CVE-2022-23747

Sony OMX audio decoder • Sony Xperia series 1, 5 and Pro • Classic Buffer Overflow
Reported on 06-Apr-22 by Slava Makkaveev (/researcher/Slava Makkaveev)
CPR-ID: 2191
Upload Date: 17-Aug-22

Information

libstagefright_soft_somcalacdec.so HAL library is responsible for decoding the Apple iTunes ALAC/AAC-LC audio format on Sony Xperia smartphones. Privileged android.hardware.media.omx service loads the ALAC library when requested by an Android app, and then calls it to decode the supplied audio frames.

In the libstagefright_soft_somcalacdec.so, an out of bound memory access can occur due to lack of validation of the number of frames being passed during music playback.

The size of the internal output buffer mMixBufferU can be specified using the csd-0 configuration parameter. The numSamples audio parameter is encoded in the audio frame: https://github.com/macosforge/alac/blob/master/codec/ALACDecoder.cpp#L252 (https://github.com/macosforge/alac/blob/master/codec/ALACDecoder.cpp#L252). We can write data outside the mMixBufferU output buffer because no size check is performed: https://github.com/macosforge/alac/blob/master/codec/ALACDecoder.cpp#L320 (https://github.com/macosforge/alac/blob/master/codec/ALACDecoder.cpp#L320).

Impact:

The vulnerability occurs in the context of the privileged media process. If exploited, the attacker can steal media data and gain control over the video and audio stream. On some Sony Xperia devices, the ALAC decoder implemented by Sony is the default decoder. So a malformed audio file can be used for RCE.

Crash trace:

```
Build fingerprint: 'Sony/XQ-BC72/XQ-BC72:12/61.1.A.2.211/061001A0020211
03147541197:user/release-keys'
Revision: '0'
ABI: 'arm'
Cmdline: media.codec hw/android.hardware.media.omx@1.0-service
pid: 10357, tid: 13216, name: mc.alac.decoder  >>> media.codec <<<
uid: 1046
signal 0 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr --------
    r0  ea0c1ec8  r1  00000008  r2  ec24ffff  r3  00000017
    r4  00001ffd  r5  00000008  r6  00000000  r7  00007ff4
    r8  ebec3fec  r9  00000008  r10 ea0c1ec8  r11 41414141
    ip  00000007  sp  e9189ea0  lr  e9192eef  pc  e91926da

backtrace:
      #00 pc 000046da  /vendor/lib/libstagefright_soft_somcalacdec.so
(BitBufferRead+8) (BuildId: 399af8c1929aa1d50e00f65fe9ff9434)
      #01 pc 00004eeb  /vendor/lib/libstagefright_soft_somcalacdec.so
(ALACDecoder::Decode(BitBuffer*, unsigned char*, unsigned int, unsigned
int, unsigned int*)+1130) (BuildId: 399af8c1929aa1d50e00f65fe9ff9434)
      #02 pc 00003ef5  /vendor/lib/libstagefright_soft_somcalacdec.so
(android::SoftALAC::onQueueFilled(unsigned int)+208) (BuildId: 399af8c1
929aa1d50e00f65fe9ff9434)
      #03 pc 00008817  /vendor/lib/libstagefright_softomx.so (android::
SimpleSoftOMXComponent::onMessageReceived(android::sp<android::AMessage
> const&)+266) (BuildId: 95ea0acdaa7e72cc6a7c88be5f296aa9)
      #04 pc 00009a15  /vendor/lib/libstagefright_softomx.so (android::
AHandlerReflector<android::SimpleSoftOMXComponent>::onMessageReceived(a
ndroid::sp<android::AMessage> const&)+52) (BuildId: 95ea0acdaa7e72cc6a7
c88be5f296aa9)
      #05 pc 0000fe05  /vendor/lib/vndk/libstagefright_foundation.so (a
ndroid::AHandler::deliverMessage(android::sp<android::AMessage> const&)
+24) (BuildId: 96b372f41b97c8470a38f16dd9b85bef)
      #06 pc 00012467  /vendor/lib/vndk/libstagefright_foundation.so (a
ndroid::AMessage::deliver()+86) (BuildId: 96b372f41b97c8470a38f16dd9b85
bef)
      #07 pc 0001057d  /vendor/lib/vndk/libstagefright_foundation.so (a
ndroid::ALooper::loop()+488) (BuildId: 96b372f41b97c8470a38f16dd9b85be
f)
      #08 pc 0000ef61  /apex/com.android.vndk.v30/lib/libutils.so (andr
oid::Thread::_threadLoop(void*)+304) (BuildId: 373fcfc8fb18977f88e89ad09552a738)
      #09 pc 0000ea15  /apex/com.android.vndk.v30/lib/libutils.so (thread_data_t::trampoline(thread_data_t const*)+256) (BuildId: 373fcfc8fb18977f88e89ad09552a738)
      #10 pc 00080e57  /apex/com.android.runtime/lib/bionic/libc.so (__pthread_start(void*)+40) (BuildId: 91ef3dc3105c19cbfe9eaa06c9cd1fcb)
      #11 pc 00039e33  /apex/com.android.runtime/lib/bionic/libc.so (__start_thread+30) (BuildId: 91ef3dc3105c19cbfe9eaa06c9cd1fcb)
```

References:
https://research.checkpoint.com/2022/bad-alac-one-codec-to-hack-the-whole-world/ (https://research.checkpoint.com/2022/bad-alac-one-codec-to-hack-the-whole-world/)

Share this:

(http://www.reddit.com/submit?url=https://cpr-zero.checkpoint.com/vulns/cprid-2191/&title=CVE-2022-23747 - CPR-Zero)
Y (http://news.ycombinator.com/submitlink?u=https://cpr-zero.checkpoint.com/vulns/cprid-2191/&t=CVE-2022-23747 - CPR-Zero)
(https://twitter.com/intent/tweet?via=_CPResearch_&url=https://cpr-zero.checkpoint.com/vulns/cprid-2191/&text=CVE-2022-23747 - CPR-Zero)
in (https://www.linkedin.com/shareArticle?mini=true&url=https://cpr-zero.checkpoint.com/vulns/cprid-2191/&title=CVE-2022-23747)