MWS Cup 2017

課題2: 静的解析

中津留 勇
石淵 一三
石丸 傑

KASPERSKY lab

課題2 担当

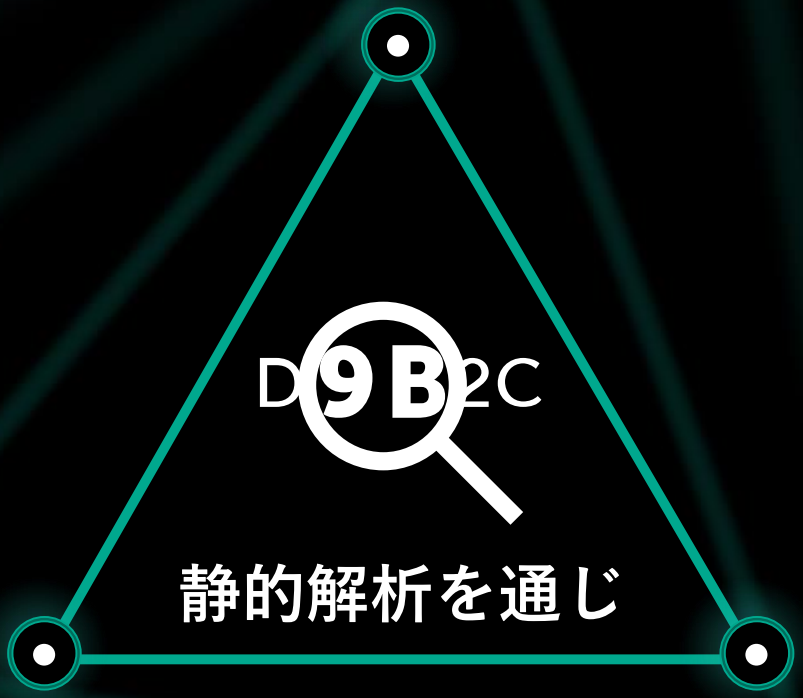
中津留 勇
ynakatsuru[at]secureworks.com
SecureWorks Japan
株式会社

石淵 一三
kazumi.ishibuchi.hh[at]hitachi.com
日立製作所

石丸 傑
suguru.Ishimaru[at]kaspersky.com
株式会社 カスペルスキー

課題2 変わらぬテーマ

マルウェアを正しく理解する



最新情報を得る

実務に近い作業

MWS Cup 2017

課題2 解説

課題2 oni.exe

https://www.cylance.com/ja_jp/blog/jp-oni-ransomware-globeimposter.html

課題2 設問

設問1

呼び出し規約

当該検体の（Windows APIやライブラリ関数以外の）関数呼び出しに使用されている呼出規約を以下から選べ(4点)

設問2

関数の挙動解析

関数 sub_401000 が何をする関数か答えよ (4点)

設問3

被害対象の特定

暗号化の対象となるドライブの種類をすべて選択せよ (4点)

設問4

関数の挙動解析

関数 sub_406306 が何をする関数か答えよ (4点)

設問5

引数の意味

当該検体が SHA-256 ハッシュ値を求めるために使用される関数 sub_404C3Eは、4番目の引数によってどのような処理の変化が起きるか答えよ (4点)

設問6

格納データの意味

当該検体のリソース EXDATA に格納されている112番のデータは、復号しても意味不明なデータとなるが、本来どのような情報が格納されているべきか、コードから推測せよ(5点)

KASPERSKY lab

課題2 解析のポイント

難しく考えない

アセンブラ言語
そのものは単純

わかりやすい
文字を探す

- 文字列
- 定数
- API

コード全てを
読む必要はない

必要な部分を
必要な分だけ
集中して読む

**困った時の
Web検索**

Google先生や
MSDNは貴重
な情報元

設問1 呼び出し規約

当該検体の（Windows APIやライブラリ関数以外の）関数呼び出しに使用されている呼出規約を以下から選べ(4点)

- ~~__cdecl~~ 引数をスタックに経由で渡す
- ~~__stdcall~~ 引数をスタック経由で渡す
- __fastcall 引数をECX, EDX,スタックを経由して渡す
- ~~__thiscall~~ 引数をスタックに、thisポインターをECXに格納
- ~~others~~ 他に適切な答えがある為却下

スタックだけでなくECX、EDX経由で値を渡していることがわかる

```
text:00406306
text:00406306                           ; int __cdecl arg3(DWORD dwBytes)
text:00406306                           arg3            proc near               ; CODE XREF: sub_4059AB+31↑p
text:00406306                                                                   ; sub_405A64+8F↑p
text:00406306                                                                   ; sub_405BAD+518↑p
text:00406306                                                                   ; sub_405BAD+553↑p
text:00406306
text:00406306                           enc_buffer      = dword ptr -0Ch
text:00406306                           var_8           = dword ptr -8
text:00406306                           rc4key          = dword ptr -4
text:00406306                           dwBytes         = dword ptr  4          ; arg3
text:00406306
text:00406306 000 83 EC 0C                              sub     esp, 0Ch
text:00406309 00C 53                                    push    ebx
text:0040630A 010 55                                    push    ebp
text:0040630B 014 56                                    push    esi
text:0040630C 018 57                                    push    edi
text:0040630D 01C 6A 00                                 push    0               ; dwMaximumSize
text:0040630F 020 68 00 10 00 00                        push    1000h           ; dwInitialSize
text:00406314                           arg1 = ecx
text:00406314 024 8B D9                                 mov     ebx, arg1       ; ecx = rc4key
text:00406316                           arg2 = edx
text:00406316 024 89 54 24 18                           mov     [esp+24h+enc_buffer], arg2
text:0040631A 024 6A 00                                 push    0               ; flOptions
text:0040631C 028 89 5C 24 24                           mov     [esp+28h+rc4key], ebx
text:00406320 028 FF 15 B0 10 41 00                     call    ds:HeapCreate
text:00406326 01C 8B 6C 24 20                           mov     ebp, [esp+1Ch+dwBytes] ; arg3
```

設問2 関数の挙動解析

関数 sub_401000 が何をする関数か答えよ (4点)

全プロセス中からsql、outlook、ssms、postgreに一致したプロセス名のPIDを基にtaskkillを用いてプロセスツリーごと強制終了する関数

関数のアドレスを取得

- Process32FirstW
- Process32NextW
- CreateToolhelp32Snapshot

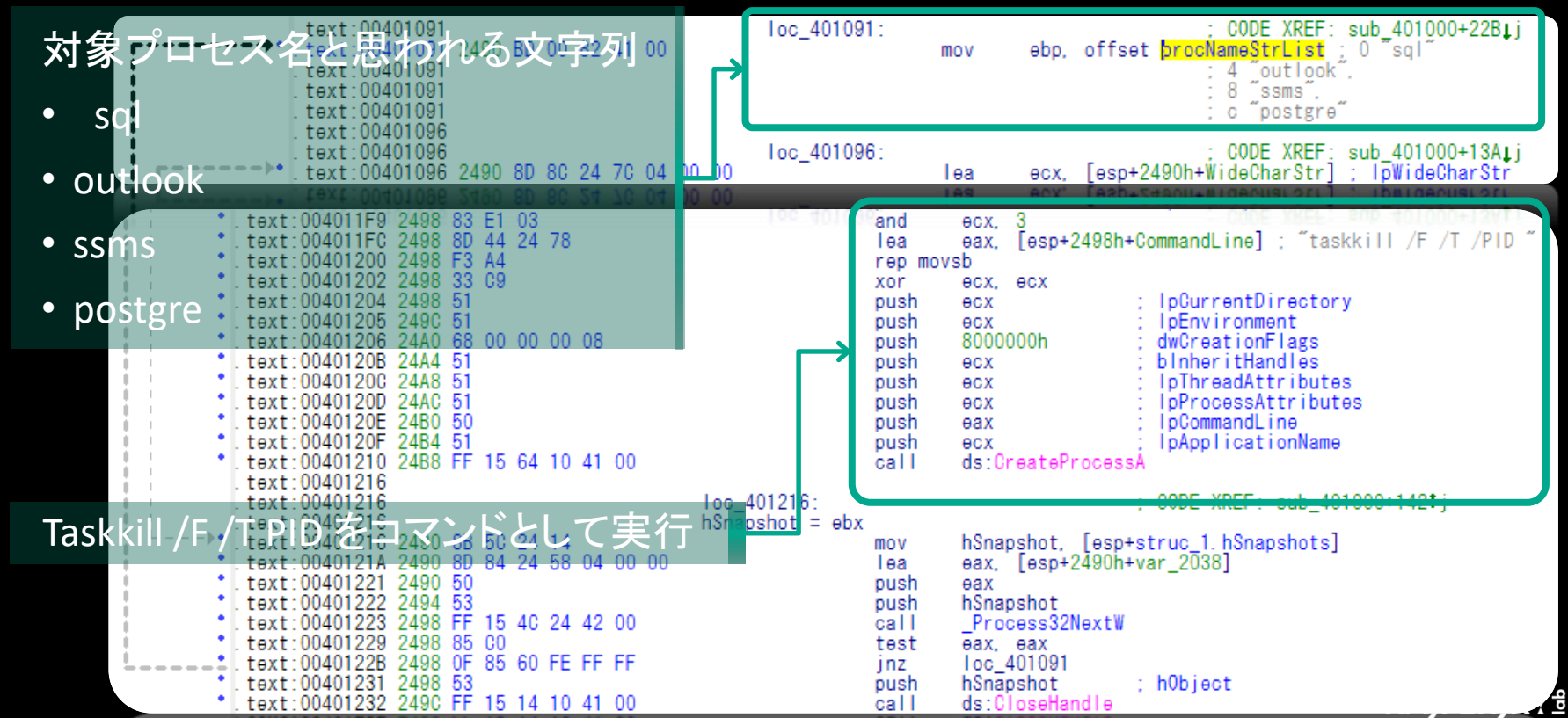
CreateToolhelp32Snapshotでスナップショットを作成

Process32FirstWでスナップショット内の最初のプロセス情報を取得

設問2 関数の挙動解析

関数 sub_401000 が何をする関数か答えよ (4点)

全プロセス中からsql、outlook、ssms、postgreに一致したプロセス名のPIDを基にtaskkillを用いてプロセスツリーごと強制終了する関数

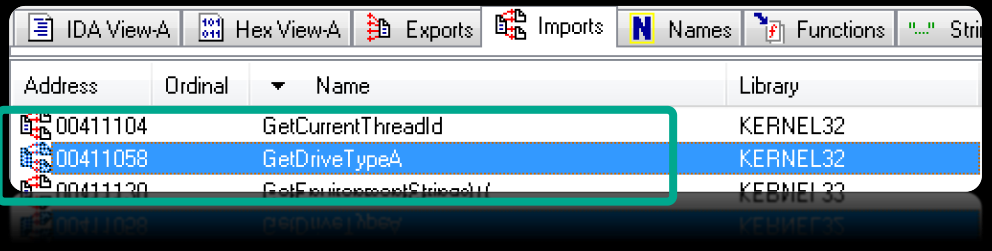


設問3 被害対象の特定

暗号化の対象となるドライブの種類をすべて選択せよ(4点)

- 固定ドライブ
- 取り外し可能ドライブ
- ネットワークドライブ
- ~~書き込み可能なCD/DVDドライブ~~

GetDriveTypeAをImportsから探す



戻り値と比較している値の意味を調べる

```
.text:004058B8
.text:004058B8 F6 C3 01
.text:004058BB 74 46
.text:004058BD 8D 85 F8 EF FF FF
.text:004058C3 50
.text:004058C4 FF 15 58 10 41 00
.text:004058CA 83 F8 03
.text:004058CD 74 0A
.text:004058CF 83 F8 02
.text:004058D2 74 05
.text:004058D4 83 F8 04
.text:004058D7 75 2A
```

```
loc_4058B8:                             ; CODE XREF: sub_405865+A6↓j
        test    bl, 1
        jz      short loc_405903
        lea     eax, [ebp+String2]
        push    eax                     ; lpRootPathName
        call    ds:GetDriveTypeA
        cmp     eax, DRIVE_FIXED
        jz      short loc_4058D9
        cmp     eax, DRIVE_REMOVABLE
        jz      short loc_4058D9
        cmp     eax, DRIVE_REMOTE
        jnz     short loc_405903
```

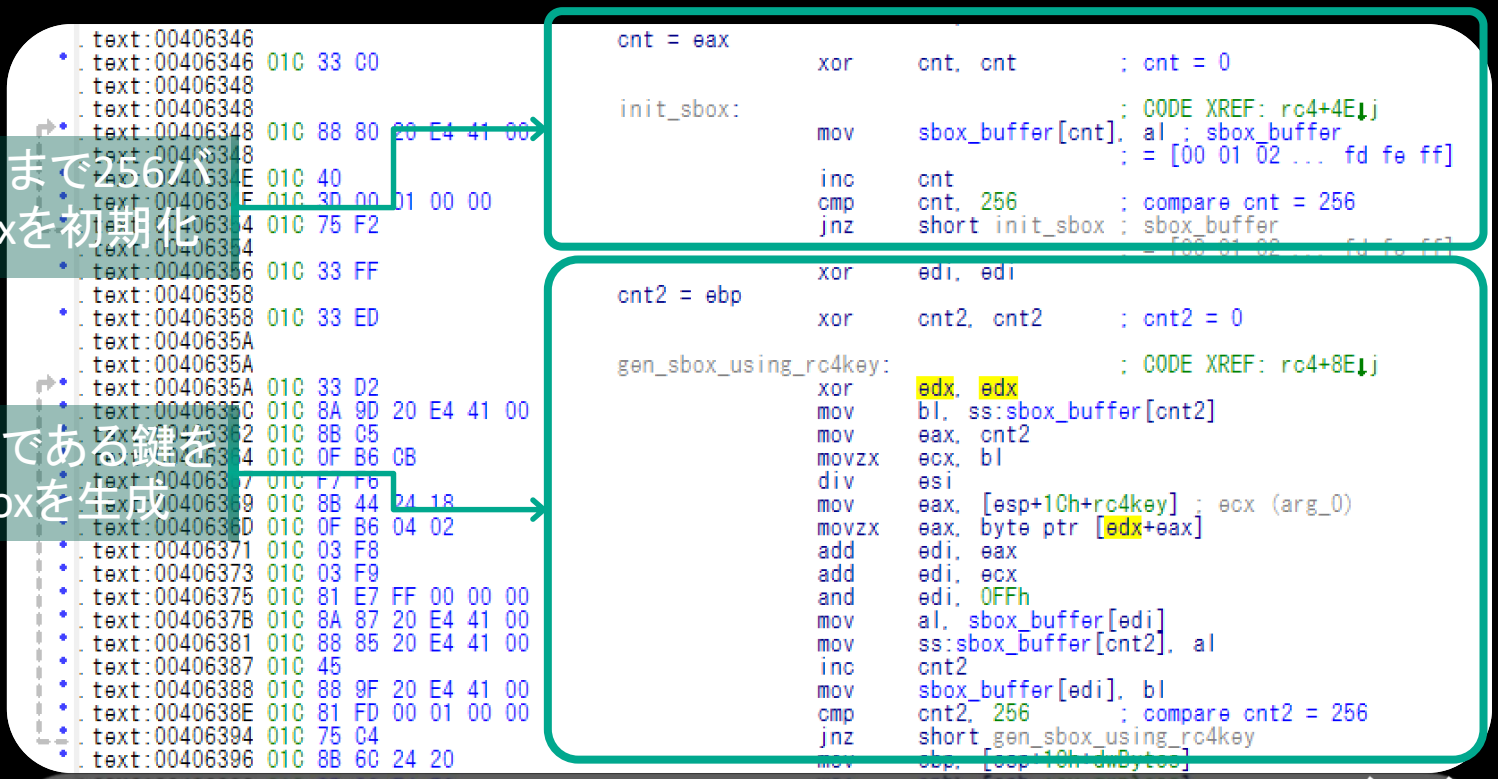
設問4 関数の挙動解析

関数 sub_406306 が何をする関数か答えよ (4点)

RC4で復号化する関数。第一引数に暗号データへのアドレス、第二引数に鍵、第三引数にデータサイズがそれぞれ入り、戻り値は復号データへのアドレスが入る

00 から ff まで256バイトのsboxを初期化

第一引数である鍵を用いてsboxを生成



設問4 関数の挙動解析

関数 sub_406306 が何をする関数か答えよ (4点)

RC4で復号化する関数。第一引数に暗号データへのアドレス、第二引数に鍵、第三引数にデータサイズがそれぞれ入り、戻り値は復号データへのアドレスが入る

1バイトづつXOR

ストリーム暗号

```
.text:004063B2 01C 29 7C 24 10                  sub     [esp+1Ch+enc_buffer], edi ; enc_data = [esp+1ch] + edi
.text:004063B6                          dec_buffer = ebx
.text:004063B6 01C 8B DF                        mov     dec_buffer, edi
.text:004063B8 01C 8B FD                        mov     edi, ebp
.text:004063BA 01C 8B 6C 24 10                  mov     ebp, [esp+1Ch+enc_buffer]
.text:004063BE
.text:004063BE                          rc4_loop:                               ; CODE XREF: rc4+10B↓j
.text:004063BE 01C 42                           inc     edx
.text:004063BF 01C 81 E2 FF 00 00 00            and     edx, 0FFh
.text:004063C5 01C 8A 8A 20 E4 41 00            mov     cl, sbox_buffer[edx]
.text:004063CB 01C 0F B6 C1                     movzx   eax, cl
.text:004063CE 01C 03 F0                        add     esi, eax
.text:004063D0 01C 81 E6 FF 00 00 00            and     esi, 0FFh
.text:004063D6 01C 8A 86 20 E4 41 00            mov     al, sbox_buffer[esi]
.text:004063DC 01C 88 8E 20 E4 41 00            mov     sbox_buffer[esi], cl
.text:004063E2 01C 88 82 20 E4 41 00            mov     sbox_buffer[edx], al
.text:004063E8 01C 0F B6 8E 20 E4 41 00         movzx   ecx, sbox_buffer[esi]
.text:004063EF 01C 0F B6 C0                     movzx   eax, al
.text:004063F2 01C 03 C8                        add     ecx, eax
.text:004063F4 01C 81 E1 FF 00 00 80            and     ecx, 800000FFh
.text:004063FA 01C 79 08                        jns     short loc_406404
.text:004063FC 01C 49                           dec     ecx
.text:004063FD 01C 81 C9 00 FF FF FF            or      ecx, 0FFFFFF00h
.text:00406403 01C 41                           inc     ecx
.text:00406404
.text:00406404                          loc_406404:                             ; CODE XREF: rc4+F4↑j
.text:00406404 01C 8A 81 20 E4 41 00            mov     al, sbox_buffer[ecx]
.text:0040640A 01C 32 04 2B                     xor     al, [dec_buffer+ebp] ; enc_data
.text:0040640D 01C 88 03                        mov     [dec_buffer], al
.text:0040640F 01C 43                           inc     dec_buffer
.text:00406410 01C 4F                           dec     edi
.text:00406411 01C 75 AB                        jnz     short rc4_loop
.text:00406413 01C 8B 7C 24 14                  mov     edi, [esp+1Ch+var_8]
```

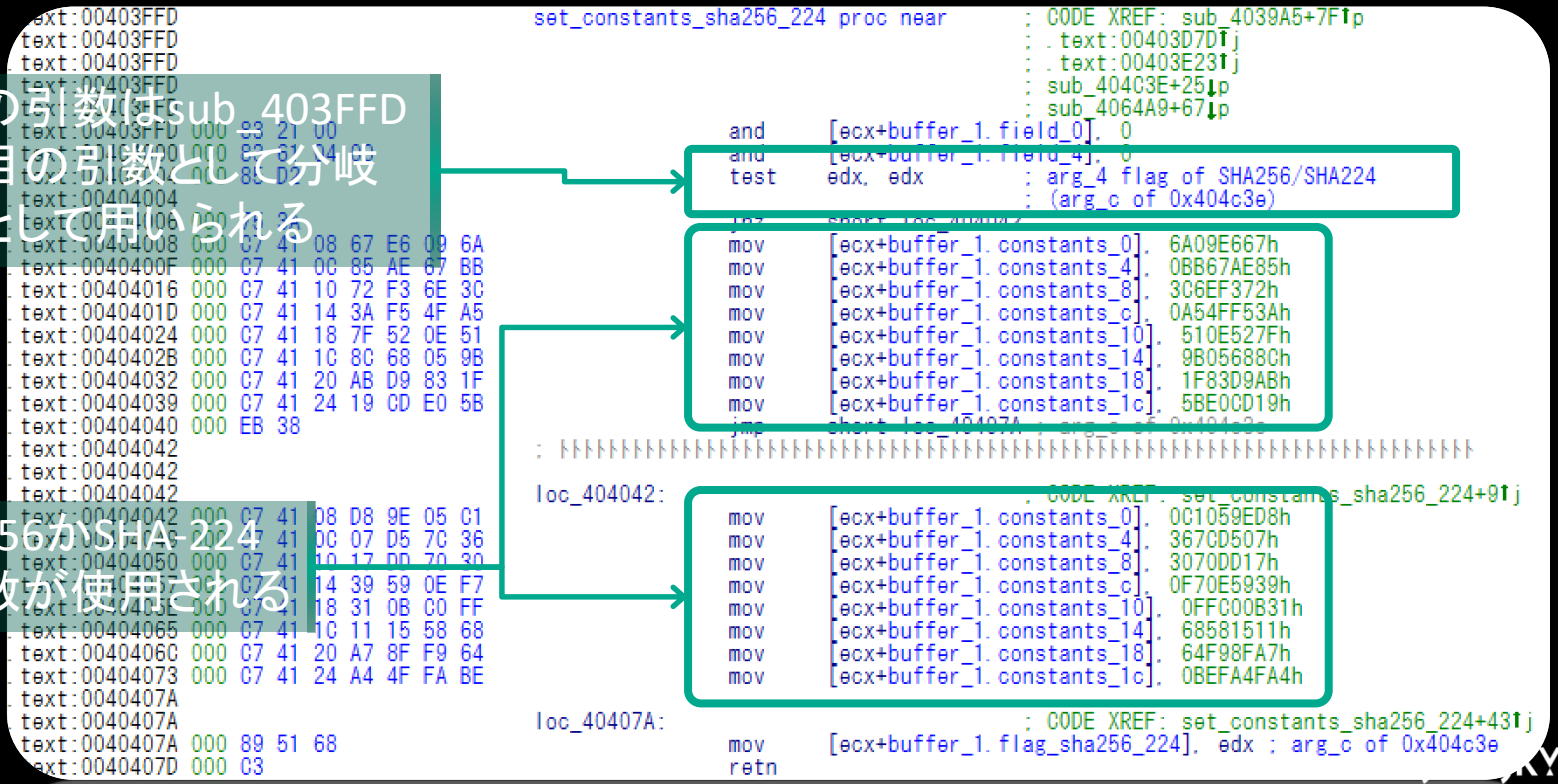
設問5 引数の意味

当該検体が SHA-256 ハッシュ値を求めるために使用される関数 sub_404C3Eは、4番目の引数によってどのような処理の変化が起きるか答えよ (4点)

引数が0の場合はSHA-256のハッシュ値、それ以外の場合にはSHA-224のハッシュ値を求めるように変化する。

4番目の引数はsub_403FFDの2番目の引数として分岐フラグとして用いられる

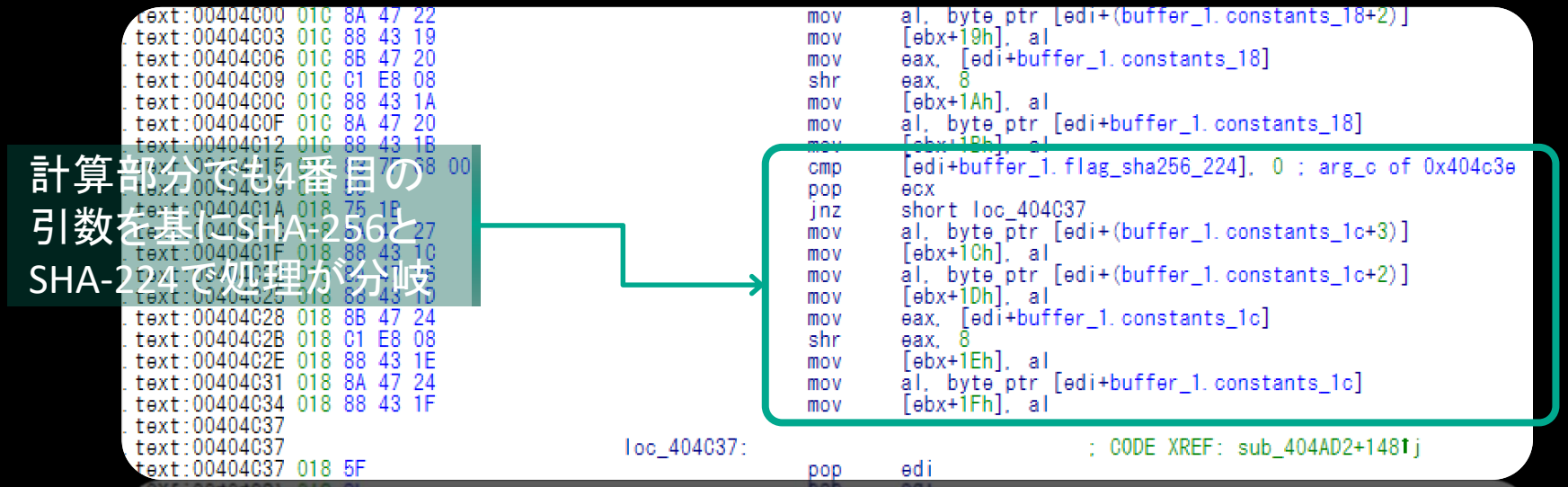
SHA-256かSHA-224の定数が使用される



設問5 引数の意味

当該検体が SHA-256 ハッシュ値を求めるために使用される関数 sub_404C3Eは、4番目の引数によってどのような処理の変化が起きるか答えよ (4点)

引数が0の場合はSHA-256のハッシュ値、それ以外の場合にはSHA-224のハッシュ値を求めるように変化する。



設問5 引数の意味

当該検体が SHA-256 ハッシュ値を求めるために使用される関数 sub_404C3Eは、4番目の引数によってどのような処理の変化が起きるか答えよ (4点)

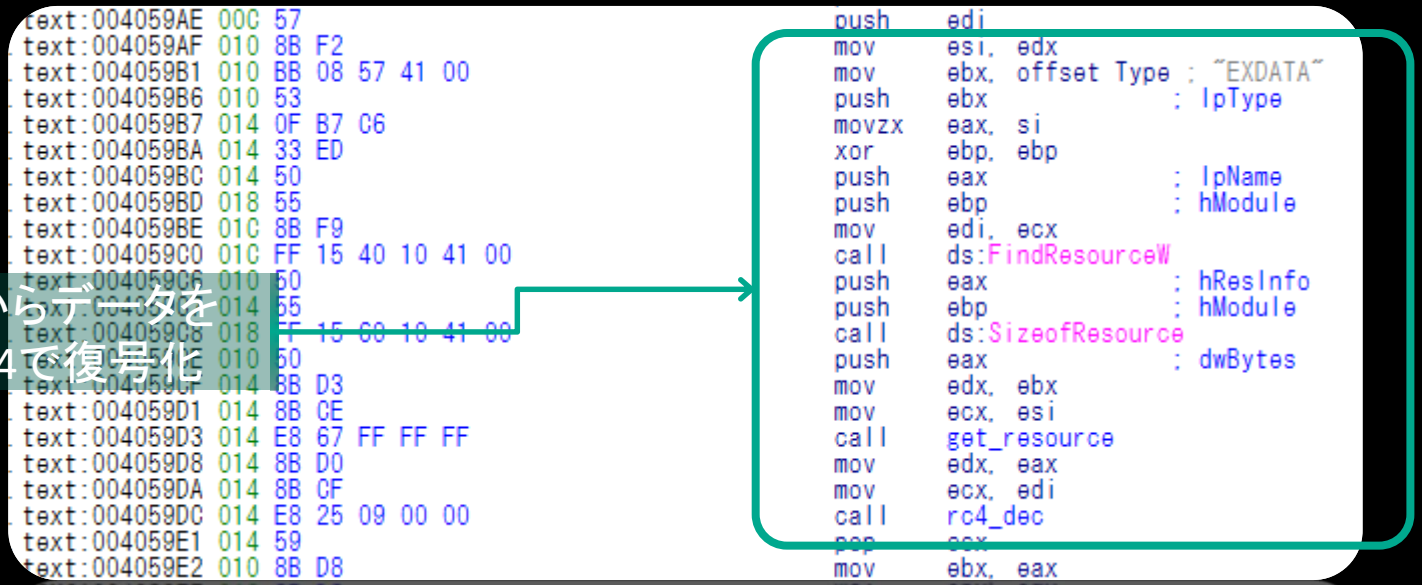
引数が0の場合はSHA-256のハッシュ値、それ以外の場合にはSHA-224のハッシュ値を求めるように変化する。

設問6 格納データの意味

当該検体のリソース EXDATA に格納されている112番のデータは、復号しても意味不明なデータとなるが、本来どのような情報が格納されているべきか、コードから推測せよ(5点)

拡張子リストが“,”（カンマ）を区切りとして格納されているべきと考えられる。また、最初の区切り文字を“ ”（スペース）にも変更することが可能であり、当該拡張子を持つファイルはファイルサイズに関わらずデータ全体を暗号化対象とする

EXDATAからデータを取得、RC4で復号化



設問6 格納データの意味

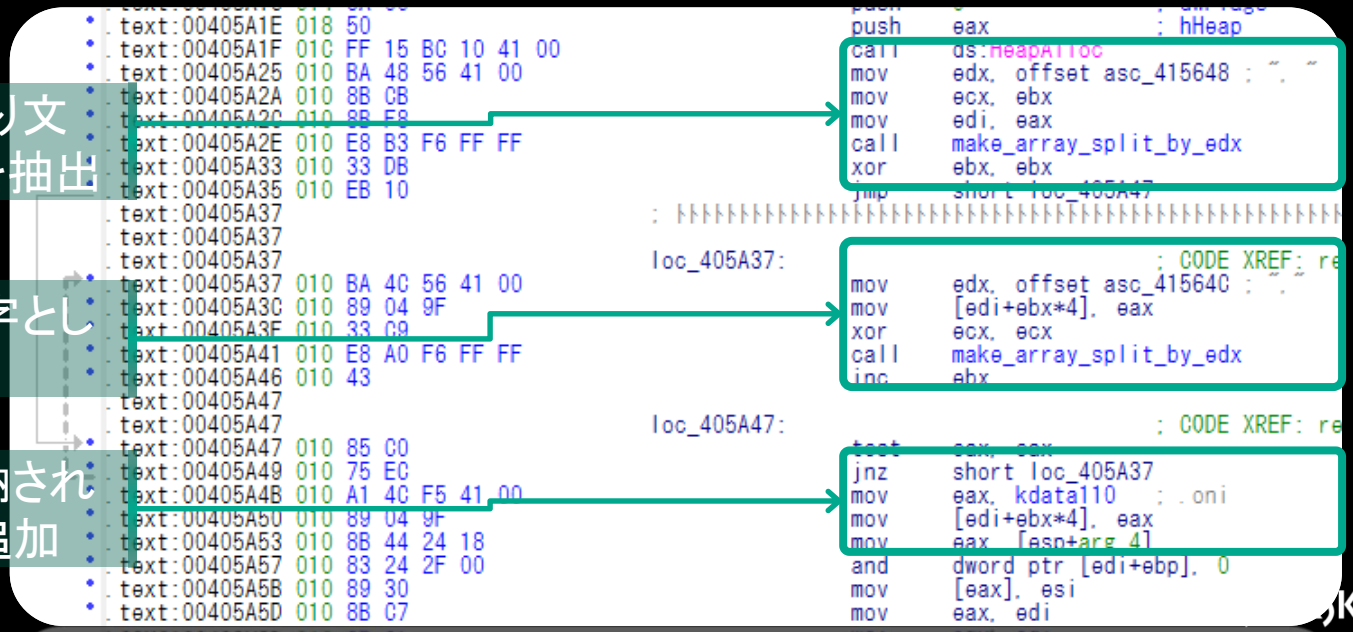
当該検体のリソース EXDATA に格納されている112番のデータは、復号しても意味不明なデータとなるが、本来どのような情報が格納されているべきか、コードから推測せよ(5点)

拡張子リストが“,”（カンマ）を区切りとして格納されているべきと考えられる。また、最初の区切り文字を“ ”（スペース）にも変更することが可能であり、当該拡張子を持つファイルはファイルサイズに関わらずデータ全体を暗号化対象とする

“, ”と“ ”を区切り文字としてデータを抽出

“, ”を区切り文字としてデータを抽出

KDATA110に格納されている“.oni”を追加

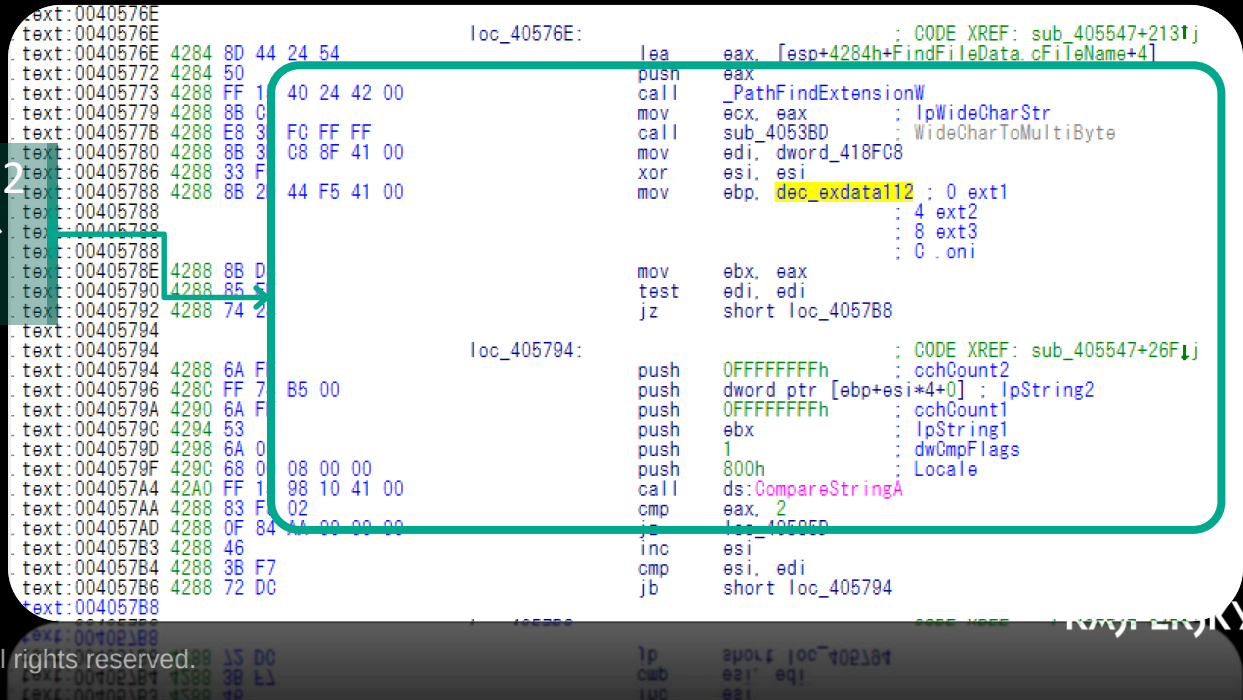


設問6 格納データの意味

当該検体のリソース EXDATA に格納されている112番のデータは、復号しても意味不明なデータとなるが、本来どのような情報が格納されているべきか、コードから推測せよ(5点)

拡張子リストが“,”（カンマ）を区切りとして格納されているべきと考えられる。また、最初の区切り文字を“ ”（スペース）にも変更することが可能であり、当該拡張子を持つファイルはファイルサイズに関わらずデータ全体を暗号化対象とする

復号化されたEXDATA112と暗号化対象となるファイルの拡張子を比較



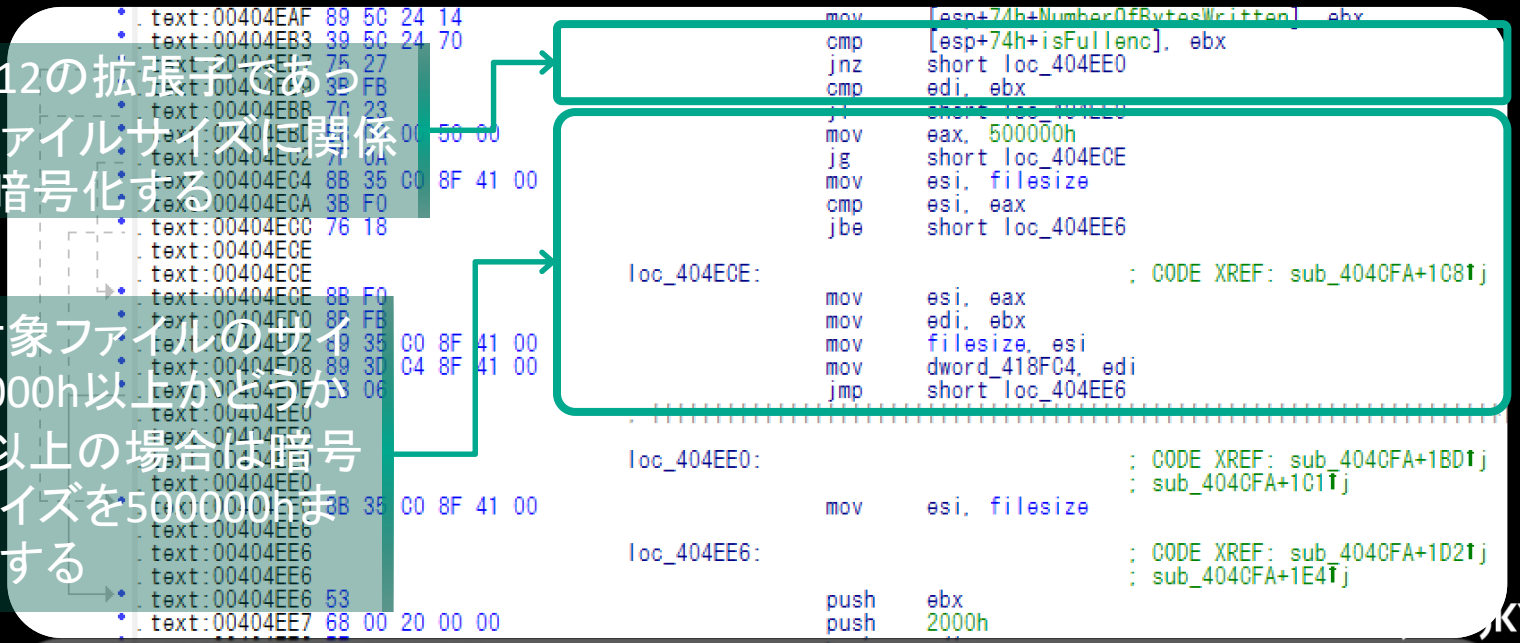
設問6 格納データの意味

当該検体のリソース EXDATA に格納されている112番のデータは、復号しても意味不明なデータとなるが、本来どのような情報が格納されているべきか、コードから推測せよ(5点)

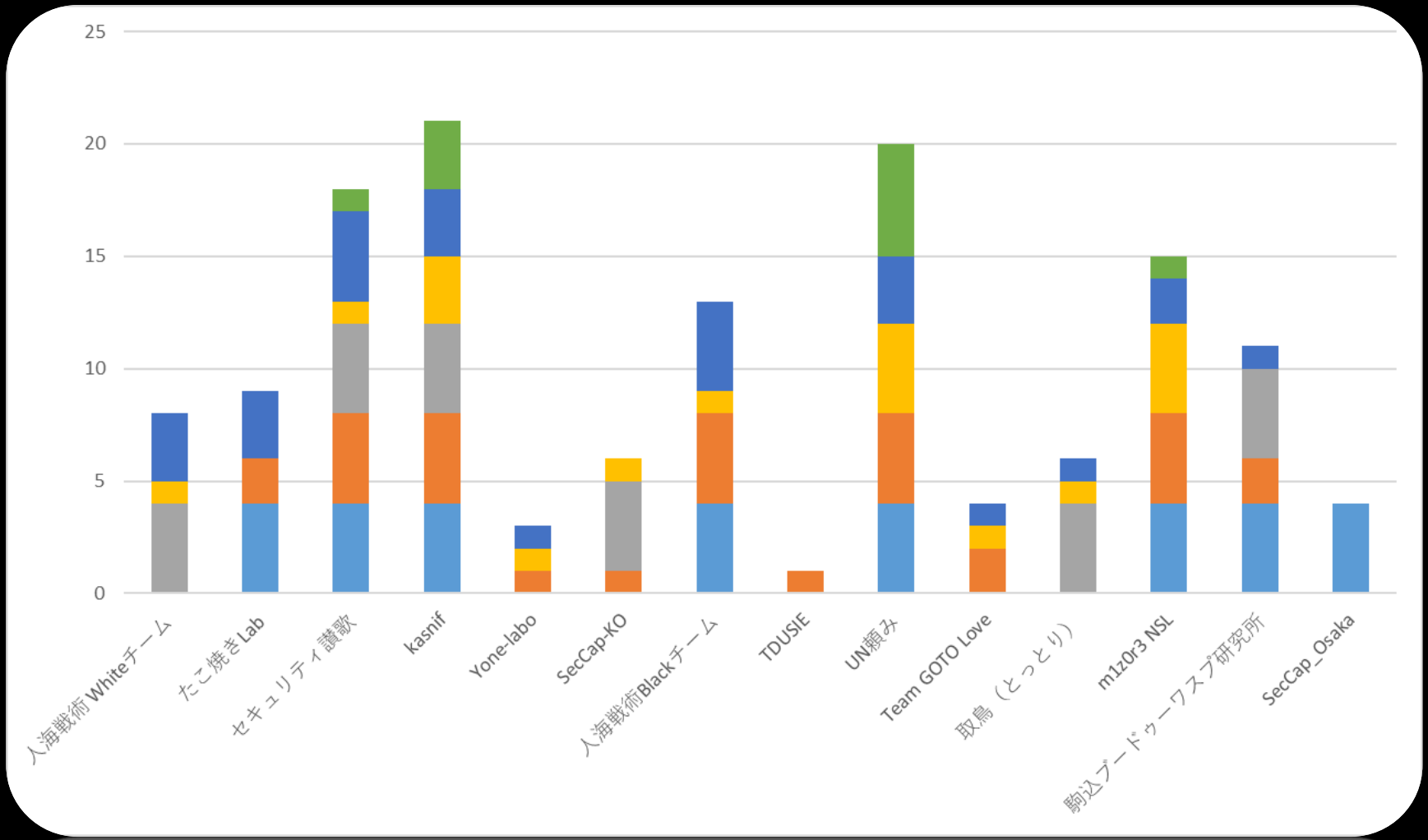
拡張子リストが“,”(カンマ)を区切りとして格納されているべきと考えられる。また、最初の区切り文字を“ ”(スペース)にも変更することが可能であり、当該拡張子を持つファイルはファイルサイズに関わらずデータ全体を暗号化対象とする

EXDATA112の拡張子であった場合ファイルサイズに関係なく全て暗号化する

暗号化対象ファイルのサイズが500000h以上かどうか確認し、以上の場合は暗号化するサイズを500000hまでに設定する



課題２ 採点結果



MWS Cup 2017

課題2 今後の課題

アンケートの意見コメント

今回の課題2に関して良かった点や悪かった点、意見やコメント等あればお願いします

9 件の回答

特になし (2)

（すみません静的解析の担当でないのでわからないです。）

関数の動作を解答する問題で、単に関数の動作のみを解答すればよいのか、マルウェアとしてその関数がどのような悪性な動きをするのかを解答すればよいのかわからなかったです。

難易度、問題数的にもちょうど良かったように思います。

解いてないです

現実的な課題でよかったです

競技中のヒントが有用な情報だった。問題文が一部不明確であり、どの程度詳細に回答すればよいかがわからなかった。

近年流行しているランサムウェアを対象とした静的解析を行うことができ，楽しく課題に取り組むことができました．

問題作成の課題

テーマ選定の難しさ

自動化などを静的解析で適用することの困難さ

より実務に近い検体、2時間で解ける
課題として昇華させる難しさ

問題文の出し方

問題文が一部が伝わりにくかった

どの程度の解答を求めているのか
明示できていなかった

作成委員不足

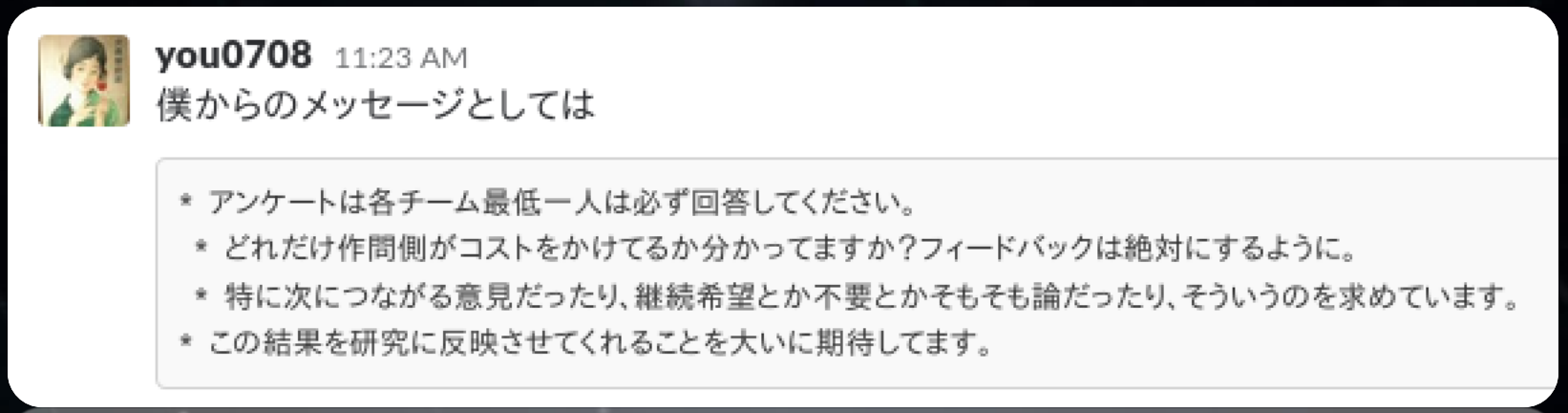
協議中のプレゼンが公開処刑すぎる為
蚤の心臓では耐えらない

問題作成はコードと向き合う必要がある為勉強
になること、なにより楽しい

参加者側の課題

過去問や過去の解答をもとに復習

Write-up書いてみると勉強になる



アンケート答えてくださいね。お願いします。

勉強する際の参考資料

セキュリティ・キャンプ全国大会2015でのマルウエア分析講義(2015-09-10)
https://www.jpcert.or.jp/magazine/acreport-seccamp.html
リバースエンジニアリング入門 - @IT
http://www.atmarkit.co.jp/ait/series/2614/
Edomae 2015 - マルウェアを解析してみよう
http://www.slideshare.net/SatoshiMimura/edomae-2015


WEB

Practical Malware Analysis | No Starch Press
https://www.nostarch.com/malware
O'Reilly Japan - アナライジング・マルウェア
https://www.oreilly.co.jp/books/9784873114552/


書籍

ML に投げてるので、それを入手する
データセットに入っている資料を読む


過去問