Malware Used by BlackTech after Network Intrusion - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2019-09-17 · Archived: 2026-04-05 17:14:01 UTC

- Tool
- BlackTech

Previously, we explained about malware "TSCookie" and "PLEAD" which are used by an attack group BlackTech. Their activities have been continuously observed in Japan as of now. We have been seeing that a new malware variant is being used after they successfully intruded into a target network. This article explains the details of the variant.

TSCookie used after intrusion

The malware consists of 2 files (TSCookie Loader and TSCookie) as in Figure 1.

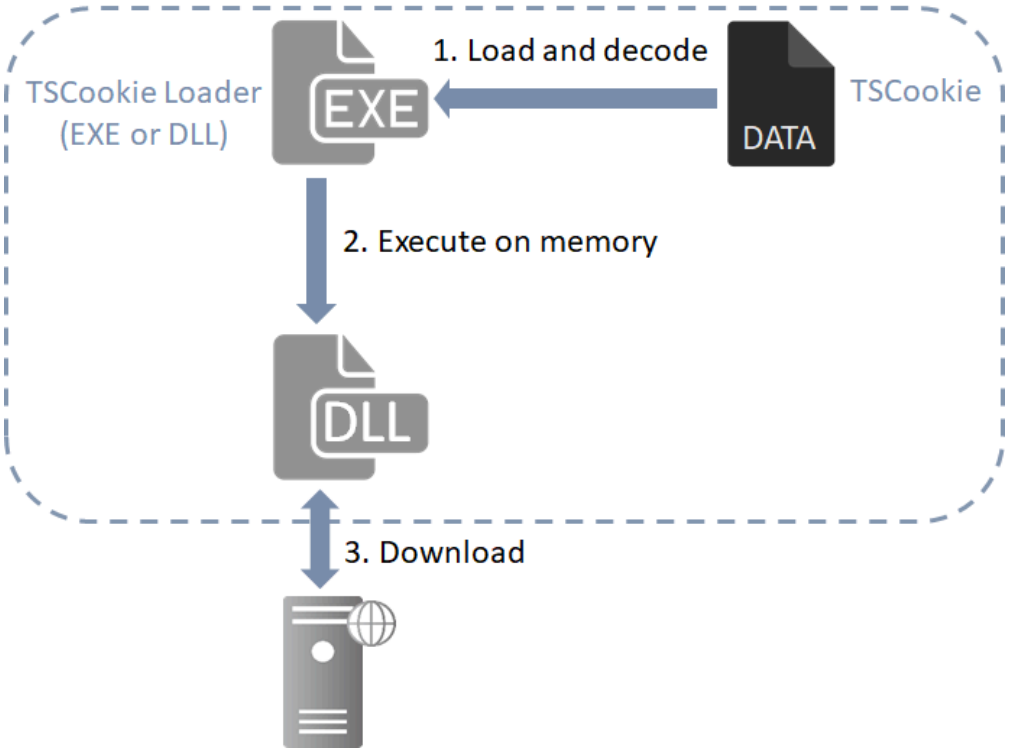


Figure 1: Overview of TSCookie Loader and TSCookie

TSCookie Loader is either in EXE or DLL format, and it reads and executes specific files stored in the same folder or the following locations. (The folders may vary depending on the sample.)

- C:\Windows
- C:\ProgramData\Microsoft

- C:\Users\Public\Documents
- C:\Program Files\Internet Explorer

It reads files that match the following file names:

- desktop.db
- Files with name that match *7????*. (wildcard)
- Files with name that match *8????*. (wildcard)

For example, files names such as KB78E7269.log and PM89E7267.xml have been confirmed.
TSCookie is RC4-encrypted and can be decoded by TSCookie Loader before being executed on the memory. TSCookie itself is a downloader and operates according to modules downloaded from an external server. Some characteristics such as configuration and communication protocols differ between TSCookie and the variant. Details of TSCookie behaviour is described in the following section.

TSCookie behaviour

TSCookie supports multiple communication protocols (HTTP, HTTPS and custom protocol). The protocol that each sample uses is described in its configuration. (Please see Appendix A for the details of the configuration.)
If it is configured to use HTTP protocol, the following HTTP POST request is sent:

```
POST /index?o=E7E168C4EC82E HTTP/1.1
Cache-Control: no-cache
Pragma: no-cache
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Proxy-Connection: Keep-Alive
Content-Length: [size]
Host: [host name]

[Data]
```

There are several patterns of URL path, which are dynamically created with the following random strings and values. (Some of them are described with format specifiers. Other patterns also exist.)

- /news?%c=%X%X
- /index?%c=%X%X
- /?id=%X%X
- /Default.aspx?%c=%X%X
- /m%u.jsp?m=%d
- /N%u.jsp?m=%d

Sent data is RC4-encrypted. Please see Table B-1 and B-2 in Appendix B for the format of the data.
Data downloaded by the HTTP POST request is RC4-encrypted by the 8-byte value consisting of the RC4 key

(Table A-1 in Appendix A) and another value in the received data (RC4 key as in Table B-3 in Appendix B). The downloaded data contains modules, and they are executed on the memory.

TSCookie decoding tool

We have developed a tool to decode TSCookie files and extract configuration. This is available on GitHub for your use.

JPCERTCC/aa-tools - GitHub
https://github.com/JPCERTCC/aa-tools/blob/master/tscookie_data_decode.py

In closing

We have received many reports about TSCookie infection. Please make sure that there is no infection in your organisation, referring to the file names and communication protocols described in this article. The hash value of the samples described in this article are listed in Appendix C.

Shusei Tomonaga
(Translated by Yukako Uchida)

Appendix A: TSCookie Configuration

Table A: Configuration

Offset	Contents	Remarks
0x000	Destination server and port number	Multiple hosts can be specified by listing with a semicolon ";"
0x400	RC4 key	Used for encryption
0x404	Sleep times	
0x42C	Mutex	
0x44C	Communication mode	- 1,2,3: HTTP protocol supporting authentication proxy - 6,7,8: HTTPS protocol - 0: Custom protocol - 5: HTTP protocol
0x454	HTTP connection keep	
0x458	ICMP recipient setting	Receive information of the destination server by ICMP
0x4D4	IP address to receive ICMP communication	
0x624	Process injection mode	- 0: Launch - 1: Already running

		- 2: Launch offset 0x62C
0x628	Process to be injected	- 0: svchost.exe - 1: iexplorer.exe - 2: explorer.exe - 3: Default Browser - 4: Process in offset 0x62C
0x62C	Process name	
0x72C	Proxy server	
0x76C	Proxy port number	
0x770	Proxy username	
0x790	Proxy password	
0x7B0	Proxy mode	- 1: Use configuration data - 0: Detect Proxy automatically
0x7B4	Proxy authentication process	AuthScheme

- Some samples may not inject processes.

Appendix B: Data exchanged by TSCookie

Table B-1: Format of sent data

Offset	Length	Contents
0x00	4	Number of received data (begins with 0xFFFFFFFF)
0x04	4	Length of data sent
0x08	4	Times of communication
0x0C	4	Fixed value (Set to 0x5322 at the beginning, then to 0x5324 or 0x5325 while receiving modules)
0x1C	4	Random data (RC4 key)
0x20	-	Random data after first communication (See Table B-2 for first communication)

- Up to offset 0x1C, the contents are RC4-encrypted with the key in the configuration and random data.

Table B-2: Data format of first communication after offset 0x20

Offset	Length	Contents

0x00	4	0x9A65001E
0x04	4	Process ID
0x08	4	0x5322
0x0C	4	Random data
0x10	4	Data size from offset 0x14
0x14	-	Random data (RC4 key)

- Up to offset 0x14, the contents are RC4-encrypted with the key in the configuration and random data.

Table B-3: Format of received data

Offset	Length	Contents
0x00	4	Number of received data
0x04	4	Length of received data
0x0C	4	-
0x10	4	Whether the contents from offset 0x20 is encrypted
0x1C	4	RC4 key
0x20	-	Module data

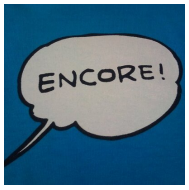
- Up to 0x1C, the contents are RC4-encrypted with the key contained in the configuration and another key in the received data.

Appendix C: SHA-256 value of the samples

TSCookie Loader

- 072f24d2691fb3930628be91bc46cefb8bc3364d1d09d72ab0cb3863681cb107
- f49956f498042feb237c3e898f74a8e14500c27cda2746efca2d973a5390baa8
- 3e12938df72380e4ae7a2dcb3322e563de3da102f5f32b26a29662ba594e73d1
- 23ca1a3ca26ada00502bbd1abf4d42302343dafba32cbc0711847d52884ff8e1
- 6ec56de53ef1ea66c81b3e48f9a9b3cf3dc8e3ebda1ec08bf95cc21228a4c7b3
- bd89b972de19c8ab2be0fb3e2aa44638a95e465e4b52920c94e6f59c25ce4693
- c5d7e5a12c8eab9c14f008c93d92e0070f84f358d39f28ac089ee917c652f5a8
- 85536a139b9d44157aea2908a6a6e53e4ac19077355680b69edd8e84c70254bc
- 0d00d12d71dd080d2861e9da89906a67bb822c64366b4c6b72a55bb8c26a4ea3
- 81dfce847a9fd6a3a0080a927bbb740709bdcc099bfe1b0cfc99958f6ddeb52f
- 48fdc29e7f47e5d38c88a89667ed85740628bf4f4ce95045019f7ebfeb4bbb5c

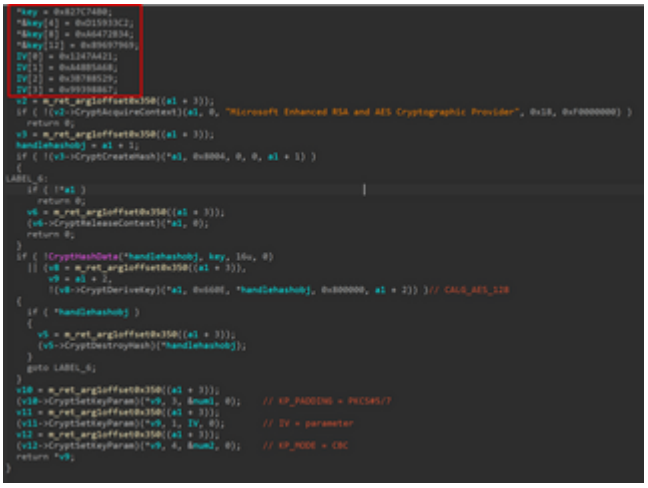
- d5909d06ddb394dea114052e9e174fa1e88324d805d153edb6076c53842fd2f2
- 9e10a1abbff4d421eaee20040fb2a9270c4efb6d75ee6cd728b09bac1042bfa6
- ae5528cc802c81946f2787c7e884656416acebc89466989eeca9379fa066ad96
- 69b07aae04af6ca57d6066fdcbfeeb4c4849bfd2cd65b01c1e576f45b1c24d79
- 784b331d30d46ee9e7a264ecb45e3a39d7cef135d189bf0e712e89935728c13f
- 0eb9947a1ef4b810517f6cba175a321c4d69c3058d688bdd73492d54e7932c86



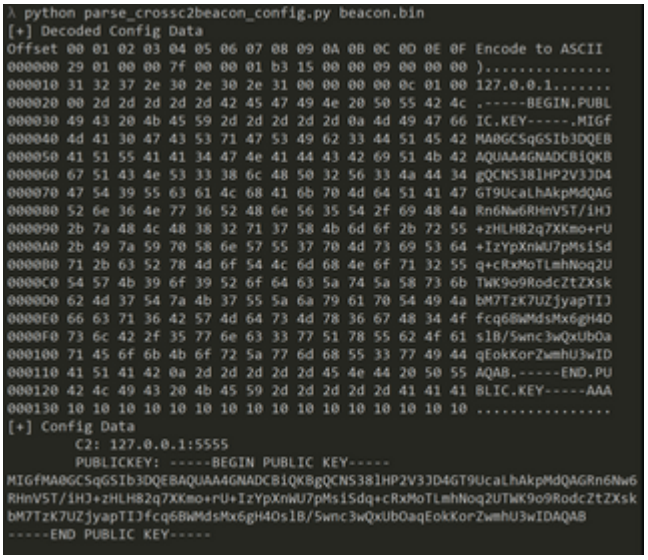
朝長 秀誠 (Shusei Tomonaga)

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.

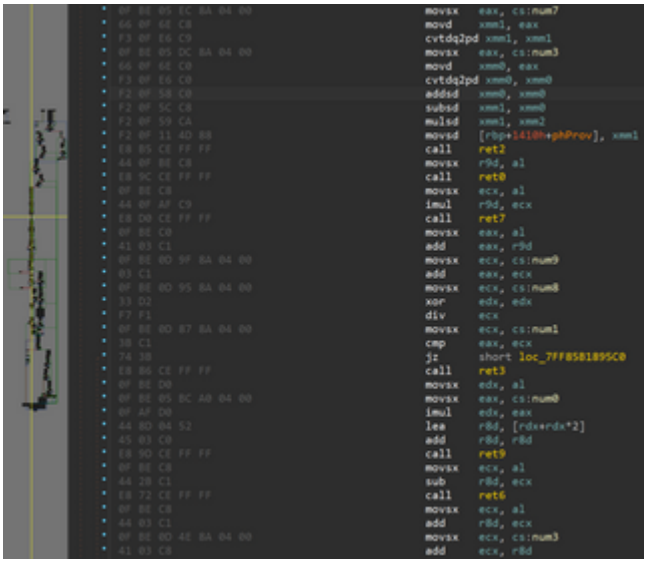
Related articles



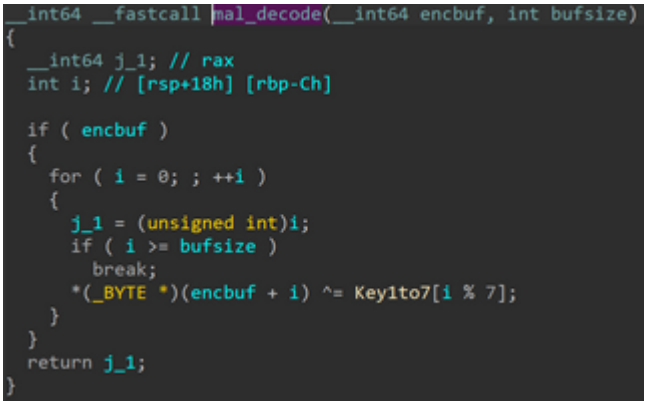
- Update on Attacks by Threat Group APT-C-60



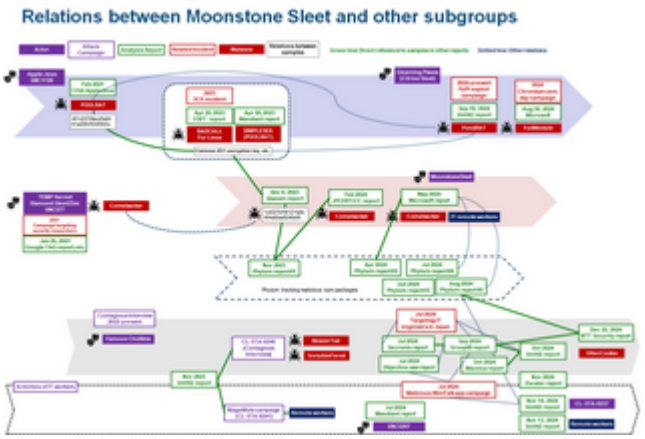
- [CrossC2 Expanding Cobalt Strike Beacon to Cross-Platform Attacks]



- [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/2019/09/tscookie-loader.html