FileInsight-plugins: Decoding toolbox for malware analysis

萬谷 暢崇

Bluebox @ CODE BLUE 2019

自己紹介

- 政府職員
  - 警察庁情報技術解析課サイバーテロ対策技術室（サイバーフォースセンター）専門官
  - 過去にマルウェア解析やデジタルフォレンジックに従事
- 趣味のプログラマ
  - 2001年から FreeBSD プロジェクトメンバー (ports committer)
  - オープンソースソフトウェアが大好き！ 🥰

FileInsight-plugins

- McAfee FileInsight バイナリエディタのプラグイン集
  - 2019年10月時点で67個のプラグイン
- マルウェア解析における様々なデコード作業に便利
- 2012年に開発を開始
- **個人プロジェクトで自宅で開発**
  （日本国政府のものではありません） 😉
- 入手はこちら https://github.com/nmantani/FileInsight-plugins

背景

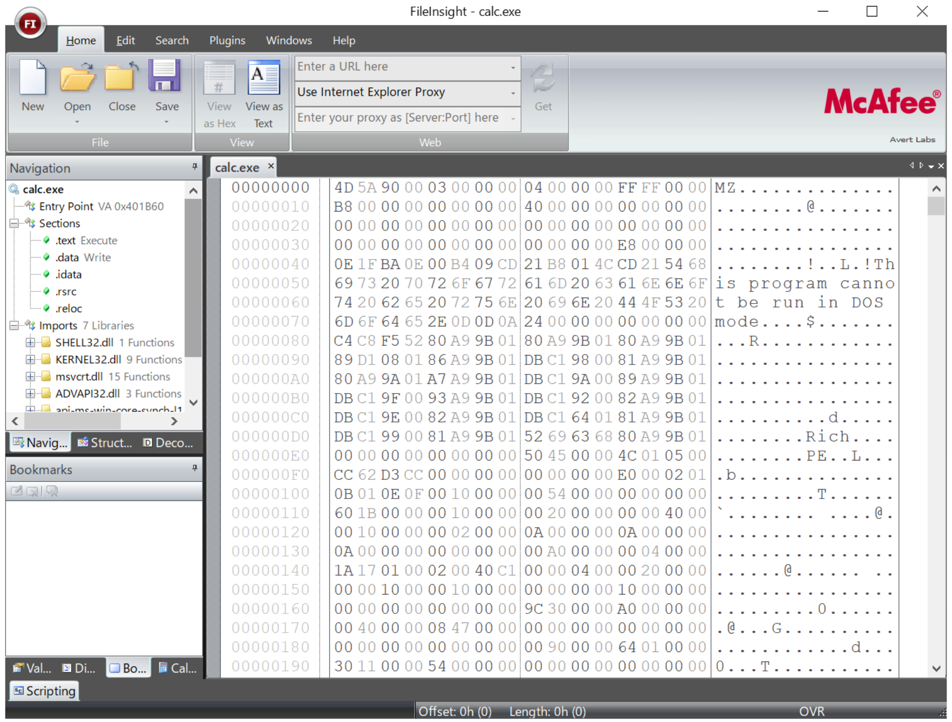
- 攻撃者は解析を妨げたり意図を隠すためにマルウェア中の様々なデータを難読化する
  - 埋め込まれた実行ファイルやおとり文書ファイル
  - 設定情報（C2ホスト名やポート番号）
  - 重要な文字列（ファイルパス、レジストリパスやコマンド）
  - コード
- マルウェア解析者はさらなる解析のために可読化する必要がある

開発の動機

- 当初（2012年）
  - Microsoft Office（と Adobe Flash Player）や Adobe Reader の脆弱性がマルウェア感染によく悪用されていた
  - 解析環境のソフトウェアのバージョンが exploit コードのターゲットのバージョンと異なっていて動的解析できないことが時々あった
  - そのため、C2 ホスト名をいち早く知って通信をブロックするために難読化されて埋め込まれたマルウェアの実行ファイルを手作業で抽出したい
- 現在
  - ただただ**無料でパワフルなバイナリエディタ**が欲しい！！ 😈

FileInsight

- マカフィー社が開発したフリーのバイナリエディタ
- 便利な組み込み機能
    - デコーダ（XOR, ビットローテート, BASE64等）
    - x86 逆アセンブラ
    - ブックマーク
    - データ構造ビューア（HTML, OLE と PE）
    - Python / JavaScript でのスクリプティング
- **Python プラグインで拡張可能！**



FileInsight

- しかし・・・2009年以降アップデートされていません
- **McAfee Free Tools ウェブサイトからも消えました** 😵
- FileInsight のインストーラはこちらで入手可能です
  http://downloadcenter.mcafee.com/products/mcafee-avert/fileinsight.zip

FileInsight-plugins の詳細

プラグインの分類

- 67 のプラグインが8つのカテゴリに分類されている（2019年10月現在）
  - Basic operations
  - Compression operations
  - Crypto operations
  - Encoding operations
  - Misc operations
  - Parsing operations
  - Search operations
  - XOR operations

アルゴリズムとフォーマットの追加サポート

圧縮

- aPLib
- Bzip2
- Deflate (without zlib header)
- Gzip
- LZMA
- LZNT1
- XZ

暗号

- AES
- ARC2
- ARC4
- Blowfish
- ChaCha20
- DES
- Salsa20
- Triple DES

エンコード

- Binary data <-> Hex text
- Binary data <-> Binary text
- Custom BASE64
- ROT13 (variable amount)
- Quoted printable

暗号利用モード

- ECB
- CBC
- CFB
- OFB

追加された機能

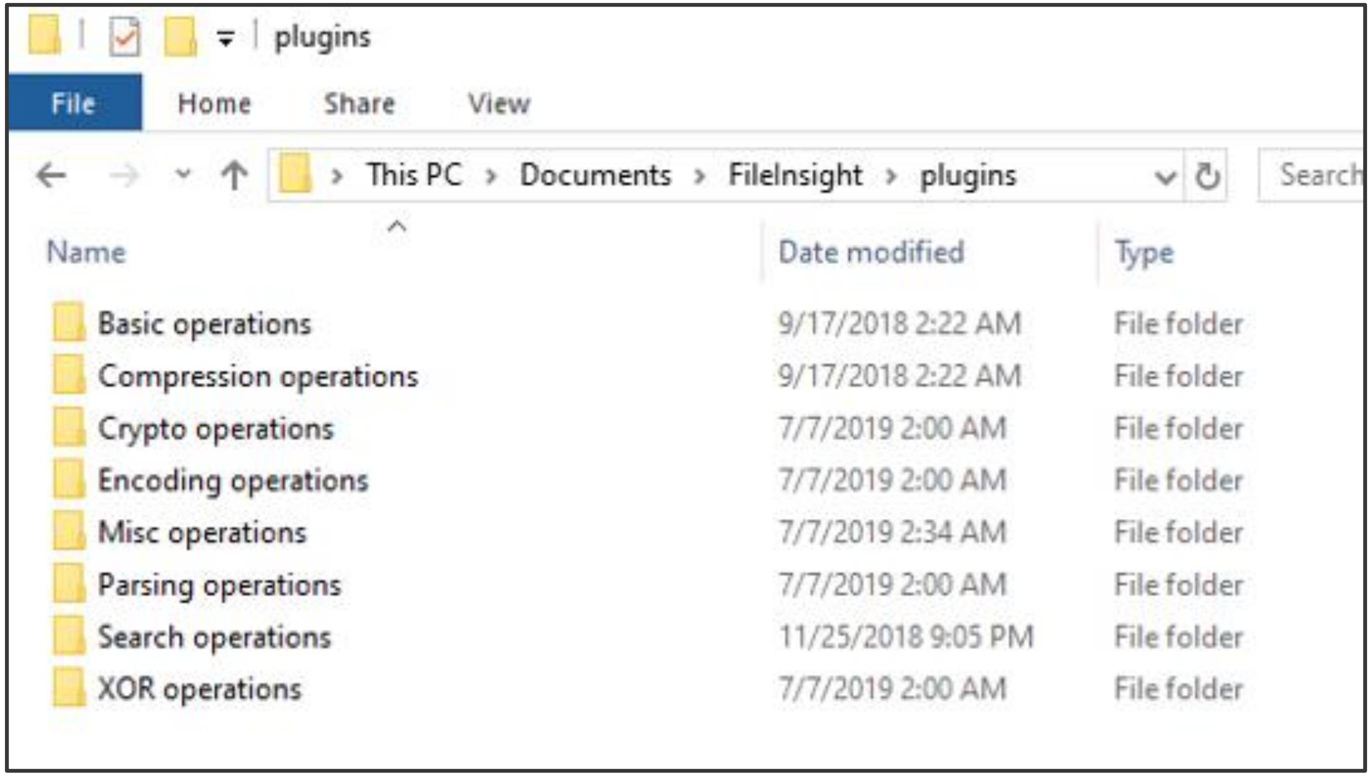
- 選択したデータを新しいタブで開く
- ハッシュ値の計算
- ファイルタイプの判別
- 埋め込まれたファイルの発見
- 正規表現を使った検索 / 置換
- XOR / ビットローテートされたデータの検索
- 0x00 をスキップする XOR (Null-preserving XOR)
- 256 バイトの XOR キー推定（キーをインクリメント / デクリメントする XOR を使われたデータに有効）
- データを外部ツールで開く（JSON ファイルでカスタマイズ可能）
- ファイル比較
- YARAルールでのスキャン
- その他いろいろ！

必要なもの

- Python 2 (x86)
  - **FileInsight は Python 2 (x64) と Python 3 と互換性がありません**
- Python モジュール等
  - aPLib
  - binwalk
  - PyCryptodomex
  - backports.lzma
  - python-magic
  - pefile
  - yara-python

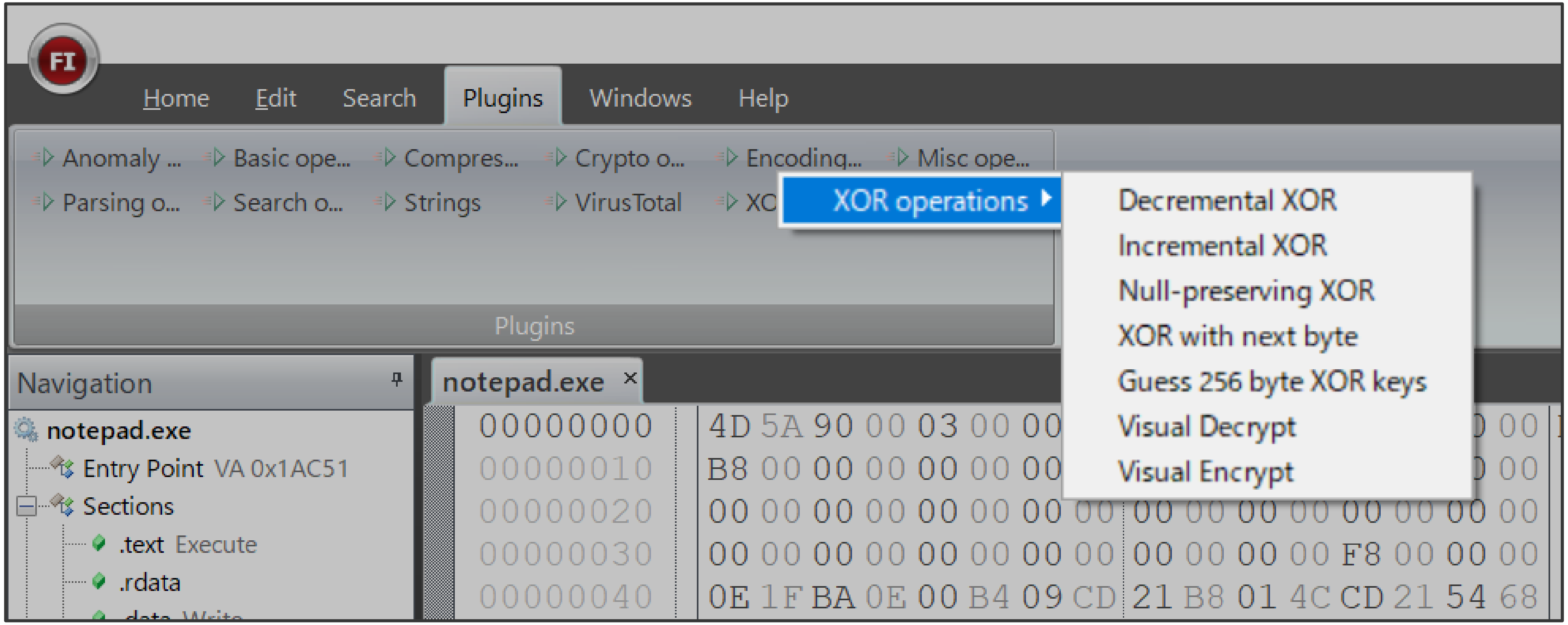
インストール

- “plugins” フォルダを %USERPROFILE%\Documents\FileInsight にコピーします



使い方

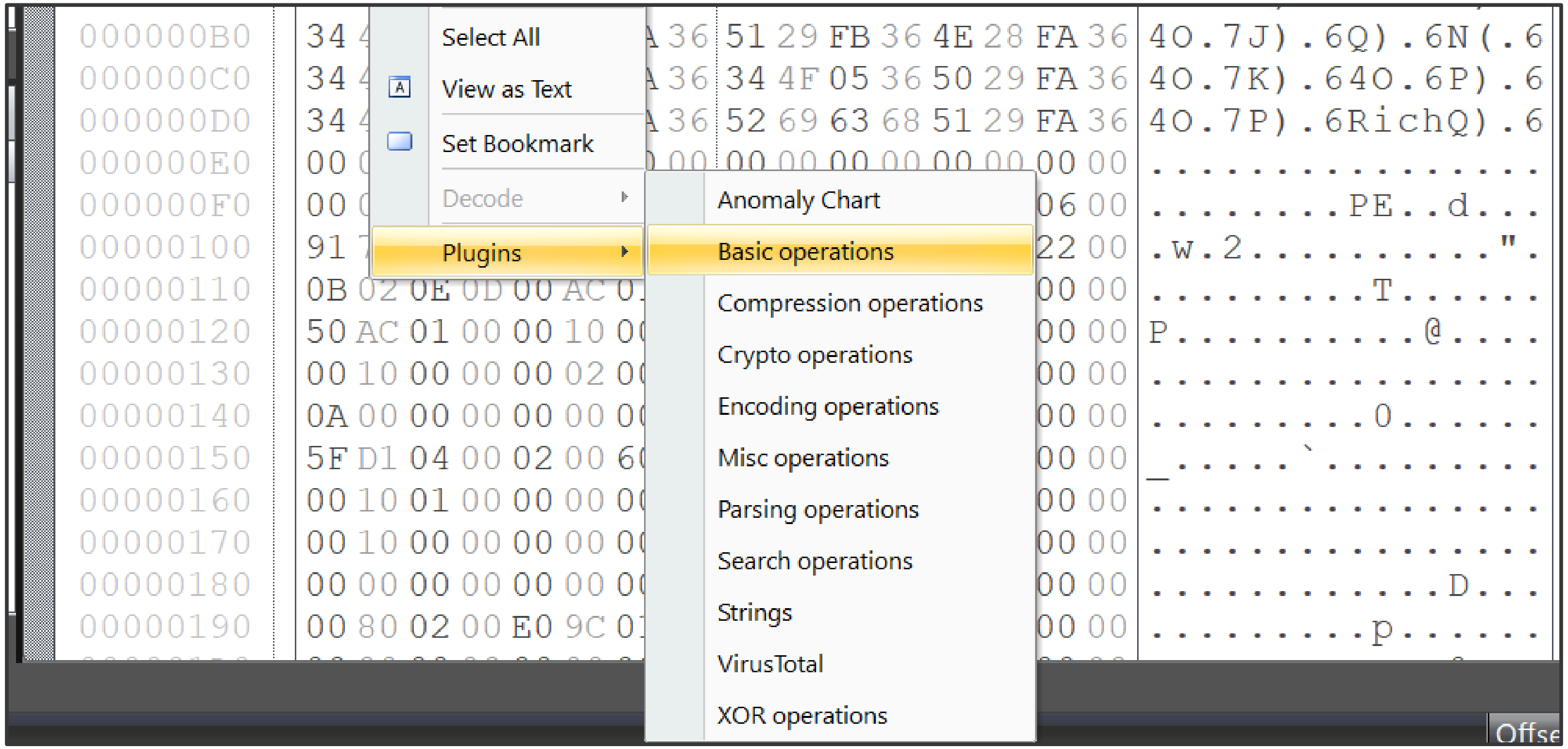
- “Plugins” タブからカテゴリをクリックしてプラグインを選択します



* Anomaly Chart, Strings と VirusTotal のプラグインは FileInsight に同梱

使い方

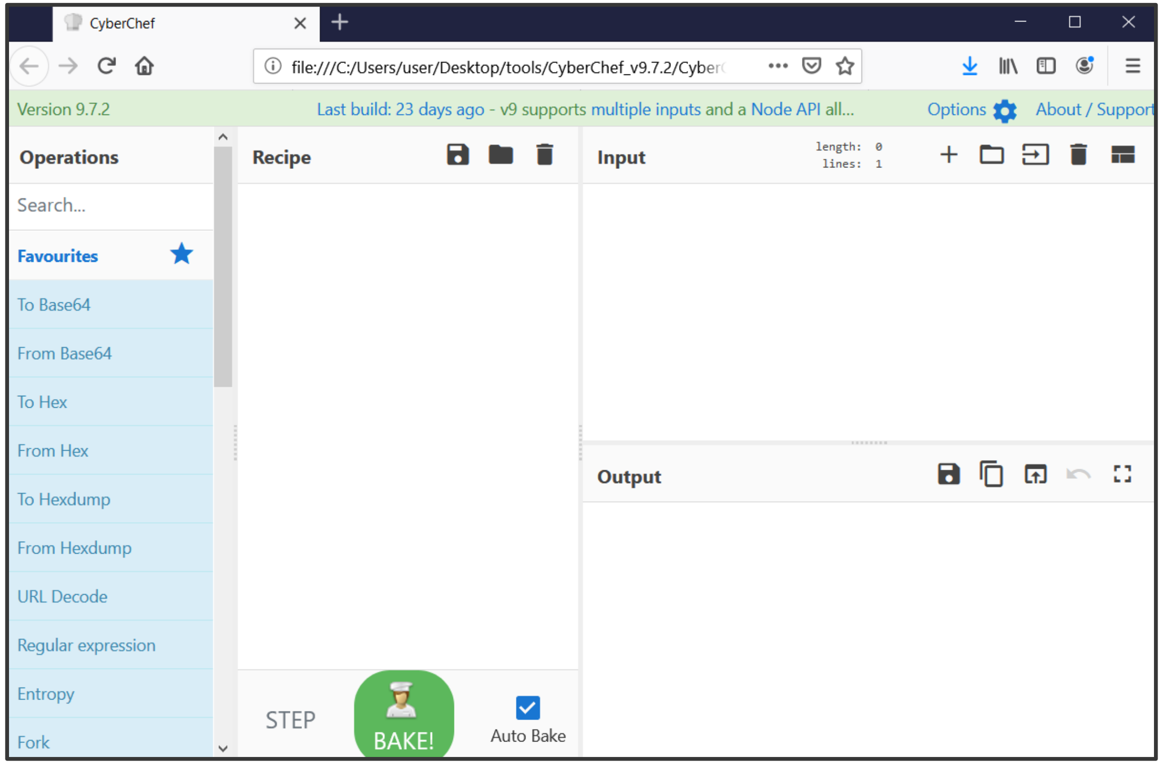
- 又は右クリックのメニューからカテゴリを選択します



* Anomaly Chart, Strings と VirusTotal のプラグインは FileInsight に同梱

FileInsight-plugins のお友達: CyberChef

- **超パワフルなデコードツール** 😍
- ウェブブラウザ上で動作
- 英国政府（GCHQ）と世界中の開発者によってアクティブに開発
- 手作業でのファイル編集は苦手
- **“Send to” プラグインで CyberChef に直接データを送り込める（12KBまで）**



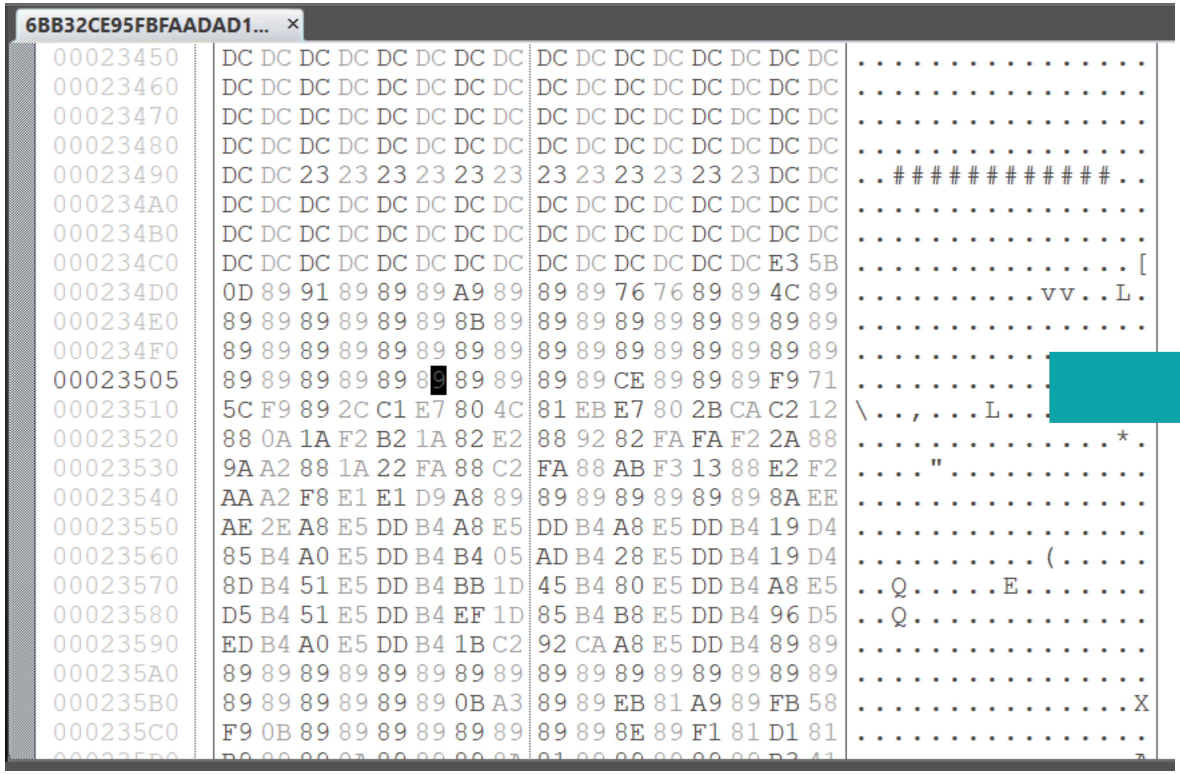
デモ

デモ 1

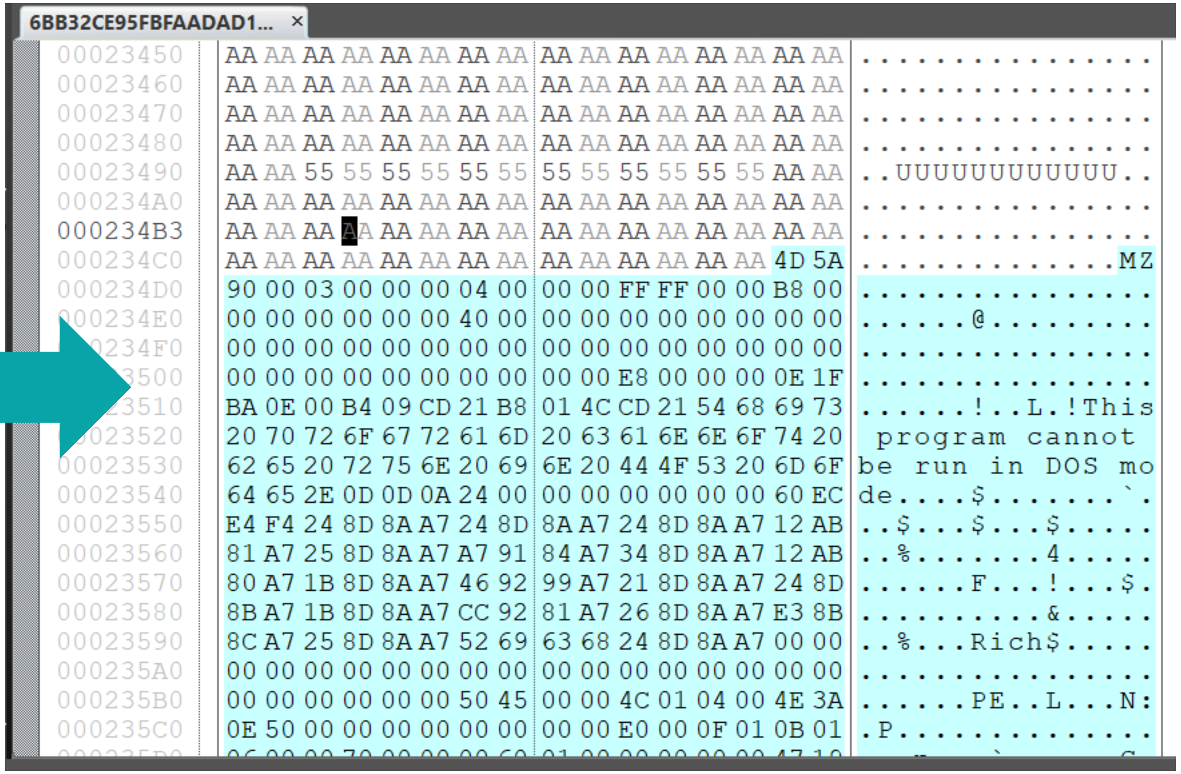
- Microsoft Excel ファイル
    - マルウェアの実行ファイルが XOR とビットローテートで難読化されて埋め込まれている
- 使用するプラグイン
    - Search operations -> XOR text search
    - Paring operations -> Find PE file
    - Misc operations -> Send to

デモ 1

可読化前：



可読化後：

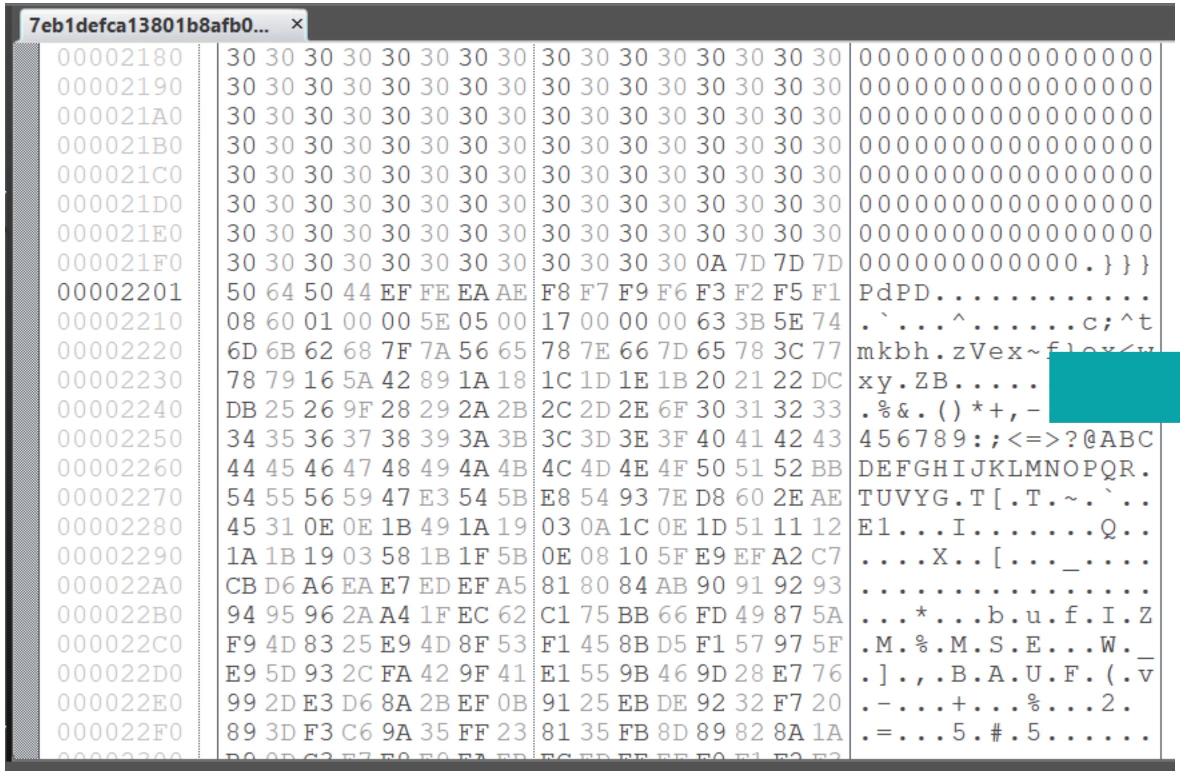


デモ 2

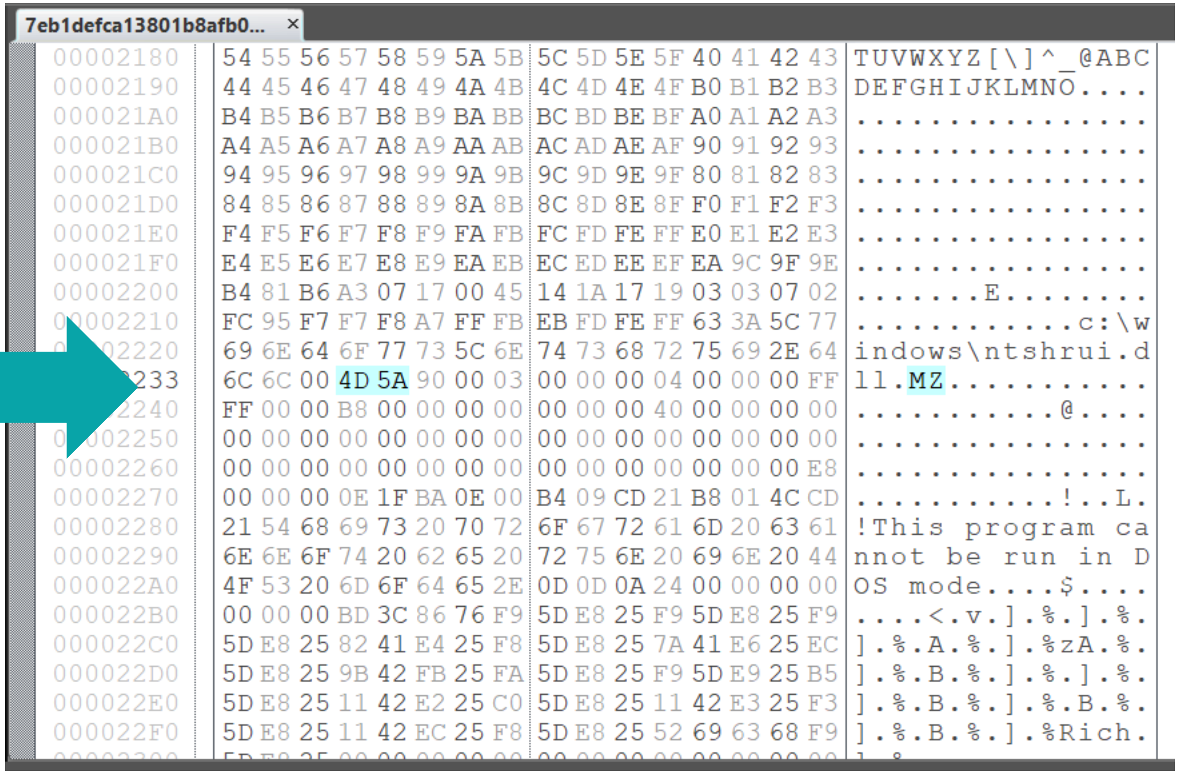
- リッチテキストファイル
  - マルウェアの実行ファイルがキーをインクリメントする XOR（rolling XOR）で難読化されて埋め込まれている
- 使用するプラグイン
  - XOR operations -> Guess 256 byte XOR keys

デモ 2

可読化前：



可読化後：



デモ 3

- PHP ウェブシェル
    - コードは BASE64, ROT13, Deflate 等で難読化されている
- 使用するプラグイン：
    - Encoding operations -> ROT13
    - Basic operations -> Reverse order
    - Encoding operations -> Custom BASE 64 decode
    - Compression operations -> Raw inflate
    - Misc operations -> Send to

デモ 3

可読化前：

4e837338fb5a31b638f9... ×

```
<?php
eval(gzinflate(base64_decode(strrev(str_rot13('=8u///5557/ss//xBzBpXjJ5/nVhIuMLepXM09DIgR/ah/IuT36cltrk
?>
```

可読化後：

4e837338fb5a31b638f912... × New file* × New file* × New file* × New fi × New file* ×

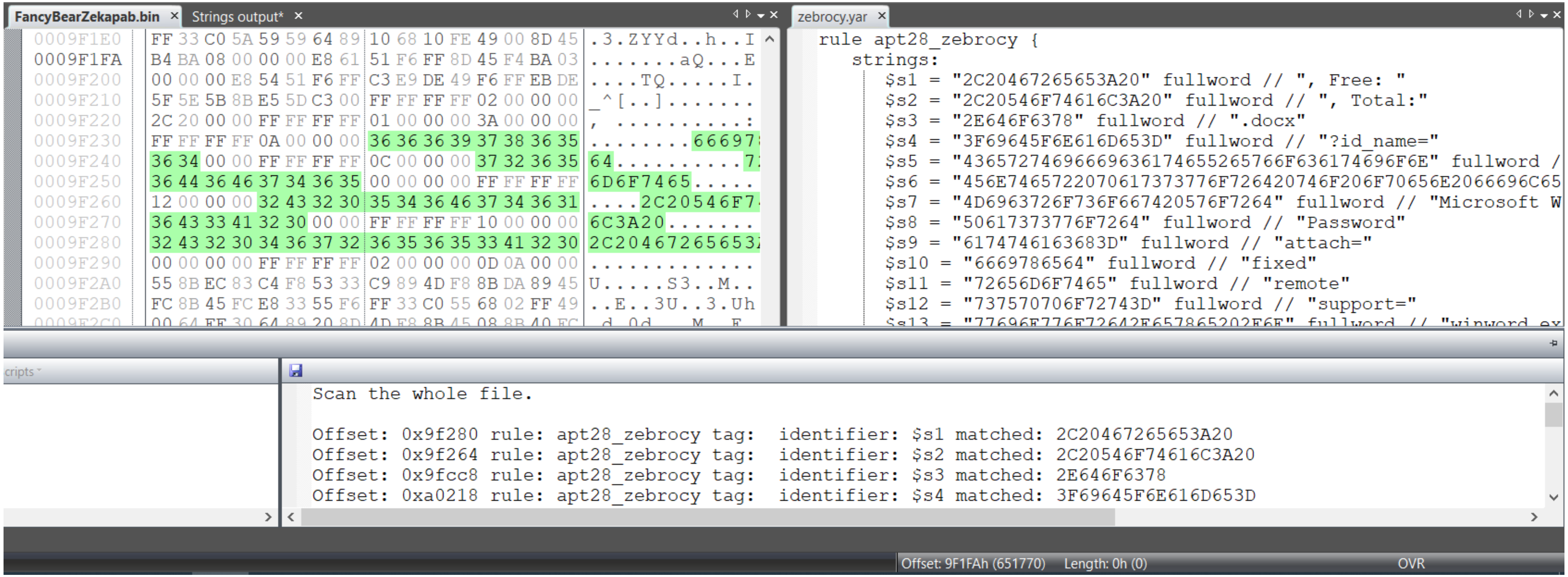
```
?><?php
/*******************************************/
/* robotcop xxixixixixixxxx@xxx        */
/* Re-coded and modified By robot */
/* #love@beautiful              */
/*******************************************/
$sh_id = "U0FQVV5KQUdBRCBTaGVsbHMgLXJAIA==";
$sh_ver = "No Fucking Rules";
$sh_name = base64_decode($sh_id).$sh_ver;
$sh_mainurl = "http://legalref.ru/config/";
```

デモ 4

- Zebrocy (Zekapab) マルウェア
  - いくつかの文字列が16進数にエンコードされている
- 16進文字列を抽出してデコードする
- この文字列に基づく YARA ルールでスキャンする
- 使用するプラグイン：
  - Parsing operations -> Strings
  - Search operations -> YARA scan

デモ 4

“YARA scan” プラグインで FileInsight を YARA ルールエディタとして使用可能



デモ 5

- 隠しメッセージを含むバイナリファイル（マルウェアではありません）
- CyberChef とのコンビネーション
- 使用するプラグイン：
  - Parsing operations -> File type
  - Compression operation -> XZ decompress
  - Misc operations -> Send to

デモ 5

可読化前：

可読化後：

Output

Thank you for listening my talk! Let's enjoy CODE BLUE 2019!

まとめ

- FileInsight-plugins は FileInsight バイナリエディタをアイアンマンスーツのようにもっとパワフルにします
- マルウェア解析における様々なデコード作業に便利
- 入手はこちら https://github.com/nmantani/FileInsight-plugins
- フィードバックをお待ちしています！（プルリクエスト、バグレポート、機能追加の要望）😉

ありがとうございました！

スライドは私の GitHub リポジトリからダウンロードできます：

https://github.com/nmantani/FileInsight-plugins

付録

FileInsight の API 関数の一覧

関数の詳細については FileInsight のヘルプをご覧ください。

- getLength()
- getByteAt()
- setByteAt()
- setBookmark()
- getSelection()
- getSelectionOffset()
- getSelectionLength()
- gotoBookmark()
- download()
- newDocument()
- showSimpleDialog()
- decode()

FileInsight の API 関数の一覧

関数の詳細については FileInsight のヘルプをご覧ください。

- getDocument()
- setDocument()
- getDocumentName()
- getDocumentCount()
- getDocumentURL()
- activateDocumentAt()

プラグインのサンプルコード

```
# 選択した部分又はファイル全体を新しいタブで開く
length = getSelectionLength() # 選択部分のサイズを取得
if length > 0:
    data = getSelection() # 選択部分のデータを取得
else:
    data = getDocument() # ファイル全体のデータを取得
newDocument(“New file”, 1) # hex view モードで新しいタブを開く
setDocument(data) # データを新しいタブにコピーする
```