18:10 Easy SHA-1 Colliding PDFs with PDFLaTeX.

by Ange Albertini

In the summer of 2015, I worked with Marc Stevens on the re-usability of a SHA1 collision: determining a prefix could enable us to craft an infinite amount of valid PDF pairs, with arbitrary content with a SHA-1 collision.

```
000:  .% .P .D .F .- .1 .. .3 \n .% E2 E3 CF D3 \n \n
010:  \n .1    .0    .o .b .j \n .< .< ./ .W .i .d .t
020:  .h    .2    .0    .R ./ .H .e .i .g .h .t    .3
030:     .0    .R ./ .T .y .p .e    .4    .0    .R ./
040:  .S .u .b .t .y .p .e    .5    .0    .R ./ .F .i
050:  .l .t .e .r    .6    .0    .R ./ .C .o .l .o .r
060:  .S .p .a .c .e    .7    .0    .R ./ .L .e .n .g
070:  .t .h    .8    .0    .R ./ .B .i .t .s .P .e .r
080:  .C .o .m .p .o .n .e .n .t    .8 .> .> \n .s .t
090:  .r .e .a .m \n FF D8 FF FE 00 24 .S .H .A .- .1
0a0:     .i .s    .d .e .a .d .! .! .! .! .! 85 2F EC
0b0:  09 23 39 75 9C 39 B1 A1 C6 3C 4C 97 E1 FF FE 01
0c0:  ??
```

The first SHA-1 colliding pair of PDF files were released in February 2017.[61] I documented the process and the result in my "Exploiting hash collisions" presentation.

The resulting prefix declares a PDF, with a PDF object declaring an image as object 1, with references to further objects 2–8 in the file for the properties of the image:

```
PDF signature                 000: %PDF-1.3
non-ASCII marker              009: %âãÏÓ
object declaration            011: 1 0 obj
image object properties       019: <</Width 2 0 R/Height 3 0 R/Type 4 0 R
                                    /Subtype 5 0 R/Filter 6 0 R
                                    /ColorSpace 7 0 R/Length 8 0 R
                                    /BitsPerComponent 8>>
stream content start          08e: stream
JPEG Start Of Image           095: FF D8        length: 36
JPEG comment                  097: FF FE 00 24
hidden death statement        09b:   SHA-1 is dead!!!
randomization buffer          0ad:   85 2F .. .. 97 E1
JPEG comment                  0bd: FF FE 01
start of collision block      0c0:           ??   byte with a xor
                                                  difference of 0x0C
                                    length: 01??
```

The PDF is otherwise entirely normal. It's just a PDF with its first eight objects used, and with a image of fixed dimensions and colorspace, with two different contents in each of the colliding files.

The image can be displayed one or many times, with optional clipping, and the raw data of the image can be also used as page content under specific readers (non browsers) if stored losslessly repeating lines of code eight times.

The rest of the file is totally standard. It could be actually a standard academic paper like this one.

We just need to tell PDFLaTeX that object 1 is an image, that the next seven objects are taken, and do some postprocessing magic: since we can't actually build the whole PDF file with the perfect precision for hash collisions, we'll just use placeholders for each of the objects. We also need to tell PDFLaTeX to disable decompression in this group of objects.

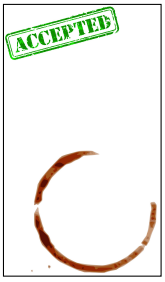
Here's how to do it in PDFLaTeX. You may have to put that even before the `documentclass` declaration to make sure the first PDF objects are not reserved yet.

```
\begingroup

 \pdfcompresslevel=0\relax

 \immediate\pdfximage width 40pt {<foo.jpg>}

 \immediate\pdfobj{65535}          %/Width
 \immediate\pdfobj{65535}          %/Height
 \immediate\pdfobj{/XObject}       %/Type
 \immediate\pdfobj{/Image}         %/SubType
 \immediate\pdfobj{/DCTDecode}     %/Filters
 \immediate\pdfobj{/DeviceGray}    %/ColorSpace
 \immediate\pdfobj{123456789}      %/Length

\endgroup
```

Then we just need to get the reference to the last PDF image object, and we can now display our image wherever we want.

```
\edef \shattered{
            \pdfrefximage\the\pdflastximage}
```

We then just need to actually overwrite the first eight objects of a colliding PDF, and everything falls into place.[62] You can optionally adjust the XREF table for a perfectly standard, SHA-1 colliding, and automatically generated PDF pair.



[61] `unzip pocorgtfo14.pdf shattered.pdf`

[62] See `https://alf.nu/SHA1` or `unzip pocorgtfo18.pdf sha1collider.zip`.