12 How to Manually Attach a File to a PDF

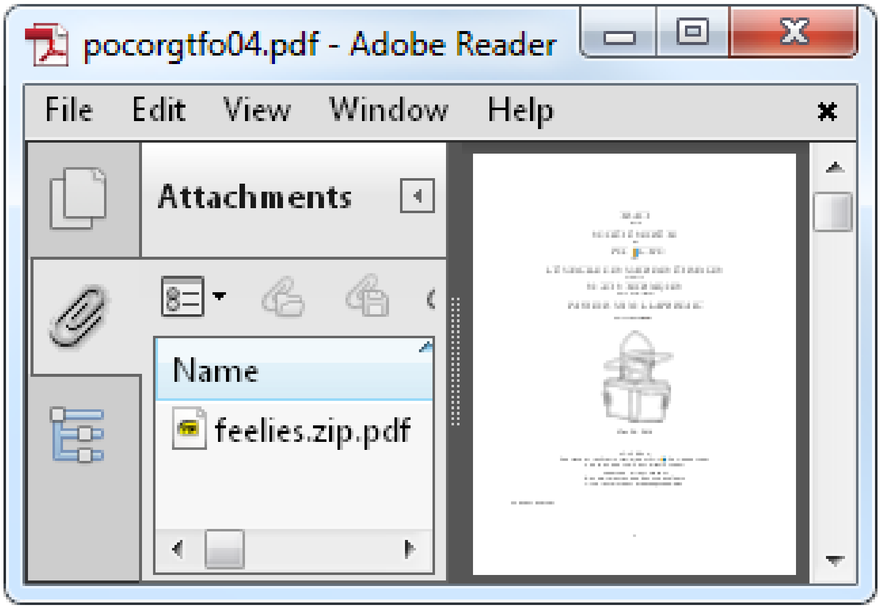
by Ange Albertini

If you followed the PoC||GTFO's March of the Polyglots to date, you may have noticed that until now the feelies were added in a dummy object at the end of the PDF document. That method kept `unzip(1)` happy, and Adobe PDF tools were none the wiser.

Yet Adobe in its wisdom created its own way of attaching files to a PDF!

> One of the great features of PDF is its ability to carry attached files, just as e-mail messages can carry attached files. Any kind of file, and any number of files, can be sucked into a PDF file. These are held internal to PDF as "stream" objects, one of the basic 8 object types from which all PDF content is built (numbers, arrays, strings, true, false, names, dictionaries and streams). Streams start with a dictionary object but then carry along an arbitrarily long sequence of arbitrary 8-bit bytes. Stream objects meet the generic description for disk files quite well.
>
> —Jim King at Adobe

So, dear reader, prepare to be sucked in into PDF feature(creep) greatness![21]



Of course, we could use Adobe software to attach the feelies, but this is not the Way of the PoC. Instead, we'll use our trusty `pdflatex(1)`.

Pdflatex allows us to directly create our own PDF objects from the TeX source, whether they are stream or standard objects. For Adobe tools to see a PDF attachment, we need to create 3 objects:

- the stream object with the attached file contents;
- a file specification object with the filename used in the document;
- an annotation object with the /FileAttachment subtype.

[21] *Some alarmist neighbors predict that the Universe will gravitationally collapse upon itself due to uncontrolled PoC||GTFO expansion. Fear not, neighbors: an international action on PoC footprint is coming! On a second thought, though, since you are all Merchants of Dire PoC now, maybe fear twice as hard? –PML*

There are a couple of things to keep in mind. First, Adobe Reader refuses to extract attachments with a ZIP extension, so we'll need to use a different one. For the plain old `unzip` still to work on the resulting PDF file (after a couple of fixes), we must make sure our attachment is stored in the PDF byte-for-byte, without additional PDF compression.

Here is the code we need. Note that after creating our PDF objects, we can refer to them via `\pdflastobj`; to output the actual value, we prepend that reference with the `\the` keyword.

```
\begingroup
  \pdfcompresslevel=0\relax
    \immediate\pdfobj stream
        attr {/Type /EmbeddedFile}  file {feelies.zip}
  \immediate\pdfobj{<<
        /Type /Filespec   /F (feelies.zip.pdf)  /EF <</F \the\pdflastobj\space 0 R>>
    >>}
  \pdfannot{
        /Subtype /FileAttachment  /FS \the\pdflastobj\space 0 R
        /F 2 % Flag: Hidden
    }
\endgroup
```

Finally, for some reason Adobe software fails to see an annotation object when it's the last one in the file. To work around this, we'll just have to make sure we have some text after that object.

12.1 Increasing compatibility

Sadly, after we use this method and put the (extension-renamed) ZIP into PDF as a standard attachment, plain old `unzip` will fail to unpack it. To `unzip`, the file doesn't look like a valid archive: the actual ZIP contents are neither located near the start of the file (because it's a TrueCrypt polyglot) nor at the end (because our document is big enough so the XREF table is bigger than the usual 64Kb threshold). Let's help `unzip` to find the ZIP structures again!

Luckily, this is easy to do. All we need is to duplicate the last structure of the ZIP file—the End of Central Directory—which points to the body, the Central Directory. This structure is just 22 bytes long, so it won't make a big difference. When duplicating, we change the offset to the Central Directory so that it's pointing to the correct place in the PDF body. We then need to adjust the offsets in each directory entry so that our files' data is still reachable—and voilà, we have an attachment that is visible both to the fancy Adobe tools and to the good old classic `unzip`!