

nmantani-Plugin訓練 座学

訓練概要

- 訓練目的

- 表層解析の技術を用いて、難読化された検体ファイルから目的のオブジェクト(実行可能ファイル, 偽装ドキュメント等)を抽出できる様にする。

- 受講対象

- マルウェアの解析作業(主に動的解析)を行う者
- マルウェア解析のサポート・支援作業を行う者

- 受講可能レベル

- マジックナンバー(フォーマット識別子)の意味を理解しており、その知識を活用して一般的なファイル種別(MS-Office, PDF, 実行ファイル等)を判別できるレベルの者
- または、相当する知識を有する者(「表層解析訓練」受講者含む)

- 事前知識

- 16進数の読み方
- (ビット)論理演算(AND／OR／NOT／XOR／シフト／ローテート)

FileInsight/nmantaniの紹介

<http://www.mcafee.com/au/downloads/free-tools/fileinsight.aspx>

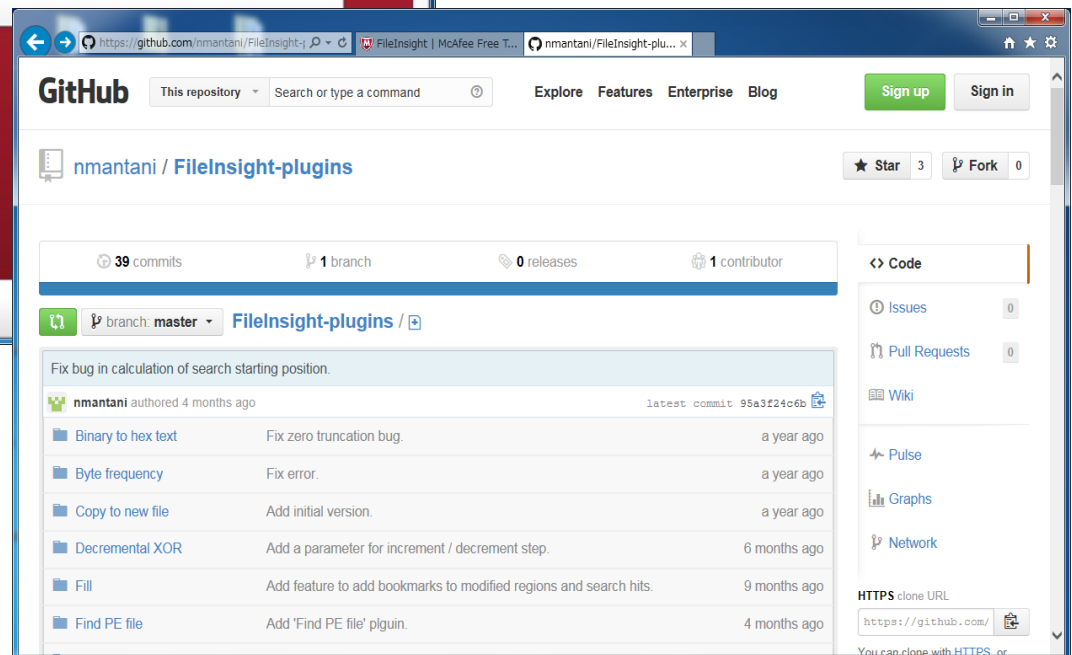


McAfee FileInsight

McAfee社が提供する無償のファイル操作ツール。マルウェア解析に有用な機能を備えている。

[nmantani/FileInsight-Plugins](https://github.com/nmantani/FileInsight-plugins)

Nobutaka Mantani氏が作成したFileInsightの機能を更に強化するプラグイン集



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

FileInsight/nmantaniによる オブジェクト抽出の可能性

• ドキュメント型マルウェアの一般的構造

- 実行ファイルをドロップするマルウェアの場合、何処かにその素がある
- 通常マルウェア作成者はシェルコードを極力小さく作りたがる
- 小さくするため、複雑な復号ロジック等の余計なコードは書きたくない

MS-Officeの場合



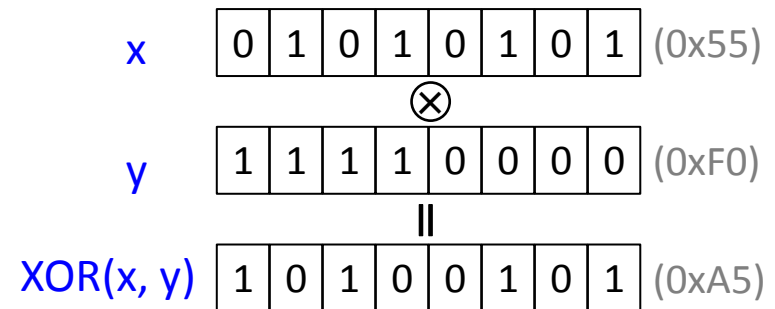
この様なタイプのマルウェアの場合、難読化と言っても高度な暗号やパックされることは少ないため**ビット演算**、**ビット操作**、**シフト演算**等を駆使して解くことが出来る可能性がある

難読化解除の基礎知識

• ビット演算

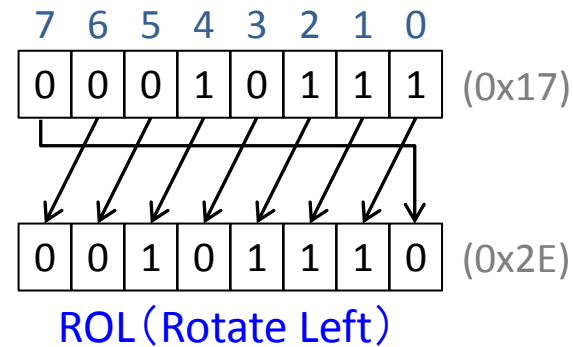
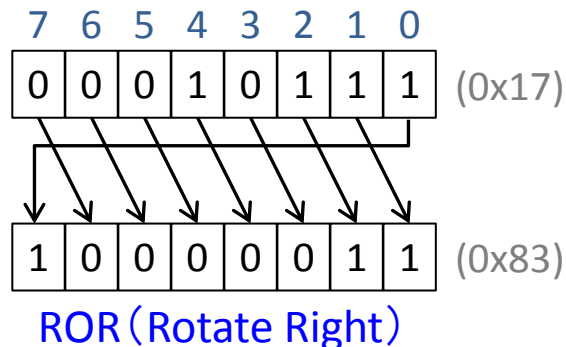
- XOR (Exclusive OR: 排他的論理和) が多用される (これだけは覚える)

x	y	AND(x, y)	OR(x, y)	XOR(x, y)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



• シフト演算

- ROR (Rotate Right) と ROL (Rotate Left) が多用される (これだけは覚える)



オブジェクト抽出の基礎知識

- マジックナンバー(ファイル識別子)

- 大抵ファイルの先頭数バイトを確認することでファイル種別が分かる

PEファイル (EXE)

PDFファイル

ZIPファイル or MSOffice

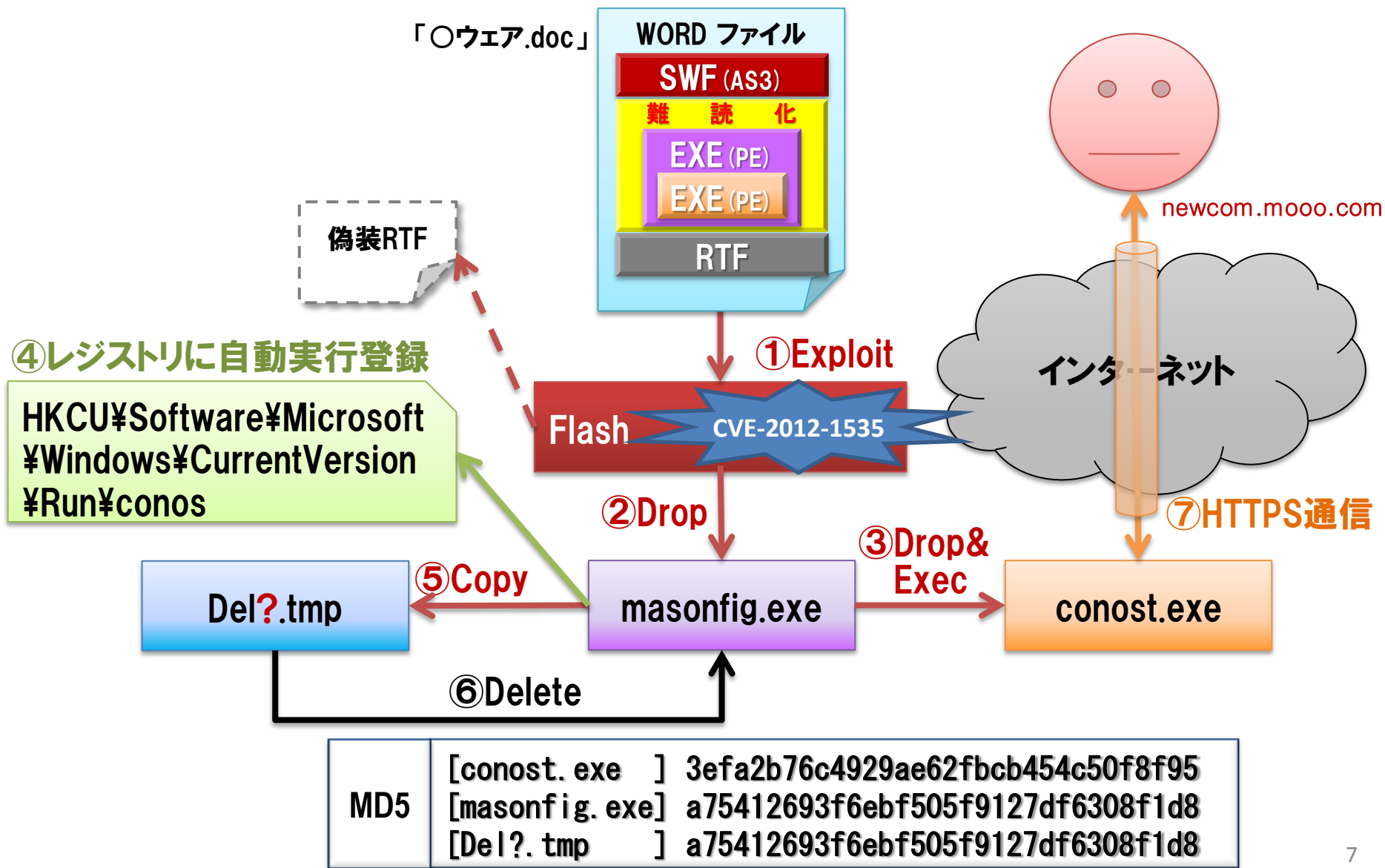
MSOffice or 一太郎

The screenshot shows a hex editor window with four files open. The files and their magic numbers are:

- calc.exe**: Magic number 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00. The text 'MZ...' is visible.
- ダイアログ.pdf**: Magic number 25 50 44 46 2D 31 2E 35 0D 25 E2 E3 CF D3 0D 0A. The text '%PDF-1.5.%' is visible.
- サンプル.docx**: Magic number 50 4B 03 04 14 00 06 00 08 00 00 00 21 00 70 73. The text 'PK...' is visible.
- アンケート.jtd**: Magic number D0 CF 11 E0 A1 B1 1A E1. The text 'ミマ.爆7...' is visible.

Red arrows point from the file type labels on the left to the corresponding file windows in the hex editor.

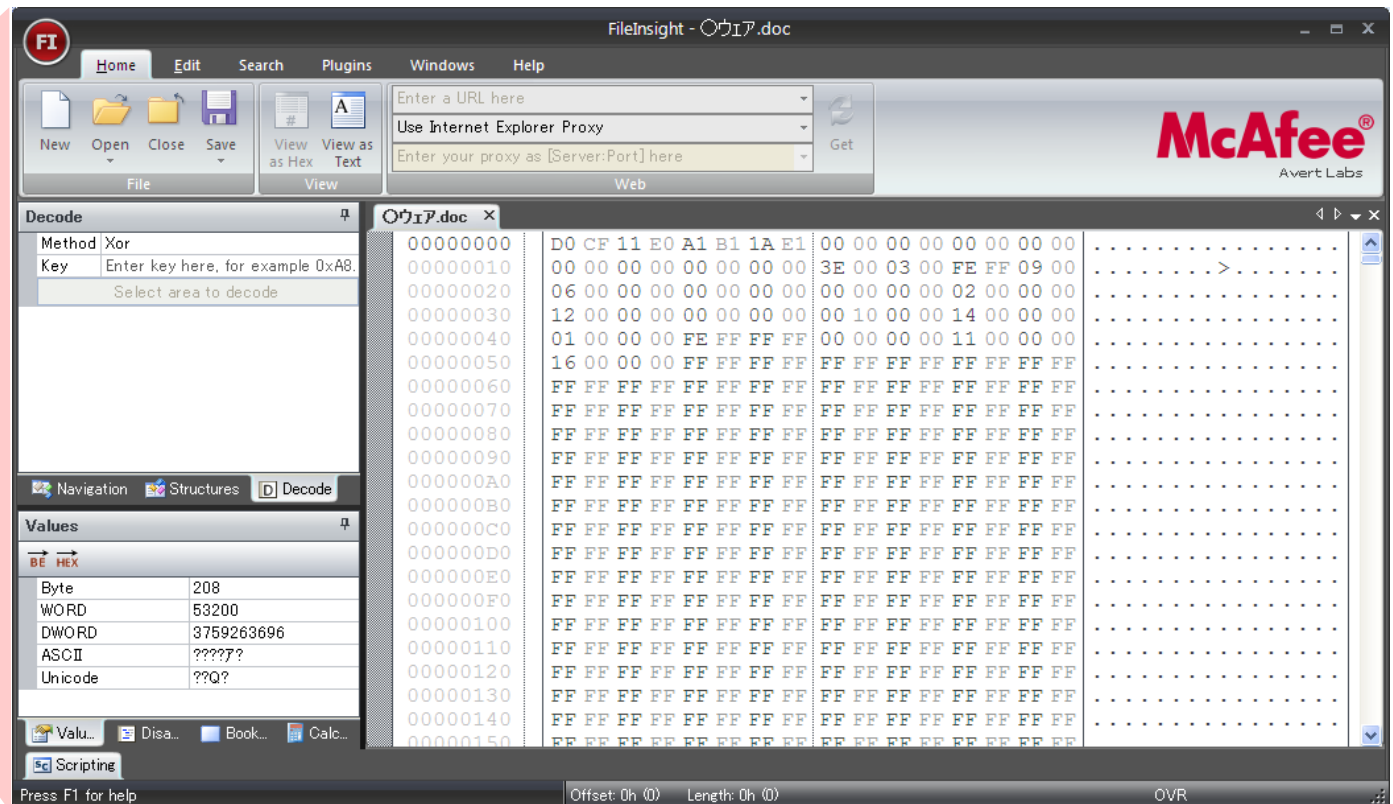
オブジェクト抽出を試みる検体



オブジェクト抽出

- 抽出作業の開始

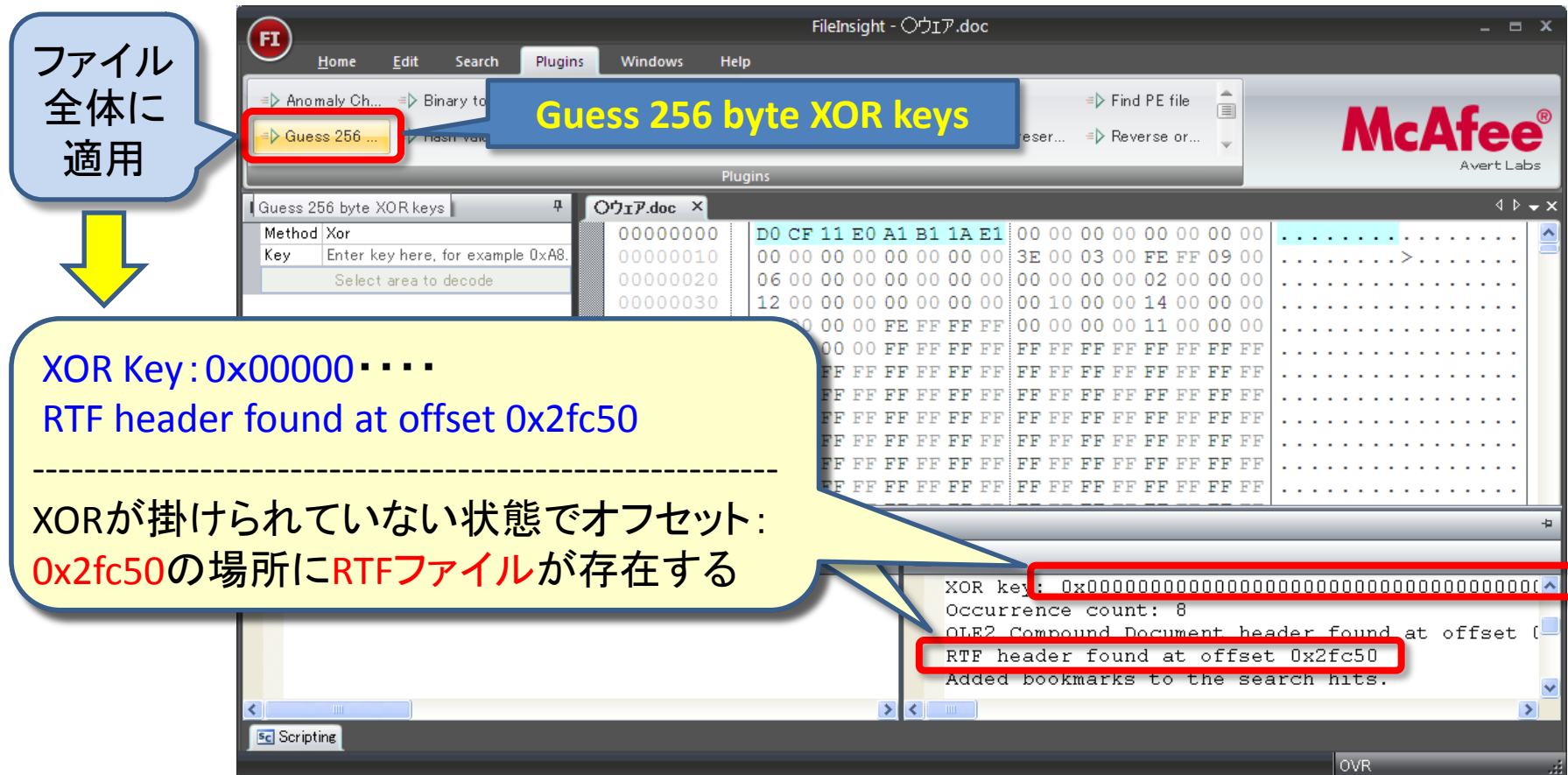
- FileInsightアイコンへマルウェア検体をドラッグ & ドロップ」



オブジェクト抽出

- ・ 隠れたオブジェクトの検索（解除Key検索）

- 「Guess 256 byte XOR keys」を用いて隠れたオブジェクトが無い検索



オブジェクト抽出

- 偽装RTFファイルを抽出してコピー&保存
 - 「Copy to new file」を用いて偽装RTF部分をコピー

The screenshot shows the FileInsight application interface. A blue callout bubble on the left points to the 'Copy to new file' button in the 'Plugins' menu, with the text: 'オフセットからRTFフォーマット部分を選択し適用' (Select and apply the RTF format part from the offset). A yellow arrow points from this bubble to a yellow callout bubble at the bottom left, which contains the text: '選択部分が新しいファイル「New File」として生成される' (The selected part is generated as a new file 'New File'). A red box highlights the 'Copy to new file' button and the 'New file*' dialog box. The main window displays a hex dump of the file 'ウエア.doc'. The 'Values' pane on the left shows the decoded content, including RTF tags like '\rtf1\ansi\ansicpg932\deflang1033\deftab840\fonttbl{\f0\froman\fpqr1\ch...'. The status bar at the bottom indicates 'Offset: 2FC50h (195664)' and 'Length: C1Dh (3101)'.

オフセットからRTFフォーマット部分を選択し適用

Copy to new file

選択部分が新しいファイル「New File」として生成される

FileInsight - ウェア.doc

Plugins

Copy to new file

Decode

Method Xor

Key Enter key here, for example 0xA8.

Select area to decode

Values

BE HEX

Byte 123

WORD 23675

DWORD 1953651835

ASCII #rtf1\ansi\ansicpg932

Unicode ?????????s????????

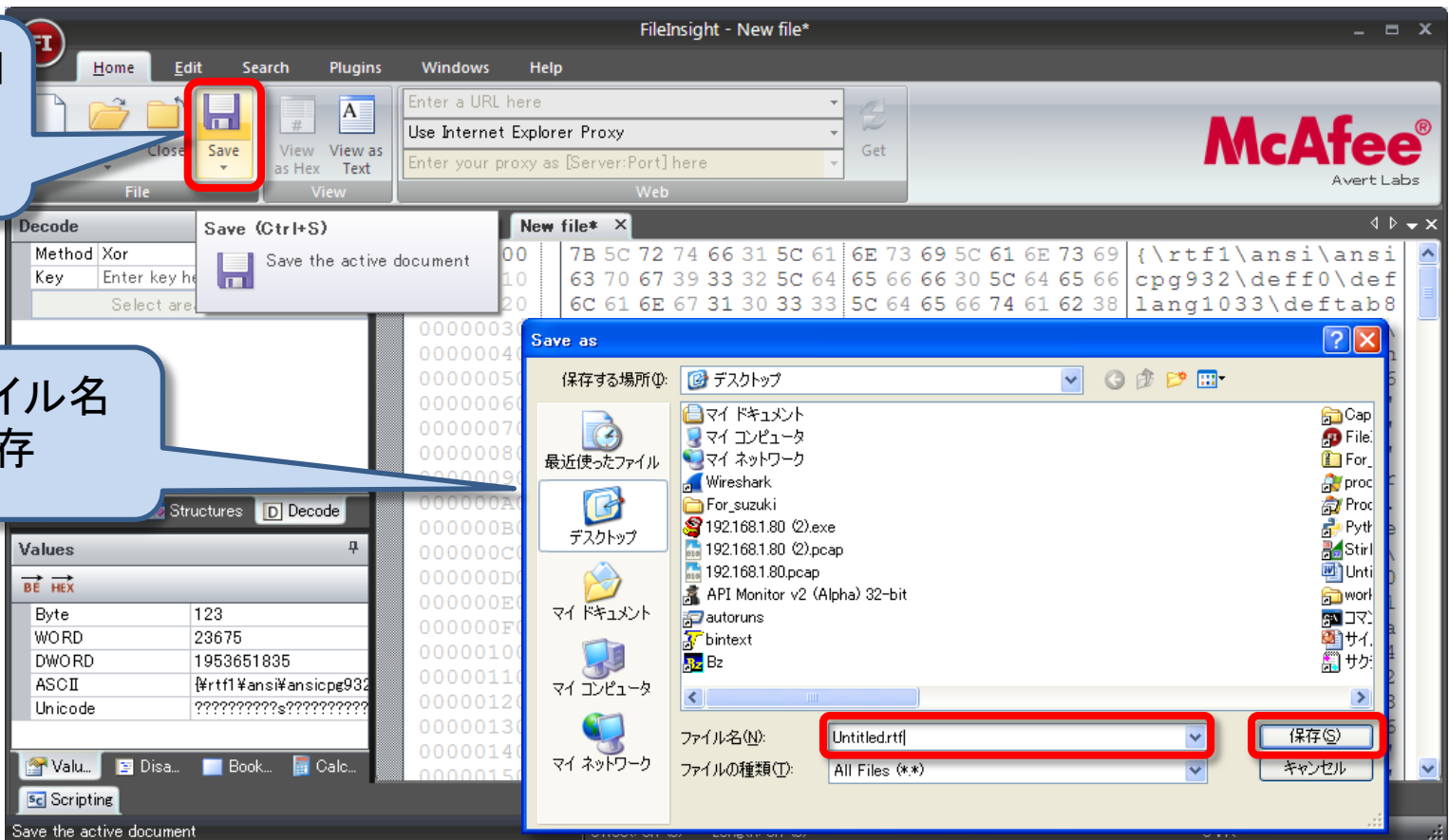
Offset: 2FC50h (195664) Length: C1Dh (3101) OVR

オブジェクト抽出

- ・ 偽装RTFファイルを抽出してコピー＆保存
 - 「Home」メニュー「Save」を選択し、偽装ファイルを保存

「New file*」
タグを選択
してSAVE

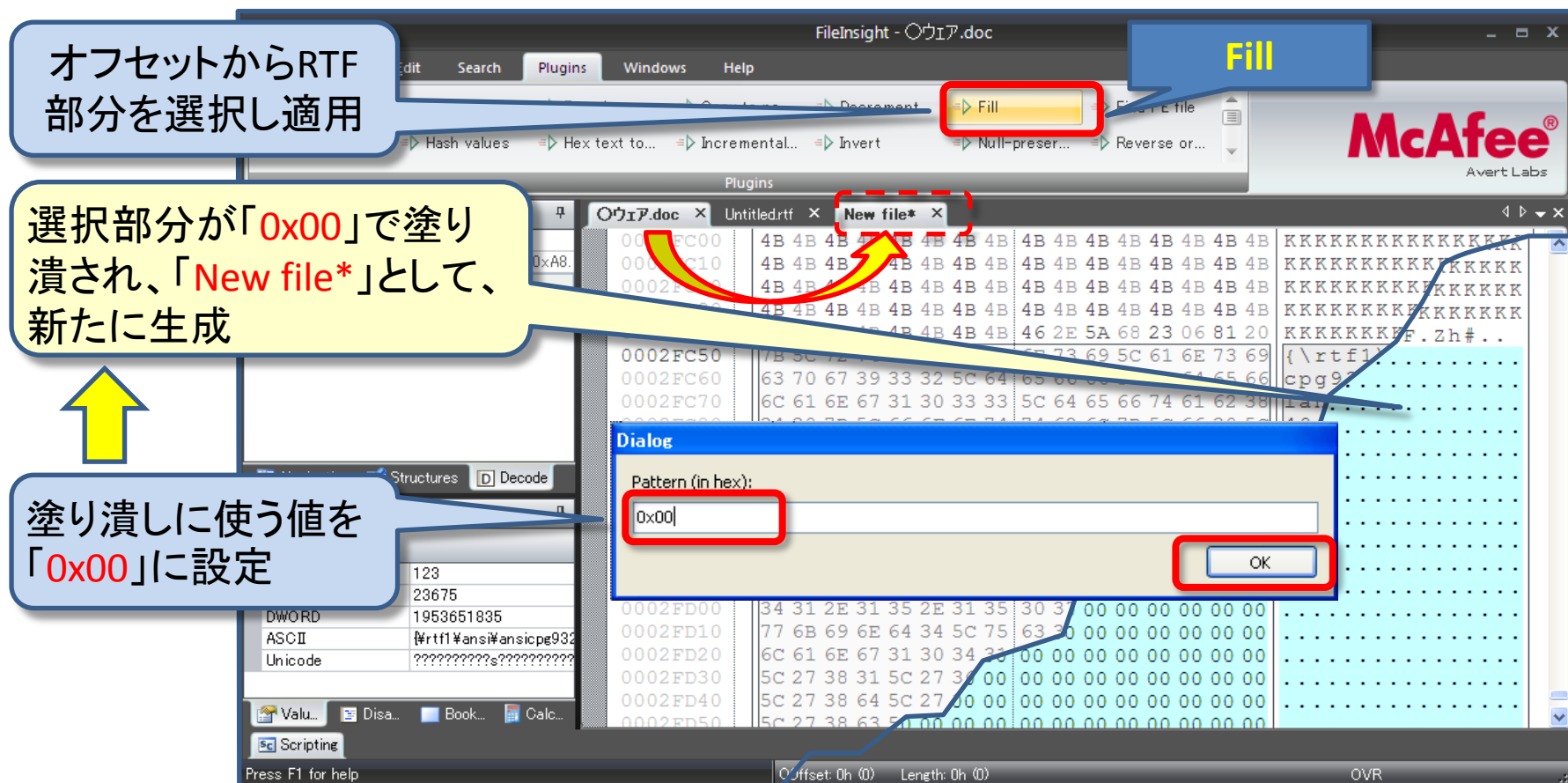
任意のファイル名
を付けて保存



オブジェクト抽出

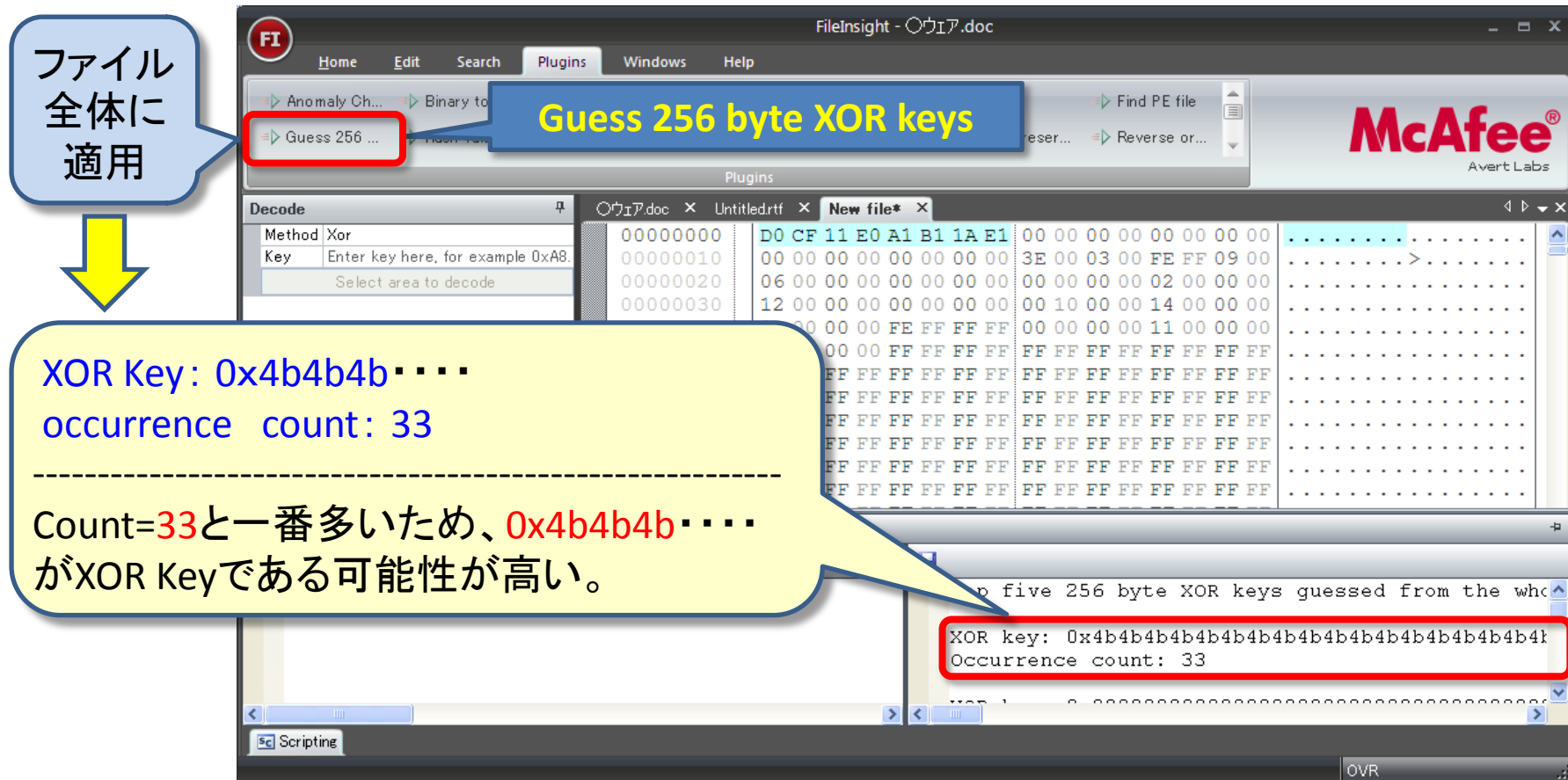
- 偽装RTF抽出後の後処理

- 解析を進める上で邪魔となるため、抽出したRTF部分をNULLで塗り潰す。



オブジェクト抽出

- **再度隠れたオブジェクトを検索（解除Key検索）**
 - 「Guess 256 byte XOR keys」を用い、改めて隠れオブジェクトを検索



オブジェクト抽出

- 難読化解除の開始

- 推測したXOR Keyを手掛かりに難読化を紐解いていく

ファイル全体に適用

Decode欄
Method: Xor
Key: 0x4b (0x4b4b4b...)
→ 「Decode」押下

「0x4b」でXOR掛けられた内容に変化

Values	BE	HEX
Byte	208	
WORD	53200	
DWORD	3759263696	
ASCII	?????	
Unicode	??Q?	

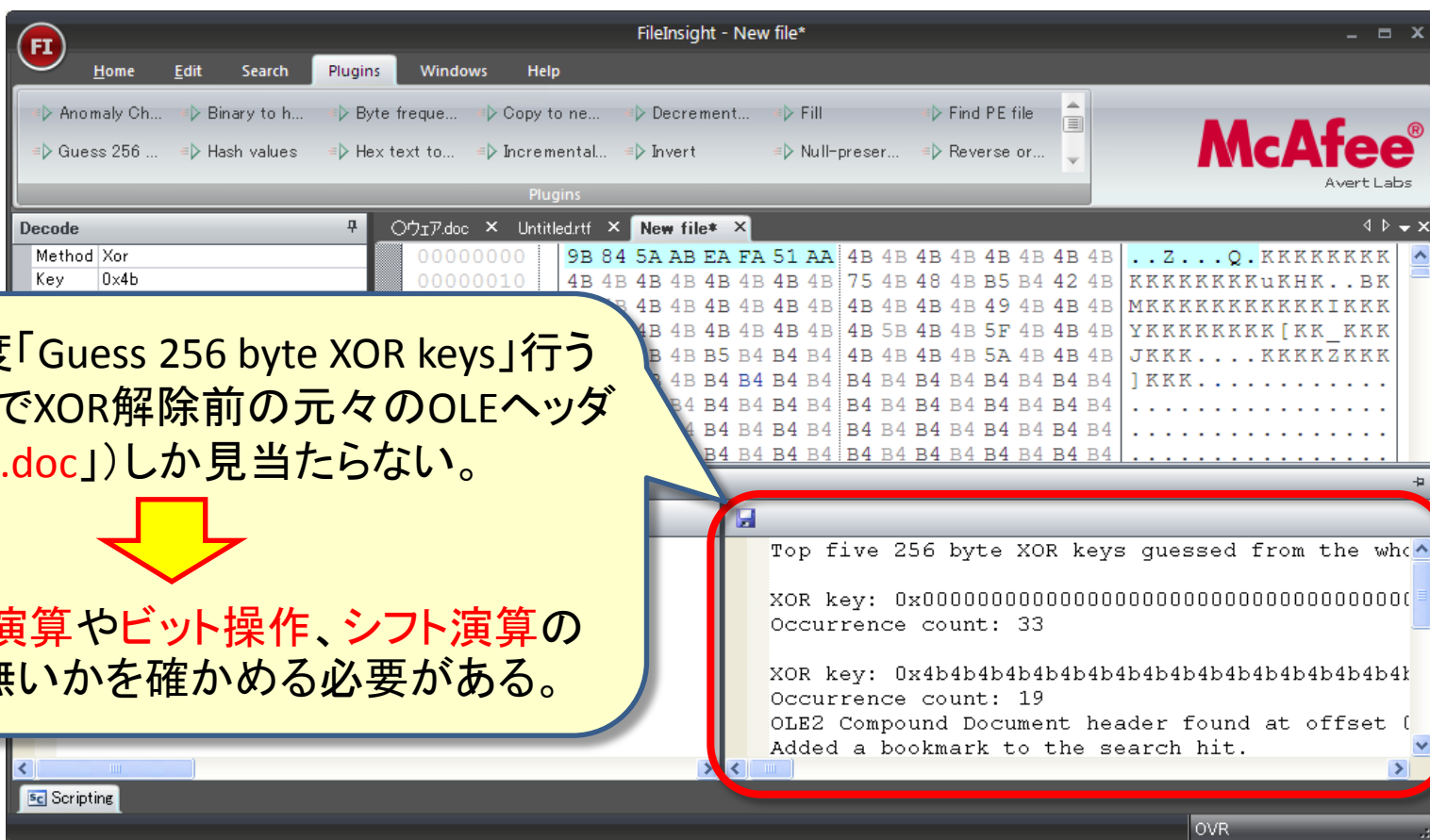
Navigation Structures Decode

Offset: 0h (0) Length: 3086Dh (198765) OVR

オブジェクト抽出

- 難読化解除方針の再考

- － XOR Keyのみでは解除できないため、次の手段を検討する



オブジェクト抽出

- 再度解除Keyを検索

- 他のビット操作やシフト演算がKeyとなる可能性の有無を調べる

ファイル全体
に適用

XOR hex search

隠れている可能性のあるEXE
ファイルのマジックナンバー
「MZ」=0x4d5a」を検索ワード
として設定(効率化のため0x9000追加)

Dialog

Search keyword (in hex):

0x4d5a9000

OK

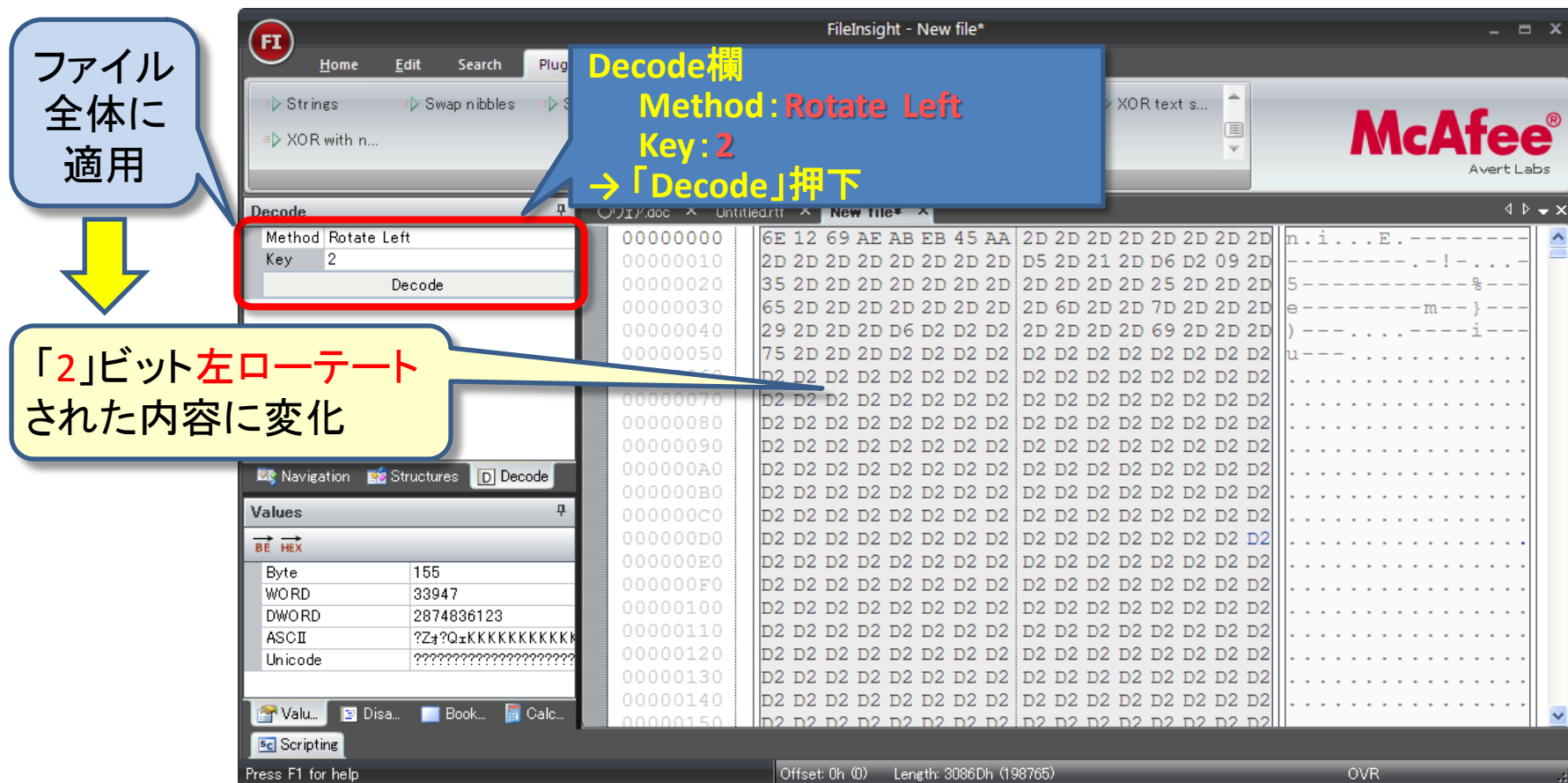
「ROL(Rotate Left):2ビット」が
Keyである可能性が高いこと
が判明。

Search XORed / bit-rotated data in the whole file.
ROL bit: 2 offset: 0x1f248 search hit: 4D5A90000
ROL bit: 2 offset: 0x289d8 search hit: 4D5A90000
Added bookmarks to the search hits.

オブジェクト抽出

● 難読化解除の再開

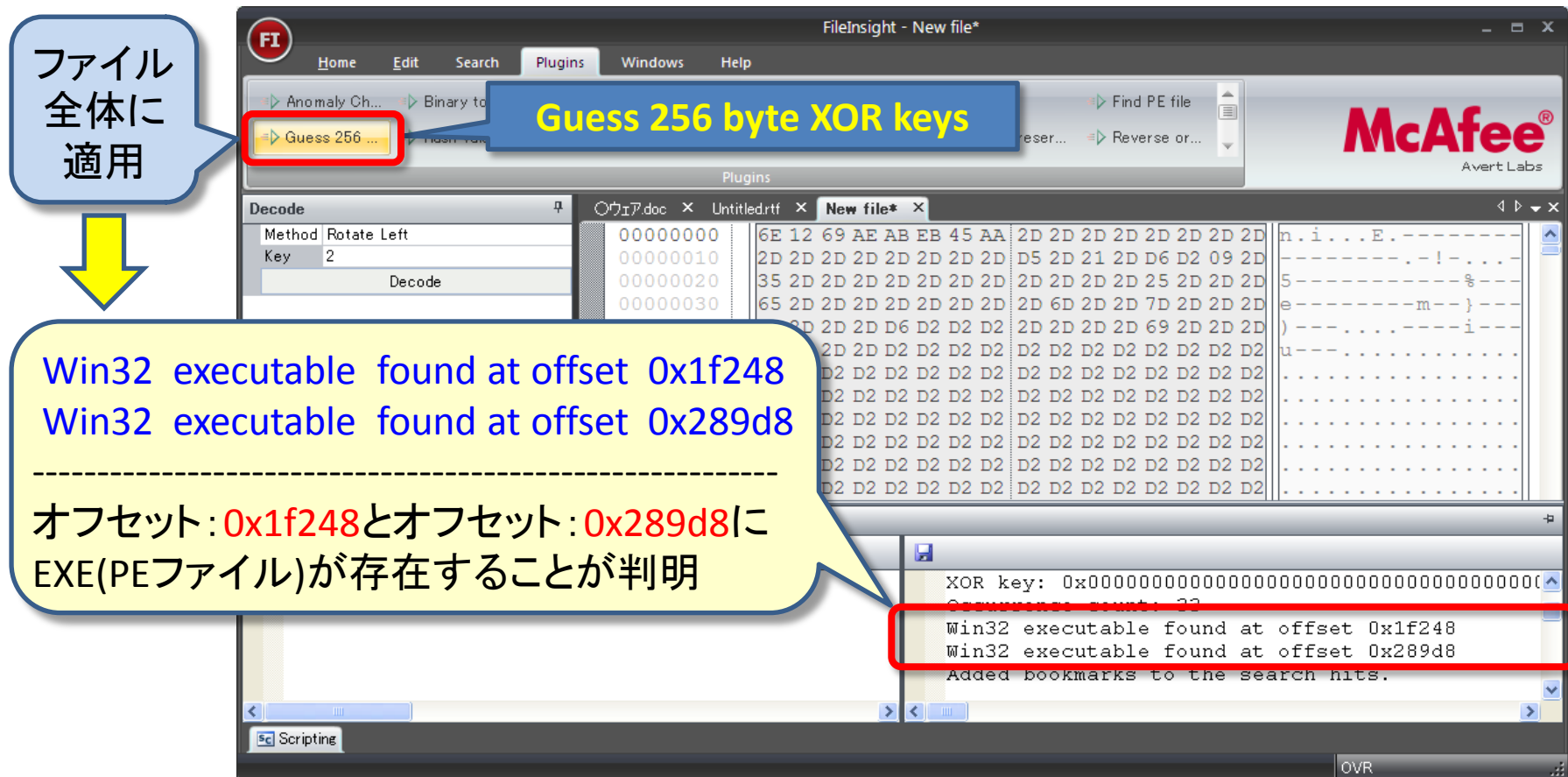
- 推測したROL Keyを用いて難読化を解除



オブジェクト抽出

- 再度オブジェクトを検索

- 「Guess 256 byte XOR keys」を用い、改めて隠れオブジェクトを検索



オブジェクト抽出

- 再度オブジェクトを検索

- 「Find PE file」を用いてサイズを含めて再度確認

ファイル
全体に
適用



○1つ目のPEファイル

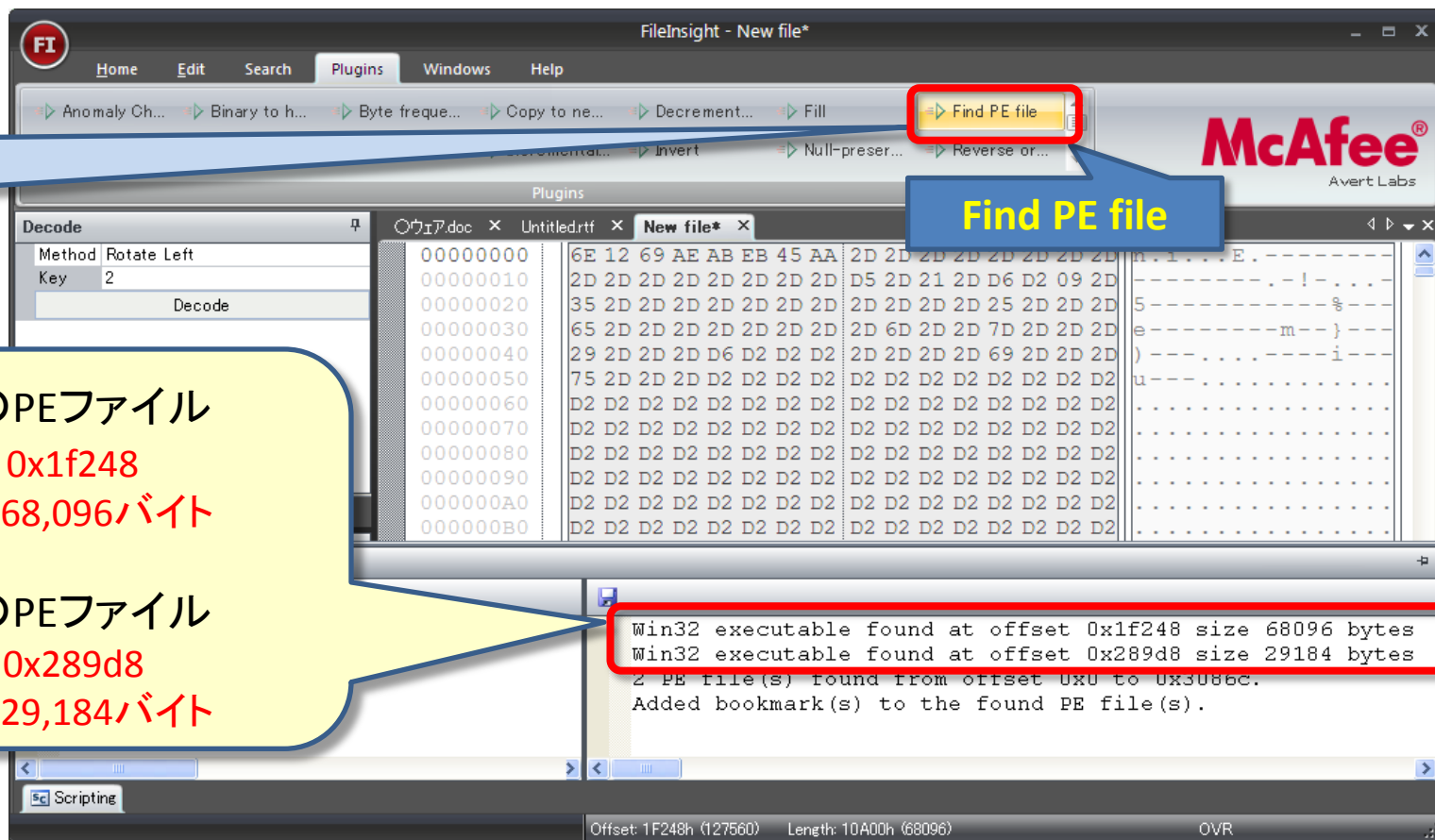
オフセット: 0x1f248

サイズ: 68,096バイト

○2つ目のPEファイル

オフセット: 0x289d8

サイズ: 29,184バイト

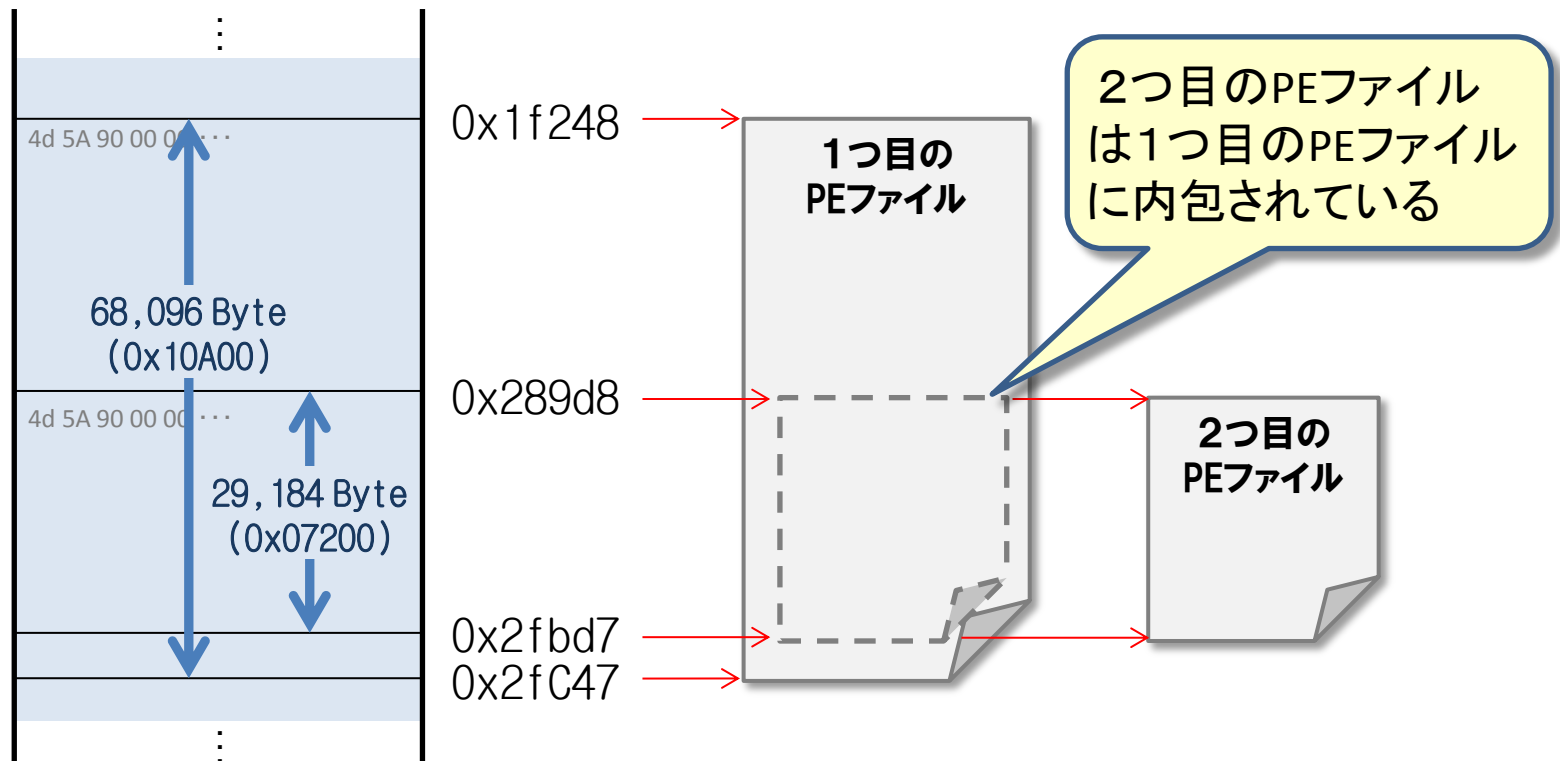


オブジェクト抽出

- 発見したPEファイル間の相関

- 各オフセット値とサイズからPEファイル間の関係を確認

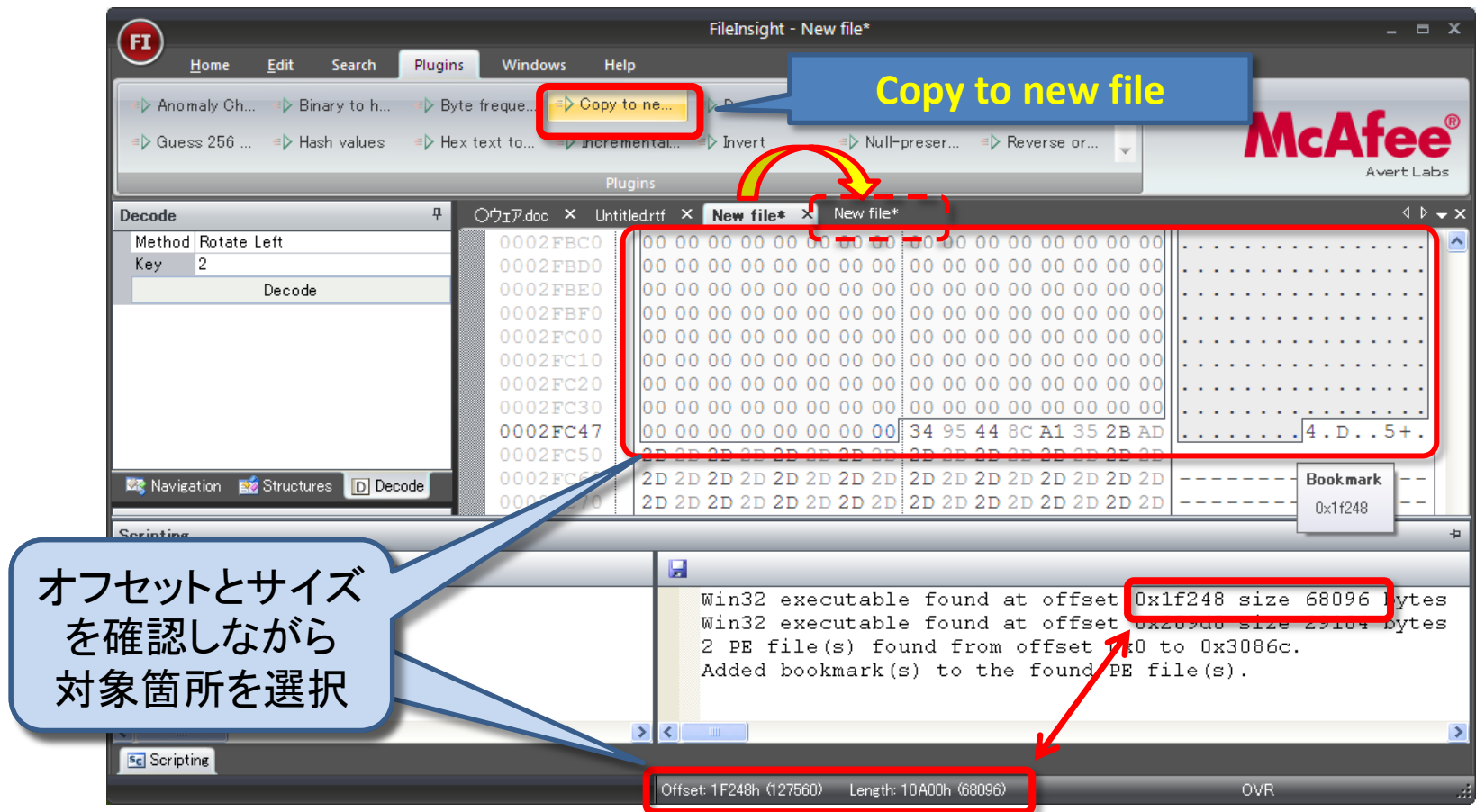
```
Win32 executable found at offset 0x1f248 size 68096 bytes  
Win32 executable found at offset 0x289d8 size 29184 bytes
```



オブジェクト抽出

- PEファイル抽出

- 1つ目のPEファイル部分を選択しコピー



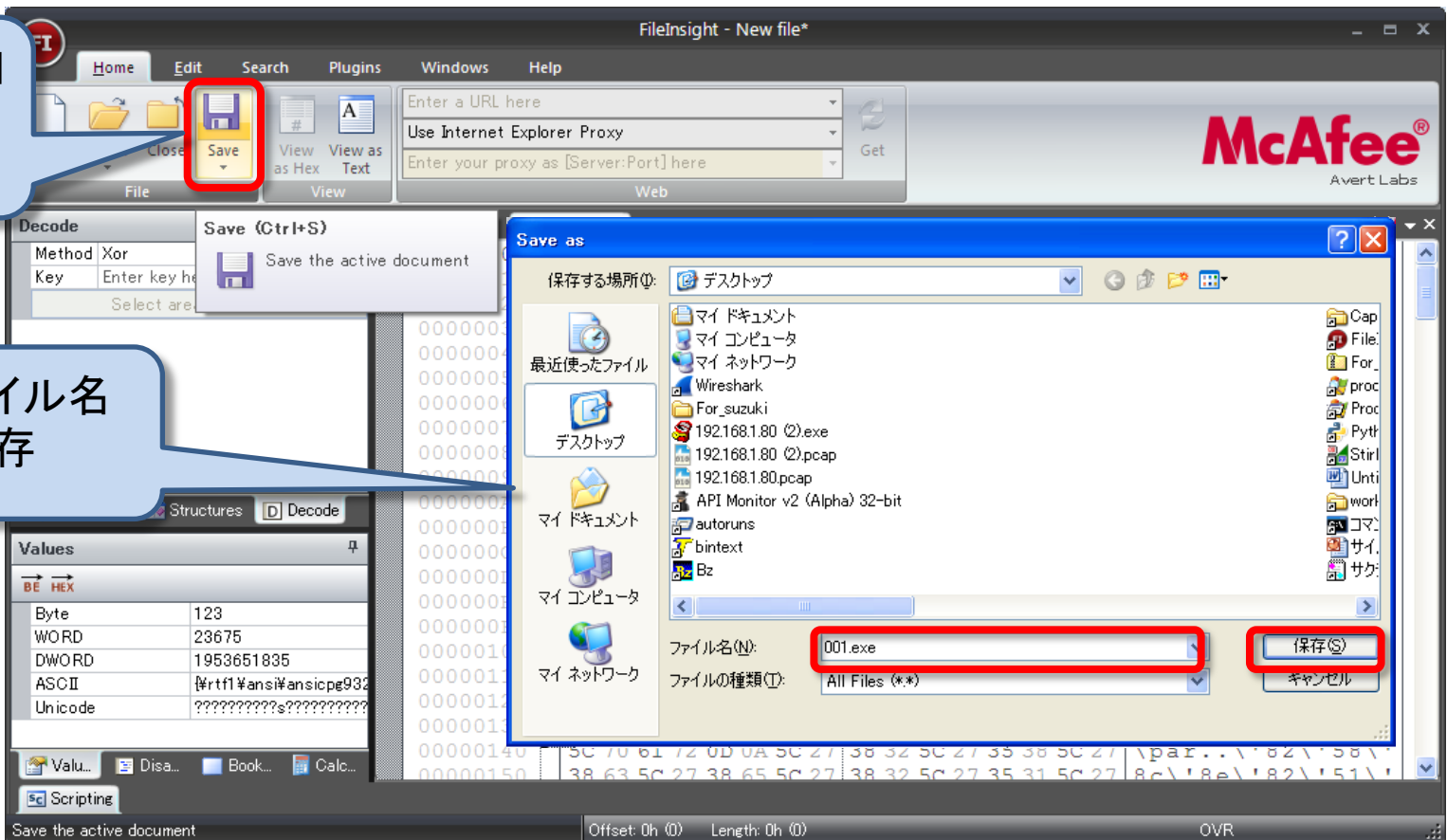
オブジェクト抽出

- PEファイル抽出

- 1つ目のPEファイル部分を任意ファイルとして保存

「New file*」
タグを選択
してSAVE

任意のファイル名
を付けて保存



オブジェクト抽出

- PEファイル抽出

- 2つ目のPEファイル部分を選択しコピー

The screenshot shows the FileInsight interface with the 'Plugins' menu open. The 'Copy to new file' option is highlighted with a red box and a blue callout bubble. A red arrow points from this option to a new file tab. The 'Decode' panel shows a hex dump with a red box highlighting a specific range of data. A blue callout bubble points to this range with the text 'オフセットとサイズを確認しながら対象箇所を選択'. The 'Scripting' panel shows the results of the extraction, with a red box highlighting the second PE file found at offset 0x289d8 and size 29184 bytes. A red arrow points from this box to the status bar at the bottom, which displays 'Offset: 289D8h (166360) Length: 7200h (29184)'.

FileInsight - New file*

Home Edit Search Plugins Windows Help

Copy to new file

McAfee®
Avert Labs

Decode

Method Rotate Left
Key 2
Decode

Navigation Structures Decode

Scripting

Win32 executable found at offset 0x1f248 size 68096 bytes
Win32 executable found at offset 0x289d8 size 29184 bytes
2 PE file(s) found from offset 0x0 to 0x30000.
Added bookmark(s) to the found PE file(s).

Offset: 289D8h (166360) Length: 7200h (29184) OVR

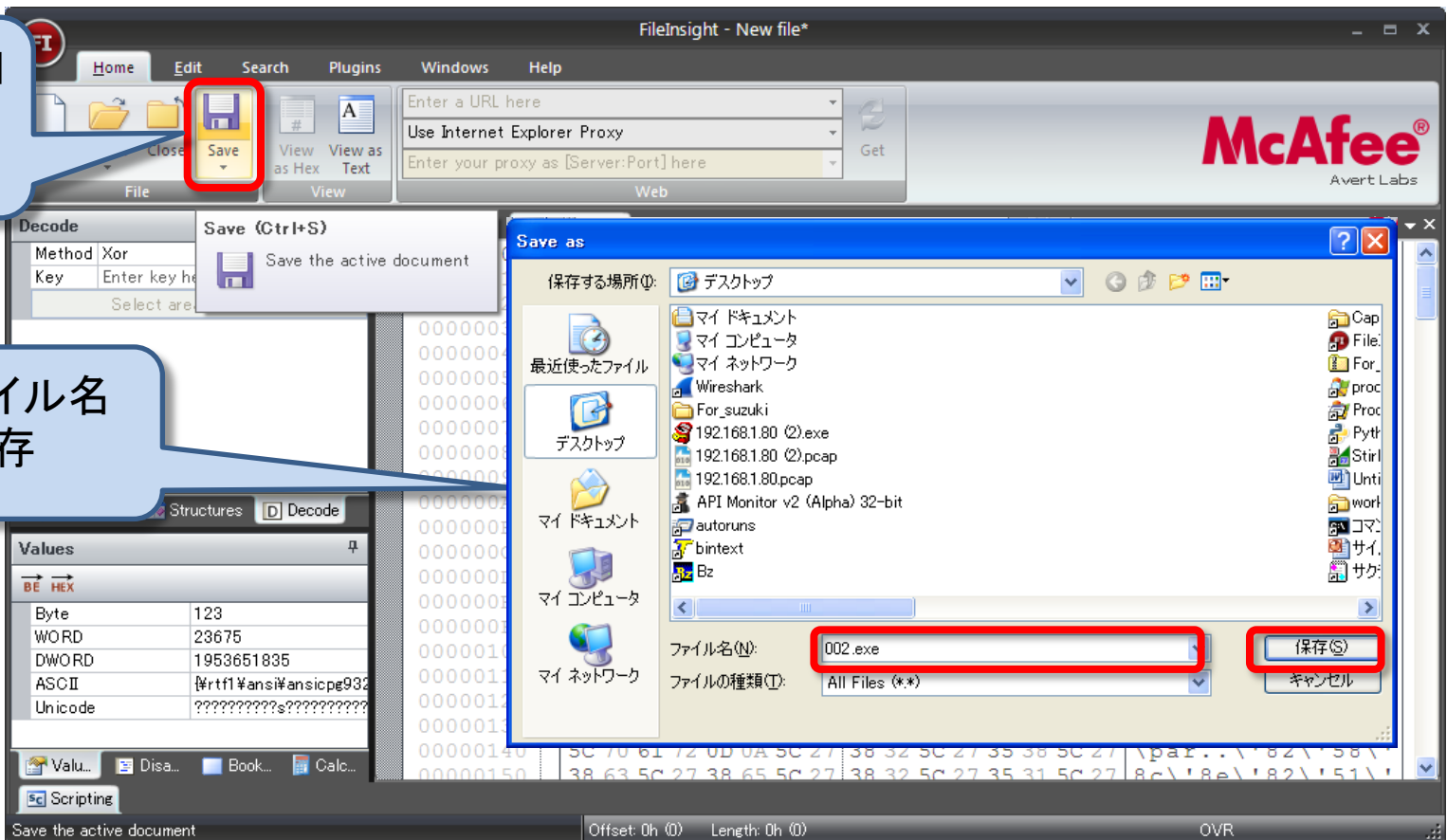
オブジェクト抽出

- PEファイル抽出

- 2つ目のPEファイル部分を任意ファイルとして保存

「New file*」
タグを選択
してSAVE

任意のファイル名
を付けて保存



オブジェクト抽出の結果

- 抽出したPEファイルの確認

- 動的解析にて抽出したPEファイルと比較してみる

```
コマンド プロンプト
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Work>md5deep C:\Work¥001.exe
3efa2b76c4929ae62fbc454c50f8f95 C:\Work¥001.exe

C:\Work>md5deep C:\Work¥002.exe
a75412693f6ebf505f9127df6308f1d8 C:\Work¥002.exe

C:\Work>
```



オブジェクト抽出の結果

- **その他の成果**

- 動的解析の様に環境を汚染させることなく抽出ができた
- 静的解析の様にシェルコードの構造を調べることなく抽出ができた
- 動的解析では失敗していた「偽装ファイル」の抽出にも成功した
- 攻撃対象アプリケーションのバージョン等を気にせず抽出ができた

- **今後の課題**

- シェルコードの挙動を調べていないため、複雑なマルウェアの場合、構成要素間の相関や因果関係が逆に分かり辛くなる可能性がある
- 高度な暗号や圧縮、パックされたマルウェアには通用しない
- ファイルをドロップするマルウェアでない場合は通用しない可能性有