

# Trusting files (and their formats)



Ange Albertini - Hack.Lu 2015



# ANGE ALBERTINI

reverse engineering

VISUAL DOCUMENTATION

@angealbertini

ange@corkami.com

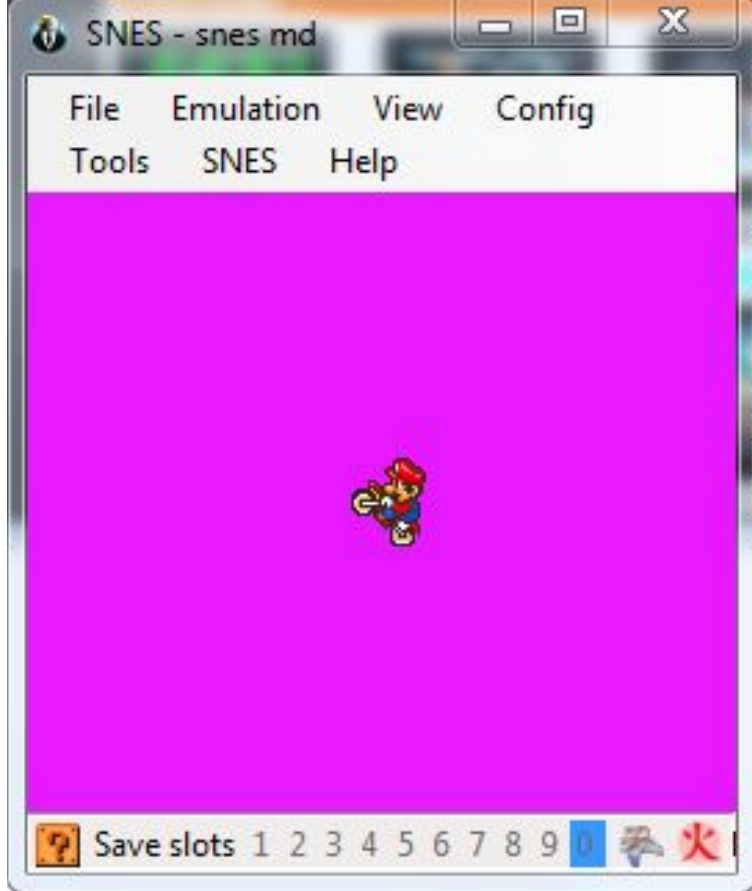
<http://www.corkami.com>



Welcome to my talk!



My resume is a PDF. What could go wrong ? ;)

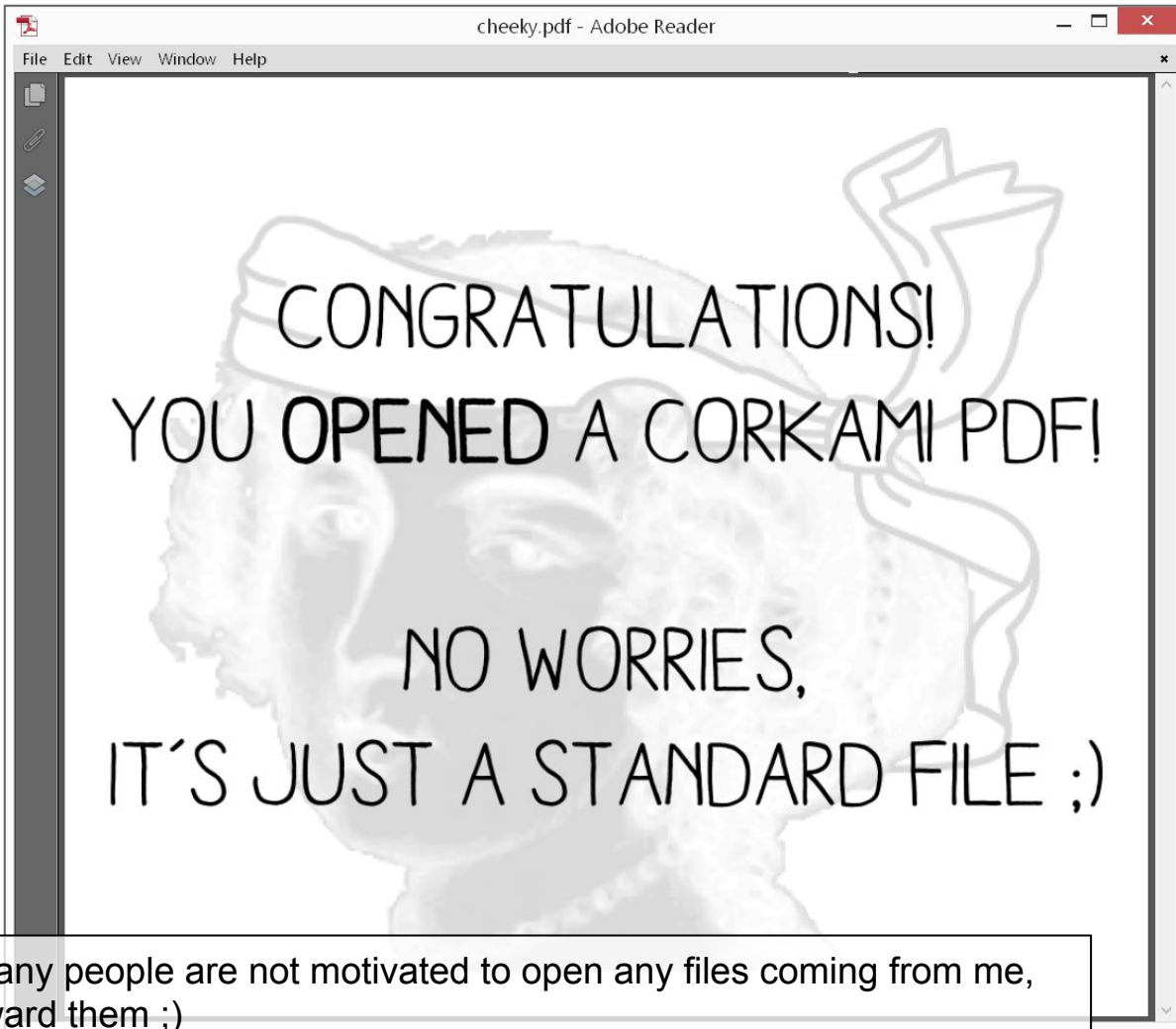


;) )

**NINTENDO**



**Sega®**



For some reason, many people are not motivated to open any files coming from me, so I made this to reward them ;)

# Print



Printer: **Microsoft XPS Document Writer** ▼

Properties

Advanced

[Help](#) ?

Copies:  ▲ ▼

Print in grayscale (black and white)

## Pages to Print

All

Current page

Pages

▶ More Options

## Page Sizing & Handling



Size



Poster



Multiple



Booklet

Fit

Actual size

Shrink oversized pages

Custom Scale:  %

Choose paper source by PDF page size

Orientation:

Auto portrait/landscape

## Comments & Forms

Document and Markups ▼

Summarize Comments

Scale: 72%

11.69 x 8.27 Inches

CONGRATULATIONS!  
YOU PRINTED A CORKAMI PDF!  
NO WORRIES,  
IT'S JUST A STANDARD FILE ;)

"standard file" ;)

**Yes, I write files 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 >>

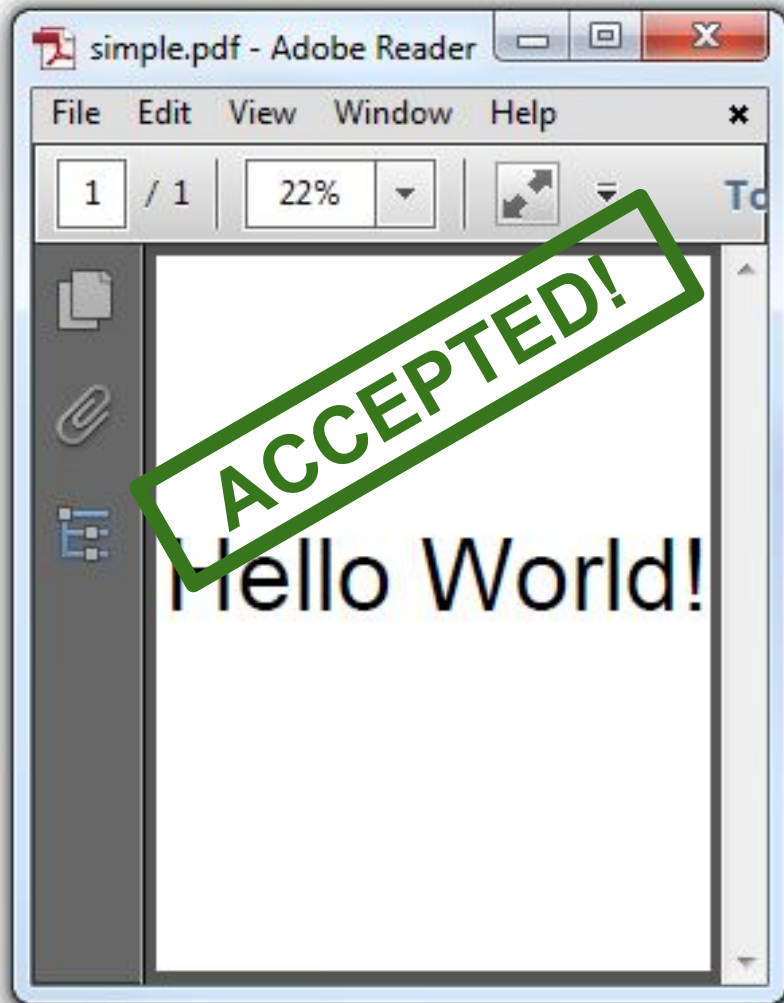
>>

/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)



**File formats are  
my playground  
(and I'm beyond recovery already)**

# **Files or file formats?**

It's not a real question.

Fichier sans format n'est que ruine de l'âme ;)



LE GOUVERNEMENT  
DU GRAND-DUCHÉ DE LUXEMBOURG  
Administration des contributions directes

modèle 100 F

[www.impotsdirects.public.lu](http://www.impotsdirects.public.lu)

Bureau d'imposition:

Réinitialiser

# Déclaration pour l'impôt sur le revenu de l'année 2014

Ce formulaire est destiné aux personnes physiques résidentes et non résidentes. La déclaration est à remettre remplie et signée pour le 31 mars 2015 au bureau d'imposition compétent sous peine d'un [supplément d'impôt pour dépôt tardif ou non-dépôt](#). Les personnes physiques qui n'ont pas leur domicile fiscal ou leur séjour habituel au Luxembourg doivent remplir la rubrique "non-résidents" à la page 3.

## signalétique

	contribuable	contribuable conjoint/partenaire
nom	<input type="text"/> 101	<input type="text"/> 102
prénom	<input type="text"/> 103	<input type="text"/> 104
date de naissance /	<input type="text"/> 105	<input type="text"/> 106

To share information, you need to use a common standard.

# Forms & file formats

moving to a different country == making a PDF/SNES polyglot

same problems: all similar, but all different

only difference: forms are rarely required to evolve

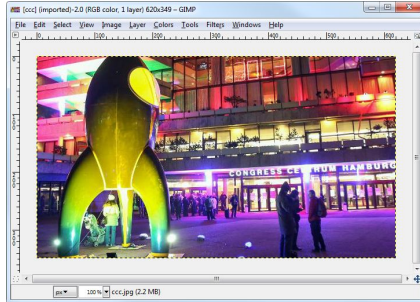
**Trusting files comes with  
trusting their format.**

**Knowing that the specs will  
be useful and reliable.**

**Retrospective**



JPG



```
>java -jar ccc.jpg
Hello world! [Java]
```

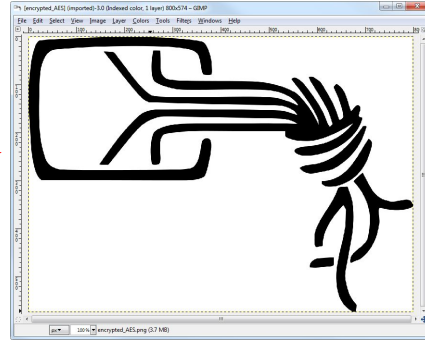
JAR  
(ZIP + CLASS)

3DES

PDF



$AES_{K_1}$



PNG



$AES_{K_2}$



FLV

I gave an entertaining presentation with many funky binary creations. Check it if you want more binary magic tricks ;)

## 6 Abusing file formats; or, Corkami, the Novella

by Ange Albertini

First, you must realize that a file has no intrinsic meaning. The meaning of a file—its type, its validity, its contents—can be different for each parser or interpreter.

Like beef cuts, which vary with the country's standards by which the animal is cut, a file is subject to interpretations of the standard. The beauty of standards is that there are so many interpretations to choose from!

Because these standards are sometimes unclear, incomplete, or difficult to understand, a variety of abuses are possible, even if the files are considered valid by individual parsers.

A *Polyglot* is a file that has different types simultaneously, which may bypass filters and avoid security counter-measures. A *Schizophrenic* file is one that is interpreted differently depending on the parser. These files may look innocent (or corrupted) to one interpreter, malicious to another. A *Chimera* is a polyglot where the same data is interpreted as different types, which is a more advanced kind of filter bypass.

This paper is a classification of various file techniques, many of which have already been mentioned in

I wrote a technical paper classifying all my file format abuses.  
So far, I was (only) playing with file formats.

view and comparison of them, not to necessarily

# Why?

Why are all these abuses possible?  
What could we (try to) do about it?



# PDF

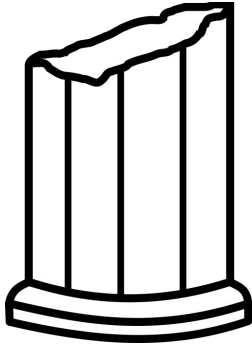
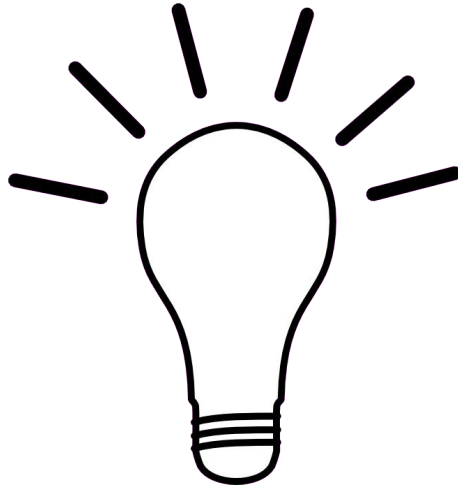
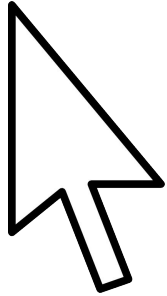


Myths

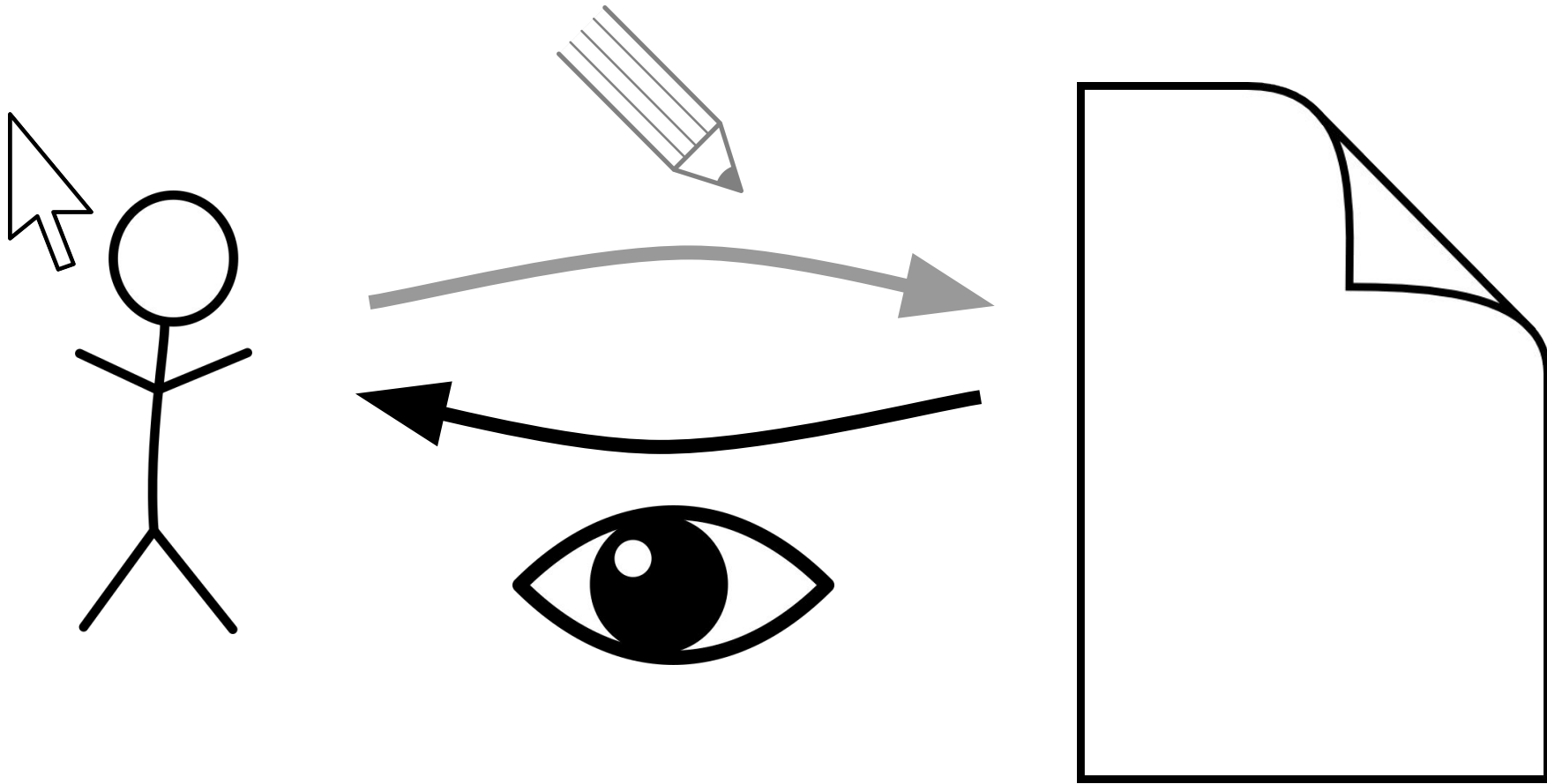
VS

FALLS

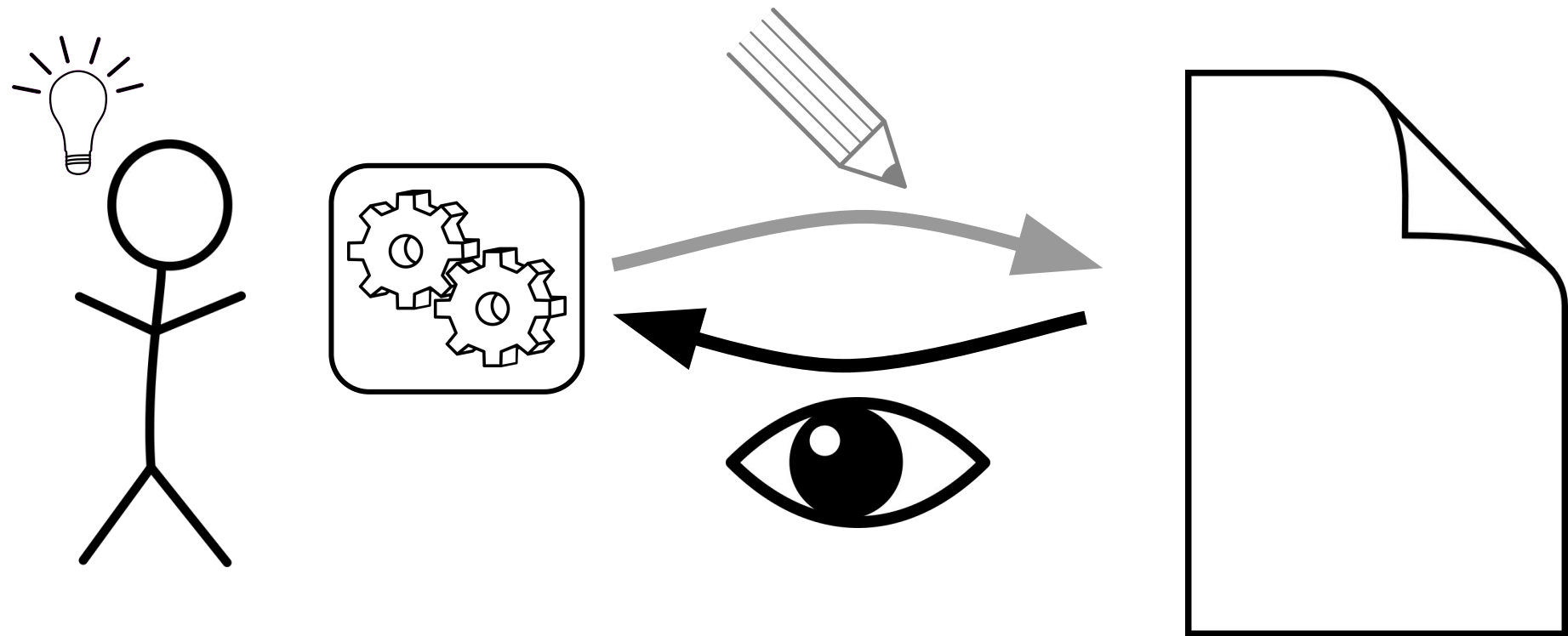
Now, I'm (also) in contact with people analyzing or designing file formats.  
I presented at a DigiPres con about Infosec: today, it's the other way around.



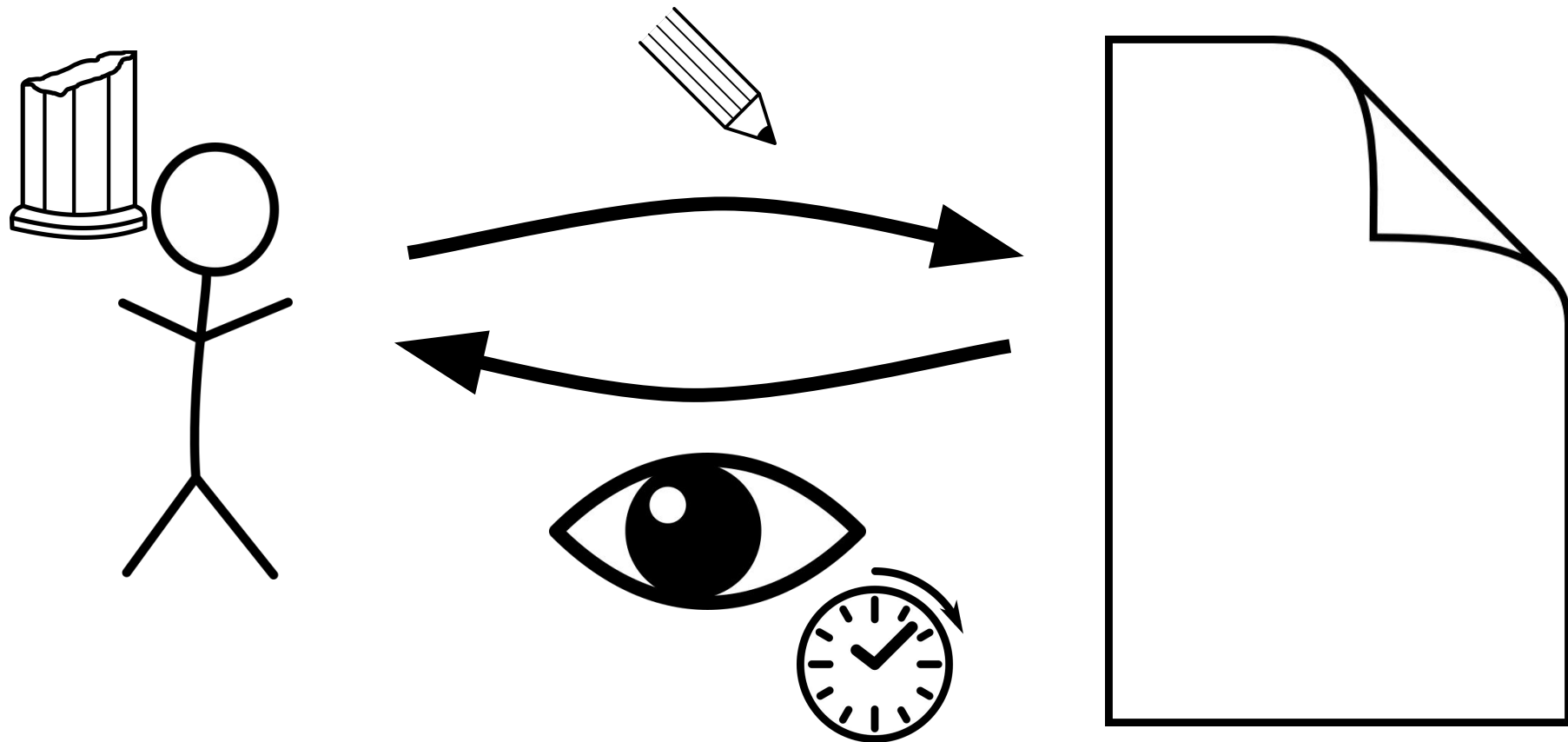
There are different kinds of expectations for files.



The end-user just wants to view external files, store his own information and re-use it.

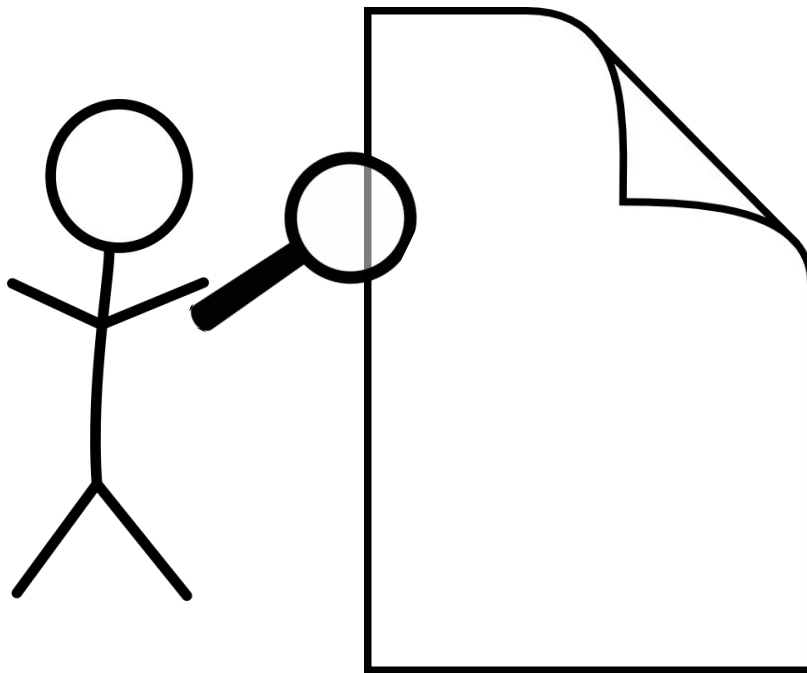


The developer relies on the specifications to add support in his library.

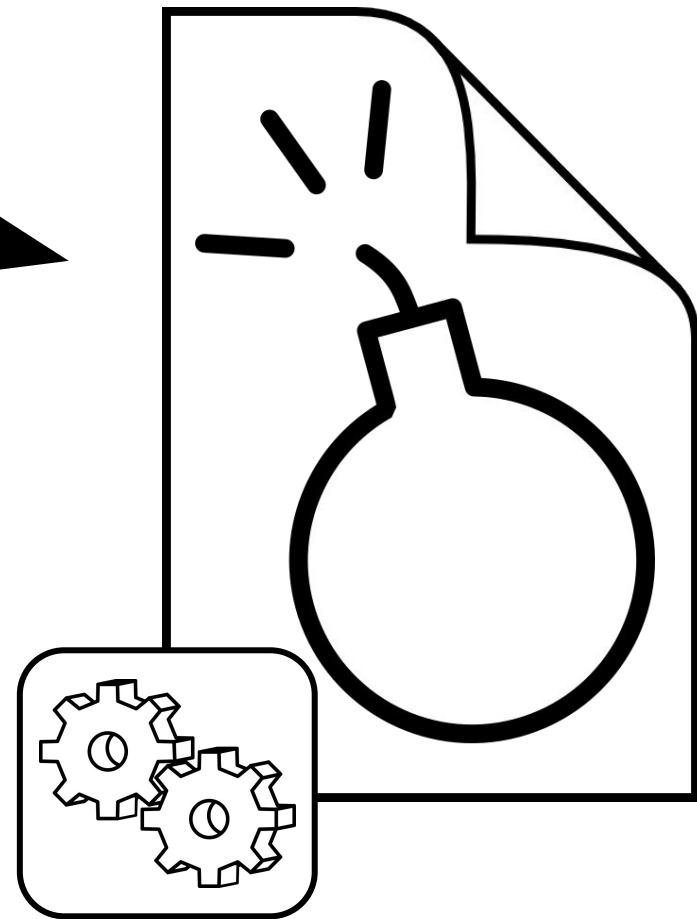
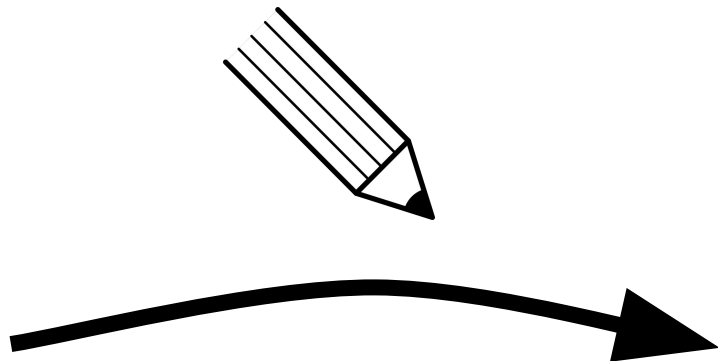
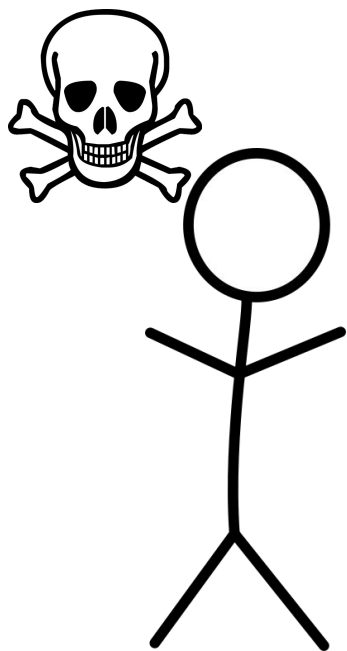


The archivist wants to make sure that his data will be re-usable *much* later.

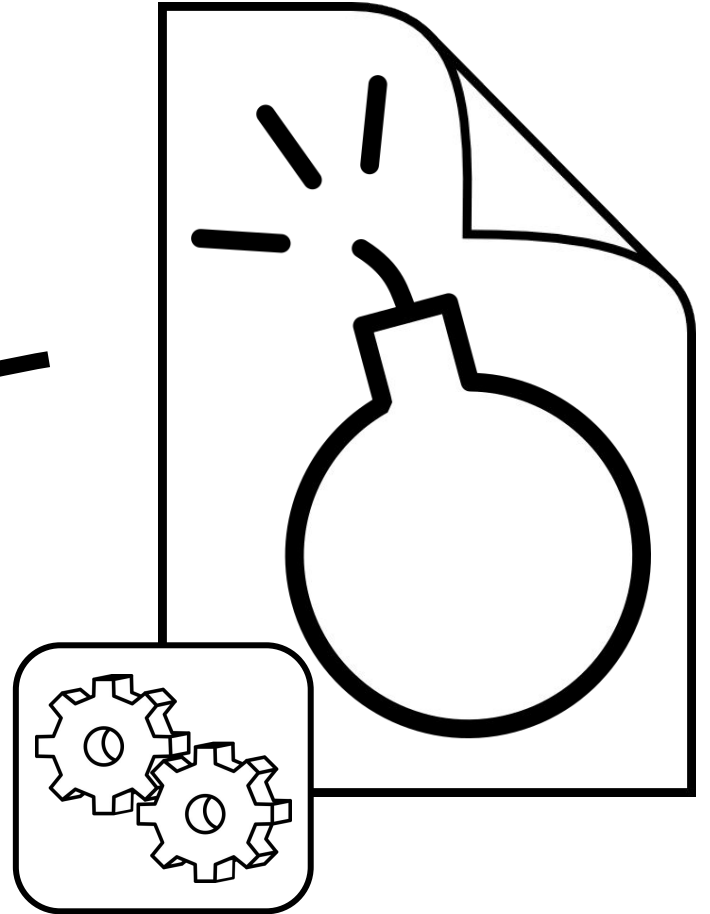
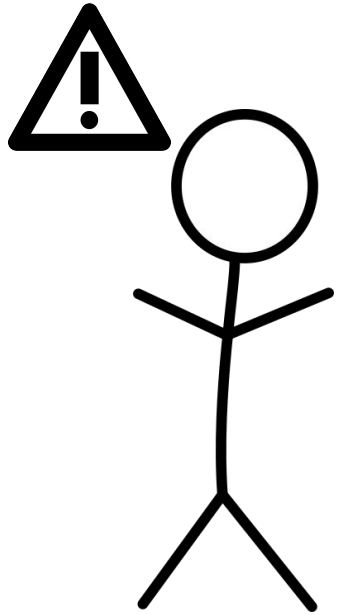




The digital investigator looks for clues in a suspect's system.



An attacker tries to craft dangerous files,



while a defender wants to prevent it from happening.

# Common points

We're blind believers:

- believing that we'll be able to reuse our information
- believing that in any case, we can just rely on the specs to help us, like a religious book.

"the cult of the (useless) specs" ;)

**It's not just an Infosec problem.**

**Bad specs make it harder for  
devs, DFIR, digipres, defenders...**

**Theory: check official specs**

**Reality:**

**check unofficial specs & blog posts**

**analyse/reverse libraries**

**gather ITW (clean & malware) samples**

Does it ring a bell ?

**Bad specs are why attackers  
and DFIR devs can make so  
much money ;)**

It's not specs reading anymore, it's reversing.

**Not all abuse of file formats  
turn into exploits.**

**But why should we only fix  
what's pwning you?**

"Short term fix" anyone?



**We just care about code,  
and "cyber attacks".**

**Files tricks go under the radar.**

Usually... a few exceptions...

The screenshot shows a Windows desktop with several icons on the left: Computer, Recycle Bin, Cygwin64 Terminal, Google Chrome, Kaspersky Anti-Virus, IDA Pro (32-bit), IDA Pro (64-bit), and WinDbg (x86). The main window is a File Explorer showing the contents of 'C:\cygwin\home\Tavis Ormandy'. The file 'exploit' is selected. A 'Properties' dialog for 'calc.exe' is open, showing details such as Version: 6.1.7601.17514, Path: C:\Windows\System32\calc.exe, and Parent: avp.exe(2532).

The screenshot shows Process Explorer displaying a list of running processes. The 'exploit' process is highlighted in red. The table below represents the data shown in the Process Explorer window:

Process	CPU	I/O Read B...	I/O Write By...	Private Bytes	Working Set	PID	Description	Company
System Idle Process	95.69			0 K	24 K	0		
System	0.03	95.6 MB	21.8 MB	204 K	1,860 K	4		
System	1.51			0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		32.5 KB	20 B	468 K	1,148 K	364	Windows Session Manager	Microsoft
csrss.exe		224.7 KB		2,184 K	4,684 K	460	Client Server Runtime Process	Microsoft
wininit.exe		7.0 KB		1,504 K	4,536 K	512	Windows Start-Up Application	Microsoft
services.exe		475.0 KB	3.2 MB	5,144 K	9,208 K	608	Services and Controller app	Microsoft
svchost.exe	< 0.01			4,128 K	9,444 K	740	Host Process for Windows S...	Microsoft
vmacthlp.exe		233 B		1,460 K	4,176 K	800	VMware Activation Helper	VMware,
svchost.exe	0.02			4,112 K	8,348 K	844	Host Process for Windows S...	Microsoft
svchost.exe	0.03	3.7 MB	757.8 KB	16,704 K	17,984 K	932	Host Process for Windows S...	Microsoft
audiodg.exe				15,660 K	15,660 K	1672	Windows Audio Device Grap...	Microsoft
svchost.exe		1.3 KB	388 B	5,656 K	13,236 K	984	Host Process for Windows S...	Microsoft
dwm.exe				1,880 K	5,480 K	1580	Desktop Window Manager	Microsoft
svchost.exe	< 0.01	1.3 KB		7,032 K	12,640 K	108	Host Process for Windows S...	Microsoft
svchost.exe	< 0.01	22.7 MB	2.4 MB	17,132 K	30,164 K	372	Host Process for Windows S...	Microsoft
svchost.exe		116 B	160 B	2,060 K	5,196 K	816	Host Process for Windows S...	Microsoft
svchost.exe	< 0.01	1.6 MB	43.0 KB	9,688 K	12,276 K	1116	Host Process for Windows S...	Microsoft
spoolsv.exe		1.2 KB	320 B	7,324 K	13,616 K	1244	Spooler SubSystem App	Microsoft
svchost.exe		48.2 MB	450.6 KB	11,576 K	14,132 K	1284	Host Process for Windows S...	Microsoft
taskhost.exe	< 0.01	1.9 MB	678.0 KB	11,608 K	10,884 K	1384	Host Process for Windows T...	Microsoft
svchost.exe	< 0.01	54.3 KB		4,612 K	9,492 K	1556	Host Process for Windows S...	Microsoft
lpdverUsbSvc.exe		53.8 KB		8,248 K	11,036 K	1640		Microsoft
sqlwriter.exe		116 B	160 B	1,928 K	6,180 K	1860	SQL Server VSS Writer - 64 Bit	Microsoft
SyslogAgent.exe	0.03			5,024 K	8,200 K	1916	SyslogAgent	Dalagran
vmtoolsd.exe	0.04	125.4 KB	13.6 KB	6,972 K	15,532 K	1964	VMware Tools Core Service	VMware,
dihost.exe	< 0.01	2.0 MB	3.5 KB	4,260 K	11,448 K	2380	COM Surrogate	Microsoft
msdtc.exe				3,548 K	8,098 K	2468	Microsoft Distributed Transa...	Microsoft
SearchIndexer.exe		6.8 MB	1.3 MB	17,712 K	13,824 K	2744	Microsoft Windows Search L...	Microsoft
svchost.exe		12.9 MB	1.9 KB	51,476 K	22,988 K	2032	Host Process for Windows S...	Microsoft
lsass.exe		93.8 KB	138.1 KB	3,804 K	10,168 K	636	Local Security Authority Proc...	Microsoft
lsm.exe				2,544 K	4,232 K	644	Local Session Manager Serv...	Microsoft
csrss.exe	0.18	1.6 MB		13,648 K	11,344 K	532	Client Server Runtime Process	Microsoft
winlogon.exe		14.1 KB		2,768 K	7,528 K	584	Windows Logon Application	Microsoft
explorer.exe	0.11	4.6 MB	80.9 KB	42,608 K	61,488 K	1632	Windows Explorer	Microsoft
vmtoolsd.exe	0.09	11.5 KB	2.0 MB	11,840 K	19,312 K	1500	VMware Tools Core Service	VMware,
procep.exe		118.7 KB	1.3 MB	2,236 K	7,188 K	2968	Sysinternals Process Explor...	Sysintern
PROCEXP64.exe	1.09	250.8 KB	33.7 KB	14,364 K	24,096 K	2988	Sysinternals Process Explorer	Sysintern
avpui.exe	1.16	1.3 MB	6.2 KB	67,816 K	2,988 K	392	Kaspersky Anti-Virus	Kaspersk
calc.exe				4,772 K	9,080 K	1592	Windows Calculator	Microsoft

Tavis Ormandy's ZIP/DLL polyglot exploit for Kaspersky

Browser window showing a "This webpage is not available" error message. The address bar contains `https://192.168.237.1/whatever`. The error message includes "ERR\_CONNECTION\_RESET" and a "Reload" button.

Process Explorer window showing system processes. The table below lists the processes and their resource usage:

Process	CPU	I/O Read B...	I/O Write By...	Private Bytes	Working Set	PID	Description
svchost.exe	< 0.01	25.4 MB	1.9 KB	44,900 K	41,224 K	3032	Host Process for Windo
WmiApSvc.exe	0.01			1,816 K	6,048 K	2832	WMI Performance Rieve
lsass.exe		50.6 KB	107.0 KB	4,144 K	11,016 K	508	Local Security Authority
lsmd.exe				2,436 K	4,196 K	516	Local Session Manager
csrss.exe	0.07	779.5 KB		17,168 K	13,400 K	408	Client Server Runtime P
winlogon.exe	< 0.01	15.7 KB	160 B	2,800 K	7,420 K	460	Windows Logon Applic
explorer.exe	0.04	3.6 MB	480 B	23,320 K	36,660 K	3016	Windows Explorer
vmtoolsd.exe	0.07	10.0 KB	4.1 KB	7,736 K	14,016 K	2920	VMware Tools Core Ser
chrome.exe	0.58	104.5 MB	19.1 MB	57,052 K	87,516 K	3654	Google Chrome
chrome.exe		39.4 MB	52.9 KB	37,496 K	65,128 K	3756	Google Chrome
chrome.exe		39.9 MB	240.4 KB	31,456 K	57,736 K	3768	Google Chrome
chrome.exe		43.0 MB	196.4 KB	47,248 K	73,796 K	3944	Google Chrome
chrome.exe	0.08	39.7 MB	63.5 KB	34,208 K	53,636 K	3668	Google Chrome
chrome.exe		45.3 MB	90.0 KB				
chrome.exe	0.08	39.7 MB	11.1 KB				
chrome.exe		40.7 MB	948.3 KB				
nacl64.exe		1.1 KB	100 B				
nacl64.exe		3.3 MB	32.5 KB				
AvastUI.exe	0.04	8.9 MB	911.1 KB				
calc.exe		60 B					

Avast Free Antivirus notification: "Avast Web Shield has blocked access to this page because of SRS..."

Calculator window showing a numeric keypad and basic arithmetic functions.

Avast Free Antivirus interface showing system status and performance alerts.

**YOU ARE** Not registered (29 days remaining) [Register](#)

Self defense is turned off [Turn on](#)

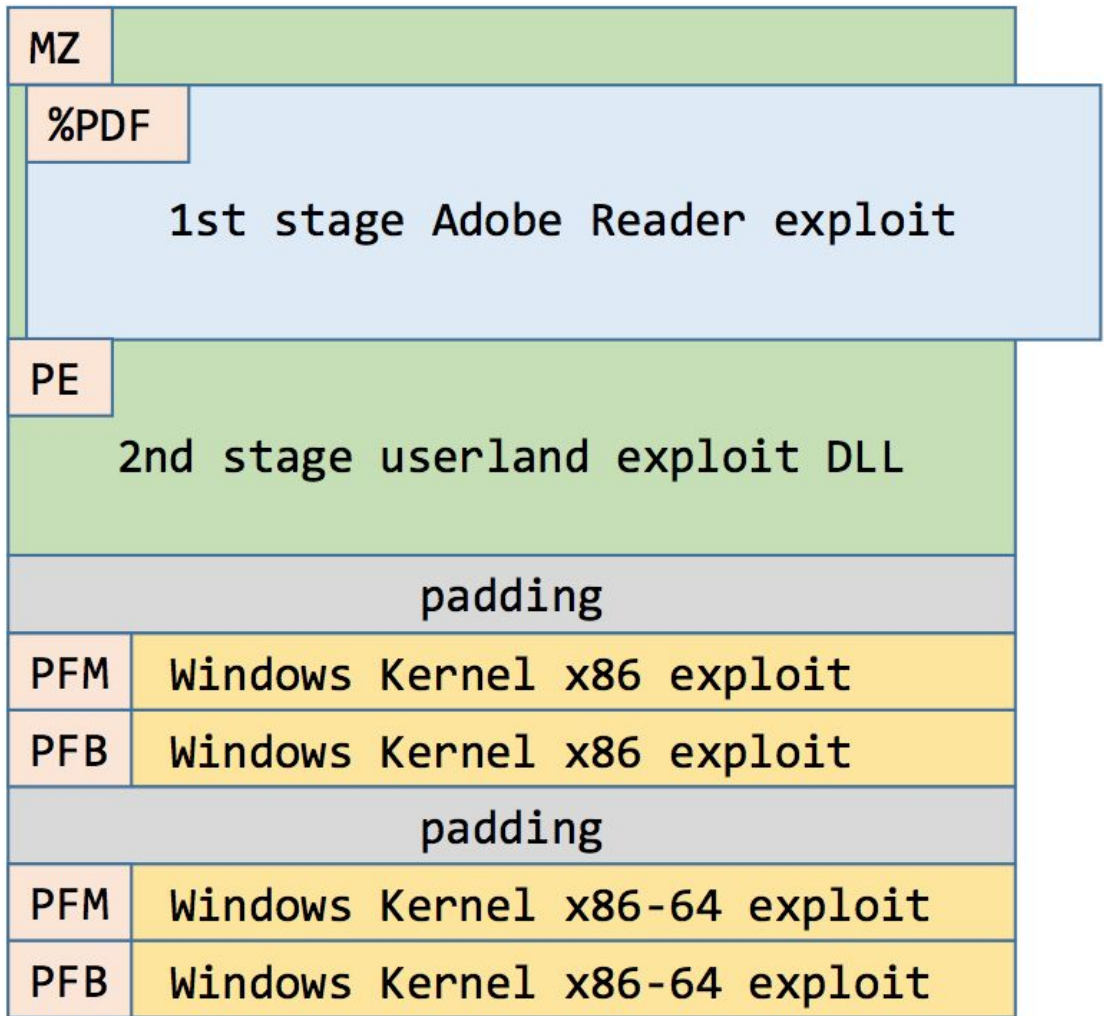
All shields active

**Attention:** Your computer might be sluggish and running slowly!

29 useless programs can be deactivated.  
3 system settings issues can be tweaked.  
300 MB can be freed up on your hard drive.

[Improve performance](#)

# Tavis Ormandy's "HTML in certificate" exploit for Avast



**If we don't understand  
how it really works,  
we can't: parse it, preserve it,  
tell if corrupted or malicious.**

# **Crafting a file format**

# **File format is not just "data structure"**

Protobuf / XML doesn't solve everything.  
They're just the high-level layer.

Data structure need to be logical and make sense from a dev perspective.

So at least, use a magic number/signature, and enforce version numbers, sizes... ;)

# Failure is still possible

Office file format is a ... filesystem!

You can defragment it!

And it has different kinds of FAT ;)



## We're a bunch of developers from IBM, ask us anything!

**TECHNOLOGY** submitted 1 month ago \* by CrazyAboutCode

Hey Reddit! We're a bunch of developers who like to talk to people. So stereotypes be damned. We work at IBM and like to talk about app infrastructure, app delivery and app tool projects (some of our favorite projects:



Top 200 Comments [show 500](#)

sorted by: [top](#) ▼



[\[-\]](#) **Acredit** 2084 points 1 month ago 🌟

What is the worst piece of software you've worked on and why is it Lotus Notes?

[permalink](#)

# **A file format is not just an "algorithm"**

Your algorithm is great, but the file format will be the interface between your algorithm and all its users and other applications.

finish your specs! double-check them!  
provide test cases!

# A file format is a map

Every street should follow the same rules,  
Otherwise you must expect many violations.  
Wherever there is a 'surprise', bad things  
happen.

Consistency ^ (Compatibility || Schizophrenia)

# "PSD makes inconsistency an art form"

```
XeePhotoshopLoad x
C https://code.google.com/p/xee/source/browse/XeePhotoshopLoader.m?r=f16763d221dfca6253983824b470adf553a19e06#108
107 off_t nextchunk=[lh.offsetInFile]+((chunklen+3)&~3);
108 // At this point, I'd like to take a moment to speak to you about the Adobe PSD format.
109 // PSD is not a good format. PSD is not even a bad format. Calling it such would be an
110 // insult to other bad formats, such as PCX or JPEG. No, PSD is an abysmal format. Having
111 // worked on this code for several weeks now, my hate for PSD has grown to a raging fire
112 // that burns with the fierce passion of a million suns.
113 // If there are two different ways of doing something, PSD will do both, in different
114 // places. It will then make up three more ways no sane human would think of, and do those
115 // too. PSD makes inconsistency an art form. Why, for instance, did it suddenly decide
116 // that *these* particular chunks should be aligned to four bytes, and that this alignment
117 // should *not* be included in the size? Other chunks in other places are either unaligned,
118 // or aligned with the alignment included in the size. Here, though, it is not included.
119 // Either one of these three behaviours would be fine. A sane format would pick one. PSD,
120 // of course, uses all three, and more.
121 // Trying to get data out of a PSD file is like trying to find something in the attic of
122 // your eccentric old uncle who died in a freak freshwater shark attack on his 58th
123 // birthday. That last detail may not be important for the purposes of the simile, but
124 // at this point I am spending a lot of time imagining amusing fates for the people
125 // responsible for this Rube Goldberg of a file format.
126 // Earlier, I tried to get a hold of the latest specs for the PSD file format. To do this,
127 // I had to apply to them for permission to apply to them to have them consider sending
128 // me this sacred tome. This would have involved faxing them a copy of some document or
129 // other, probably signed in blood. I can only imagine that they make this process so
130 // difficult because they are intensely ashamed of having created this abomination. I
131 // was naturally not gullible enough to go through with this procedure, but if I had done
132 // so, I would have printed out every single page of the spec, and set them all on fire.
133 // were it within my power, I would gather every single copy of those specs, and launch
134 // them on a spaceship directly into the sun.
135 //
136 // PSD is not my favourite file format.
137
138 if(sign!='PTM') break; // sanity check
```

# Not just specs

A default open implementation?

with test cases for the code, and free-licenced examples cases provided.

Too many 'features from the specs' are never seen in the wild.

# Life of a file format

1. define a format (if possible)
2. implement it in your software
3. end :(

if you're lucky:

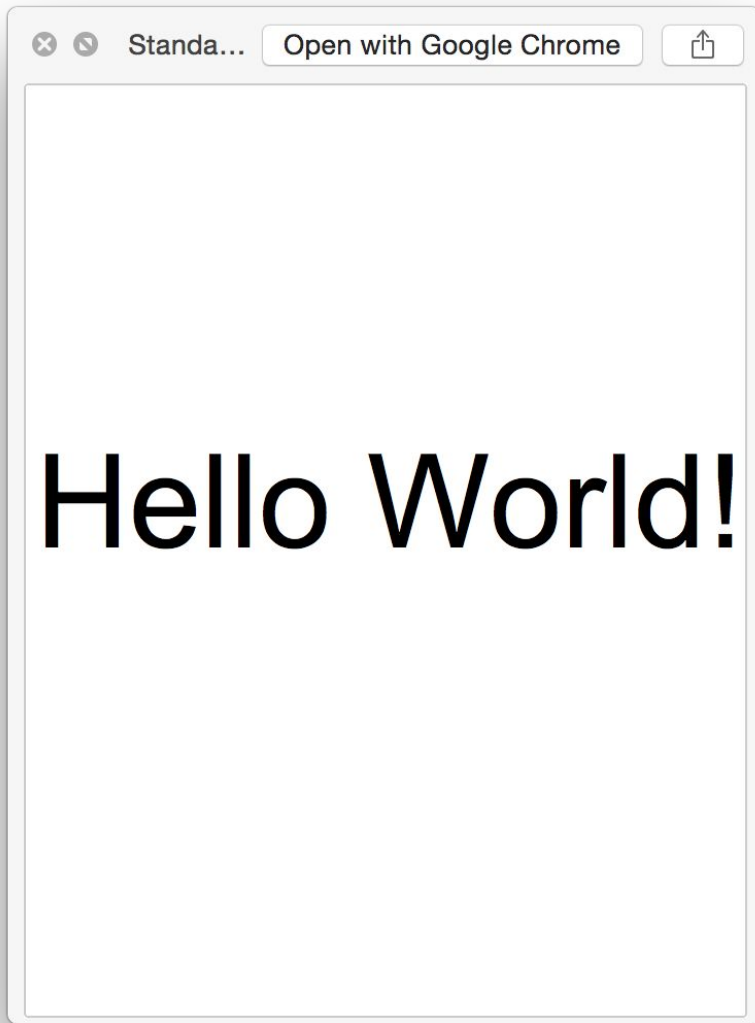
your software becomes standard  
along with its file format. That's all.

**Becoming a de-facto standard  
doesn't require anything:  
it's your niche market.  
No official requirements.  
Just business directions.  
no "long term plan"**

**You end up with a standard  
that was never properly  
designed or documented in  
the first place.**

Have fun preserving it or making it secure!





I wrote a simple "Hello World" PDF,  
that works on every reader.  
Yet, it's not 100% standard (only 99%)  
That's a bad start :(

# Thinking about bundling?

Hint: don't.

```
int bundle(trust){return trust--;} 
```



Adobe Acrobat Reader DC

Version 2015.008.20082  
[System requirements](#)

Your system:

**Optional offer:**

Yes, install the free **McAfee Security Scan Plus** utility to check the status of my PC security. It will not modify existing antivirus program or PC settings.

 **McAfee** Security Scan Plus

[Learn more](#)



**change.org** [Start a petition](#) [Browse](#) [Search](#) [Log in](#)

Petitioning [CEO of Oracle Corporation Larry Ellison](#) and 2 others

## Oracle Corporation: Stop bundling Ask Toolbar with the Java installer

 **Saeid Nourian** Lowell, MA

# **Evolution of a format**

(divergence)

# Evolution

1. Tool X creates bogus file
2. StandardTool adapts silently to support them
3. Now StandardTool goes beyond the specs

Specs are now even more useless.

Ex: ColorTrac scanners, PDF readers

**Implementations slowly  
diverge from the specs**

**⇒ the specs become  
theoretical and  
useless in the wild.**

Yet nothing exists to replace them.

**Once it's a standard,  
it's too late to fix it.**

**Before it's a standard,  
no one really cares.**

And too few people care anyway ;)

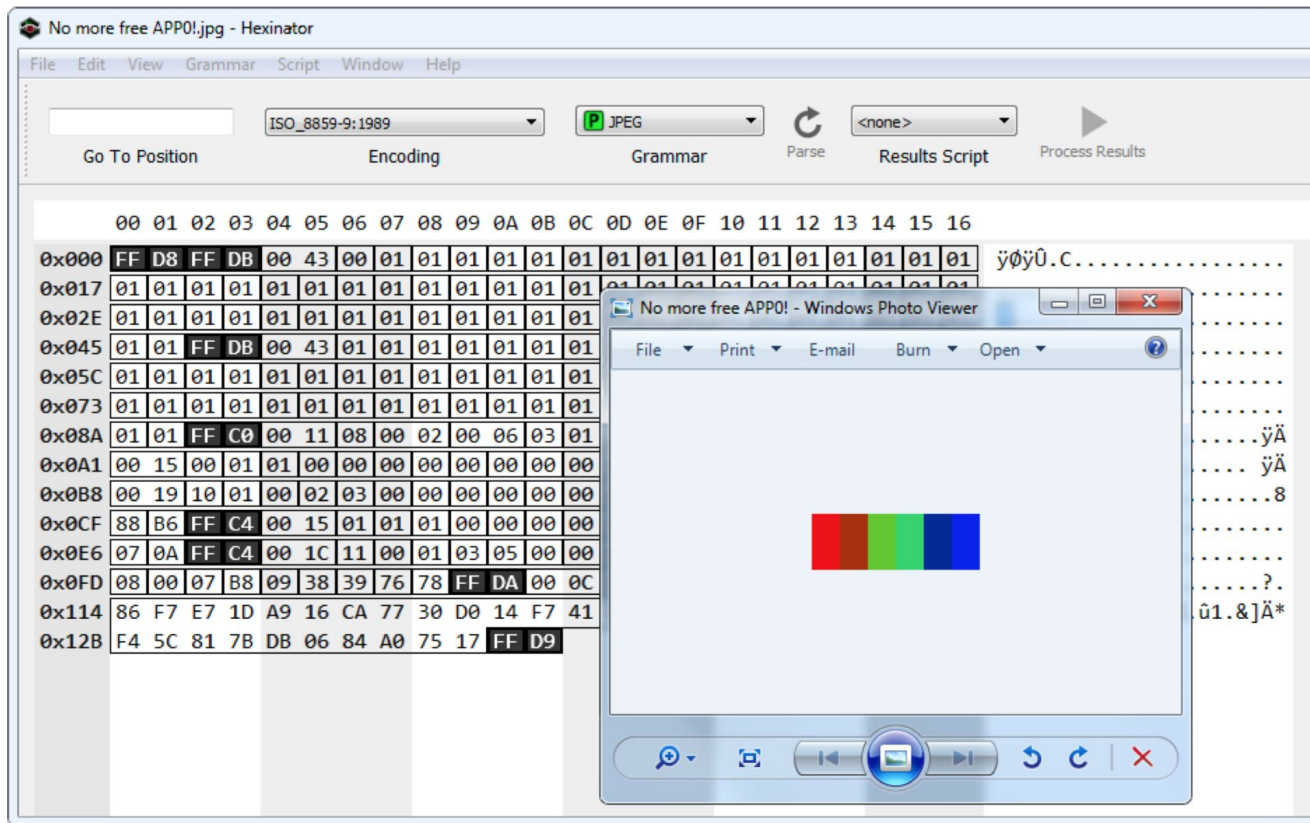
# JPEG 1/2

JPEG (1992) is not a file format!

Open source library: LibJPEG → that's great!

LibJPEG goes beyond the specs:

- recovers standard types of App0 chunks
  - including the one specific to Adobe
- unnecessary functions (headless JPEG (!))
  - "let's add this in case" ↔ design by committee ?



A JPG without a 'required' APPx segment

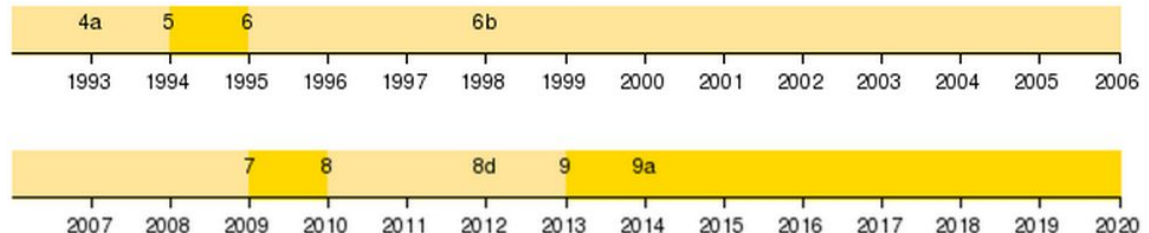


# JPEG 2/2

JPEG is 'de facto' libJPEG-turbo v6b.

Explore corner-cases, and then you fail Adobe or Safari:

⇒ their test cases are not big enough



# Major problems (so many!)

specs really come last: absent, or TBD

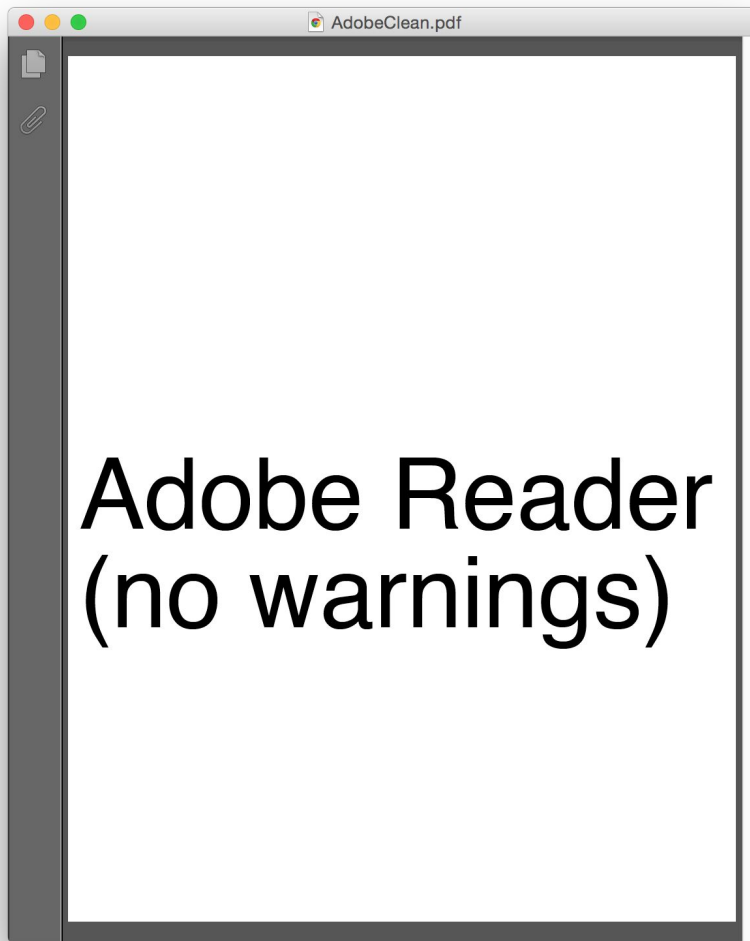
incomplete specs: BPG, ZIP, PDF

incoherent specs: PDF

non-free specs.

# Recovering broken files

AKA "hidden mode"



Take a fully working PDF.

Adobe Reader



**There was an error opening this document.  
There was a problem reading this  
document (109).**

OK

Change one byte at the wrong place (in the XREF) ⇒ OMG it's corrupted!



**Do you want to save the changes you made in the document "AdobeCleanBroken.pdf"?**

Your changes will be lost if you don't save them.

Don't Save

Cancel

Save

But if you remove its XREF entirely, it now miraculously works, with just a (misleading) dialog on closing, that actually means: "we found some bugs, do you want to save as a valid but bloated file?"

**Standard programs  
typically embed a  
(silent) recovery mode.**

**Nightmare for devs/defenders**

**These modes try their best  
to recover "broken" files.  
Far beyond the specs.**



**To improve security  
and format reliability:  
turn auto-recovery into  
dialog box warnings?  
or reject these files and log the error?  
That would make vendors act.  
"This file is not correct,  
please contact your vendor"...**

**"helping" the end-user  
by triggering no warning?  
(even temporarily) OK**

**What about identifying bad  
practices to make them stop  
eventually?**

# Forcibly deprecate?

Like crypto? Sounds good, but...

Not going to happen:

Broken crypto leads to fast and mass pwnage.

Broken file formats mostly just lead to headache - no incentive to avoid that.

Not enough "Android master key" bugs yet.

**"one" standard ?**



Adobe Reader

Poppler

F

Firefox

(warnings)

Chrome/Foxit

MuPDF

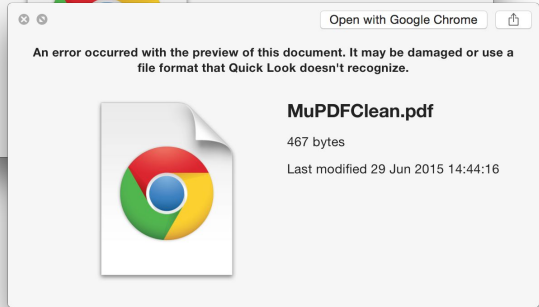
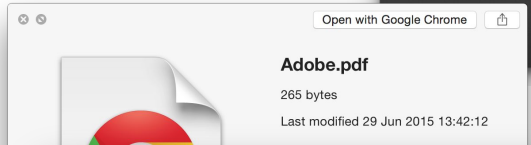
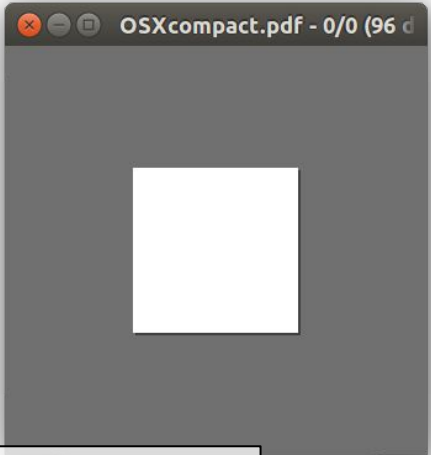
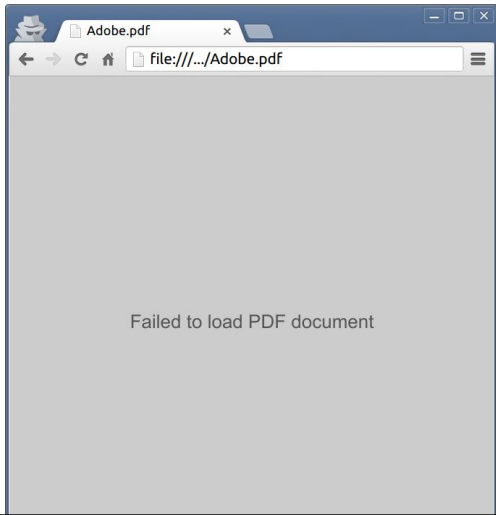
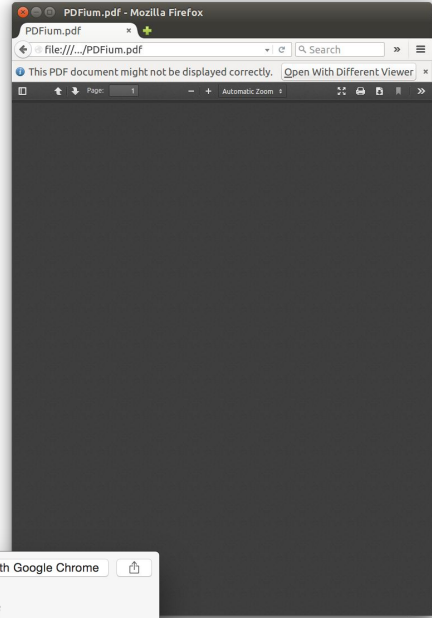
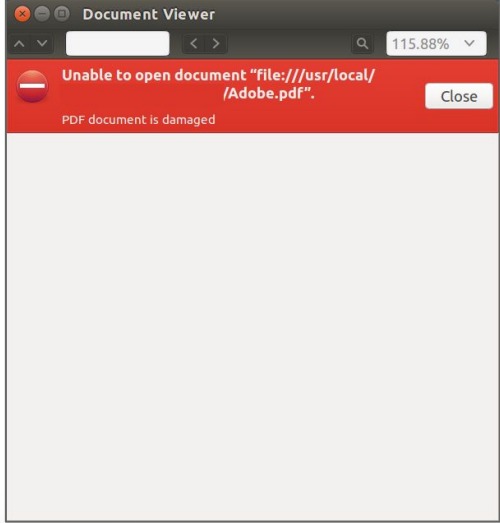
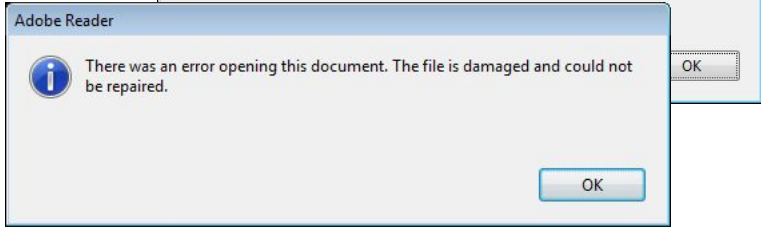
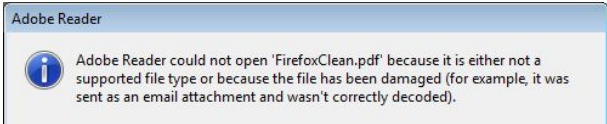
OS  
pre

OS X  
preview

OS X  
preview  
Safari

OS X  
preview  
Safari

I made extremely custom PDFs for each reader.



These "extreme" PDFs fail on any other reader.

# Consequence

We have 6 PDF reader 'standards' in practice:  
these may be extreme examples, but  
OTOH "Hello World" is not so complex

"Nothing to fix"

"Specs are subject to interpretation"

# PDF Schizophrenia?

- Sumatra / Chrome-1 / Others
- Chrome-2 / Others
- Safari / Others
- Poppler / Others

It's not even funny anymore...

⇒ any unclear area may lead to schizophrenia



# PDF = portable?

Most readers are okay to read 'standard' docs.  
any advanced functions? Adobe Reader  
(printing, forms, JavaScript, 3D).  
Also, no more Linux version.

# PDF, a clean standard?

Non-free specs.

Only the "standard" 1.7 doc is free.

No free examples.

Incomplete + missing specs  
no shareable samples

**Non-free specs?**

**No free sample-set?**

**And you wish to stay  
a "standard" in 2015?**

# PDF for archiving?

PDF/A already has 8 sub-standards

Adobe Preflight is not very updated

⇒ Preservation is **not** a business model,  
nor a legal requirement of any kind.

How long before "support is discontinued"?

# PDF 2.0

No new security stuff, specs are now 170 CHF.  
New printing features, new insecure features:  
embedding files anyone?

# What's unique about PDF?

And why PDF will live forever

PDF Association | October 6, 2015

<http://www.pdfa.org/2015/10/whats-unique-about-pdf/>

The screenshot shows the PDF Association website. At the top, there's a navigation bar with links for 'Members' News', 'Products', 'Downloads', 'Videos', 'Discussion Forums', and 'Login'. Below that is a search bar and a language selector set to 'English'. The main content area features the article title 'What's unique about PDF?' and sub-header 'And why PDF will live forever', both highlighted by a red callout box. The article text describes the PDF format's attributes and its role as an 'electronic document'. A superhero illustration with 'PDF' on its chest is also present. On the right, there's a sidebar with 'Tags' including '3rd International PDF/A Conference 4th PDF/A Conference 2012 PDF Technical Conference accessibility accessible PDF A11M archive Archiving assistive technology barrierefrei compression conversion convert digital signature DMS Expo document conversion ECM encryption ISO Standard JAVA long-term archiving metadata OCR'.

## What's unique about PDF?

And why PDF will live forever

PDF Association | October 6, 2015

The Portable Document Format possesses a variety of attributes that are, themselves, more subtle than what one normally thinks of as "features". Taken together, however, they describe a format of such flexibility and power that it will define the essential "electronic document" concept forever.

### It's a document

Wikipedia, among others, defines "document" in terms of content (text and graphics, together in a layout) as it exists at a given moment in time. The need for a sharable electronic document drove the fundamental design of PDF. The format allows pages – that is, a fixed layout of text and graphics – to be shared with total fidelity to the author's intent. However they were made, PDF documents look the same way to everyone. This feature was critical to establishing PDF as a candidate for the standard electronic document format, but it was not enough.

### One format can hold it all

Closely related to PDF's ability to reliably share fixed-layout content is the fact that a PDF document may include pages from many (any) different source. Users can (and often do) mix PDF pages produced from MS Word with PDF pages from scanned documents, screen-captures, CAD images and more. No electronic document format could replace PDF unless it was also able to allow users to mix pages together to form the documents (see above) they need.



2015 October 19-20  
San Jose, California

#### Tags

3rd International PDF/A Conference 4th PDF/A Conference 2012 PDF Technical Conference accessibility accessible PDF A11M archive Archiving assistive technology barrierefrei compression conversion convert digital signature DMS Expo document conversion ECM encryption ISO Standard JAVA long-term archiving metadata OCR

I'm not so sure about it - after all, we're killing Flash for security reasons.

# A (tiny) ray of hope

VeraPDF.org:

open source PDF/A validator.

## Definitive PDF/A Validation



veraPDF is a purpose-built, open source, file-format validator covering all PDF/A parts and conformance levels.

veraPDF is designed to meet the needs of digital preservationists and is supported by the PDF software developer community.

# Preservation

portable compiler + toolchain

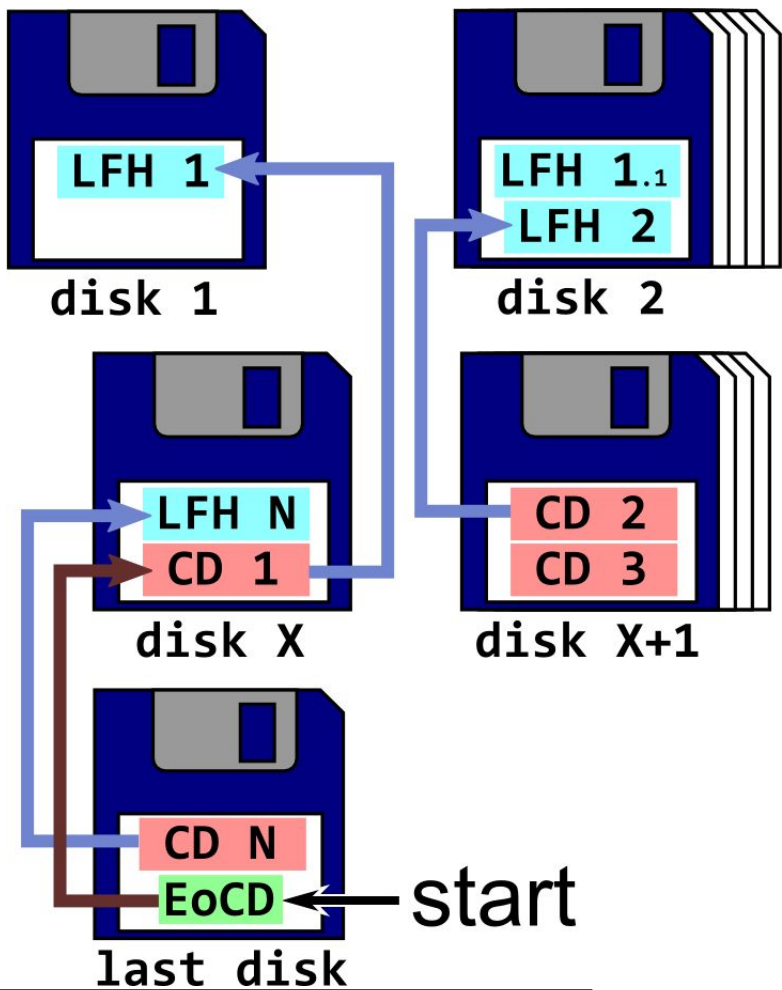
portable source

no OS dependency at all ?

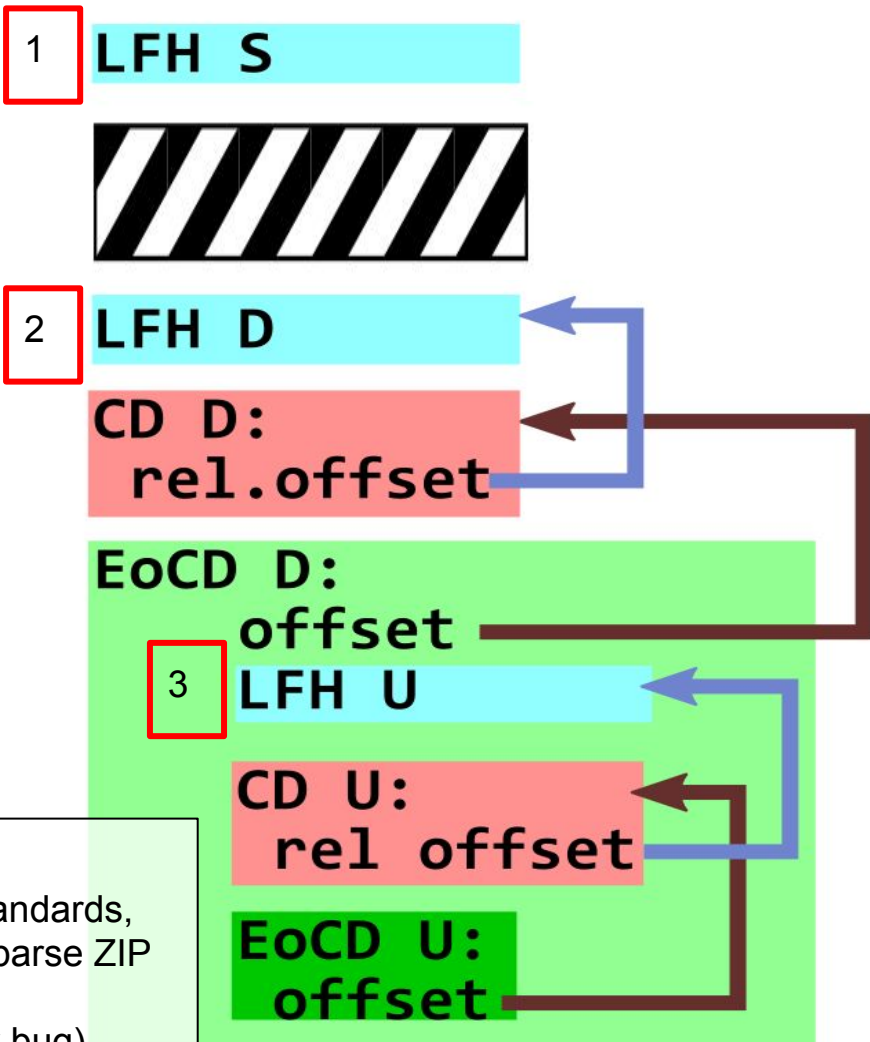
preservation via closed-source software?

⇒ "emulation as a service" has a great future :(





ZIP archives already made for multiple floppy support.



Because it's awkward and suboptimal for modern standards, there are now 3 ways ITW to parse ZIP (can be abused like in the Android Master Key bug)

**SCHIZO PHRENIC FILES**

**GUYVAEL COLDWIND**  
SECURITY RESEARCHER, GOOGLE  
DRAGON SECTOR CAPTAIN  
LIKES HAMBURGERS  
<http://guyvael.coldwind.pl/>

**ANGE ALBERTINI**  
reverse engineering & VISUAL DOCUMENTATIONS  
corkami.com

# Do we still need floppy support?

ZIP (1989) is still updated.

ZIP added AES, LZMA, 64 bits, Unicode.

But still this awkward obsolete structure?

Why not just reorder structures, enforcing values, and slowly preventing abuses ?

Not re-inventing the format, just forking it.

# Seriously

Do we still really need Tape Archives?

Floppy-oriented, backward-parsed ZIP?

Any generated PDF that doesn't have its magic at offset 0?

FTR:

OpenSSL still supports WinCE, BeOS....

Windows bitmap fonts are stored as 16 bits NE executables (copyright 1989).

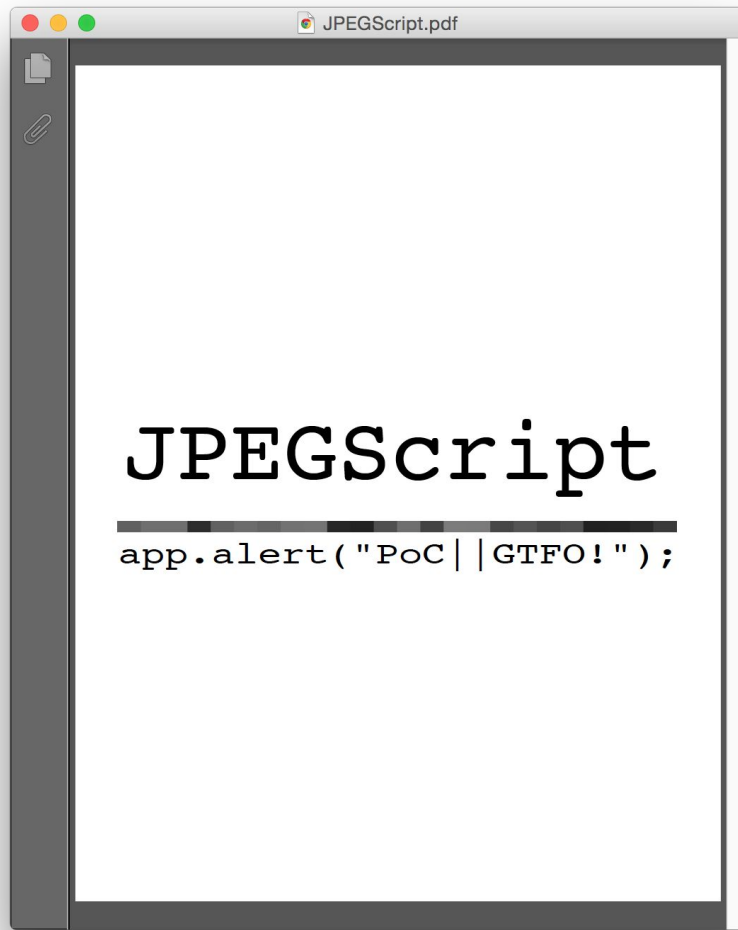
# Pure digital preservation

New documents are born digital:

the problem is shifted:

the 'master' copy already depends on:

source+compiler+toolchain+(OS+CPU).



A PDF with a JPEG-compressed script

# JPEG, but not an image?

It's not against the specs,  
but it was removed without any warning nor  
tracking.

⇒ breaks backward compatibility

If your document was using it, now it's broken.

If this document is born digital, you lost your  
source document.

# Backward compatibility

Everywhere.

In case, you never know.

The customer is always right.

Perhaps except for security things ;)

Our kids will probably ask us one day  
why we kept all these things for so long...



# Windows compatibility

Windows is becoming progressively (but silently) more strict for the PE format, slowly killing several packers.

Have you heard anyone complaining?

(the official PE doc still totally sucks though)

# breaking backward compatibility

It's ok if it's for valid reasons,  
but keep track of changes, enforce version  
numbers, and update the specs accordingly at  
the same time!

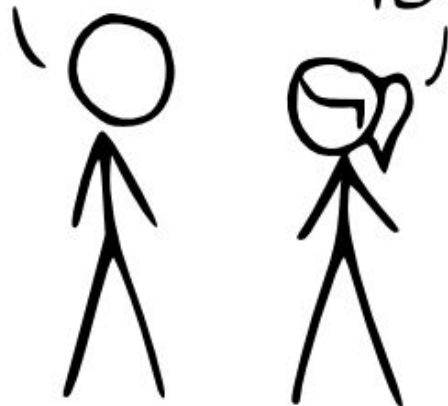
Nowadays,  
a file format is an evolving entity for security reasons,  
not something sacred written in stone

# HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGES, CHARACTER ENCODING, INSTANT MESSAGING, ETC)

SITUATION:  
THERE ARE  
14 COMPETING  
STANDARDS

14?! RIDICULOUS!  
WE NEED TO DEVELOP  
ONE UNIVERSAL STANDARD  
THAT COVERS EVERYONE'S  
USE CASES.



SOON:

SITUATION:  
THERE ARE  
15 COMPETING  
STANDARDS

**Multiple formats is not the problem:  
we have different needs.**

**But documentation never reflect  
reality in any case.**

**There are many benefits to know definitively what a valid file is or isn't.**

# Cleaning up

Terse Executable is a cleaned-up version of the Portable Executable (but for UEFI, not to replace it).

Only example of forking that makes sense?  
We just stack features...

# **There's no standard for file format specifications**

different style of writing, may be incomplete  
unclear, non free...

**Something I tried**





```

BFF9AF2770      mov     edi,07027AFF9 ;'p'>
B068           3mov     al,068 ;'h'
AA             stosb
B800102900      mov     eax,000291000 --f4
AB             stosd
66B8C300       mov     ax,000C3 ;'t'
AA             stosb
89D8           mov     eax,ebx
0000           add     [eax],al
0000           add     [eax],al

```

Number	Name	UirtSize	RVA	PhysSize	Offset	Flag
65524		00007000	70226000	00000000	00280200	E00000C0
65525		00007000	7022D000	00000000	00280200	E00000C0
65526		00007000	70234000	00000000	00280200	E00000C0
65527		00007000	7023B000	00000000	00280200	E00000C0
65528		00007000	70242000	00000000	00280200	E00000C0
65529		00007000	70249000	00000000	00280200	E00000C0
65530		00007000	70250000	00000000	00280200	E00000C0
65531		00007000	70257000	00000000	00280200	E00000C0
65532		00007000	7025E000	00000000	00280200	E00000C0
65533		00007000	70265000	00000000	00280200	E00000C0
65534		00007000	7026C000	00000000	00280200	E00000C0
65535		00007000	70273000	00000000	00280200	E00000C0

```

0000           add     [eax],al
c:\ Windows 7 x64
>65535sects.exe
* 65535 physically identical, virtually executed sections

```

```

4D 5A FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
FF FF FF FF-FF FF FF FF-FF FF FF FF-FF FF FF FF
50 45 00 00-4C 01 01 00-FF FF FF FF-FF FF FF FF
FF FF FF FF-E0 00 02 01-0B 01 FF FF-FF FF FF FF
FF FF FF FF-FF FF FF FF FF-00 10 00 00-FF FF FF FF
FF FF FF FF-00 00 40 00-00 10 00 00-00 02 00 00
FF FF FF FF-FF FF FF FF FF-04 00 FF FF-FF FF FF FF
00 20 00 00-60 01 00 00-FF FF FF FF-03 00 7F FA
FF FF FF

```

Name	RVA	Size
Export	FFFFFFFF	FFFFFFFF
Import	00001050	FFFFFFFF
Resource	00000000	FFFFFFFF
Exception	FFFFFFFF	FFFFFFFF
Security	FFFFFFFF	FFFFFFFF
Fixups	FFFFFFFF	FFFFFFFF
Debug	FFFFFFFF	00000000
Description	FFFFFFFF	FFFFFFFF
MIPS GP	FFFFFFFF	FFFFFFFF
ILS	00000000	FFFFFFFF
Load config	00000000	FFFFFFFF
Bound Import	00000000	FFFFFFFF
Import Table	00000000	FFFFFFFF
Delay Import	FFFFFFFF	FFFFFFFF
COM Runtime	00000000	FFFFFFFF
(Reserved)	FFFFFFFF	FFFFFFFF

```

Intel1386
FFFFFFFF
010B
65535.
4.65
FFFFFFFF
FFFFFFFF
00000160
FFFFFFFF
Console
00000200
FFF/00001FFF
4294967295

```

```

00 00 00
00 00 00
00 00 00 00-00 00 00
00 00 00 00-00 00 00
68 18 10 40-00 FF 15
00 FF 15 00-10 40 00
c:\ Windows XP
>maxvals.exe
* a PE with a maximal values in the headers

```

```

tls_noEP.exe  ↓FRO ----- a32 PE .00401000 [High] 8_21 (c:\SSB
tls1:  push     000401024 ;' * Exiting TLS with no EP
00401005: call     printf
0040100B: add     esp,4
0040100E: nop
0040100F: mov     d,1&tls1],tls2 --f2 --f3
00401019: nop
0040101A: push   0
0040101C: call   ExitProcess
00401022: int    3
00401024: 1and   [edx],ch
00401026: and    [ebp][078],a

```

```

c:\ Windows XP
>tls_noEP.exe
* Exiting TLS with no EP
# 1st ILS call, ExitPr
# 2nd TLS call

```

```

00011000 4883ec28      sub     rsp,28h
00011004 8d0d12000000     lea    ecx,[image00000000_00010000+0x101c
0001100a ff15d8000000     call   qword ptr [image00000000_00010000+
00011010 31c9             xor    ecx,ecx

```

```

IBKNOR~1.EXE  ↓FRO ----- a64 PE+.0FFFFFFF'FFFF1000
FFFF1000: 483EC28      sub     rsp,028 ;'<
FFFF1004: 8D0D12000000 lea    ecx,10FFFFFFF'FFFF101C1 ;' * ke
FFFF100A: FF15D8000000 call   printf
FFFF1010:
FFFF1012:
FFFF1018:
FFFF101C:
FFFF101E:
FFFF1021:
FFFF1022:

```

Count of sections		
Symbol table	00000000[00000000]	Thu Jan 01 01:
Size of optional header	0000	Magic optional header
Linker version	0.00	OS version
Entry point	FF000000	System version
		Size of code

c:\ Windows 7

```

>ibknoreloc64.exe
* kernel IB + RIP-relative code (PE32+)

```

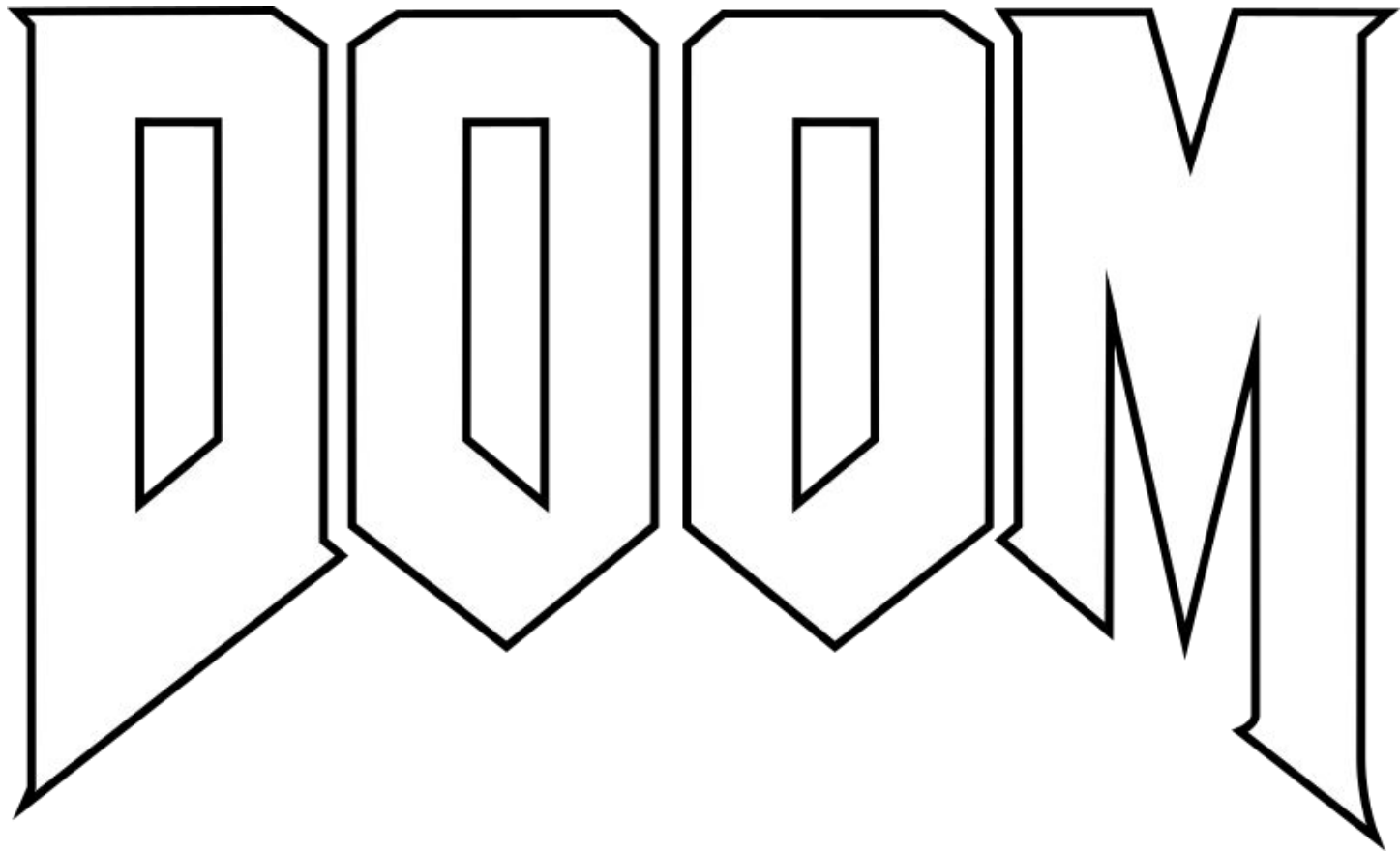
Some of these failed a lot of software.

# Consequence?

- 'corkami-proof' software
- raises the bar for everyone
- become a hub of knowledge
  - "I can't share the sample", but from the knowledge, my own file will be shared  
⇒ even useful for the original contact

**Conclusion**

We're



-ed

**We probably have to witness  
the burning of  
*a digital* "Library of Alexandria"  
before we change anything.**

(because money)

**No matter the kind of format,  
we can't trust files:  
"specifications" ?  
more like *gentle introductions!***

**Or maybe something like  
religious texts (with philosophical suggestions)  
not accurate descriptions of reality.**

**Many more file abuses will come!  
It doesn't get you any bug bounty,  
but plenty of new classes  
of abuse to discover:  
compression, network, cryptography,  
file systems...**



# Rules of thumb

- abuse your own format
  - double-check your specs -- with a twisted mind!
- open-source, unit-tested library
- consistency, technical common sense
- stop stacking features!

# How you can help?

## test-case binaries

- share your testing suite
- fuzzing results (seen from code coverage)

⇒ raises the bar for all industries

# A format evolves

- deprecate!
- enforce version numbers
- make it public

we can set open ultimatum for crypto,  
we should do the same for bad files.

# Ack

Phil Paul Arindam Jacob Alex

Christophe Travis Tavis

Sergey Kurt Gabor Miki Gyn

Mat Bart Max

...

**Thank you!**

# **Corkami: 10 years! time to evolve !**

More PoCs, posters, book(s)...  
+ some side projects

⇒ no more [personal] presentations for now

# **FAQ: "do you have any recommended PDF reader"**

Only Adobe Reader handles complex documents and functionalities.

Other are more or less equivalent.

Not a very satisfying answer, I know ;)

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

