

New issue

Jump to bottom

heap-buffer-overflow in in msadpcm_decode_block #687

Closed andreafioraldi opened this issue on Jan 15, 2021 · 22 comments

Labels Bug
Milestone v1.1.0

andreafioraldi commented on Jan 15, 2021 • edited

Hi,
fuzzing sndfile-info with AFL++ I found a heap-buffer-overflow in in msadpcm_decode_block /home/andrea/real/libsndfile/src/ms_adpcm.c:279

I'm on an x86-64 Ubuntu 20.04 with Clang 10.

The AddressSanitizer report is the following:

```
=====
==24888==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x621000001238 at pc 0x0000005ebdc bp 0x7ffced651bd0 sp 0x7ffced651bc8
WRITE of size 2 at 0x621000001238 thread T0
#0 0x5ebedb in msadpcm_decode_block /home/andrea/libsndfile/src/ms_adpcm.c:279:31
#1 0x5e9cf8 in wavlike_msadpcm_init /home/andrea/libsndfile/src/ms_adpcm.c:171:3
#2 0x566da9 in wav_open /home/andrea/libsndfile/src/wav.c:258:14
#3 0x4cc6d2 in psf_open_file /home/andrea/libsndfile/src/sndfile.c:3080:13
#4 0x4caa5e in sf_open /home/andrea/libsndfile/src/sndfile.c:359:9
#5 0x4c57dd in cart_dump /home/andrea/libsndfile/programs/sndfile-info.c:479:14
#6 0x4c36c3 in main /home/andrea/libsndfile/programs/sndfile-info.c:96:13
#7 0x7f114ca61bf6 in __libc_start_main /build/glibc-S9d23N/glibc-2.27/csu/../csu/libc-start.c:310
#8 0x41b4c9 in _start (/home/andrea/libsndfile/programs/sndfile-info+0x41b4c9)

0x621000001238 is located 0 bytes to the right of 4408-byte region [0x621000000100,0x621000001238)
allocated by thread T0 here:
#0 0x493d82 in calloc (/home/andrea/libsndfile/programs/sndfile-info+0x493d82)
#1 0x5e90b4 in wavlike_msadpcm_init /home/andrea/libsndfile/src/ms_adpcm.c:138:27
#2 0x566da9 in wav_open /home/andrea/libsndfile/src/wav.c:258:14
#3 0x4cc6d2 in psf_open_file /home/andrea/libsndfile/src/sndfile.c:3080:13
#4 0x4caa5e in sf_open /home/andrea/libsndfile/src/sndfile.c:359:9
#5 0x4c57dd in cart_dump /home/andrea/libsndfile/programs/sndfile-info.c:479:14
#6 0x4c36c3 in main /home/andrea/libsndfile/programs/sndfile-info.c:96:13
#7 0x7f114ca61bf6 in __libc_start_main /build/glibc-S9d23N/glibc-2.27/csu/../csu/libc-start.c:310

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/andrea/libsndfile/src/ms_adpcm.c:279:31 in msadpcm_decode_block
Shadow bytes around the buggy address:
 0x0c427fff81f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c427fff8200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c427fff8210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c427fff8220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0x0c427fff8230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c427fff8240: 00 00 00 00 00 00 00[fa]fa fa fa fa fa fa fa
 0x0c427fff8250: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c427fff8260: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c427fff8270: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c427fff8280: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
 0x0c427fff8290: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
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

==24888==ABORTING
```

To reproduce on git master:

```
export CC='clang-10 -fsanitize=address'
export CFLAGS='-g'
./configure --disable-shared
make
./programs/sndfile-info --cart ./sndfile_heap_overflow
```

The testcase that triggers the bug is (decompress it before):

[sndfile_heap_overflow.tar.gz](#)

evpobr commented on Jan 16, 2021

Member

Hi @andreafioraldi, thanks for report.

zodf0055980 commented on Jan 25, 2021 • edited

Contributor

I think this problem is about BlockAlign is not match NumChannels * BitsPerSample / 8 .
In this poc, BlockAlign = 1280, NumChannels = 128, BitsPerSample (Bit Width) = 1.
In wavpack , it have [this check](#) and this POC not pass.

```
if (!WaveHeader.NumChannels || WaveHeader.NumChannels > 256 ||  
    WaveHeader.BlockAlign / WaveHeader.NumChannels < (config->bits_per_sample + 7) / 8 ||  
    WaveHeader.BlockAlign / WaveHeader.NumChannels > 4 ||  
    WaveHeader.BlockAlign % WaveHeader.NumChannels)  
    supported = FALSE;
```

zodf0055980 commented on Jan 26, 2021

Contributor

Oops, maybe blockalign == bitwidth * channels / 8 is only for WAVE_FORMAT_PCM .

libsndfile/src/wavlike.c
Lines 150 to 155 in 251a435

```
150     if (wav_fmt->format == WAVE_FORMAT_PCM && wav_fmt->min.blockalign == 0  
151         && wav_fmt->min.bitwidth > 0 && wav_fmt->min.channels > 0)  
152     {  
153         wav_fmt->min.blockalign = wav_fmt->min.bitwidth / 8 + (wav_fmt->min.bitwidth % 8 > 0 ? 1 : 0);  
154         wav_fmt->min.blockalign *= wav_fmt->min.channels;  
155         psf_log_printf (psf, " Block Align : 0 (should be %d)\n", wav_fmt->min.blockalign);  
156     }
```

In the make test test case, ima_adpcm.wav doesn't have this feature.

```
File : ima_adpcm.wav  
Length : 4668  
RIFF : 4660  
WAVE  
fmt : 20  
Format : 0x11 => WAVE_FORMAT_IMA_ADPCM  
Channels : 1  
Sample Rate : 16000  
Block Align : 512  
Bit Width : 4  
Extra Bytes : 2  
Samples/Block : 1017  
Bytes/sec : 8055 (should be 15)  
fact : 4  
frames : 9153  
data : 4688  
End
```

 andreafioraldi mentioned this issue on Feb 3, 2021

heap-buffer-overflow in wavlike_ima_decode_block #697

 Closed

galaktipus commented on Jul 21, 2021


Hi, is there a plan of fixing the issue?
Or is the fix for #697 sufficient for this issue as well? That is, [9e0e55f](#)

evpobr commented on Jul 21, 2021

Member

Hi @galaktipus , it was fixed: <https://oss-fuzz.com/testcase-detail/5696502087024640>. I guess i forgot to close this issue.

You can use master branch, we tagged unofficial release 1.1.0beta1 there.

 evpobr added this to the v1.1.0 milestone on Jul 21, 2021

ajakk commented on Jul 21, 2021

Hi @galaktipus , it was fixed: <https://oss-fuzz.com/testcase-detail/5696502087024640>. I guess i forgot to close this issue.

You can use master branch, we tagged unofficial release 1.1.0beta1 there.

I don't see a tag, maybe it wasn't pushed? That link is also behind a login, can you point directly to a fix?

 1

evpobr commented on Jul 22, 2021

Member

@galaktipus , actually you can use master branch.

@SoapGentoo , could you create tag for 1.1.0beta1?



SoapGentoo commented on Jul 22, 2021

Member

@evpobr sorry, forgot to push for beta1, pushed now

carnil commented on Jul 23, 2021

[CVE-2021-3246](#) appears to have been assigned for this issue.

evpobr commented on Jul 24, 2021

Member

@carnil, thanks.

evpobr closed this as completed on Jul 24, 2021

krisfed commented on Aug 11, 2021

Thank you very much for fixing this! Just to confirm, is this the [commit](#) and the pull request [#713](#) that fixes this issue?

I would also appreciate any information about the approximate timeline for the next official release... I think this fix is not in 1.0.31, and we are somewhat limited in ability to address CVEs by upgrading to an unofficial beta releases or by patching. So any info about the plans for the next release would be very helpful!



evpobr commented on Aug 12, 2021

Member

Thank you very much for fixing this! Just to confirm, is this the [commit](#) and the pull request [#713](#) that fixes this issue?

Yes.

I would also appreciate any information about the approximate timeline for the next official release... I think this fix is not in 1.0.31, and we are somewhat limited in ability to address CVEs by upgrading to an unofficial beta releases or by patching. So any info about the plans for the next release would be very helpful!

Unfortunately, there are no exact dates. I hope we will release 1.1.0 this year. We're waiting for @arthurt to finish whatever he wanted about the MP3 format, but he is probably very busy at work or maybe resting. We'll wait for him to respond.



arthurt commented on Aug 12, 2021

Member

Uh oh, I thought we were waiting on something else!

evpobr commented on Aug 13, 2021

Member

No 🤔

DerDakon commented on Aug 30, 2021

The CVE states that only 1.0.30 is affected. I think that [e4cc9d3](#) introduced the incomplete check, which is in 1.0.26. But given that there was no check before I assume that even older versions are affected. Can you confirm this, at least that >1.0.26 are affected, and update the CVE information, please?



andreafioraldi commented on Aug 30, 2021

Author

I found the bug fuzzing the 1.0.28 fyi



ajakk commented on Aug 30, 2021

The CVE states that only 1.0.30 is affected. I think that [e4cc9d3](#) introduced the incomplete check, which is in 1.0.26. But given that there was no check before I assume that even older versions are affected. Can you confirm this, at least that >1.0.26 are affected, and update the CVE information, please?

Anyone can request CVE data to be updated:

<https://cveform.mitre.org>

evpobr commented on Aug 30, 2021

Member

Hi everybody. It would be nice to update CVE. Maybe it is better to wait for 1.1.0 with fix and then update description?

krisfed mentioned this issue on Nov 4, 2021

opus: how much difference is expected after writing and reading data back? #788

Closed

MT2017090 mentioned this issue on Dec 14, 2021

1.1.0 official release ? #799

Closed

KamicDemon commented on Jan 18

Hello all,
I'm testing a black box Linux system that supposedly uses a version of libsndfile vulnerable to the CVEs related to this issue. Has made a .wav file PoC that would induce a crash indicative of vulnerability available? It would be helpful to test systems that don't have arbitrary execution access enabled on them. Thank you!

evpobr commented on Jan 18

Member

Hi @KamicDemon

Has made a .wav file PoC that would induce a crash indicative of vulnerability available?

Hmm, i didn't quite understand, do you need a text file or did you create it?

ajak commented on Jan 18

Hello all, I'm testing a black box Linux system that supposedly uses a version of libsndfile vulnerable to the CVEs related to this issue. Has made a .wav file PoC that would induce a crash indicative of vulnerability available? It would be helpful to test systems that don't have arbitrary execution access enabled on them. Thank you!

The test case is in the original report.

KamicDemon commented on Jan 18

Hi @KamicDemon

Has made a .wav file PoC that would induce a crash indicative of vulnerability available?

Hmm, i didn't quite understand, do you need a text file or did you create it?

Hi, thanks for the reply. I may have misunderstood the nature of the vulnerability in question. I assumed it could be triggered simply by playing a crafted WAV file on a device using the vulnerable version of Libsnd. I was asking if anyone had made a .wav PoC file, but forgot to type out the word "anyone"

Assignees

No one assigned

Labels

Bug

Projects

None yet

Milestone

v1.1.0

Development

No branches or pull requests

11 participants

