ELFUTILS - readelf

Source Code Version

Affected Version : Release 0.168
Source Code URL : ftp://sourceware.org/pub/elfutils/0.168/elfutils-0.168.tar.bz2

ELFUTILS is managed privately. We are only able to submit bugs through bugzilla. The team uses git version control under git://sourceware.org/git/elfutils.git . We are unable to provide commit ID as it is manages internally with the develop team.

Whether the PoC is downloadable from Internet

Yes. It is downloadable at https://github.com/asarubbo/poc/blob/master/00226-elfutils-heapoverflow-ebl_object_note_type_name?raw=true

CVE ID

CVE-2017-7608
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-7608

The detailed procedures that trigger the crash

How the project programs are compiled :

1. Decompress the tarball using `tar xvf elfutils-0.168.tar.bz2`
2. Go to the decompressed directory
3. Make a build folder using `mkdir build`
4. Go into the *build* directory
5. Invoke the configure script, points compiler to afl compiler and enable the use of Address Sanitizer. This can be done by this line `AFL_USE_ASAN=1 ./../configure CC=afl-gcc CXX=afl-g++ LD=afl-gcc--disable-shared`
6. Compile/Make the code by using `AFL_USE_ASAN=1 make`

Note: `AFL_USE_ASAN=1` is to enable the use of Address Sanitizer

The exact running arguments

1. Go to the *src* directory after compiling the code

2. Recreate the crash by running the following code ./readelf -a $FILE where the *$FILE* is the input provided

Description about the crashes (program locations of crash, program locations of the root cause)

The program crashes due to heap-buffer over read. The input file can have an empty name file and it does not have any ‘\0’ character to terminate the array character/string read. Hence, the array character will read heap-buffer that it is not supposed to read.

Explanation about the bug fixes

Rather than putting the name into to a char array/string directly. We should check if the name buffer is null.
By changing from:

```
const char *name = data->d_buf + name_offset;
```

to:

```
const char *name = nhdr.n_namesz == 0 ? "" : data->d_buf + name_offset;
```

will prevent the heap-based over read. As it will put a “” / ’\0’ into the char array/string when the name is null.