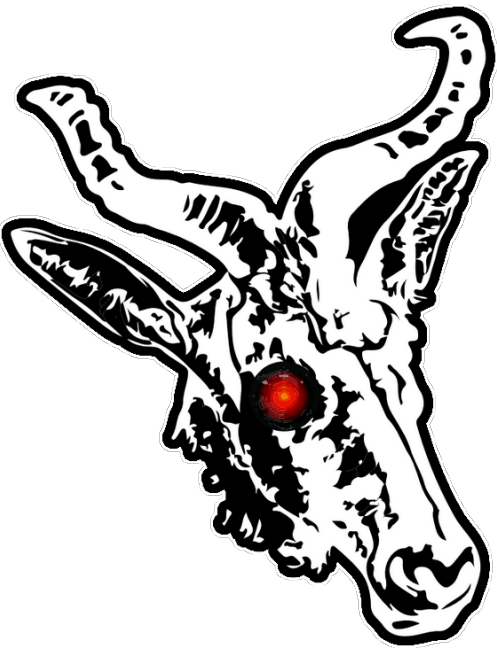


An overview of potential leaks via PDF

Ange Albertini



ANGE ALBERTINI

reverse engineering

VISUAL DOCUMENTATIONS

@angealbertini

ange@corkami.com

<http://www.corkami.com>



Yet another talk on PDF from me?

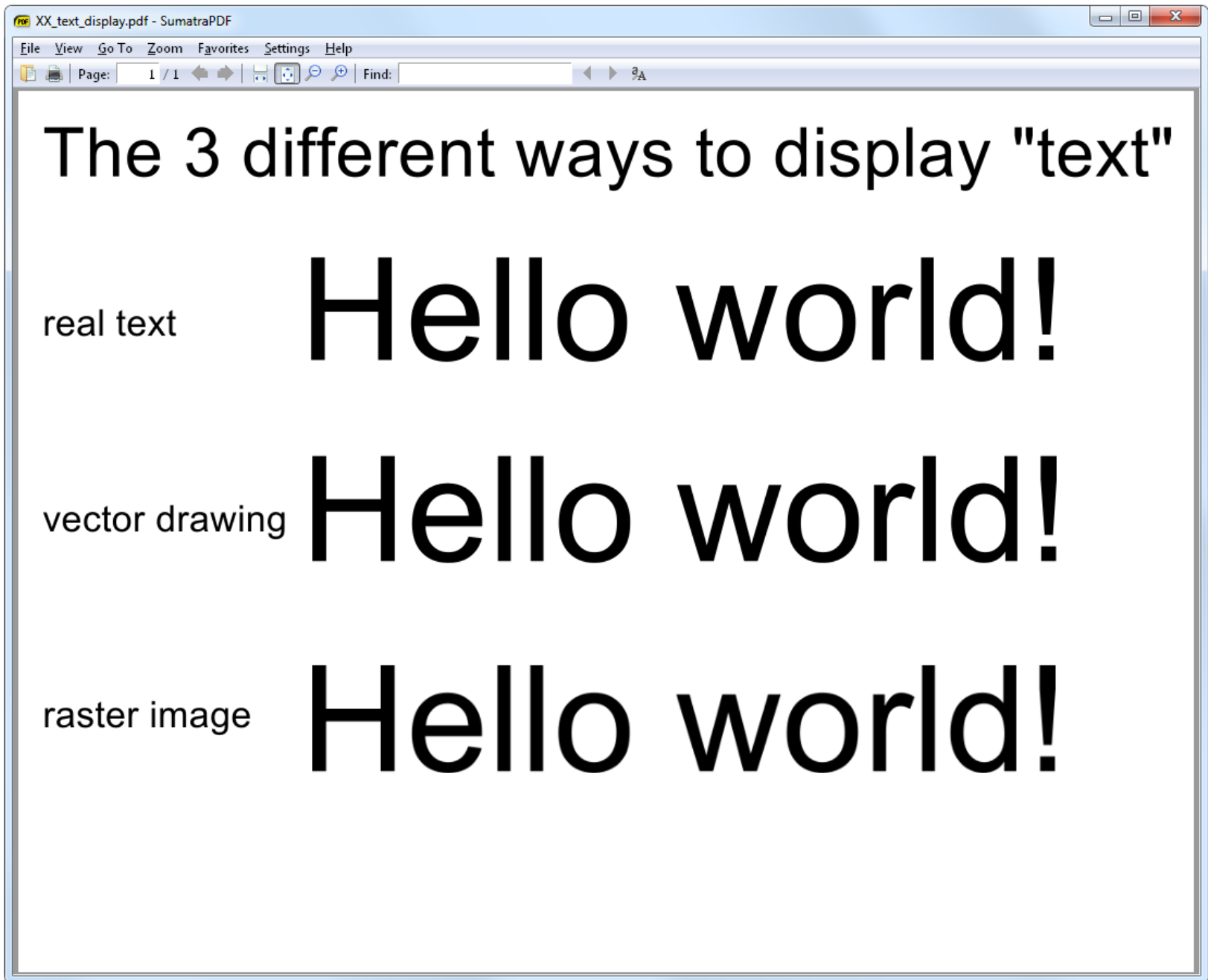
- this one is high-level
 - awareness without the hardcore details
- a new kind of leak happened ITW recently
 - ⇒ it's still worth spreading the knowledge!



[illegible]

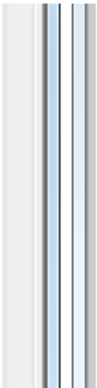
text, image, drawing

potential leaks
via the standard page elements



Pages are made of 3 kinds of 'visual' elements (that can look identical).


```
BT
230 500 Td
/Arial 117 Tf
(Hello world!)Tj
ET
```



real text

Hel

```
BT
230 500 Td
/Arial 117 Tf
(Be llo world!)Tj
ET
```



real text

Bel

1: Text

‘string of the text in the document’

Text

- explicitly spelled in the data
- can be
 - invisible
 - white, invisible style, covered
 - forbidden to copy/paste
 - but this can be disabled instantly
 - mapped to some weird unicode

but still technically there!

⇒ it can still be extracted, often automatically

```
pdftotext -layout ...
```


/ B m

H

raster image

Even if the image is not used (displayed),

Images

- embedded as a dedicated object
 - can be automatically extracted
 - `pdfimages -j -layout ...`
- then referenced in pages' contents
 - useful for multiple uses

⇒ images can be present (and extracted)
even if not used

Images

- JPEG are stored **as-is** (the complete file)

Extra risk: leak via thumbnail, EXIF, RDF

```
% the first l is drawn by a rectangle  
231.691 245.727 10.547 85.898 re % re = rectangle
```

```
% the second l is drawn by 4 points
```

```
258.41 245.727 m
```

```
258.41 331.625 l
```

```
268.957 331.625 l
```

```
268.957 245.727 l
```

```
h
```

vector drawing **Hello**

```
% the first l is drawn by a rectangle  
231.691 245.727 10.547 85.898 re % re = rectangle
```

```
% the second l is drawn by 4 points
```

```
258.41 245.727 m
```

```
258.41 331.625 l
```

```
268.957 331.625 l
```

```
368.957 245.727 l
```

```
h
```

vector drawing **Hel** 

3: drawings
sequences of graphical operators

Drawings

(rectangles, lines...)

- the information is not trivial to extract
- can still be modified without any problem
 - remove covering layers (concealship)



Importing a **specific** part of a **confidential** PDF



With OSX Preview: select **area**, then paste in a new document...


```
$ du -b cropme.pdf cropped.pdf
595      cropme.pdf
10203    cropped.pdf
```



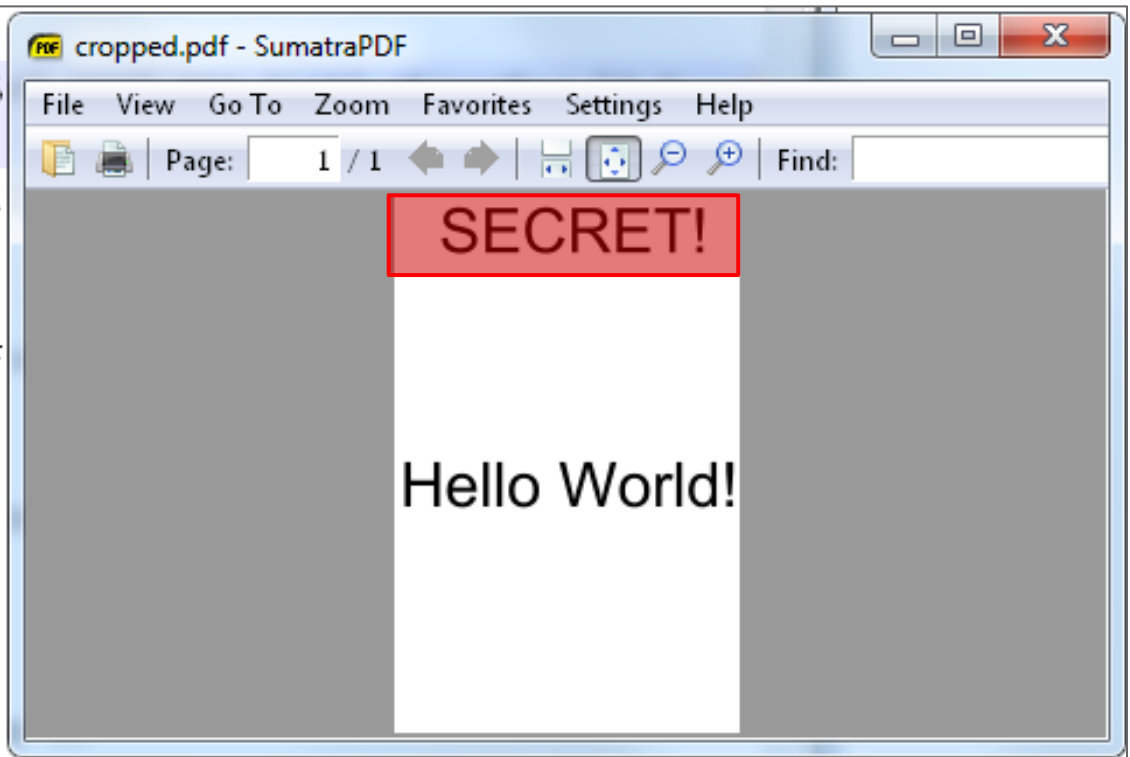
So you get a new document, showing only what you wanted...
(cropme.pdf is much smaller because it was hand-written, while cropped.pdf is bloated)

```
/MediaBox [0 0 612 950]  
/CropBox [6.123547 377.4666 6  
612 950] /TrimBox  
[0 0 612 950] /ArtBox [0 0 61  
endobj  
6 0 obj  
<< /ProcSet [ /PDF /Text ] /F  
endobj  
3 0 obj  
<< /Type /Pages /MediaBox [0  
R ] >>  
endobj  
8 0 obj  
<< /Type /Catalog /Pages 3 0
```



Risk: it's actually the **same** content with an extra 'limiting view'!

```
/MediaBox [0 0 612 950]  
/CropBox [6.123547 377.4666 612 950] /TrimBox  
[0 0 612 950] /ArtBox [0 0 612 950] endobj  
6 0 obj  
<< /ProcSet [ /PDF /Text ] /Resources << /Font << /F1 << /Type /Font /Subtype /Type1 /BaseFont /Helvetica /Encoding /WinAnsiEncoding /ToUnicode << /Type /Pages /MediaBox [0 0 612 950] /Resources << /Font << /F1 << /Type /Font /Subtype /Type1 /BaseFont /Helvetica /Encoding /WinAnsiEncoding /ToUnicode << /Type /Catalog /Pages 3 0 R >> endobj  
8 0 obj  
<< /Type /Catalog /Pages 3 0 R >> endobj
```



If you remove the “CropBox”, you get back the **original** content.

Importing

- Copy/paste from OSX preview
- Import via LaTeX
- ...?

What it actually does:

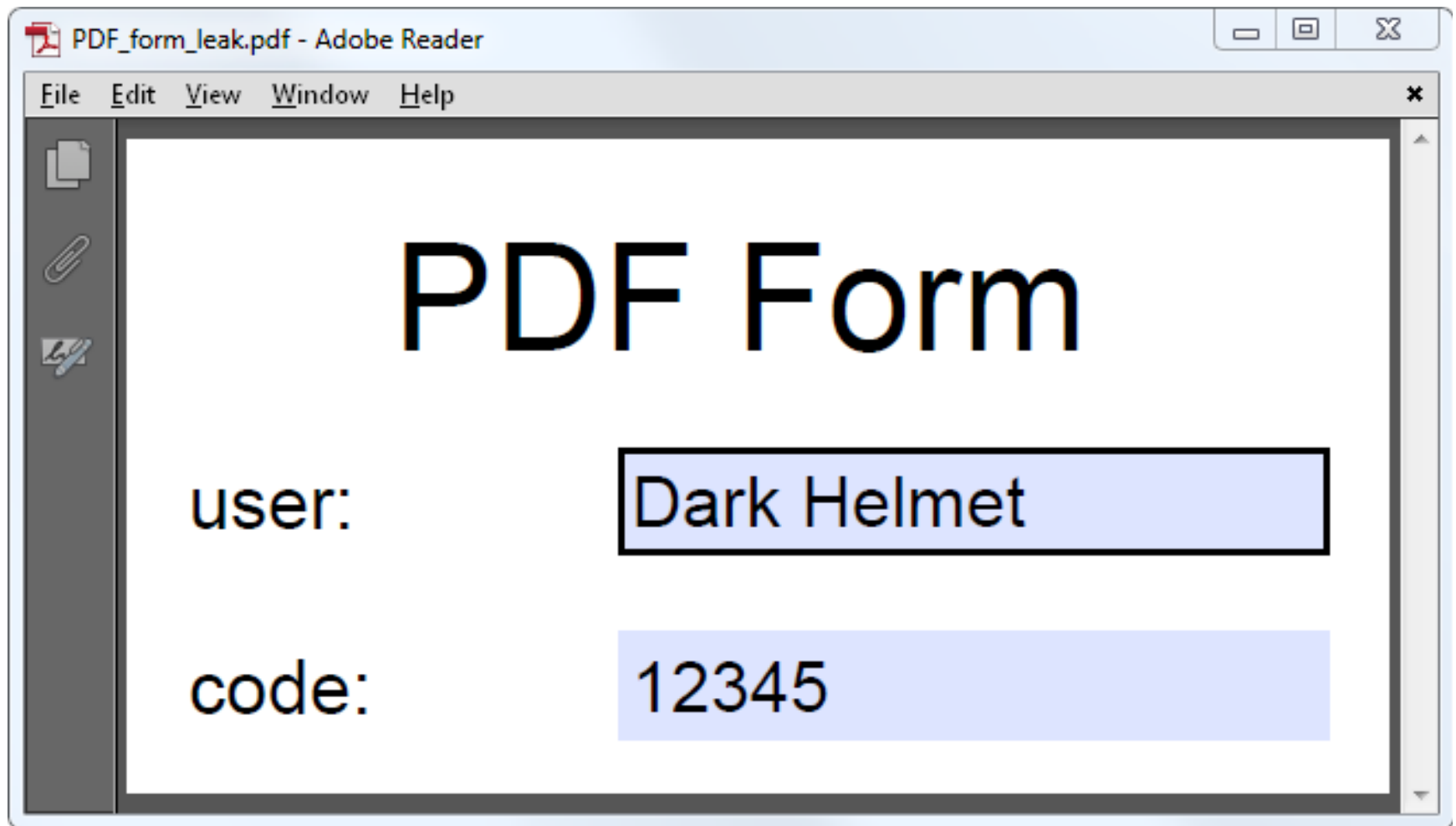
- 1/ imports the whole doc
(to prevent incompatibilities)
- 2/ adds a limiting view

Risk: the original content is still there!

Incremental updates

updates (even deletions) are appended,
like in Microsoft Office, etc...

⇒ “save as...” a new document to prevent it



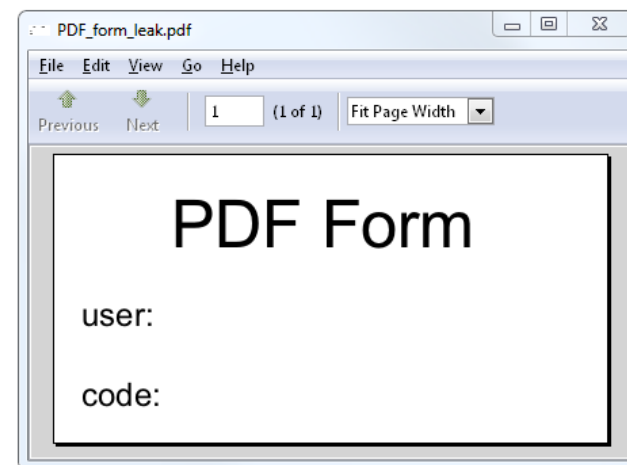
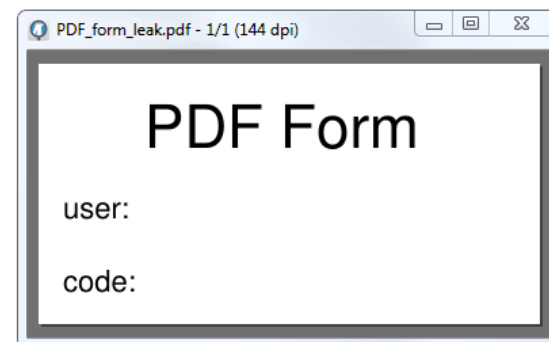
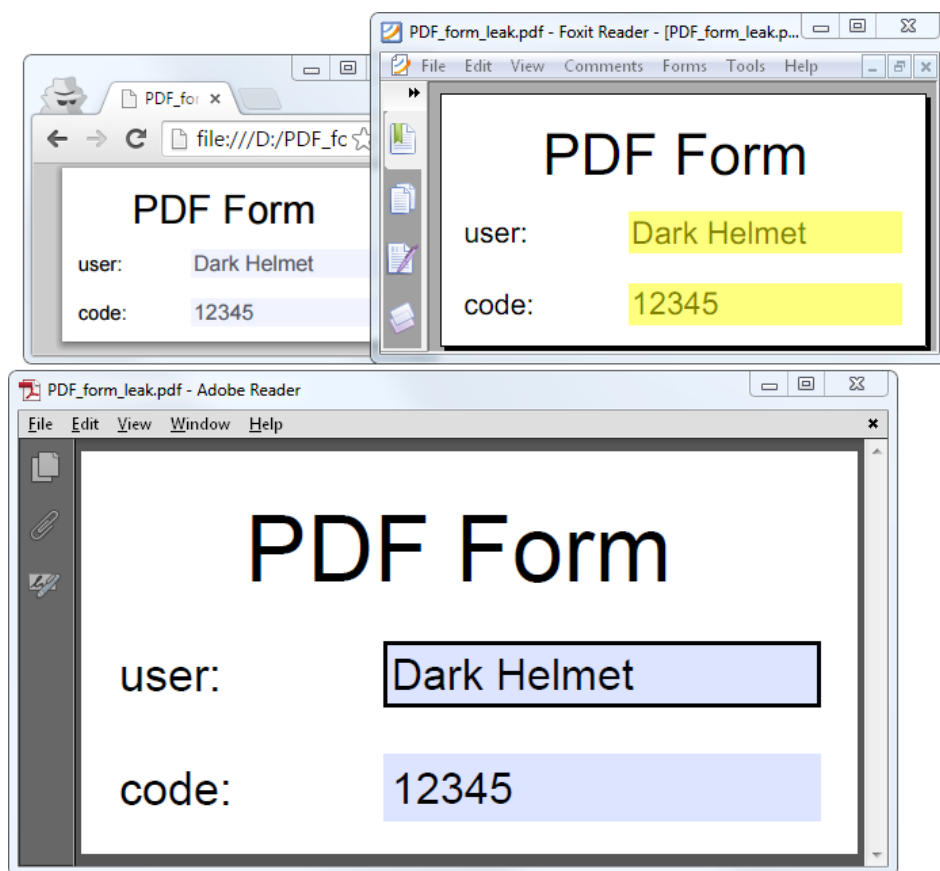
Forms

Forms

- Time saver:
 - type (copy/paste) your info in the doc, **then** print!
 - you can even **save** the info in the doc
 - this info is not stored like standard text

Risk:

you spread an updated document
containing private info!



Some readers may **not** show the saved information!

Forms

- Forms are not always supported
 - you won't even get a warning!
- Content is not stored like standard text
 - not as easy to extract, but still there!

Bigger risk :

Just opening the file to double-check
may be not enough!

The only *fully* reliable way ?

(the one that *NSA* uses...)

Convert pages to pictures !

Just use Imagemagick convert
then import to a **new** PDF

Damn ugly, but fully reliable.

Conclusion

PDF sucks to prevent leaks

PDF is a monster for attack surface
(and metadata embedding)

No free PDF 'dissector'
because we only focus on malware

⇒ No solution anytime soon

(Btw, how much is worth the map of a petroleum reservoir ?)

Questions?

That was just ITW examples of leaks,
other kind of leaks may be possible.

@angealbertini

Note:
this PDF is also a ZIP,
containing the PoCs
shown in the document.

