huntr

Buffer Over-read at parse_rawml.c:1416 in bfabiszewski/libmobi

0

Valid Reported on Apr 25th 2022

Description

Heap-based Buffer Overflow at parse_rawml.c:1416

Build

```
git clone https://github.com/bfabiszewski/libmobi.git
cd libmobi

export CFLAGS="-g -O0 -lpthread -fsanitize=address"
export CXXFLAGS="-g -O0 -lpthread -fsanitize=address"
export LDFLAGS="-fsanitize=address"

./autogen.sh

./configure --disable-shared

make
```

POC

```
./tools/mobitool -e -o ./tmp/ ./poc.mobi
```

poc.mobi

Asan

```
Title: Libmobi sample file
```

```
Author: Bartek Fabiszewski
Subject: Dictionaries
Language: pl (utf8)

Dictionary: pl => en

__
Mobi version: 7
Creator software: kindlegen 2.9.0 (linux)

Reconstructing source resources...
=================================================================
==1088449==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61000
READ of size 8 at 0x610000000160 thread T0
    #0 0x4f8d37 in mobi_reconstruct_infl /home/fuzz/libmobi/src/parse_rawml
    #1 0x4fabbc in mobi_reconstruct_orth /home/fuzz/libmobi/src/parse_rawml
    #2 0x4fd1fb in mobi_reconstruct_links_kf7 /home/fuzz/libmobi/src/parse_
    #3 0x4fd916 in mobi_reconstruct_links /home/fuzz/libmobi/src/parse_rawm
    #4 0x5011d3 in mobi_parse_rawml_opt /home/fuzz/libmobi/src/parse_rawml.
    #5 0x4ff78f in mobi_parse_rawml /home/fuzz/libmobi/src/parse_rawml.c:2
    #6 0x4c98d4 in loadfilename /home/fuzz/libmobi/tools/mobitool.c:852:20
    #7 0x4c8b36 in main /home/fuzz/libmobi/tools/mobitool.c:1051:11
    #8 0x7ffff7a7a0b2 in __libc_start_main /build/glibc-sMfBJT/glibc-2.31/
    #9 0x41d57d in _start (/home/fuzz/libmobi/tools/mobitool+0x41d57d)

Address 0x610000000160 is a wild pointer.
SUMMARY: AddressSanitizer: heap-buffer-overflow /home/fuzz/libmobi/src/pars
Shadow bytes around the buggy address:
  0x0c207fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c207fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c207fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c207fff8000: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c207fff8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c207fff8020: fa fa fa fa fa fa fa fa fa fa fa fa[fa]fa fa fa
  0x0c207fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
```

```
    Heap left redzone:       fa
    Freed heap region:       fd
    Stack left redzone:      f1

    Stack mid redzone:       f2
    Stack right redzone:     f3
    Stack after return:      f5
    Stack use after scope:   f8
    Global redzone:          f9
    Global init order:       f6
    Poisoned by user:        f7
    Container overflow:      fc
    Array cookie:            ac
    Intra object redzone:    bb
    ASan internal:           fe
    Left alloca redzone:     ca
    Right alloca redzone:    cb
    Shadow gap:              cc
  ==1088449==ABORTING
```

Impact

The bug causes the program reads data past the end of the intented buffer. Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash.

CVE
CVE-2022-1534
(Published)

Vulnerability Type
CWE-126: Buffer Over-read

Severity
Medium (6.6)

Registry
Other

Affected Version
*

Visibility

Public

Status

Fixed

Found by

TDHX ICS Security

@jieyongma

pro

Fixed by

Bartek Fabiszewski

@bfabiszewski

unranked

This report was seen 626 times.

We are processing your report and will contact the **bfabiszewski/libmobi** team within 24 hours.
7 months ago

We have contacted a member of the **bfabiszewski/libmobi** team and are waiting to hear back
7 months ago

Bartek Fabiszewski validated this vulnerability 7 months ago

TDHX ICS Security has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Bartek Fabiszewski marked this as fixed in **0.11** with commit **fb1ab5** 7 months ago

Bartek Fabiszewski has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Chat with us

Jamie Slome 7 months ago Admin

Jamie Slome 7 months ago Admin

CWE changed to CWE-126. Read comments in other report.

Bartek 7 months ago Maintainer

@jieyongma
Recently I've received multiple bug reports from different researchers fuzzing libmobi.
Is there any coordinated project aiming at fuzzing the library?
I am asking because I would like to know whether I should still wait for new reports to come or the fuzzing is done and I can publish new release. Do you have any idea?
Thanks for your efforts to secure libmobi!

TDHX 7 months ago Researcher

@bfabiszewski
AFAIK, there is no coordinated project aiming at fuzzing the library.
There is no more crash report when I fuzz the library last week.
Maybe I will give it another try someday later.

TDHX 7 months ago Researcher

@bfabiszewski
It's like fishing, if I saw someone caught a fish in a pond, I will definitely gave it a try ;-)

Bartek 7 months ago Maintainer

@jieyongma
Ok. Thanks for the info!

Bartek 6 months ago Maintainer

@jieyongma
Could you tell me what is the procedure to assign severity levels to CVEs?
This vulnerability, for example, has high severity score in NVD. I cannot agree, as the real security impact of this bug is low. The worst scenario I can think of is crashing user application. You must also force the user to use crafted file. Without address sanitizer there will not even be a crash in such case.
How can it be considered high severity problem?

Chat with us

TDHX 6 months ago Researcher

@bfabiszewski
Please check the following URL:

CVE: https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures
NVD: https://en.wikipedia.org/wiki/National_Vulnerability_Database

As NVD mentioned on there website:
"NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA."

I believe there is someone in NVD(NVD Analysts) assign severity levels to CVEs in their database (NVD). And it's widely used by other parties.

For this vulnerability, their detail assessment information could be found at following URL:

https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2022-1534&vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:H&version=3.1&source=NIST

Bartek 6 months ago Maintainer

@jieyongma
Thanks, I'll check that!

TDHX 6 months ago Researcher

@bfabiszewski
You are welcome :-)
Don't take the severity level too seriously. Maybe at the hacker's point of view, they could cause some serious problem by using this type of vulnerability (Buffer Over-read).

Bartek 6 months ago Maintainer

@jieyongma
I always took it seriously. :)
Now I see that the system is not reliable and misleading.

Sign in to join this conversation

Chat with us