

Analysis:

Command: "pdftocairo -png poppler-pdftocairo-poc temp.png"

Assert() at line 1293 in pdftocairo.cc

```
1289 #ifndef NDEBUG
1290     // Clear the cairo font cache. If all references to font faces or
1291     // scaled fonts have not been released this function will
1292     // assert. If this occurs we have found a memory leak.
1293     cairo_debug_reset_static_data();
1294 #endif
1295
1296     return 0;
1297 }
1298
```

⊙ (afl_env) (base) user@ubuntu:~/zgd/AFLProject/pdf_parsers/poppler-25.04.0/build/utis\$./pdftocairo -png ../../../../pdf_fuzz/poppler-master/pdftocairo/analyze_crashes/afl++/poppler-pdftocairo-poc temp.png
Internal Error: cairo context error: error occurred in libfreetype
cairo error: error occurred in libfreetype
Syntax Warning: Unknown font type: 'Type1!'
pdftocairo: ../../../../src/cairo-ft-font.c:503: _cairo_ft_unscaled_font_fini: Assertion `unscaled->face == NULL' failed.
Aborted (core dumped)

We run the program with gdb, run to `cairo_debug_reset_static_data()`; and dump the heap memory, and find all the stream information leaks of 6 0 obj objects.

000C6FC0	6A 0A 0A 36 20 30 20 6F 62 6A 0A 20 20 3C 3C 20	j . . 6 0 o b j . < <
000C6FD0	2F 4C 65 6E 67 74 68 20 38 38 20 3E 3E 0A 73 74	/ L e n g t h 8 8 > > . s t
000C6FE0	72 65 61 6D 0A 20 20 71 20 31 20 30 2E 33 20 30	r e a m . q 1 0 . 3 0
000C6FF0	2E 33 20 72 67 20 30 20 30 20 35 39 35 20 38 34	. 3 r g 0 0 5 9 5 8 4
000C7000	32 20 72 65 20 46 20 51 20 42 54 20 2F 46 31 20	2 r e F Q B T / F 1
000C7010	32 32 20 54 66 20 33 30 20 38 30 30 20 54 64 20	2 2 T f 3 0 8 0 0 T d
000C7020	28 54 68 69 73 20 69 73 20 74 68 65 20 32 6E 64	(T h i s i s t h e 2 n d
000C7030	20 70 61 67 65 29 20 54 6A 20 45 54 0A 65 6E 64	p a g e) T j E T . e n d
000C7040	73 74 72 65 61 6D 0A 65 6E 64 6F 62 6A 0A 0A 78	s t r e a m . e n d o b j . . x
000C7050	72 65 66 0A 30 20 37 0A 30 30 30 30 30 30 30 30	r e f . 0 7 . 0 0 0 0 0 0 0 0
000C7060	30 30 20 36 35 35 33 35 20 66 20 0A 30 30 30 20	0 0 6 5 5 3 5 f . 0 0 0

Fix recommendation:

zero- fill or securely free heap buffers holding PDF streams before calling `cairo_debug_reset_static_data()`.