

[New issue](#)[Jump to bottom](#)

Heap use-after-free still exists in the bit_copy_chain #497

Open 0xdd96 opened this issue on Jun 17 · 1 comment

Assignees



Labels

bug fuzzing

0xdd96 commented on Jun 17

Vulnerability description

version: [0.12.4.4608](#) & latest commit [f2dea29](#)poc: [poc](#)

command: ./dwgrewrite poc

This is similar to issue [#364](#) and others, but it seems that the patch [e95cc1e](#) has not fully fixed them.

Here is the trace reported by ASAN:

```
==28024==ERROR: AddressSanitizer: heap-use-after-free on address 0x7ffff3b65800 at pc
0x5555564f67f6 bp 0x7fffffff6760 sp 0x7fffffff6750
READ of size 1 at 0x7ffff3b65800 thread T0
#0 0x5555564f67f5 in bit_read_RC libredwg/src/bits.c:317
#1 0x5555564f67f5 in bit_copy_chain libredwg/src/bits.c:3352
#2 0x555556105ec6 in obj_flush_hdlstream libredwg/src/encode.c:833
#3 0x555556105ec6 in dwg_encode_PLANESURFACE_private libredwg/src/dwg.spec:9150
#4 0x5555563a57df in dwg_encode_PLANESURFACE libredwg/src/dwg.spec:9136
#5 0x5555563a57df in dwg_encode_variable_type libredwg/src/classes.inc:247
#6 0x5555563ab3d0 in dwg_encode_add_object libredwg/src/encode.c:4432
#7 0x5555563c914c in dwg_encode libredwg/src/encode.c:2769
#8 0x55555575ca00 in dwg_write_file libredwg/src/dwg.c:429
#9 0x555555758a3f in main libredwg/programs/dwgrewrite.c:350
#10 0x7ffff726f0b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x240b2)
#11 0x55555575924d in _start (libredwg/build-ASAN/dwgrewrite+0x20524d)

0x7ffff3b65800 is located 0 bytes inside of 208896-byte region [0x7ffff3b65800,0x7ffff3b98800)
freed by thread T0 here:
#0 0x7ffff7699ffe in __interceptor_realloc (/lib/x86_64-linux-gnu/libasan.so.5+0x10dffe)
```

```
#1 0x5555564f532c in bit_chain_alloc_size libredwg/src/bits.c:3046
#2 0x5555564f532c in bit_chain_alloc libredwg/src/bits.c:3062
#3 0x5555564f532c in bit_copy_chain libredwg/src/bits.c:3339
#4 0x1900000105 (<unknown module>)
```

previously allocated by thread T0 here:

```
#0 0x7ffff7699ffe in __interceptor_realloc (/lib/x86_64-linux-gnu/libasan.so.5+0x10dffe)
#1 0x5555564f396f in bit_chain_alloc_size libredwg/src/bits.c:3046
#2 0x5555564f396f in bit_chain_alloc libredwg/src/bits.c:3062
#3 0x31fff (<unknown module>)
```

Vulnerability analysis

When running to `bit_copy_chain`, both `dat->chain` and `tmp_dat->chain` point to `0x7ffff3b65800` (see the gdb output below).

This will lead to use-after-free, since line 3339 calls `realloc`, which frees the chunk `0x7ffff3b65800`, and line 3352 ties to read from the freed chunk.

Also note that the comment says `bit_copy_chain` *Copy the whole content of tmp_data to dat, and reset tmp_dat*, so why `dat->chain = tmp_dat->chain` in this PoC should be further investigated.

[libredwg/src/bits.c](#)

Lines 3333 to 3360 in f2dea29

```
3333 void bit_copy_chain (Bit_Chain *restrict dat, Bit_Chain *restrict tmp_dat)
3334 {
3335     unsigned long i;
3336     unsigned long dat_bits = bit_position (tmp_dat);
3337     unsigned long size = tmp_dat->byte;
3338     while (dat->byte + size > dat->size)
3339         bit_chain_alloc (dat);
3340     // check if dat is byte aligned, tmp_dat always is. we can use memcpy then.
3341     if (!dat->bit)
3342     {
3343         assert(!tmp_dat->bit);
3344         memcpy (&dat->chain[dat->byte], &tmp_dat->chain[0], size);
```

```
pwndbg> p *dat
```

```
$6 = {
  chain = 0x7ffff3b65800,
  size = 208896,
  byte = 204890,
  bit = 6 '\006',
  opts = 1 '\001',
  version = R_2000,
  from_version = R_2004,
  fh = 0x0
}
```

```
pwndbg> p *tmp_dat
$10 = {
  chain = 0x7ffff3b65800,
  size = 208896,
  byte = 204882,
  bit = 6 '\006',
  opts = 1 '\001',
  version = R_2000,
  from_version = R_2004,
  fh = 0x0
}
```


  **rurban** self-assigned this on Aug 15

  **rurban** added **bug** **fuzzing** labels on Aug 15

rurban commented on Aug 15

Contributor


I think I found some thinkos in the obj_flush_hdlstream logic

 **rurban** added a commit that referenced this issue on Aug 15

 fix obj_flush_hdlstream GH [#497](#) ...

✓ ccc533c

 **rurban** added a commit that referenced this issue on Aug 15

 fix obj_flush_hdlstream GH [#497](#) ...

✓ 2f36577

 **rurban** added a commit that referenced this issue on Sep 5

 fix obj_flush_hdlstream GH [#497](#) ...

✓ 0657a28

 **rurban** added a commit that referenced this issue on Oct 16

 fix obj_flush_hdlstream GH [#497](#) ...

✓ 0b203b7

 **rurban** added a commit that referenced this issue 20 hours ago



Assignees



rurban

Labels

bug **fuzzing**

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

