2022年
台湾のランサムウェア事例とマルウェア解析

2022年12月

H. Murakami

CYCRAFT

紹介事項

発表者： ごく普通の町のデジタルフォレンジック屋さん (村上)

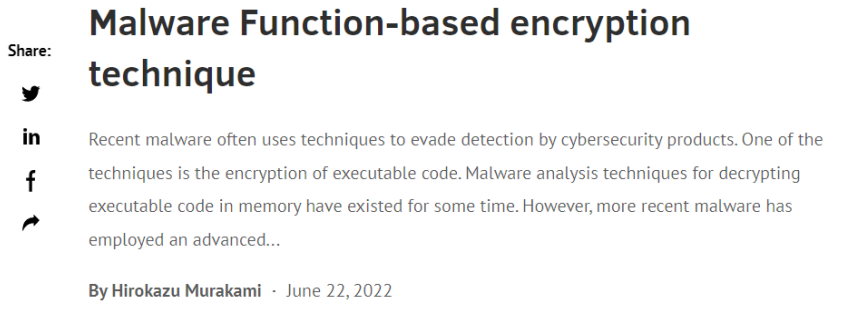
所　属： CyCraft Japan (本社は台湾)

東京電機大学　サイバーセキュリティ研究所　研究員

静岡大学　創造科学技術大学院　博士課程（西垣研究室）

第001404号

Analyst Number 5549


Share:

Malware Function-based encryption technique

Recent malware often uses techniques to evade detection by cybersecurity products. One of the techniques is the encryption of executable code. Malware analysis techniques for decrypting executable code in memory have existed for some time. However, more recent malware has employed an advanced...

By Hirokazu Murakami · June 22, 2022

SANS Whitepaper


Share:

Reverse Engineering of WannaCry Worm and Anti Exploit Snort Rules

Today, a lot of malware is being created and utilized. To solve this problem, many researchers study technologies that can quickly respond automatically to detected malware. Using artificial intelligence (AI) is such an example. However, modern AI has difficulty responding to new attack methods. On...

By Hirokazu Murakami · May 27, 2018

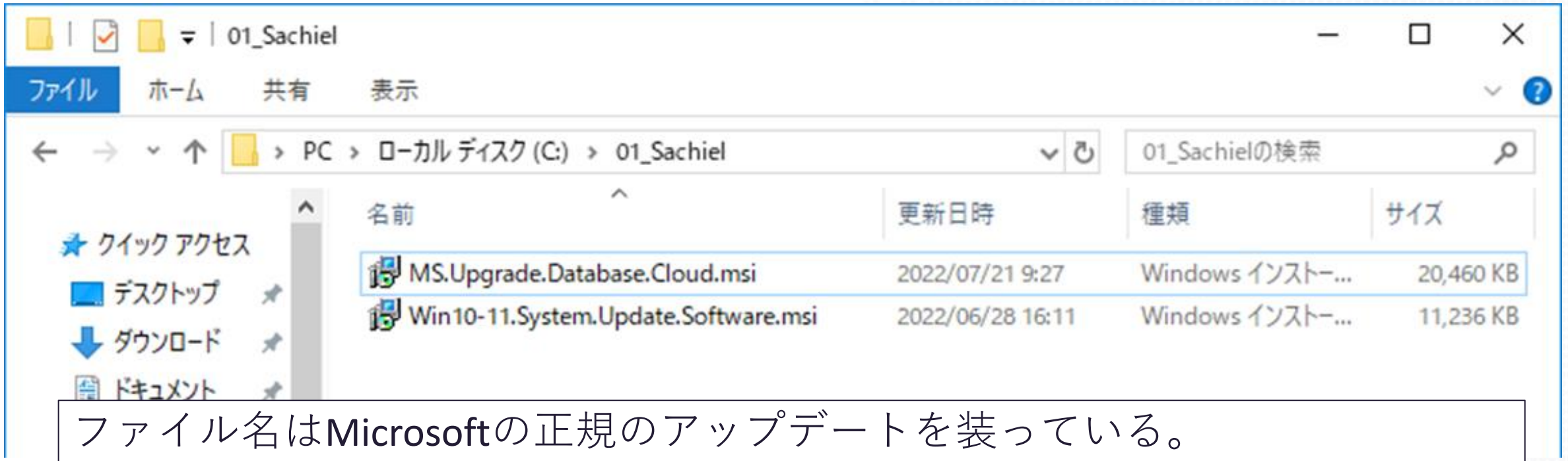
SANS Whitepaper

1. Case 1:.msi形式のMagniberランサムウェア

- 特　徴
  - 2022年6月頃から広がりを確認（顧客から問い合わせあり）。
  - インストーラの.msi形式になっており、基本的にユーザを騙してインストールさせる方法。
  - .msiファイルにDLL形式のマルウェアが埋め込まれている。
  - ファイルの暗号化はAES128 CBCモード、IV、AES共通鍵は乱数で作成し、RSA公開鍵で暗号化。
  - 実行中のプロセスのうち、**オープンできるプロセスに対しかたっぱしから暗号化コードをインジェクション**してスレッドを実行する。
  - **syscallの利用**で検知回避を狙っているとみられる。

1. Case 1:.msi形式のMagniberランサムウェア

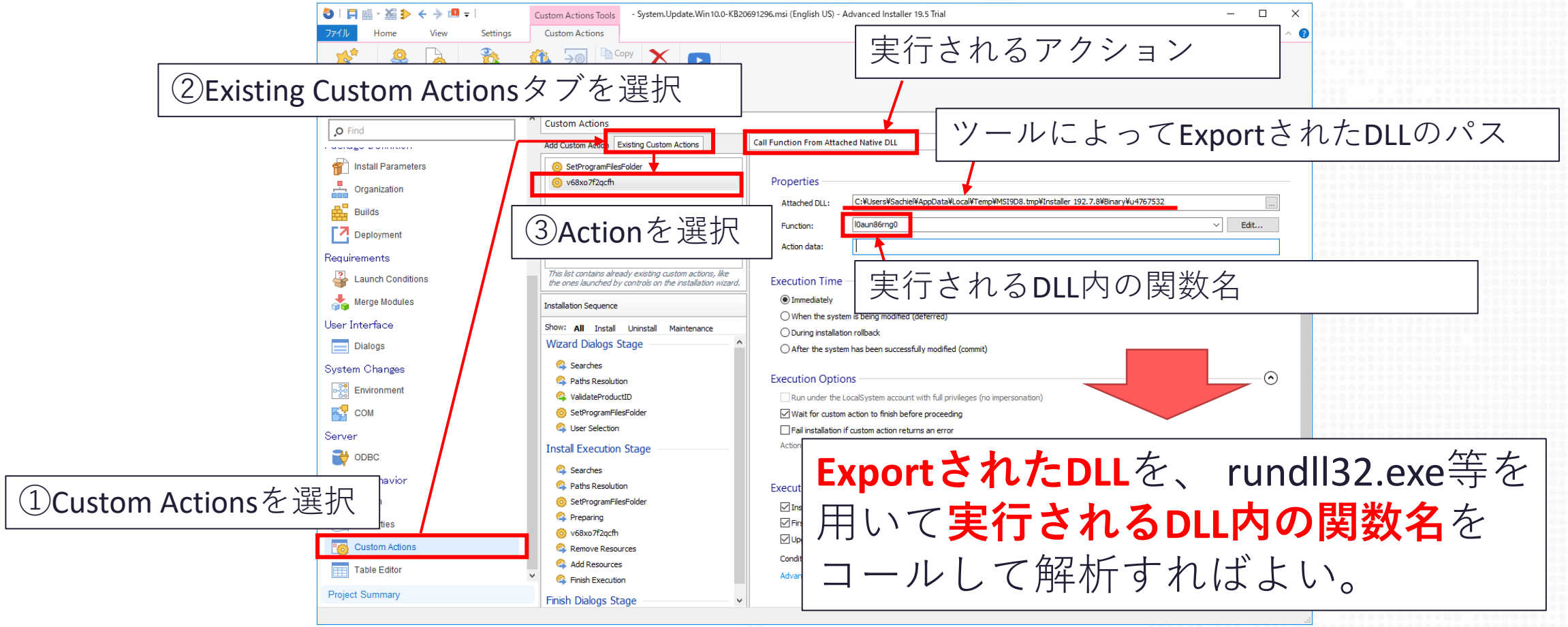
- 当該ファイル



ファイル名はMicrosoftの正規のアップデートを装っている。
偽サイトに掲載したり、管理者になりすましたメールに添付するなどして
ユーザに誤認させて実行させ、感染させることを意図しているとみられる。

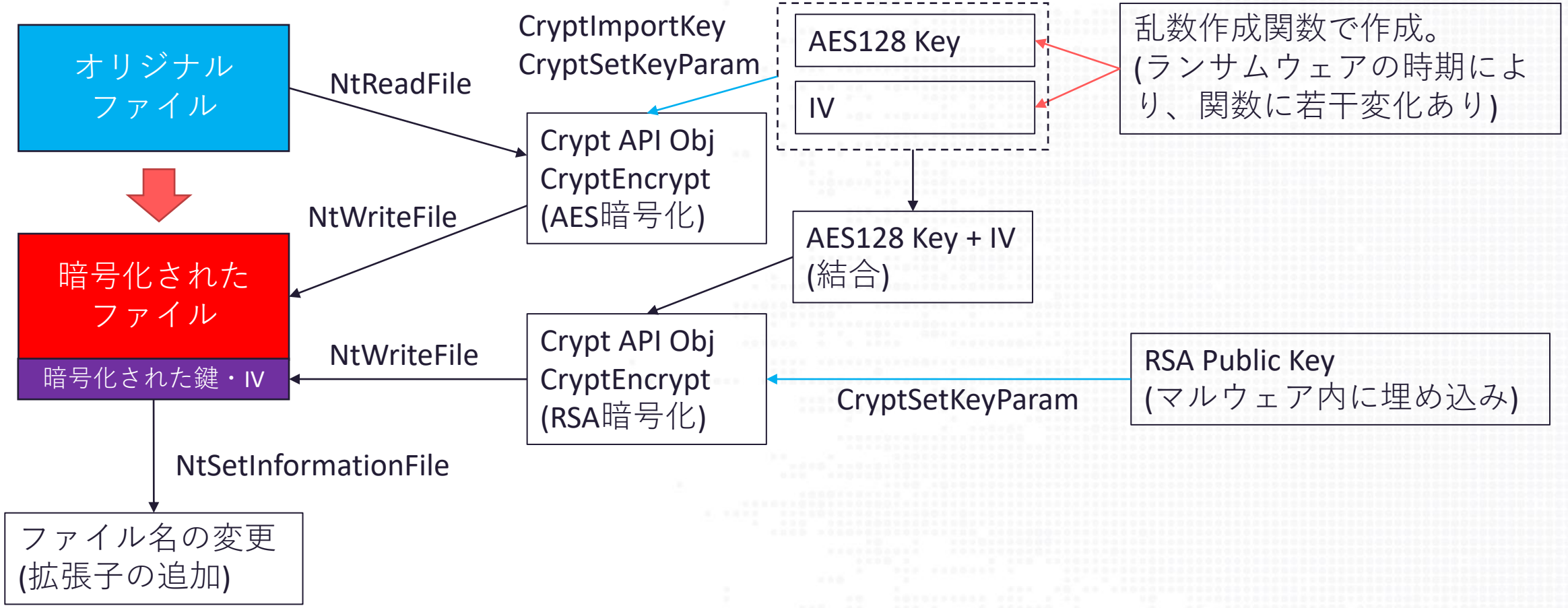
1. Case 1:.msi形式のMagniberランサムウェア

- .msi形式のマルウェアの解析：AdvancedInstallerの利用



1. Case 1:.msi形式のMagniberランサムウェア

- 暗号化プロセス



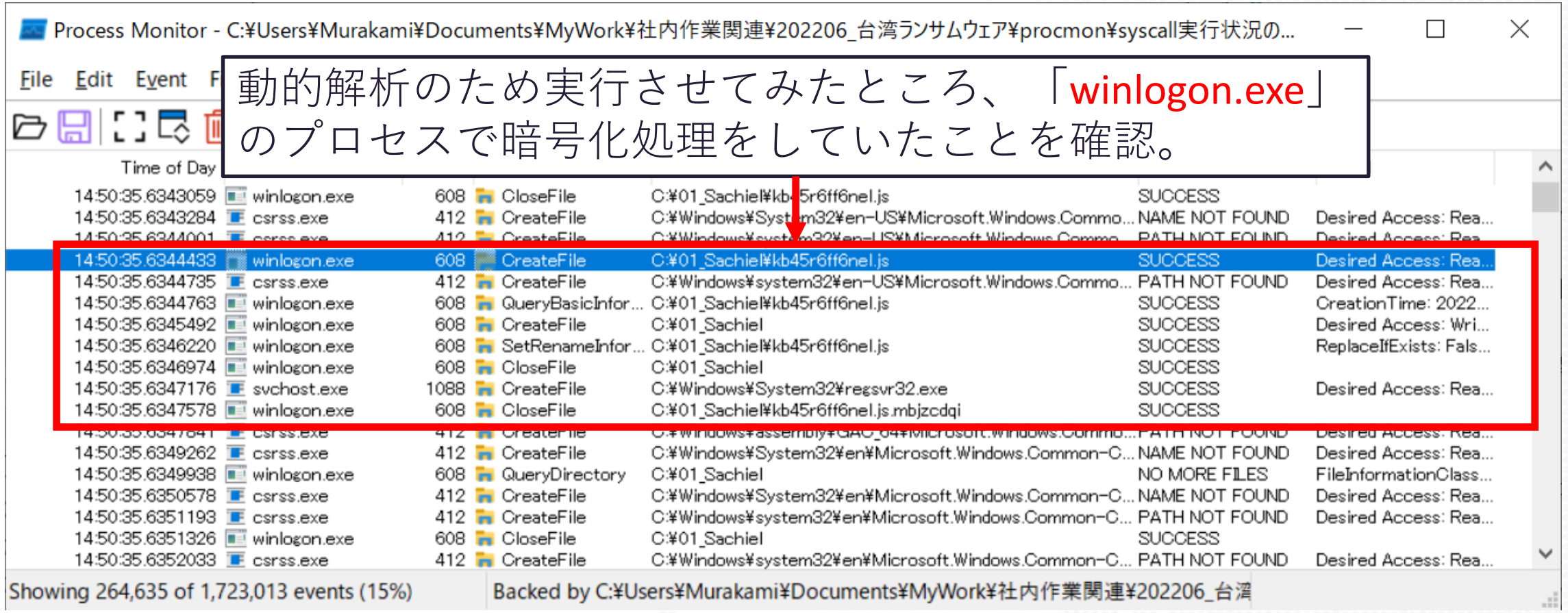
1. Case 1:.msi形式のMagniberランサムウェア

- プロセスサーチおよびプロセスインジェクション
  - syscallで**SystemProcessInformation**を実行してシステム上のプロセス情報を取得する（あっ、いつもの**CreateToolhelp32Snapshot**じゃないんだー（棒））。
  - プロセス毎にNtOpenProcessを実行し、**プロセスのオープンに成功した場合**はそのプロセスのメモリ上に暗号化する**ランサムウェアのコードをコピーし、NtCreateThreadEx で実行**する。

    →多くのプロセスで暗号化処理が行われるため、**かなり重く**なる。
    →暗号化済みかどうかの判定は、拡張子でおこなっている。

1. Case 1:.msi形式のMagniberランサムウェア

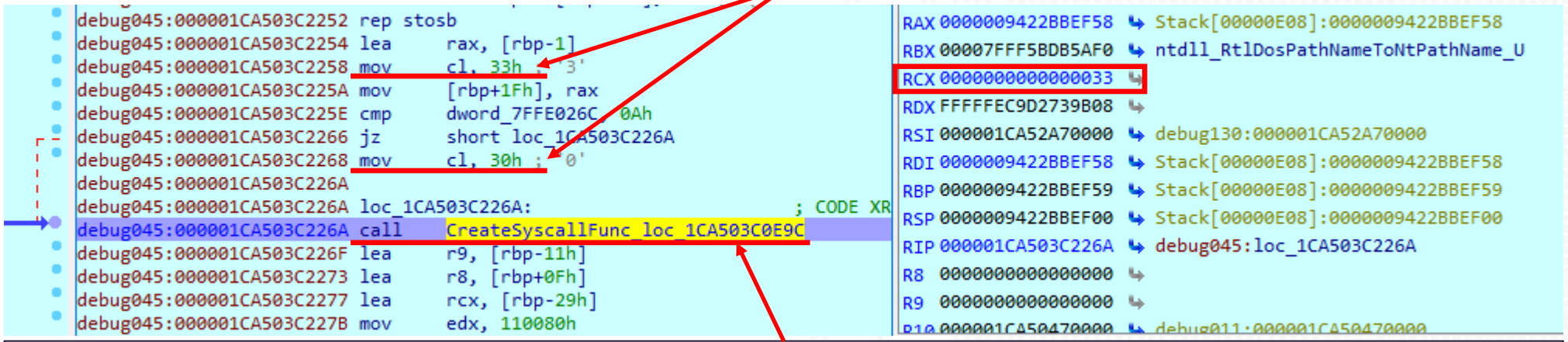
- プロセスサーチおよびプロセスインジェクション



1. Case 1:.msi形式のMagniberランサムウェア

- syscallの頻繁な利用

同じAPIでも、**OSのバージョンによってsyscallで使うコードが異なる**ため、OSのバージョン情報を元にコードを判定。



NtAllocateVirtualMemoryで実行可能領域を確保し、得られたコードでsyscallをするという**小さな関数を作成**。

syscallテーブル参考：https://j00ru.vexillium.org/syscalls/nt/64/

1. Case 1:.msi形式のMagniberランサムウェア

- syscallの頻繁な利用

```
debug045:000001CA503C2268 mov     cl, 30h ; '0'
debug045:000001CA503C226A
debug045:000001CA503C226A loc_1CA503C226A:                        ; CODE XREF: debug045:000001CA503C2266↑j
debug045:000001CA503C226A call    CreateSyscallFunc_loc_1CA503C0E9C
debug045:000001CA503C226F lea     r9, [rbp-11h]
debug045:000001CA503C2273 lea     r8, [rbp+0Fh]
debug045:000001CA503C2277 lea     rcx, [rbp-29h]
debug045:000001CA503C227B mov     edx, 110080h
debug045:000001CA503C2280 mov     dword ptr [rsp+28h], 20h ; ' '
debug045:000001CA503C2288 mov     dword ptr [rsp+20h], 7
debug045:000001CA503C2290 call    rax                             ; NtOpenFile
debug045:000001CA503C2292 test    eax, eax
debug045:000001CA503C2294 jns     short loc_1CA503C229D
debug045:000001CA503C2296
debug045:000001CA503C2296 loc_1CA503C2296:                        ; CODE XREF: debug045:000001CA503C22F2↓j
debug045:000001CA503C2296                                         ; debug045:000001CA503C2371↓j ...
debug045:000001CA503C2296 xor     eax, eax
debug045:000001CA503C2298 jmp     loc_1CA503C23B4
```

```
debug131:000001CA52A80000 ; Segment type: Pure code
debug131:000001CA52A80000 ; Segment permissions: Read/Write/Execute
debug131:000001CA52A80000 debug131 segment byte public 'CODE' use64
debug131:000001CA52A80000 assume cs:debug131
debug131:000001CA52A80000 ;org 1CA52A80000h
debug131:000001CA52A80000 assume es:ntdll_dll, ss:ntdll_dll, ds:ntdll_dll, fs:ntdll_dll, gs
debug131:000001CA52A80000 mov     r10, rcx
debug131:000001CA52A80003 mov     eax, 33h ; '3'
debug131:000001CA52A80008 syscall                                 ; Low latency system call
debug131:000001CA52A8000A retn
debug131:000001CA52A8000A ; ---------------------------------------------------------------
```

syscallによるNtOpenFileのコール例

syscall実行用領域を解放していない
バグ持ち（ふざけんなド下手糞）

1. Case 1:.msi形式のMagniberランサムウェア

- その他の特徴
  - パッカー部は、単純なXORによる暗号化を利用。ただし、ワンステップずつjmpすることで若干の難読化を試みている。
  - スクリプトを展開し、 vssadmin、 wbadmin、 bcdeditを利用してVolume Shadow Copyの消去、バックアップカタログの削除、システム状態バックアップの削除、全てのブートエラーを無視するよう表示ポリシーを変更、スタートアップ修復の無効化をしている。

2. Case 2:Lockbit 2.0 ランサムウェア

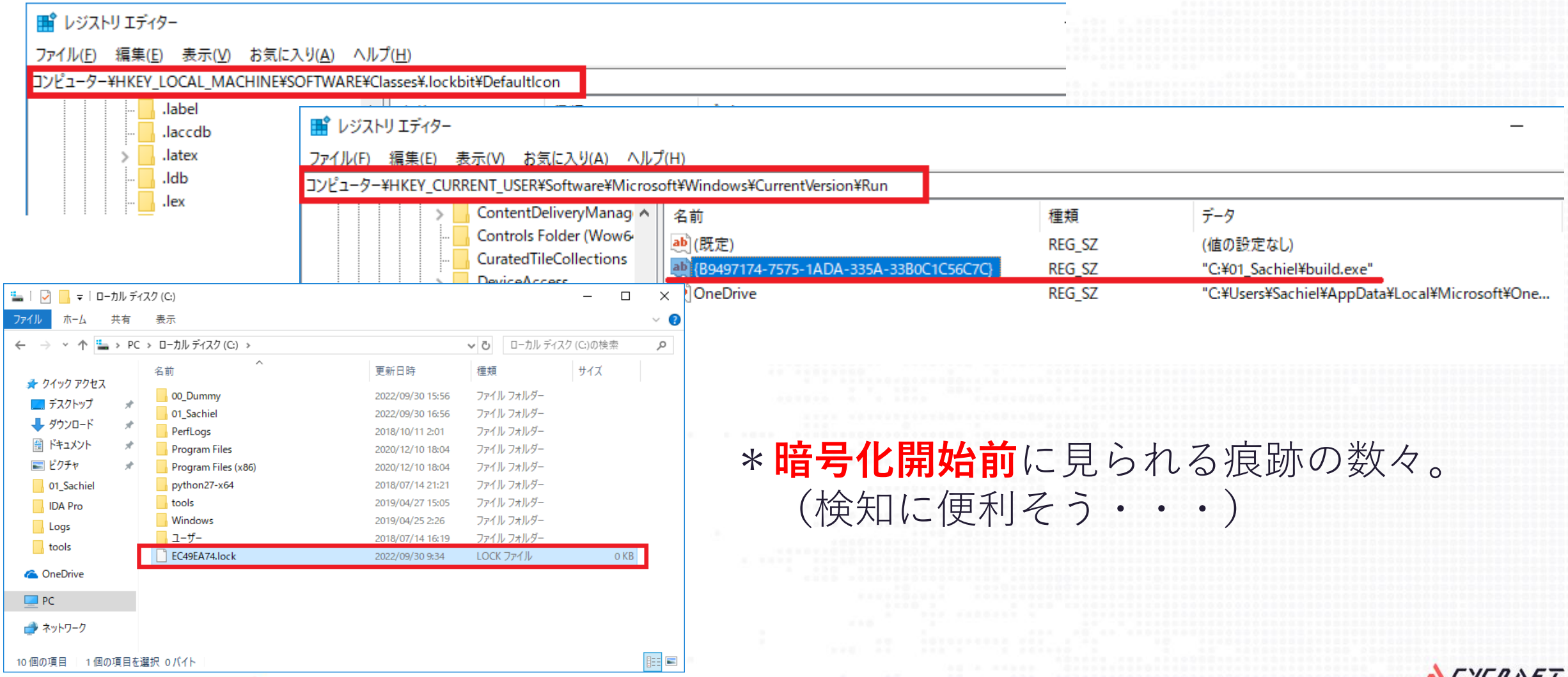
- 特　徴
  - 2022年2月、台湾で被害者の情報がDark Webにリークしていることを発見したことから調査開始となった。
  - このランサムウェアのファイルの暗号化はAES128 CBCモードで、**ファイルの先頭4,096byteしか暗号化しない**。
  - 解析回避、アンチフォレンジックのテクニックがほのかには入っている（ただし、あんまり気にならなかった）が、マルウェアはパックもされていない。
  - プリンタで9999枚脅迫文を印刷、壁紙をマルウェアが頑張って描画など、変なところに力をいれているマルウェアでもある。

MBSDさんの解析が詳しい：　https://www.mbsd.jp/research/20211019/blog/

2. Case 2:Lockbit 2.0 ランサムウェア

- 解析した感想・・・
  - マルウェアが**暗号化を開始する前に色々やってくれる**ので、それらの振る舞いを検知・遮断できるEDR等の機能があれば、割と対策が可能だと思われる。
  - ビルダーでランサムウェアを生成するようにしている影響かもしれないが、コードの構造化や関数化が不十分で、ダラダラ長い＆ifのネストが深い（＝**コードのできが悪い**）印象。
  - 解析回避、アンチフォレンジック等はあるが、既存のライブラリ等を集めて取って付けたような感じで、マルウェア全体の設計は不十分に見える（あちこちに雑さが残っている）。

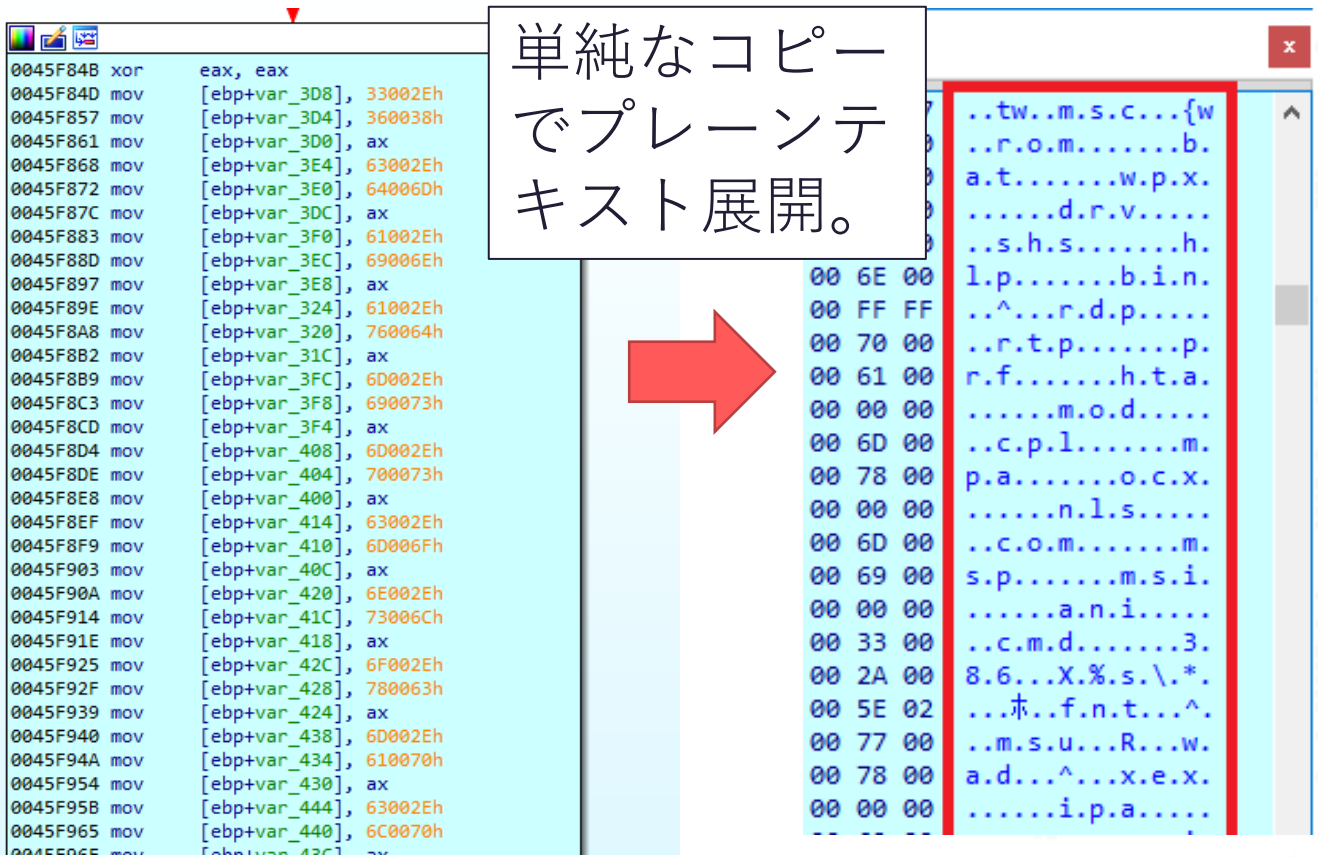
2. Case 2:Lockbit 2.0 ランサムウェア



＊**暗号化開始前**に見られる痕跡の数々。
（検知に便利そう・・・）

2. Case 2:Lockbit 2.0 ランサムウェア

- コードは単調でパラメータも暴露しやすく、解析難易度は高くはない。



2. Case 2:Lockbit 2.0 ランサムウェア

- 解析回避例：

```
004BFF90 var_208= byte ptr -208h
004BFF90
004BFF90 push    ebp
004BFF91 mov     ebp, esp
004BFF93 and     esp, 0FFFFFFF8h
004BFF96 mov     eax, large fs:30h
004BFF9C sub     esp, 480h
004BFFA2 test    byte ptr [eax+68h], 70h
004BFFA6 push    esi
004BFFA7 push    edi
004BFFA8 jz      short loc_4BFFB2
```

PEBのパラメータで判定
→EIPを正常ルートへ

```
004BA73A
004BA73A loc_4BA73A:
004BA73A lea     ecx, [ebp+var_20]
004BA73D push    ecx
004BA73E push    0
004BA740 push    [ebp+var_18]
004BA743 push    [ebp+var_1C]
004BA746 push    0
004BA748 push    0
004BA74A call    eax ; CreateThread_Addr_dword_4F07EC
004BA74C mov     esi, eax
004BA74E cmp     esi, 0FFFFFFFFh
004BA751 jz      short loc_4BA768
```

```
004BA753 lock inc dword_4F0864
004BA75A push    0
004BA75C push    0
004BA75E push    11h
004BA760 push    esi
004BA761 call    Get_NtSetInformationThread_Addr_sub_4154F0
004BA766 call    eax
```

THREADINFOCLASS
のパラメータ

```
004BA768
004BA768 loc_4BA768:
```

デバッガ
からの秘匿

```
004BA748 push    0
004BA74A call    eax ; CreateThread_Addr_dword_4F07EC
004BA74C mov     esi, eax
004BA74E cmp     esi, 0FFFFFFFFh
004BA751 jz      short loc_4BA768
```

```
004BA753 lock inc dword_4F0864
004BA75A nop
004BA75B nop
004BA75C nop
004BA75D nop
004BA75E nop
004BA75F nop
004BA760 nop
004BA761 nop
004BA762 nop
004BA763 nop
004BA764 nop
004BA765 nop
004BA766 nop
004BA767 nop
```

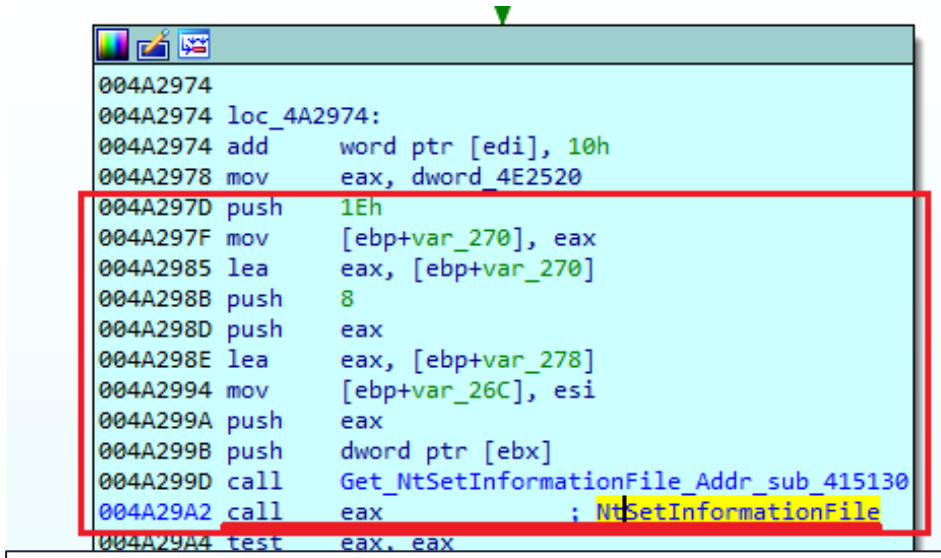
```
004BA768
004BA768 loc_4BA768:
```

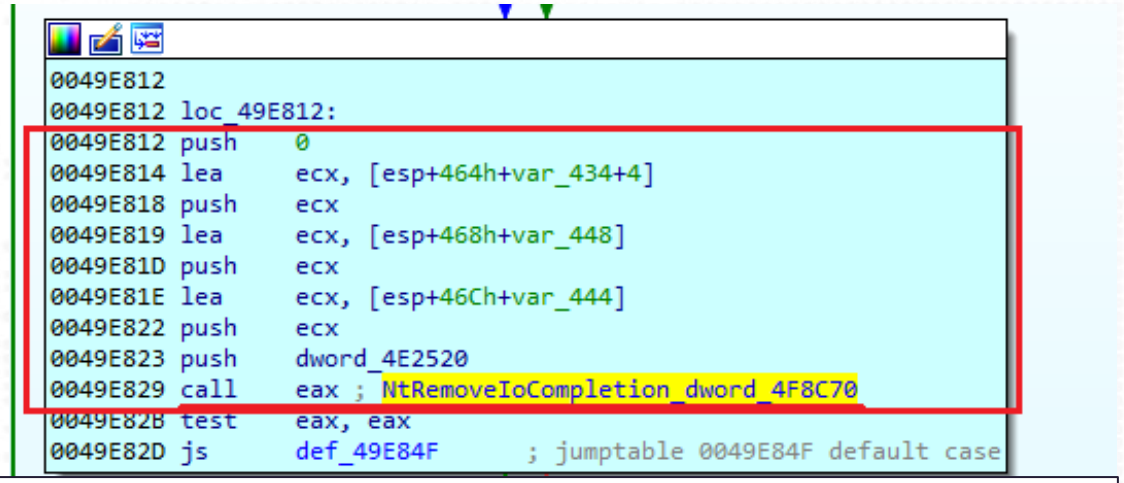
コード除去

2. Case 2:Lockbit 2.0 ランサムウェア

- スレッド間通信

  NtSetInformationFile、 NtRemoveIoCompletionを使って通信。



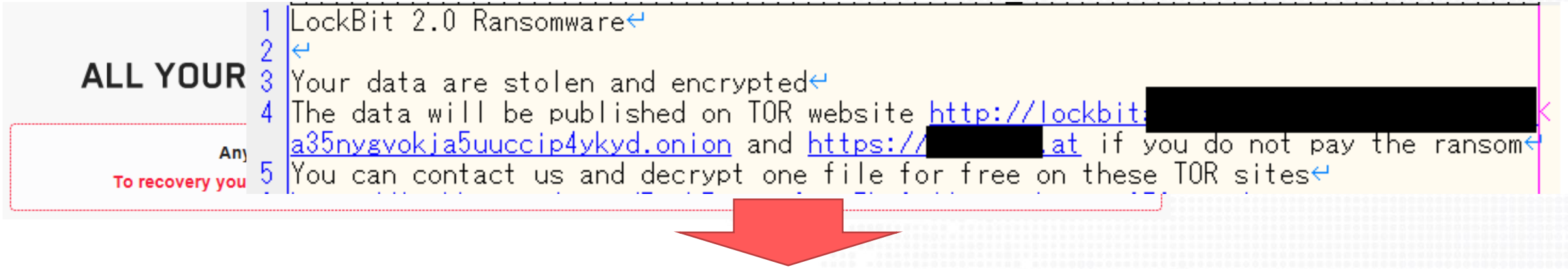


送信元スレッド：
暗号化対象のチェックおよびファイルのオープン、読み込み、IVとAES鍵の生成

受信先スレッド：
暗号化の実行、暗号化結果の書き込み、ファイル名変更（.bitlock拡張子の追加）、ファイルクローズ

2. Case 2:Lockbit 2.0 ランサムウェア

- 発生した事象：
  - Lockbit 2.0ランサムウェアに感染し、被害端末のデータが暗号化された。
  - Lockbit 2.0の脅迫文には、さらに指定のWebサイトやTorのサイトで盗まれたデータを暴露すると脅迫。

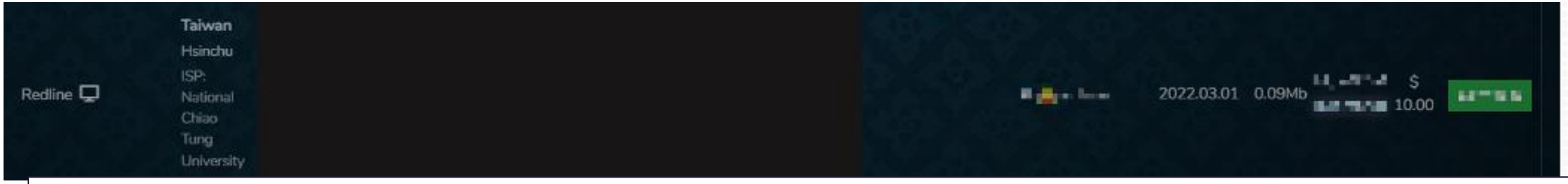


ん？Lockbit 2.0にデータ収集やC&Cサーバへの転送機能って無くない？

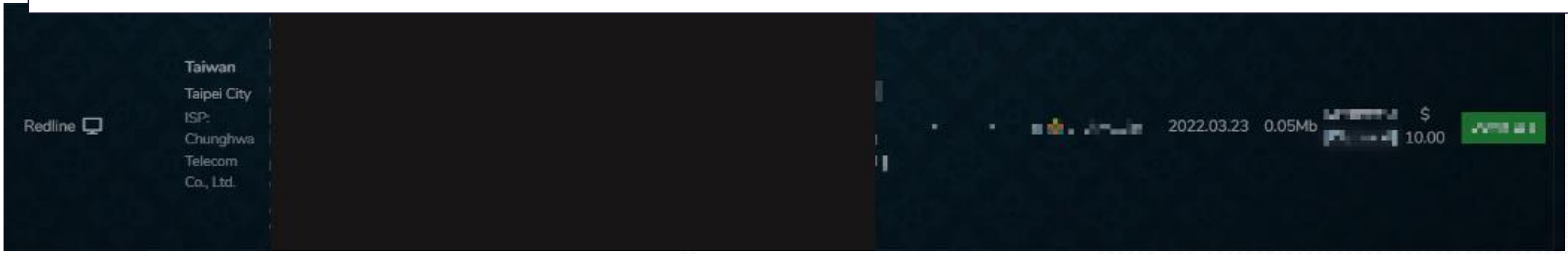
2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（1）Dark Webでの認証情報漏洩



「Redline」という情報窃盗組織により、2022年3月に台湾のユーザとみられる認証情報が複数件販売されていることを確認。

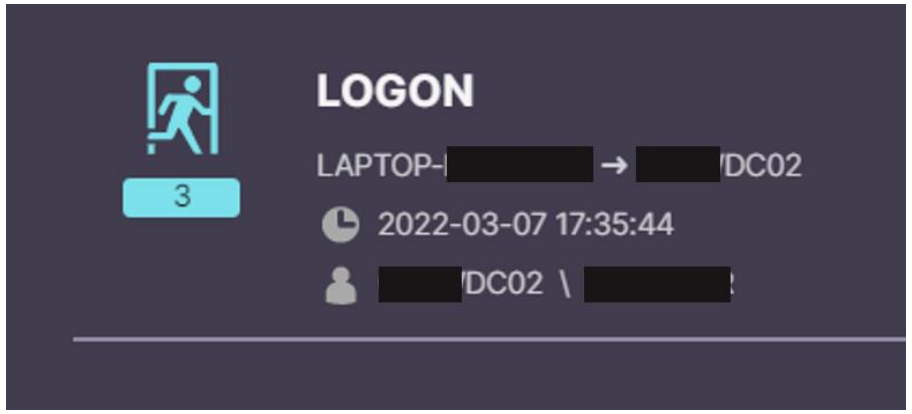


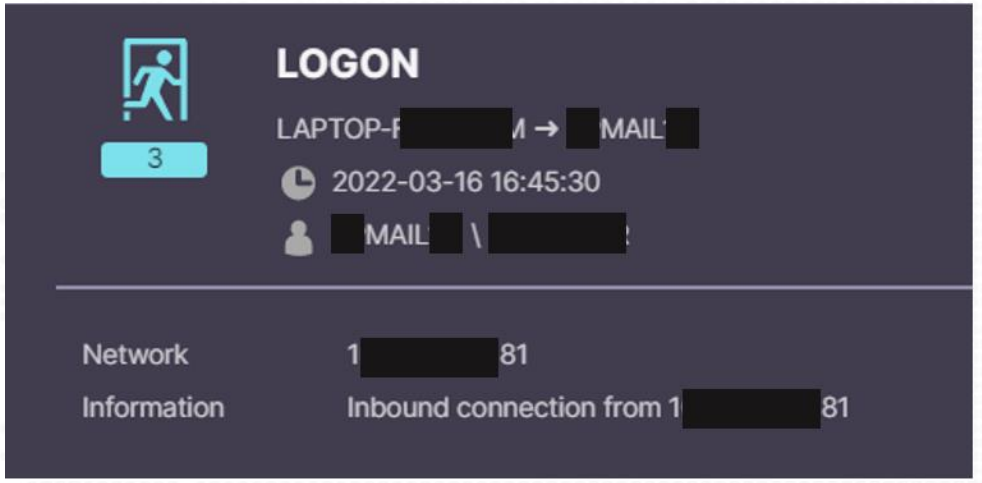
2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（2）正体不明の攻撃者によるアクセス

漏洩した情報を利用してログインした痕跡を確認。





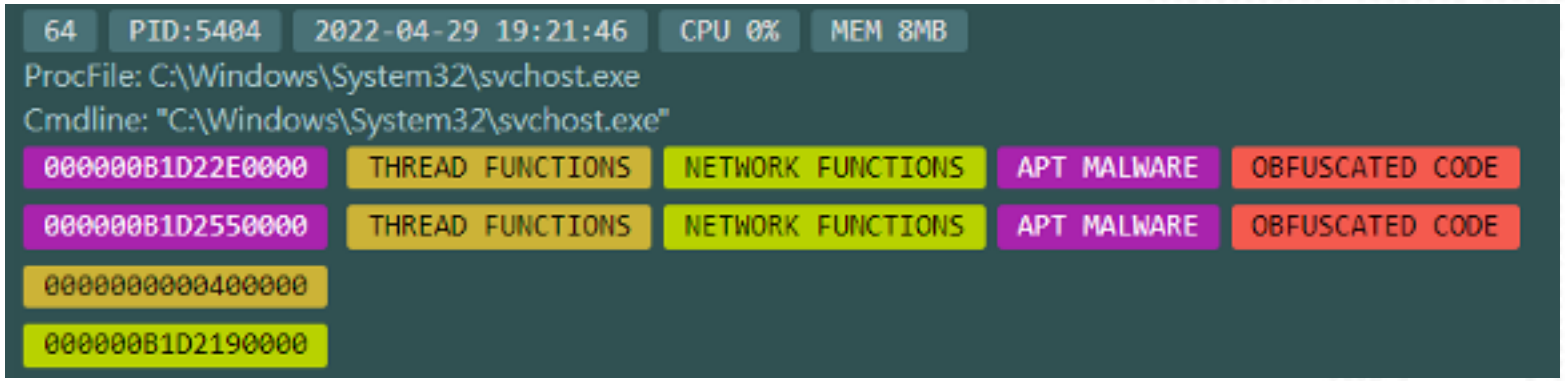
正しい認証情報を用いてログインに成功すると、**高脅威度としての検知は難しい！**

2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（3）攻撃のエスカレート

```
2022-04-13 20:01:30 [redacted]  c:\windows\temp\s3110.bat
2022-04-13 22:12:20             c:\windows\temp\prt3389.bat
2022-04-13 22:13:21             10.[redacted] >> C:\Windows\Temp\prt3389.bat
2022-04-13 22:13:23             10.[redacted] >> C:\Windows\Temp\prt3389.bat
2022-04-13 22:13:23             10.[redacted] >> C:\Windows\Temp\KB2604121_20130219_101408765.html
2022-04-13 22:13:24             10.[redacted] >> C:\Windows\Temp\KB2656351_20130219_101157363-Microsoft .NET Framework 4 Client Profile-MSP
```

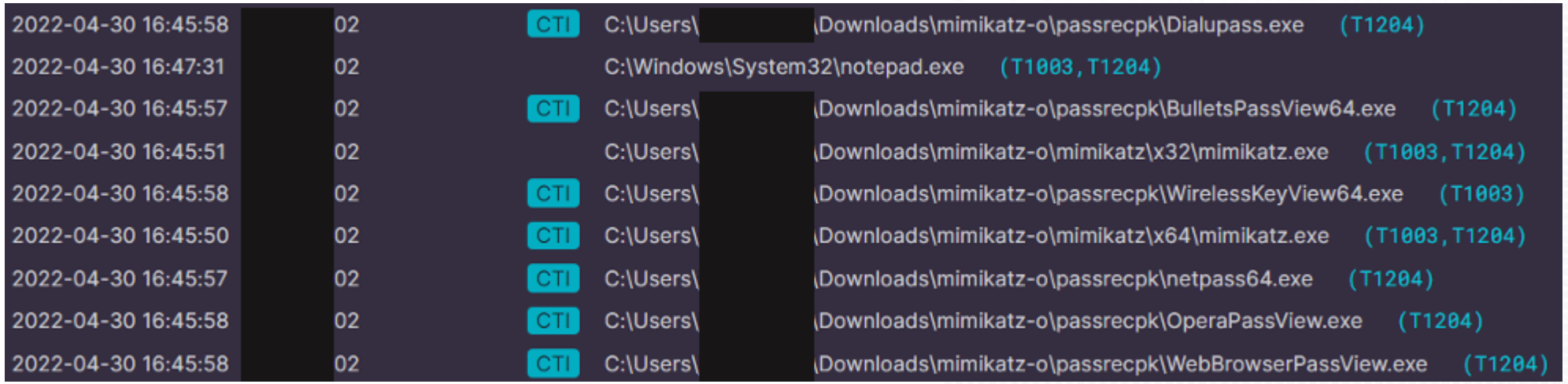


バッチファイルの展開および実行、svchost.exeへのコードインジェクションを確認。

2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（4）被害組織内での認証情報窃盗

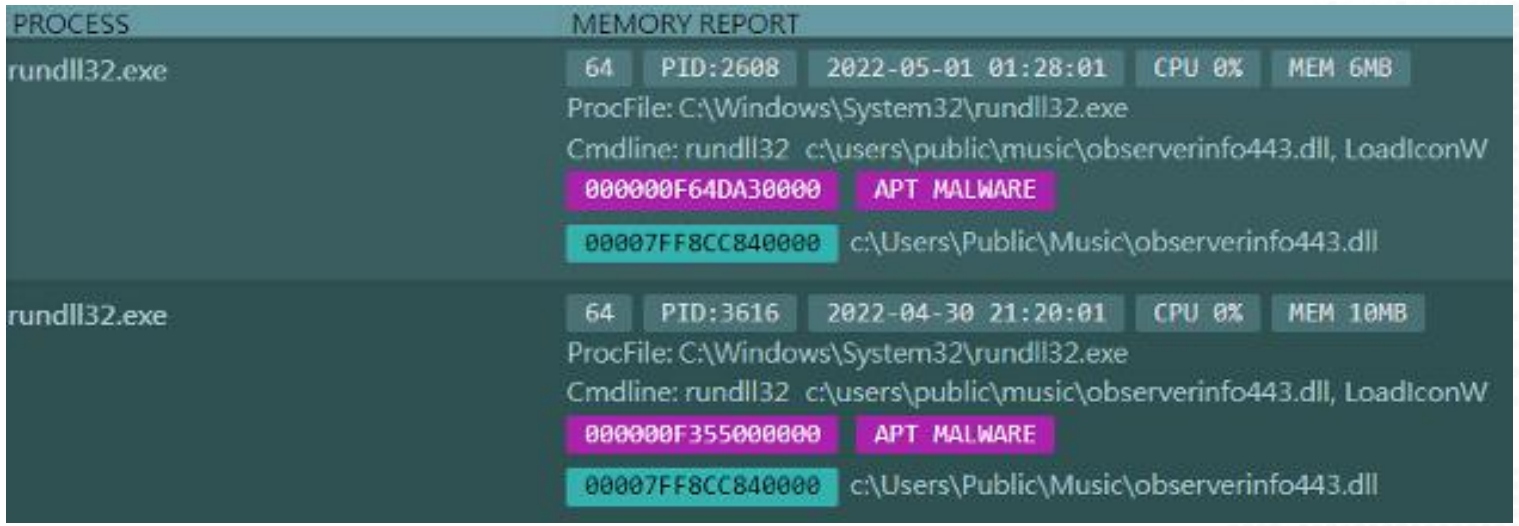


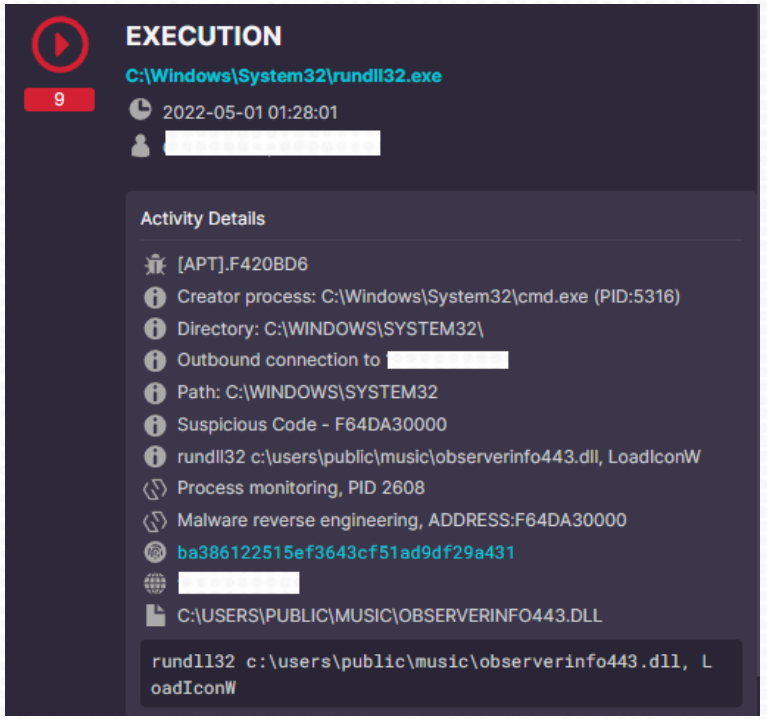
認証情報をダンプするmimikatzを展開、実行した痕跡を確認。

2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（5）リモート操作のためのマルウェアの実行





CobaltStrikeとみられるマルウェアの実行の痕跡を確認。

このケースでは、これを用いた情報窃盗までは確認できず。

2. Case 2:Lockbit 2.0 ランサムウェア

- 事件の顛末：

（6）Lockbitの実行

CobaltStrikeを利用してダウンロード、実行したとみられる。

解析したLockbit 2.0には、情報を収集して外部に送信する機能が見つからない？

ここまでの不正アクセスで既に情報を収集し盗んでいるか、「データを公開する」ということ自体がブラフの可能性。

2. Case 2:Lockbit 2.0 ランサムウェア

- 判明した攻撃の流れのまとめ：
  - Dark Web等に盗まれた認証情報が出回る
  - 盗まれた資格情報を使って不正アクセス
  - 組織内のシステムからの認証情報窃盗を実施
  - 組織内の端末からCobaltStrikeによるC&Cサーバへのアクセス
  - Lockbitをダウンロードし実行

Lockbitの実行は、攻撃の最後の段階で使われる傾向

2. Case 2:Lockbit 2.0 ランサムウェア

- おまけ：Lockbit 3.0 も入手したので解析してみた。

類似点：

- ファイルを暗号化し、その鍵を公開鍵？で暗号化している。
- 多重起動抑止をしている。
- 色々な解析回避技術を入れている。
- ファイルオープン処理、暗号化処理のスレッドを分けている。
- AD環境を利用した拡散処理がある。
- アイコンの設定や壁紙の変更、プリンタ出力を行う。
- 自身を削除する。

2. Case 2:Lockbit 2.0 ランサムウェア

- おまけ：Lockbit 3.0 も入手したので解析してみた。

相違点：

- ファイルの暗号化は、**xorを使った暗号方式に変更**。
- **先頭の4096バイトしか暗号化しないおサボりを止めた。**
- 多重起動防止も、使用するAPIを変更(NtCreateMutant → OpenMutexとCreateMutexの併用)。
- 解析回避技術は以前より執拗になり、特にデバッガ回避が増えていた（ブログに幾つか書いた）。

2. Case 2:Lockbit 2.0 ランサムウェア

- おまけ：Lockbit 3.0 も入手したので解析してみた。

相違点：

- 被害ファイルの拡張子を、固定の「.lockbit」からランダムなものに変更。これにより、ちょっと検知が面倒くさくなった。アイコンはこの拡張子に紐づけするよう変更。
- ファイルを開くスレッドと暗号化スレッド(複数起動)を分け、I/O Completion でスレッド間通信を行っているが、使用しているAPIが全く異なる。

2. Case 2:Lockbit 2.0 ランサムウェア

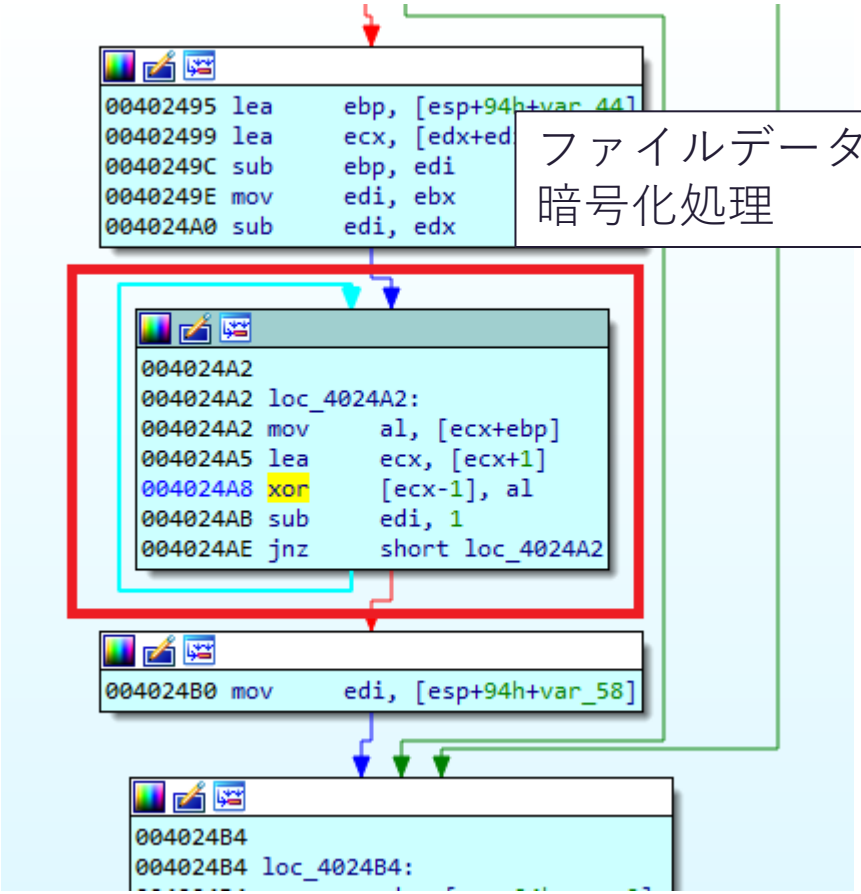
- おまけ：Lockbit 3.0 も入手したので解析してみた。

相違点：

- 拡張子の判定が**ベタ書きから32ビットハッシュによる判定に変更**。
- 無駄に？力が入っていた壁紙描画は、逆にえらくシンプルに。
- プリンタの出力枚数が**9999枚→1000枚に減少**。
- 自身を削除する方法は、削除プログラムを解凍、プロセス実行。

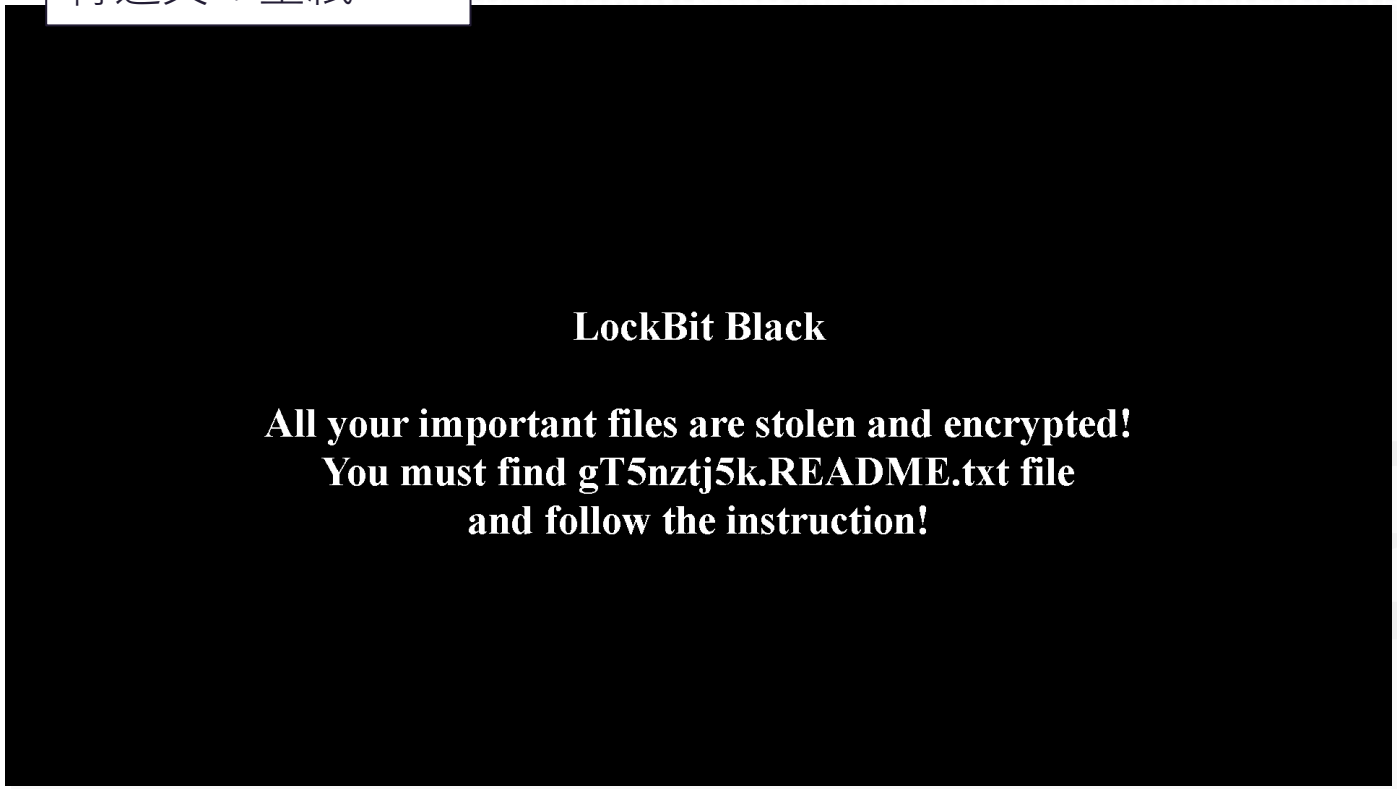
2. Case 2:Lockbit 2.0 ランサムウェア

- おまけ：Lockbit 3.0 も入手したので解析してみた。



ファイルデータの暗号化処理

脅迫文の壁紙

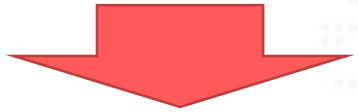


2. Case 2:Lockbit 2.0 ランサムウェア

- おまけ：Lockbit 3.0 も入手したので解析してみた。

感　想：

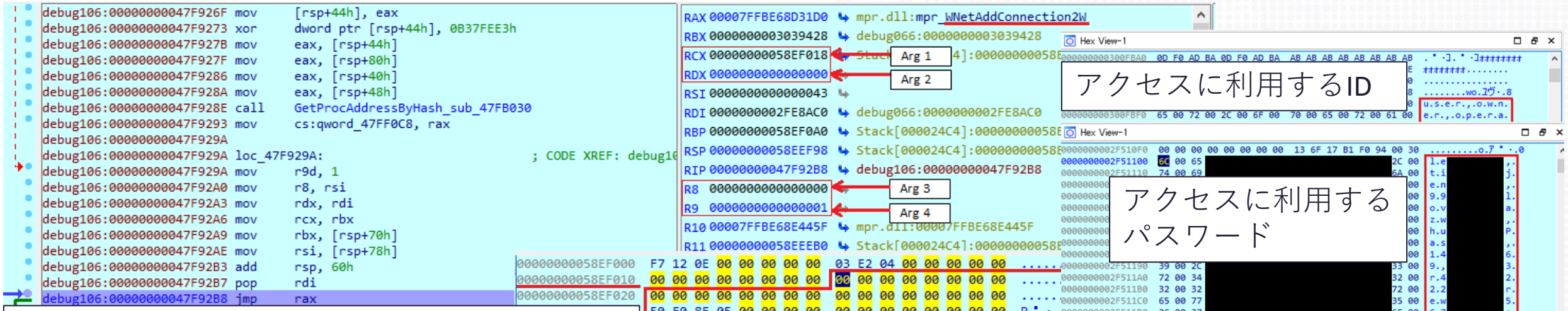
- 殆ど流用したコードが無く、**１から作り直したレベル**。
  →以前よりもコードのレベルは高くなっていた。



- ハッキリ言って、**要求仕様を変えないまま別人が作ったレベル**。
  →研究しているマルウェアの筆跡鑑定やインテリジェンス抽出の対象にすると面白そう！（**喜ぶんじゃぁない・・・**）

3. 台湾でもEmotet事例あり

- Emotetは、C&Cから落ちてくる追加コードが重要。
- 以前と異なり、ファイルに出力せず実行するようになっている。
- 落ちてきた機能例：WNetOpenEnumWを利用して感染拡大する機能。



Arg1: 0x058EF018 - NETRESOURCE structure
Arg2: 0x00000000 - Password
Arg3: 0x00000000 - User name
Arg4: 0x00000001 - Flag (CONNECT_UPDATE_PROFILE)

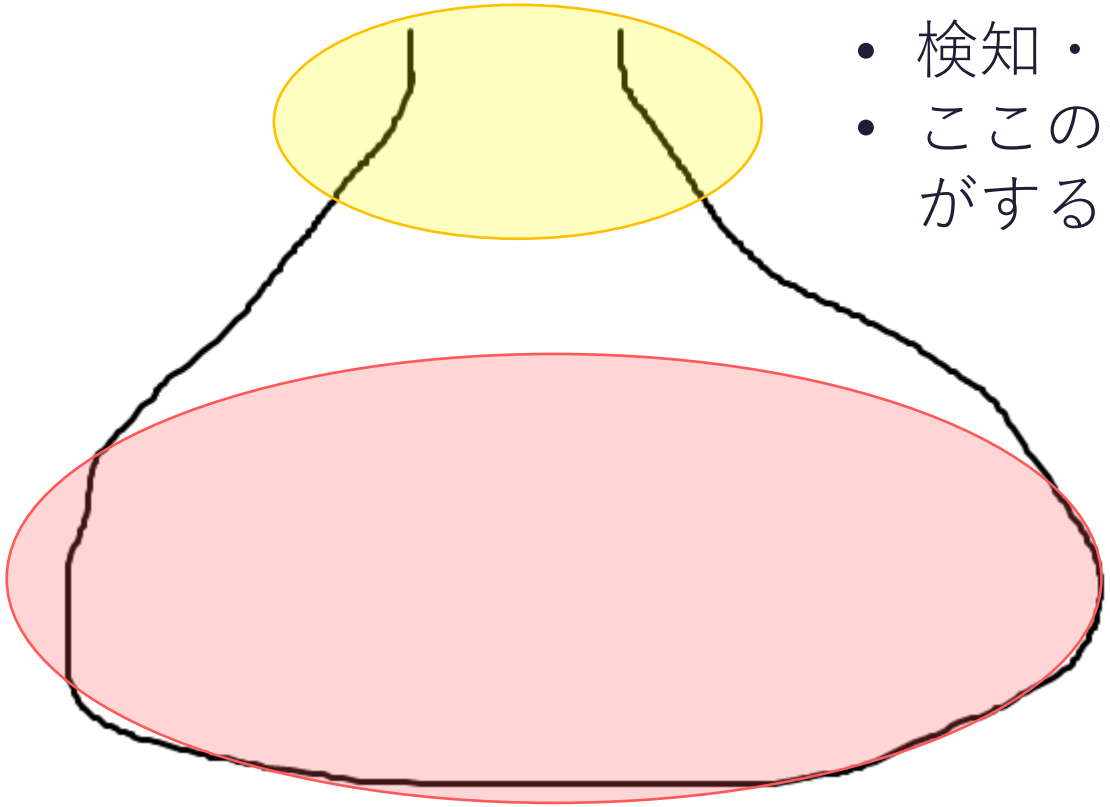
4. まとめ

- ２つのランサムウェアとEmotetの解析結果の紹介
- ランサムウェアの感染に繋がる認証情報の漏洩を発端とした不正アクセスの事例を紹介

- マルウェアの解析では、IoCだけでなく、どのようなことができるか、またはできないかといった能力の分析も、サイバー攻撃の分析では重要。
- 攻撃の手口やマルウェアの機能を知ることで、検知やブロックできる点を発見し、対策に生かすことができる。
- 不足している対策が何かについて気づいてもらいたい。

4. まとめ

- マルウェア解析のイメージ？



- 検知・識別に使うならこのあたりの情報で十分。
- ここの省力化・合理化の研究が多く、進んでる気がする？

- 時間がめっちゃかかる。
- 古手のアセンブラプログラマ世代が昔取った杵柄でやってるケースが多め？
- ここで得られる情報の活用がまだ不十分？

CyCraft |Website

CyCraft |Medium

CyCraft | Twitter