SecHack365

デモ動画

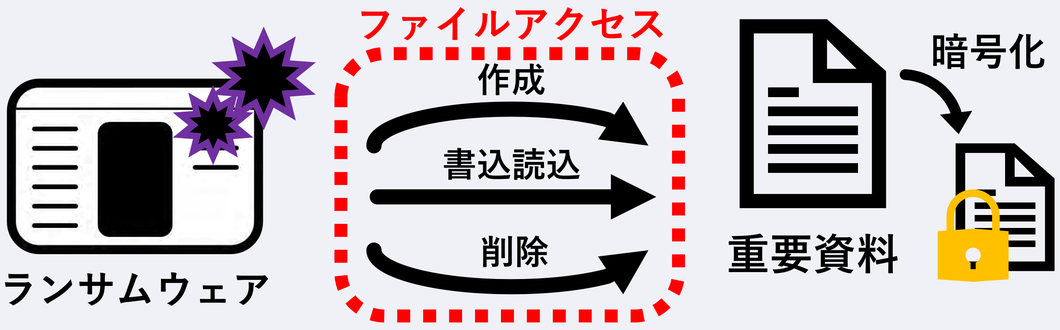
軽量化を図ったランサムウェア特化型ディフェンダー

ランサムウェアワクチン

38R 研究駆動コース
守田向輝

ランサムウェアワクチン

管理者権限のいらない軽量なディフェンダーを作る

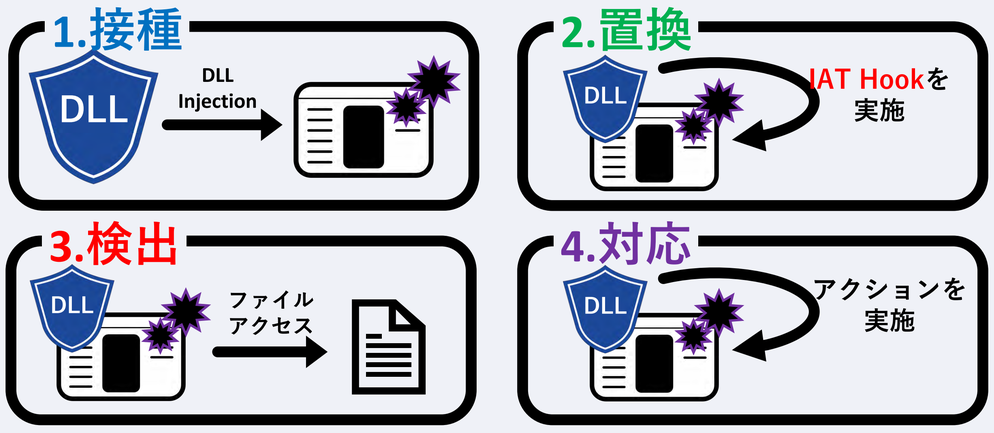


- ランサムウェアは多くの場合**ファイルアクセス**を行う
- そのアクセスを簡易的で低い権限で行えるプログラムで**ブロック**

➢ **省リソースで低い権限でランサムウェアの攻撃をブロック**

- ブロックする手法として**IAT Hook**, **DLL Injection**を用いる

仕組み



1.接種

- ランサムウェアワクチンがプログラム実行の前に**DLL Injection**を実施
  - ➢ Dll Main関数が**2.置換**の関数を実行プロセス内で呼び出し

2.置換

- ファイルアクセス関数を**IAT Hook**する
  - ➢ Hookによりファイルアクセスは行われず自作関数が呼ばれる
  - ➢ この時点でアクセス関数が呼ばれないのでアクセスが行われない

3.検出

- IAT Hookにより各関数に対応した自作関数が呼ばれる
  - ➢ 呼び出された関数と引数を取得、検出

4.対応

- ユーザーの設定と3.検出で取得した情報を元にアクションを実行
  - ➢ 元の関数呼び出しの制御, 対象ファイルや操作内容の通知etc.

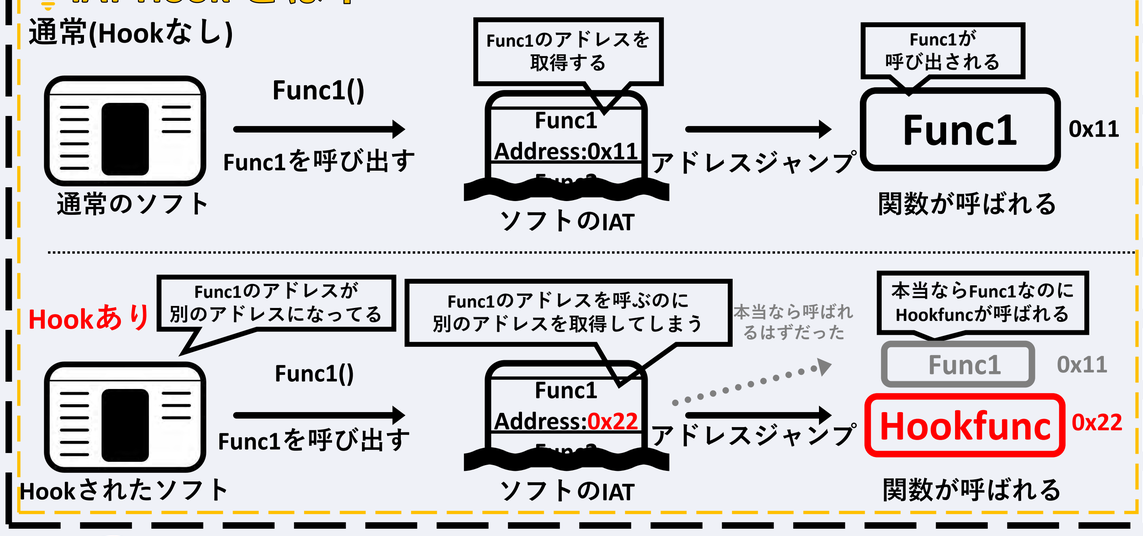
通知例

元ファイルを読み込み、
暗号化ファイルに暗号化したものを書き込み、元ファイルを削除していることが分かる。

```
CreateFile for   ../Documents/ID_PASSWORD_FILE.xlsx.Locked
ReadFile for   ../Documents/ID_PASSWORD_FILE.xlsx
WriteFile for   ../Documents/ID_PASSWORD_FILE.xlsx.Locked
DeleteFile for   ../Documents/ID_PASSWORD_FILE.xlsx
```

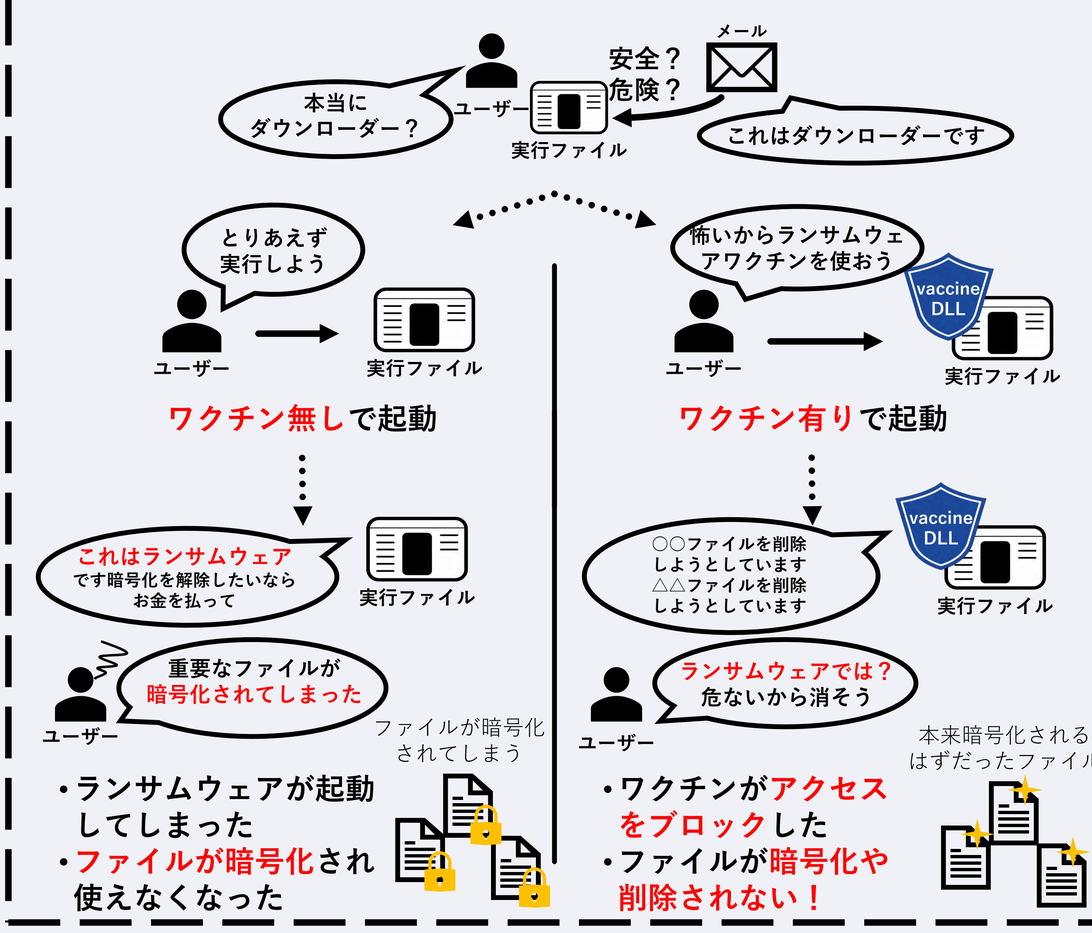
DLL Injection とは？

IAT Hook とは？



ユースケース

- ランサムウェアをランサムウェアワクチン無/有で実行
  - サンプルファイルの**被害状況を比較**



実装と評価

【実装】

- C言語, コード合計約1500行
- Virtual Box, CPU: Intel Core i7-12700Hのプロセッサー6個, Memory:8192MB

【防御率の評価】

- ファミリーの違うランサムウェアを実行
- **防御率の計測**,ファミリーの記録

➢ ブロック**成功:3/7** (7件中3件)

➢ ブロック失敗したものの影響 接種成功:**2/4**件

【省リソース化の評価】

- Windows Defender, ランサムウェアワクチンで各々ブロック
- 防御率の**最高値のCPU, メモリ使用率を比較**

➢ ランサムウェアワクチンの方がWindows Defenderより平均約**CPU50%, Memory337.8MB軽く**防御することが出来た

ランサムウェア＼ディフェンダー	ランサムウェアワクチン	Windows Defender
Akira	CPU:**11.4%**, Memory:**1.6MB**	CPU:**86.2%**, Memory:**319.8MB**
kawa	CPU: **0%**, Memory:**1.3MB**	CPU:**42.0%**, Memory:**323.7MB**
Nitrogen	CPU: **1.9%**, Memory:**1.3MB**	CPU:**35.2%**, Memory:**374.1MB**

※ランサムウェアワクチンのリソースはランサムウェアのリソースに依存する
※全てのファイルアクセスをブロックするオプションで実行
※ランサムウェアが検閲or親プロセスが削除されるまでの時間の最高値

社会実装

【防御できない技術・手法】

- shell codeやsyscall
- ターミナルコマンドを用いた攻撃
- レジストリの改変
- メモリ書き込み
- 既存プロセスへのInjection

↓解決策

- ターミナル書き込み関数のHook
- レジストリ変更の関数をHook
- メモリ書き込みの関数のHook
- 既存プロセスへのアタッチ関数のHook

【新たな活用法】

- **モニター,デバッガーの+α**
  - ➢ 関数呼び出しのモニター
  - ➢ ブレイクポイント設置機能の追加
  - ➢ 高レイヤーなモニター
- **仮想環境の+α**
  - ➢ Hookする関数の指定
    - ➢ 用途ごとに関数をまとめる
- **通知・ログの方式を追加**

防御範囲が狭い, 漏れがある可能性がある
→ないよりはあるほうがいいもの

- 既存のモニターやデバッガーは情報量が多く低レイヤーで読みづらい
  - ➢ 高レイヤーで具体的な通知を出す**+αモニター**としての活用
- 非常に省リソース
  - ➢ 汎用性が高く、**アンチウイルス以外の技術**にも使用可能

今後の進め方

- アンチウイルスソフトとしての機能以外の改善, **論文作成/学会発表**

NICT National Institute of Information and Communications Technology