Move your mouse down here!

horrifying-pdf-experiments

Like many of you, I always thought of PDF as basically a benign format, where the author lays out some text and graphics, and then the PDF sits in front of the reader and doesn't do anything. I heard offhand about vulnerabilities in Adobe Reader years ago, but didn't think too much about why or how they might exist.

That was why Adobe made PDF at first[1], but I think we've established that it's not quite true anymore. The 1,310-page PDF specification (actually a really clear and interesting read) specifies a bizarre amount of functionality, including:

- Embedded Flash
- Audio and video annotations
- 3D object annotations (!)
- Web capture metadata
- Custom math functions (including a Turing-incomplete subset of PostScript)
- Rich text forms using a subset of XHTML and CSS
- File and file-collection attachments

but most interestingly...

- JavaScript scripting, using a completely different standard library from the browser one

Granted, most PDF readers (besides Adobe Reader) don't implement most of this stuff. *But Chrome does implement JavaScript!* If you open a PDF file like this one in Chrome, it will run the scripts. I found this fact out after following this blog post about how to make PDFs with JS.

There's a catch, though. Chrome only implements a *tiny* subset of the enormous Acrobat JavaScript API surface. The API implementation in Chrome's PDFium reader mostly consists of stubs like these:

```
FX_BOOL Document::addAnnot(IJS_Context* cc,
                           const CJS_Parameters& params,
                           CJS_Value& vRet,
                           CFX_WideString& sError) {
  // Not supported.
  return TRUE;
}
FX_BOOL Document::addField(IJS_Context* cc,
```

[1] In fact, I got interested in PDF a couple weeks ago because I'd been reading these random Don Hopkins posts about NeWS, the system supposedly like AJAX but done in the 80s, and so I got interested in PostScript.

Ironically, PDF was a reaction to PostScript, which was too expressive (being a full programming language) and too hard to analyze and reason about. PDF remains a big improvement in that sense, I think, but it's still funny how it's grown all these features.