

PDF



Myths  
VS  
FACTS

Oxford University 15th July 2015 Ange Albertini

# ANGE ALBERTINI

reverse engineering

VISUAL DOCUMENTATION

[@angealbertini](https://www.instagram.com/angealbertini)

[ange@corkami.com](mailto:ange@corkami.com)

<http://www.corkami.com>



# Disclaimer:

# this is my first digipres event

I come here with a very different perspective:

I might sound pessimistic (or provocative/killjoy)...  
Give me hope, give me peace on earth ;)

I might be entirely **wrong** - *please* let me know!

# **I used to think: “PDF is perfect”**

Complex documents,  
yet uniform rendering on any system  
(no wonder it's omnipresent)  
⇒ I believed the myth...

# **Professionally, I analyse PDFs**

Malware, security

(It originally happened by “accident”,  
but I’ve been doing it since then...)

# PORTABLE DOCUMENT FORMAT

## HEADER

```

%PDF-1.1          SIGNATURE & VERSION INFORMATION
<< [D VALUE]* >>
1 0 obj
<<
  /Pages 2 0 R
endobj
  
```

DICTIONARY  
 OBJECT REFERENCE:  
 IDENTIFIER (WITH /)

## BODY

```

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
4 0 obj
<< /Length 50 >>
stream
BT
  /F1 110 Tf
  10 400 Td
  (Hello World!)Tj
ET
endstream
endobj
  
```

ARRAY  
 STREAM PARAMETERS:  
 LENGTH, COMPRESSION...  
 BEGIN TEXT  
 FONT F1 (ARIAL) SET TO SIZE 110  
 MOVE TO COORDINATE 10, 400  
 OUTPUT TEXT 'HELLO WORLD!'  
 END TEXT

## XREF TABLE

```

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

CROSS REFERENCE  
 CROSS REFERENCES  
 5 OBJECTS, STARTING AT INDEX 0  
 (STANDARD FIRST EMPTY OBJECT 0  
 OFFSET TO OBJECT 1, REV 0  
 TO OBJECT 2...)

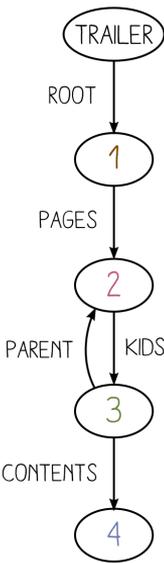
## TRAILER

```

trailer
<< /Root 1 0 R
>>
startxref
413
%%EOF
  
```

## PARSING

%PDF-1. ? IS CHECKED  
 startxref POINTS TO XREF  
 xref POINTS TO EACH OBJECT  
 trailer IS PARSED  
 REFERENCES ARE FOLLOWED  
 DOCUMENT IS RENDERED



# PDF operators

## PATH

construction  
 x y m move points (begin subpath)  
 x y l line  
 x1 y1 x2 y2 x3 y3 c cubic bezier  
 x2 y2 x3 y3 v cubic bezier  
 x1 y1 x3 y3 y cubic bezier  
 h close subpath  
 x y width height re rectangle

## painting

S stroke the path  
 s close and stroke (h S)  
 f fill  
 F fill (deprecated)  
 f\* even-odd fill  
 B fill & stroke  
 B\* even-odd fill & stroke  
 b close, fill & stroke  
 b\* close, even-odd fill & stroke  
 n end path (no filling/stroking)



## clipping

W intersect clipping path  
 W\* even-odd intersect clipping path

## GRAPHICS STATE

general  
 lineWidth w set line width  
 lineCap J set line cap  
 lineJoin j set line join  
 miterLimit M set miter limit  
 dashArray dashPhase d set line dash  
 intent ri set color rendering intent  
 flatness i set flatness tolerance  
 dictName gs set graphics state

special  
 q save current state  
 Q restore current state  
 a b c d e f cm modify current transformation matrix

## TEXT

object  
 BT begin text  
 ET end text

position  
 x y Td move to next line  
 x y TD move to next line + set leading  
 a b c d e f Tm text & text line matrixes  
 T\* move to next line + reset leading

showing  
 string TJ show string  
 ' string ' move to next line + show string  
 a\_w a\_c string " move to next line + show string + set word & character spacing  
 array TJ show string(s) with glyph positioning

# I created fact sheets about PDF

I gave  
presentations  
about PDF

Advanced

# PDF TRICKS

An overview of  
potential leaks via PDF

PDF  
Secrets

hiding & revealing secrets in PDF documents

# Personally, I *play* with PDF

proactive, and *fun*

PDF is more than a file format.

PDF is an abuse playground!

The logo for 'PDF disaster' features a stylized icon of a document with a red square at the top left corner, followed by the text 'PDF' in a large, bold, black sans-serif font, and 'disaster' in a smaller, bold, black sans-serif font below it.

**PDF**  
disaster

**Yes, I write PDFs by hand...**

[...and I open them in hex editors]

```
%PDF-1.
```

```
1 0 obj
<< /Kids [ <<
  /Parent 1 0 R
  /Resources <<>>
  /Contents 2 0 R
  >> ]
>>
```

```
2 0 obj
<<>>
stream
BT
  /F1 110 Tf
  10 400 Td
  (Hello World!) Tj
ET
endstream
endobj
```

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

...like this one

%PDF-1.

truncated signature

1 0 obj

<< /Kids [ <<

    /Parent 1 0 R

    /Resources <<>

    /Contents 2 0 R

>>]

>>

missing parent /Type  
/Kids should be indirect  
missing /Font  
missing kid /Type  
missing /Count

missing endobj

2 0 obj

<<>

stream

BT

    /F1 110 Tf

    10 400 Td

    (Hello World!) Tj

ET

endstream

endobj

missing /Length

missing xref

trailer <<

    /Root << /Pages 1 0 R >>

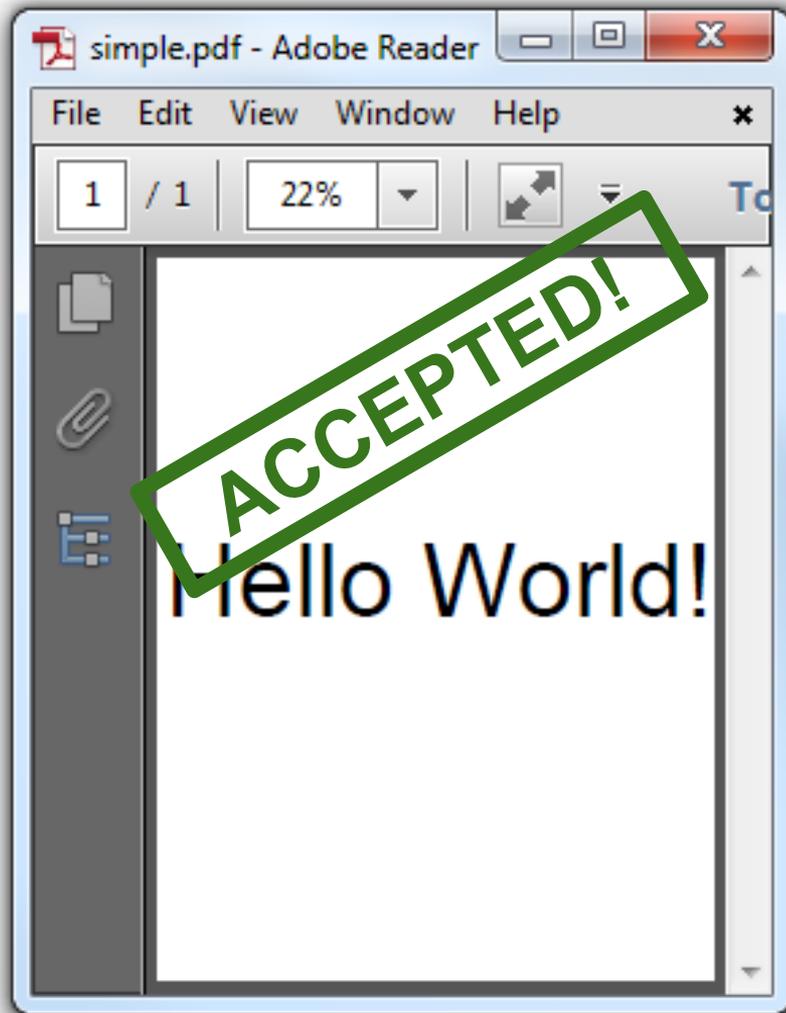
>>

/Root should be indirect, missing /Size, missing root /Type  
missing startxref, %%EOF



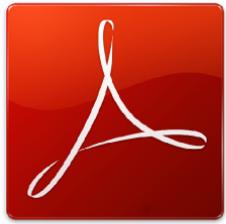
It's *not*  
standard...

...but it works  
exactly as planned!  
(without any reported error)



# Binary art

PDF + creativity = ... ?



PDF slides

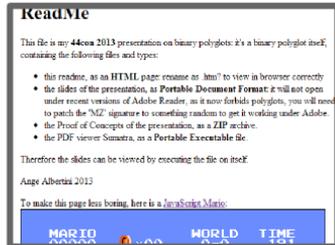


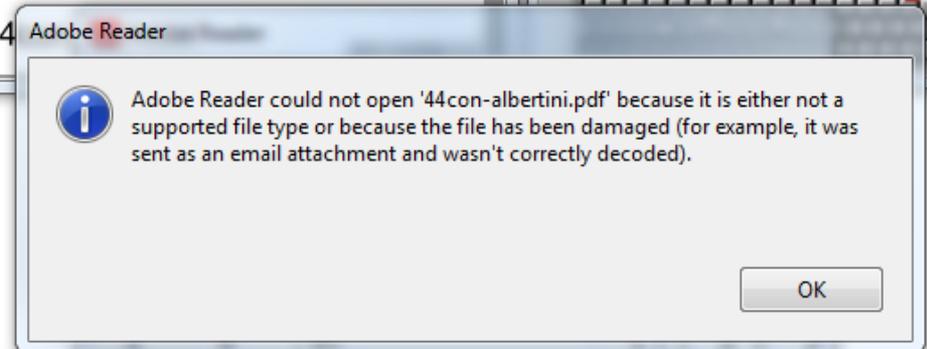
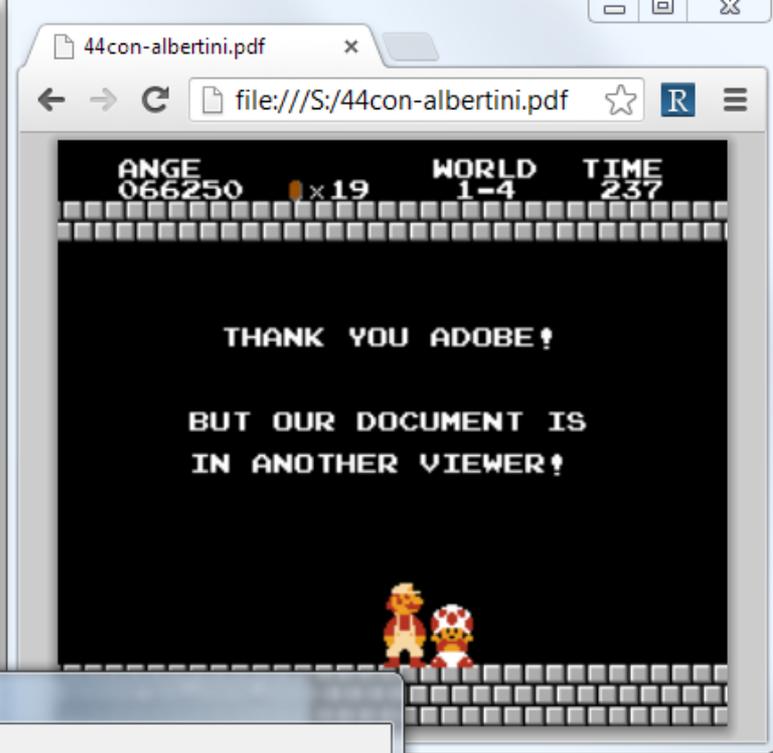
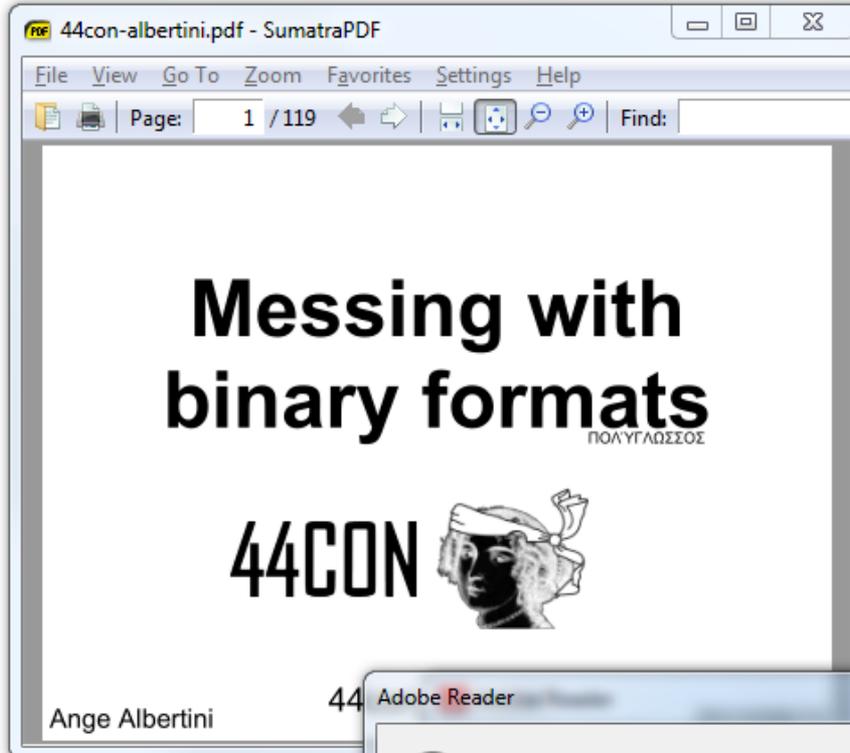
PDF viewer



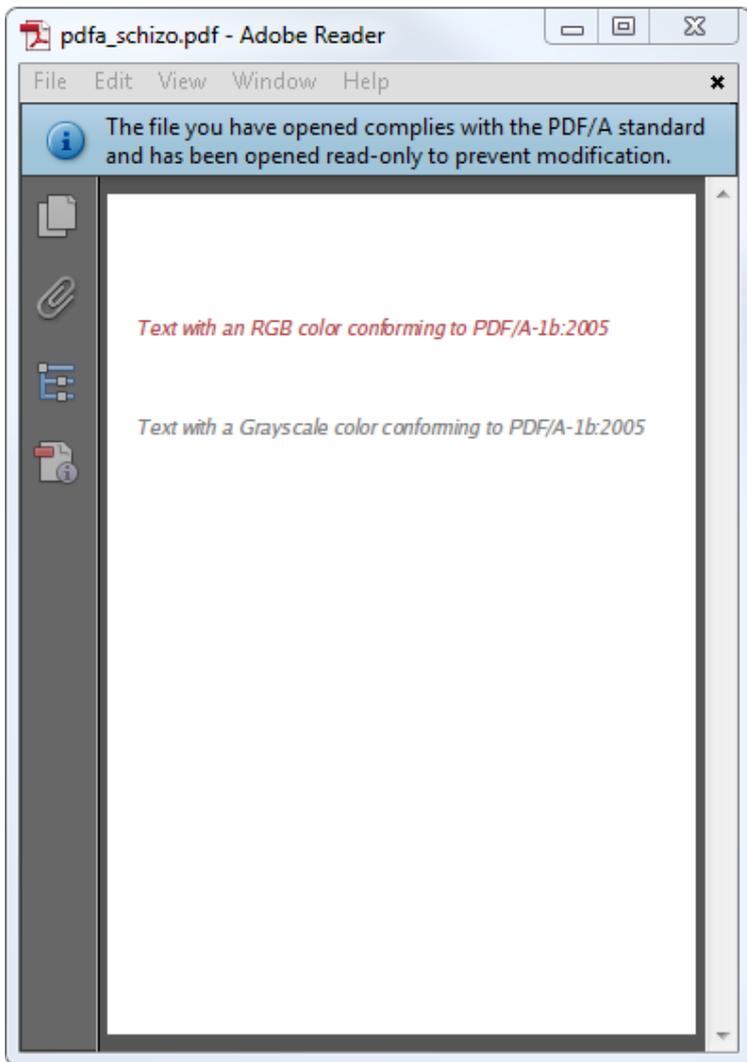
the slides for my talk at 44Con are distributed as a file that is *simultaneously*

a PDF *and* a PE (a PDF viewer) so that the slides can view themselves (oh, and it's also HTML + Java)...





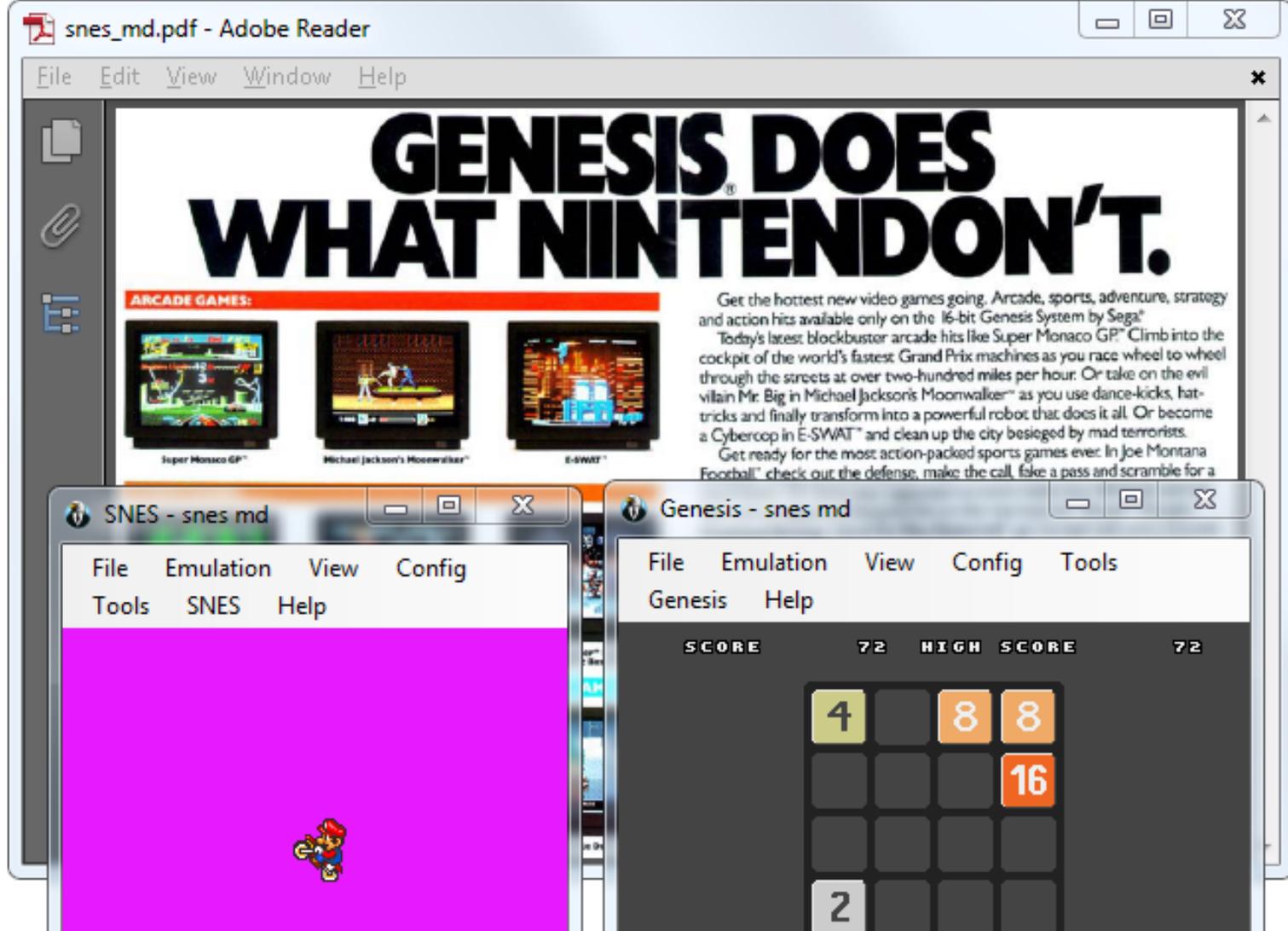
...and it's also schizophrenic (PDF documents appear different with different readers)

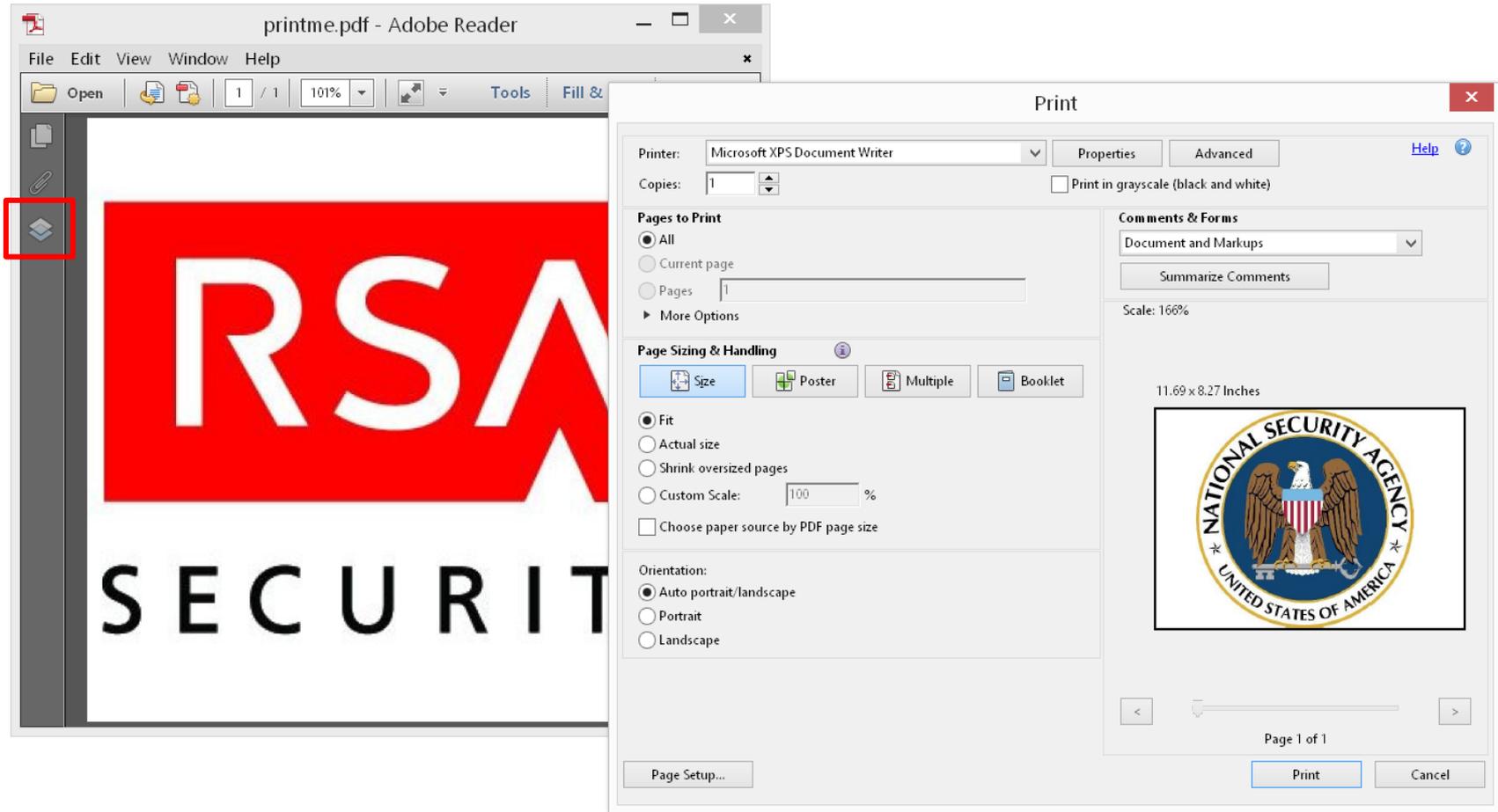


(Also available in PDF/A flavour)

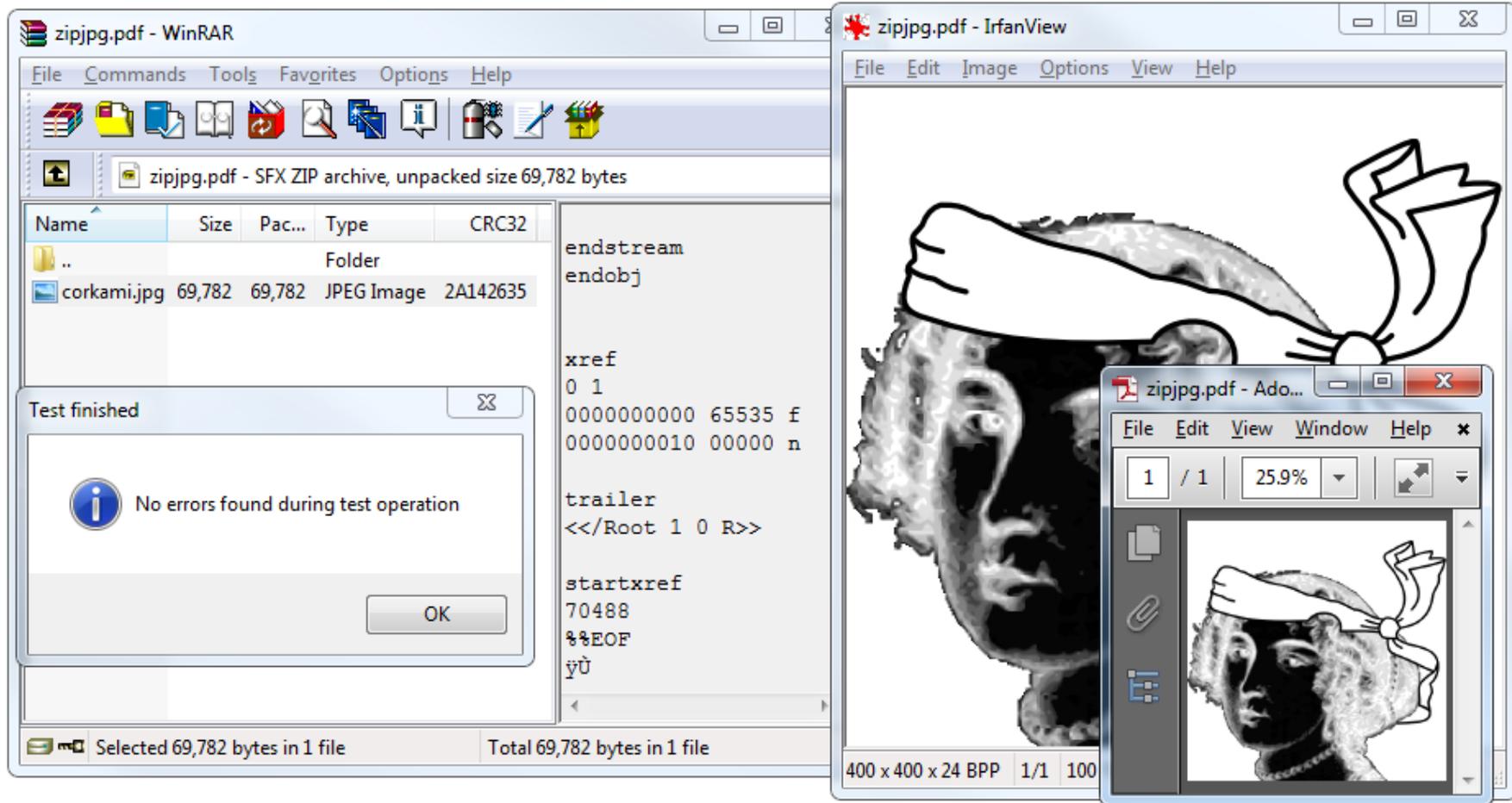


Super NES  
Megadrive

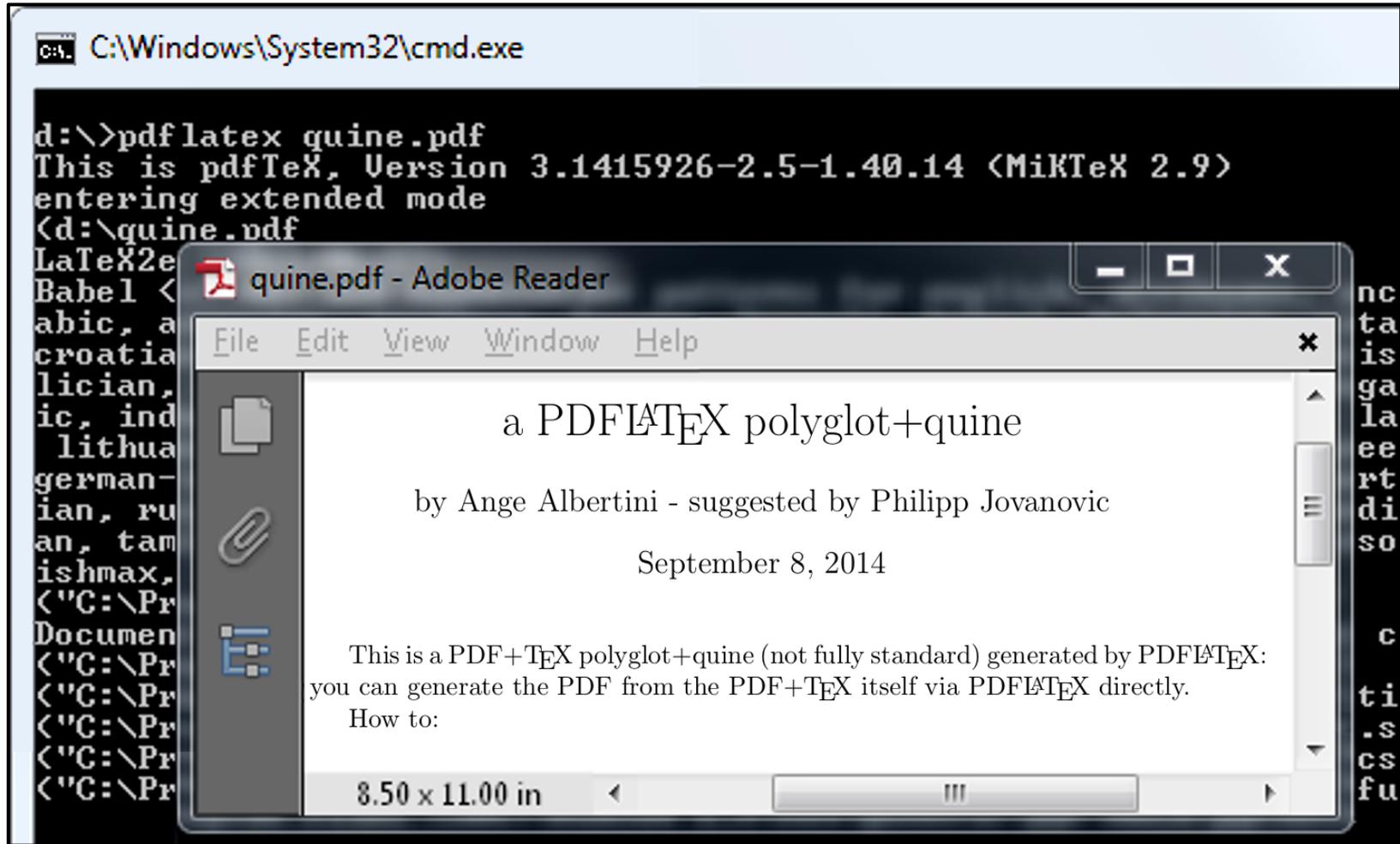




What you see is not always what you print - when you use Layers [Optional Content Groups]!  
Fun fact: you **can't** change the printing output with Adobe Reader ;)

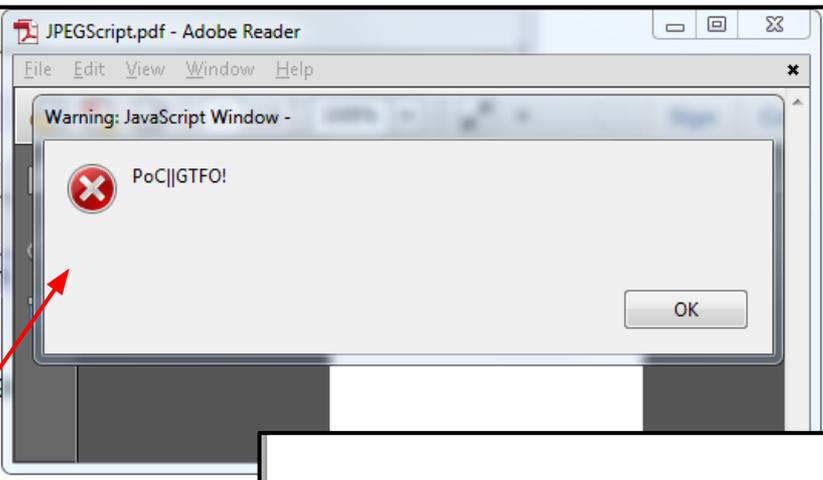


JPEG + ZIP + PDF Chimera (3 headers but only 1 image data)



PDFLaTeX quine (the document is its own source)

```
JPEGScript.pdf
trailer << /Size 2 /Root <</Pages >>>
/OpenAction << /S/JavaScript /JS 1 0 R
>>
1 0 obj<</Filter[/ASCIIHexDecode/DCTDecode]/Width
/ColorSpace/DeviceGray >>
stream
ffd8ffe000104a46494600010100000100010000ffdb00430
endstream
endobj
```



script == picture

```
JPEGScrip
app.alert("PoC||GTFO!");
```

# JPEG-encoded JavaScript

(deprecated)

# PoC||GTFO

International Journal of Proof-of-Concept or Get The F\*\*\* Out

the “new” 2600 / Phrack...

Distributed as **PDF** ⇒ each issue is a PoC

```

QEMU
-----
Berliner Spargel Operating System
Mein Deutsch is nicht so gut, aber es ist Spargel zeit!
by Travis Goodspeed

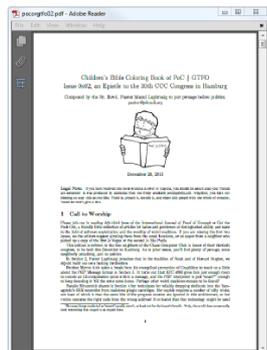
m -- Memory Viewer
a -- About

This is a minimal operating system by Travis Goodspeed for 16-bit Real
Mode 8086 on an IBM PC.  It was written in order to learn about the
8086, and it quite likely will serve no use for you.  It is free
without any strings attached, but please give credit were credit is
due if you fork it.

Also, and this is very important, you should use the included hex viewer
to poke around this machine's memory.  The boot sector at 0000:7C000
is likely a good place to start.
Press the 'any' key to continue._

```

# MBR (bootable) + PDF + ZIP



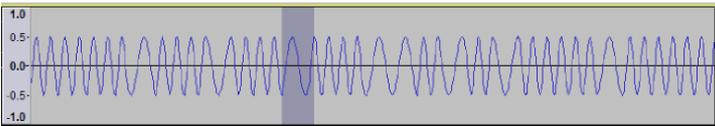
```

Archive: pocorgtfo02.pdf
warning [pocorgtfo02.pdf]: 8016414 extra bytes at beginning or within zipfile
(attempting to process anyway)
  Length  EAS  ACLs  Date    Time    Name
  -----
    852    0    0    12/06/13 16:25  README.txt
   6794    0    0    12/06/13 16:25  coda.txt
  20164    0    0    12/06/13 16:25  feeling.txt
  12618    0    0    12/06/13 16:25  harrison.txt
     0     0    0    12/06/13 16:25  pppquine/
   275     0    0    12/06/13 16:25  pppquine/Makefile
  1006     0    0    12/06/13 16:25  pppquine/inflate.c
   5323    0    0    12/06/13 16:25  pppquine/quine.c
  203706   0    0    12/06/13 16:25  rfc4880.txt
 2046109   0    0    12/06/13 16:25  tamagotchi.zip
   15565   0    0    12/06/13 16:25  thewub.txt
   278598   0    0    08/05/13 13:06  pocorgtfo00.pdf
  3790438   0    0    10/13/13 02:47  pocorgtfo01.pdf
  -----
 6381448   0    0                      13 files

```

FILE	JPEG	PDF
0000: ff d8	"START OF IMAGE" MARKER	
0002: ff e0 <size.16> <content>	"APP0" MARKER (REQUIRED   HEADER)	
0014: ff fe <size.16>	"COMMENT" MARKER	
+4: %PDF-1.5	COMMENT CONTENT	PDF SIGNATURE
999 0 obj		STARTING A DUMMY BINARY OBJECT
<<>		
stream		
0039: ...	(OTHER MARKERS, ORIGINAL JPEG DATA)	
xx : ff d9	"END OF IMAGE" MARKER	
xx+2 : endstream		CLOSING THE DUMMY OBJECT
endobj		
xx+14: %PDF-1.5 ...		ORIGINAL PDF" CONTENTS (MULTIPLE SIGNATURES ARE IGNORED)

\*REPLACED WITH 00 00 TO BYPASS ADOBE FILTER



# AngeCryption: getting valid files after encryption

1 CONTROLLING FIRST ENCRYPTED BLOCK

PLAINTEXT BLOCKS: P1, P2

ENCIPHERING:  $C1 = ENC(P1 \oplus IV)$ ,  $C2 = ENC(P2 \oplus IV)$

DECRYPTING:  $IV = DEC(C1) \oplus P1$ ,  $IV = DEC(C1) \oplus P1$

EXAMPLE WITH AES:

```

KEY: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
IV: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ENC(PDF SIGNATURE) = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  
```

2 CONTROLLING ENDING ENCRYPTED BLOCKS

"ENC(A) = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00"

"DEC(B) = [icon of a person with a lightbulb]"

"ENC(C) = [icon of a person with a lightbulb]"

3 SKIPPING UNCONTROLLED BLOCKS

(1) PNG SIGNATURE STARTING A DUMMY CHUNK

RANDOMLY ENCRYPTED DATA

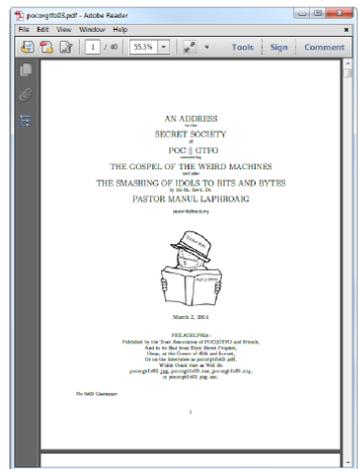
ENDING DUMMY CHUNK: 99 99 99 99

(2) STARTING CONTROLLED DATA

END OF IMAGE: .. 00 00 00 00 00 .. I . N . D RC 42 60 02

ANGE ALBERTIN  
WITH THE HELP OF JEAN-PHILIPPE AUMASSON

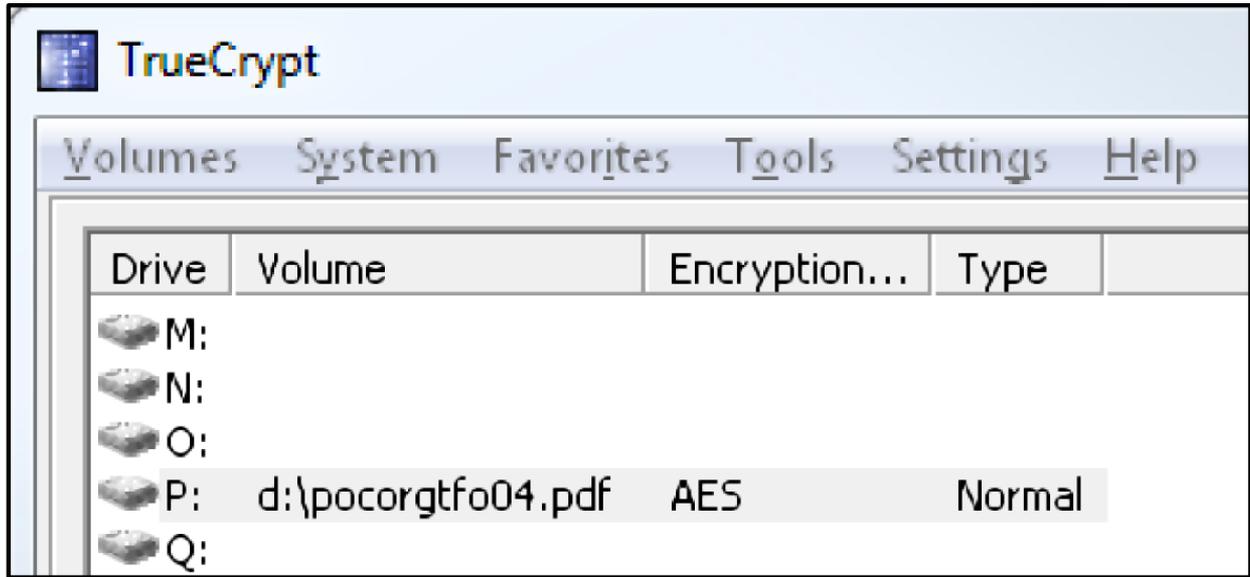
raw audio +  
JPG + AES(PNG)  
+ PDF + ZIP



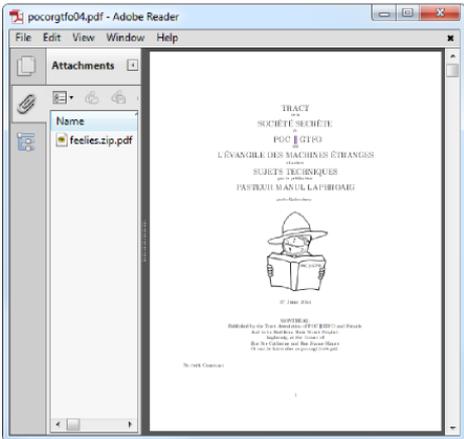
```

Archive: pocorgtfo03.pdf
warning [pocorgtfo03.pdf]: 12224072 extra bytes at beginning or within zipfile
(attempting to process anyway)
  
```

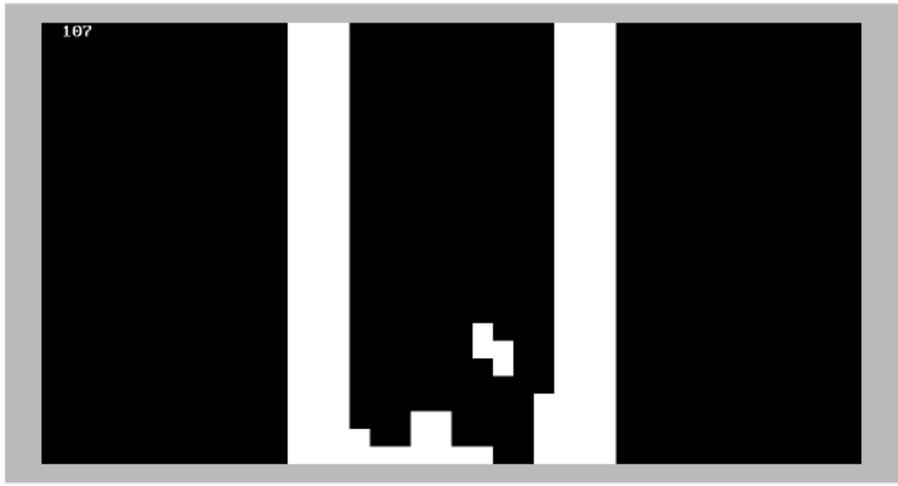
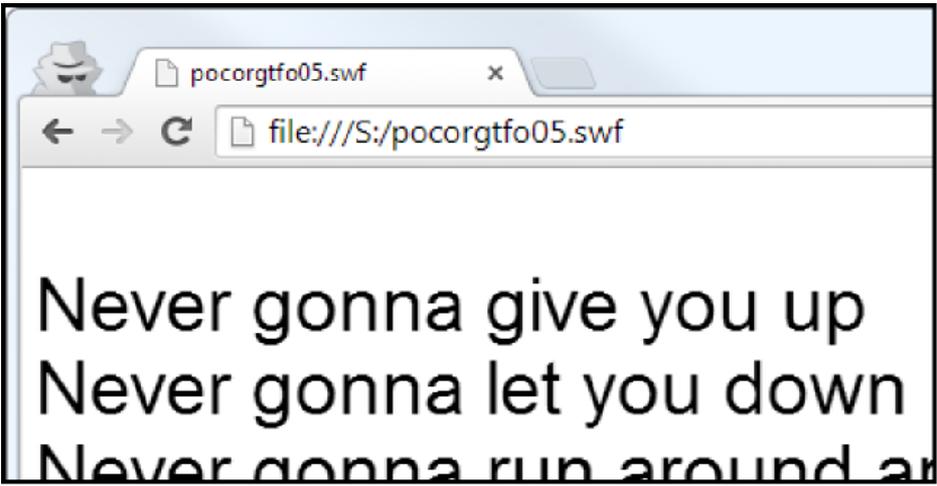
Length	EAS	ACLs	Date	Time	Name
2561	0	0	02/10/14	06:23	alexander.txt
7848	0	0	02/08/14	20:20	bochs-2.6.2.patch
6135	0	0	02/08/14	20:21	bochs-20140203.patch
7248	0	0	02/09/14	08:35	defusing.zip
4830	0	0	12/01/13	15:48	despair.txt
14892	0	0	11/27/13	19:03	lasta.txt
26325	0	0	02/07/14	21:06	lastq.txt
473449	0	0	02/07/14	21:06	netwatch-337f8b1.tar.gz
131930	0	0	02/24/14	20:32	nokiacypher.png
14645	0	0	02/17/14	18:52	packed
2129	0	0	02/07/14	21:06	saucers.txt
3144	0	0	02/07/14	21:06	tamadec.txt
6227	0	0	02/07/14	21:06	tetraglix.tar.bz2
14109425	0	0	02/07/14	21:06	pocorgtfo02.pdf
322	0	0	03/03/14	01:28	pocorgtfo03-encrypt.py
14811110	0	0			15 files



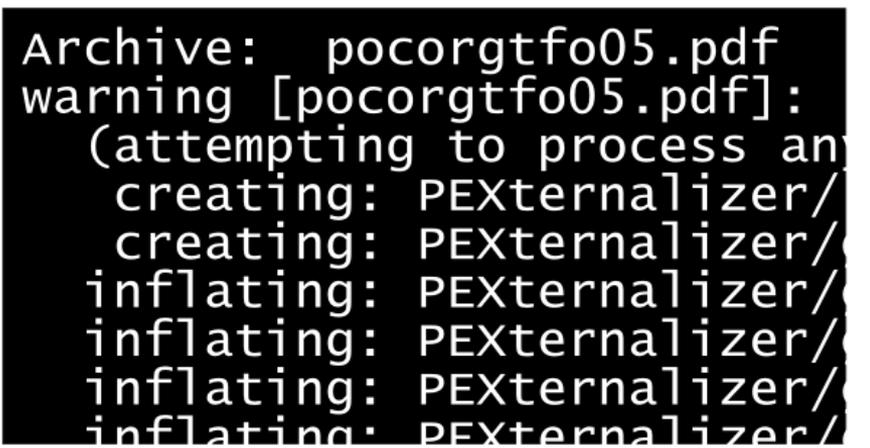
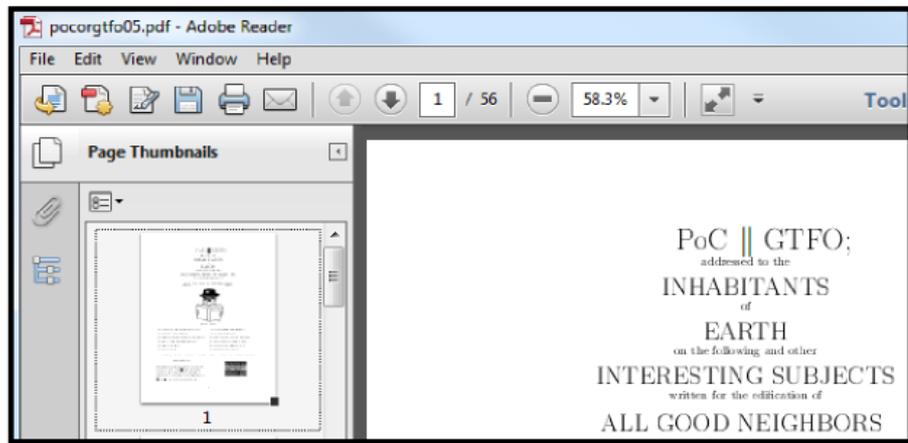
TrueCrypt  
+ PDF + ZIP



```
Archive: pocorgtfo04.pdf
warning [pocorgtfo04.pdf]: 798586 extra bytes at beginning or within zipfile
(attempting to process anyway)
error [pocorgtfo04.pdf]: reported length of central directory is
-798586 bytes too long (Atari STZip zipfile? J.H.Holm ZIPSPLIT 1.1
zipfile?). Compensating...
Length      EAS      ACLS      Date      Time      Name
-----
0           0         0         06/24/14  18:56    bin2png/
5010        0         0         06/24/14  18:56    bin2png/bin2png.py
18025       0         0         06/24/14  18:56    bin2png/LICENSE
1141        0         0         06/24/14  18:56    bin2png/README.md
140413     0         0         06/24/14  18:56    darfsteller.txt
2841        0         0         06/24/14  18:56    gods.txt
0           0         0         06/24/14  18:56    lenticrypt/
36445      0         0         06/24/14  18:56    lenticrypt/lenticrypt.py
18025       0         0         06/24/14  18:56    lenticrypt/LICENSE
776         0         0         06/24/14  18:56    lenticrypt/README.md
2709        0         0         06/24/14  18:56    lenticrypt/test.py
3111965    0         0         06/24/14  18:56    pocorgtfo.png
25986       0         0         06/24/14  18:56    theveldt.txt
239224     0         0         06/24/14  18:56    tsb-20140401.zip
26750864   0         0         06/24/14  18:56    pocorgtfo03.pdf
-----
30353424   0         0
15 files
```



Flash + bootable ISO + PDF + ZIP



# TAR + PDF + ZIP

```
$ tar -tvf pocorgtfo06.pdf
-rw-r--r-- Manul/Laphroaig    0 2014-10-06
-rw-r--r-- Manul/Laphroaig 525849 2014-10-
-rw-r--r-- Manul/Laphroaig 273658 2014-10-
```

```
$ unzip -l pocorgtfo06.pdf
Archive:  pocorgtfo06.pdf
warning [pocorgtfo06.pdf]: 10672929 extra
(attempting to process anyway)
```

Length	Date	Time	Name
-----	-----	-----	-----
4095	11/24/2014	23:44	64k.txt
818941	08/18/2014	23:28	acsac13_zadd
4564	10/05/2014	00:06	burn.txt
342232	11/24/2014	23:44	davinci.tgz.
3785	11/24/2014	23:44	davinci.txt
5111	09/28/2014	21:05	declare.txt
0	08/23/2014	19:21	ecb2/

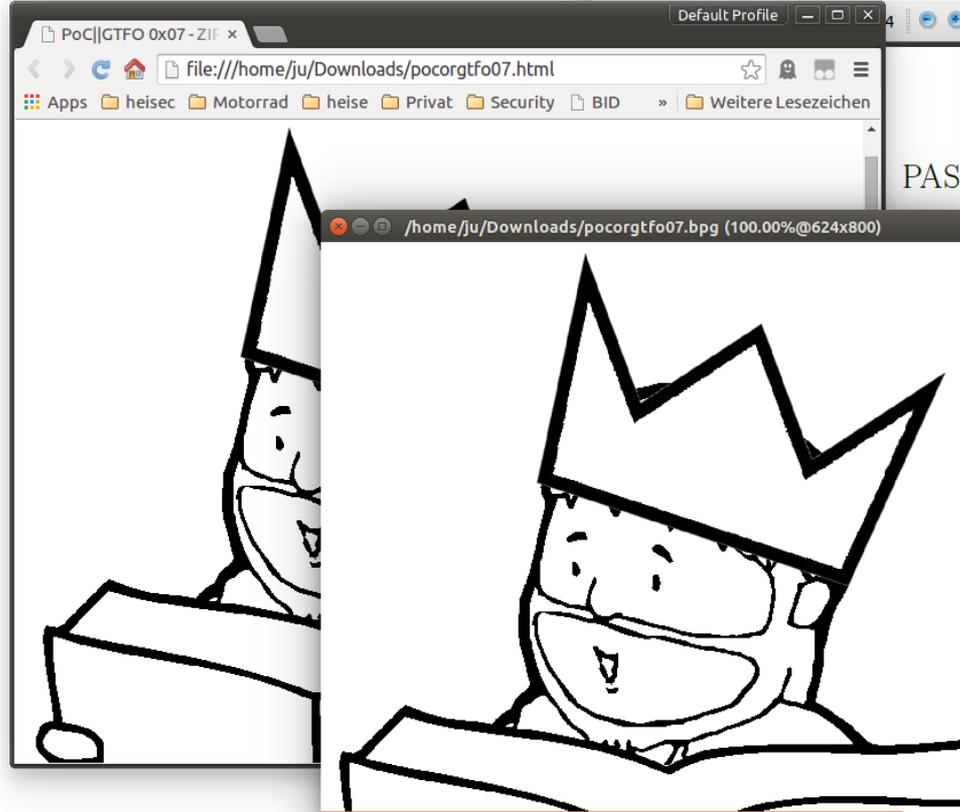
PoC || GTFO;  
brings that  
OLD TIMEY EXPLOITATION  
with a  
WEIRD MACHINE JAMBOREE  
and our world-famous  
FUNKY FILE FLEA MARKET  
not to be ironic, but because  
WE LOVE THE MUSIC!

November 25, 2014

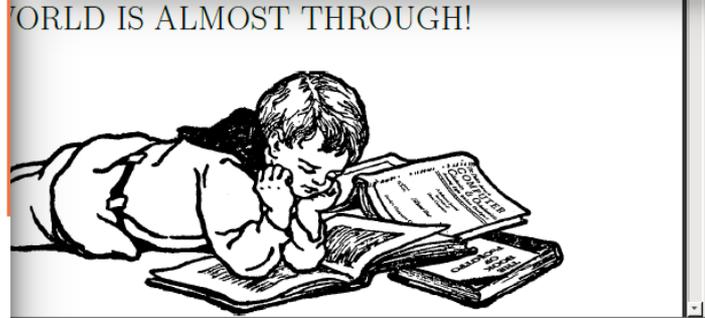
6:2 On Giving Thanks	6:7 Cracking AngeCryption with ECB.py
6:3 Dolphin Emulator Internals (PPC)	6:8 PCB Reverse Engineering
6:4 TAR/PDF Polyglots	6:9 Davinci Self-Extractor
6:5 Pong Easter Eggs in VMWare	6:10 Observable Metrics
6:6 Anti-Emulation for MIPS	6:11 Donate to Laphroaig's Oday Charity

# BPG + HTML (incl. a BPG viewer in JS)

+ PDF + ZIP



```
ju@loki:
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
Archive: pocorgtfo07.zip
***** PWNED *****
dumping credentials...
*****
Length      Date       Time      Name
-----
6325      2015-02-02  20:56     500miles.txt
0         2015-03-19  15:51     abusing_file_formats/
370375    2015-03-06  21:51     abusing_file_formats/3in1.png
512       2015-03-06  21:51     abusing_file_formats/abstract.tar
143360    2015-03-06  21:51     abusing_file_formats/AppleII.pdf
116       2015-03-06  21:51     abusing_file_formats/asciizip
420852    2015-03-06  23:27     abusing_file_formats/cameron.png
41902     2015-03-19  15:51     abusing_file_formats/chimera.png.zip
122880    2015-03-07  19:16     abusing_file_formats/chimera.tar.pdf
590538    2015-03-06  21:51     abusing_file_formats/corkami_spectrogram.bmp
537654    2015-03-06  21:51     abusing_file_formats/dr_nsa.bmp
10213     2015-03-06  21:51     abusing_file_formats/I love reliable forensics sof
ware.png
```



# Shell script + PDF + ZIP

```
$ echo "terrible raccoons achieve their escapades" | ./pocorgtfo08.pdf -d 4321  
good neighbors secure their communications
```

```
$ unzip -l pocorgtfo08.pdf
```

```
Archive:  pocorgtfo08.pdf
```

Length	EAS	ACLs	Date	Time	Na
988446	0	0	08/06/15	22:46	EC
440648	0	0	09/06/15	20:36	ai
522633	0	0	09/06/15	19:18	ai
1546	0	0	08/06/15	22:46	al
118696	0	0	08/06/15	22:46	br
31337	0	0	08/06/15	22:46	ex
38109	0	0	08/06/15	22:46	ge
303926	0	0	08/06/15	22:46	if
160225	0	0	08/06/15	22:46	jt
3149	0	0	08/06/15	22:46	le
2244652	0	0	08/06/15	22:46	ma
4662	0	0	08/06/15	22:46	ni



AS EXPLOITS SIT LONELY,  
FORGOTTEN ON THE SHELF  
YOUR FRIENDLY NEIGHBORS AT  
PoC || GTFO  
PROUDLY PRESENT

# **... and others**

Bootable quine in assembly,  
2 switchable PDFs via ROT13,  
hash collisions,  
GameBoy + Sega Master System...

# **You get the idea...**

The worst case for preservation?

I explore corner cases, before attackers do it

# How is it possible?

- signature offset not enforced
- stream object (containing anything)
- comments can contain binary data
- appended data
- objects tolerated between XREF and startxref

and a few specific abuses (some are fixed now)

# What is PDF ?

I asked online...

Postscript Derived Failure

Potential Disaster Forever

Practically Destructive File

Pretty Demented Format

Preservation Dooming Format

POC||GTFO Demonstration Format

Paper Dimensions Fixed

Payload Deployment File

Posterity Depends on Forensics

**Preserving Document Forever**

Pathetic & Dangerous Format

Posthoc Depression Format

Perversely Designed Format

Penile Dysfunction Format

Proprietary Document Fee

Public Domain Farce

Postscript Didn't Fit

Penetrate Dodgy Firewall

Please Don't Fail / Again

Proven Dysfunctional Format

Pants-Down Format

Polyglot (Definition|Deployment|Delivery) Framework

PDF is a Disaster for the Future

Perpetually Disagreeable Format

...and I wasn't disappointed :)

**More seriously...**

(from my *personal* point of view)

# **A miracle?**

Fonts are embedded in the document

Rendering is following complex rules  
(overly-complex, from a security standpoint)



# An open format?

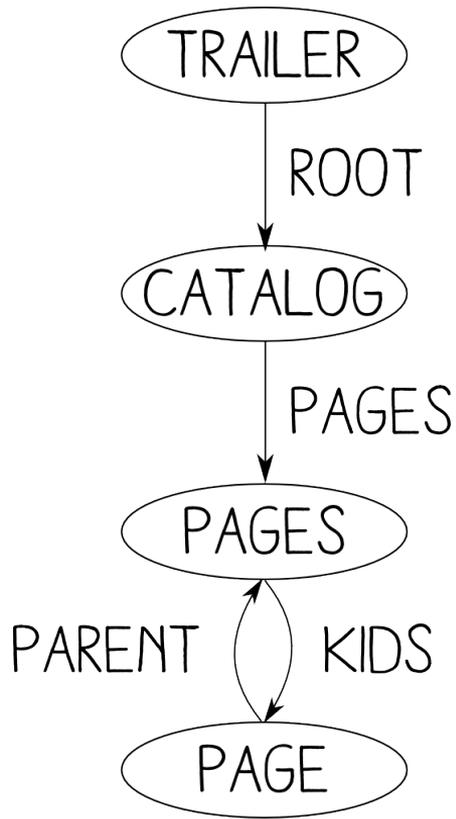
ISO \$pec\$ = 200\$

These specs only cover the main part :(

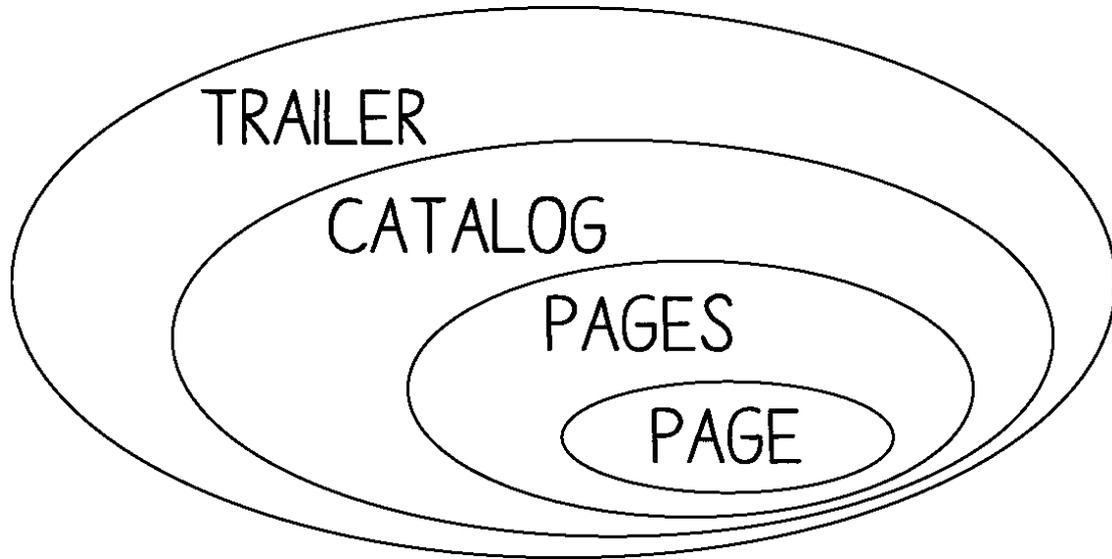
They are unclear - no formal guarantee :(

# **A strict format ?**

No reader completely enforces the specs  
⇒ recovery mode (sometimes 'explicit')  
signature, stream length, XREF...



standard structure  
(each object should be distinct)



non-standard but tolerated structure  
(inlined objects)

Many possible malformations handled specifically by each reader (high level)...

# Many possible abuses

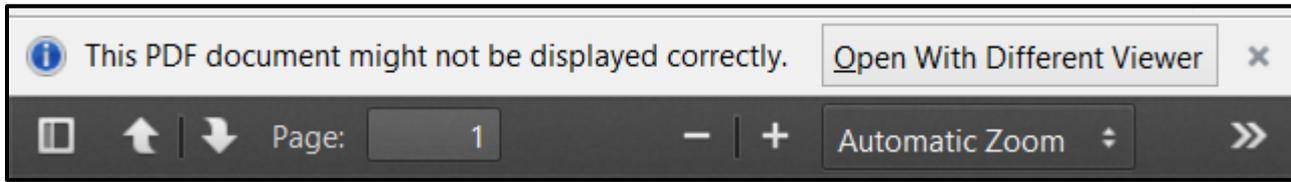
follows the specs

corruption tolerated

absence tolerated

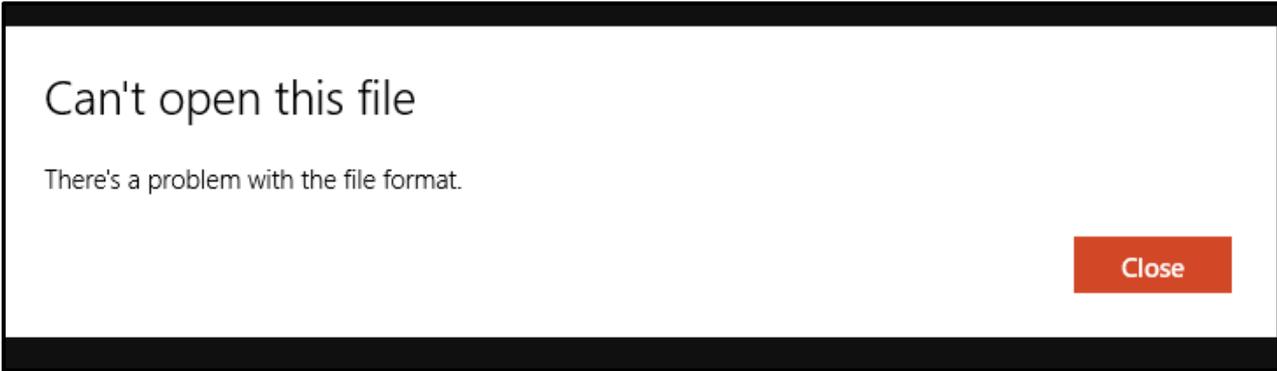
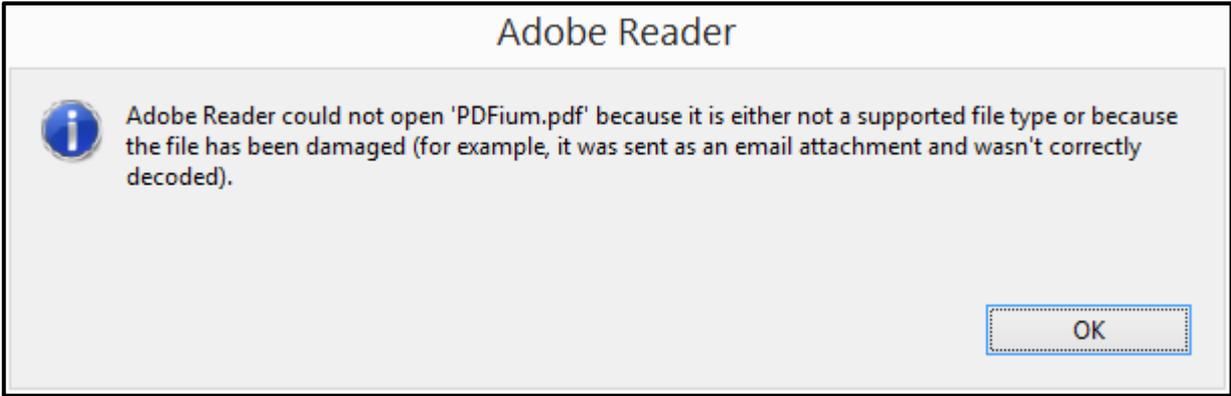
	signature	endobj	/Count	text operators	/Font	font use	xref	/Resources	trailer
Adobe Reader	corruption tolerated	corruption tolerated	absence tolerated	follows the specs	absence tolerated	follows the specs	absence tolerated	follows the specs	follows the specs
MuPDF	absence tolerated	absence tolerated	follows the specs	corruption tolerated	follows the specs	follows the specs	absence tolerated	follows the specs	absence tolerated
PDF.js	absence tolerated	follows the specs	follows the specs	absence tolerated	follows the specs	follows the specs	corruption tolerated	follows the specs	corruption tolerated
PDFium	corruption tolerated	absence tolerated	absence tolerated	absence tolerated	absence tolerated	absence tolerated	absence tolerated	absence tolerated	follows the specs
Poppler	absence tolerated	absence tolerated	follows the specs	absence tolerated	follows the specs	follows the specs	absence tolerated	follows the specs	follows the specs

... different readers have different tolerances ...



Failed to load PDF document

Error loading D:\FireFox.pdf



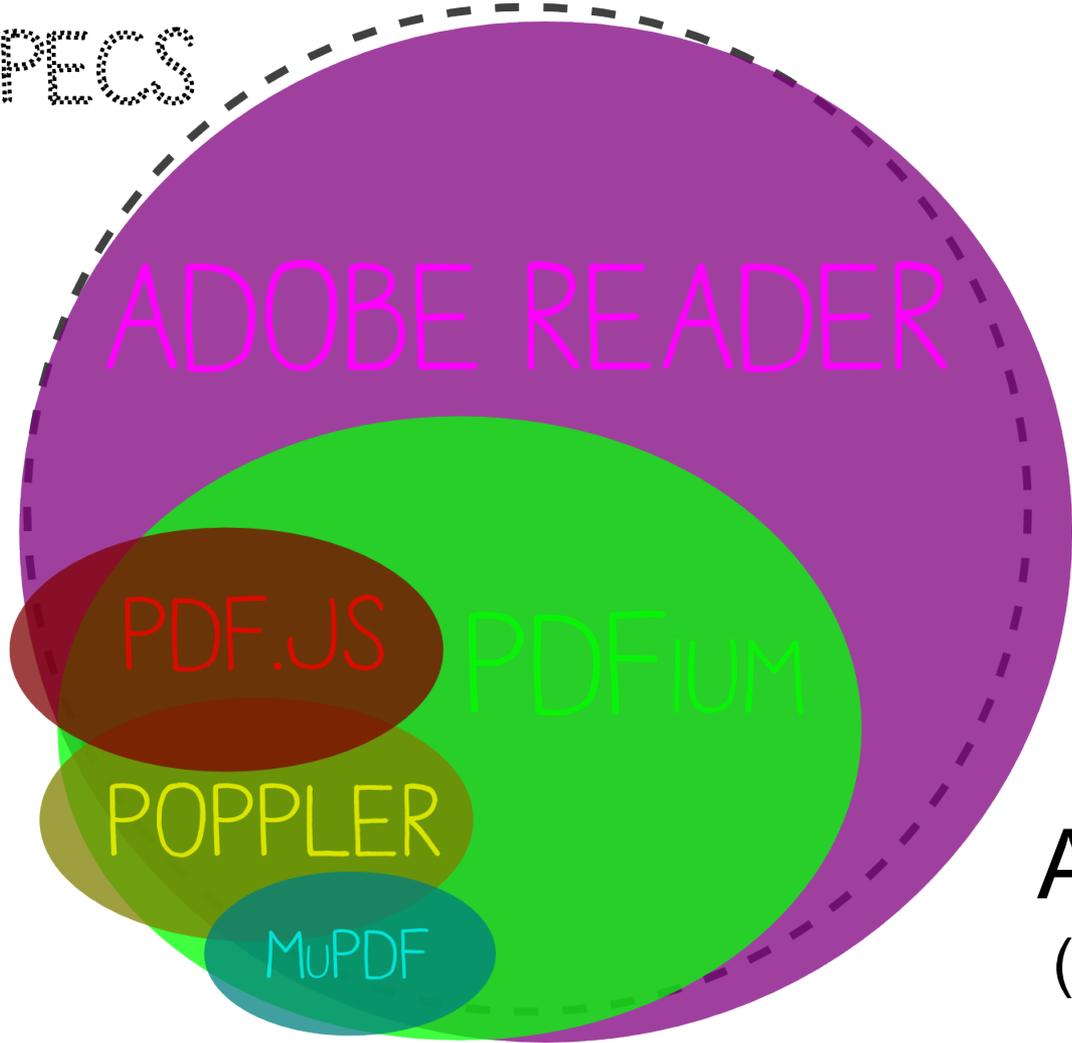
...so a PDF specifically crafted for one reader, may fail with all other readers.

# A uniform format?

Many free readers, but...

- Many (useful) features only available in Adobe Reader:  
forms, signature, layers...  
(it's Adobe's business model)
- Other readers just aim to support “standard” PDFs

SPECS



ADOBE READER

PDF.JS

PDFIUM

POPPLER

MuPDF

**A beautiful mess!**  
(an artist's interpretation)

# A consistent format?

Adobe Reader is closing security issues.

This is good, but...

⇒ Some features are not supported anymore

⇒ Potential lack of backward compatibility

# It's a complex patchwork!

JPGs are stored *entirely* as-is, but PNG have to be *converted* to raw

Forms as XML

PostScript Transfer function

Web (Flash, JavaScript...)

3D objects

# A coherent format?

- text + line comments, yet binary
- unusual whitespace, binary also in comments
- different escaping
- read forward+no separator and object reference
- hex as nibbles and odd-numbered
- bottom up but also possibly top down (who wins?)
- corrupted ZLIB still tolerated
- image compression for non-images

# What if...

...Adobe would stop supporting PDF ?  
We're just left with the 'specs' ?

# **After all...**

...Flash is being killed for security reasons,  
after becoming progressively redundant.

PDF could be converted to something else.

# PDF & preservation

- JPG + OCR'ed text = simple  
...so simple that we wouldn't need PDF ?  
other PDFs = complex (Adobe-dependent)

Is PDF/A the solution? more \$pec\$

# “Backward compatibility”

...is a beautiful utopia!

And it leads to saying  
“we've always done it this way”  
even after several generations :(

# “Backward compatibility”

...can be incompatible with security fixes

JPEG-encoded JavaScript  
PDF polyglots

# Brace yourself...

PDF 2.0 is coming!

It's not improving stability and preservability  
Will Adobe adhere to it ? Since it's distinct now...

# Conclusion

“a *complex* puzzle because the original picture is messy”



Adobe

[WWW.ADOBE.COM/SECURITY](http://WWW.ADOBE.COM/SECURITY)

# Conclusion

- PDF is very useful - omnipresent for a reason
- it's still involved in computer security
  - recent complete takeover of Windows 8.1 by @j00ru
- it's quite a monster
  - I'm merely scratching the surface
  - its specs were messy from the beginning
- it's far from perfect
  - "if only Adobe Reader was open"

# ACK

Paul Wheatley

@doegox @pdfkunfoo @newsoft

@internot @insertscript @avlidienbrunn

@foxgrrl @chrisjohnriley @travisgoodspeed

and everybody for the PDF suggestions :)

@angealbertini  
corkami.com

