


FileInsight-plugins: Decoding toolbox for malware analysis

Nobutaka Mantani

About me

- Government official
 - Assistant director of Cyber Force Center, National Police Agency of Japan
 - Experiences in malware analysis and digital forensics
- Hobbyist programmer
 - Member of FreeBSD Project (ports committer) since 2001
 - I love open source software! 


FileInsight-plugins

- Collection of plugins for McAfee FileInsight hex editor
 - 67 plugins as of October 2019
- Useful for various kind of decoding tasks in malware analysis
- Development started in 2012
- **Private project and developed at home**
(not a product of Japanese government) 🤗
- Available at <https://github.com/nmantani/FileInsight-plugins>

Background

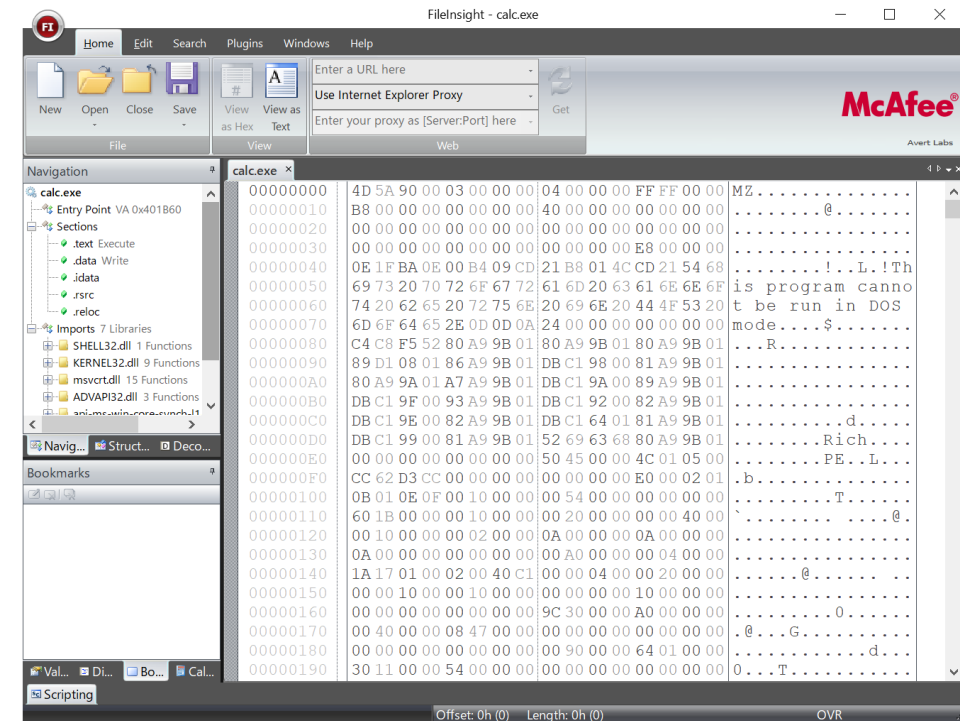
- Attackers obfuscate various data in malware to hamper analysis and hide their intent
 - Embedded executable files and decoy document files
 - Configurations (such as C2 host names and port numbers)
 - Important text strings (such as file paths, registry paths and commands)
 - Codes
- Malware analysts have to deobfuscate them for further analysis

Motivation of development

- At the beginning (2012)
 - Vulnerabilities of Microsoft Office (with Adobe Flash Player) and Adobe Reader were frequently used for malware infection
 - Sometimes exploit code did not work in dynamic analysis due to target version mismatch
 - So I wanted to manually extract obfuscated malware executable files from malicious documents to find C2 host names and block them earlier
- These days
 - I just want moooooooooooooooooooooore **powerful free hex editor!** 

FileInsight

- Free hex editor developed by McAfee, LLC
- Useful built-in functions
 - Decoders (XOR, bit rotate, BASE64 and so on)
 - x86 disassembler
 - Bookmarks
 - Structure viewer (HTML, OLE and PE)
 - Python / JavaScript scripting
- **Extendable with Python plugins!**



FileInsight

- But... no update since 2009
- **Disappeared from the McAfee Free Tools website** 🤪
- Installer of FileInsight is still available at
<http://downloadcenter.mcafee.com/products/mcafee-avert/fileinsight.zip>

Details of FileInsight-plugins

Operation categories

- 67 plugins are categorized into 8 operation categories (as of October 2019)
 - Basic operations
 - Compression operations
 - Crypto operations
 - Encoding operations
 - Misc operations
 - Parsing operations
 - Search operations
 - XOR operations

Added support of algorithms and formats

Compression

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

Encoding

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

Crypto

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

Block cipher modes of operation

- ECB
- CBC
- CFB
- OFB

Added functions

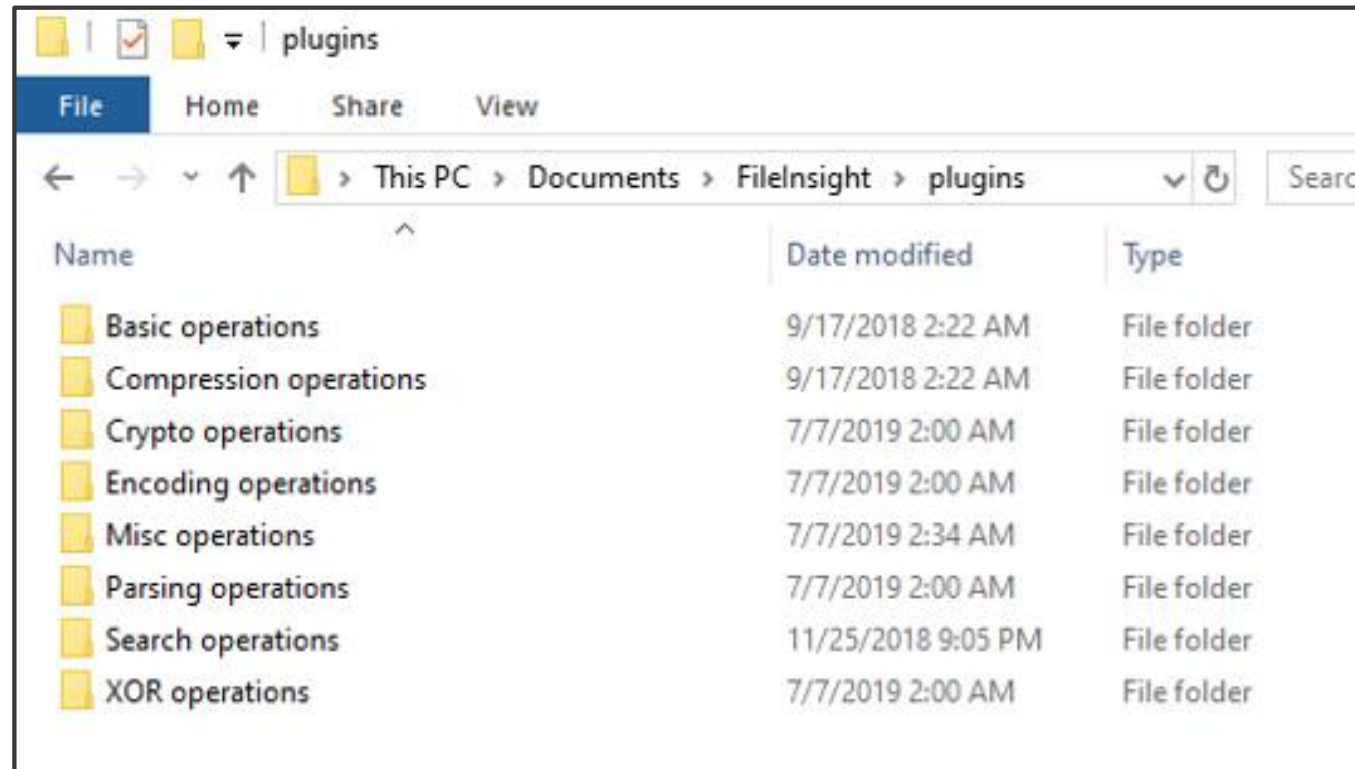
- Opening selected data in a new tab
- Hash value calculation
- File type detection
- Finding embedded files
- Search / replace with regular expressions
- Searching for XORed / bit-rotated data
- Null-preserving XOR
- Guessing 256 bytes XOR keys (effective against XORed data with key increment / decrement)
- Opening data with external tools (customizable with JSON config file)
- File comparison
- Scan with YARA rules
- And more!

Pre-requisites

- Python 2 (x86)
 - **FileInsight is not compatible with Python 2 (x64) and Python 3**
- Some Python modules and so on
 - aPLib
 - binwalk
 - PyCryptodomex
 - backports.lzma
 - python-magic
 - pefile
 - yara-python

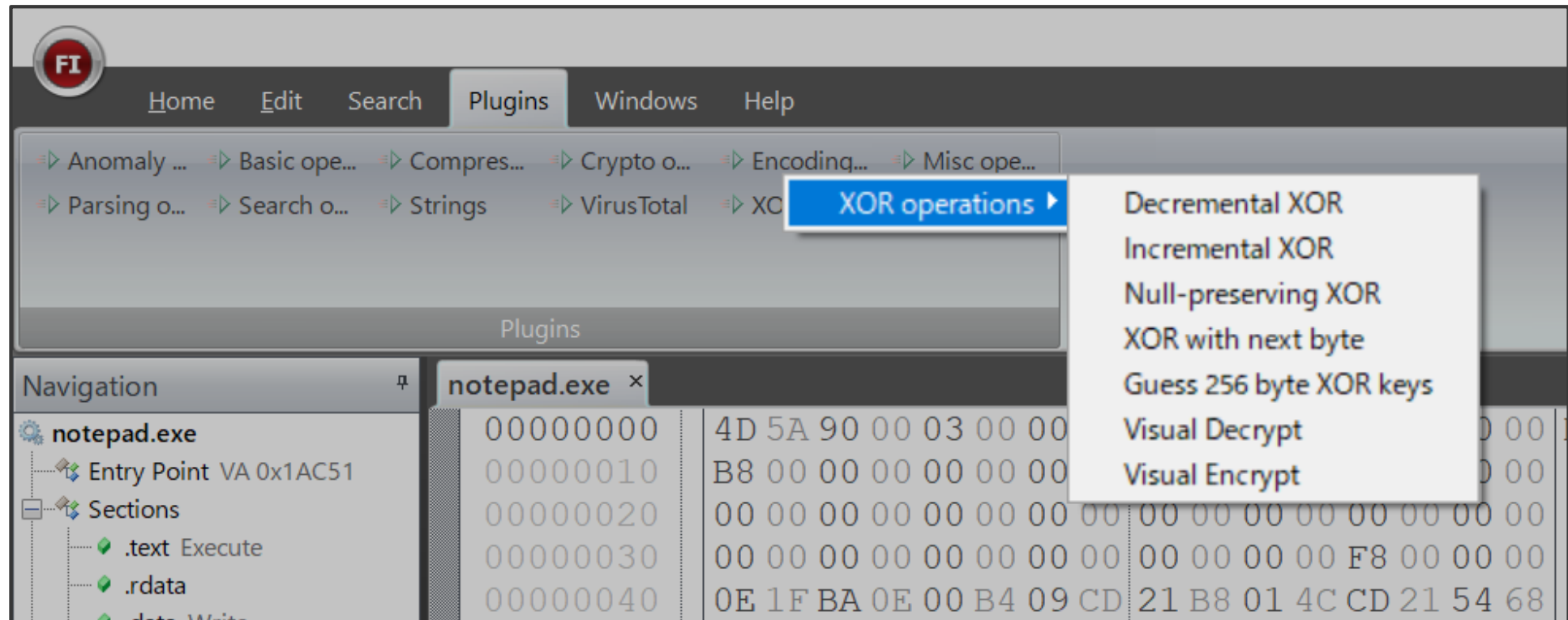
Installation

- Please copy "plugins" folder into %USERPROFILE%\Documents\FileInsight



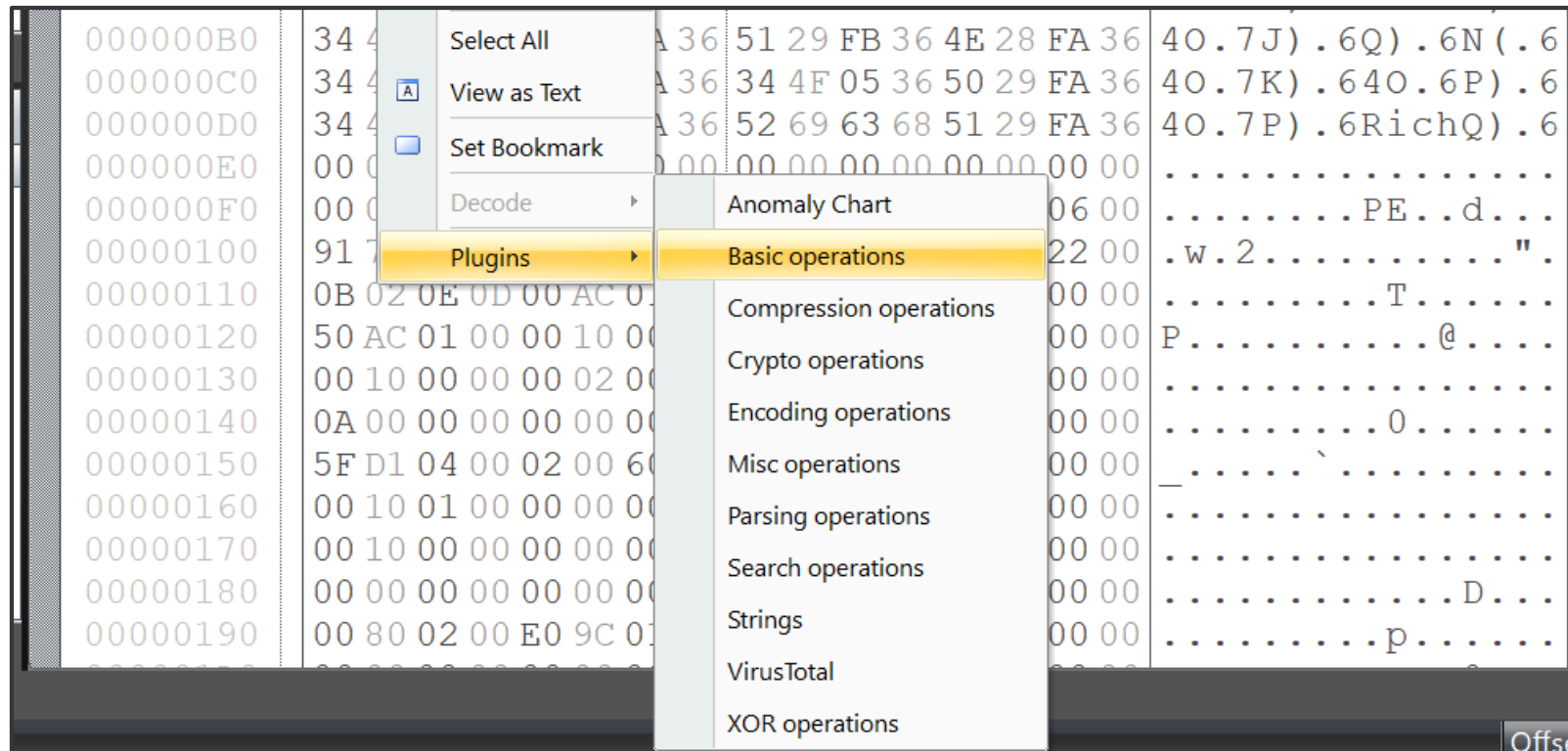
How to use

- Please click a category from the "Plugins" tab then select a plugin



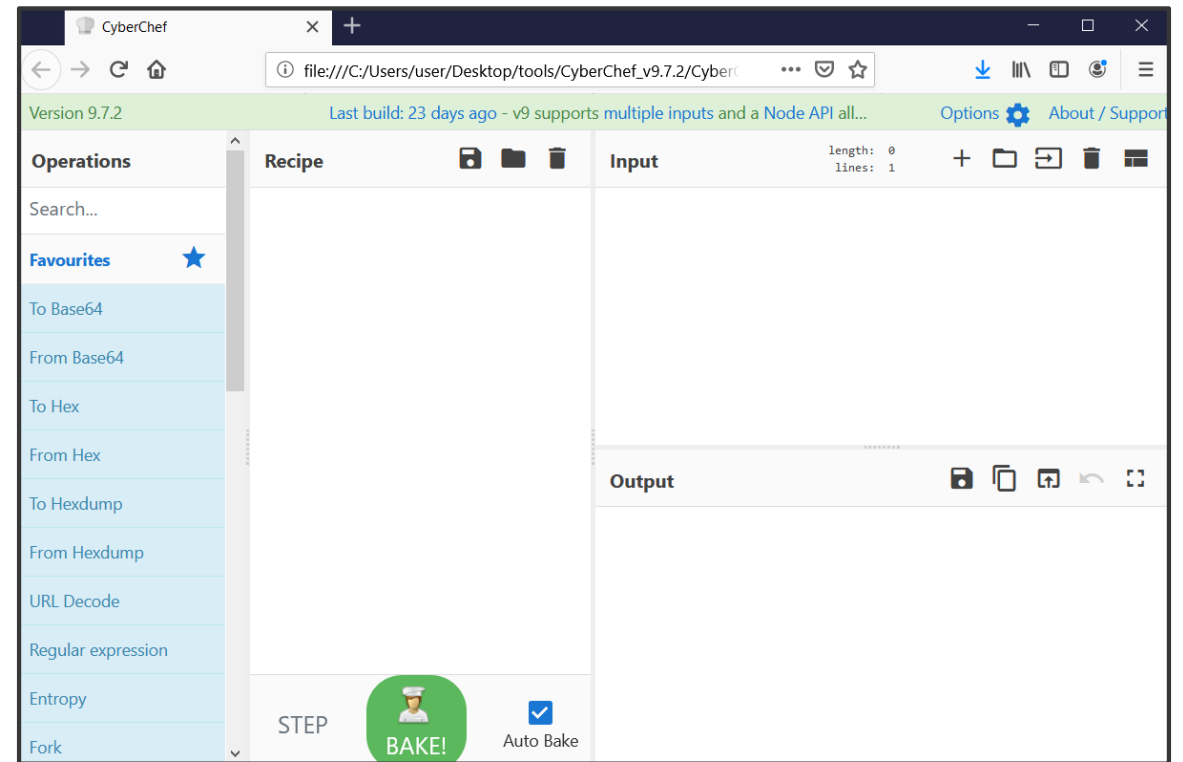
How to use

- Or please select a category from the right-click menu



FileInsight-plugins' good friend: CyberChef

- **Super powerful decoding tool** 🥰
- Works on web browsers
- Actively developed by British government (GCHQ) and many developers in the world
- Not good at manual file editing
- **“Send to” plugin can directly send data to CyberChef (up to 12KB)**



Demo

Demo 1

- Malicious Microsoft Excel file
 - Malware executable file is embedded and obfuscated with XOR and bit-rotate
- Plugins that will be used:
 - Search operations -> XOR text search
 - Paring operations -> Find PE file
 - Misc operations -> Send to

Demo 2

- Malicious rich text file
 - Malware executable file is embedded and obfuscated with XOR while incrementing XOR key (rolling XOR)
- Plugin that will be used:
 - XOR operations -> Guess 256 byte XOR keys

Demo 2

Before:

```
7eb1defca13801b8afb0... x
00002180 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
00002190 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021A0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021B0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021C0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021D0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021E0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 000000000000000000
000021F0 30 30 30 30 30 30 30 30 30 30 0A 7D 7D 7D 00000000000000.}}}}
00002201 50 64 50 44 EF FE EA AE F8 F7 F9 F6 F3 F2 F5 F1 PdPD.....
00002210 08 60 01 00 00 5E 05 00 17 00 00 00 63 3B 5E 74 .`...^.....c;^t
00002220 6D 6B 62 68 7F 7A 56 65 78 7E 66 7D 65 78 3C 77 mkbh.zVex~flox<u
00002230 78 79 16 5A 42 89 1A 18 1C 1D 1E 1B 20 21 22 DC xy.ZB.....
00002240 DB 25 26 9F 28 29 2A 2B 2C 2D 2E 6F 30 31 32 33 .%&.( ) *+, -
00002250 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 456789:;<=>?@ABC
00002260 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 BB DEFGHIJKLMNOPQR.
00002270 54 55 56 59 47 E3 54 5B E8 54 93 7E D8 60 2E AE TUVYG.T[.T.~.`.
00002280 45 31 0E 0E 1B 49 1A 19 03 0A 1C 0E 1D 51 11 12 E1...I.....Q...
00002290 1A 1B 19 03 58 1B 1F 5B 0E 08 10 5F E9 EF A2 C7 ....X..[..._.....
000022A0 CB D6 A6 EA E7 ED EF A5 81 80 84 AB 90 91 92 93 .....
000022B0 94 95 96 2A A4 1F EC 62 C1 75 BB 66 FD 49 87 5A ...*...b.u.f.I.Z
000022C0 F9 4D 83 25 E9 4D 8F 53 F1 45 8B D5 F1 57 97 5F .M.% .M.S.E...W._
000022D0 E9 5D 93 2C FA 42 9F 41 E1 55 9B 46 9D 28 E7 76 .] ., .B.A.U.F. (.v
000022E0 99 2D E3 D6 8A 2B EF 0B 91 25 EB DE 92 32 F7 20 .-...+...%...2.
000022F0 89 3D F3 C6 9A 35 FF 23 81 35 FB 8D 89 82 8A 1A .=...5.#.5.....
```

After:

```
7eb1defca13801b8afb0... x
00002180 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 40 41 42 43 TUVWXYZ [\ ] ^ _ @ABC
00002190 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F B0 B1 B2 B3 DEFGHIJKLMNÖ....
000021A0 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF A0 A1 A2 A3 .....
000021B0 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF 90 91 92 93 .....
000021C0 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F 80 81 82 83 .....
000021D0 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F F0 F1 F2 F3 .....
000021E0 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF E0 E1 E2 E3 .....
000021F0 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF EA 9C 9F 9E .....
00002200 B4 81 B6 A3 07 17 00 45 14 1A 17 19 03 03 07 02 .....E.....
00002210 FC 95 F7 F7 F8 A7 FF FB EB FD FE FF 63 3A 5C 77 .....c:\w
00002220 69 6E 64 6F 77 73 5C 6E 74 73 68 72 75 69 2E 64 indows\ntshruid
00002230 6C 6C 00 4D 5A 90 00 03 00 00 00 04 00 00 00 FF ll.MZ.....
00002240 FF 00 00 B8 00 00 00 00 00 00 00 00 40 00 00 00 00 .....@....
00002250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00002260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E8 .....
00002270 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD .....!..L.
00002280 21 54 68 69 73 20 70 72 6F 67 72 61 6D 20 63 61 !This program ca
00002290 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20 44 nnot be run in D
000022A0 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 OS mode....$.
000022B0 00 00 00 BD 3C 86 76 F9 5DE8 25 F9 5DE8 25 F9 ....<.v.]%.].%.
000022C0 5DE8 25 82 41 E4 25 F8 5DE8 25 7A 41 E6 25 EC ].%.A.%.].%zA.%.
000022D0 5DE8 25 9B 42 FB 25 FA 5DE8 25 F9 5DE9 25 B5 ].%.B.%.].%.].%.
000022E0 5DE8 25 11 42 E2 25 C0 5DE8 25 11 42 E3 25 F3 ].%.B.%.].%.B.%.
000022F0 5DE8 25 11 42 EC 25 F8 5DE8 25 52 69 63 68 F9 ].%.B.%.].%Rich.
```



Demo 3

- PHP web shell
 - Code is obfuscated with BASE64, ROT13, Deflate and so on
- Plugins that will be used:
 - Encoding operations -> ROT13
 - Basic operations -> Reverse order
 - Encoding operations -> Custom BASE 64 decode
 - Compression operations -> Raw inflate
 - Misc operations -> Send to


Demo 3

Before:

```
4e837338fb5a31b638f9... x
<?php
eval(gzinflate(base64_decode(strrev(str_rot13('=8u///5557/ss//xBzBpXjJ5/nVhIuMLepXM09DIgR/ah/IuT36cltrk
?>
```

After:

```
4e837338fb5a31b638f912... x New file* x New file* x New file* x New file* x New file* x
?><?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/";
```



Demo 4

- Zebrocy (Zekapab) malware
 - Some text strings are encoded as hex text
- Extracting and decoding hex strings
- Scanning with a YARA rule based on the string
- Plugins that will be used:
 - Parsing operations -> Strings
 - Search operations -> YARA scan

Demo 4

FileInsight can be used as a YARA rule editor with the "YARA scan" plugin

The screenshot displays the FileInsight application interface. At the top, there are three tabs: 'FancyBearZekapab.bin', 'Strings output*', and 'zebrocy.yar'. The 'FancyBearZekapab.bin' tab shows a hex dump with columns for offset, hex bytes, and ASCII characters. Several hex values are highlighted in green, including '36 36 36 39 37 38 36 35', '66697', '64', '6D6F7465', '2C20546F7', '6C3A20', and '2C20467265653'. The 'zebrocy.yar' tab shows a YARA rule named 'rule apt28_zebrocy' with a 'strings' section containing 13 identifiers (\$s1 through \$s13) with their corresponding fullword patterns. Below the tabs, a 'cripts' window is open, displaying the command 'Scan the whole file.' and the following scan results:

```
Offset: 0x9f280 rule: apt28_zebrocy tag: identifier: $s1 matched: 2C20467265653A20
Offset: 0x9f264 rule: apt28_zebrocy tag: identifier: $s2 matched: 2C20546F74616C3A20
Offset: 0x9fcc8 rule: apt28_zebrocy tag: identifier: $s3 matched: 2E646F6378
Offset: 0xa0218 rule: apt28_zebrocy tag: identifier: $s4 matched: 3F69645F6E616D653D
```

At the bottom of the interface, the status bar shows 'Offset: 9F1FAh (651770) Length: 0h (0) OVR'.

Demo 5

- Binary file that contains a hidden message (not malware)
- Combination with CyberChef
- Plugins that will be used:
 - Parsing operations -> File type
 - Compression operation -> XZ decompress
 - Misc operations -> Send to

Demo 5

Before:

```
message.bin x
00000000  D 37 7A 58 5A 00 00 04 E6 D6 B4 46 02 00 21 01 .7zXZ.....F...!.
00000010  16 00 00 00 74 2F E5 A3 01 02 56 89 50 4E 47 0D .....t/.....V.PNG.
00000020  0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 00 B9 00 .....IHDR.....
00000030  00 00 B9 08 00 00 00 00 51 C6 C1 80 00 00 02 1E .....Q.....
00000040  49 44 41 54 78 DA ED DC 41 72 84 30 0C 05 51 EE IDATx...Ar.0..Q.
00000050  7F E9 C9 32 1B 2C FA DB 43 25 B6 9A 55 66 42 E0 ...2.,...C%.UfB.
00000060  99 2A 64 D9 52 E5 FA EC 7A 5C CA 95 2B 57 AE 5C .*d.R...z\...+W.\
00000070  F9 7F 91 5F CF C7 DD C9 77 1F 7F BF 1B 5E B9 BE ..._.....w.....^..
00000080  0A 11 28 57 DE 50 3E 7E 95 47 AC 9A 7A 77 4A 3D ..(W.P>~.G...zwJ=
00000090  42 2C 50 AE BC AB 7C 18 06 F2 31 D4 84 FA 1E C3 B,P...|...
000000A0  DF 2A 57 AE FC 02 57 05 E1 A2 4E 2E 86 CF 44 B9 *W...W...
000000B0  72 E5 EB B1 85 62 A2 8F CA 95 2B 5F CA CF 87 C b...
000000C0  77 3D 1A 70 A3 B7 57 16 CA 95 6F 2F 07 3B 4C 2F ...
000000D0  FF F4 E2 EE 9C 72 E5 3B CB 71 81 E6 79 4F 0A EC ...
000000E0  47 D3 88 F2 CD 0A 97 72 E5 3B CB C1 3B 0E B6 93 G.....
000000F0  A3 AD B1 FA 02 0F 69 86 72 E5 BDE4 B4 A4 03 56 .....i...
00000100  D0 F9 4A 3B 8F 32 CA 95 37 93 03 25 98 F3 EB 80 ..J;.2..7...
00000110  30 99 6B DC AE C3 95 2B EF 25 8F C2 05 88 19 93 0.k....+...
00000120  53 FB EC EC AF 5C F9 E1 72 5A C3 A9 5F FE 3A 41 S....\...r?
00000130  C8 0B C5 78 F6 57 AE BC A1 1C 4C FC 24 0C C0 84 ...x.W...
00000140  3C CA DE 95 2B 6F 26 07 6B 5F 7A 0A 9D DF E9 63 <...+o&.k
00000150  23 F9 B9 72 E5 87 CB E9 86 13 D8 E8 AA 3B 2C A2 #.r.....
00000160  F6 A7 6C FF 5C B9 F2 C3 E5 51 E3 03 E8 C9 98 5C ..l.\....Q.....\
00000170  28 93 0E 65 E5 CA 1B CA A3 7C FA E1 D2 73 39 04 (...e.....|...s9.
00000180  CF EF 0F 2B CF 2B CF DF 7C 70 22 69 60 07 F2 16
```

After:

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



Wrap-up

- FileInsight-plugins makes FileInsight hex editor more powerful like Iron Man suits
- Useful for various kind of decoding tasks in malware analysis
- Available at <https://github.com/nmantani/FileInsight-plugins>
- I am waiting for your feedbacks (pull requests, bug reports and feature requests)! 😊

Thank you!

This slide deck will be available at my GitHub repository:

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

Appendix

List of FileInsight API functions

Please see FileInsight help for details of the functions

- `getLength()`
- `getBytesAt()`
- `setBytesAt()`
- `setBookmark()`
- `getSelection()`
- `getSelectionOffset()`
- `getSelectionLength()`
- `gotoBookmark()`
- `download()`
- `newDocument()`
- `showSimpleDialog()`
- `decode()`

List of FileInsight API functions

Please see FileInsight help for details of the functions

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

Sample plugin code

```
# Open selected region or entire file in a new tab
length = getSelectionLength() # Get length of selected region
if length > 0:
    data = getSelection() # Get selected region
else:
    data = getDocument() # Get entire file
newDocument("New file", 1) # Create a new tab with hex view mode
setDocument(data) # Copy data to the new tab
```