oss-sec mailing list archives

By Date By Thread

List Archive Search

Disclosure: CVE-2021-3744: crypto: ccp - fix resource leaks in ccp_run_aes_gcm_cmd()

From: Marcus Meissner <meissner () suse de>
Date: Tue, 14 Sep 2021 17:29:11 +0200

Hi,

CVE-2021-3744: crypto: ccp - fix resource leaks in ccp_run_aes_gcm_cmd()

This was reported by Tencent researcher <minihanshen () tencent com> to linux-distros, with disclosure date agreed to September 6th.

It was not followed up by timely disclosure so far, also everyone in the thread went silent for unknown reasons, even with 3 seperate reminders to publish.

As its now 1 week after the proposed embargoe end and has also expired the maximum 14 days embargo timeline, the linux-distros team publishes it to oss-security.

I am quoting the original report email, and the bugfix email from Dan Carpenter for Linux security.

Ciao, Marcus

Hello,

We found a vulnerability similar with CVE-2019-18808(https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-18808) which could allows attackers to cause a denial of service (memory consumption). Next is our analysis.

The vulnerability also appared on ccp_run_aes_gcm_cmd() funciton in driver in the Linux kernel through 5.14.

```
// CODE-1
ccp_run_aes_gcm_cmd(struct ccp_cmd_queue *cmd_q, struct ccp_cmd *cmd)
{
  struct ccp_aes_engine *aes = &cmd->u.aes;
  struct ccp_dm_workarea key, ctx, final_wa, tag;
  struct ccp_data src, dst;
  struct ccp_data aad;
  struct ccp_op op;
  unsigned int dm_offset;
  unsigned int authsize;
  unsigned int jobid;
  unsigned int ilen;
  bool in_place = true; /* Default value */
  __be64 *final;
  int ret;

  struct scatterlist *p_inp, sg_inp[2];
  struct scatterlist *p_tag, sg_tag[2];
  struct scatterlist *p_outp, sg_outp[2];
  struct scatterlist *p_aad;

  if (!aes->iv)
return -EINVAL;
. . . .
The structure aad,src,dst.. is defined in CODE-1
//CODE-2
....

op.init = 1;
  if (aes->aad_len > 0) {
    /* Step 1: Run a GHASH over the Additional Authenticated Data */
    ret = ccp_init_data(&aad, cmd_q, p_aad, aes->aad_len,
            AES_BLOCK_SIZE,
            DMA_TO_DEVICE); // init `aad`
    if (ret)
      goto e_ctx;

    op.u.aes.mode = CCP_AES_MODE_GHASH;
    op.u.aes.action = CCP_AES_GHASHAAD;

    while (aad.sg_wa.bytes_left) {
      ccp_prepare_data(&aad, NULL, &op, AES_BLOCK_SIZE, true);

      ret = cmd_q->ccp->vdata->perform->aes(&op);
      if (ret) {
        cmd->engine_error = cmd_q->cmd_error;
        goto e_aad;
      }

      ccp_process_data(&aad, NULL, &op);
      op.init = 0;
    }
  }
// CODE-3
  op.u.aes.mode = CCP_AES_MODE_GCTR;
  op.u.aes.action = aes->action;

  if (ilen > 0) {
    /* Step 2: Run a GCTR over the plaintext */
    in_place = (sg_virt(p_inp) == sg_virt(p_outp)) ? true : false;

    ret = ccp_init_data(&src, cmd_q, p_inp, ilen,
            AES_BLOCK_SIZE,
            in_place ? DMA_BIDIRECTIONAL
               : DMA_TO_DEVICE);
    if (ret)
      goto e_ctx; // whill free the value and return.

    if (in_place) {
      dst = src;
    } else {
      ret = ccp_init_data(&dst, cmd_q, p_outp, ilen,
              AES_BLOCK_SIZE, DMA_FROM_DEVICE);
      if (ret)
        goto e_src;
}
. . . . .
In CODE-2 `aad` will init which will be alloc a memory and then into CODE-3 if `src` init failed it will got
`e_ctx`(following code show it) which not free `aad` until the function execute end.
....

e_tag:
  ccp_dm_free(&final_wa);

e_dst:
  if (ilen > 0 && !in_place)
    ccp_free_data(&dst, cmd_q);
e_src:
  if (ilen > 0)
    ccp_free_data(&src, cmd_q);

e_aad:
  if (aes->aad_len)
    ccp_free_data(&aad, cmd_q);

e_ctx:
  ccp_dm_free(&ctx);

e_key:
  ccp_dm_free(&key);

  return ret;
}
```

And then this code is used to support AMD's cryptographic co-processor.

The above is our analysis, I look forward to hearing from you soon

Have a nice day
Best wishes

Peanuts
Tencent Security XuanwuLab

From: Dan Carpenter <dan.carpenter () oracle com>
Subject: [vs-plain] [PATCH RESEND] crypto: ccp - fix resource leaks in ccp_run_aes_gcm_cmd()

There are three bugs in this code:

1) If we ccp_init_data() fails for &src then we need to free aad.
   Use goto e_aad instead of goto e_ctx.
2) The label to free the &final_wa was named incorrectly as "e_tag" but
   it should have been "e_final_wa". One error path leaked &final_wa.
3) The &tag was leaked on one error path. In that case, I added a free
   before the goto because the resource was local to that block.

Fixes: 36cf515b9bbe ("crypto: ccp - Enable support for AES GCM on v5 CCPs")
Reported-by: "minihanshen(沈明航)" <minihanshen () tencent com>
Signed-off-by: Dan Carpenter <dan.carpenter () oracle com>
Reviewed-by: John Allen <john.allen () amd com>
Tested-by: John Allen <john.allen () amd com>

Resending because I screwed up the CC list and left off linux-distros.
Sorry!

```
 drivers/crypto/ccp/ccp-ops.c | 14 ++++++++------
 1 file changed, 8 insertions(+), 6 deletions(-)

diff --git a/drivers/crypto/ccp/ccp-ops.c b/drivers/crypto/ccp/ccp-ops.c
index bb88198c874e..aa4e1a500691 100644
--- a/drivers/crypto/ccp/ccp-ops.c
+++ b/drivers/crypto/ccp/ccp-ops.c
@@ -778,7 +778,7 @@ ccp_run_aes_gcm_cmd(struct ccp_cmd_queue *cmd_q, struct ccp_cmd *cmd)
                                    in_place ? DMA_BIDIRECTIONAL
                                             : DMA_TO_DEVICE);
                if (ret)
-                       goto e_ctx;
+                       goto e_aad;

                if (in_place) {
                        dst = src;
@@ -863,7 +863,7 @@ ccp_run_aes_gcm_cmd(struct ccp_cmd_queue *cmd_q, struct ccp_cmd *cmd)
        op.u.aes.size = 0;
        ret = cmd_q->ccp->vdata->perform->aes(&op);
        if (ret)
-               goto e_dst;
+               goto e_final_wa;

        if (aes->action == CCP_AES_ACTION_ENCRYPT) {
                /* Put the ciphered tag after the ciphertext. */
@@ -873,17 +873,19 @@ ccp_run_aes_gcm_cmd(struct ccp_cmd_queue *cmd_q, struct ccp_cmd *cmd)
                ret = ccp_init_dm_workarea(&tag, cmd_q, authsize,
                                           DMA_BIDIRECTIONAL);
                if (ret)
-                       goto e_tag;
+                       goto e_final_wa;
                ret = ccp_set_dm_area(&tag, 0, p_tag, 0, authsize);
-               if (ret)
-                       goto e_tag;
+               if (ret) {
+                       ccp_dm_free(&tag);
+                       goto e_final_wa;
+               }

                ret = crypto_memneq(tag.address, final_wa.address,
                                    authsize) ? -EBADMSG : 0;
                ccp_dm_free(&tag);
        }

-e_tag:
+e_final_wa:
        ccp_dm_free(&final_wa);

 e_dst:
--
2.20.1
```

By Date By Thread

Current thread:

- **Disclosure: CVE-2021-3744: crypto: ccp - fix resource leaks in ccp_run_aes_gcm_cmd()** ***Marcus Meissner (Sep 14)***

Site Search

Nmap Security Scanner
Ref Guide
Install Guide
Docs
Download
Nmap OEM

Npcap packet capture
User's Guide
API docs
Download
Npcap OEM

Security Lists
Nmap Announce
Nmap Dev
Full Disclosure
Open Source Security
BreachExchange

Security Tools
Vuln scanners
Password audit
Web scanners
Wireless
Exploitation

About
About/Contact
Privacy
Advertising
Nmap Public Source License