

PDF

Myths

FACTS



# 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

```
DICTIONARY << 1 0 obj
<< OBJECT NUMBER >> REVISION NUMBER > R
<< /Pages 2 0 R
>> IDENTIFIER (WITH /)
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
```

STREAM PARAMETERS:

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

BEGIN TEXT  
FONT F1 (Arial) SET TO SIZE 110  
MOVE TO COORDINATE 10,400  
OUTPUT TEXT 'HELLO WORLD!'  
END TEXT

## BODY

## XREF TABLE

CROSS REFERENCE	xref	CROSS REFERENCES
	5 0	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-1.1 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 <sub>1</sub> y <sub>1</sub> x <sub>2</sub> y <sub>2</sub> x <sub>3</sub> y <sub>3</sub>	c cubic bezier
x <sub>2</sub> y <sub>2</sub> x <sub>3</sub> y <sub>3</sub>	v cubic bezier
x <sub>1</sub> y <sub>1</sub> x <sub>3</sub> y <sub>3</sub>	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

w	set line width
J	set line cap
j	set line join
M	set miter limit
dashArray dashPhase d	set line dash
intert	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 matrices
T*	move to next line + reset leading

### showing

string	Tj show string
string	' move to next line + show string
a <sub>w</sub> a <sub>c</sub> 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  
COOKIES

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.

```
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 >>
>>
```

**INVALID?**

truncated signature

missing parent /Type  
/Kids should be indirect

missing /Font  
missing kid /Type  
missing /Count

missing endobj

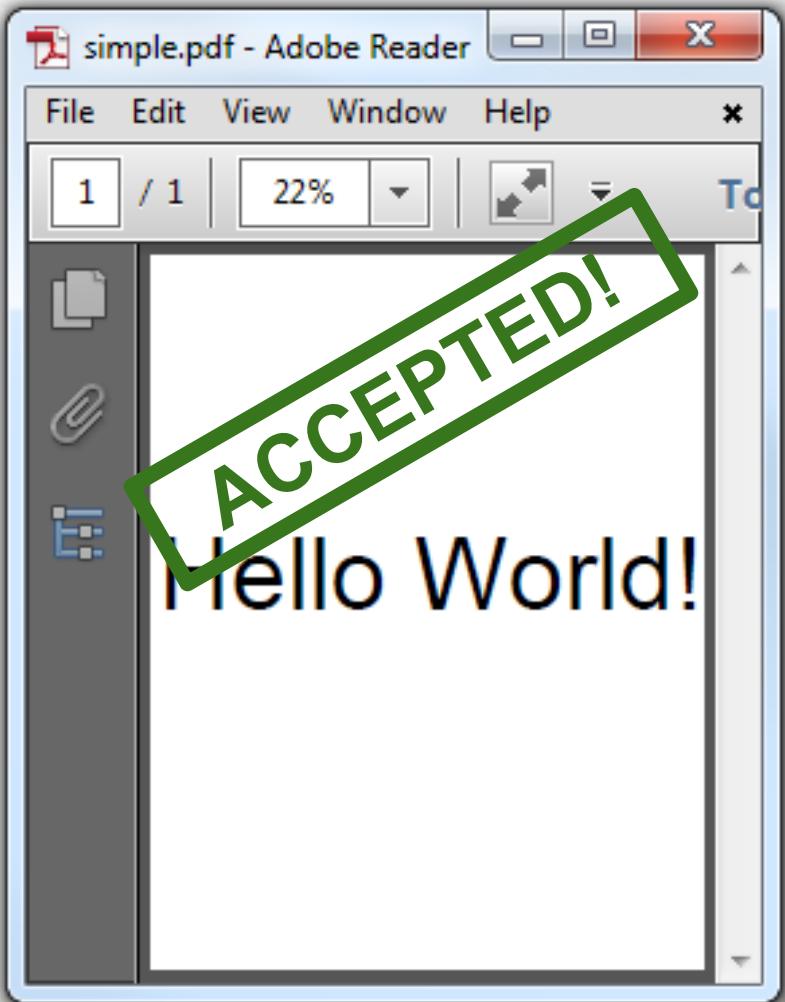
missing /Length

missing xref

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

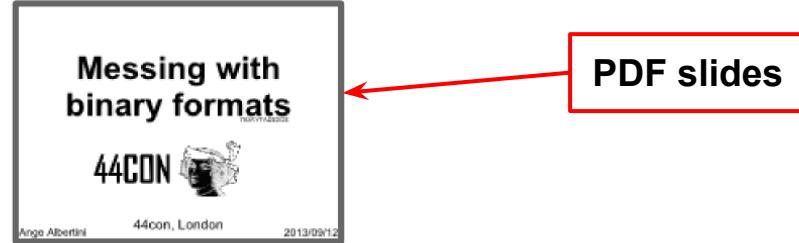
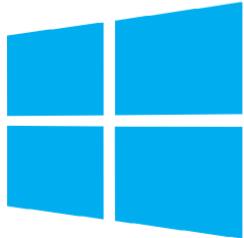
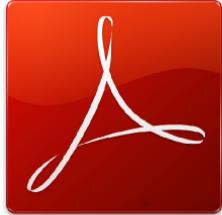
**It's *not*  
standard...**

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

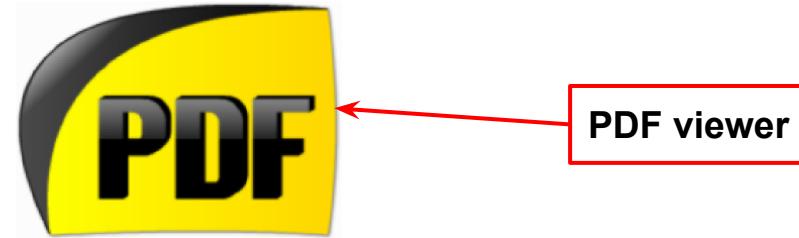


# **Binary art**

PDF + creativity = ... ?



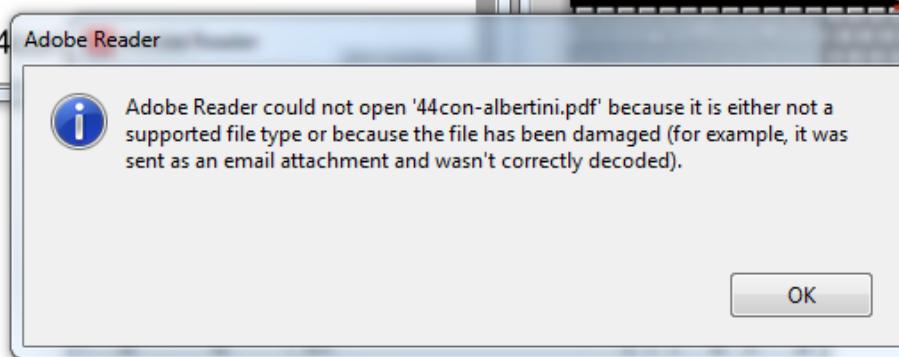
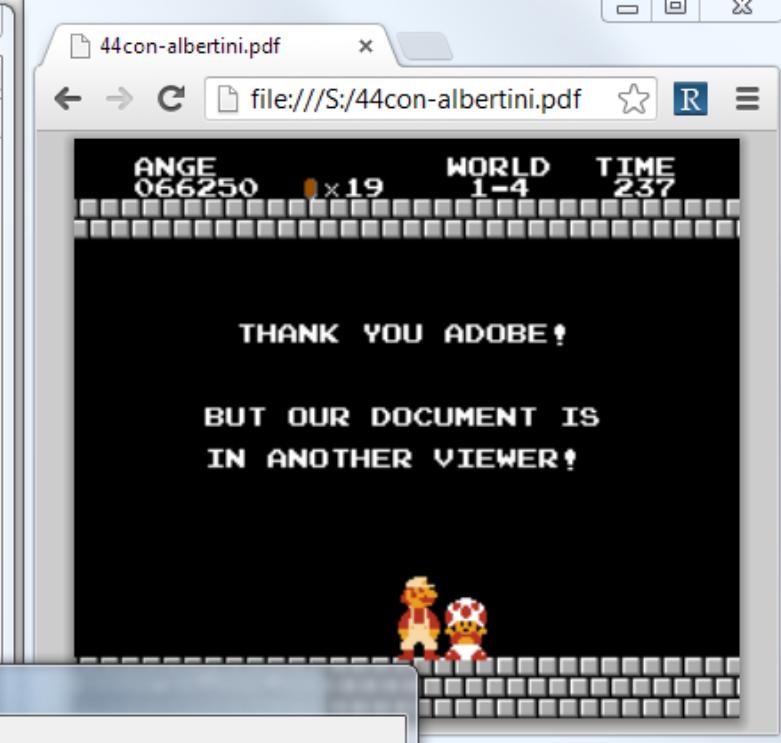
PDF slides



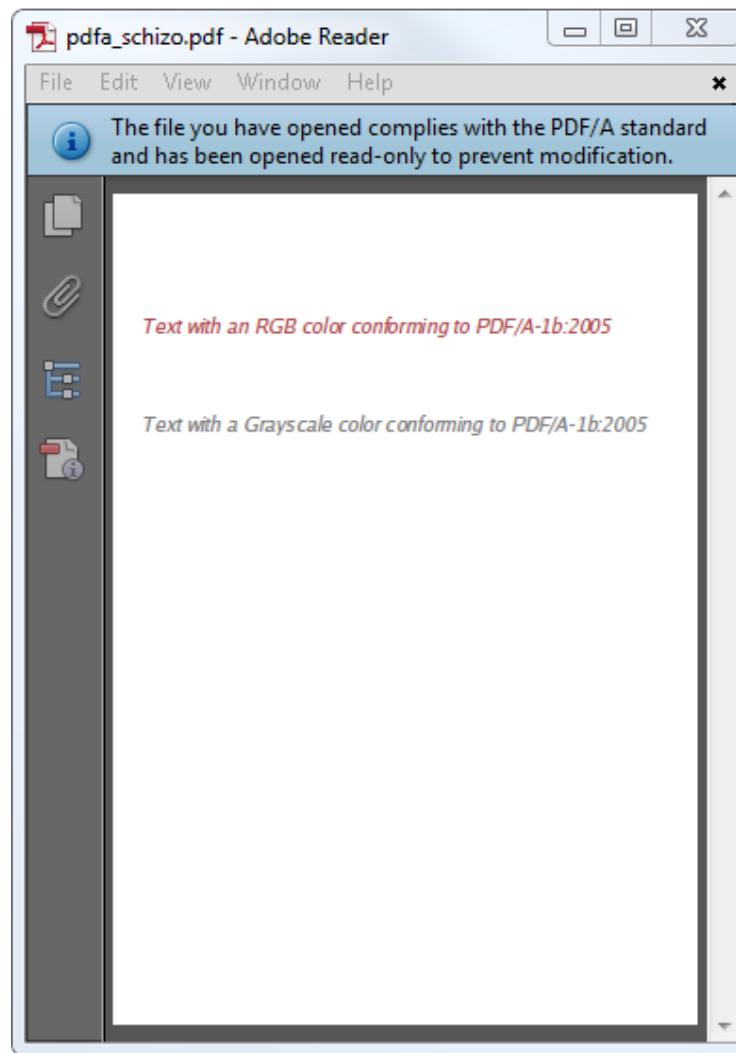
PDF viewer



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)

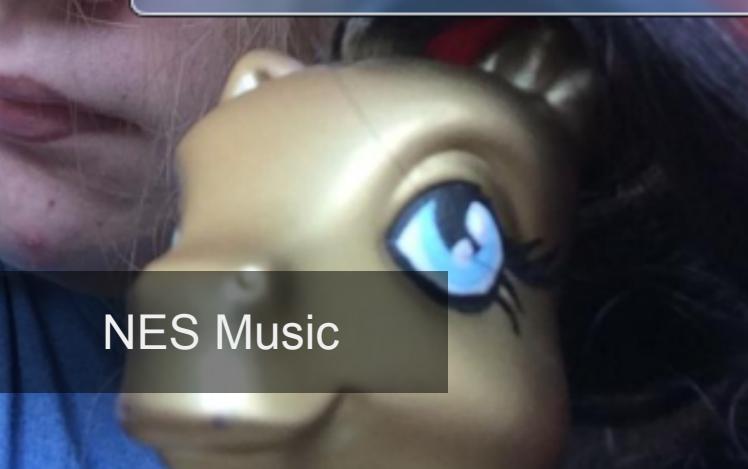


(Also available in PDF/A flavour)

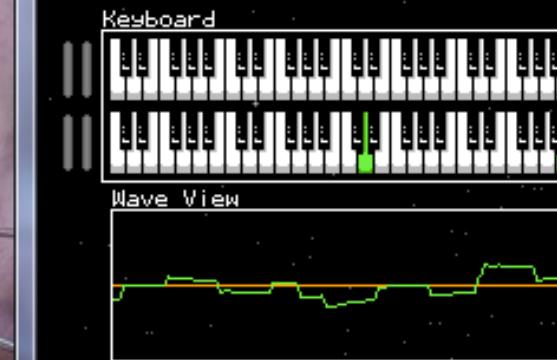
# The SSL Smiley Song

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



**Title : "SSL Smiley Song :-)"**

Artist: "Melissa Elliott"

(C) : "2014 0xabad1dea"

No : 00 VRC6 VRC7 FDS MMC5 N106 SN5E

## Keyboard



Wave View

# Super NES Megadrive

snes\_md.pdf - Adobe Reader

File Edit View Window Help

# GENESIS DOES WHAT NINTENDON'T.

Get the hottest new video games going. Arcade, sports, adventure, strategy and action hits available only on the 16-bit Genesis System by Sega®

Today's latest blockbuster arcade hits like Super Monaco GP™. Climb into the cockpit of the world's fastest Grand Prix machines as you race wheel to wheel through the streets at over two-hundred miles per hour. Or take on the evil villain Mr. Big in Michael Jackson's Moonwalker™ as you use dance-kicks, hat-tricks and finally transform into a powerful robot that does it all. Or become a Cybercop in E-SWAT™ and clean up the city besieged by mad terrorists.

Get ready for the most action-packed sports games ever. In Joe Montana Football™, check our the defense, make the call, fake a pass and scramble for a

SNES - snes md

File Emulation View Config  
Tools SNES Help



Genesis - snes md

File Emulation View Config Tools  
Genesis Help

SCORE 72 HIGH SCORE 72

4	8	8
16		
2		

printme.pdf - Adobe Reader

File Edit View Window Help

Open Tools Fill &

1 / 1 101% Tools

Print

Printer: Microsoft XPS Document Writer Properties Advanced Help ?

Copies: 1 Print in grayscale (black and white)

Pages to Print: All

Current page Pages 1 More Options

Page Sizing & Handling: Fit

Actual size Shrink oversized pages Custom Scale: 100 % Choose paper source by PDF page size

Orientation: Auto portrait/landscape Portrait Landscape

11.69 x 8.27 Inches

NATIONAL SECURITY AGENCY  
UNITED STATES OF AMERICA

Page Setup... Page 1 of 1 Print Cancel

The screenshot shows the Adobe Reader interface with a document open. The document features a large red background with the white text 'RSA' and 'SECURITY' below it. On the left side of the reader window, there is a toolbar with various icons, one of which is highlighted with a red box. The main content area displays the document's text. Overlaid on the document is the 'Print' dialog box. In the 'Print' dialog, the printer is set to 'Microsoft XPS Document Writer'. The 'Copies' field is set to 1. Under 'Pages to Print', the 'All' option is selected. In the 'Page Sizing & Handling' section, 'Fit' is chosen. The 'Orientation' is set to 'Auto portrait/landscape'. The 'Comments & Forms' section shows 'Document and Markups' selected. The 'Scale' is set to 166%. The preview area shows the RSA logo and the NSA seal. At the bottom of the dialog, the status 'Page 1 of 1' is visible along with 'Print' and 'Cancel' buttons.

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 ;)

zipjpg.pdf - WinRAR

File Commands Tools Favorites Options Help

zipjpg.pdf - SFX ZIP archive, unpacked size 69,782 bytes

Name	Size	Pac...	Type	CRC32
..			Folder	
corkami.jpg	69,782	69,782	JPEG Image	2A142635

**Test finished**

No errors found during test operation

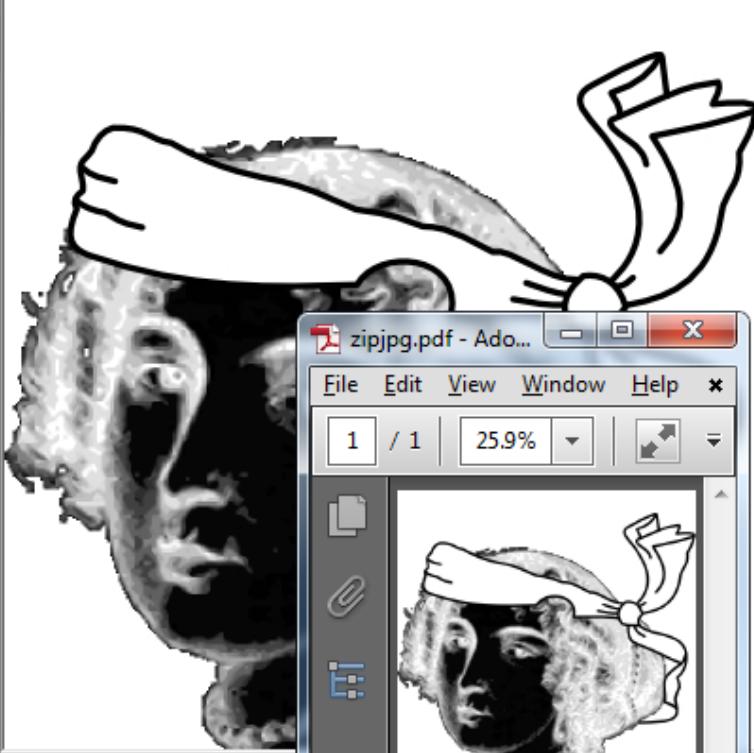
OK

Selected 69,782 bytes in 1 file Total 69,782 bytes in 1 file

endstream  
endobj  
  
xref  
0 1  
000000000000 65535 f  
000000000010 00000 n  
  
trailer  
<</Root 1 0 R>>  
  
startxref  
70488  
%%EOF  
ÿÙ

zipjpg.pdf - IrfanView

File Edit Image Options View Help



zipjpg.pdf - Ado...

File Edit View Window Help

1 / 1 25.9%

400 x 400 x 24 BPP 1/1 100

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

C:\Windows\System32\cmd.exe

```
d:\>pdflatex quine.pdf
This is pdfTeX, Version 3.1415926-2.5-1.40.14 (MiKTeX 2.9)
entering extended mode
<d:\quine.pdf
```

```
\LaTeX2e
Babel <
abics, a
croatia
lician,
ic, ind
lithua
german-
ian, ru
an, tam
ishmax,
C:\Pr
Documen
C:\Pr
C:\Pr
C:\Pr
C:\Pr
C:\Pr
```

quine.pdf - Adobe Reader

File Edit View Window Help

# a PDFLATEX polyglot+quine

by Ange Albertini - suggested by Philipp Jovanovic

September 8, 2014

This is a PDF+T<sub>E</sub>X polyglot+quine (not fully standard) generated by PDFLATEX:  
you can generate the PDF from the PDF+T<sub>E</sub>X itself via PDFLATEX directly.

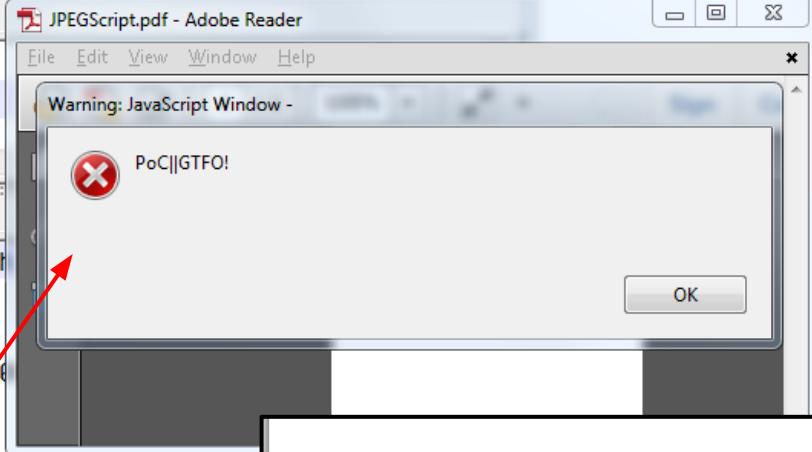
How to:

8.50 x 11.00 in

PDFLaTeX quine (the document is its own source)

JPEGScript.pdf

```
trailer << /Size 2 /Root <</Pages <>>
/OpenAction << /S/JavaScript /JS 1 0 R
>>
>>
JPGScript.pdf
1 0 obj<</Filter[/ASCIIHexDecode/DCTDecode]/Width
/ColorSpace/DeviceGray >>
stream
ffd8ffe000104a46494600010100000100010000ffdb0043e
endstream
endobj
```



script == picture

JPEGScript  
app.alert ("PoC||GTFO!");

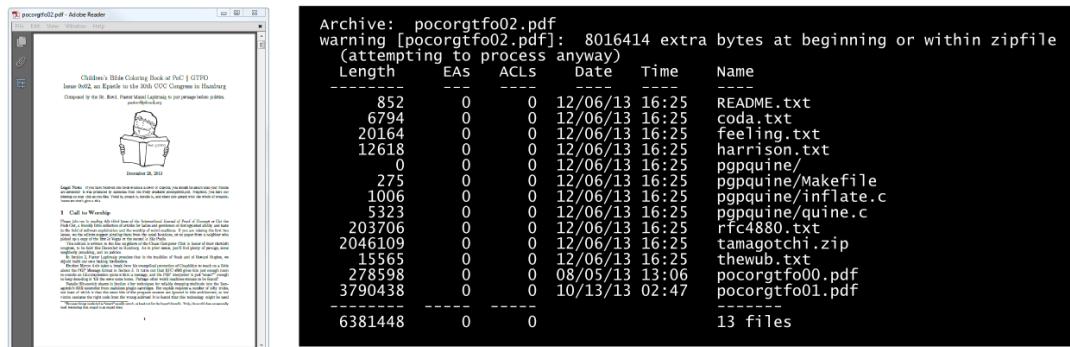
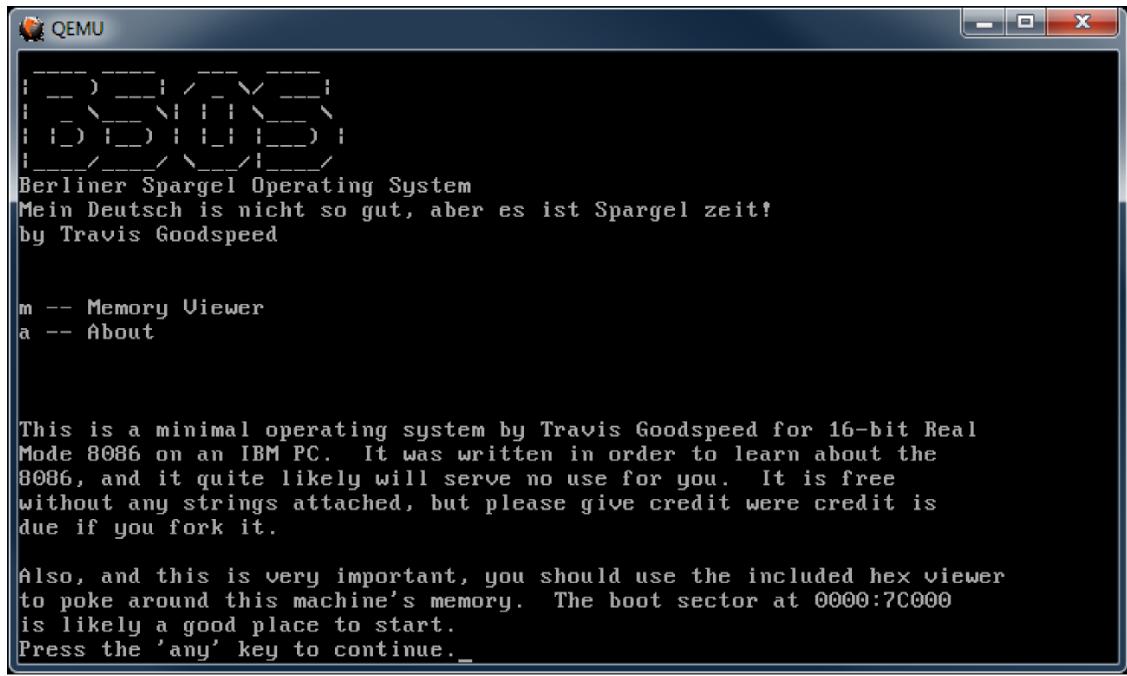
# JPEG-encoded JavaScript (deprecated)

# 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



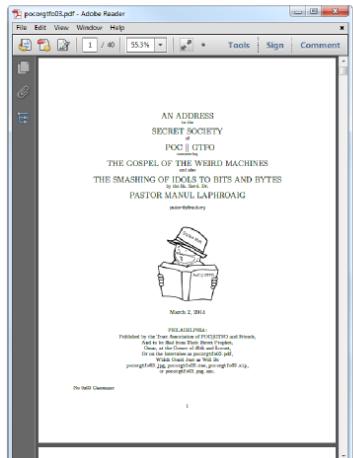
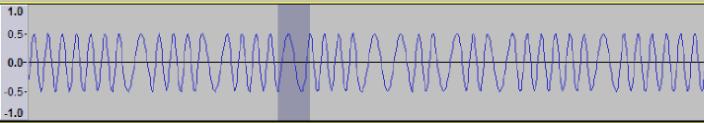
MBR (bootable)  
+ PDF + ZIP

FILE

```
00000: ff d8          'START OF IMAGE' MARKER
00002: ff e0<size.16> <content>      'APP0' MARKER (REQUIRED) (HEADER)
00014: ff fe <size.16>      'COMMENT' MARKER
+4: xPDF-1.5           COMMENT CONTENT
    999 8 obj
    <>>
    stream

00039: ...             (OTHER MARKERS, ORIGINAL JPEG DATA)

xx   : ff d9          'END OF IMAGE' MARKER
xx+2 : endstream
endobj
```

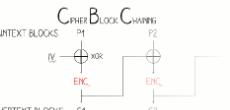


JPEG

PDF



## 1 CONTROLLING FIRST ENCRYPTED BLOCK



$$\begin{aligned} C1 &= \text{ENC}(P1 \wedge IV) \\ \text{DEC}(C1) &= P1 \wedge IV \\ IV &= \text{DEC}(C1) \wedge P1 \end{aligned}$$

```
EXAMPLE WITH AESI  
KEY: my_own_key_12345  
V:0F Bd ee 1c 96 4c 5f 1e 84 19 4a 38 81 ef b7 f6  
D:\C:\PDF\MSWin008.DJ\99.PDF Bd ee 1a 8a 00 00 00 00 00 00 HDR'
```



$$\text{ENC}_{\text{key}}(\Delta) = \begin{matrix} \text{red box} \\ + \\ \text{blue box} \\ = \\ \text{green box} \end{matrix}$$

## UNCONTROLLED BLOCKS



(1) PNG SIGNATURE  
STARTING A DUMMY CHUNK



(2) STARTING CONTROLLED DATA . . . . . 98 98 98 9d .I .H .D .  
CONTINUATION



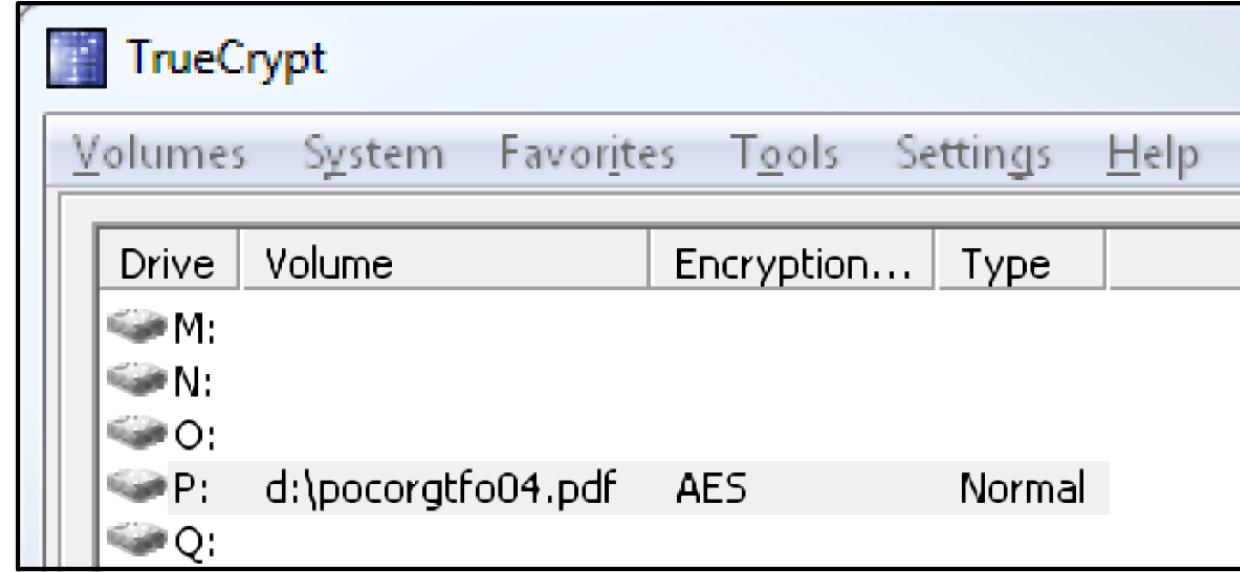
... 99 99 99 99 T E N D R E 42 68 1

ANGE ALBERTIN  
WITH THE HELP OF JEAN-PHILIPPE AUMASSON

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

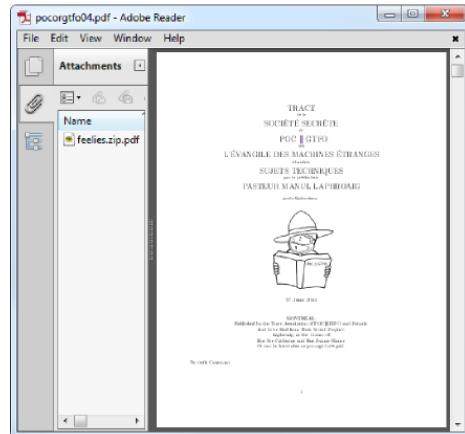
Length	EAs	ACLS	Date	Time	Name
2561	0	0	02/10/14	06:23	alexander.txt
7848	0	0	02/08/14	20:20	bochs-2.6.2.patch
6135	0	0	02/08/14	20:21	bochs-20140203.patch
7248	0	0	02/09/14	08:35	defusing.zip
4830	0	0	12/01/13	15:48	despair.txt
14892	0	0	11/27/13	19:03	lasta.txt
26325	0	0	02/07/14	21:06	lastq.txt
473449	0	0	02/07/14	21:06	netwatch-337f8b1.tar.g
131930	0	0	02/24/14	20:32	nokicipher.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

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

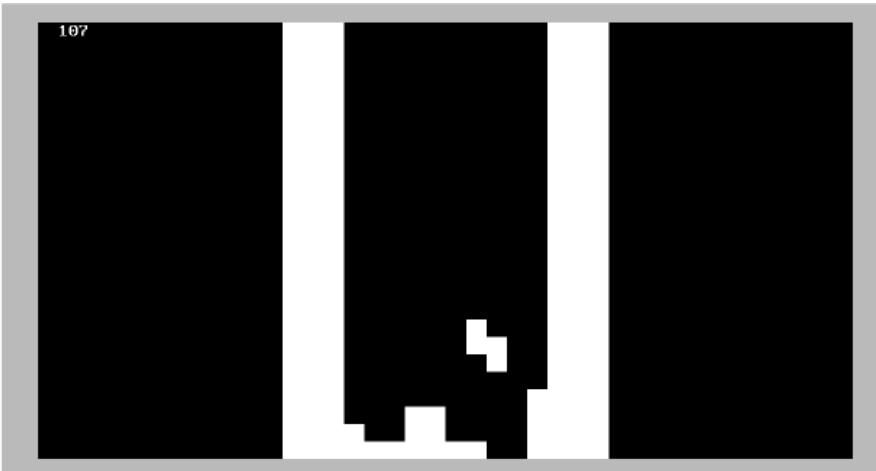
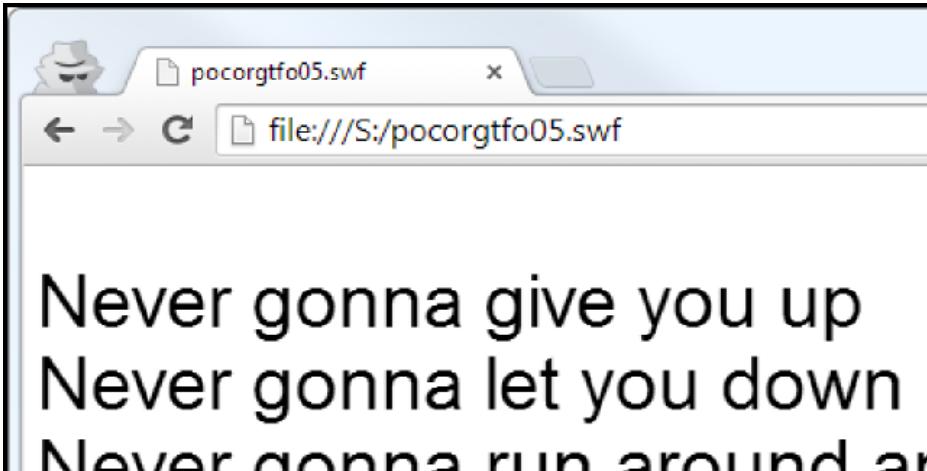


# TrueCrypt

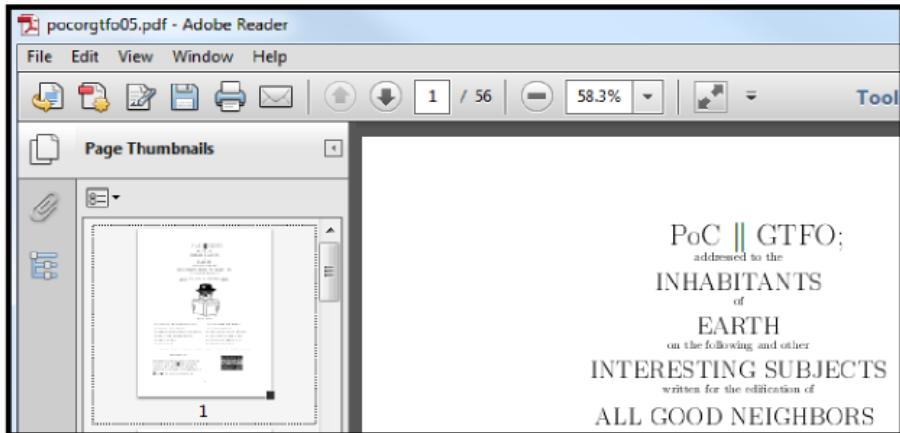
## + PDF + ZIP



```
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 ZIPSSPLIT 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  tsb20140401.zip
26750864        0        0 06/24/14 18:56  pocorgtfo03.pdf
-----  -----
 30353424        0        0                               15 files
```



Flash + bootable ISO + PDF + ZIP



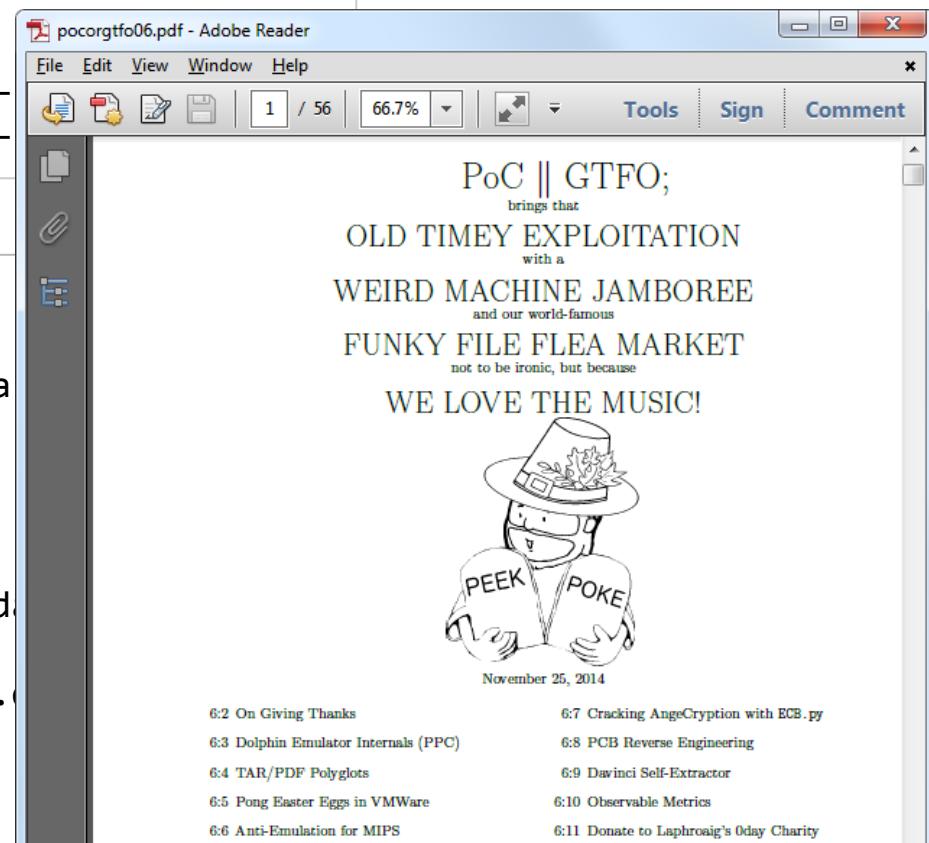
Archive: pocorgtfo05.pdf  
warning [pocorgtfo05.pdf]:  
(attempting to process an)  
creating: PEXternalizer/  
creating: PEXternalizer/  
inflating: PEXternalizer/  
inflating: PEXternalizer/  
inflating: PEXternalizer/  
inflating: PEXternalizer/

# 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-
```

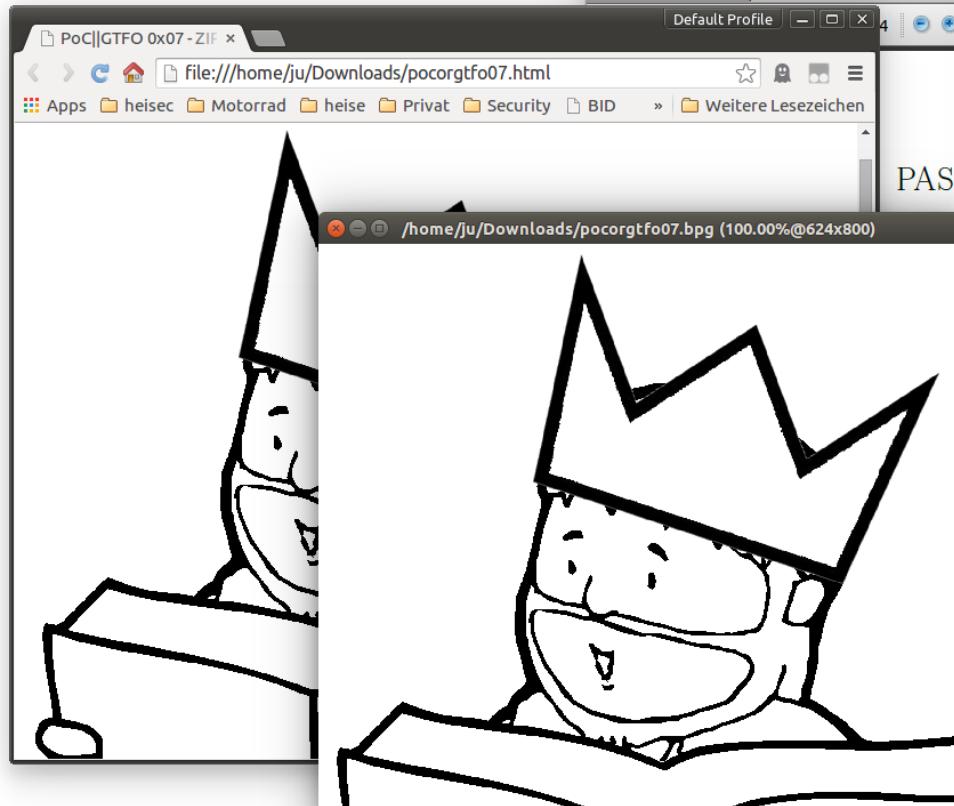
```
$ unzip -l pocorgtfo06.pdf
Archive: pocorgtfo06.pdf
warning [pocorgtfo06.pdf]: 10672929 extra
(attempting to process anyway)
```

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

A terminal window titled "ju@loki: ~" showing the contents of a ZIP archive named "pocorgtfo07.zip". The terminal output includes:

```
***** PWNED *****
dumping credentials...
*****
Length      Date    Time   Name
-----
6325 2015-02-02 20:56
0       2015-03-19 15:51
370375 2015-03-06 21:51
512    2015-03-06 21:51
143360 2015-03-06 21:51
116    2015-03-06 21:51
426852 2015-03-06 23:27
41902 2015-03-19 15:51
122880 2015-03-07 19:16
596538 2015-03-06 21:51
537654 2015-03-06 21:51
10213 2015-03-06 21:51
:[]
```

Below the terminal window is a cartoon illustration of a person lying down, reading several books.

# 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	Name
988446	0	0	08/06/15	22:46	ED
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
Posterity Depends on Forensics	<b>Preserving Document Forever</b>	Pathetic & Dangerous Format
Perversely Designed Format	Posthoc Depression Format	
Proprietary Document Fee		Penile Dysfunction Format
Postscript Didn't Fit	Public Domain Farce	
Please Don't Fail / Again	Penetrate Dodgy Firewall	Proven Dysfunctional Format
Pants-Down Format	Polyglot (Definition Deployment Delivery) Framework	
Perpetually Disagreeable Format		PDF is a Disaster for the Future
...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?

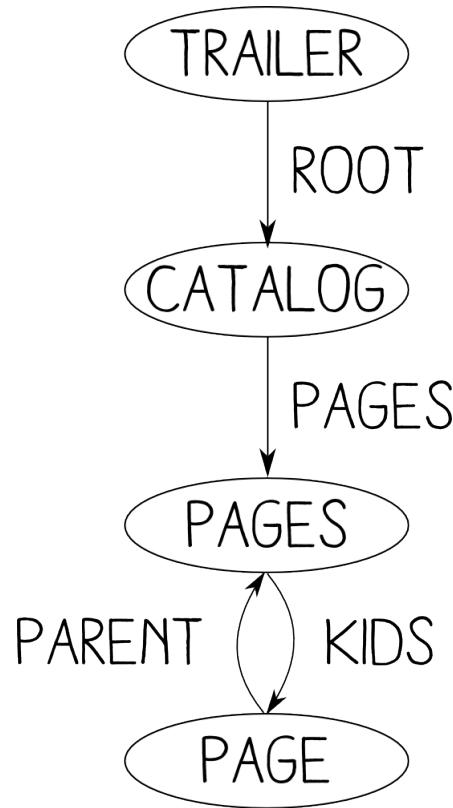
ISO \$pec\$ = 200\$

These specs only cover the main part :(

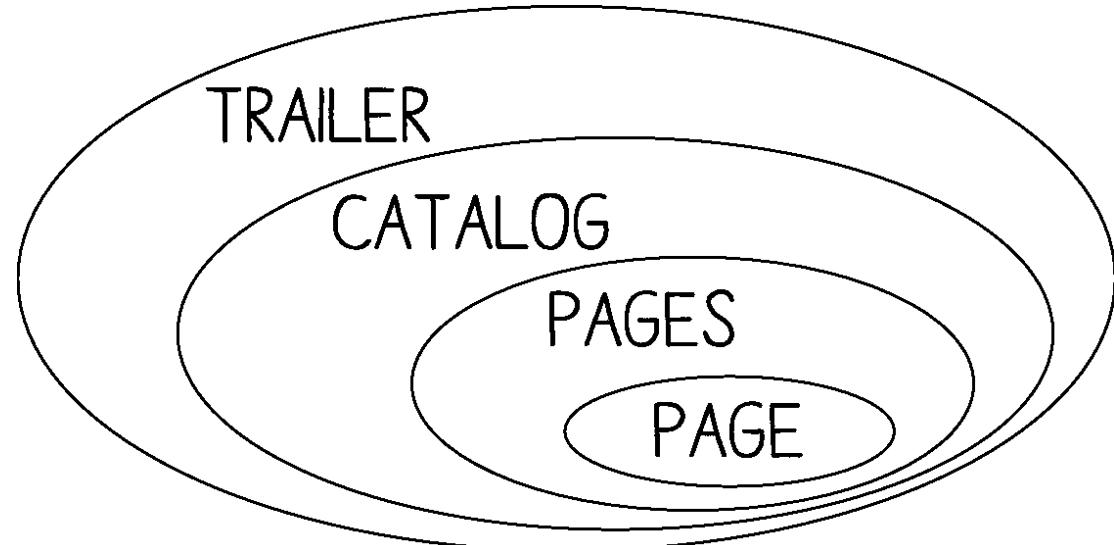
They are unclear - no formal guarantee :(

# A strict format ?

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



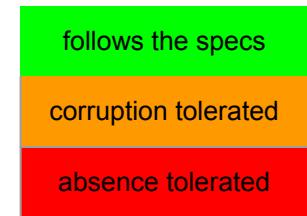
standard structure  
(each object should be distinct)



non-standard but tolerated structure  
(inlined objects)

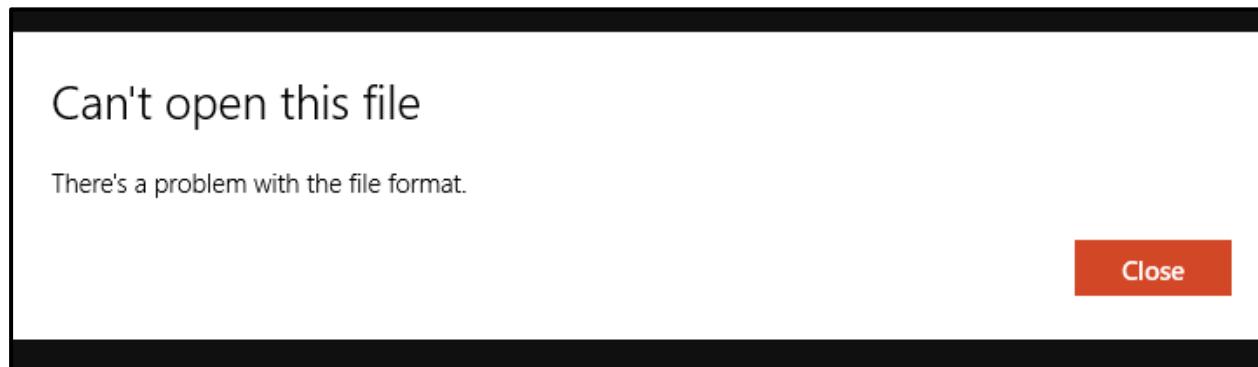
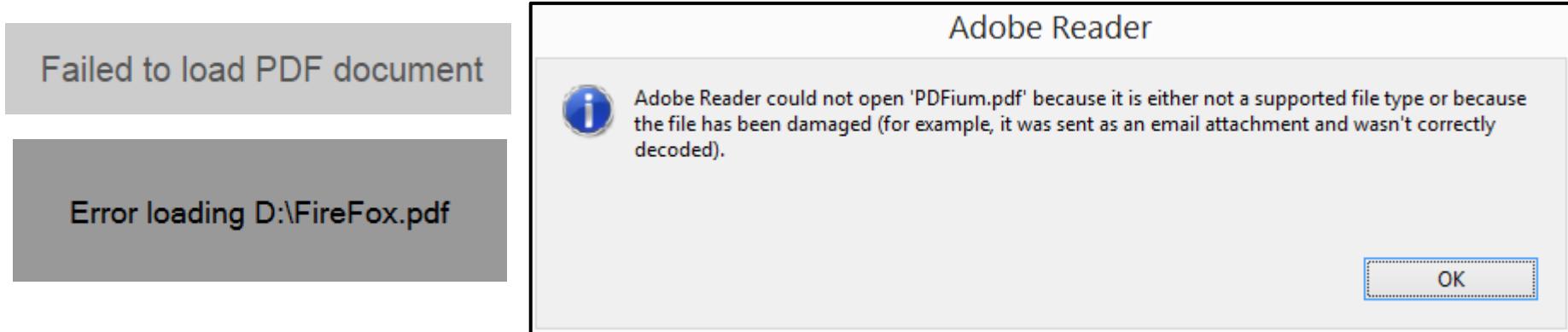
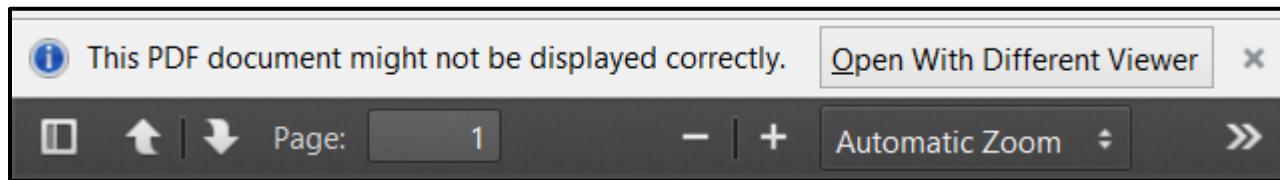
Many possible malformations handled specifically by each reader (high level)...

# Many possible abuses



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

... different readers have different tolerances ...



...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

ADOBED READER

PDF.JS

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

