CVE-2022-22675: AppleAVD Overflow in AVC_RBSP::parseHRD

Natalie Silvanovich

The Basics

Disclosure or Patch Date: March 31, 2022

Product: Apple iOS, MacOS

Advisory:

iOS: https://support.apple.com/en-us/HT213219

Mac: https://support.apple.com/en-us/HT213220

Affected Versions:

Reachable by thumbnailing media file: MacOS 12.3 / iOS 15.4

Reachable from local code only: MacOS 12.2.1 / iOS 15.3.1 and previous

First Patched Version: MacOS 12.3.1 / iOS 15.4.1

Issue/Bug Report: N/A

Patch CL: N/A

Bug-Introducing CL: N/A

Reporter(s): an anonymous researcher

The Code

Proof-of-concept:

Partial PoC below triggers patch log output, but does not crash

https://googleprojectzero.github.io/0days-in-the-wild//0day-RCAs/2022/CVE-2022-22675.mp4

Exploit sample: N/A

Did you have access to the exploit sample when doing the analysis? N/A

The Vulnerability

Bug class: Buffer overflow

Vulnerability details:

There is a buffer overflow when processing the Hardware Reference Device (HRD) of an H.264 stream in the function *AVC_RBSP::parseHRD*. The AppleAVD.kext kernel module reads values describing the bitrates of the HRD from the stream in a loop and copies them into a buffer. This buffer has a fixed size of 32 elements, meanwhile the number of elements copied is determined by the *cpb_cnt_minus1* value read from the stream, which can have a maximum value of 255, allowing the buffer to be overflowed.

Note that while the advisories describe the impact of this issue as a local privilege escalation, it is theoretically possible to exploit it to achieve fully-remote code execution in MacOS 12.3/iOS 15.4. These versions use AppleAVD to perform thumbnailing of incoming images in iMessage, so this code path is available to a fully-remote attacker.

Patch analysis:

The patch tests whether the *cpb_cnt_minus1* value is less than 32. If the check fails, it logs “ERROR: hrd.cpb_cnt_minus1” and returns, which terminates decoding

Thoughts on how this vuln might have been found *(fuzzing, code auditing, variant analysis, etc.)*:

The vulnerability causes an overflow into other members of the decoder struct that contains the HRD buffer and does not extend into other allocations on the heap. Despite an in-depth analysis of this issue, we were unable to find a proof-of-concept media file that crashed the system, even after fuzzing with the trigger file above as a template for several days. This means that the condition that allowed the exploit was probably quite subtle, it is unlikely that the bug was found by fuzzing, so it was probably found by a manual audit of the binary.

The Next Steps

Variant analysis

Areas/approach for variant analysis (and why):

Several parsing functions in AppleAVD have recently became reachable when H.264 and H.265 streams are processed, including during thumbnailing of media files. This entire attack surface could use review.

Found variants: None

Structural improvements

Other potential improvements:

The impact of similar vulnerabilities could be reduced by removing the H.264 parameter parsing code from the kernel and running it in a lower-privileged context.

Other References

H.264 specification: https://www.itu.int/rec/T-REC-H.264