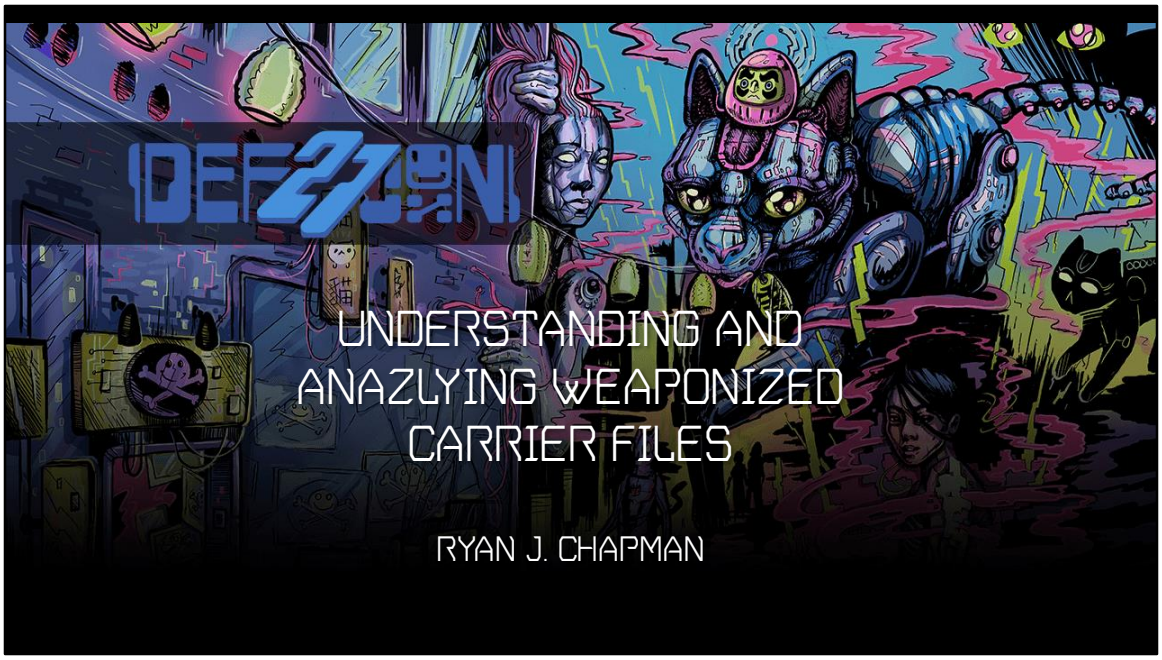

DEF CON
UNDERSTANDING AND
ANAZLYING WEAPONIZED
CARRIER FILES
RYAN J. CHAPMAN



WORKSHOP MATERIALS

- You will need TWO VMs:
  - Windows (malware) VM
    - w/ MS Office + PDFStreamDumper + OffViz
  - REMnux VM
    - Update the bad boy, you'll have all you need
- If you don't have either VM:
  - Come grab a USB in front of the room
  - The copy process will be slow, **so prep**



HEADS-UP ya'll! I will be adding a ton of step-by-step instructions the days leading up to DefCon. As we come closer to the date, you can grab an updated file from my GitHub to find all the fun instructions. Get it!

[additional notes here closer to the big day]



WORKSHOP MATERIALS CONT.

- Virtual handouts here:
  - https://github.com/rj-chap/CFWorkshop
- Malware samples available here:
  - https://incidentresponse.training/cfworkshop-samples.zip
- Copy the **cfworkshop-samples.zip** file to both VMs
  - Remember to disable file sharing after copying
  - Unzip those bad boys



HEADS-UP ya'll! I will be adding a ton of step-by-step instructions the days leading up to DefCon. As we come closer to the date, you can grab an updated file from my GitHub to find all the fun instructions. Get it!

[additional notes here closer to the big day]

Do you have the malware samples? If not, hit the repo!
https://github.com/rj-chap/CFWorkshop

Copy these bad boys to both your Windows malware VM and your REMnux VM. Remember to **disable file sharing after, as we'll be playing with live malware.**

[additional notes here closer to the big day]



CAUTION!! LIVE MALWARE AHEAD!

- We'll be working with live malware

- Careful!
  - **Don't** download malware onto your host
  - **Don't** double-click malware samples
  - **Don't** forget DefCon's **"3-2-1"** rule ☺

- **If you have questions, please ask!**



Stolen from the official DefCon FAQ 4.0:

"At a MINIMUM follow the **3-2-1 Rule** Daily - Three hours of sleep, Two Meals, One Shower. And if you only take One Shower or > a day, Lather on the deodorant. For the ultimate **DefCon** Experience, you need to be an ACTIVE participant."
- You're already doing your part as a participant by taking this workshop!

[additional notes here closer to the big day]



AGENDA

- Environment Setup
- Carrier File Overview
- PDF File Overview
- PDF Analysis
- **Break**
- Office File Overview
- Office Document Analysis



In my previous role, I ran a 5-week SOC baseline training course (many, many times!). In those courses, I dedicated a full day to PDF analysis along with a full day to Office file analysis. Today, we have a total of four (4) hours. As such, we'll want to make the best use of our time.

Proposed Workshop Agenda:
0.5 hr: Intro, VM Setup, and Carrier File Overview
0.5 hr: PDF Overview
1.0 hrs: PDF Analysis
0.5 hr: Bathroom break(s) and buffer
0.5 hr: Office Document Overview
1.0 hr:s Office Document Analysis

Actual Workshop Agenda:
WHO KNOWS?! This is a DefCon workshop yo! I'm sure we'll run into some fun tangents, some random issues, blah blah blah.
Regardless, the instructions within this document will allow you to follow through the intended agenda at your leisure. My goal is to provide step-by-step instructions, so feel free to finish up anything we aren't able to hit within our time limit whenever

you feel like doing so. And don't forget, I'm always around to answer questions online.



ABOUT ME



- Principal IR Consultant
  - All things BLUE!
  - Incident command
  - Host/Network forensics
  - Malware analysis
- Hobbies
  - Retro video games
  - Getting tapped on the mats
  - Hangin' with my Boogie →







[additional notes here closer to the big day]



IT TAKES A VILLAGE

- We have a large class (~90 people)
- But we have some helpers!!
- If you need assistance, **raise your hand**
  - A helper will come to your aid
- If still stuck, skip that section for now
  - I can provide individual assistance:
    - During the break
    - After the workshop in person
    - After the workshop online (hit me @rj_chap!)



[additional notes here closer to the big day]



VM SETUP

- Hopefully, you prepared!
- If not, make sure to:
  - **DISABLE networking**
  - Enable shared folders to copy the samples .zip
    - Then disable!
  - Windows: Disable Windows Defender!
  - REMnux: Make sure the bad boy is up-to-date
- **Snapshot your VMs before you start!**



If you need a Windows VM, please see https://zeltser.com/free-malware-analysis-windows-vm/#step2
\

[additional notes here closer to the big day]


DEF CON
CARRIER FILE OVERVIEW
(A.K.A. MALDOCS)



CARRIER FILE DEFINITION

A **carrier file** (a.k.a. "**MalDoc**") is a document that carries with it a malicious payload.

The most common carrier files are **Office documents** along with **Portable Document Format** (PDF) files.

[additional notes here closer to the big day]



CARRIER FILE INFOZ

- Often attached or linked to within email
- Email attachments are the #1 malware entry vector
  - Email links are also in the top 10
- The median company received over 90% of their detected malware by email
  - Office-based carrier files made up 45% of delivered file types (Verizon, 2019)



The data from this slide is taken from Verizon's 2019 Data Breach Investigations Report, which can be found here: https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

[additional notes here closer to the big day]



MOAR CARRIER FILE INFOZ

- Users often transfer documents via email
- As such, they are more prone to open the bad boys
- Common schemes used:
  - Purchase Orders / Invoices
  - Resumes / CVs
  - Receipts / Bills
  - Contract Proposals
- Wide-net vs. more targeted approach





[additional notes here closer to the big day]

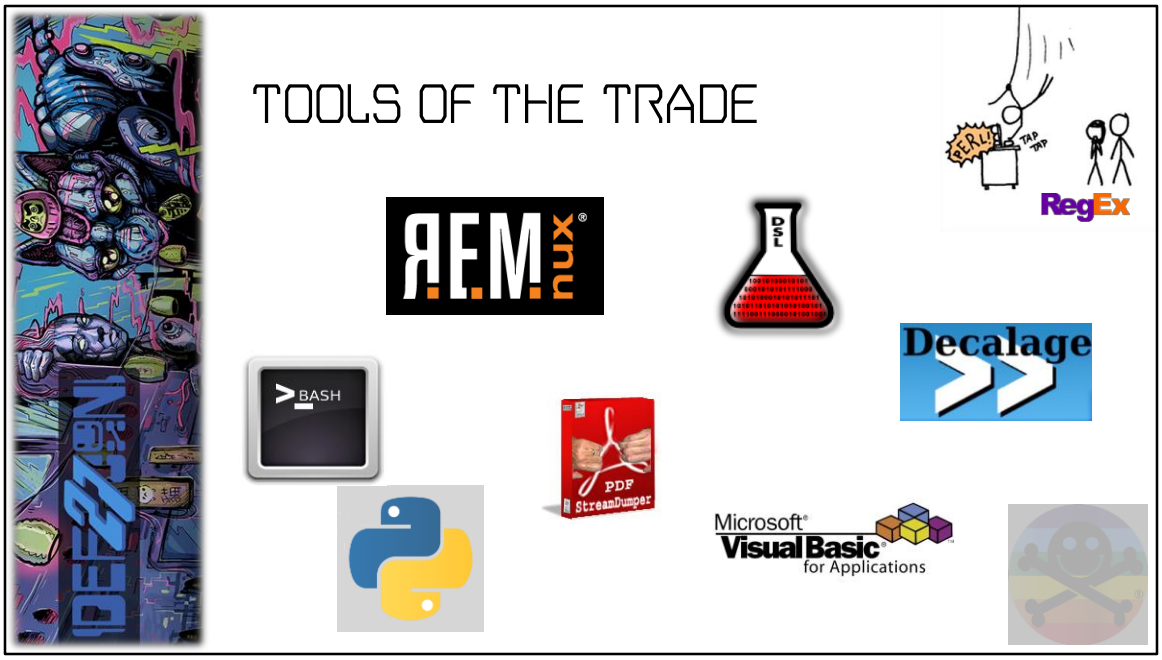


DOWNLOADER VS. DROPPER

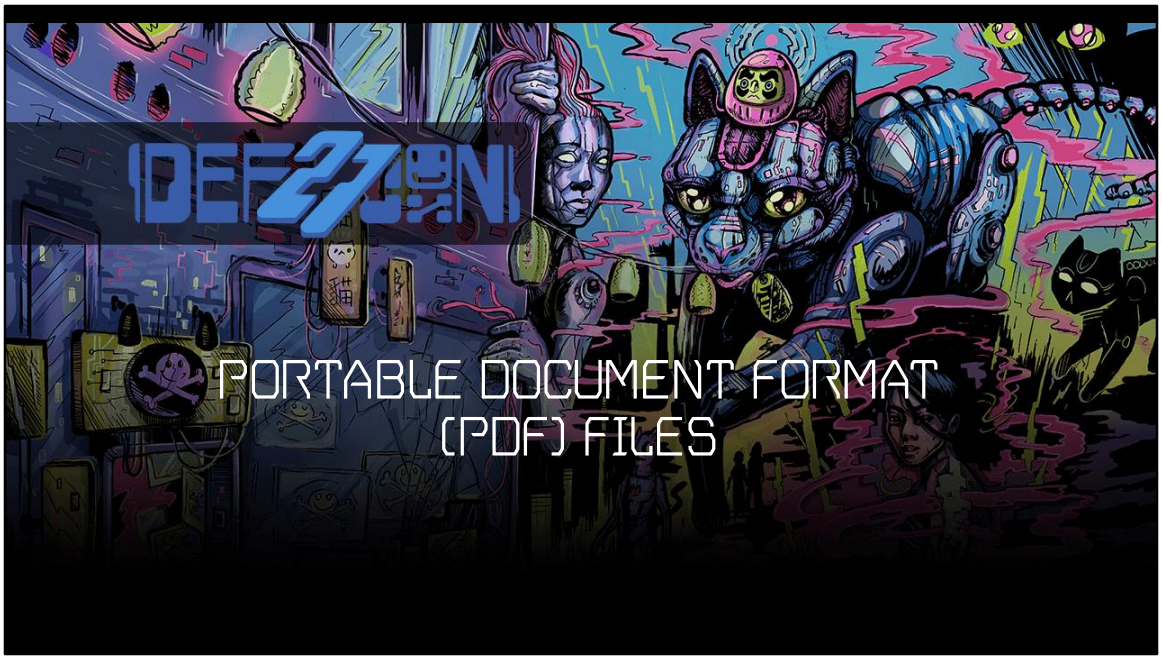
- Downloader
  - Reaches out to external resource via Internet
  - Downloads malware > executes on host
  - When opening, requires Internet access to P0wn
- Dropper
  - Malware contained within document
  - Drops malware onto host -> executes on host
  - *Droppers don't require initial Internet access*



[additional notes here closer to the big day]



[additional notes here closer to the big day]


DEF CON
PORTABLE DOCUMENT FORMAT
(PDF) FILES

ORIGIN AND PURPOSE

- Created by Adobe over 25 years ago
- ISO 32000 as of 2008
- Based on PostScript
- Intent is to include EVERYTHING needed to render a document within the document itself
  - Event fonts are embedded
  - Hence: Portability!





When you need peer-reviewed references, ALWAYS go with Wikipedia (lol):
https://en.wikipedia.org/wiki/PDF#History_and_standardization



PDF WEAPONIZATION

- Simple links (e.g. URI objects)
- Scripting
  - JavaScript (JS) the most common
  - Other script types can be executed
- Exploits
  - CVEs son!
  - Often target JS libraries provided by Adobe
- Embedded file (relies on scripting to exec)



[additional notes here closer to the big day]

https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/js_api_reference.pdf

https://www.malware-traffic-analysis.net/2018/09/04/index.html

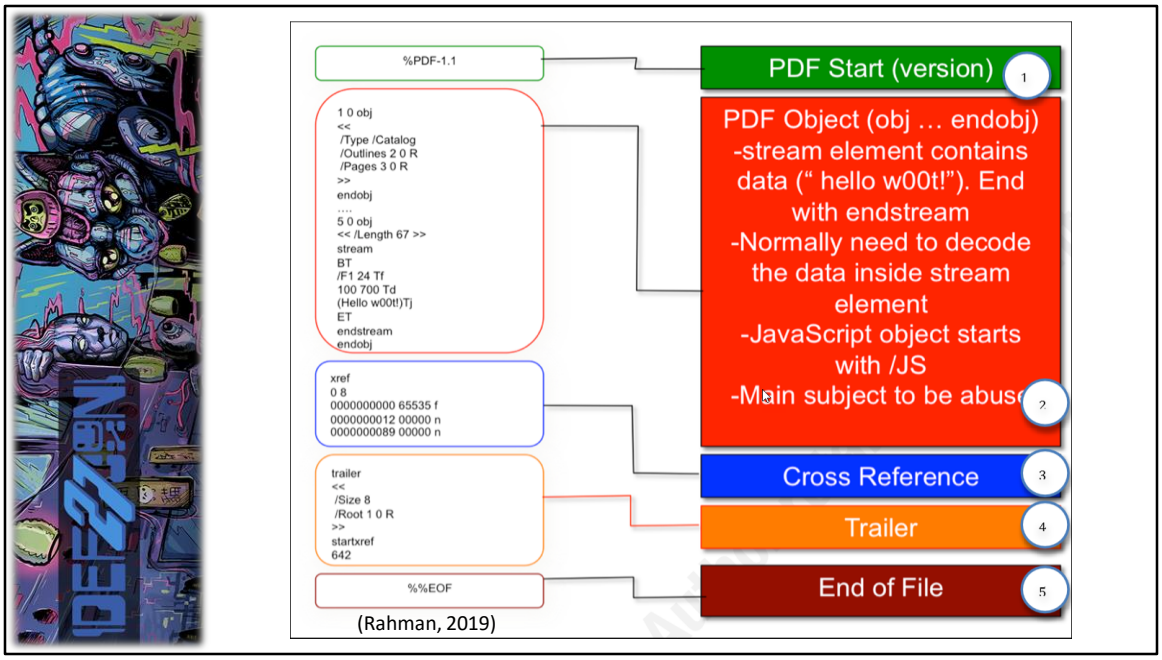


PDF STRUCTURE

- Header
  - Includes PDF specification # [%PDF-X.X])
- Objects
  - Root
  - Catalog
  - Metadata
- Xref
- Trailer
- EOF
- Order can be random



Link to Adobe format guide
[additional notes here closer to the big day]



(Rahman, 2019)

[additional notes here closer to the big day]



PDF OBJECTS

- Example object on next slide!

- Reference # along with revision #
- Types
- Stream objects
  - Filters!

[additional notes here closer to the big day]



EXAMPLE OBJECT

```
<<
     /Type/Catalog/Pages 47 0 R/Lang(en-US)/MarkInfo
     <<
          /Marked true
     >>
     /Metadata 50 0 R/StructTreeRoot 28 0 R/ViewerPreferences
     <<
          /DisplayDocTitle true
     >>

>>
```





PDF STREAM OBJECTS

```
5 0 obj
<< /Length 42 >>
stream
BT /F1 24 Tf 100 700 Td (Hello World)Tj ET
endstream
endobj
```

Indirect Object ID

Dictionary

Stream

Byte Sequence

(Stevens, 2008)



[additional notes here closer to the big day]



PDF STREAM FILTERS

```
5 0 obj
<<
  /Length 55
  /Filter /ASCII85Decode
>>
stream
6<#'\7PQ#@1a#b0+>GQ(+?(u.+B2ko-qIocCi:FtDfTZ).9(%)78s~>
endstream
endobj
```

Filter

(Stevens, 2008)



[additional notes here closer to the big day]



PDF STREAM FILTERS

- ASCIIHexDecode
- ASCII85Decode
- LZWDecode
- FlateDecode
- RunLengthDecode
- CCITTFaxDecode
- JBIG2Decode
- DCTDecode
- JPXDecode
- Crypt

[notes here pre-DefCon – wait ‘fer me!]



PDF KEYWORDS

/Page
/Encrypt
/ObjStm
/JS
/JavaScript
/AA
/OpenAction
/JBIG2Decode
/RichMedia
/Launch
/XFA
/Annotation



[additional notes here closer to the big day]

Shamelessly copied from https://blog.didierstevens.com/programs/pdf-tools/:

Almost every PDF documents will contain the first 7 words (obj through startxref), and to a lesser extent stream and endstream. I've found a couple of PDF documents without xref or trailer, but these are rare (BTW, this is not an indication of a malicious PDF document).
/Page gives an indication of the number of pages in the PDF document. Most malicious PDF document have only one page.
/Encrypt indicates that the PDF document has DRM or needs a password to be read.
/ObjStm counts the number of object streams. An object stream is a stream object that can contain other objects, and can therefor be used to obfuscate objects (by using different filters).
/JS and /JavaScript indicate that the PDF document contains JavaScript. Almost all malicious PDF documents that I've found in the wild contain JavaScript (to exploit a JavaScript vulnerability and/or to execute a heap spray). Of course, you can also find JavaScript in PDF documents without malicious intend.
/AA and /OpenAction indicate an automatic action to be performed when the

page/document is viewed. All malicious PDF documents with JavaScript I've seen in the wild had an automatic action to launch the JavaScript without user interaction. The combination of automatic action and JavaScript makes a PDF document very suspicious.
/JBIG2Decode indicates if the PDF document uses JBIG2 compression. This is not necessarily and indication of a malicious PDF document, but requires further investigation.
/RichMedia is for embedded Flash.
/Launch counts launch actions.
/XFA is for XML Forms Architecture.
A number that appears between parentheses after the counter represents the number of obfuscated occurrences. For example, /JBIG2Decode 1(1) tells you that the PDF document contains the name /JBIG2Decode and that it was obfuscated (using hexcodes, e.g. /JBIG#32Decode).



PDF FILES IN A HEX EDITOR

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	25	50	44	46	2D	31	2E	37	0D	25	E2	E3	CF	D3	0D	0A	%PDF-1.7.%âãÏÓ..
00000010	31	20	30	20	6F	62	6A	0D	3C	3C	2F	54	79	70	65	2F	1 0 obj.<</Type/
00000020	50	61	67	65	2F	50	61	72	65	6E	74	20	34	37	20	30	Page/Parent 47 0
00000030	20	52	2F	43	6F	6E	74	65	6E	74	73	20	32	37	20	30	R/Contents 27 0
00000040	20	52	2F	4D	65	64	69	61	42	6F	78	5B	30	20	30	20	R/MediaBox[0 0
00000050	35	39	35	2E	33	32	30	30	31	20	38	34	31	2E	39	31	595.32001 841.91
00000060	39	39	38	5D	2F	41	6E	6E	6F	74	73	5B	32	20	30	20	998]/Annots[2 0
00000070	52	20	34	20	30	20	52	20	36	20	30	20	52	20	38	20	R 4 0 R 6 0 R 8
00000080	30	20	52	20	31	30	20	30	20	52	20	31	32	20	30	20	0 R 10 0 R 12 0
00000090	52	5D	2F	47	72	6F	75	70	20	31	34	20	30	20	52	2F	R]/Group 14 0 R/
000000A0	53	74	72	75	63	74	50	61	72	65	6E	74	73	20	30	2F	StructParents 0/
000000B0	54	61	62	73	2F	53	2F	52	65	73	6F	75	72	63	65	73	Tabs/S/Resources
000000C0	3C	3C	2F	45	78	74	47	53	74	61	74	65	3C	3C	2F	47	<</ExtGState<</G
000000D0	53	37	20	31	35	20	30	20	52	2F	47	53	38	20	31	36	S7 15 0 R/GS8 16
000000E0	20	30	20	52	3E	3E	2F	46	6F	6E	74	3C	3C	2F	46	31	0 R>>/Font<</F1
000000F0	20	31	37	20	30	20	52	2F	46	32	20	31	39	20	30	20	17 0 R/F2 19 0
00000100	52	3E	3E	2F	58	4F	62	6A	65	63	74	3C	3C	2F	49	6D	R>>/XObject<</Im
00000110	61	67	65	31	32	20	32	32	20	30	20	52	2F	49	6D	61	age12 22 0 R/Ima
00000120	67	65	31	34	20	32	34	20	30	20	52	2F	49	6D	61	67	ge14 24 0 R/Imag
00000130	65	31	35	20	32	35	20	30	20	52	3E	3E	3E	3E	3E	3E	e15 25 0 R>>>>>>
00000140	0D	65	6E	64	6F	62	6A	0D	32	20	30	20	6F	62	6A	0D	.endobj.2 0 obj.
00000150	3C	3C	2F	54	79	70	65	2F	41	6E	6E	6F	74	2F	53	75	<</Type/Annot/Su
00000160	62	74	79	70	65	2F	4C	69	6E	6B	2F	52	65	63	74	5B	btype/Link/Rect[
00000170	32	30	33	2E	35	39	20	33	34	30	2E	30	37	30	30	31	203.59 340.07001
00000180	20	33	37	30	2E	37	36	30	30	31	20	33	39	36	2E	37	370.76001 396.7



[additional notes here closer to the big day]



PDF ANALYSIS TOOLS

- pdf-tools
  - pdfid
  - pdf-parser
- peepdf
- PDFStreamDumper

- Many more exist, but we'll be using these



[additional notes here closer to the big day]

Additional tools exist, such as those listed here:
https://www.decalage.info/fr/file_formats_security/pdf



SIMPLE URI

- **301d285ee0d3151d67f80bf4709c0aae**
- And we pull it apart…
- [step-by-step notes closer to DefCon!]



[additional notes here closer to the big day]

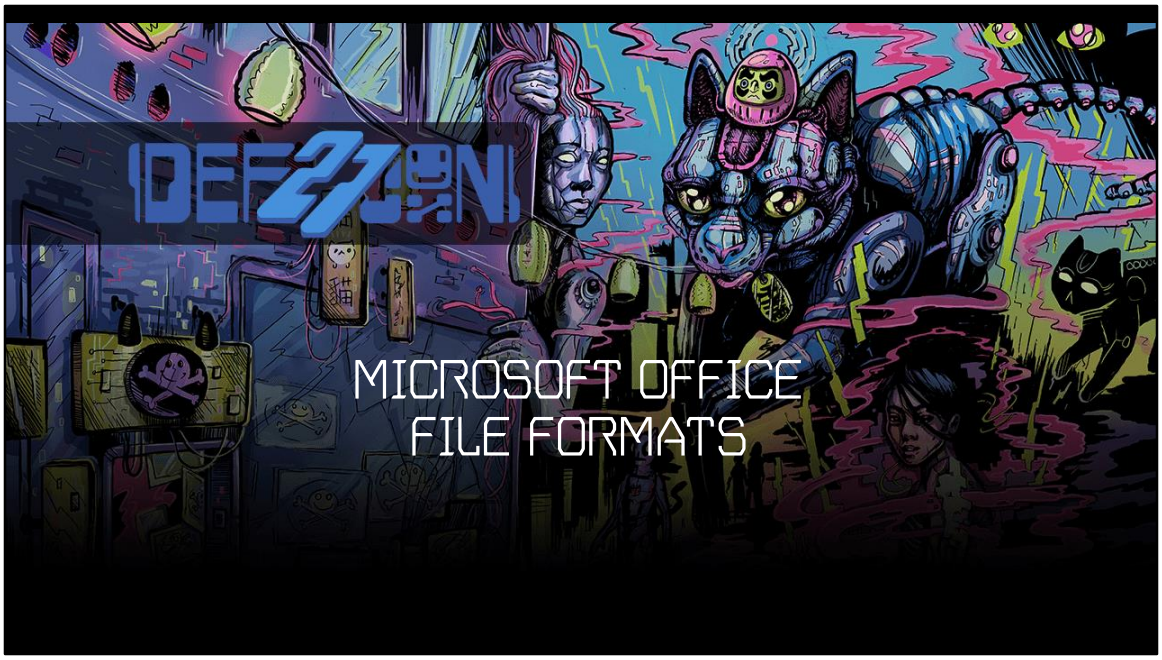


EXPLOIT W / SHELLCODE

- **F281c01164c0120d12e996a66b05542f7065116c01 9cf546e44412164a28890e**
- And we pull it apart…
- [step-by-step notes closer to DefCon!]



[additional notes here closer to the big day]


DEF CON
MICROSOFT OFFICE
FILE FORMATS



OFFICE FILE STRUCTURES

- Object Linking and Embedding Compound File
  - **Office 97-2007**
  - a.k.a. “OLE CF”
  - It’s a darn file system!
- DOCX
  - **Office 2007+**
  - It’s just a ZIP file with an XML structure!
- Rich Text Format (RTF)
  - Can embed raw OLE documents



http://www.forensicswiki.org/wiki/Word_Document_(DOC)
http://www.forensicswiki.org/wiki/OLE_Compound_File
http://www.forensicswiki.org/wiki/Word_Document_(DOCX)

[additional notes here closer to the big day]



OFFICE FILE WEAPONIZATION

- Often use Visual Basic for Applications (VBA)
  - Macros use VBA
- RTF files obfuscating raw hex data
- Equation Editor exploits
- VBA stomping
  - Not common due to compatibility requirements
  - But ridiculously awesome!



[additional notes here closer to the big day]



OFFICE ANALYSIS TOOLS

- OfficeMalScanner
- oletools
- oledump
- MS VBA Editor
  - Included w/ MS Office
  - One of the best tools for analysis (Thanks MS!)
- OffViz



Visual Basic for Applications (VBA) Overview
https://en.wikipedia.org/wiki/Visual_Basic_for_Applications Whether through macros or just straight scripting, VBA is the weaponization method of choice for Office files Walk folks through enabling the Developer ribbon in Office
https://support.office.com/en-us/article/show-the-developer-tab-e1192344-5e56-4d45-931b-e5fd9bea2d45 Keyboard shortcut: Alt + F11
OfficeMalScanner
http://www.reconstructer.org/code.html open, review Use inflate on previous 2007+ files, then use info on the .bin files to find VBA
oletools
https://www.decalage.info/python/oletools
https://github.com/decalage2/oletools/wiki/Install Pay special attention to olevba
https://github.com/decalage2/oletools/wiki/olevba Use me:
https://github.com/decalage2/oletools/blob/master/cheatsheet/oletools_cheatsheet.pdf
oledump
https://blog.didierstevens.com/programs/oledump-py/ Not part of training, but fun to know about (this is the tool Kelcey used in the incident overview video):
OffVis

https://blogs.technet.microsoft.com/srd/2009/09/14/offvis-updated-office-file-format-training-video-created/ https://www.aldeid.com/wiki/OffVis

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	D0	CF	11	E0	A1	B1	1A	E1	00	00	00	00	00	00	00	00	ÐÏ.à¡±.á........
00000010	00	00	00	00	00	00	00	00	3E	00	03	00	FE	FF	09	00	........>...þÿ..
00000020	06	00	00	00	00	00	00	00	00	00	00	00	01	00	00	00	....
00000030	30	00	00	00	00	00	00	00	00	10	00	00	32	00	00	00	0...........2...
00000040	01	00	00	00	FE	FF	FF	FF	00	00	00	00	2F	00	00	00	....þÿÿÿ..../...
00000050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000170	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000190	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000200	EC	A5	C1	00	6B	00	09	04	00	00	F0	12	BF	00	00	00	ì¥Á.k.....ð.¿...

OLE CF in HEX EDITOR



[additional notes here closer to the big day]



DOCX IN A HEX EDITOR

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	50	4B	03	04	14	00	06	00	08	00	00	00	21	00	DF	A4	PK.....!.ß¤
00000010	D2	6C	5A	01	00	00	20	05	00	00	13	00	08	02	5B	43	ÒlZ... .....[C
00000020	6F	6E	74	65	6E	74	5F	54	79	70	65	73	5D	2E	78	6D	ontent_Types].xm
00000030	6C	20	A2	04	02	28	A0	00	02	00	00	00	00	00	00	00	l ¢..(........
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....



[additional notes here closer to the big day]



WORD SAMPLE 1 OLE W / VBA

- **62a83453d87810a25cf7ef7952f8cbed211064f2de4 e7e39ddcff42c8c855759**
- And we pull it apart…
- [step-by-step notes closer to DefCon!]

[additional notes here closer to the big day]



WORD SAMPLE 2 OLE W / VBA

- **a0cf68c10d1d9ce0e73208e6e6bad32b2405d2c74c662dc7b20f9b567c4dda64**
- And we pull it apart…
- [step-by-step notes closer to DefCon!]



[additional notes here closer to the big day]



RTF SAMPLE 1 EMBEDDED OLE

- **0de3f4380b642e59d0cde5570ed13bfc727000b94a034ce10e1f87bfac3fac79**
- And we pull it apart…
- [step-by-step notes closer to DefCon!]

[additional notes here closer to the big day]



QUESTIONS?

- If we have time:
  - *Ask me anything!*
- If we don't have time:
  - Feel free to contact me whenever!
  - Twitter: **@rj_chap**

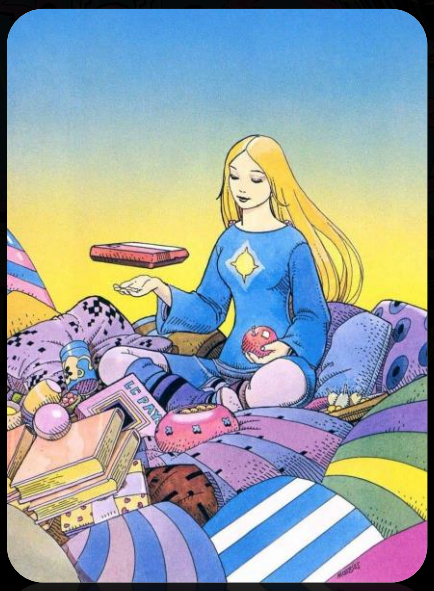




Ryan Chapman

@rj_chap

DFIR/Malware analyst. Trainer/Speaker. PluralSight author. Comedy & BJJ chump. R2V0IGl0IGd1cmwhIQ==





ADDITIONAL MATERIALS

- https://zeltser.com/media/docs/analyzing-malicious-document-files.pdf
- https://www.decalage.info/fr/file_formats_security/pdf



REFERENCES

Rahman, M. A. (2019). Getting owned by malicious pdf – analysis. Retrieved from https://www.sans.org/reading-room/whitepapers/malicious/paper/33443

Stevens, D. (2008). PDF stream objects. Retrieved from https://blog.didierstevens.com/2008/05/19/pdf-stream-objects/

Verizon. (2019). 2019 data breach investigations report. Retrieved from https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

Weyne, Felix. (2016). Image entitled “maldocs_header.png”. Retrieved from https://www.uperesia.com/analyzing-malicious-office-documents