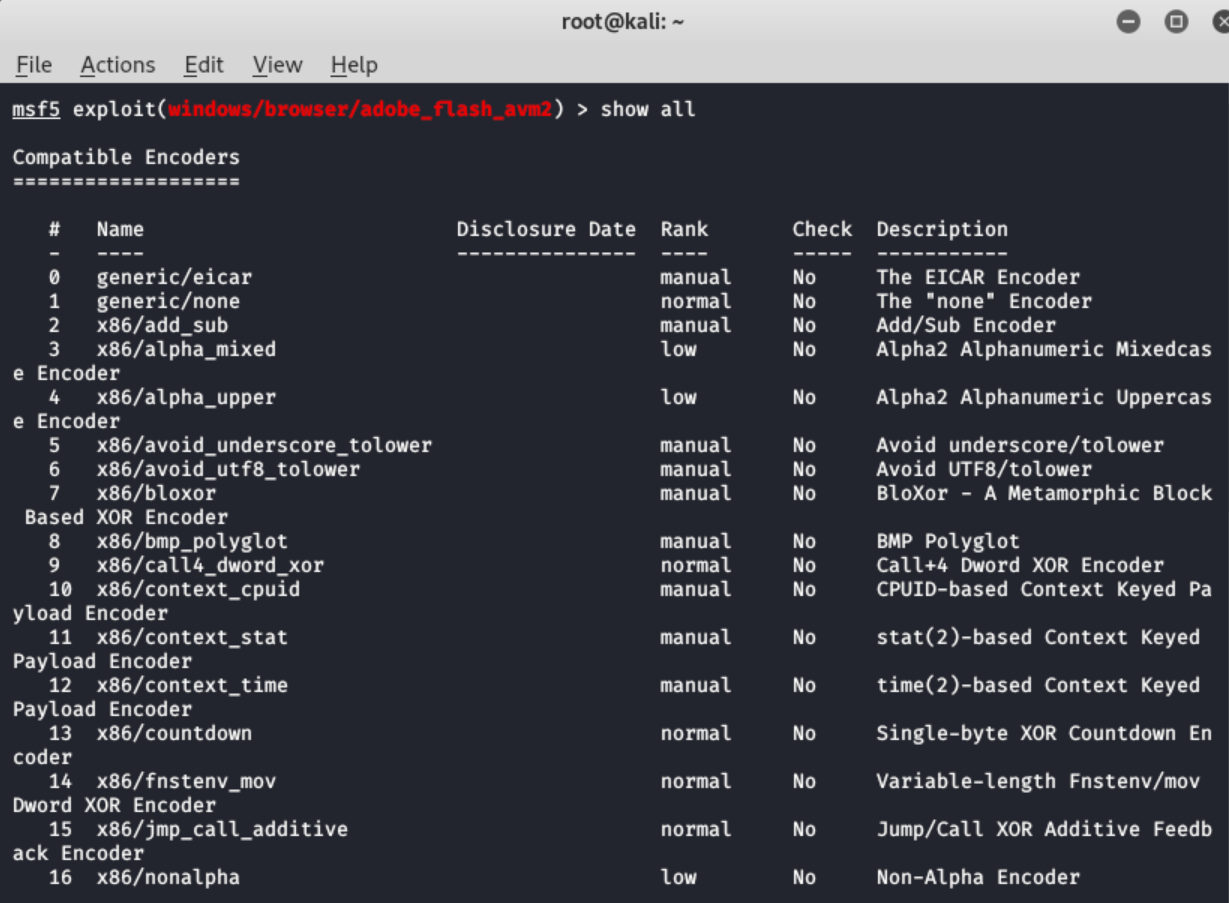


**Command: use**

Here, “use exploit/windows/browser/adobe\_flash\_avm2”: used to exploit adobe flash plugin.



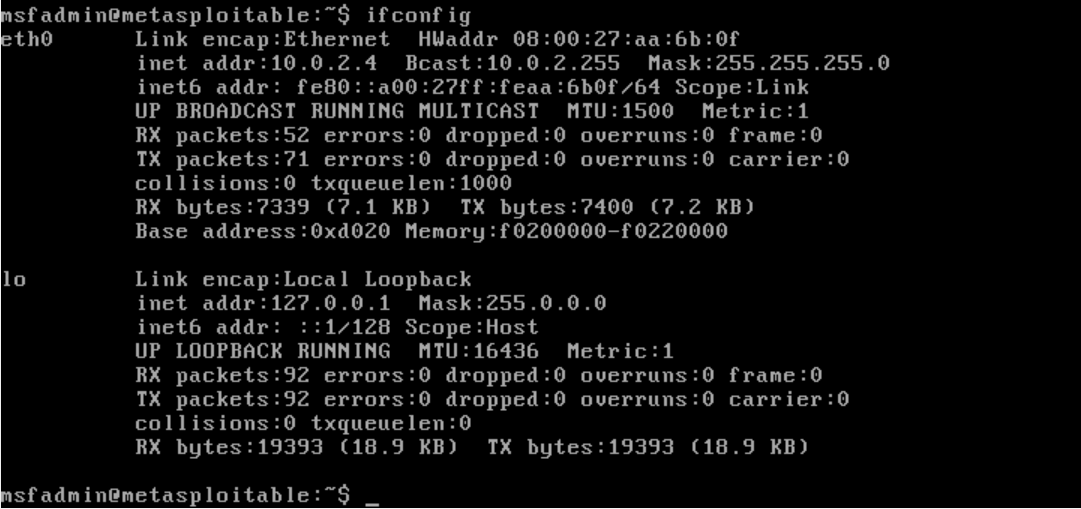
**Command: show all: this is used to give the information of a particular module**



**The typical process used by attackers in exploitation and privilege escalation:**

* **Find an open port and running service:** As in our case we are using metasploitrable 2 as the target machine we will have to get its IP address manually, for that we will run the command “ifconfig” in the command shell of metasploitable 2 machine.
* **Determine that the service has a vulnerability:** This step basically comes under information gathering in which network scanning tools like Nmap are used. In our project we have used Nmap. Nmap uses the IP packets to identify all the devices connected to the network. It also provides information on the services and operating systems they are running on.
* **Determine whether that vulnerability has an exploit available:** After knowing what all vulnerabilities are there in the device we search if there are any exploit corresponding to the vulnerabilities.

**IP ADDRESS OF THE METSPLOITABLE 2 MACHINE:**



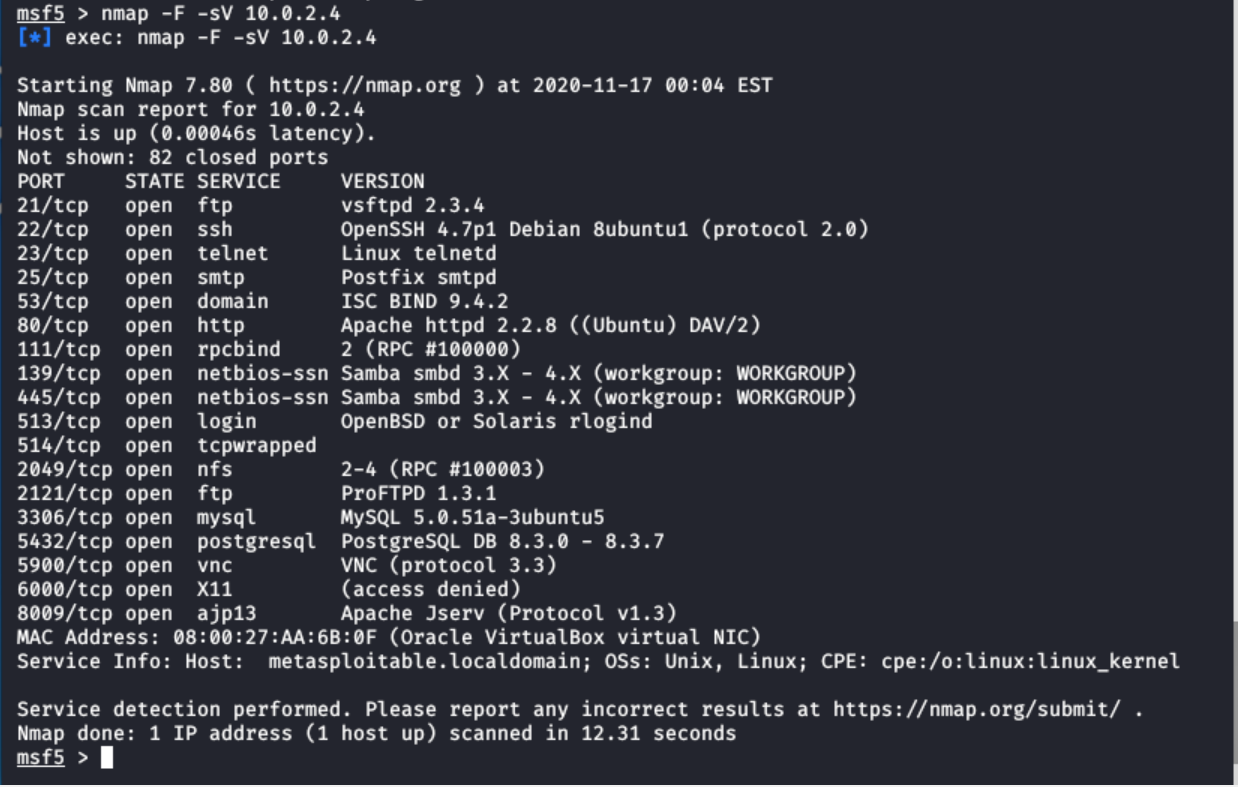
**BASIC EXPLOITATIONs**

For the project we have perform three basic exploitations:

* 1. Related to the FTP port
  2. Related to the SSH port
  3. Related to the SMB port

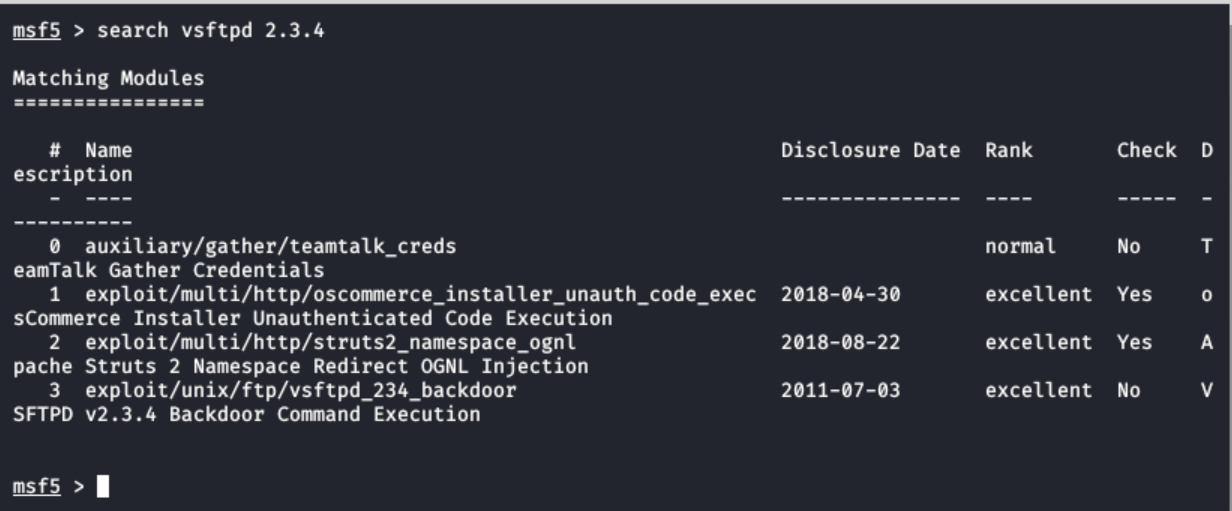
**FTP BACKDOOR COMMAND EXECUTION**

Nmap to scan the vulnerabilities: (scanning all the ports)

It shows the ports that are open e.g. the ftp port.

Now we are going to search for an exploit on the ftp port using the “search” command. After performing the search command, we will get a list of the exploit with their rank, disclosure date, etc.

e.g. - exploit/unix/ftp/vsftpd\_234\_backdoor – a backdoor command execution

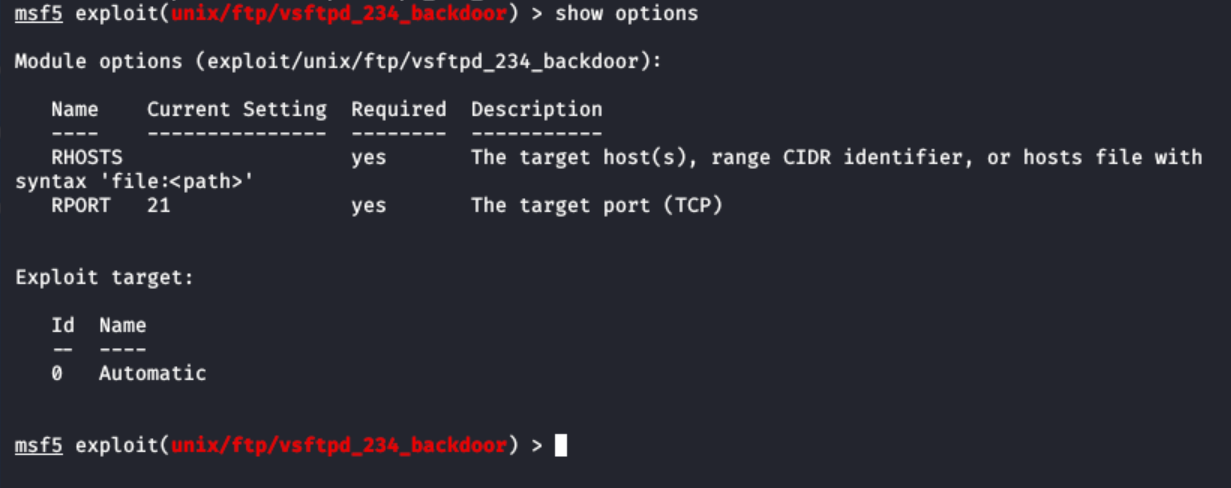


Now we will use this exploit:

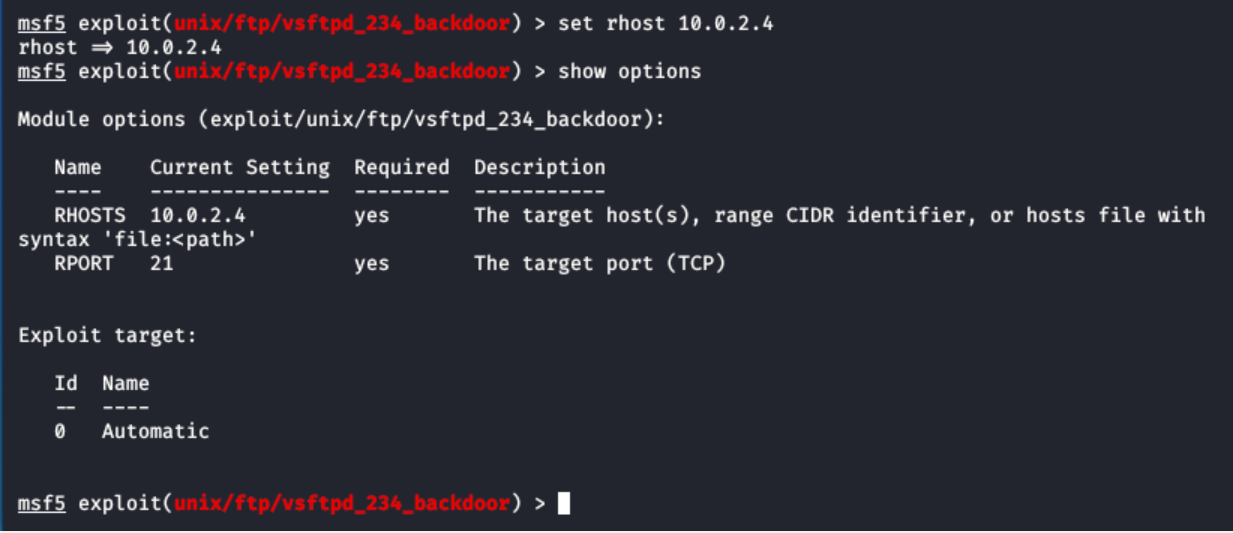


Now we will be showing the options available with the exploit:

(with different exploits and different modules, we will have different options)



Now to run this particular exploit we need to set the option – “rhosts” which will be the ip address of our target machine i.e. metasploitable 2



Now we have set all the option field and we can further move on to run the “exploit” command.

In case “exploit” command will basically open the backdoor to the target machine. It has opened the command (/reverse) shell (Linux sys).

**NEW TERM ALERT:**

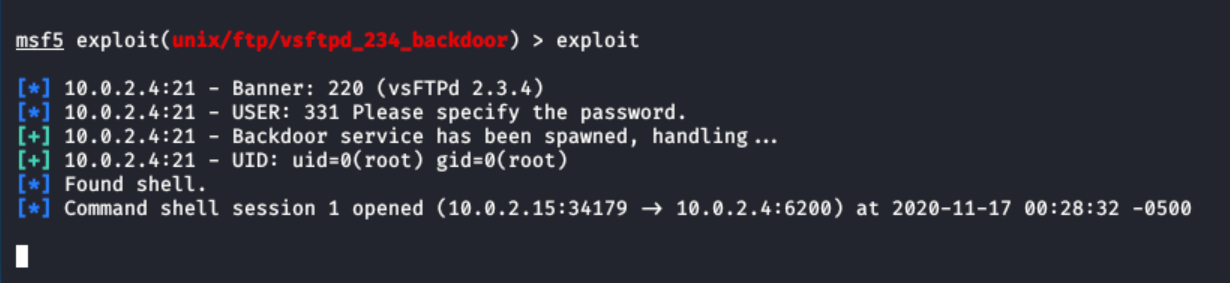
**WHAT IS REVERSE SHELL?**

**Ref to:** <https://www.netsparker.com/blog/web-security/understanding-reverse-shells/>

A reverse shell is a shell session established on a connection that is launched from a remote machine, not from the local host. Attackers who successfully exploit a [**remote command execution vulnerability**](https://www.netsparker.com/blog/web-security/code-injection/) can use a reverse shell to achieve an interactive shell session on the target machine and continue their attack. A reverse shell (also called a connect-back shell) can also be the only way to gain remote shell access across a NAT or firewall.

Here, we have the access to the server

We can perform the Linux commands in here.



We can use the Linux commands to do our stuff on our target machine.

e.g. here, “ls” command is used to list out the files and we can do anything to the files, we can even create new files, etc

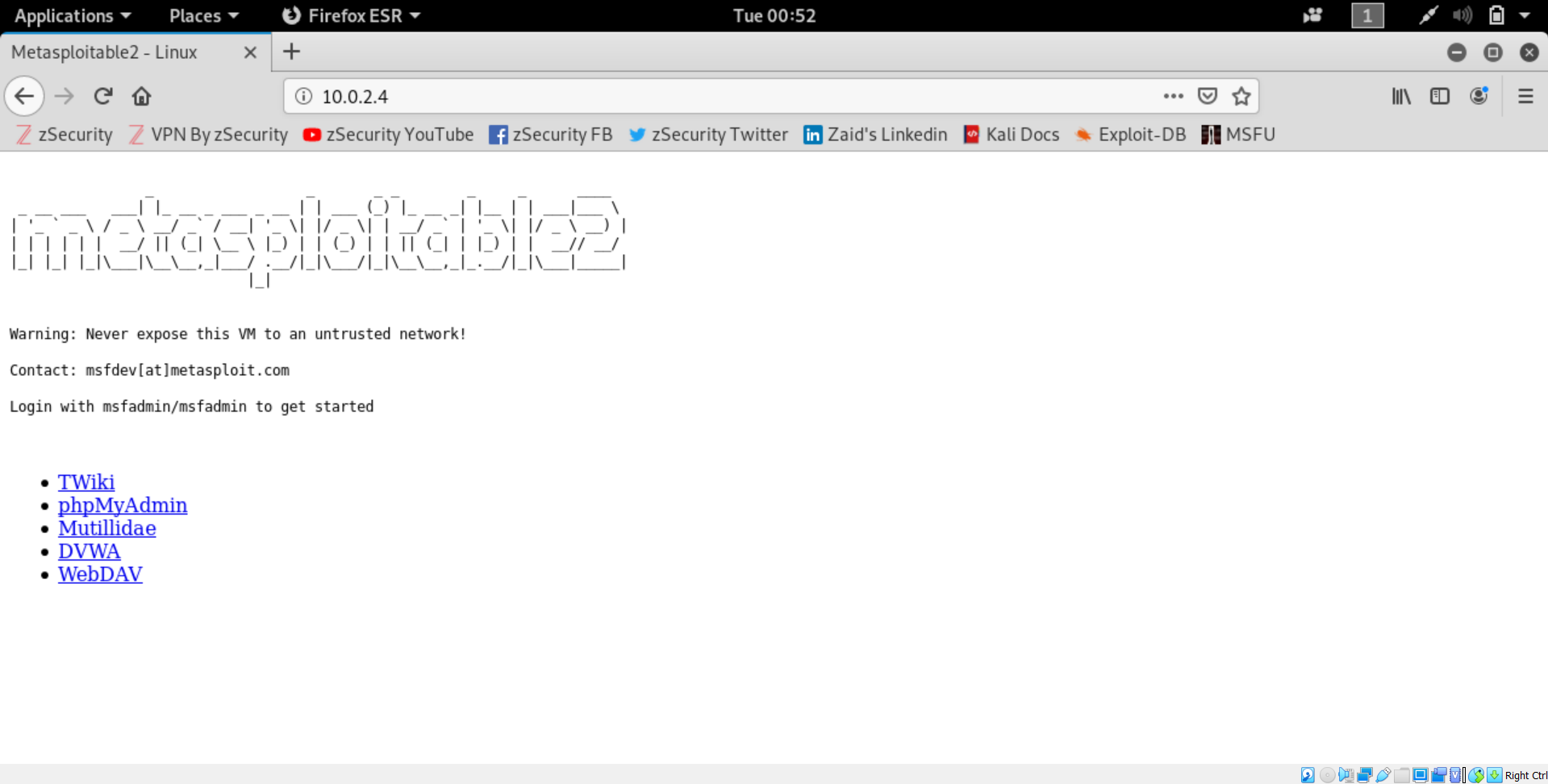


**HACKING WEB SERVERS**

Metasploitable is going to act as the web server:

When we enter its ip address in the browser we can access it as a server

So, this means there is no security anyone can login into the server



**NEW TERM ALERT:**

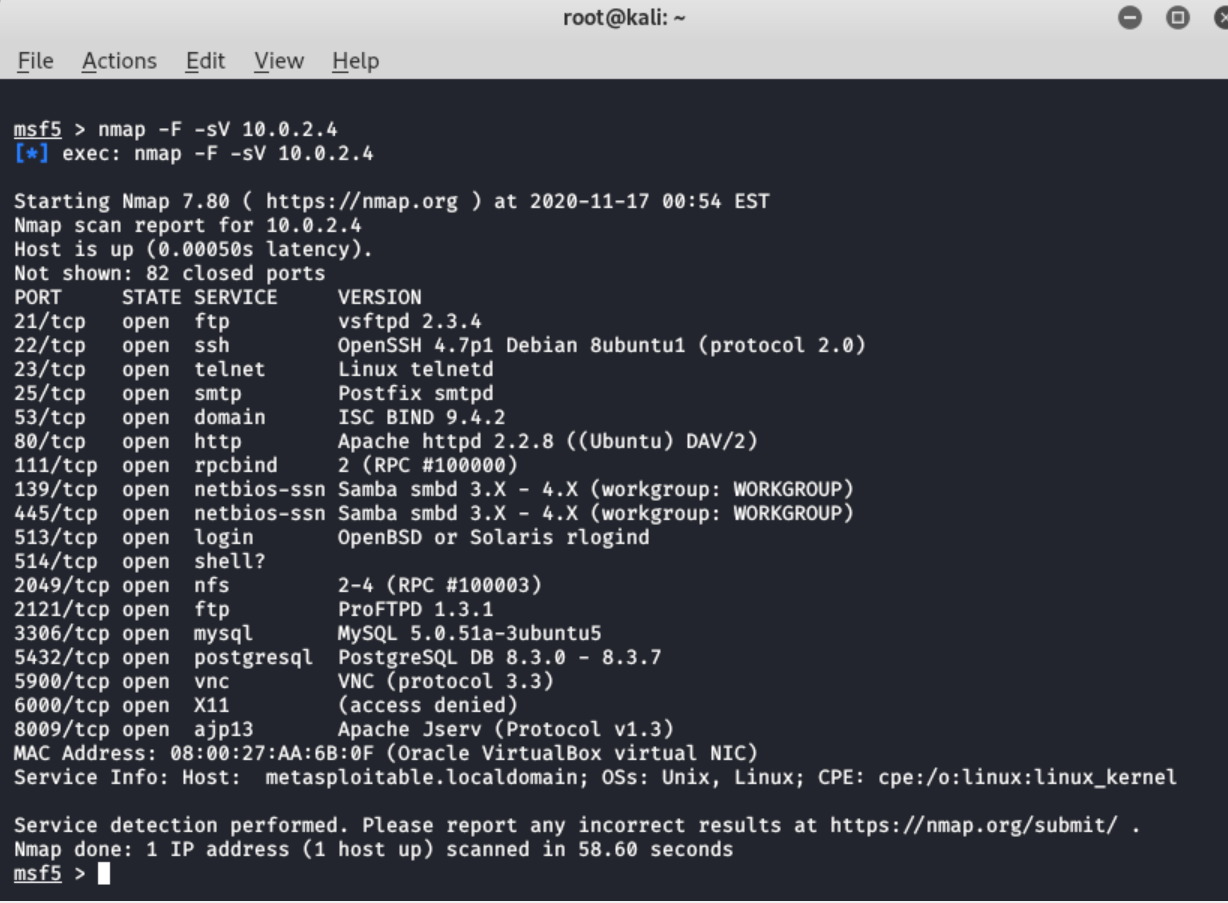
**WHAT IS SSH?**

SSH (SSH client) is a program used for logging into a remote machine and for executing commands on a remote machine. It is intended to provide secure encrypted communications between two untrusted hosts over an insecure network. The default port for Secure Shell (SSH) is port 22. It listens for the incoming connections on this port.

Because SSH provides remote access into systems, it is critical that access be tracked and controlled. Since many organizations do not have centralized oversight and control of SSH, the risk of unauthorized access is increasing.

SSH Is essentially a secure shell which means we can connect to the server (in our case Metasploitable 2) granted we know the username and password

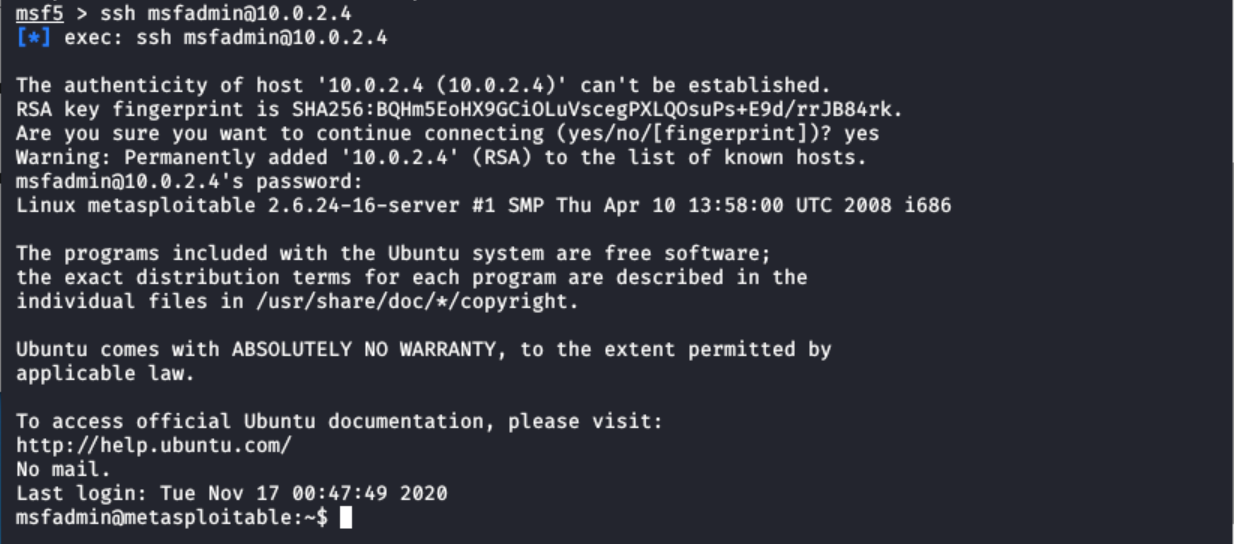
Doing the nmap scan: the ssh port is open

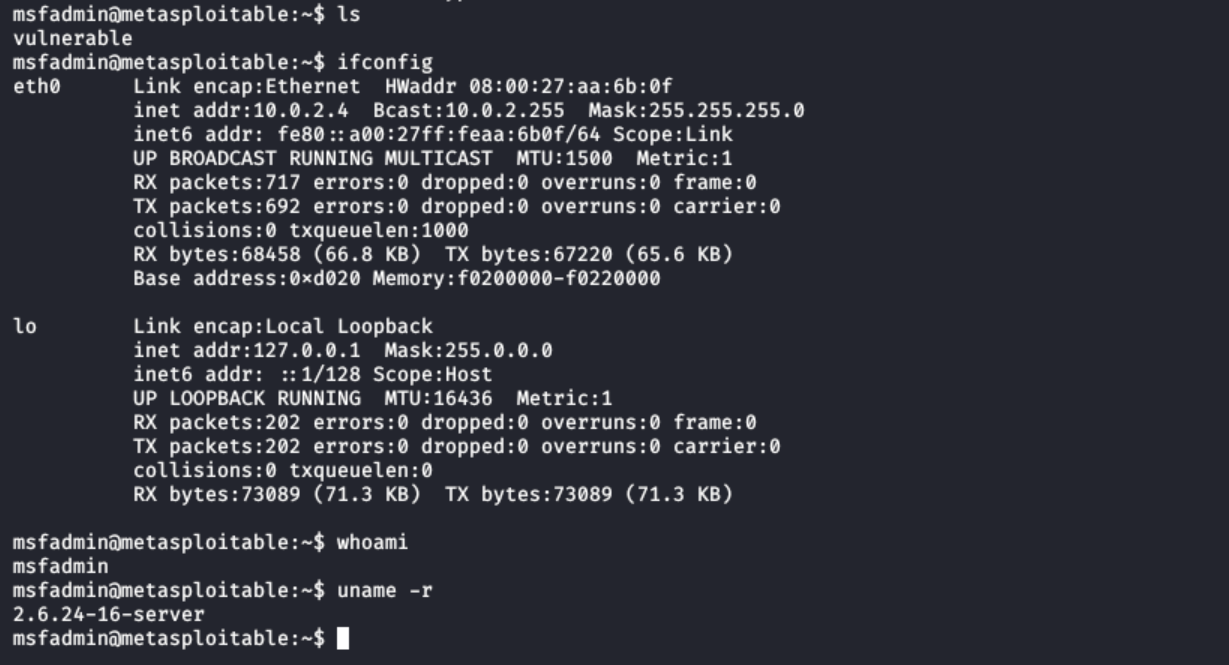


Connecting to the ssh port of the target machine

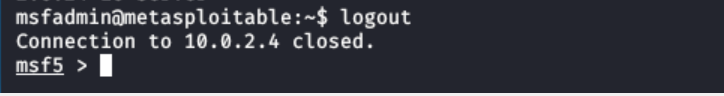
(For this we need to know the RSA key: the password of the target machine as stated before)

After executing the above command, we are in the metasploitable machine: we can list the files, and view ip address, etc





To close out the connection



**SAMBA COMMAND EXECUTIONS**

Samba “username map script” Command execution

**Ref to:** <https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script/>

This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands. No authentication is needed to exploit this vulnerability since this option is used to map usernames before authentication!

This exploit does not have a backdoor installed already, it’s pure vanilla, to exploit anything here we would need to use a payload because it does have a buffer overflow and the only way we use a buffer overflow is by using a payload.

(In our case we did not need to set the payload because it was already set)

**NEW TERM ALERT!**

**What is buffer Overflow**?

A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold or when a program attempts to put data in a memory area past a buffer.

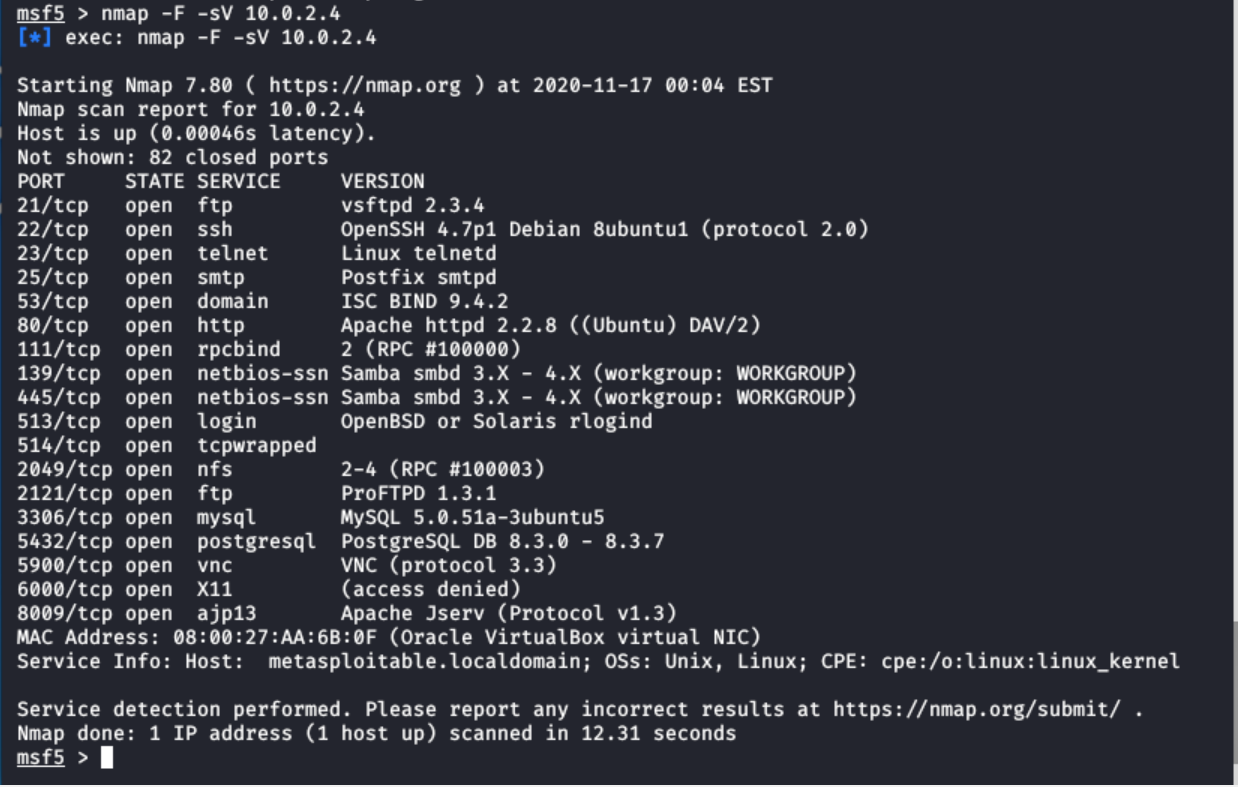
Attackers use buffer overflows to corrupt the execution stack of a web application. By sending carefully crafted input to a web application, an attacker can cause the web application to execute arbitrary code – effectively taking over the machine

**What is pure vanilla**?

Something used without any customizations or no updates are applied to them.

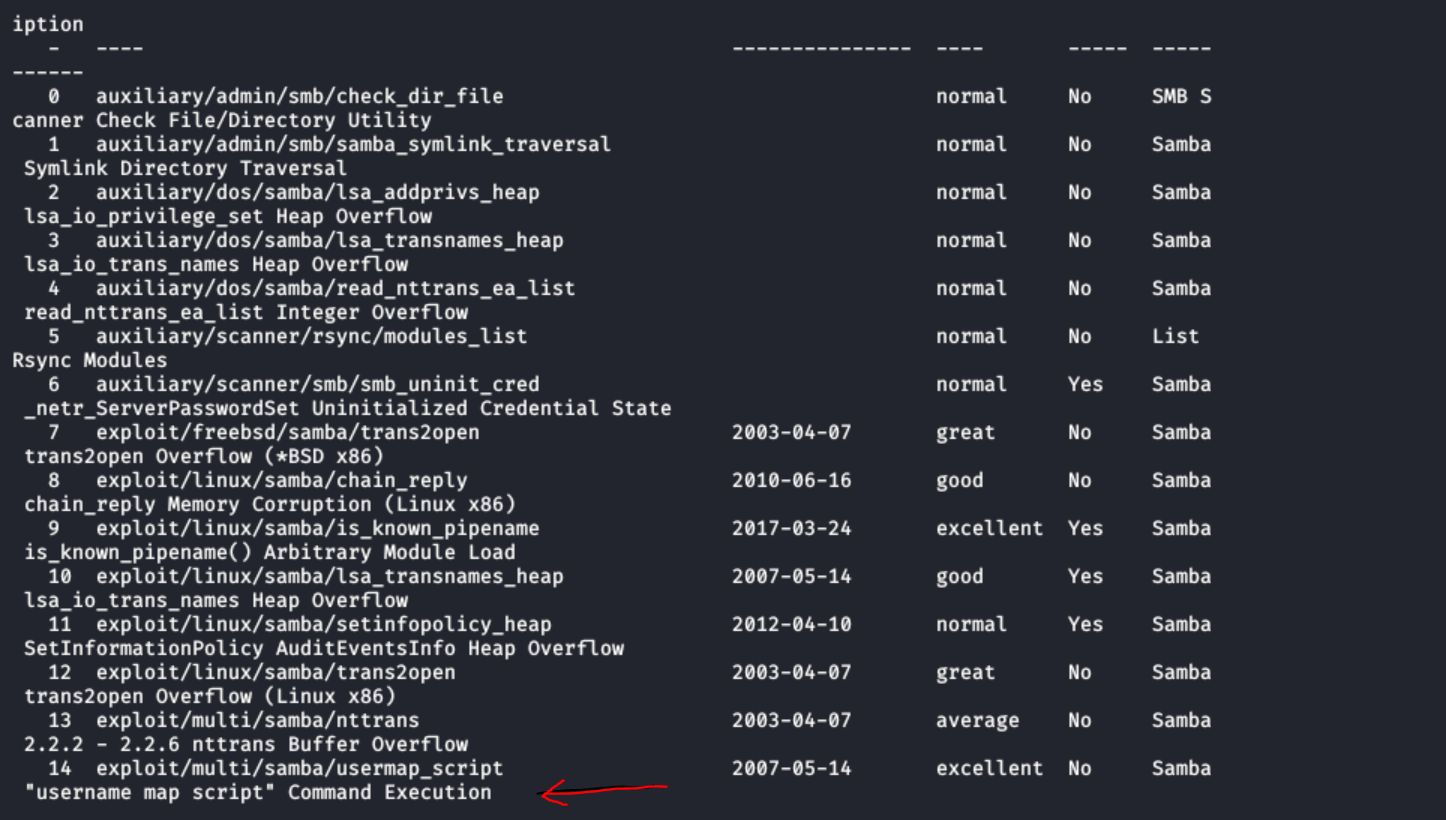
So basically, the exploit that we are going to use in this case is pure and raw and no customisations are made to it.

Doing the nmap scan:



Now to get the list of the exploits we will use the “search” command:

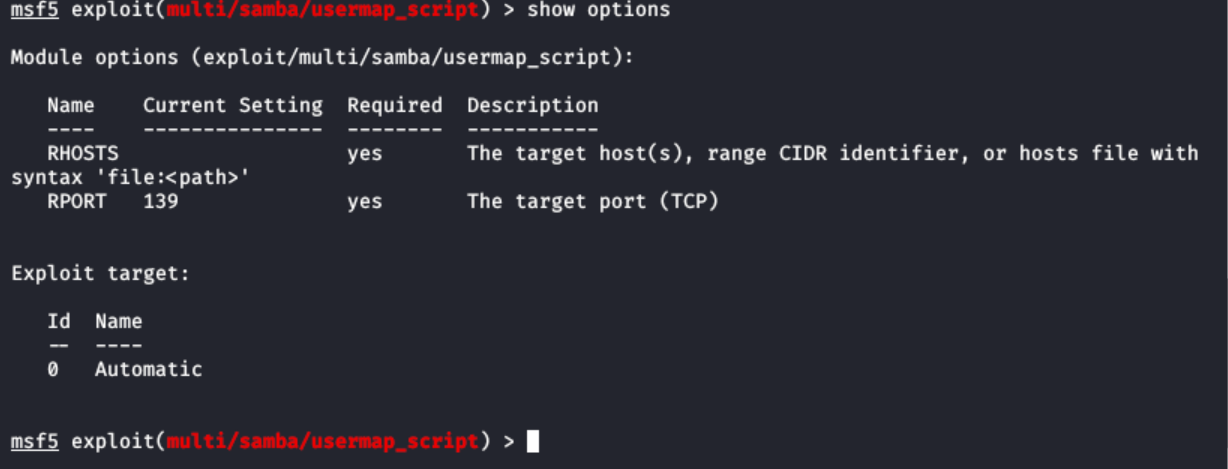


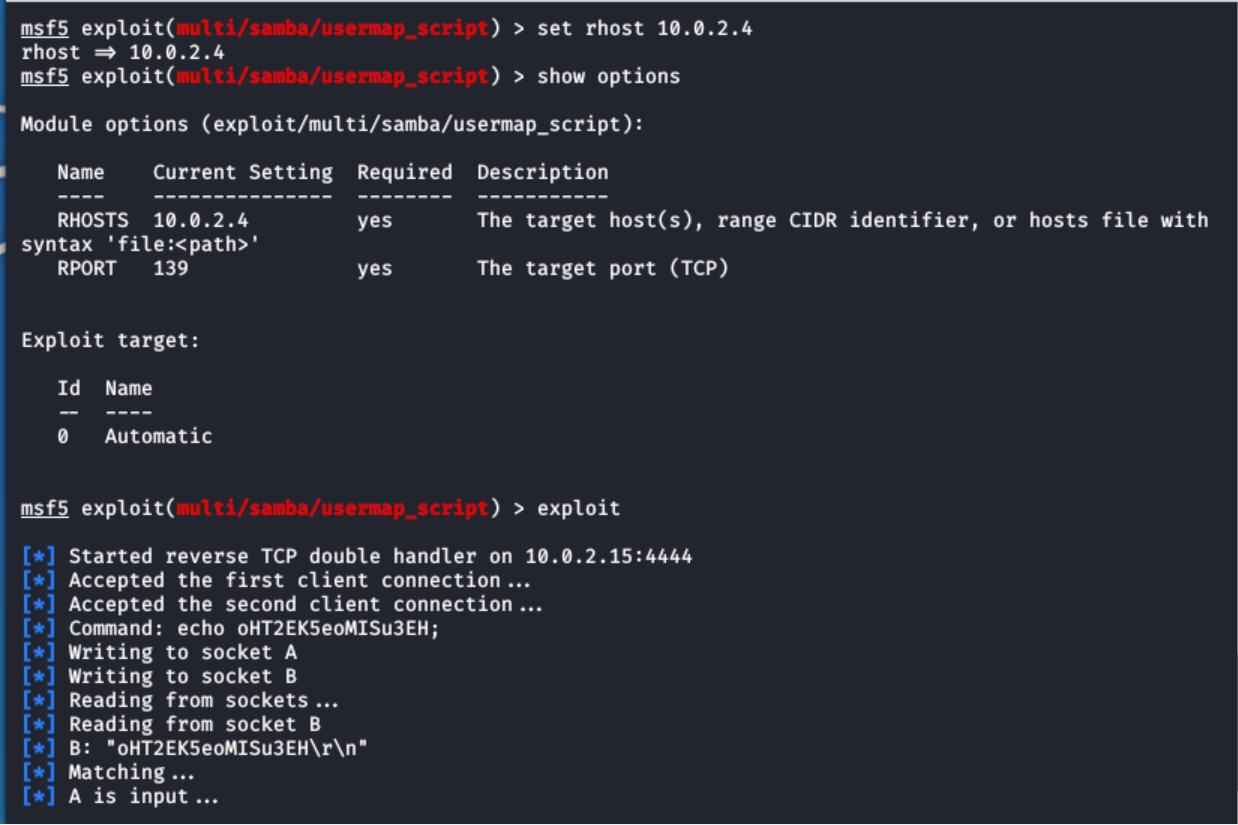


In this case we will be using the “username map script” exploit as stated before

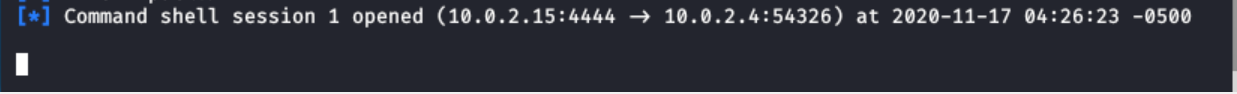


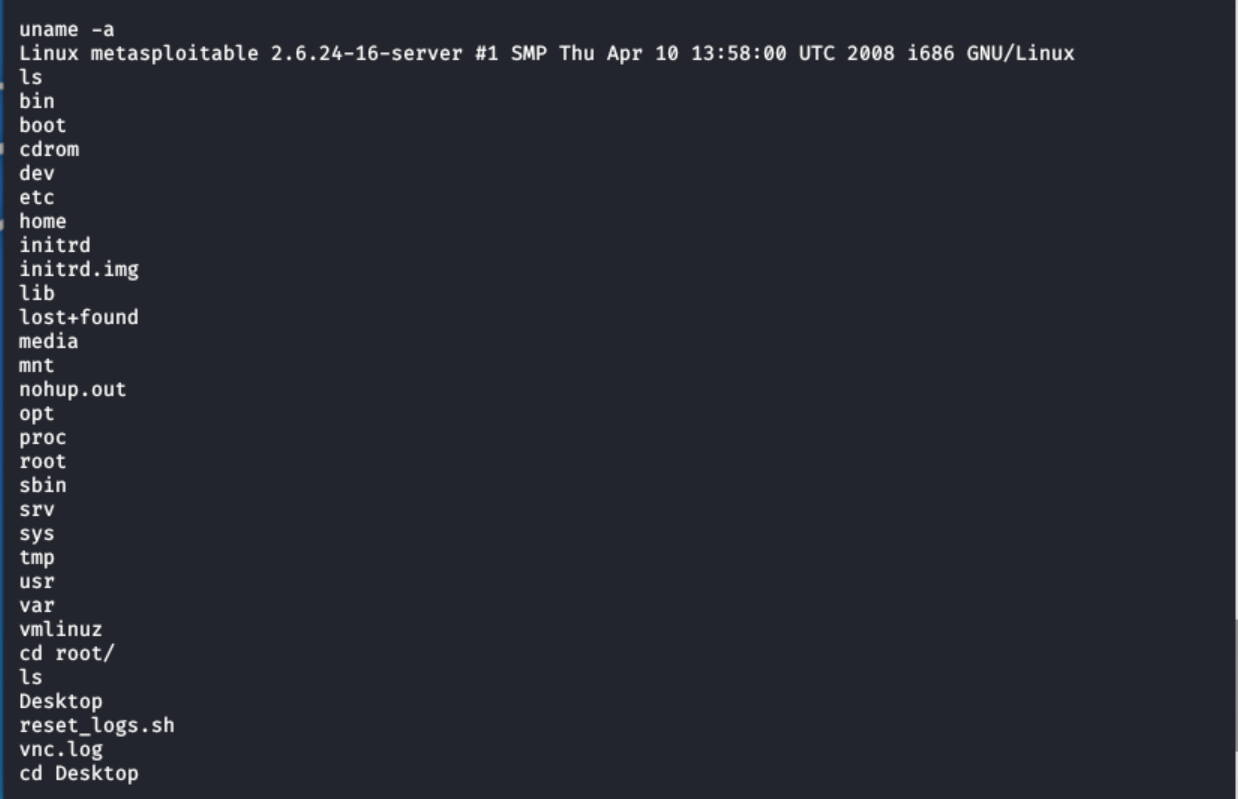
To set the options that are required for this exploit we will first list them out using the “show” command and then set the required fields using “set”





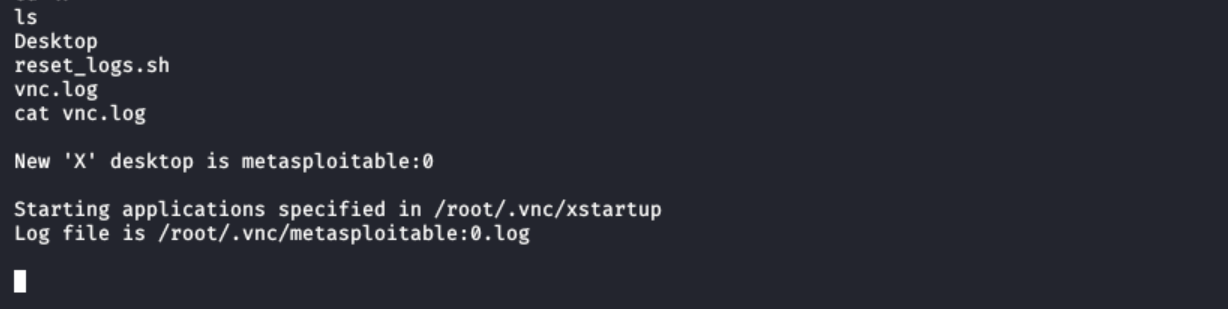
The Reverse shell has started and now we have the access to the machine





Now we have the access we can create files in the target system as well,

**cat command** allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.



**EVADING ANTI VIRUS SOFTWARE WITH VEIL EVASION**

**ANTIVIRUS BYPASS**

We also explored about how we can generate payloads that can bypass the antiviruses.

**(Generating a payload)**

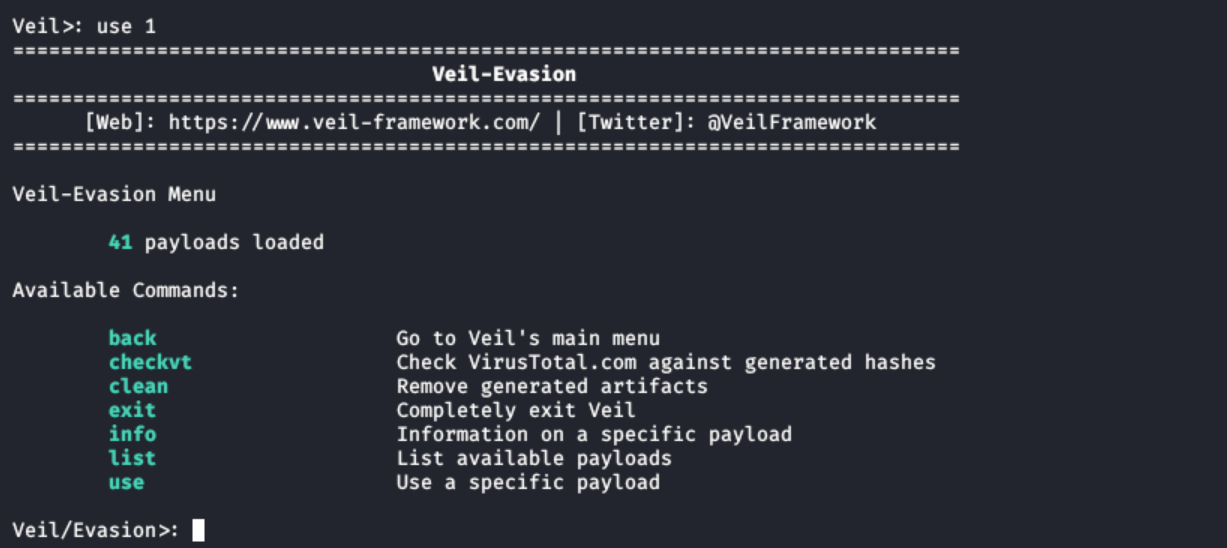
**Antivirus** software is one of the oldest and the most ever-present security control against malware and various types of malicious software. In the past it was focused on blocking viruses only, then eventually evolved into blocking all sort of other malware. Lately, however, attacks have been growing more sophisticated, specifically trying to stay under the radar using administrator toolkits and evading virus signatures to bypass these formerly effective standalone security control. At this point, antivirus technology has been outpaced by endpoint detection and prevention technology that is behavioural in nature and uses virtualized malware detonation technology.

**Ref to:** <https://www.nopsec.com/pen-testing-toolkit-tools-techniques-used-to-evade-antivirus-software/>

**Veil-Evasion** is another popular framework written in python. We can use this framework to generate payloads that can evade majority of Antiviruses.

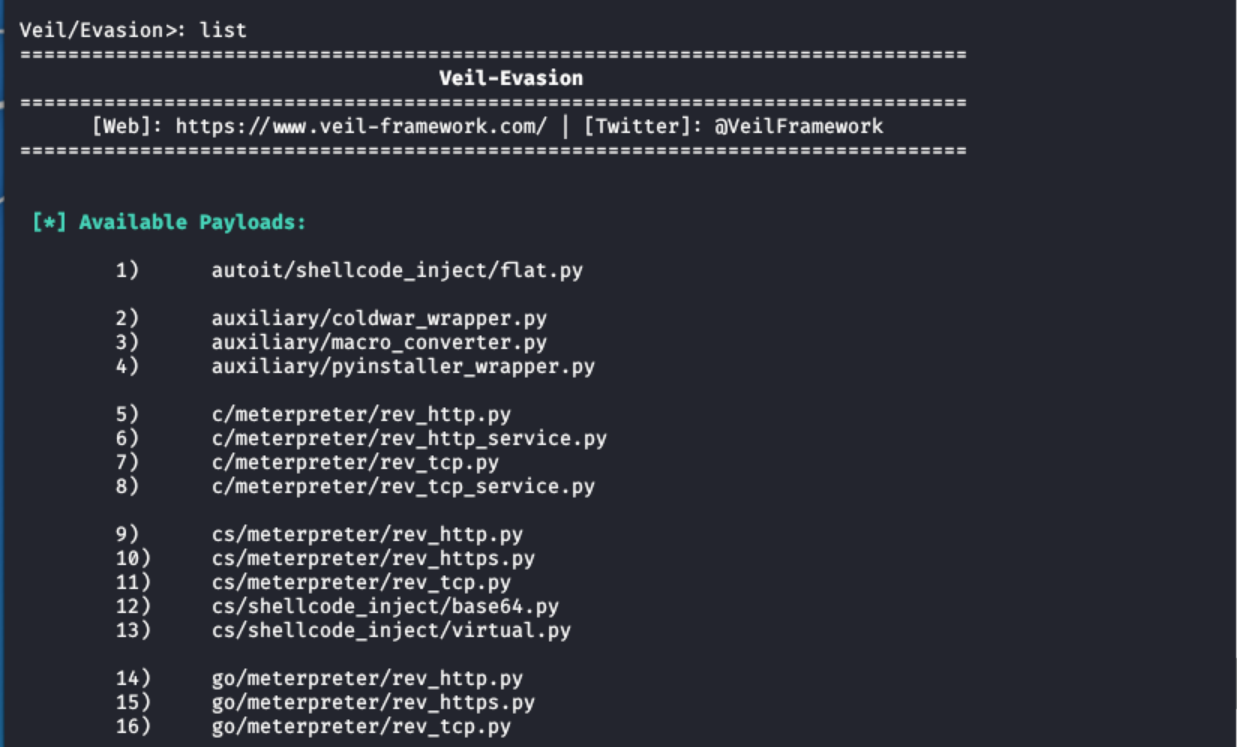


In veil there are two tools namely Evasion and Ordnance, we want to use the first tool i.e. the evasion



When we use the “evasion” tool we see that there are 41 payloads

The list of all the payloads



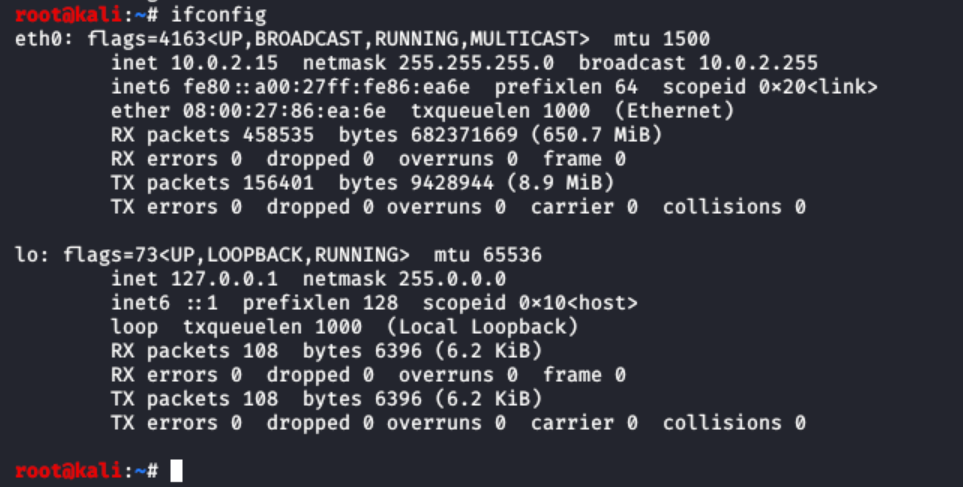
From the list we saw that some payloads are written in high level languages (e.g. python) and some are written in low level languages (e.g. go-lang).

The payloads written in high level languages are hard to detect for the antivirus whereas the payloads written in low level languages are easy to detect.

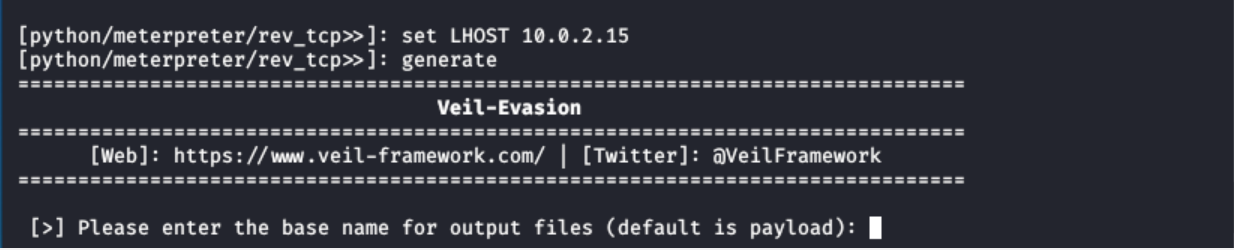
For our project we decided to use payloads written in Python, a high level language.

Use “python/meterpreter/rev\_https.py”

Now we need to set the “Lhost” field as our ip address:

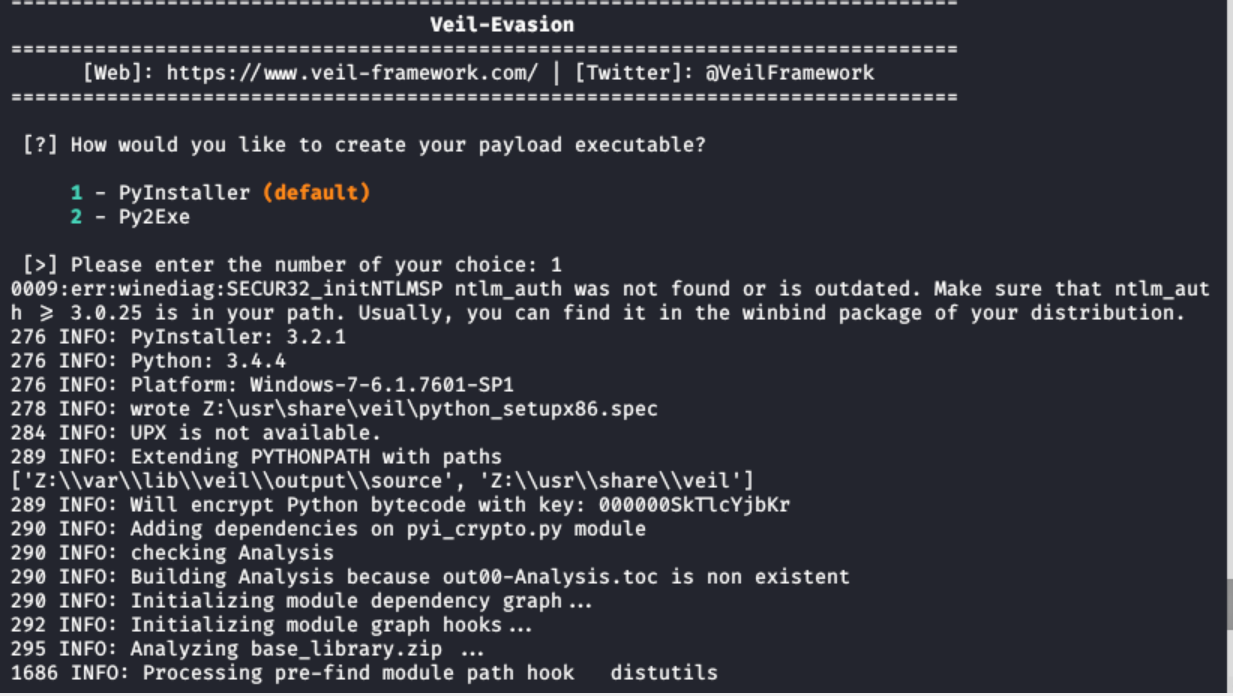


Setting Lhost and generating the payload:



We set the base name of the output file as “python\_setupx86.exe”

The base name of output file, Pyinstaller is used to create the payload executable file



The generated executable file:

