Cobalt Strike 4.2 – Everything but the kitchen sink

blog.cobaltstrike.com/2020/11/06/cobalt-strike-4-2-everything-but-the-kitchen-sink/

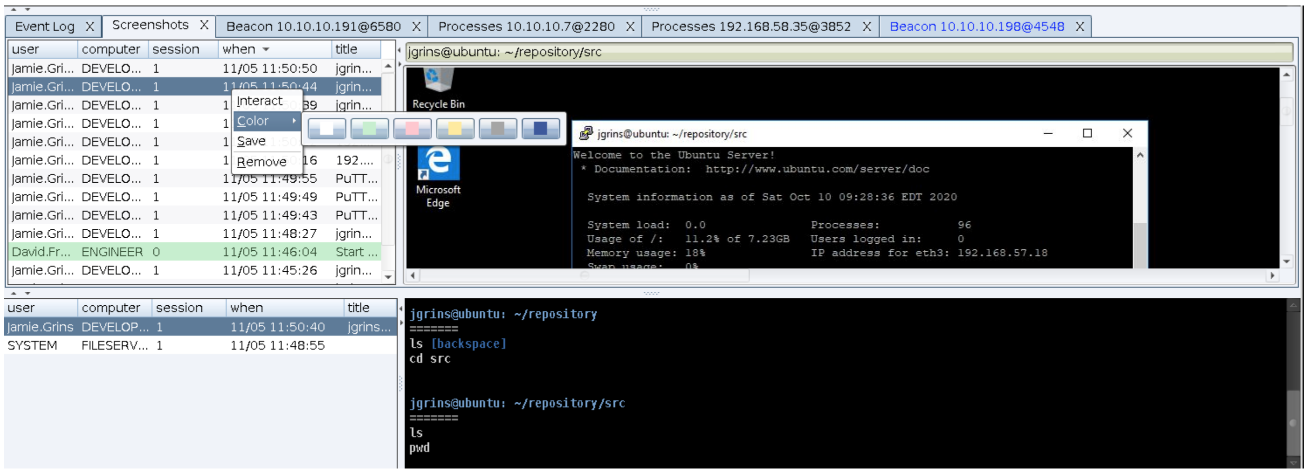
Raphael Mudge November 6, 2020

Cobalt Strike 4.2 is now available. This release overhauls our user exploitation features, adds more memory flexibility options to Beacon, adds more behavior flexibility to our post-exploitation features, and makes some nice changes to Malleable C2 too.

User Exploitation Redux

Cobalt Strike's screenshot tool and keystroke logger are examples of user exploitation tools. These capabilities are great for risk demonstration and story telling. But, with good UX, these features are also powerful capabilities to collect information that aids moving closer to an objective in a network.

Cobalt Strike's screenshot tool and keystroke logger now report active window, username, and desktop session with each of their results. This context helps our GUI, logs, and reports display where this information came from. It's a subtle change, but a big enhancement:



Event Log X
Screenshots X
Beacon 10.10.10.191@6580 X
Processes 10.10.10.7@2280 X
Processes 192.168.58.35@3852 X
Beacon 10.10.10.198@4548 X
jgrins@ubuntu: ~/repository/src
Recycle Bin
Microsoft Edge
Interact
Color
Save
Remove
jgrins@ubuntu: ~/repository/src

user	computer	session	when	title
Jamie.Gri...	DEVELO...	1	11/05 11:50:50	jgrin...
Jamie.Gri...	DEVELO...	1	11/05 11:50:44	jgrin...
Jamie.Gri...	DEVELO...	1	1[illegible]39	jgrin...
Jamie.Gri...	DEVELO...	1	1[illegible]	[illegible]
Jamie.Gri...	DEVELO...	1	1[illegible]	[illegible]
Jamie.Gri...	DEVELO...	1	1[illegible]16	192....
Jamie.Gri...	DEVELO...	1	11/05 11:49:55	PuTT...
Jamie.Gri...	DEVELO...	1	11/05 11:49:49	PuTT...
Jamie.Gri...	DEVELO...	1	11/05 11:49:43	PuTT...
Jamie.Gri...	DEVELO...	1	11/05 11:48:27	jgrin...
David.Fr...	ENGINEER	0	11/05 11:46:04	Start ...
Jamie.Gri...	DEVELO...	1	11/05 11:45:26	jgrin...

```
Welcome to the Ubuntu Server!
 * Documentation:  http://www.ubuntu.com/server/doc

  System information as of Sat Oct 10 09:28:36 EDT 2020

  System load:  0.0               Processes:           96
  Usage of /:   11.2% of 7.23GB   Users logged in:     0
  Memory usage: 18%               IP address for eth3: 192.168.57.18
  Swap usage:   0%
```

user	computer	session	when	title
Jamie.Grins	DEVELOP...	1	11/05 11:50:40	jgrins...
SYSTEM	FILESERV...	1	11/05 11:48:55	

```
jgrins@ubuntu: ~/repository
=======
ls [backspace]
cd src


jgrins@ubuntu: ~/repository/src
=======
ls
pwd
```

The right-click menu in the screenshot and keystroke has updates too. You can now remove a keystroke buffer or screenshot from the interface. Highlight a row with a color. And, save the keystroke buffer or screenshot to a local file as well.

We also split screenshot into two commands: screenshot and screenwatch. The **screenshot** command takes a single screenshot. The **screenwatch** command (which can fork&run or inject into a specific process) takes screenshots continuously until it's stopped with the **jobkill** command.

As believers in offense in depth, we've added new options to acquire screenshots and log keystrokes too. The **printscreen** command forces a PrintScr keypress and grabs the screenshot from the clipboard. This command was inspired by the creative and awesome Advanced Post-Exploitation Workshop given by @zerosum0x0 and @aleph__naught at 2017's DEF CON 25. And, we've added a **post-ex -> keylogger** Malleable C2 option to change the keystroke logger between GetAsyncKeyState and SetWindowsHookEx methods.

More In-memory Flexibility

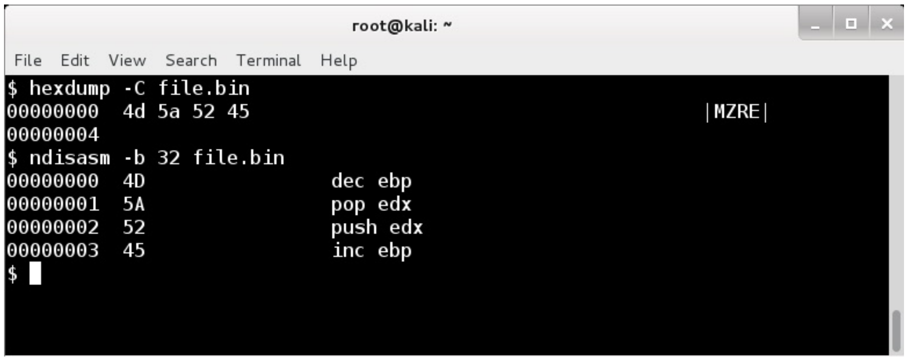
Cobalt Strike has long had an interest in in-memory detection and evasion. I love in-memory detections, because I think these tactics put real pressure on post-exploitation survival. But, it's also important to challenge security teams that rely on these tactics, to force thinking beyond that "one easy trick" that's working right now.

Cobalt Strike 4.2 continues to build Beacon's in-memory flexibility.

Beacon's Reflective Loader now has two added options for allocating memory for the Beacon payload in memory. Set **stage -> allocator** to *HeapAlloc* to use RWX heap memory for the Beacon payload. Set **stage -> allocator** to MapViewOfFile to stick Beacon in mapped memory. This is in addition to **VirtualAlloc** (the default) and 3.11's module stomping, which is our way of putting Beacon into image memory.

We've also added new options for content flexibility. Beacon's Reflective DLL follows Metasploit's conventions to make itself self-bootstrapping. We patch the beginning of the DLL (the part with that MZ header) with instructions to call the Reflective Loader with a few arguments. A common in-memory detection trick is to scan executable memory regions for MZ and PE content that looks like a PE/COFF file. These items are easy "it has to be there" content to find a Reflective DLL. Not anymore though.

Set **magic_mz_x86** to change the x86 Reflective DLL MZ header in Beacon. This updates the other parts of the loading process that depend on the value. The catch is that valid x86 instructions are required for magic_mz_x86. For example, MZ in x86 are the instructions dec ebp, pop edx. You don't want to begin execution with unexpected state, so it's also recommended to undo any state changes made by your instructions. This is why the default magic_mz_x86 value is MZRE. The R and E bytes decode to push edx, inc ebp. They undo the effect of the MZ instructions. The same idea applies for set **magic_mz_x64** too.



```
$ hexdump -C file.bin
00000000  4d 5a 52 45                                       |MZRE|
00000004
$ ndisasm -b 32 file.bin
00000000  4D                dec ebp
00000001  5A                pop edx
00000002  52                push edx
00000003  45                inc ebp
$ 
```

We've also added set **magic_pe** which changes the PE header magic bytes (and code that depends on these bytes) to something else. You can use whatever you want here, so long as it's two characters.

Post-ex Omikase Shimasu: Second serving

Cobalt Strike 3.14 introduced a lot of options to change behaviors, characteristics, and content in Cobalt Strike's post-exploitation DLLs. Cobalt Strike 4.2 continues this work.

We've added **post-ex -> pipename** to Malleable C2. This is an option to specify a comma-separated list of pipenames. Cobalt Strike will choose one of these when it executes its post-exploitation jobs. #s in the pipename are replaced with a [a-f0-9] character. We've also added set **ssh_pipename** to change the named pipe used by Cobalt Strike's SSH client. Multiple pipenames are allowed in this option too.

Some of Beacon's post-exploitation DLLs do create threads. In a keystroke logger that uses GetAsyncKeyState, this isn't easily avoidable. 🙂 To help these situations, we've introduced **post-ex -> thread_hint** to Malleable C2. When set, our post-ex DLLs will create a suspended thread with the specified module!function+0x[offset] address. Update the thread to run the post-exploitation capability. And, resume it. This is a way to push back on point-in-time analysis that looks for moduleless thread start addresses.

And, one of my favorite features in Cobalt Strike is the obfuscate and sleep feature. This is the ability for Beacon to mask its content [and code, if you're RWX] in memory during long blocking periods. Obfuscate and sleep is a method to push back on point-in-time analysis that looks for known strings in memory. Now, when **post-ex -> obfuscate** is true, Cobalt Strike's execute-assembly, keystroke logger, screenshot, and SSH client features will mask many of their strings while they're running.

Malleable C2 Things…

And, we've made several updates to the communicate side of Malleable C2. We've doubled the maximum size of the global **useragent** field. We've also doubled the max size of the the **http-get -> client** and **http-post -> client** programs too. This will help those of you that had run into the previous limits in your profile writing.

This release adds the global **headers_remove** option. This option affects http-get and http-post client blocks. It's a late-in-the-transaction option to remove unwanted HTTP client headers from the communication. WinINet, based on its configuration, will force some headers into the transaction. You may specify specify multiple unwanted headers (separate them with commas) in this option.

And, Cobalt Strike 4.2 introduces a data content jitter as well. Set **data_jitter** to a number value and Cobalt Strike will append a random content and length string (up to the data_jitter value bytes) to its http-get and http-post server output.

Check out the release notes to see a full list of what's new in Cobalt Strike 4.2. Licensed users may run the update program to get the latest. To procure Cobalt Strike (or ask about evaluation options), please contact us for more information.