What is PDF, word docs, VBA

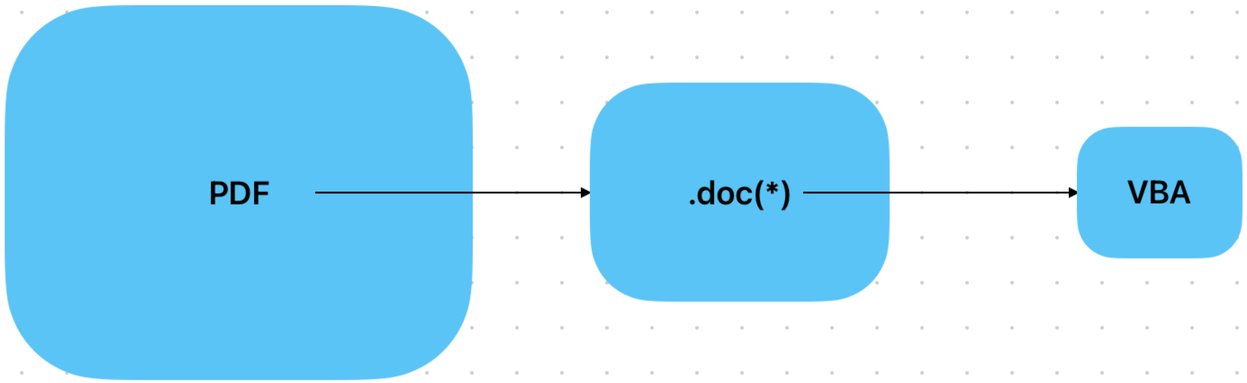
With Malware sample analysis and how to make one (simple)

Tanaka 10/05/2023

Agenda

What is PDF, word docs and VBA

- PDF
  - keywords (OpenAction, JS)
  - example malware analysis
  - file structure
  - how to make one
  - Mitgations



- word docs
  - keywords (macro/VBA)
  - file structure(OLE2, OOXML)
  - example malware analysis
  - detection evasion techniques
  - mitigations

- VBA
  - keywords ()
  - example malware analysis
  - detection evasion techniques

PDF

PDF structure

HEADER

```
%PDF-1.1
```

BODY

```
1 0 obj
<<
  /Pages 2 0 R
>>
endobj

2 0 obj
<<
  /Type /Pages
  /Count 1
  /Kids [3 0 R]
>>
endobj

3 0 obj
<<
  /Type /Page
  /Contents 4 0 R
  /Parent 2 0 R
  /Resources <<
    /Font <<
      /F1 <<
        /Type /Font
        /Subtype /Type1
        /BaseFont /Arial
      >>
    >>
  >>
>>
endobj
```

FILE

```
<< /Length 47 >>
stream
BT
  /F1 110
  Tf
  10 400 Td
  (Hello World!)Tj
ET
endstream
endobj
```

XREF TABLE

CROSS REFERENCE

```
xref
0 5
0000000000 65535 f
0000000010 00000 n
0000000047 00000 n
0000000111 00000 n
0000000313 00000 n
```

TRAILER

```
trailer
<<
  /Root 1 0 R
>>

startxref
416
%%EOF
```

010:

```
1 0 obj
<<
  /Pages 2 0 R
>>
endobj
```

047:

```
2 0 obj
<<
  /Type /Pages
  /Count 1
  /Kids [3 0 R]
>>
endobj
```

111:

```
3 0 obj
<<
  /Type /Page
  /Contents 4 0 R
  /Parent 2 0 R
  /Resources <<
    /Font <<
      /F1 <<
        /Type /Font
        /Subtype /Type1
        /BaseFont /Arial
      >>
    >>
  >>
>>
endobj
```

313:

```
4 0 obj
<< /Length 47 >>
stream
BT                      BEGIN TEXT
  /F1 110                 FONT F1 (ARIAL) SET TO SIZE 110
  Tf                      SELECT THIS FONT
  10 400 Td               MOVE TO COORDINATE 10, 400
  (Hello World!)Tj        OUTPUT TEXT 'HELLO WORLD!'
ET                      END TEXT
endstream
endobj
```

416:

```
xref                    CROSS REFERENCES
0 5                     5 OBJECTS, STARTING AT INDEX 0
0000000000 65535 f      (STANDARD FIRST EMPTY OBJECT 0
0000000010 00000 n      OFFSET TO OBJECT 1, REV 0
0000000047 00000 n      TO OBJECT 2...
0000000111 00000 n      3...
0000000313 00000 n      4

trailer
<<
  /Root 1 0 R
>>

startxref
416
%%EOF
```

BASICS

PDF IS TEXT BASED, WITH BINARY STREAMS

TYPES

0: STRING
  EX: `(Hello World!)`
/NAME (IDENTIFIERS)
  EX: `/Count 1`
<<>>: DICTIONARY
  EX: `<</key1 value1 /key2 value2>>`
[]: ARRAY
  EX: `[0 1 2 3 4]`

OBJECT REFERENCES

CONTENT IS STORED IN OBJECT
MOST CONTENT CAN BE INLINED OR REFERENCED IN A SEPARATE OBJECT

`/Key1 value` IS EQUIVALENT TO `/Key1 3 0 R` (<OBJECT NUMBER> <REVISION NUMBER> R)

```
[...]
3 0 obj
value
endobj
```

BINARY STREAMS

BINARY STREAM ARE STORED IN SEPARATE OBJECTS LIKE THIS:

```
<object number> <object revision> obj
<< <STREAM METADATA> >>
stream
<STREAM CONTENT>
endstream
endobj
```

(<STREAM METADATA>: STREAM LENGTH, COMPRESSION PARAMETERS...)

TRIVIA

THE PDF WAS FIRST SPECIFIED BY ADOBE SYSTEMS IN 1993

INITIAL VERSIONS OF ADOBE ACROBAT WERE NOT FREE

FILE STRUCTURE

HEAD OF THE FILE

THE '%PDF-' SIGNATURE IDENTIFIES THE FORMAT AND REQUIRED VERSION

XREF

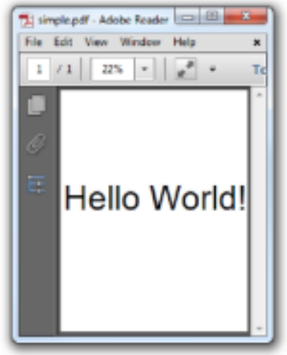
`xref`
<STARTING OBJECT> <OBJECT COUNT>
FOLLOWED BY XREF ENTRIES:
IF (OBJECT IN USE)
  <OFFSET:10> <GENERATION:5> n
ELSE
  <NEXT_FREE_OBJECT:10> <GENERATION:5> f

END OF THE FILE

`startxref`
<XREF OFFSET IN DECODED STREAM>
`%%EOF`

PARSING

THE HEADER `%PDF-1.?` SIGNATURE IS CHECKED TO IDENTIFY THE FILE FORMAT
THE XREF IS LOCATED VIA THE `startxref` OFFSET
THE `xref` TABLE GIVES OFFSET OF EACH OBJECT
THE `trailer` IS PARSED
EACH OBJECT REFERENCE IS FOLLOWED, BUILDING THE DOCUMENT
PAGES ARE CREATED, TEXT IS RENDERED



PDF[101] an Adobe document walk-through

PDF

Keywords

- PDF - portable document: can be opened in windows, macOS, linux

Risky PDF Keywords

- /OpenAction and /AA specify the script or action to run automatically.
- /JavaScript, /JS, /AcroForm, and /XFA can specify JavaScript to run.
- /URI accesses a resource by its URL, perhaps for phishing.
- /SubmitForm and /GoToR can send data to URL.
- /RichMedia can be used to embed Flash in a PDF.
- /ObjStm can hide objects inside an object stream.
- /XObject can embed an image for phishing.
- Be mindful of obfuscation with hex codes, such as /JavaScript vs. /J#61vaScript.

(See examples.)

https://zeltser.com/analyzing-malicious-documents/

PDF

Example malware: 304a28d5e9010331c8f183b5932d0420410cf5e749f84cdd02d9992abd397285

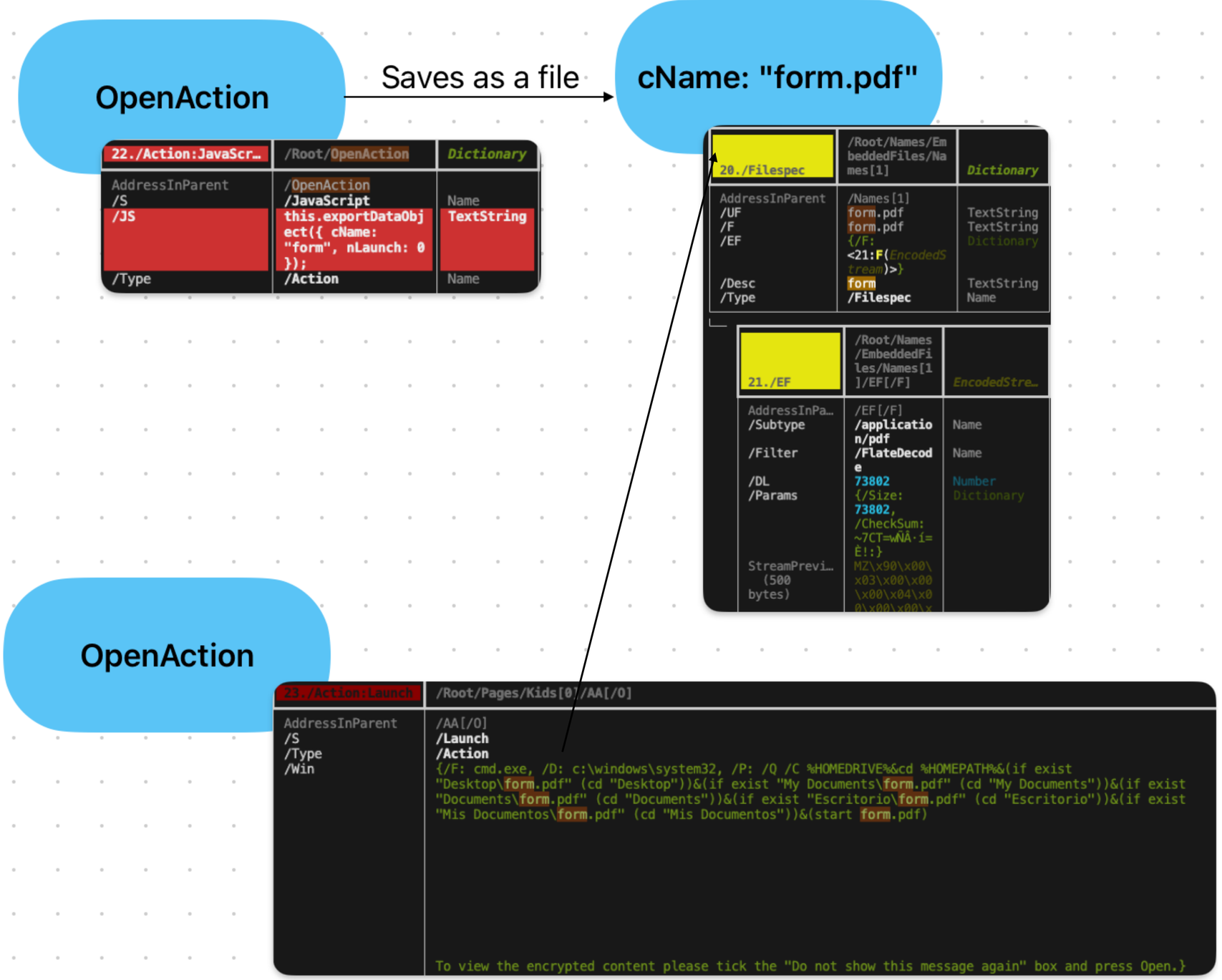
```
→  pdf-analysis python2.7 ../Tools/pdfid.py example-malware.pdf
PDFiD 0.2.8 example-malware.pdf
 PDF Header: %PDF-1.3
 obj                    25
 endobj                 25
 stream                  4
 endstream               4
 xref                    2
 trailer                 2
 startxref               2
 /Page                   2
 /Encrypt                0
 /ObjStm                 0
 /JS                     1
 /JavaScript             1
 /AA                     1
 /OpenAction             1
 /AcroForm               0
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 1
 /EmbeddedFile           0
 /XFA                    0
 /Colors > 2^24          0
```

https://github.com/DidierStevens/DidierStevensSuite/blob/master/pdfid.py
https://github.com/michelcrypt4d4mus/pdfalyzer

Execution flow

- OpenAction prompts a user to save a file “form.pdf”

- After that second OpenAction prompts a user to execute the command.

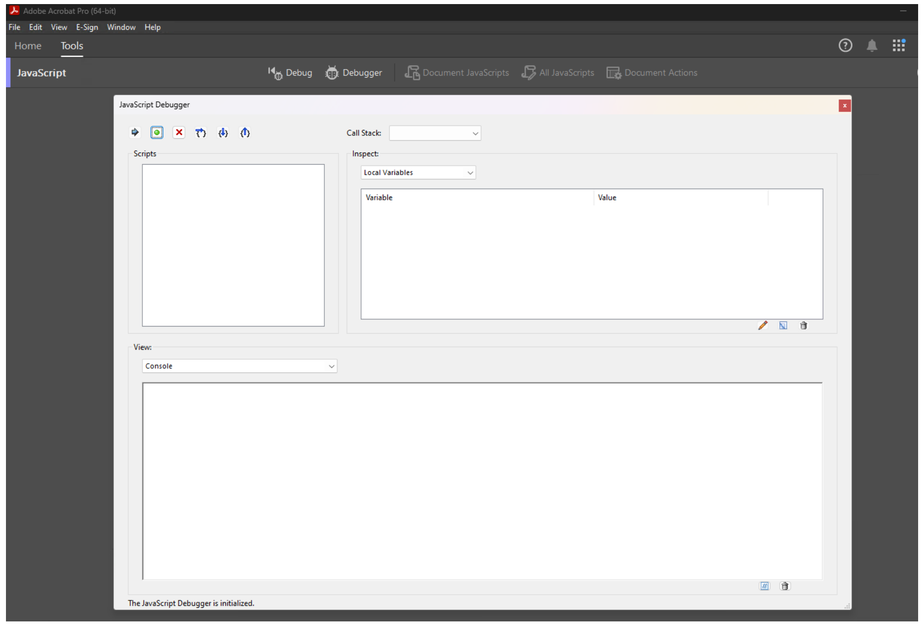


https://bufferzonesecurity.com/the-beginners-guide-to-adobe-pdf-malware-reverse-engineering-part-2/

PDF

How to make malicious pdf document

- Explain the code you wrote, add ms-settings one as demo
- It didn’t work, used Adobe Reader Pro.
- Attached .doc and added exportDataObject.
- Explain what the splitted PDF attack i

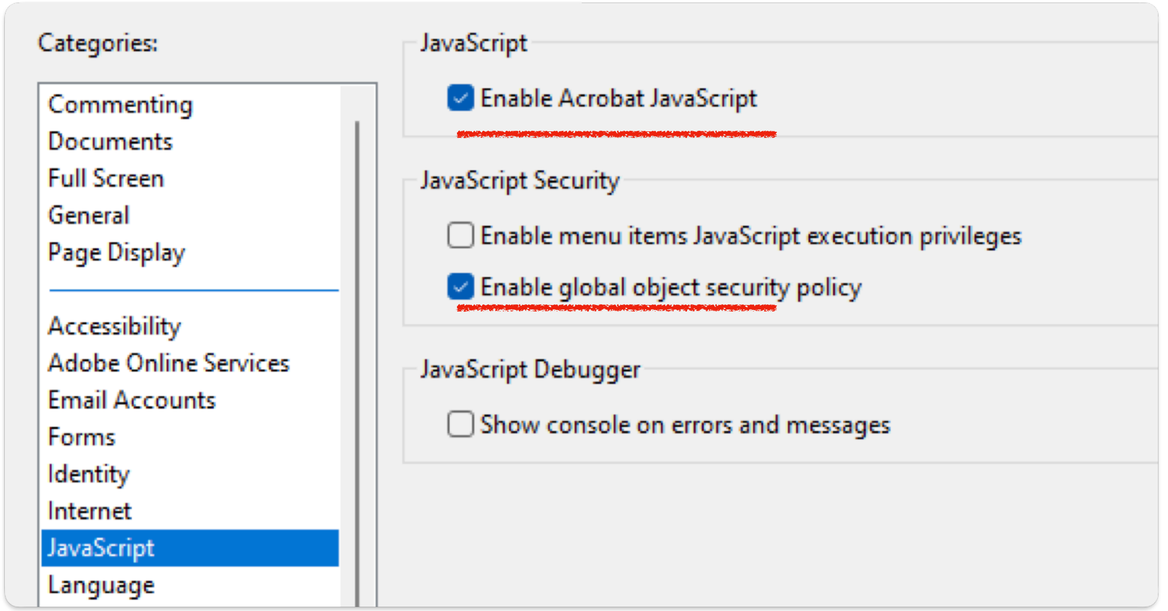


PDF

Mitigations

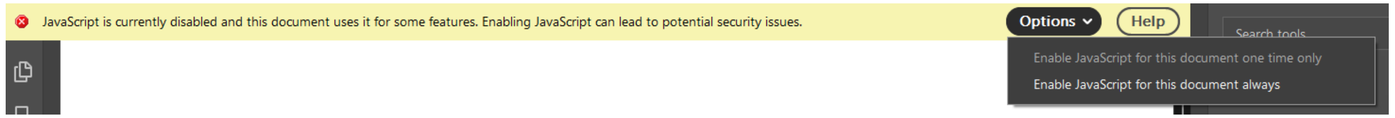
- Turn off Enable Acrobat Javascript
- Turn off Enable Global Object Security Policy : which allows JavaScript globally through APIs, or trusts specific documents containing JavaScripts.

After fresh installation, Javascript policy is enabled by default



Or modify the registry

```
reg add "HKLM\SOFTWARE\WOW6432Node\Policies\Adobe\Acrobat Reader\DC\FeatureLockDown" /v "bDisableJavaScript" /t REG_DWORD /d "0x00000001" /f
reg add "HKLM\SOFTWARE\WOW6432Node\Policies\Adobe\Acrobat Reader\DC\FeatureLockDown" /v "bDisableTrustedFolders" /t REG_DWORD /d "0x00000001" /f
```



https://www.adobe.com/devnet-docs/acrobatetk/tools/PrefRef/Windows/JSPrefs.html

MS office

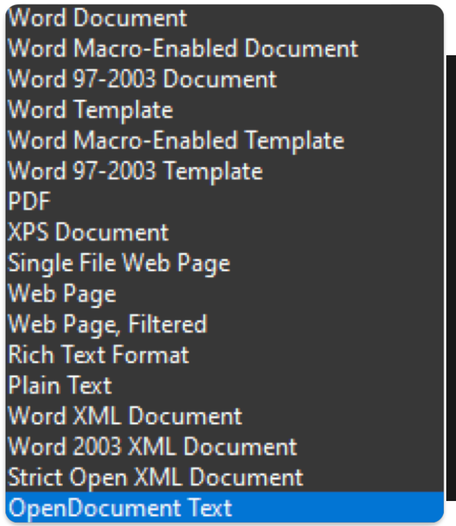
MS Office

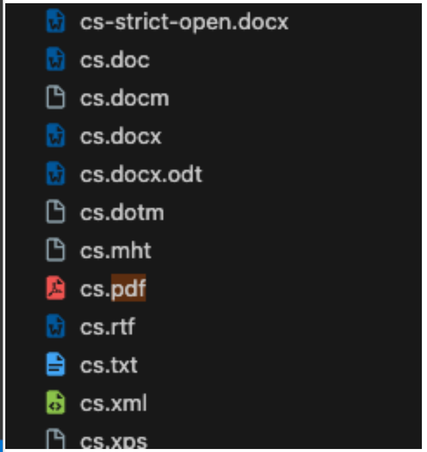
Rant and complain

- What is office document?

word document

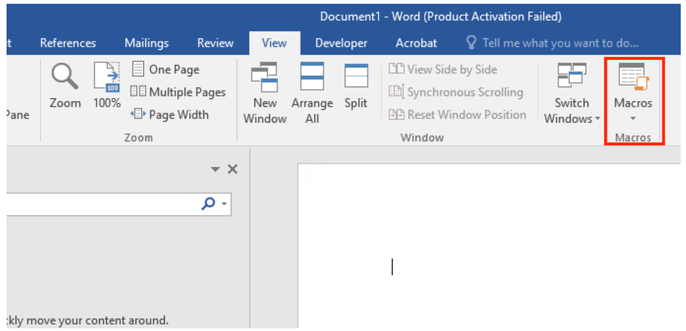
Keywords





- File Type
- .doc, .docm, .dotm, .dot , .rtf are capable to contain

"In Word, you can automate frequently used tasks
by creating and running macros. A macro is a series of commands and instructions that you group together as a single command to accomplish a task automatically."



word document

Keywords

- OLE2 and OOXML (.doc* and .docx)
- OOXML

Has magic bytes of PK, which is Zip file format

```
00000000: 504b 0304 1400 0600 0800 0000 2100 dfa4  PK..........!...
00000010: d26c 5a01 0000 2005 0000 1300 0802 5b43  .lZ... .......[C
00000020: 6f6e 7465 6e74 5f54 7970 6573 5d2e 786d  ontent_Types].xm
00000030: 6c20 a204 0228 a000 0200 0000 0000 0000  l ...(..........
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

- OLE2

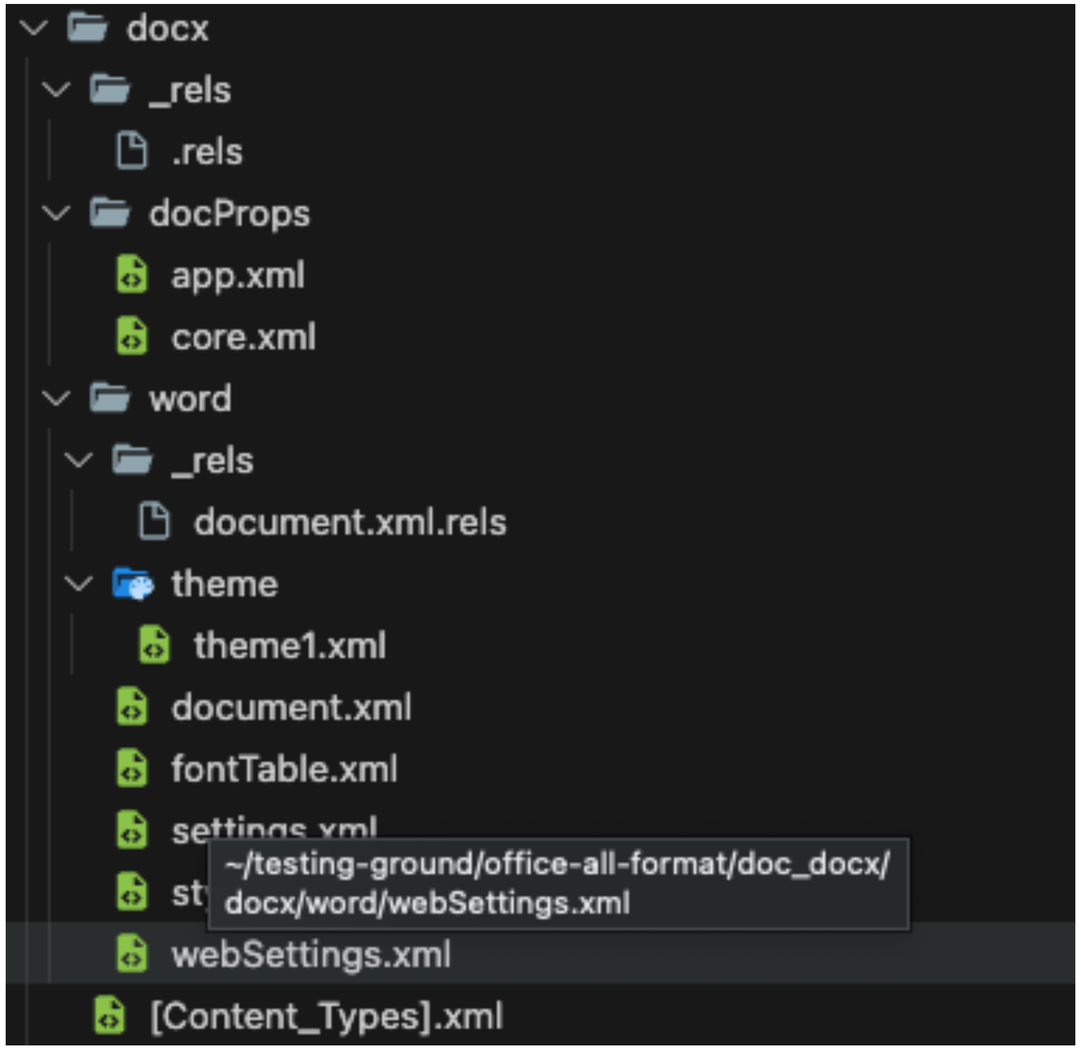
Has bytes of \xD0\xCF\x11\xE0\xA1\xB1\x1A\xE1\x00\x00\x00
Which signifies that it is a word document file
note: format is proprietary so there's no official doc

```
00000000: d0cf 11e0 a1b1 1ae1 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 3e00 0300 feff 0900  ........>.......
00000020: 0600 0000 0000 0000 0000 0000 0100 0000  ................
00000030: 2800 0000 0000 0000 0010 0000 2a00 0000  (...........*...
00000040: 0100 0000 feff ffff 0000 0000 2700 0000  ............'...
00000050: ffff ffff ffff ffff ffff ffff ffff ffff  ................
00000060: ffff ffff ffff ffff ffff ffff ffff ffff  ................
```

word document

File structure

- Upon unzipping .docx

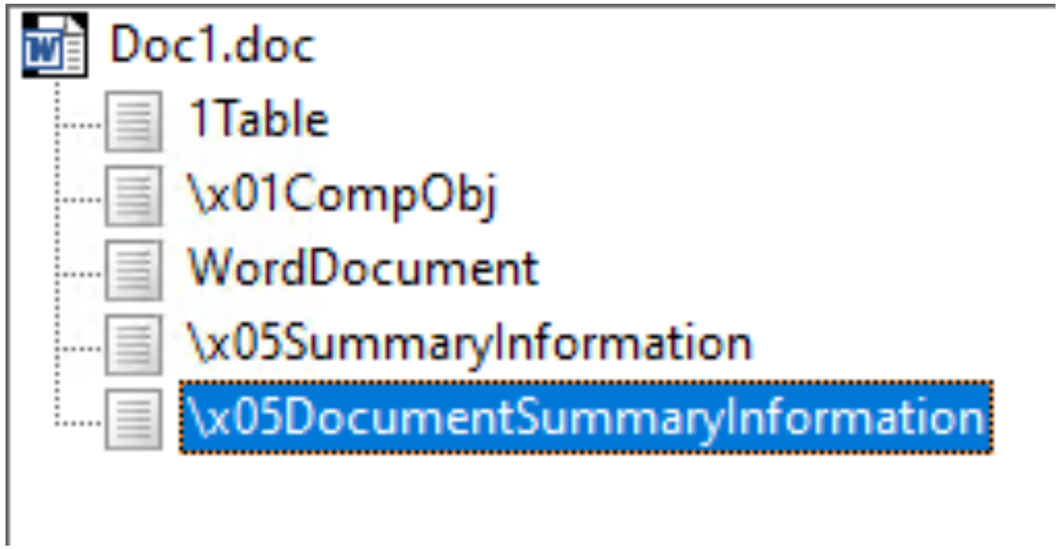


word document

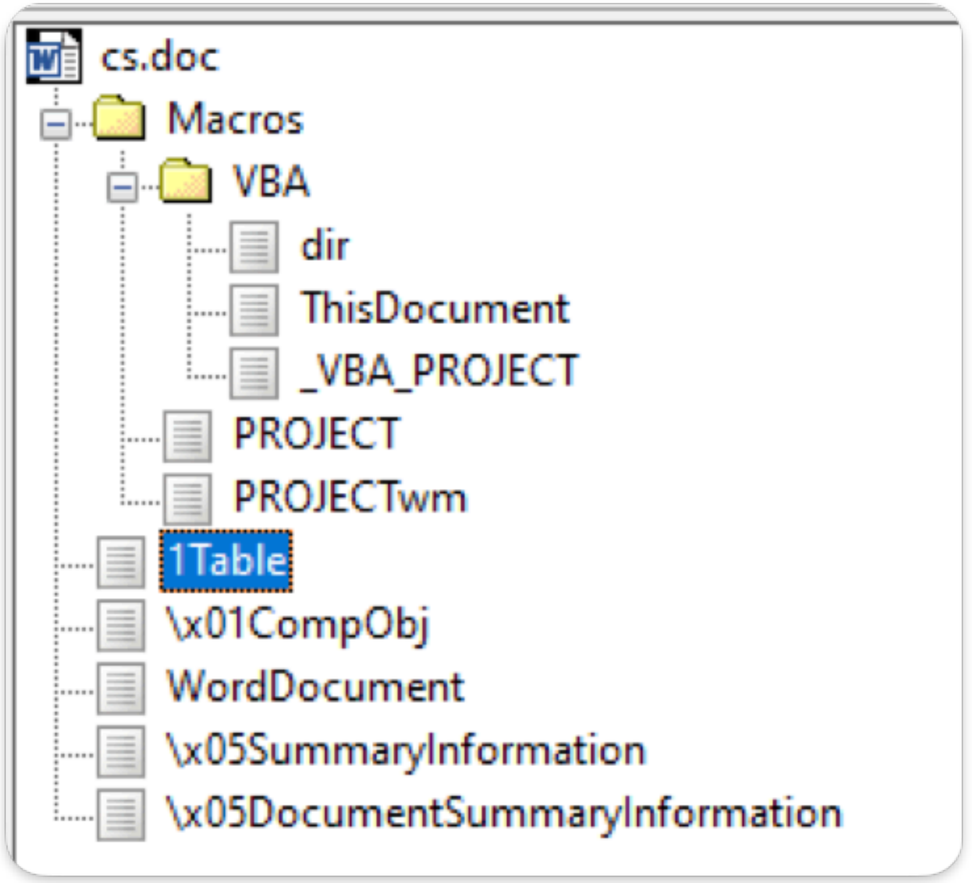
OLE2 file structure

- OLE2 - binary object

Without macro



With macro



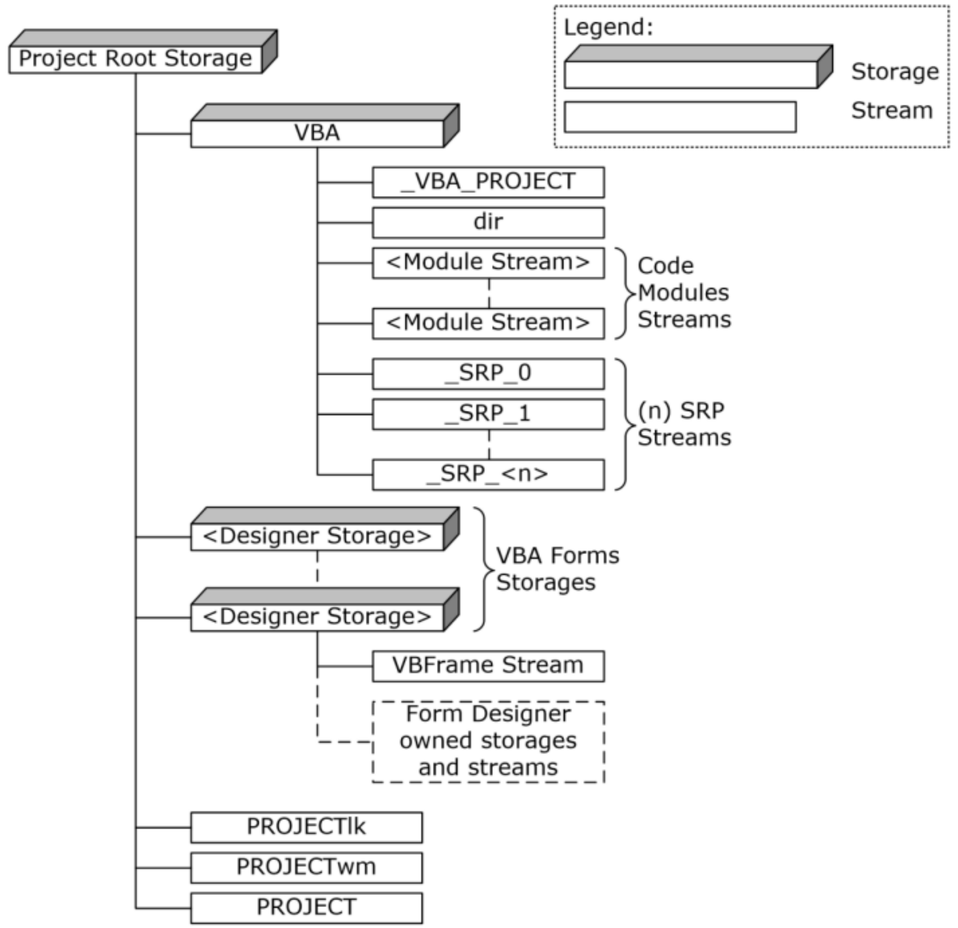
http://www.heaventools.com/flexhex-hex-editor.htm

word document

VBA storage

2.2 File Structure

Specifies a **VBA project** and contained project items. All data is stored in a structured storage as specified in [MS-CFB]. The **storages** and **streams** MUST be organized according to a hierarchy r at the Project Root Storage (section 2.2.1) as depicted in the following figure.



https://msopenspecs.azureedge.net/files/MS-OVBA/%5bMS-OVBA%5d-230221.pdf

word document

VBA storage

- _VBA_PROJECT Stream

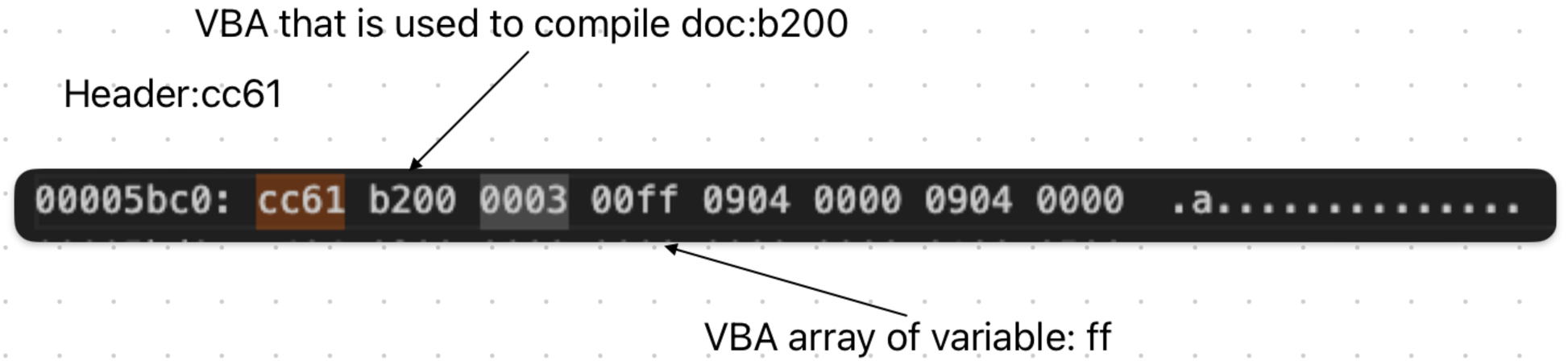
Reserved1 (2 bytes): MUST be 0x61CC. MUST be ignored.

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.

Reserved2 (1 byte): MUST be 0x00. MUST be ignored.

Reserved3 (2 bytes): Undefined. MUST be ignored.

PerformanceCache (variable): An array of bytes that forms an implementation-specific and version-dependent performance cache for the VBA project. The length of **PerformanceCache** MUST be seven bytes less than the size of **_VBA_PROJECT** Stream (section 2.3.4.1). MUST be ignored on read. MUST NOT be present on write.



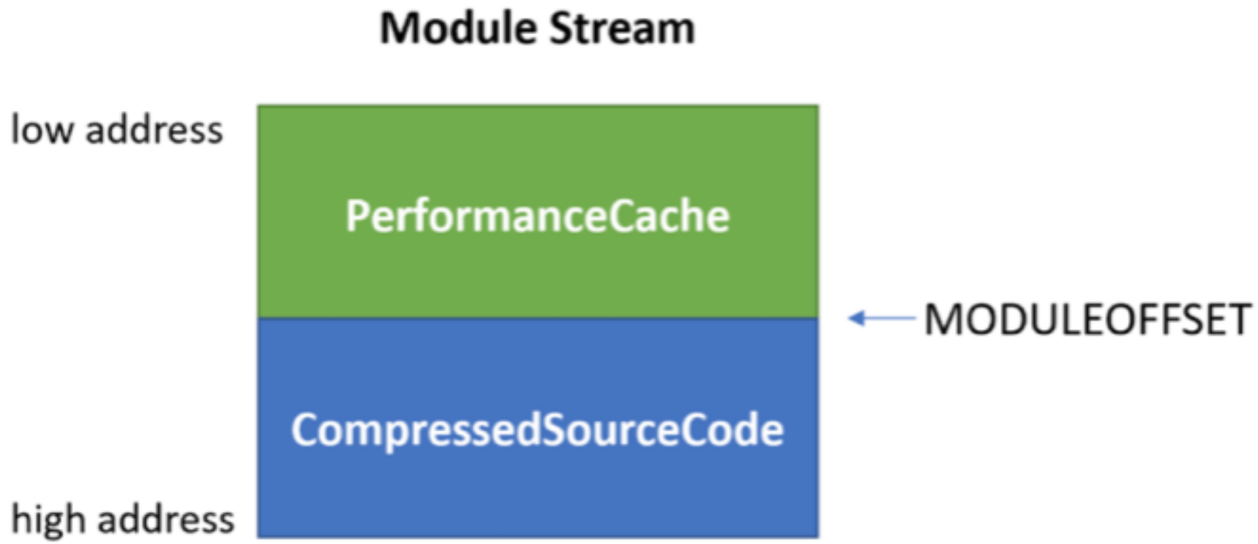
https://www.mandiant.com/resources/blog/purgalicious-vba-macro-obfuscation-with-vba-purging

https://learn.microsoft.com/en-us/openspecs/office_file_formats/ms-ovba/ef7087ac-3974-4452-aab2-7dba2214d239

word document
VBA storage

- PerformanceCache

Upon opening the document with macro, VBA interpreter loads it, as it is a cache, it loads faster.
hence the name of “Performance” cache
Note! Only loads if if MS Office version signified in _VBA_PROJECT matches
with the version of the host MS Office



- CompressedSourceCode

  This is the original source code.

https://www.mandiant.com/resources/blog/purgalicious-vba-macro-obfuscation-with-vba-purging

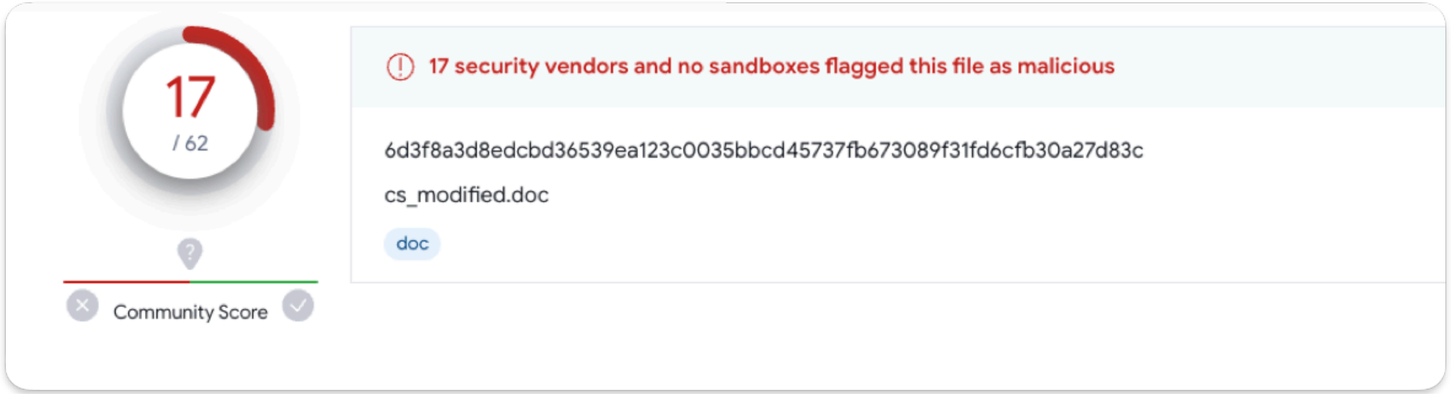
word document

VBA storage: evasion technique

word document

VBA storage: evasion technique

- VBA Stomping: delete CompressedSourceCode, execution relies on P-Cache. it was originally identified by Vesselin Bontchev in 2016.
- Pro:



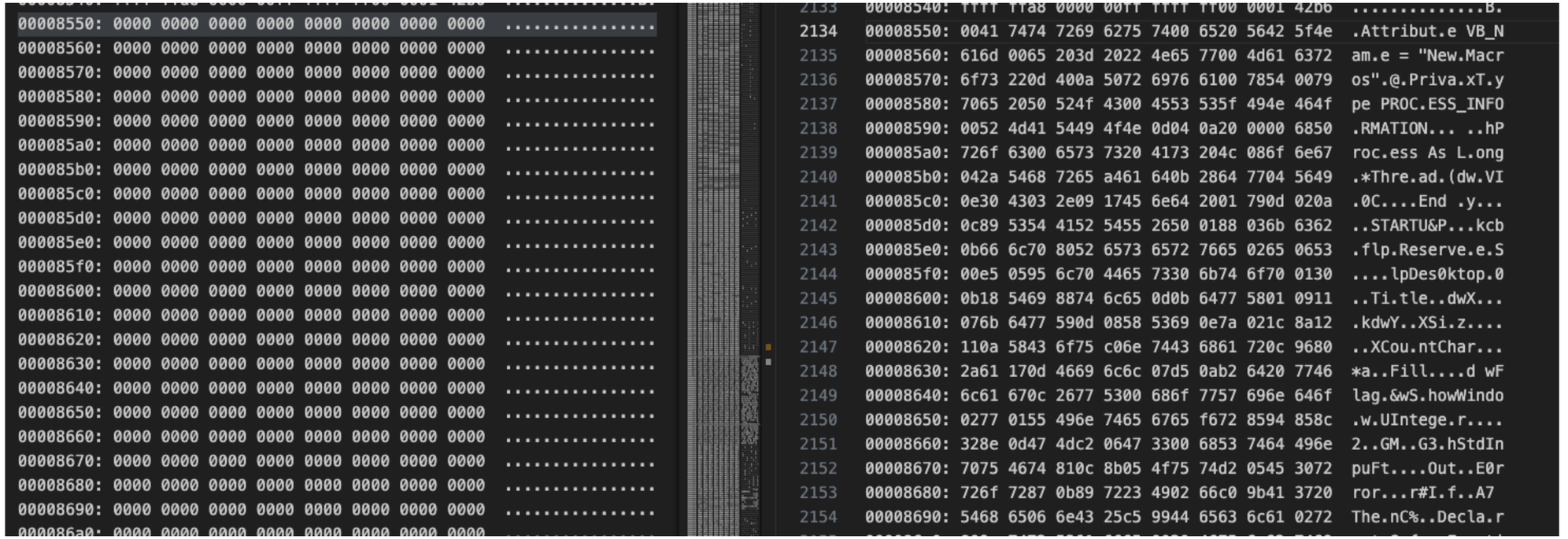
- Cons: the MS office version needs to match with the target, absolutely must be exactly the same.

https://github.com/bontchev/pcodedmp

word document

VBA storage: evasion technique

- Show ~testing-ground/office-all-format/p-code-extract/main.go



https://github.com/decalage2/oletools/wiki/olevba

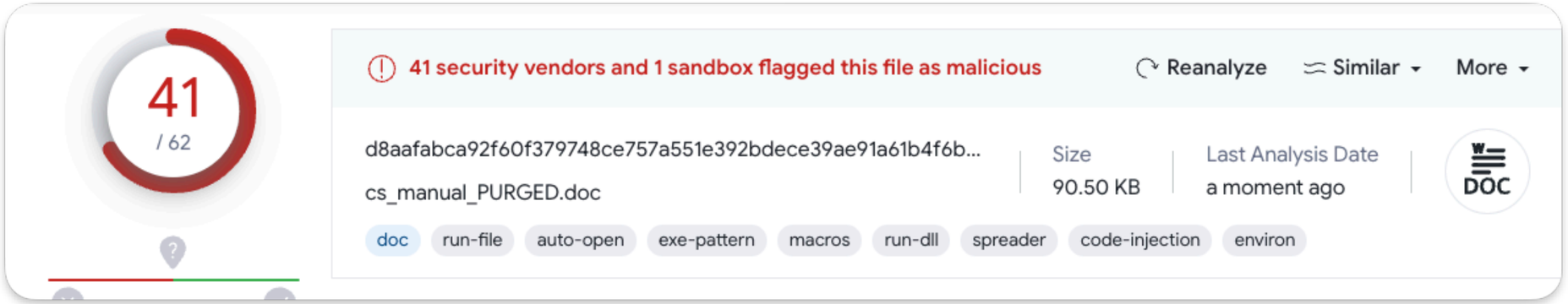
word document

VBA storage: evasion technique

- VBA Purging: delete PerformanceCache, keeps CompressedSourceCode
  Originally identified by Didier Stevens in 2020,Feb.

- Pro: It’s not dependent on the version of software like VBA Stomping

- Cons:



  not effective anymore…

https://blog.nviso.eu/2020/02/25/evidence-of-vba-purging-found-in-malicious-documents/

word document

VBA storage: evasion technique

- https://github.com/mandiant/OfficePurge

https://www.mandiant.com/resources/blog/purgalicious-vba-macro-obfuscation-with-vba-purging

word document

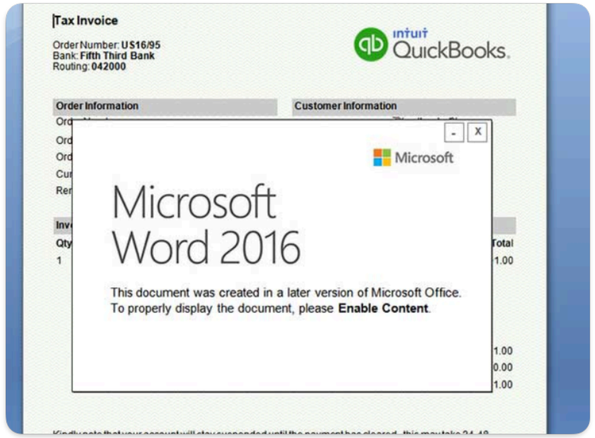
Sample malware analysis

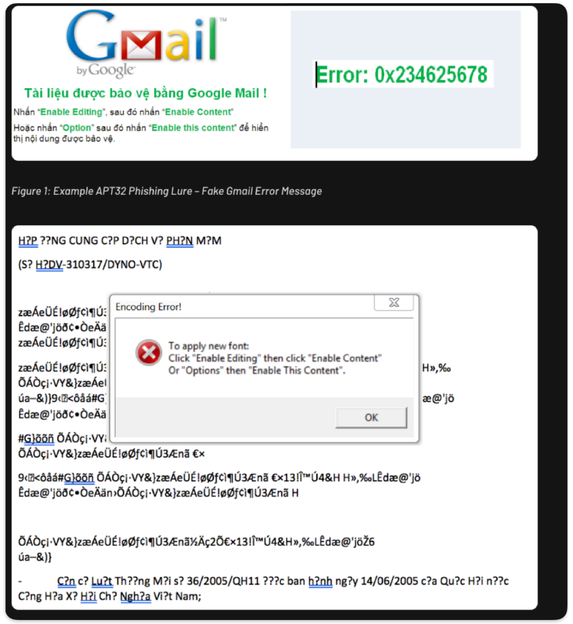
- ~~Show demo~~

word document

Social Engineering

- Says it’s outdated, press to enable macro
- Says it’s encrypted, enter pass to open it.
- Says it needs an update, install add-on (VSTO)





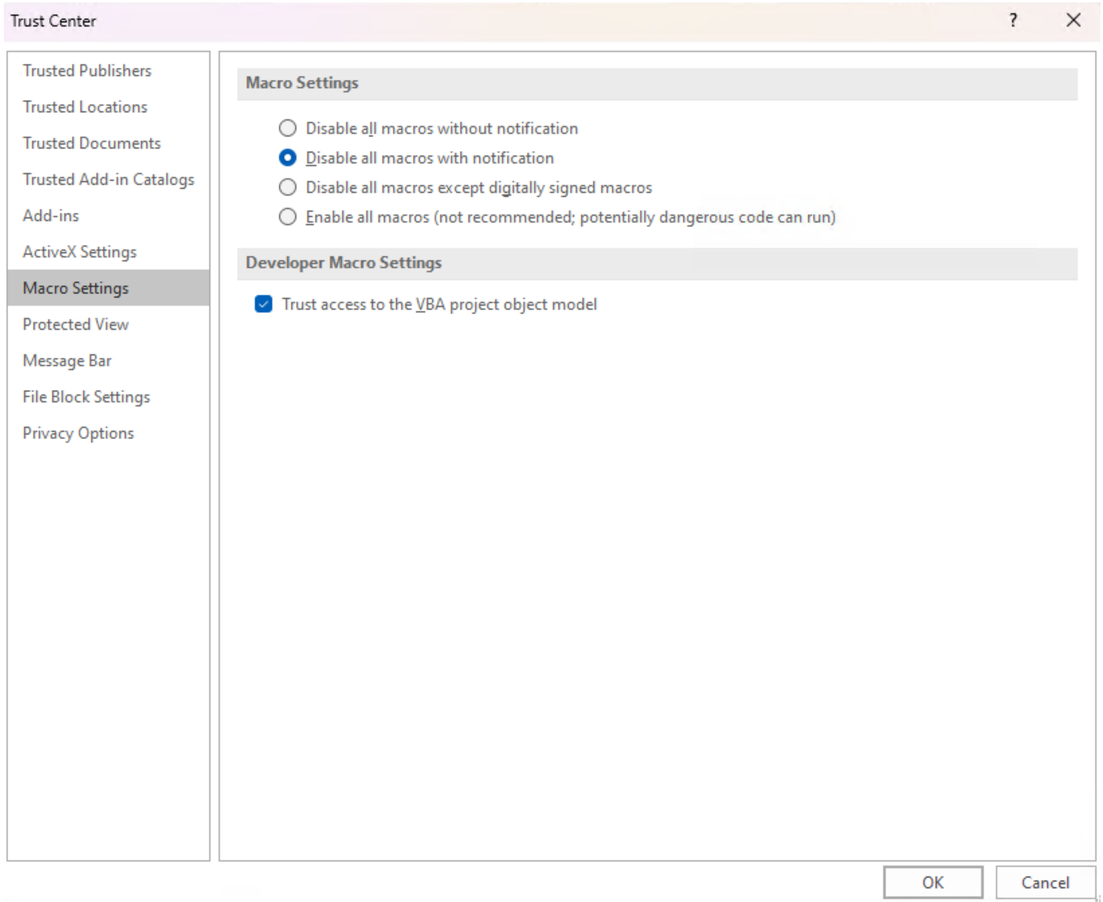
https://web.archive.org/web/20210228215649/https://www.lastline.com/labsblog/party-like-its-1999-comeback-of-vba-malware-downloaders-part-3/

word document

Mitigations

- https://outflank.nl/blog/2023/04/25/so-you-think-you-can-block-macros/

  - Trusted Folder,
  - Trusted Publishers,
  - MotW (mark of the web),
  - Add-ons,
  - Digital Signature,



Microsoft
Visual Basic
for Applications

VBA

What is VBA

- VBA is used to write programs for the Windows operating system and runs as an internal programming language in Microsoft Office

```
Attribute VB_Name = "Snippets"
Sub Variables()

'Variable Types
Dim name As String
Dim age As Integer
Dim height As Double
Dim isMarried As Boolean
Dim dateOfBirth As Date
Dim anyDataHere As Variant

name = "Ajay"
age = 28
height = 175.01
isMarried = False
dateOfBirth = #2/13/1988#

'Arrays
Dim Films(5) As String

Films(0) = "Lord of the Rings"
Films(1) = "Speed"
Films(2) = "Star Wars"
Films(3) = "The Godfather"
Films(4) = "Pulp Fiction"

Dim Films2d(1 To 5, 1 To 2) As String
Dim i As Integer, j As Integer
```

Random Script that found on git

VBA

CobaltStrike default macro payload analysis

- ~/testing-ground/office-all-format/obfuscate-VBA/cs.vba

VBA

Detection Evasion Technique

- The classic 3 steps
- 1.Check filename
- 2.Check Path that document has been saved
- 3.Sleep() and calculate the time of sleep.

https://depthsecurity.com/blog/obfuscating-malicious-macro-enabled-word-docs

VBA

Detection Evasion Technique

- 1.Check filename : AV engine change the name of the file upon emulating the execution. So we make sure that it'll be executed when filename has not been changed.

```
    'perform doc name test cs.doc in this case
    ' <> is not equal
    ' if StrComp() is true returns 0

If StrComp(ActiveDocument.Name, "cs.doc", vbTextCompare) <> 0 Then
    Exit Sub
End If
```

https://learn.microsoft.com/en-us/office/vba/api/word.application.activedocument

VBA

Detection Evasion Technique

- 2.Check Path that document has been saved

```
'perform path test
' check if the path has username C:/*/Ted/*
' AV Engine saves the file somewhere else

Dim regexObject As Object
Set regexObject = CreateObject("VBScript.RegExp")

regexObject.Pattern = Application.UserName

If regexObject.Test(ActiveDocument.path) <> True Then
Exit Sub
End If
```

https://software-solutions-online.com/vba-regex-guide/

VBA
Detection Evasion Technique

- 3.Sleep() and calculate the time of sleep.

If sleep() has not been skipped, then
the deviation of timein and timeout should
match with
seconds of sleep()

timein 00:00
sleep 10 seconds
timeout 00:10

```
' perform sleep test
Dim myTime
Dim second_time

Dim Timein As Date
Dim Timeout As Date
Dim subtime As Variant

myTime = Time
Timein = Date + myTime
Sleep 10000
second_time = Time
Timeout = Date + second_time

' if Datediff true=1, false=0
subtime = DateDiff("s", Timein, Timeout)

If subtime <> 10 Then
    Exit Sub
End If
```

VBA

Sample malware analysis

- Show demo
- ~/testing-ground/malwarebazer-doc/malware-sample.vba

Summary

PDF, Word and VBA

- PDF: files, javascript, embedded links…
- Word: various file formats, object structure,
- VBA: classic. though it gets detected more often now, it’s still dominant and TA will keep using it (I hope).
- TIL: PPAP is very bad, it’s still practiced in many Japanese corporation.


Thank you for listening!