PDF

Myths VS FACTS

Oxford University 15th July 2015 Ange Albertini

ANGE ALBERTINI

reverse engineering

VISUAL DOCUMENTATION

@angealbertini

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
```

BODY

```
DICTIONARY
<< [ID VALUE]* >>

1 0 obj                 OBJECT REFERENCE:
<<                      <OBJECT NUMBER> <REVISION NUMBER> R
  /Pages 2 0 R
>>          IDENTIFIER (WITH /)
endobj

2 0 obj
<<
  /Type /Pages
  /Count 1        ARRAY
  /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

                        STREAM PARAMETERS:
4 0 obj                 LENGTH, COMPRESSION....
<< /Length 50 >>
stream
BT                              BEGIN TEXT
  /F1 110 Tf                      FONT F1 (ARIAL) SET TO SIZE 110
  10 400 Td                       MOVE TO COORDINATE 10, 400
STRING                            OUTPUT TEXT 'HELLO WORLD!'
  (Hello World!)Tj              END TEXT
ET
endstream
endobj
```

XREF TABLE

```
CROSS
REFERENCE
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
```

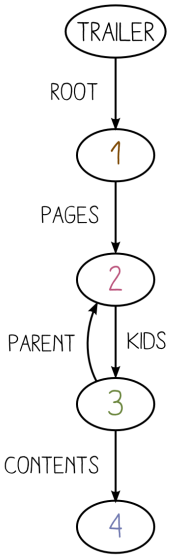
TRAILER

```
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
$x_1\ y_1\ x_2\ y_2\ x_3\ y_3$	**c** cubic bezier
$x_2\ y_2\ x_3\ y_3$	**v** cubic bezier
$x_1\ y_1\ x_3\ y_3$	**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)


even-odd fill

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!

Yes, I write PDFs by hand...

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

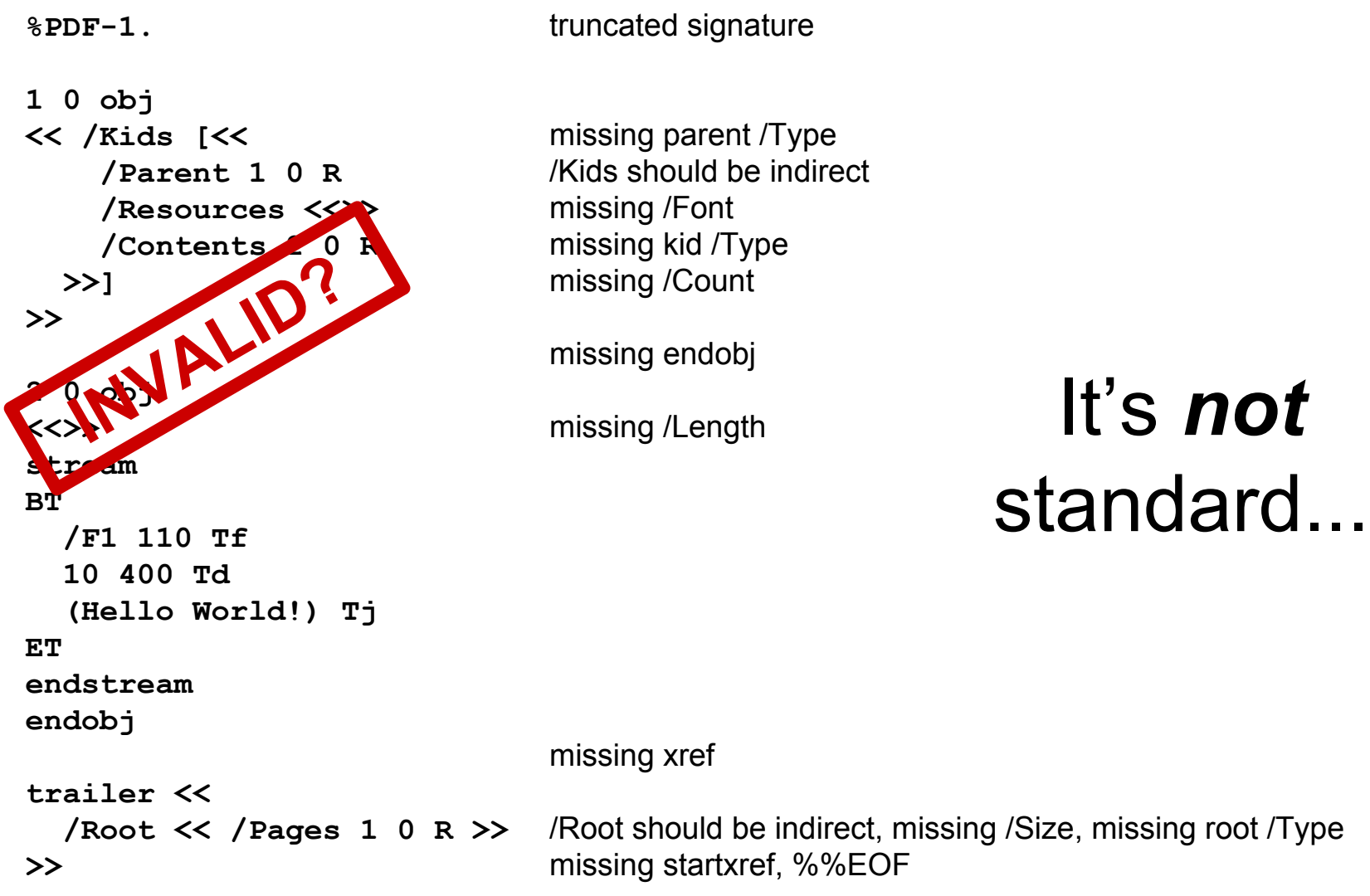
```
%PDF-1.

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 >>
>>
```

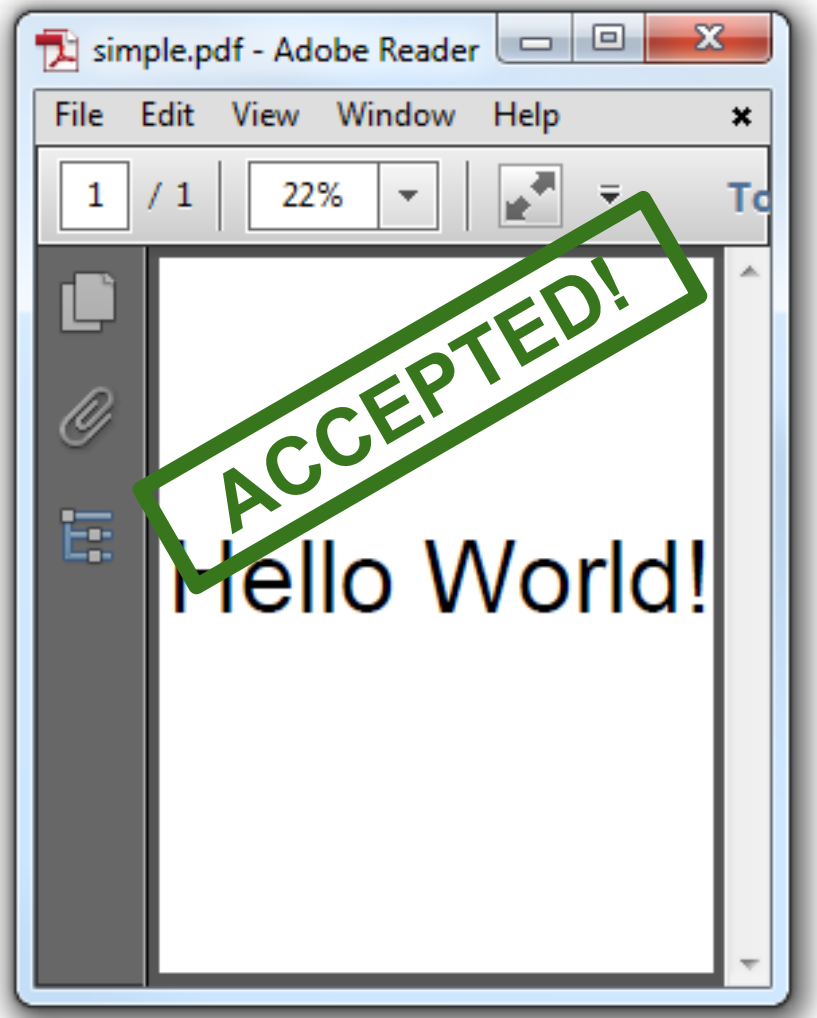
...like this one


%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
trailer <<
/Root << /Pages 1 0 R >>
>>
missing xref
/Root should be indirect, missing /Size, missing root /Type
missing startxref, %%EOF
INVALID?
It's *not* standard...

...but it works

exactly as planned!

(without any reported error)



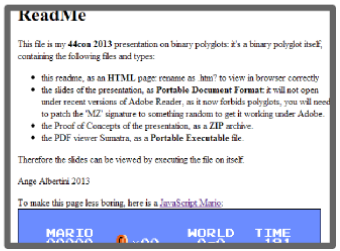
Binary art

PDF + creativity = … ?

PDF slides

PDF viewer

HTML



the slides for my talk at 44Con

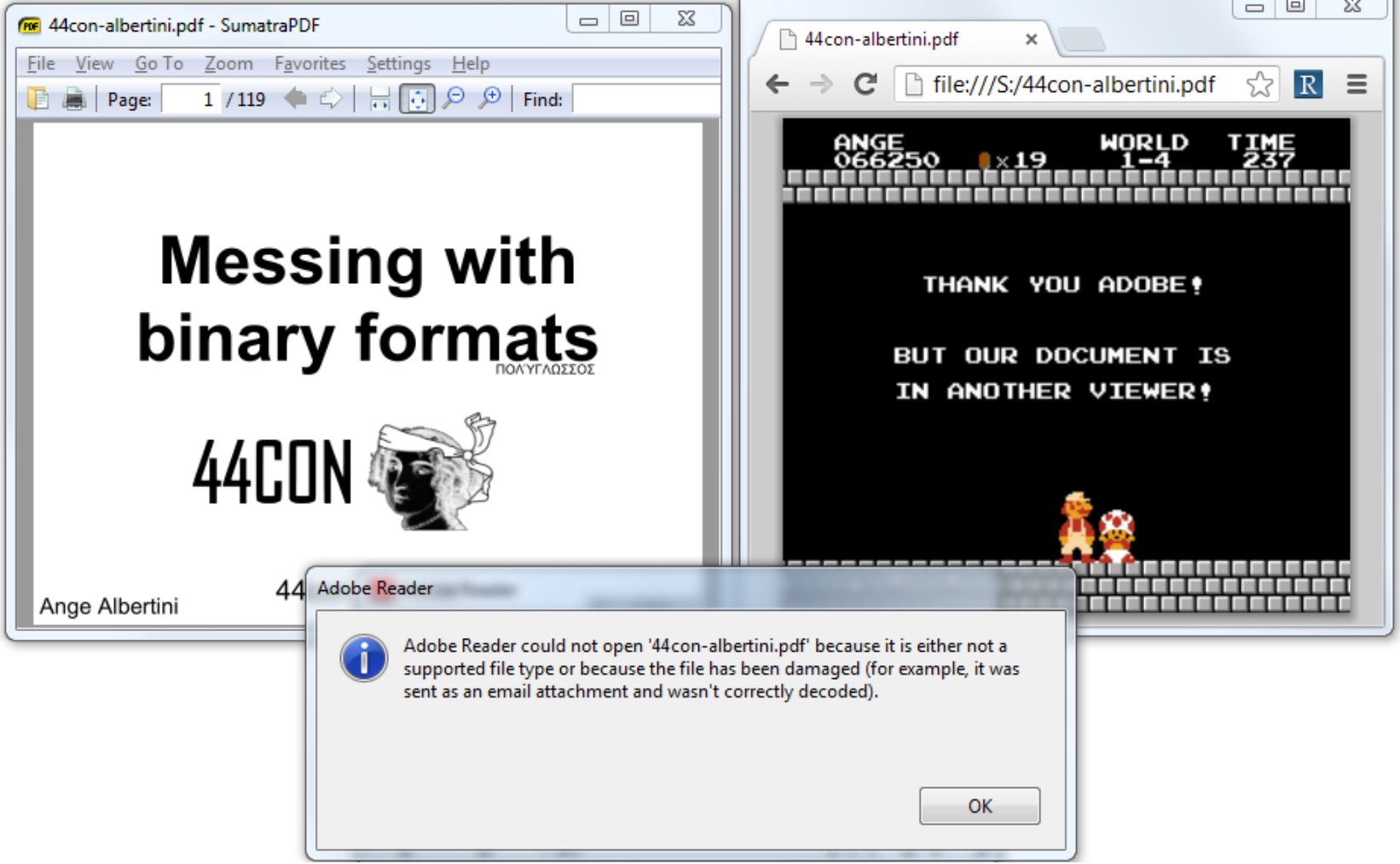
are distributed as a file that is

simultaneously

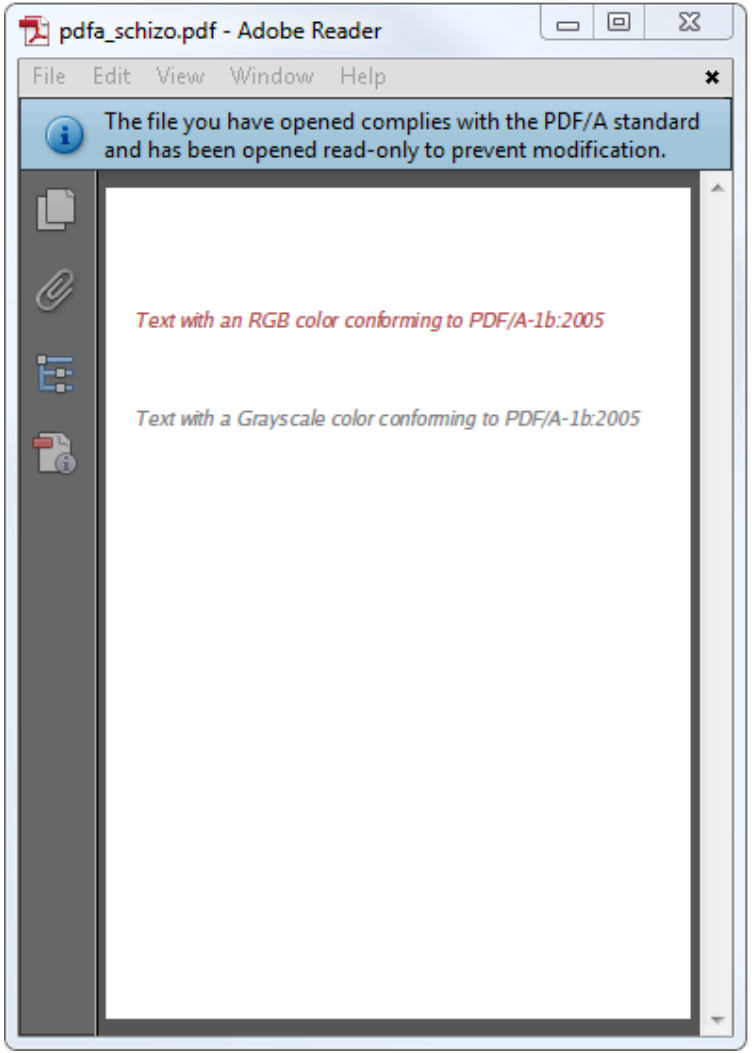
a PDF ***and*** a PE (a PDF viewer)

so that the slides can view themselves

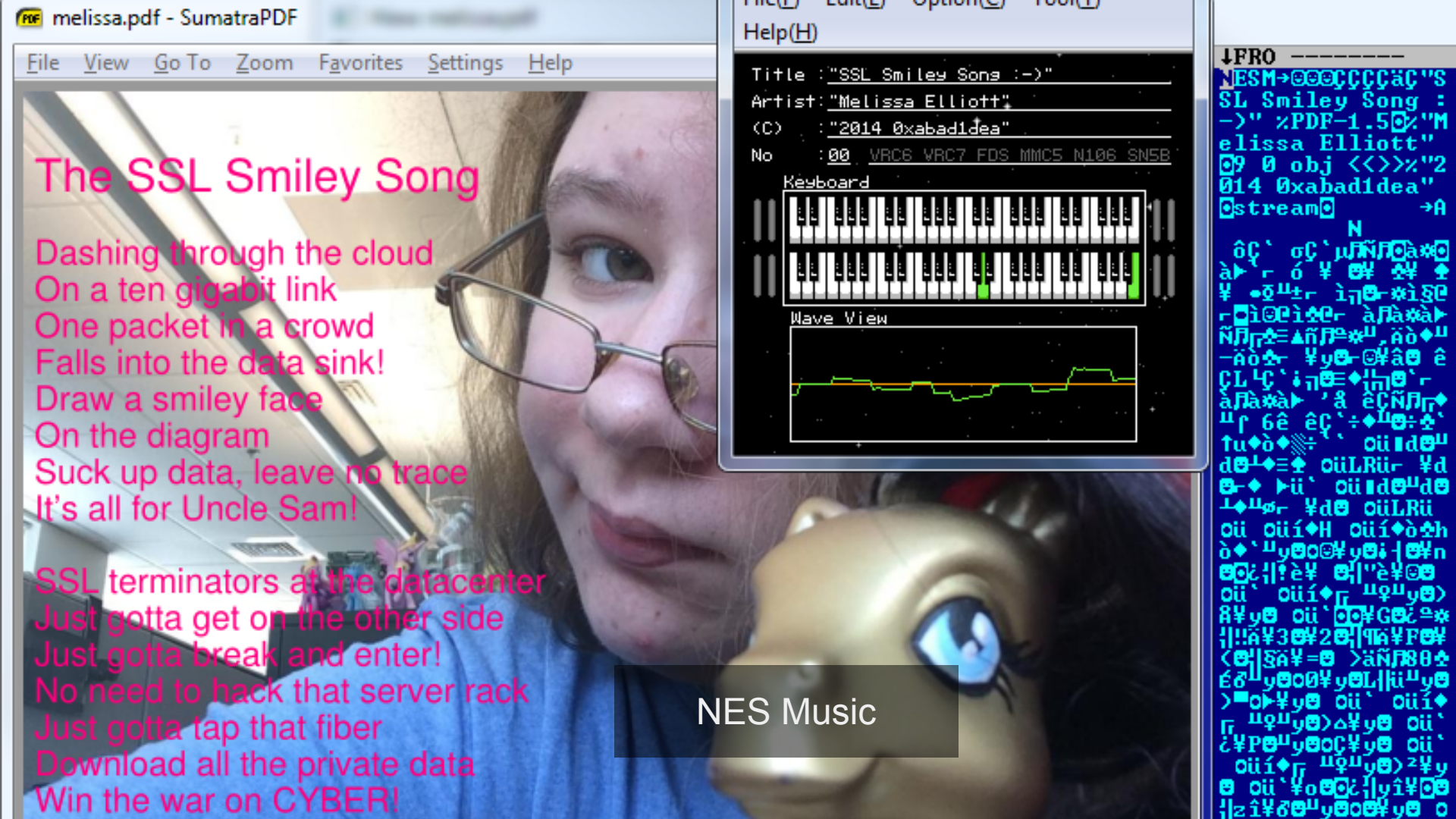
(oh, and it's also HTML + Java)...



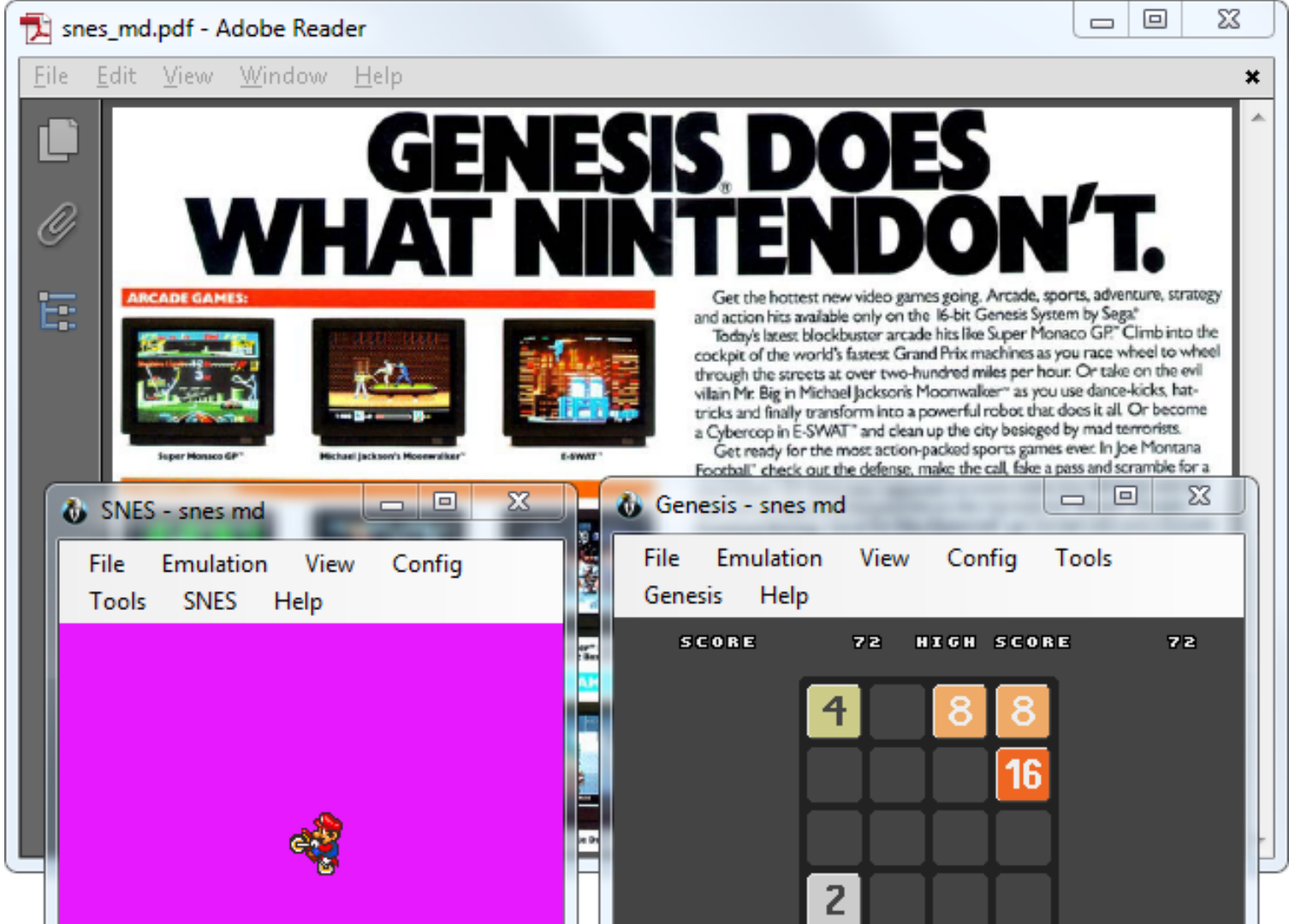
...and it’s also schizophrenic (PDF documents appear different with different readers)

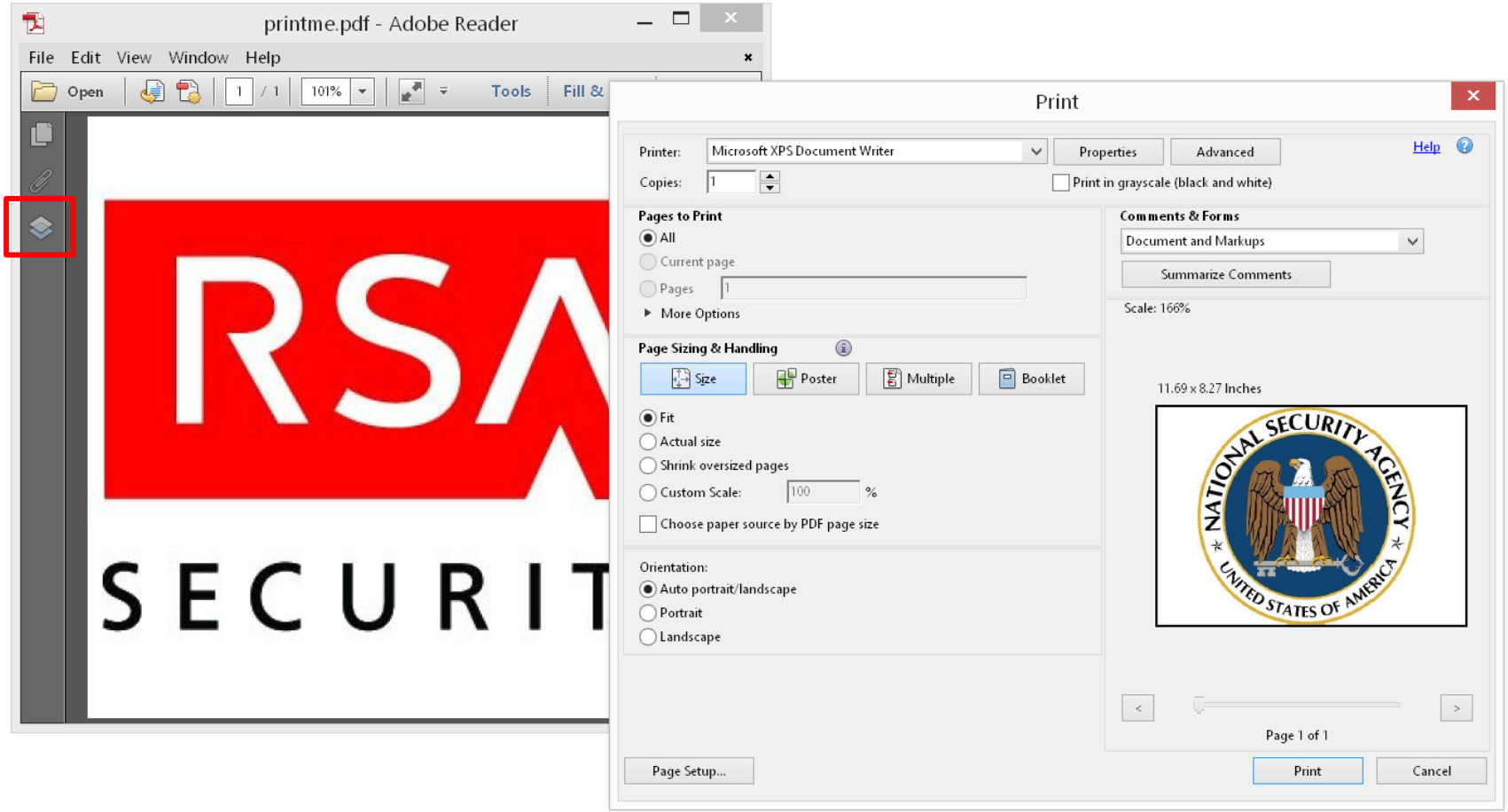


(Also available in PDF/A flavour)


melissa.pdf - SumatraPDF
File View Go To Zoom Favorites Settings Help
Help(H)
Title :"SSL Smiley Song :-)"
Artist:"Melissa Elliott"
(C) :"2014 0xabad1dea"
No :00 VRC6 VRC7 FDS MMC5 N106 SN5B
Keyboard
Wave View
The SSL Smiley Song
Dashing through the cloud
On a ten gigabit link
One packet in a crowd
Falls into the data sink!
Draw a smiley face
On the diagram
Suck up data, leave no trace
It's all for Uncle Sam!
SSL terminators at the datacenter
Just gotta get on the other side
Just gotta break and enter!
No need to hack that server rack
Just gotta tap that fiber
Download all the private data
Win the war on CYBER!
NES Music

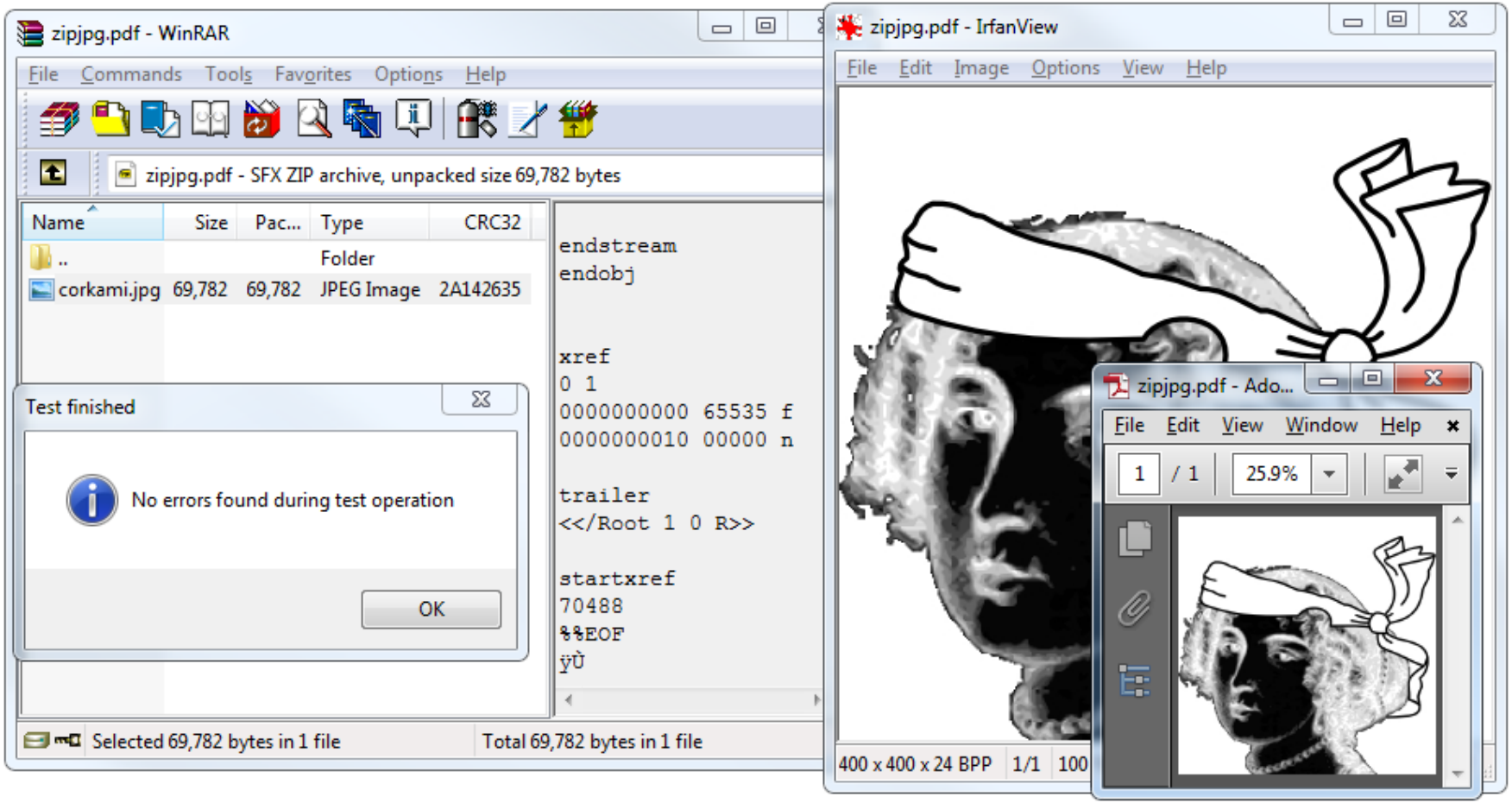
Super NES
Megadrive



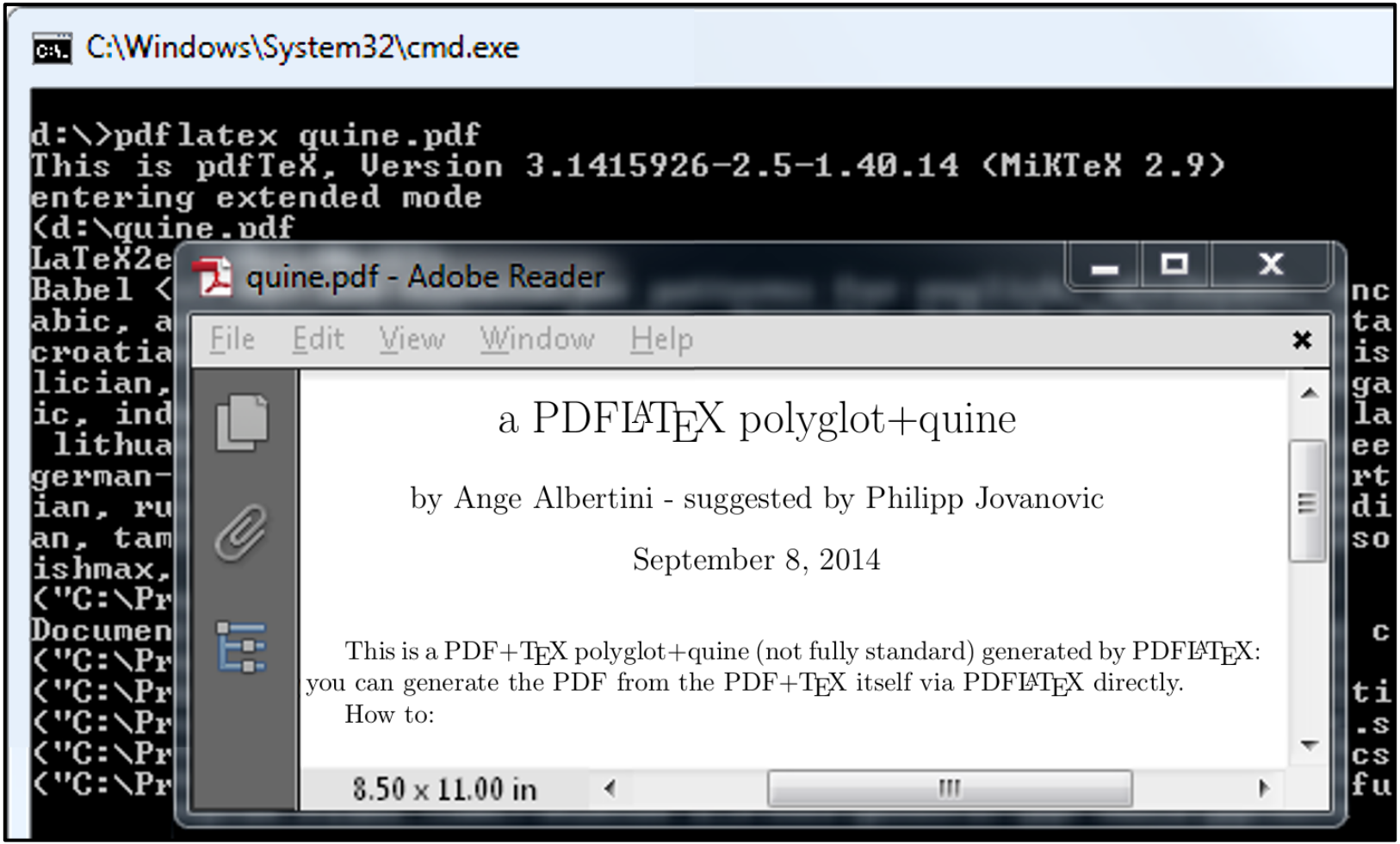


What you see is not always what you print - when you use Layers [***O**ptional **C**ontent **G**roups*]!

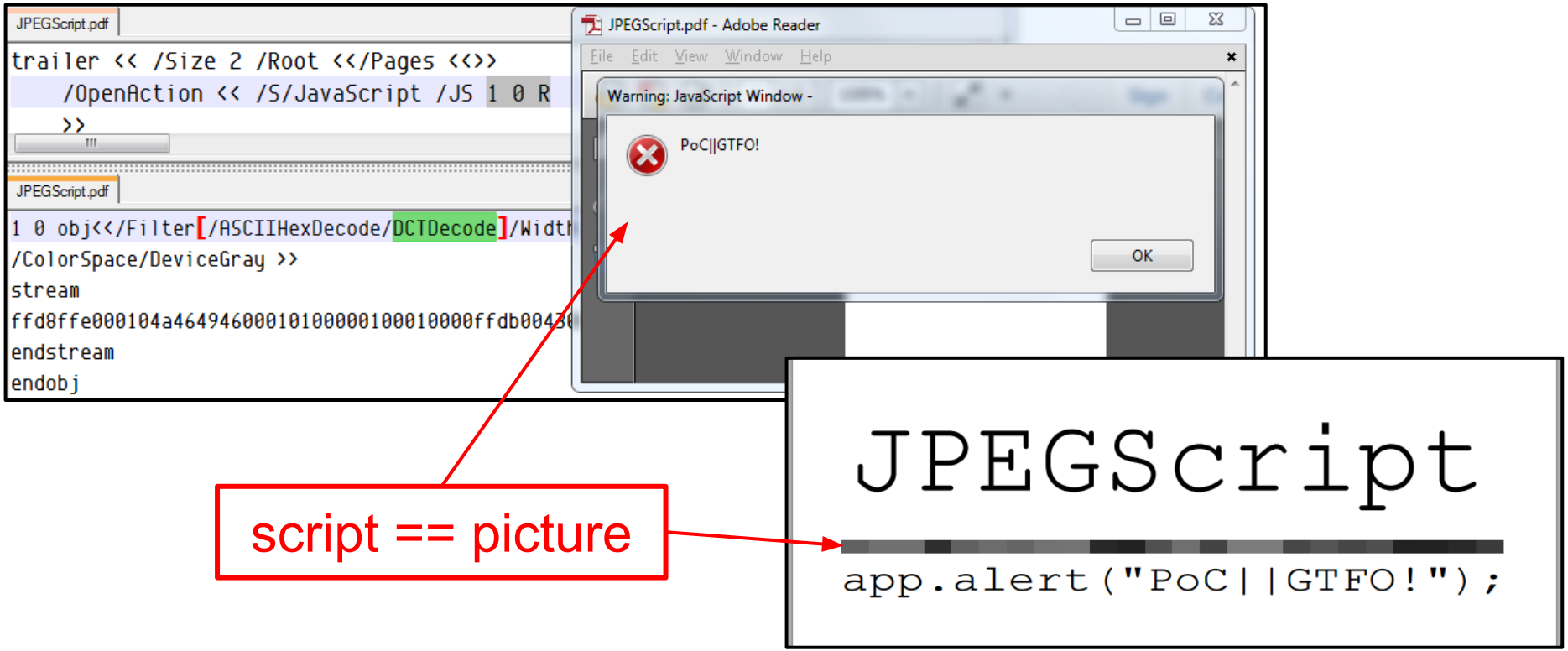
Fun fact: you **can't** change the printing output with Adobe Reader ;)



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



PDFLaTeX quine (the document is its own source)



JPEG-encoded JavaScript

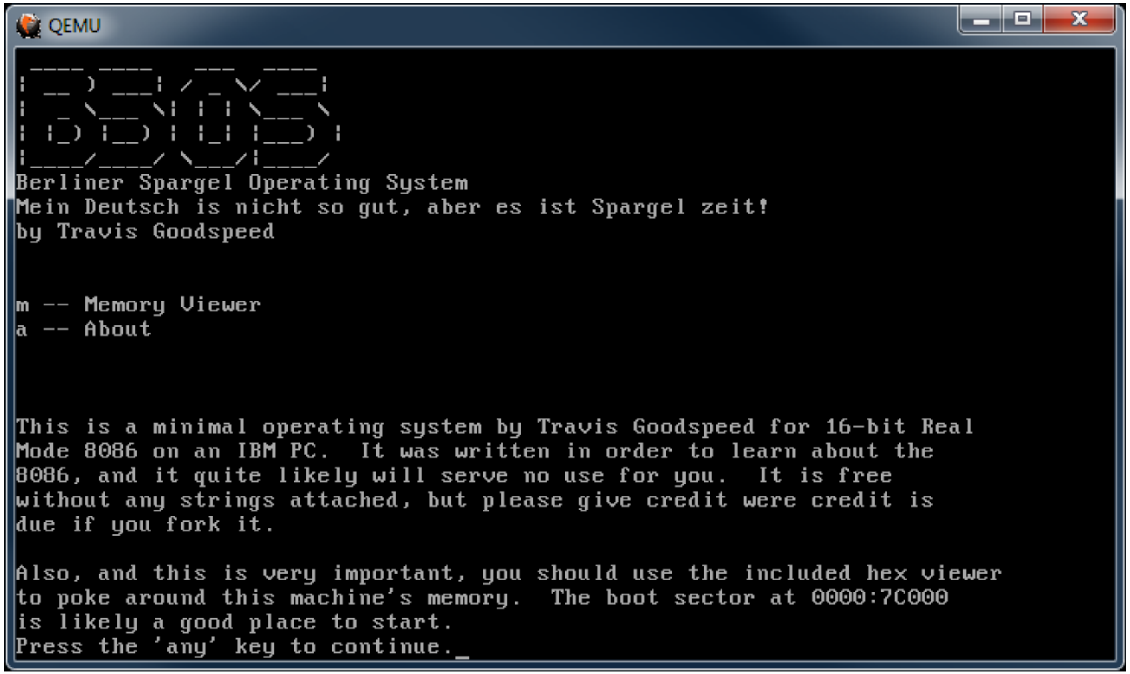
(deprecated)

PoC||GTFO

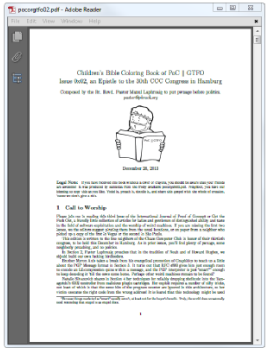
International Journal of Proof-of-Concept or Get The F*** Out

the “new” 2600 / Phrack...

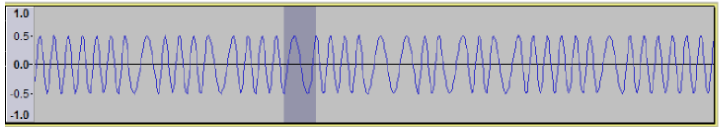
Distributed as **PDF** ⇒ each issue is a PoC

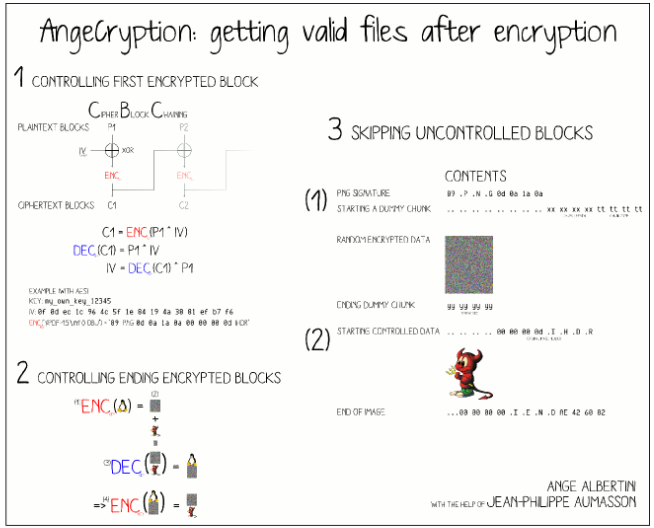


MBR (bootable) + PDF + ZIP




```
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   pgpquine/
      275       0      0  12/06/13 16:25   pgpquine/Makefile
     1006       0      0  12/06/13 16:25   pgpquine/inflate.c
     5323       0      0  12/06/13 16:25   pgpquine/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
```

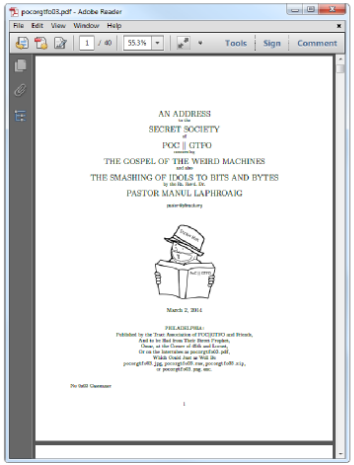




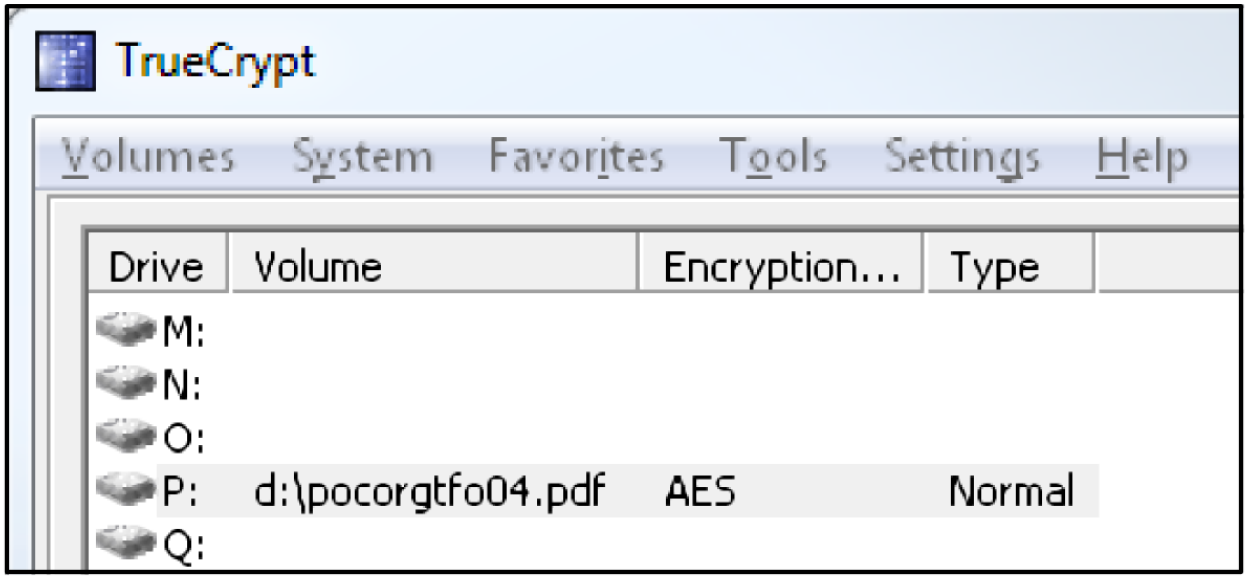
raw audio +

JPG + AES(PNG)

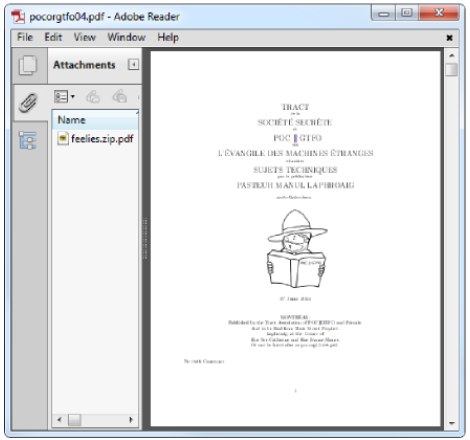
+ PDF + ZIP



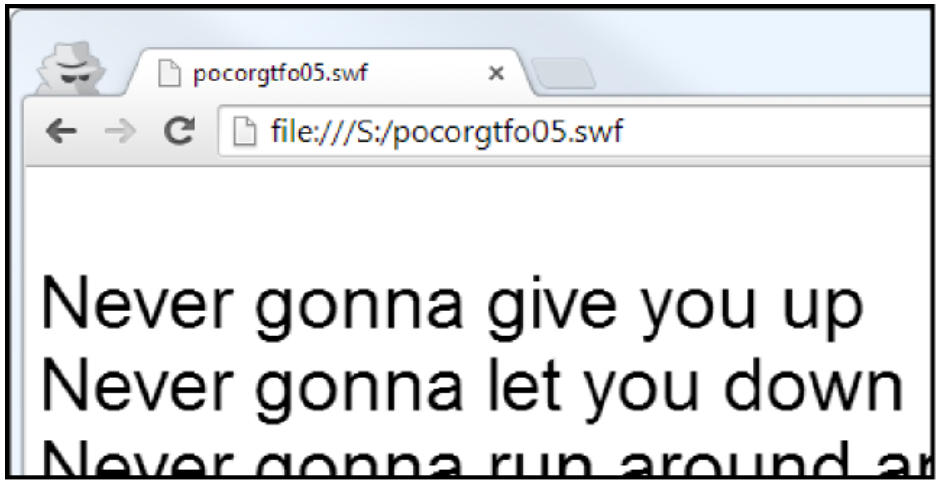

```
Archive:  pocorgtfo03.pdf
warning [pocorgtfo03.pdf]:  12224072 extra bytes at beginning or within zipfile
  (attempting to process anyway)
  Length      EAs    ACLs    Date    Time    Name
 --------     ---    ----    ----    ----    ----
     2561       0       0  02/10/14 06:23   alexander.txt
     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   nokiacipher.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   tetranglix.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
```

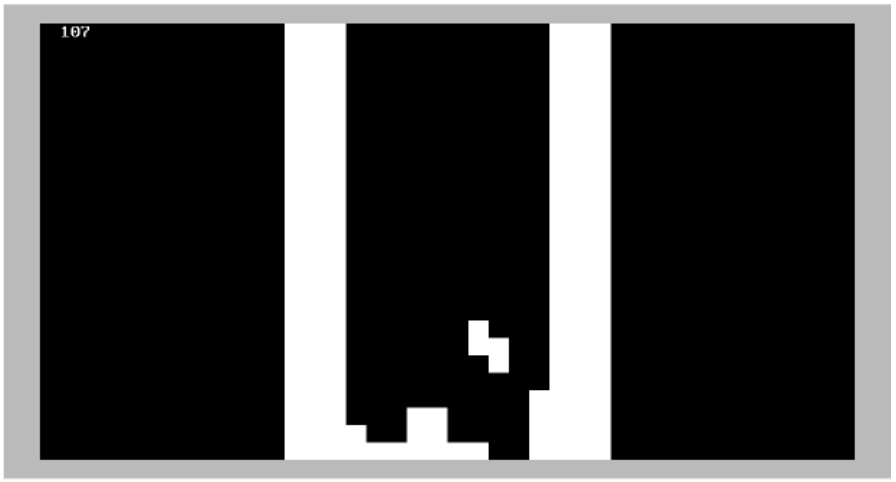


TrueCrypt
+ PDF + ZIP

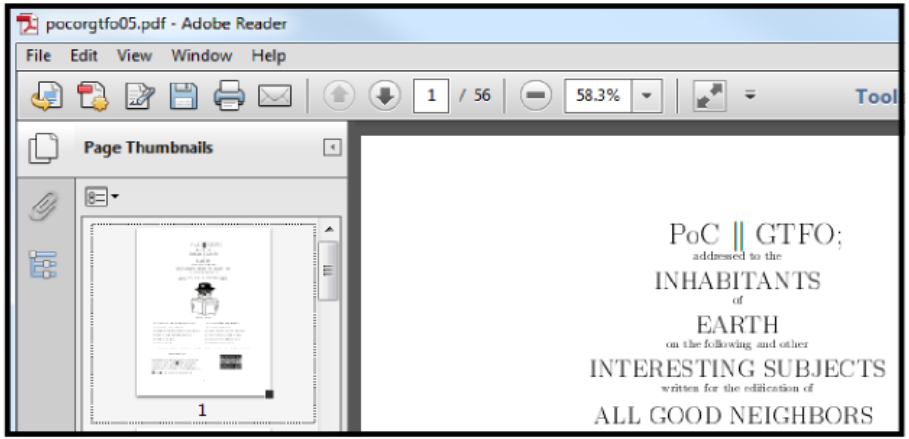


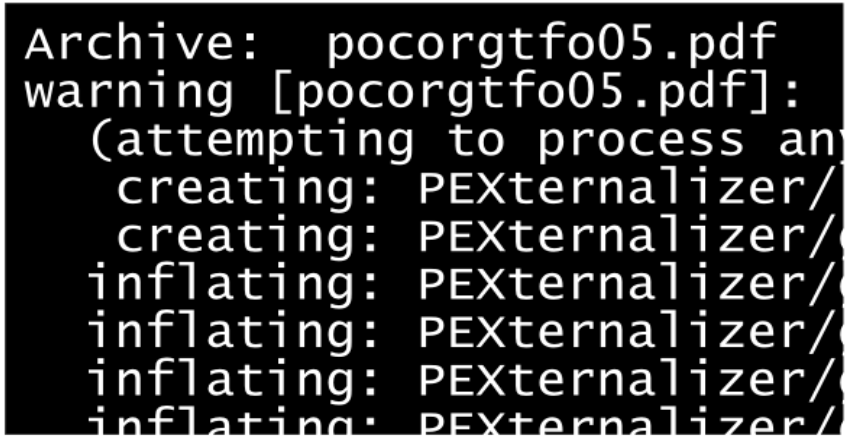

```
Archive:  pocorgtfo04.pdf
warning [pocorgtfo04.pdf]:  798586 extra bytes at beginning or within zipfile
  (attempting to process anyway)
error [pocorgtfo04.pdf]:  reported length of central directory is
  -798586 bytes too long (Atari STZip zipfile?  J.H.Holm ZIPSPLIT 1.1
  zipfile?).  Compensating...
  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

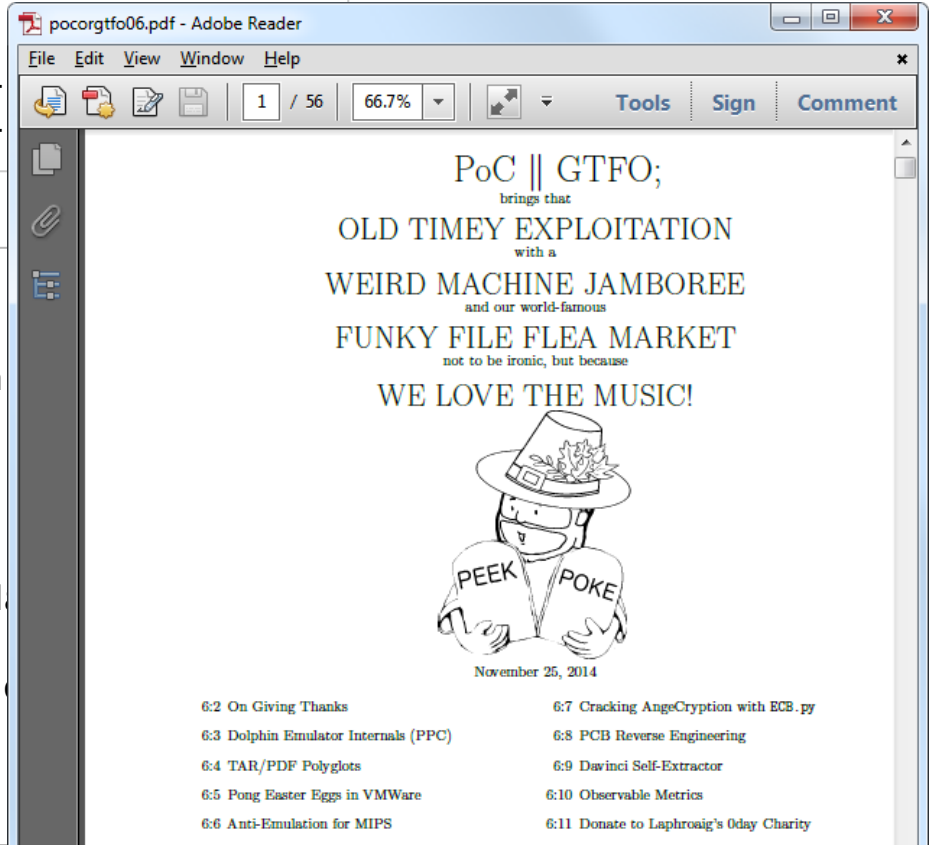




TAR + PDF + ZIP

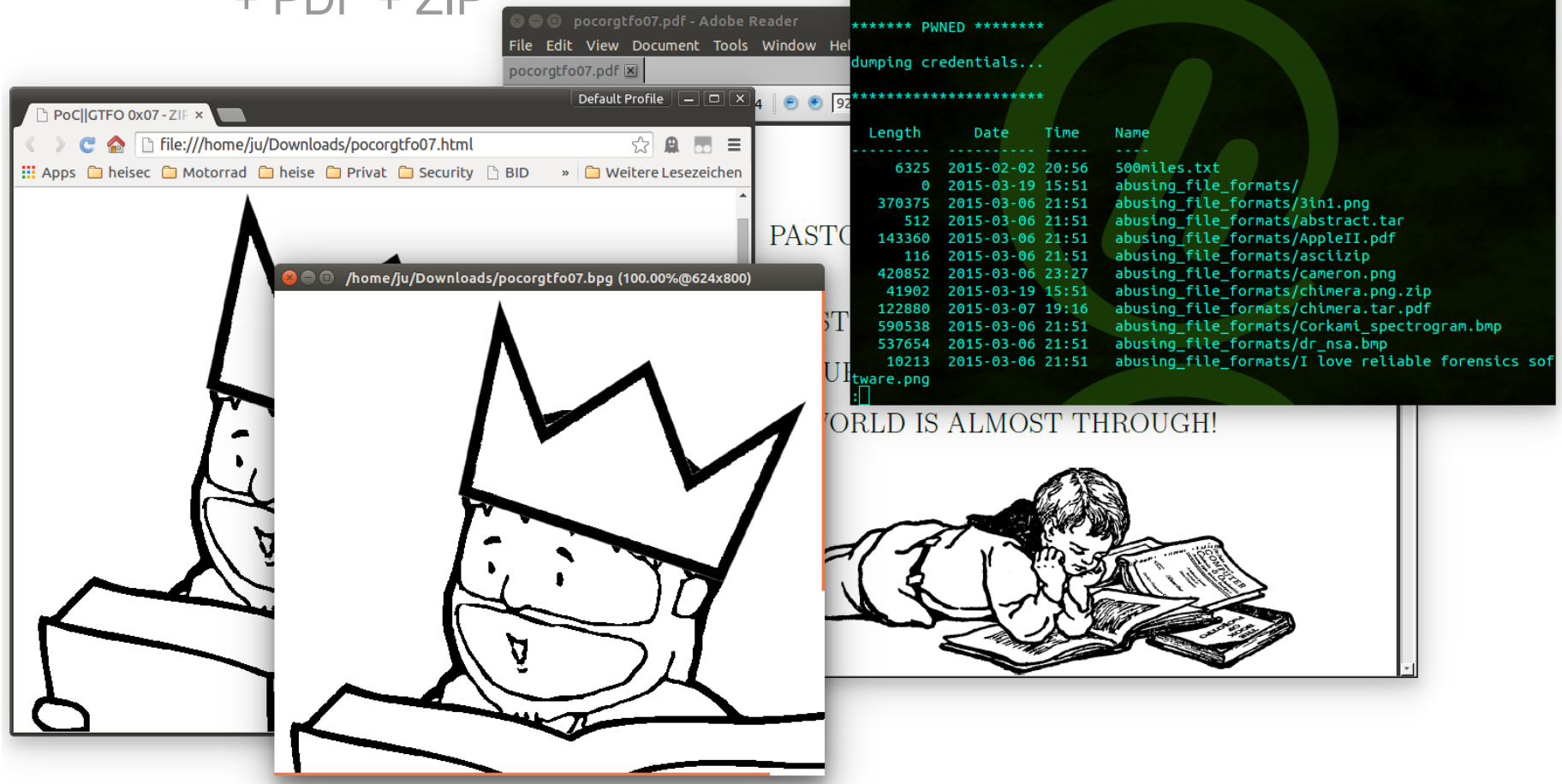
```
$ tar -tvf pocorgtfo06.pdf
-rw-r--r-- Manul/Laphroaig   0 2014-10-06
-rw-r--r-- Manul/Laphroaig 525849 2014-10-
-rw-r--r-- Manul/Laphroaig 273658 2014-10-
```

```
$ 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/
```



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

+ PDF + ZIP



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   nu
```

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

Preserving Document Forever

Posterity Depends on Forensics

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

Proven Dysfunctional Format

Please Don't Fail / Again

Polyglot (Definition|Deployment|Delivery) Framework

Pants-Down Format

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 :(

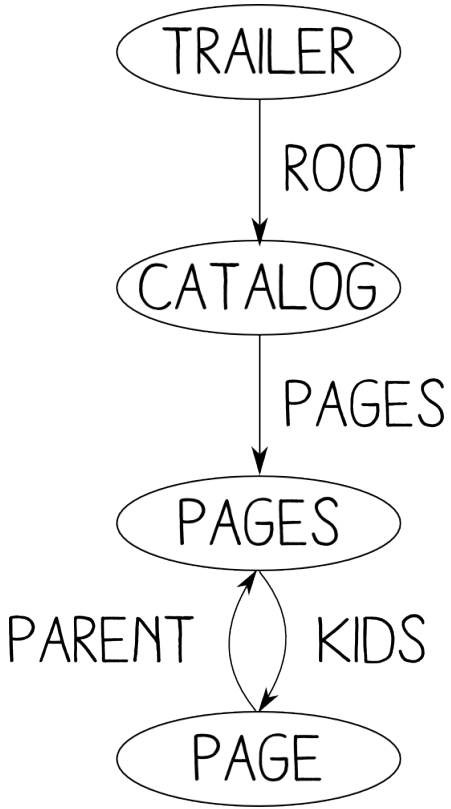
http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502

A strict format ?

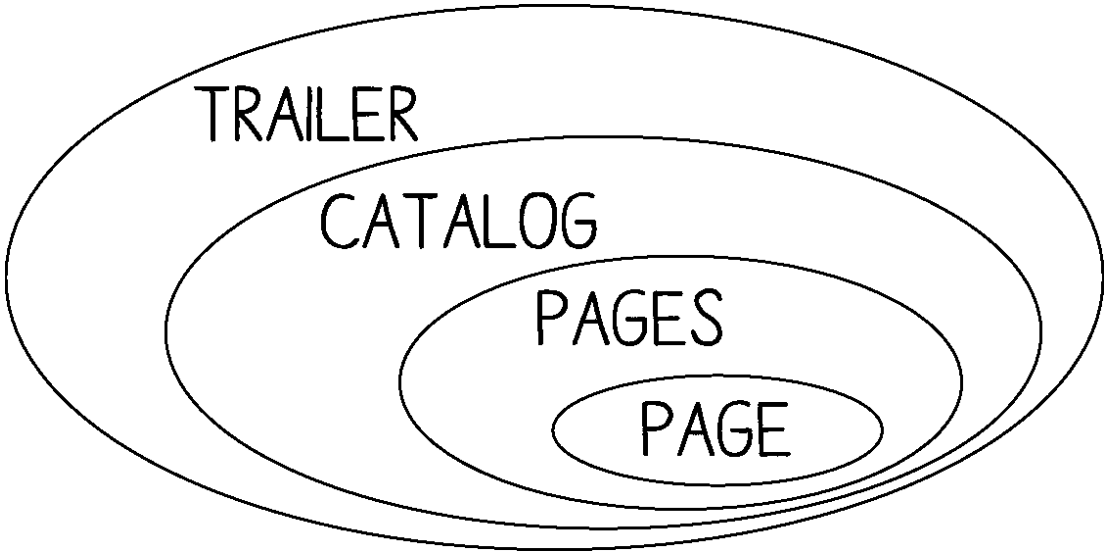
No reader completely enforces the specs

⇒ recovery mode (sometimes ‘explicit’)

signature, stream length, XREF…



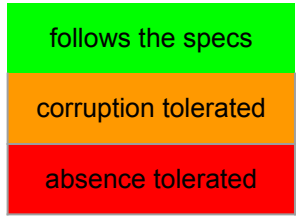
standard structure
(each object should be distinct)



non-standard but tolerated structure
(inlined objects)

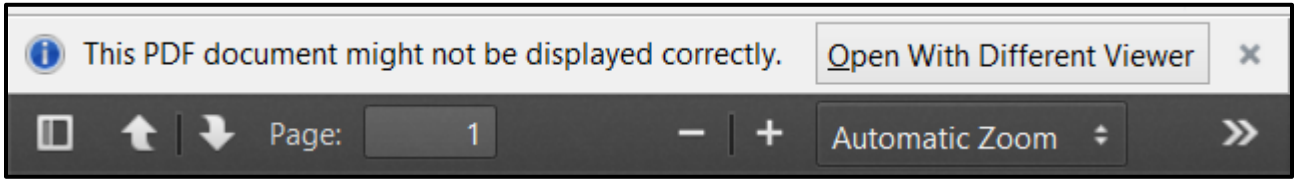
Many possible malformations handled specifically by each reader (high level)...

Many possible abuses



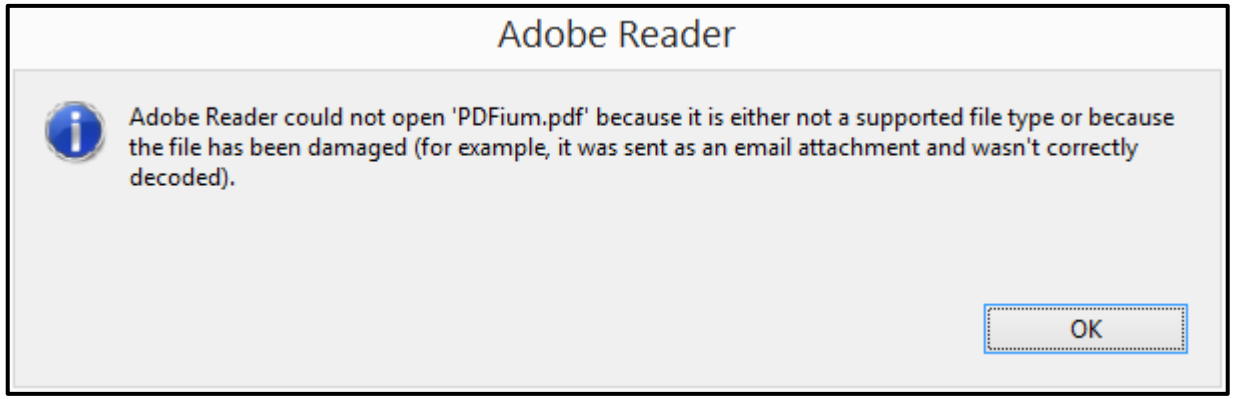
	signature	endobj	/Count	text operators	/Font	font use	xref	/Resources	trailer
Adobe Reader									
MuPDF									
PDF.js									
PDFium									
Poppler									

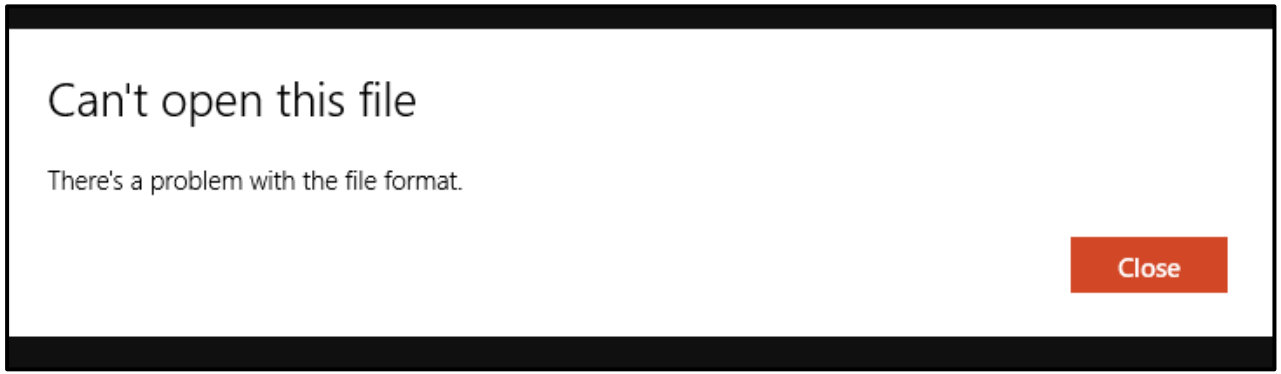
… different readers have different tolerances ...



Failed to load PDF document

Error loading D:\FireFox.pdf



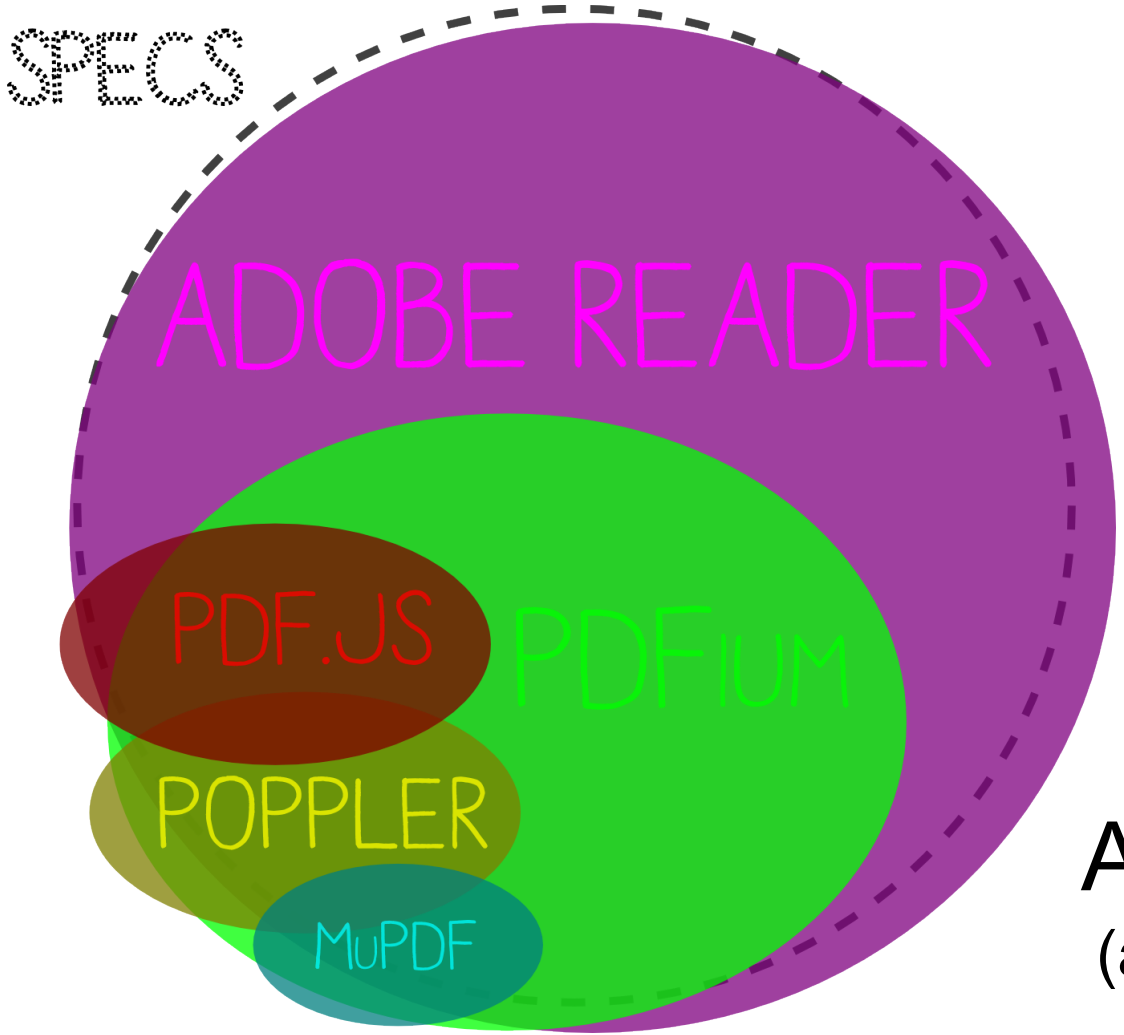


...so a PDF specifically crafted for one reader, may fail with all other readers.

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



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

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38920

“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...

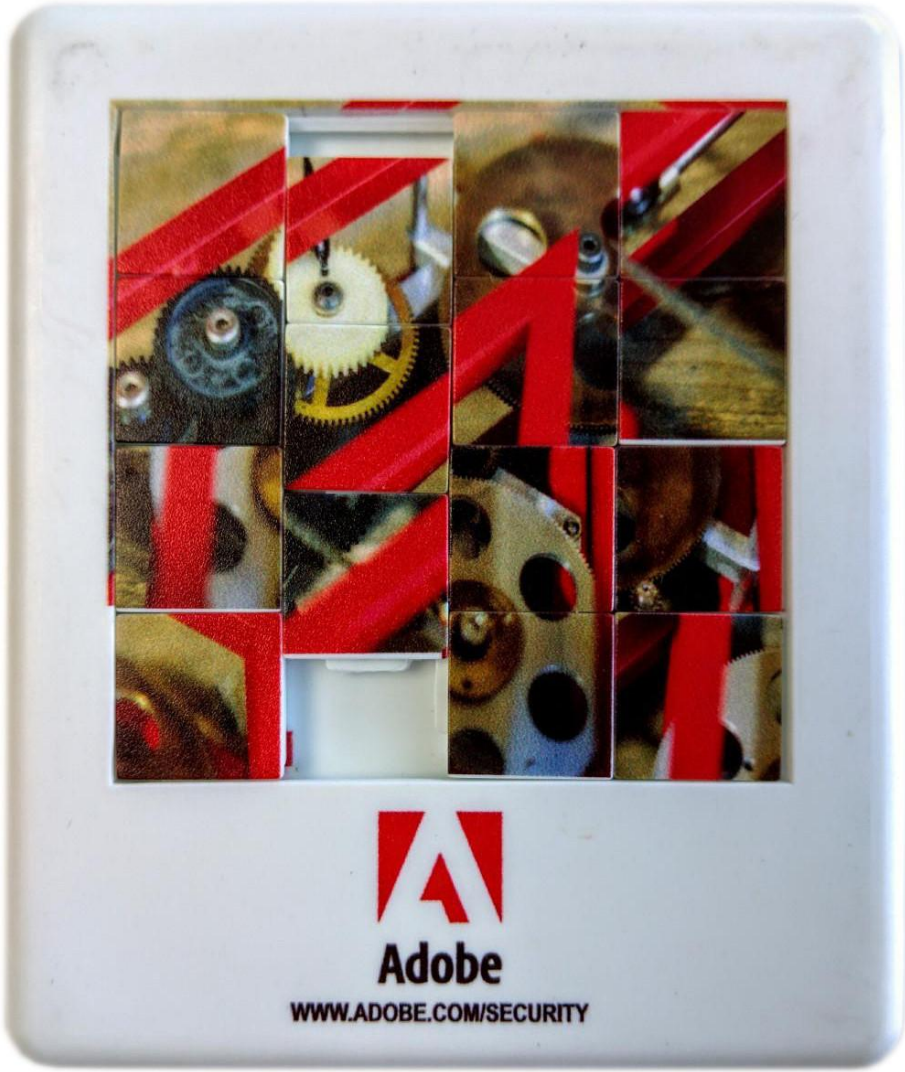
PDF 2.0 is coming!

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

*https://www.youtube.com/watch?v=wGmcTf-uMrE

Conclusion

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



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”

*https://www.youtube.com/watch?v=FVBSvjYQgq8

ACK

Paul Wheatley
@doegox @pdfkunfoo @newsoft
@internot @insertscript @avlidienbrunn
@foxgrrl @chrisjohnriley @travisgoodspeed

and everybody for the PDF suggestions :)

@angealbertini

corkami.com