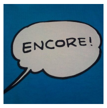
Bug in Malware “TSCookie” - Fails to Read Configuration -

blogs.jpcert.or.jp/en/2018/11/tscookie2.html



朝長 秀誠 (Shusei Tomonaga)

November 12, 2018

BlackTech

-
- Email

In a previous article we have introduced malware ‘TSCookie’, which is assumedly used by an attacker group BlackTech. We have been observing continuous attack activities using the malware until now. In the investigation of an attack observed around August 2018, we have confirmed that there was an update in the malware. There are two points meriting attention in this update:

- Communication with C&C server
- Decoding configuration information

This article will introduce the details of the update.

Communication with C&C server

In the previous version, TSCookie included encrypted contents in the Cookie header to communicate to a C&C server.

```
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:
1405D7CD01C6978E54E86DA9525E1395C4DD2F276DD28EABCC3F6201ADAA66F55C15352D29D0FFE51BC9D4

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Host:[host name]:443
```

In the new version, Cookie header is no longer used. Instead, encrypted contents are placed within the URL parameter as below:

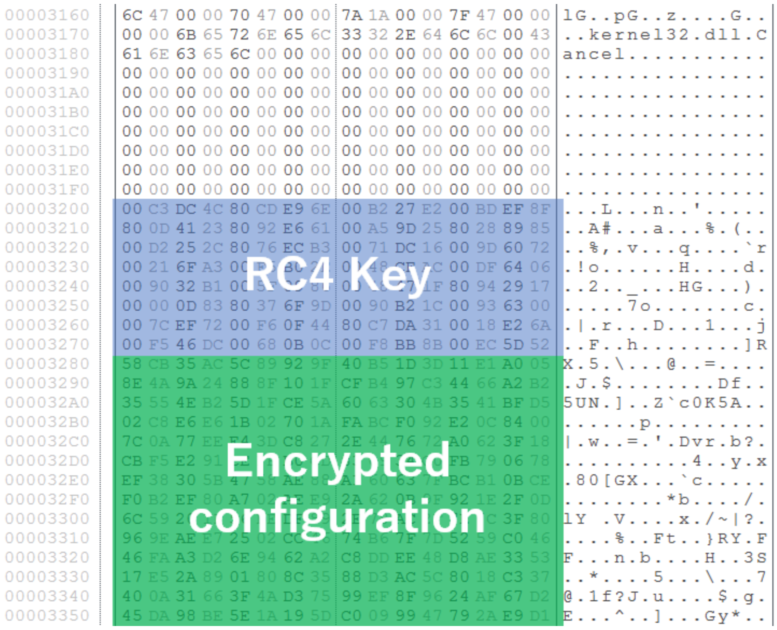
```
GET /t3328483620.aspx?m=4132641264&i=44D6CF457ADC27B2AFAAEAA&p=EF4D5069C30D6CAC9
HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32)
Host: [host name]:443
```

If received an ack from the server to this HTTP GET request, an HTTP POST request will be sent as a next step. The communication feature is the same as the previous TSCookie.
For encryption, RC4 is still used, but the key is generated differently. Here is an example code for decoding HTTP GET request parameter.

```
data = "&" + sys.argv[1] # sys.argv[1] = URL path
conf_key = sys.argv[2].decode("hex") # sys.argv[2] = Configuration key
field = data.split("&")

url_key = field[1]
i=2
encdata = ""
while i<len(field):
    value = field[i].split("=")
    encdata += value[1]
    i+=1

key1 = 0
for i in range(len(url_key)):
    key1 = ord(url_key[i]) + ROR(key1, 13)
    key1 = key1 & 0xFFFFFFFF

key2 = 0
for i in range(len(conf_key)):
    key2 = ord(conf_key[i]) + ROR(key2, 13)
    key2 = key2 & 0xFFFFFFFF

key = pack("I", key1) + pack("I", key2)

decode_data = rc4(encdata.decode('hex'), key)
```

Decoding configuration information

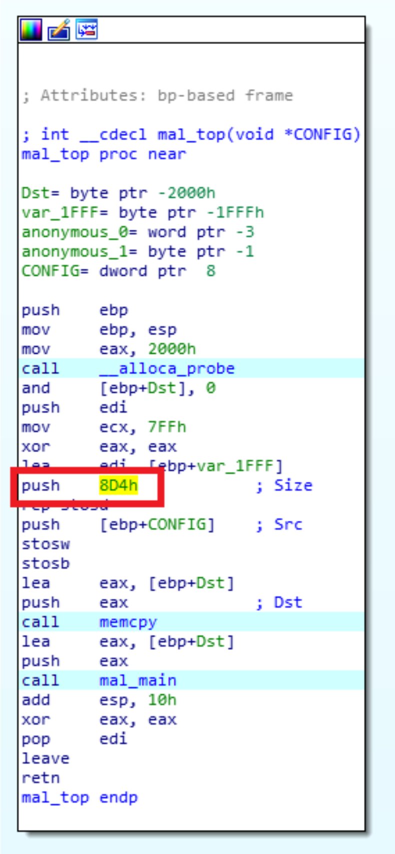
TSCookie possesses its own configuration information and operates accordingly. The details of the configuration remain the same in the new version. The difference is the decoding method of the configuration. Previously, TSCookie had its 4-byte RC4 key in the beginning of the configuration, which was used for decoding. In the new version, the size is expanded to 0x80 bytes (Figure 1).



```
00003160  6C 47 00 00 70 47 00 00 7A 1A 00 00 7F 47 00 00  lG..pG..z....G..
00003170  00 00 6B 65 72 6E 65 6C 33 32 2E 64 6C 6C 00 43  ..kernel32.dll.C
00003180  61 6E 63 65 6C 00 00 00 00 00 00 00 00 00 00 00  ancel...........
00003190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000031F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00003200  00 C3 DC 4C 80 CD E9 6E 00 B2 27 E2 00 BD EF 8F  ...L...n..'.....
00003210  80 0D 41 23 80 92 E6 61 00 A5 9D 25 80 28 89 85  ..A#...a...%.(..
00003220  00 D2 25 2C 80 76 EC B3 00 71 DC 16 00 9D 60 72  ..%,.v...q....`r
00003230  00 21 6F A3 00 [illegible] [illegible] [illegible] [illegible] 48 [illegible] [illegible] 00 DF 64 06  .!o......H....d.
00003240  00 90 32 B1 00 [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] 1F 80 94 29 17  ..2.._...HG...).
00003250  00 00 0D 83 80 37 6F 9D 00 90 B2 1C 00 93 63 00  .....7o.......c.
00003260  00 7C EF 72 00 F6 0F 44 80 C7 DA 31 00 18 E2 6A  .|.r...D...1...j
00003270  00 F5 46 DC 00 68 0B 0C 00 F8 BB 8B 00 EC 5D 52  ..F..h........]R
00003280  58 CB 35 AC 5C 89 92 9F 40 B5 1D 3D 11 E1 A0 05  X.5.\...@..=....
00003290  8E 4A 9A 24 88 8F 10 1F CF B4 97 C3 44 66 A2 B2  .J.$........Df..
000032A0  35 55 4E B2 5D 1F CE 5A 60 63 30 4B 35 41 BF D5  5UN.]..Z`c0K5A..
000032B0  02 C8 E6 E6 1B 02 70 1A FA BC F0 92 E2 0C 84 00  ......p.........
000032C0  7C 0A 77 EE F4 3D C8 27 2E 44 76 72 A0 62 3F 18  |.w..=.'.Dvr.b?.
000032D0  CB F5 E2 91 [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] FB 79 06 78  ...........4..y.x
000032E0  EF 38 30 5B 47 58 AE 88 [illegible] 60 63 7F BC B1 0B CE  .80[GX...`c.....
000032F0  F0 B2 EF 80 A7 02 [illegible] E9 2A 62 0B [illegible] 92 1E 2F 0D  ........*b..../.
00003300  6C 59 [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] 3F 80  lY .V....x./~|?.
00003310  96 9E AE E7 25 02 CC [illegible] 74 B6 7F 7D 52 59 C0 46  ....%..Ft..}RY.F
00003320  46 FA A3 D2 6E 94 62 A2 C8 DD EE 48 D8 AE 33 53  F...n.b....H..3S
00003330  17 E5 2A 89 01 80 8C 35 88 D3 AC 5C 80 18 C3 37  ..*....5...\...7
00003340  40 0A 31 66 3F 4A D3 75 99 EF 8F 96 24 AF 67 D2  @.1f?J.u....$.g.
00003350  45 DA 98 BE 5E 1A 19 5D C0 09 99 47 79 2A E9 D1  E...^..]...Gy*..
```

Figure 1: RC4 key and encrypted configuration

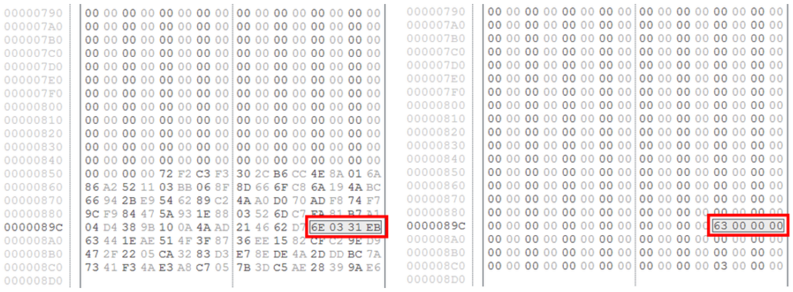
We have confirmed that this update made TSCookie fail to read part of the configuration. Figure 2 shows the code copying encrypted configuration (0x8D0 bytes) and RC4 key (0x80 bytes).



```
; Attributes: bp-based frame

; int __cdecl mal_top(void *CONFIG)
mal_top proc near

Dst= byte ptr -2000h
var_1FFF= byte ptr -1FFFh
anonymous_0= word ptr -3
anonymous_1= byte ptr -1
CONFIG= dword ptr  8

push    ebp
mov     ebp, esp
mov     eax, 2000h
call    __alloca_probe
and     [ebp+Dst], 0
push    edi
mov     ecx, 7FFh
xor     eax, eax
lea     edi, [ebp+var_1FFF]
push    8D4h            ; Size
rep stosd
push    [ebp+CONFIG]    ; Src
stosw
stosb
lea     eax, [ebp+Dst]
push    eax             ; Dst
call    memcpy
lea     eax, [ebp+Dst]
push    eax
call    mal_main
add     esp, 10h
xor     eax, eax
pop     edi
leave
retn
mal_top endp
```

Figure 2: Code copying RC4 Key and encrypted configuration

The code copies data sized 0x8D4 (0x8D0 + 4 bytes), which ignores the updated RC4 key size. To copy the updated RC4 key and configuration correctly, it needs to be set to 0x950 (0x8D0 + 0x80 bytes). With this fault, configuration cannot be decoded properly. Figure 3 describes how TSCookie configuration is decoded.



```
00000790  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000800  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000820  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000830  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000840  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000850  00 00 00 00 72 F2 C3 F3  30 2C B6 CC 4E 8A 01 6A
00000860  86 A2 52 11 03 BB 06 8F  8D 66 6F C8 6A 19 4A BC
00000870  66 94 2B E9 54 62 89 C2  4A A0 D0 70 AD F8 74 F7
00000880  9C F9 84 47 5A 93 1E 88  03 52 6D C7 FA 81 B7 A1
0000089C  04 D4 38 9B 10 0A 4A AD  21 46 62 D7 6E 03 31 EB
000008A0  63 44 1E AE 51 4F 3F 87  36 EE 15 82 CF C2 9E D9
000008B0  47 2F 22 05 CA 32 83 D3  E7 8E DE 4A 2D DD BC 7A
000008C0  73 41 F3 4A E3 A8 C7 05  7B 3D C5 AE 28 39 9A E6
000008D0
```

```
00000790  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000007F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000800  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000810  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000820  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000830  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000840  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000860  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000870  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000880  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0000089C  00 00 00 00 00 00 00 00  00 00 00 00 63 00 00 00
000008A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000008B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000008C0  00 00 00 00 00 00 00 00  00 00 00 00 03 00 00 00
000008D0
```

Figure 3: Decoded TSCookie configuration
(Left: Copy size 0x8D4, Right: Copy size 0x950)
Decoded results differ in the left figure (with the wrong, smaller copy size) and right figure (with correct, expanded copy size). Data at 0x89C byte (4 bytes) specifies the waiting time (seconds) before reconnecting to a C&C server. The attackers initially set this to 99 (0x63) seconds (as in the right figure), however, it will not be reconnected for few days since it is not read properly (left figure).

In closing

It is often the case that attackers give an update to their malware based on analysis reports provided from security vendors. We assume that this bug will be fixed sooner or later. We will update when we confirm new malware features.

The malware sample's hash value is available in Appendix A, and we also list some C&C servers in Appendix B. We hope this is helpful in identifying signs of infection.

(Translated by Yukako Uchida)

Appendix A SHA-256 Hash Value of a sample

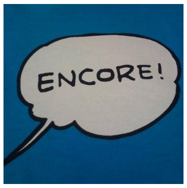
a5c75f4d882336c670f48f15bf3b3cc3dfe73dba7df36510db0a7c1826d29161

Appendix B C&C server

- mediaplayer.dnset.com
- mediaplayers.ssl443.org
- fashion.androiddatacenter.com
- sakurings.flnet.org

-
- Email

Author



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.

Was this page helpful?

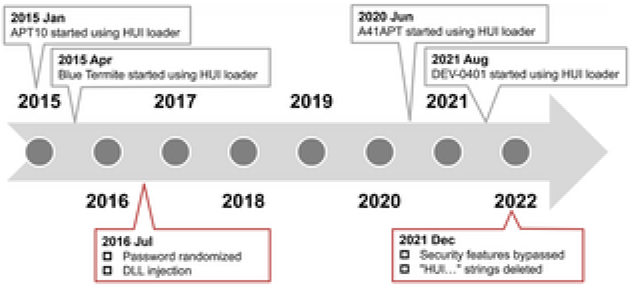
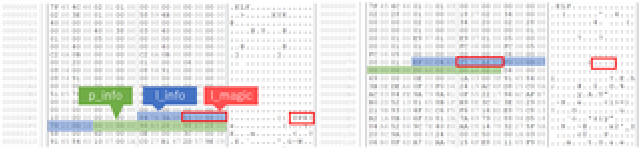
0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

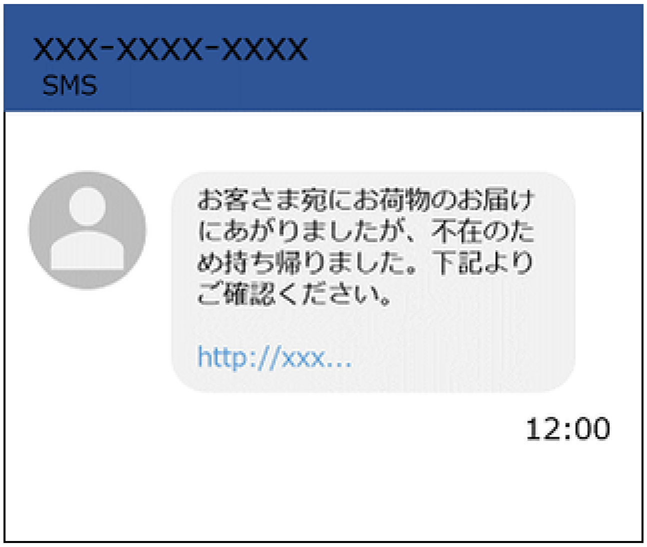
This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

please change the setting of your browser to set JavaScript valid. Thank you!

Related articles

- 
2015 Jan
APT10 started using HUI loader
2015 Apr
Blue Termite started using HUI loader
2020 Jun
A41APT started using HUI loader
2021 Aug
DEV-0401 started using HUI loader
2015
2017
2019
2021
2016
2018
2020
2022
2016 Jul
Password randomized
DLL injection
2021 Dec
Security features bypassed
"HUI..." strings deleted
Analysis of HUI Loader
- 

Anti-UPX Unpacking Technique



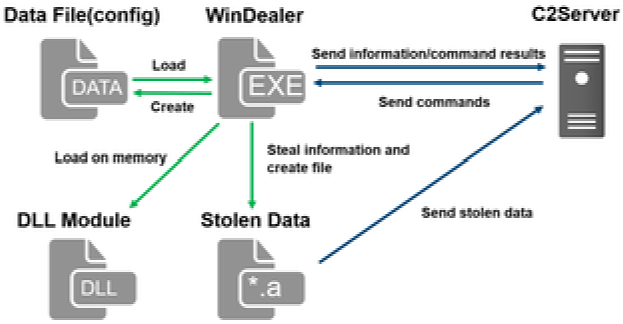
XXX-XXXX-XXXX
SMS

お客さま宛にお荷物のお届けにあがりましたが、不在のため持ち帰りました。下記よりご確認ください。

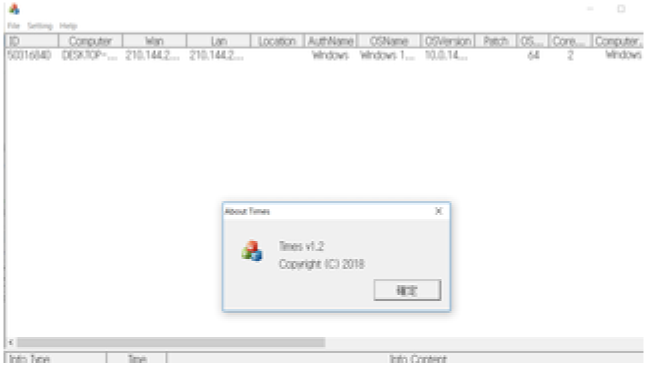
http://xxx...

12:00

- FAQ: Malware that Targets Mobile Devices and How to Protect Them



- Malware WinDealer used by LuoYu Attack Group



ID	Computer	Wan	Lan	Location	AuthName	OSName	OSVersion	Patch	OS...	Core...	Computer...
50316840	DESKTOP-...	210.144.2...	210.144.2...		Windows	Windows 1...	10.0.14...		64	2	Windows

About Times

Times v1.2
Copyright (C) 2018

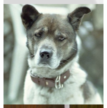
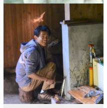
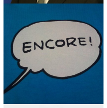
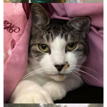
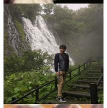
Info Type	Time	Info Content

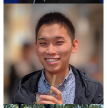
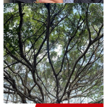
- Malware Gh0stTimes Used by BlackTech

Back
Top
Next

Authors

-

-
- 
- 
-
-
-
-
-
-
- 
ENCORE!
-
-
-
-
-
- 
- 
-
APCERT 2016
PROGRAM

-
-
- 
- 
- J