# Volatility Plugin for Detecting Cobalt Strike Beacon - JPCERT/CC Eyes

By JPCERT/CC

Published: 2018-08-02 · Archived: 2026-04-05 15:02:22 UTC

- Python

JPCERT/CC has observed some Japanese organisations being affected by cyber attacks leveraging “Cobalt Strike” since around July 2017. It is a commercial product that simulates targeted attacks [1], often used for incident handling exercises, and likewise it is an easy-to-use tool for attackers. Reports from LAC [2] and FireEye [3] describe details on Cobalt Strike and actors who conduct attacks using this tool.

Cobalt Strike is delivered via a decoy MS Word document embedding a downloader. This will download a payload (Cobalt Strike Beacon), which will be executed within the memory. Since Cobalt Strike Beacon is not saved on the filesystem, whether a device is infected cannot be confirmed just by looking for the file itself. There is a need to look into memory dump or network device logs.

This article is to introduce a tool that we developed to detect Cobalt Strike Beacon from the memory. It is available on GitHub - Feel free to try from the following webpage:

> JPCERTCC/aa-tools · GitHub
>
> https://github.com/JPCERTCC/aa-tools/blob/master/cobaltstrikescan.py

**Tool details**

This tool works as a *plugin* for The Volatility Framework (hereafter “Volatility”), a memory forensic tool. Here are the functions of cobaltstrikescan.py:

- cobaltstrikescan: Detect Cobalt Strike Beacon from memory image
- cobaltstrikeconfig: Detect Cobalt Strike Beacon from memory image and extract configuration

To run the tool, save cobaltstrikescan.py in ”contrib/plugins/malware” folder in Volatility, and execute the following command:

```
$python vol.py [cobaltstrikescan|cobaltstrikeconfig] -f <memory.image> --profile=<profile>
```

Figure 1 shows an example output of cobaltstrikescan. You can see the detected process name (Name) and process ID (PID) indicating where the malware is injected to.

Figure 1: Execution results of cobaltstrikescan

```
[root@localhost vm]# python vol.py cobaltstrikescan -f mem.image --profile=Win7SP1x64
Volatility Foundation Volatility Framework 2.5
Name                 PID      Data VA
-------------------- -------- ------------------
powershell.exe          2508 0x0000000005380000
```

Figure 2 shows an example output of cobalrstrikeconfig. Please refer to Appendix A for configuration details for Cobalt Strike Beacon.

Figure 2: Execution results of cobaltstrikeconfig

```
[root@localhost vm]# python vol.py cobaltstrikeconfig -f mem.image --profile=Win7SP1x64
Volatility Foundation Volatility Framework 2.5
config addr: 0002F35C

----------------------------------------------------------------------
Process: powershell.exe (2508)

[CobaltStrike Config Info]
BeaconType              : 1 (Hybrid HTTP and DNS)
Port                    : 80
Polling(ms)             : 60000
Unknown1                : '\x00\x10\x00('
Jitter                  : 0
Maxdns                  : 255
Unknown2                : '0\x81\x9f0\r\x06\t*\x86H\x86\xf7\r\x01\x01\x01\x05\x00\x03\x81\x8d\x000\x81\x89\x02\x81\x81\x00\xb3\
\x8cK\xdeH.\xc4W$HnCn\xb2\xf0\xf2\x92\x0f\xfe\x9f\x05\x85\x14\x1e\xe7x\x15\x9d\x96\xe7v[i\xb8_d\xa8*\x8el!\xed\x8c\xe8\xe55\xe
5h\xae\x85\x02\x03\x01\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
C2Server                : nl01.misaupdate.com,/__utm.gif,nl02.misaupdate.com,/__utm.gif,nl03.misaupdate.com,/__utm.gif
UserAgent               : Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; MATBJS)
HTTP_Method2_Path       : /___utm.gif
Unknown3                : '\x00\x00\x00\x04\x00\x00\x00\x02\x00\x00\x00\x0f\x00\x00\x00\x02\x00\x00\x00\x0f\x00\x00\x00\x02\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
Header1                 : utmac=UA-2202604-2
                        : utmcn=1
                        : utmcs=ISO-8859-1
                        : utmsr=1280x1024
                        : utmsc=32-bit
                        : utmul=en-US
                        : __utma
                        : utmcc
Header2                 : Content-Type: application/octet-stream
                        : UA-220
                        : -2
                        : utmac
                        : utmcn=1
                        : utmcs=ISO-8859-1
                        : utmsr=1280x1024
                        : utmsc=32-bit
                        : utmul=en-US
Injection_Process       :
PipeName                : \\%s\pipe\msagent_%x
Year                    : 0
Month                   : 0
Day                     : 0
DNS_idle                : 0.0.0.0
DNS_sleep(ms)           : 0
Method1                 : GET
Method2                 : POST
Unknown4                : '\x00\x00\x00\x00'
Spawnto_x86             : %windir%\syswow64\rundll32.exe
Spawnto_x64             : %windir%\sysnative\rundll32.exe
Unknown5                : '\x00\x01'
Proxy_HostName          :
Proxy_UserName          :
Proxy_Password          :
Proxy_AccessType        : 2 (use IE settings)
create_remote_thread    : Enable
```

**In closing**

Actors using Cobalt Strike continue attacks against Japanese organisations. We hope this tool helps detecting the attack in an early stage.

- Takuya Endo

*(Translated by Yukako Uchida)*

**Reference**

[1] Strategic Cyber LLC:COBALT STRIKE ADVANCED THREAT TACTICS FOR PANETRATION TESTERS

https://www.cobaltstrike.com/

[2] LAC: New attacks by APT actors menuPass (APT10) observed (Japanese)

https://www.lac.co.jp/lacwatch/people/20180521_001638.html

[3] FireEye: Privileges and Credentials: Phished at the Request of Counsel

https://www.fireeye.com/blog/threat-research/2017/06/phished-at-the-request-of-counsel.html

[4] Cybereason: Operation Cobalt Kitty: A large-scale APT in Asia carried out by the OceanLotus Group

https://www.cybereason.com/blog/operation-cobalt-kitty-apt

**Appendix A**

Table A: Configuration format

| Offset | Length | Description |
|---|---|---|
| 0x00 | 2 | index (Refer to Table B) |
| 0x02 | 2 | Data length<br>1 = 2 byte, 2 = 4 byte, 3 = as specified in 0x04 |
| 0x04 | 2 | Data length |
| 0x06 | As specified in 0x04 | Data |

Table B: Configuration

| Offset | Description | Remarks |
|---|---|---|
| 0x01 | BeaconType | 0=HTTP, 1=Hybrid HTTP and DNS, 8=HTTPS |
| 0x02 | Port number | |
| 0x03 | Polling time | |

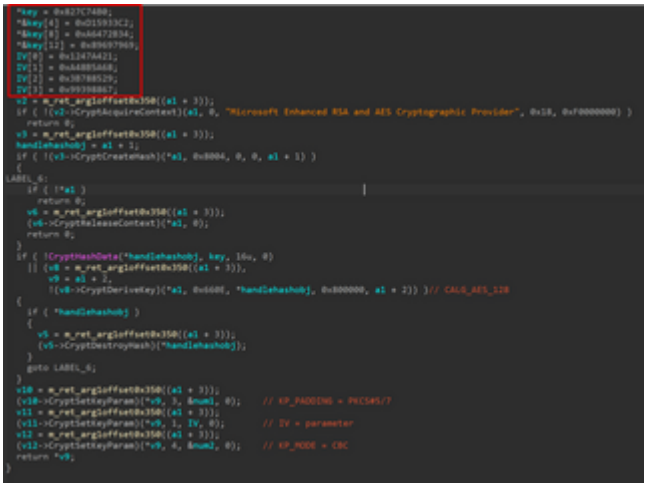
| Offset | Description | Remarks |
|---|---|---|
| 0x04 | Unknown | |
| 0x05 | Jitter | Ratio of jitter in polling time (0-99%) |
| 0x06 | Maxdns | Maximum length of host name when using DNS (0-255) |
| 0x07 | Unknown | |
| 0x08 | Destination host | |
| 0x09 | User agent | |
| 0x0a | Path when communicating HTTP_Header2 | |
| 0x0b | Unknown | |
| 0x0c | HTTP_Header1 | |
| 0x0d | HTTP_Header2 | |
| 0x0e | Injection process | |
| 0x0f | Pipe name | |
| 0x10 | Year | Stops operating after the specified date by Year, Month, Day |
| 0x11 | Month | |
| 0x12 | Day | |
| 0x13 | DNS_idle | |
| 0x14 | DNS_Sleep | |
| 0x1a | HTTP_Method1 | |
| 0x1b | HTTP_Method2 | |
| 0x1c | Unknown | |
| 0x1d | Process to inject arbitrary shellcode (32bit) | |
| 0x1e | Process to inject arbitrary shellcode (64bit) | |

| Offset | Description | Remarks |
|---|---|---|
| 0x1f | Unknown | |
| 0x20 | Proxy server name | |
| 0x21 | Proxy user name | |
| 0x22 | Proxy password | |
| 0x23 | AccessType | 1 = Do not use proxy server<br>2 = Use IE configuration in the registry<br>4 = Connect via proxy server |
| 0x24 | create_remote_thread | Flag whether to allow creating threads in other processes |
| 0x25 | Not in use | |

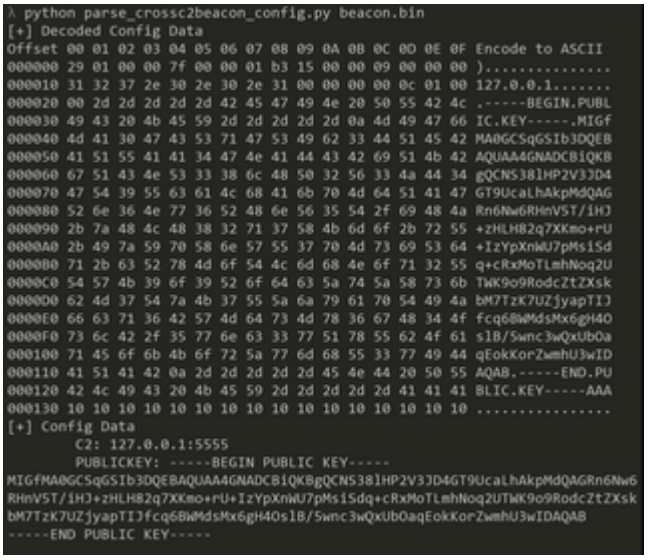
JPCERT/CC

Please use the below contact form for any inquiries about the article.

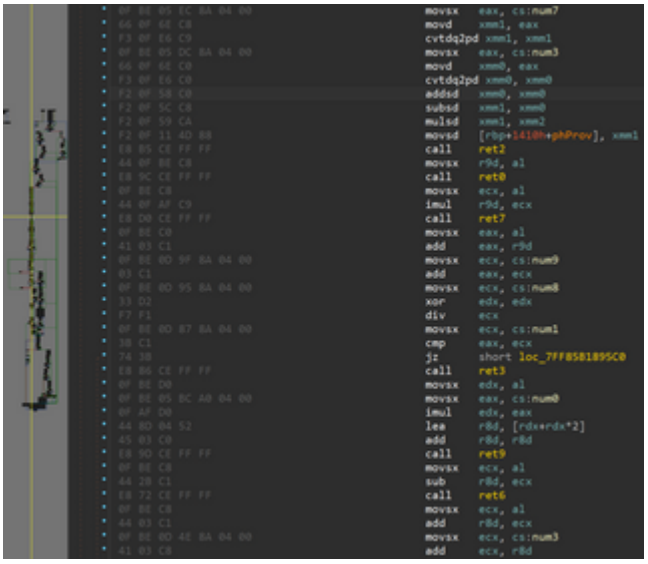
## Related articles

- 

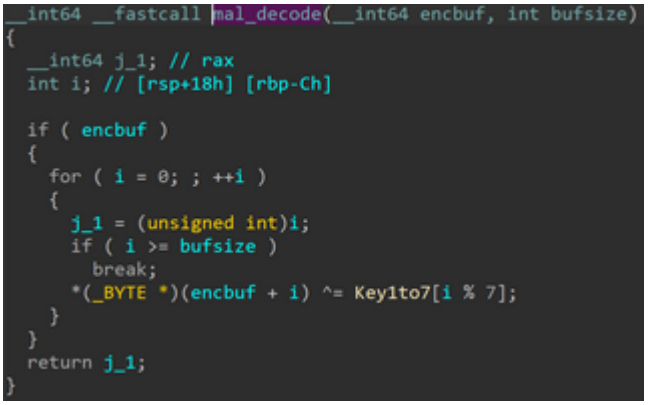
  Update on Attacks by Threat Group APT-C-60



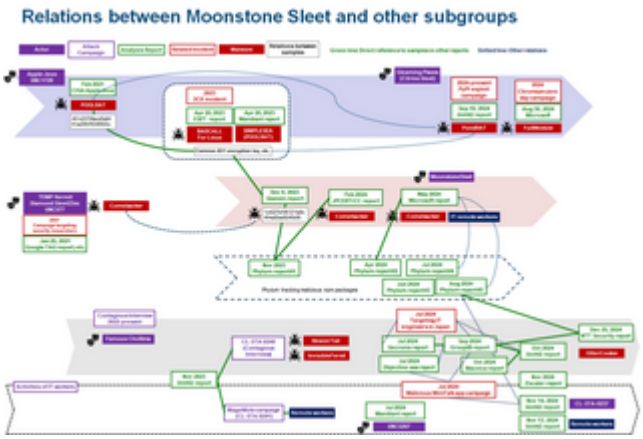

- [CrossC2 Expanding Cobalt Strike Beacon to Cross-Platform Attacks]



- [Malware Identified in Attacks Exploiting Ivanti Connect Secure Vulnerabilities]




- [DslogdRAT Malware Installed in Ivanti Connect Secure]

- 


  [Tempted to Classifying APT Actors: Practical Challenges of Attribution in the Case of Lazarus's Subgroup]

---

Source: https://blogs.jpcert.or.jp/en/2018/08/volatility-plugin-for-detecting-cobalt-strike-beacon.html