

Jump to bottom New issue

A Integer Overflow in function gf_avc_read_pps_bs_internal #1720



(⊙ Closed) treebacker opened this issue on Mar 29, 2021 ⋅ 1 comment

treebacker commented on Mar 29, 2021 • edited →
There is a integer overflow in media_tools/av_parsers.c:5349 , function gf_avc_read_pps_bs_internal . Below code:
<pre>pps_id = gf_bs_read_ue_log(bs, "pps_id"); if (pps_id >= 255) { return -1; } pps = &avc->pps[pps_id]; pps->id = pps_id; pps_id may be a negative number, so will not return. However, avc->pps only has 255 unit, so overflow, which results a crash.</pre>
More than, because of the pps->id = pps_id , the vuln may lead to an any addr write.
In command Line: gpac-info bug3 ubuntu@VM-0-3-ubuntu:~/
In gdb: Breakpoint 3, gf media avc_read_pps_bs_internal (bs=0x555557a4090, avc=0x555557acd50, nal_hdr=0) at media_tools/av_parsers.c:5339 gf_bs_enable_emulation_byte_removal(bs, GF_TRUE); (gdb) n 3341
(gdb) p psysid 3 = 113918469 (gdb) p ave-spps id = 0, entropy_coding mode_flag = 6F FALSE, pic order present = 0, redundant pic ont present = 0, alice group_count = 0, mb_slice group_map type = 0, pic_size in map for you change rate minus! = 0, status = 0, wespfred pred flag = 6F FALSE, weighted bipred idic = 0 \text{\text{Nome}} \text{ (and in the flag = 6F FALSE, pic_order present = 0, redundant pic_ont present flag = 6F FALSE, using the flag = 6F FALSE, pic_order present = 0, redundant pic_ont present flag = 6F FALSE, using the flag = 6F FALSE, verified bipred idic = 0 \text{\text{Nome}} \text{ (and in the flag = 6F FALSE, map ref idix [0] default active minus! = 0}, status = 0, weighted bipred idic = 0 \text{\text{Nome}} \text{ (and in the flag = 6F FALSE, map ref idix [0] default active minus! = 0}, num_ref idx [1] default active minus! = 0}, flag = 6F FALSE, weighted bipred idic = 0 \text{\text{Nome}} \text{ (and in the flag = 6F FALSE, map ref idix [0] default active minus! = 0}, status = 0, slice group_count pape proper = 0, pic_size in map units minus! = 0, slice group_count pape proper = 0, pic_size in map units minus! = 0, slice group_count pape proper = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_count = 0, mb_slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_map type = 0, pic_size in map units minus! = 0, slice group_count = 0, mb_slice group_count = 0,
deblocking filter_control present flag = 6f_FALSE, num_ref_adx_[0_default_active_minus] = 0, num_ref_adx_[0_defaul
S249 pps->zid = pps_id; The crafted file is in the attached zip: bug3.zip
jeanIf commented on Mar 29, 2021 (Contributor)
could not reproduce crash with latest master, but added safety checks. Thanks for the report
(a) jeanlf closed this as completed on Mar 29, 2021
Assignees No one assigned
abels None yet
Projects None yet

No milestone

Development

No branches or pull requests

2 participants

