Talos Vulnerability Report

TALOS-2020-1018

GStreamer gst-rtsp-server GstRTSPAuth Denial of Service Vulnerability

MARCH 23, 2020

CVE NUMBER

CVE-2020-6095

Summary

An exploitable denial of service vulnerability exists in the GstRTSPAuth functionality of GStreamer/gst-rtsp-server 1.14.5. A specially crafted RTSP setup request can cause a null pointer deference resulting in denial-of-service. An attacker can send a malicious packet to trigger this vulnerability.

Tested Versions

GStreamer gst-rtsp-server 1.14.5

Product URLs

https://github.com/GStreamer/gst-rtsp-server

CVSSv3 Score

7.5 - CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

CWE

CWE-690 - Unchecked Return Value to NULL Pointer Dereference

Details

gst-rtsp-server is an open source library on top of GStreamer for building a RTSP server. RTSPATT (https://github.com/Ullaakut/RTSPAllTheThings) is one of implementation for RTSP server using GStreamer/gst-rtsp-server technology.

The GstRTSPAuth object of GStreamer/gst-rtsp-server library is found to be vulnerable to an invalid pointer dereference attack. An attacker can send a crafted RTSP setup request (using Basic Authentication) with excessively long Authorization header (> 4000 chars). The attack could cause RTSPATT server to reach dereferencing a null pointer in the code path, which causes the server to crash with Segmentation Fault.

Below is an offending RTSP Setup message seen in the debug log which triggered the segmentation fault,

```
0:01:40.248001266  2509 0x555555815de0 ERROR rtspclient rtsp-client.c:2910:handle_describe_request: client 0x5555558af630: no media
0:01:40.248009380  2509 0x555555815de0 DEBUG  default gstrtspconnection.c:3734:gst_rtsp_watch_set_send_backlog: set backlog to bytes 0,
messages 100
RTSP request message 0x5555558b5a88
 request line:
   method: 'SETUP'
   uri:    'rtsp://192.168.88.146:8554/live.sdp/'
   version: '1.0'
 headers:
   key: 'CSeq', value: '3'
   key: 'Transport', value: ';unicast;client_port=49152-49153'
   key: 'User-Agent', value: 'Synopsys Test Tool'
   key: 'UUData', value: 'Service=SynopsysSvc;AppReq=View;AssetID=0|1930;ContextId=1024;FolderId=1;'
   key: 'Authorization', value: 'BasicBasicBasicBasicBasicBasicBasicBasicBasicBasic
   <truncated, it's 4041 long here>
   BasicBasicBasicBasicBasicBasicBasicBasicB'
 body:
0:01:40.276086026  2509 0x555555815de0 INFO             rtspclient rtsp-client.c:3468:handle_request: client 0x5555558af630: received a
request SETUP rtsp://192.168.88.146:8554/live.sdp/ 1.0
0:01:40.276106537  2509 0x555555815de0 DEBUG               rtspauth rtsp-auth.c:1080:gst_rtsp_auth_check:<GstRTSPAuth@0x5555557a0460> check
authorization 'auth.check.url'
0:01:40.276116900  2509 0x555555815de0 DEBUG                default gstrtspconnection.c:3734:gst_rtsp_watch_set_send_backlog: set backlog to
bytes 0, messages 0
0:01:40.276125532  2509 0x555555815de0 LOG          rtspmountpoints rtsp-mount-points.c:251:gst_rtsp_mount_points_match: Looking for mount
point path /live.sdp
0:01:40.276133806  2509 0x555555815de0 DEBUG                default rtsp-mount-points.c:79:data_item_dump: inspect: /live.sdp 0x5555557a25f0
0:01:40.276140836  2509 0x555555815de0 DEBUG                default rtsp-mount-points.c:79:data_item_dump: prefix: /live.sdp 0x5555557a25f0
0:01:40.276147116  2509 0x555555815de0 DEBUG                default rtsp-mount-points.c:79:data_item_dump: result: /live.sdp 0x5555557a25f0
0:01:40.276153786  2509 0x555555815de0 INFO         rtspmountpoints rtsp-mount-points.c:300:gst_rtsp_mount_points_match: found media factory
0x5555557a25f0 for path /live.sdp
0:01:40.276159982  2509 0x555555815de0 DEBUG               rtspauth rtsp-auth.c:1080:gst_rtsp_auth_check:<GstRTSPAuth@0x5555557a0460> check
authorization 'auth.check.media.factory.access'
```

The vulnerable code is present in the following the following function (https://github.com/GStreamer/gst-rtsp-server/blob/1.14.5/gst/rtsp-server/rtsp-auth.c#L766):

```
758 /* parse type */
759    credential = credentials;
760    while (*credential) {
761      if ((*credential)->scheme == GST_RTSP_AUTH_BASIC) {
762        GstRTSPToken *token;
763
764        GST_DEBUG_OBJECT (auth, "check Basic auth");
765        g_mutex_lock (&priv->lock);
766        if ((token =
767                g_hash_table_lookup (priv->basic,
768                    (*credential)->authorization))) {
769          GST_DEBUG_OBJECT (auth, "setting token %p", token);
770          ctx->token = token;
771          g_mutex_unlock (&priv->lock);
772          break;
773        }
774        g_mutex_unlock (&priv->lock);
775      } else if ((*credential)->scheme == GST_RTSP_AUTH_DIGEST) {
776        if (default_digest_auth (auth, ctx, (*credential)->params))
777          break;
778      }
```

During the debug session, two GstRTSPAuthCredential object's values were checked within the function default_authenticate at rtsp-auth.c:766,

That credential->authorization value was found to be a 'null' string as the rtps parsed out the overloading Authorization header from the message.

```
764     in rtsp-auth.c
(gdb) info locals
_g_boolean_var_ = <optimized out>
token = <optimized out>
priv = 0x5555557a0410
credentials = 0x7fffec05c6f0
credential = 0x7fffec05c6f0
__func__ = "default_authenticate"
(gdb) print *credential
$5 = (GstRTSPAuthCredential *) 0x7fffec0061d0
(gdb) print **credential
$6 = {scheme = GST_RTSP_AUTH_BASIC, params = 0x0, authorization = 0x0}
(gdb) print **credentials
$7 = {scheme = GST_RTSP_AUTH_BASIC, params = 0x0, authorization = 0x0}
```

The rtsp-auth module fails to check the validity of the* credential->authorization value, instead it passes down "authorization = 0x0" as key to look for in the `priv->basic` hash table using the `g_hash_table_lookup` function.

This g_hash_table_lookup function then gets the hash value of the key using the hash_function that was chosen when creating the hash table.

This hash function was selected at line 188 (https://github.com/GStreamer/gst-rtsp-server/blob/1.14.5/gst/rtsp-server/rtsp-auth.c#L188):

priv->basic = g_hash_table_new_full (g_str_hash, g_str_equal, g_free, (GDestroyNotify) gst_rtsp_token_unref);

So the `g_hash_table_lookup function` calls the `g_str_hash` function to get the hash value of the key, which causes a NULL pointer dereference because the `g_str_hash` function requires a "not nullable" value as its argument. This can be seen in the following code (https://gitlab.gnome.org/GNOME/glib/blob/2.56.4/glib/ghash.c):

```
/**
 * g_str_hash:
 * @v: (not nullable): a string key
 *
 * Converts a string to a hash value.
 *
 * This function implements the widely used "djb" hash apparently
 * posted by Daniel Bernstein to comp.lang.c some time ago.  The 32
 * bit unsigned hash value starts at 5381 and for each byte 'c' in
 * the string, is updated: `hash = hash * 33 + c`. This function
 * uses the signed value of each byte.
 *
 * It can be passed to g_hash_table_new() as the @hash_func parameter,
 * when using non-%NULL strings as keys in a #GHashTable.
 *
 * Note that this function may not be a perfect fit for all use cases.
 * For example, it produces some hash collisions with strings as short
 * as 2.
 *
 * Returns: a hash value corresponding to the key
 */
  guint
  g_str_hash (gconstpointer v)
  {
    const signed char *p;
    guint32 h = 5381;

    for (p = v; *p != '\0'; p++)
      h = (h << 5) + h + *p;

    return h;
  }
```

Crash Information

Below is a backtrace when RTSPATT crashes,

```
> backtrace
Thread 2 "pool" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7ffff3b64700 (LWP 53126)]
g_str_hash (v=0x0) at ../../../../glib/ghash.c:1894
1894    ../../../../glib/ghash.c: No such file or directory.
#0  0x00007ffff756a580 in g_str_hash (v=0x0) at ../../../../glib/ghash.c:1894
#1  0x00007ffff75699d4 in g_hash_table_lookup_node (hash_return=<synthetic pointer>, key=0x0, hash_table=0x555555898c60) at
../../../../glib/ghash.c:379
#2  0x00007ffff75699d4 in g_hash_table_lookup (hash_table=0x555555898c60, key=0x0) at ../../../../glib/ghash.c:1153
#3  0x00007ffff72ea43e in default_authenticate (auth=0x55555579eca0 [GstRTSPAuth], ctx=0x7ffff3b63a20) at rtsp-auth.c:766
#4  0x00007ffff72ea27d in ensure_authenticated (auth=auth@entry=0x55555579eca0 [GstRTSPAuth], ctx=ctx@entry=0x7ffff3b63a20) at rtsp-
auth.c:872
#5  0x00007ffff72eadab in check_factory (check=0x7ffff7316228 "auth.check.media.factory.access", ctx=0x7ffff3b63a20, auth=0x55555579eca0
[GstRTSPAuth]) at rtsp-auth.c:959
#6  0x00007ffff72eadab in default_check (auth=0x55555579eca0 [GstRTSPAuth], ctx=0x7ffff3b63a20, check=0x7ffff7316228
"auth.check.media.factory.access") at rtsp-auth.c:1033
#7  0x00007ffff730b83a in find_media (client=client@entry=0x5555558a80e0 [GstRTSPClient], ctx=ctx@entry=0x7ffff3b63a20,
path=path@entry=0x7fffec052d10 "/live.sdp", matched=0x7ffff3b638a0) at rtsp-client.c:956
#8  0x00007ffff730c3c3 in handle_setup_request (client=client@entry=0x5555558a80e0 [GstRTSPClient], ctx=ctx@entry=0x7ffff3b63a20) at rtsp-
client.c:2391
#9  0x00007ffff730e80d in handle_request (client=client@entry=0x5555558a80e0 [GstRTSPClient], request=request@entry=0x5555558aa418) at rtsp-
client.c:3592
#10 0x00007ffff731190b in gst_rtsp_client_handle_message (client=0x5555558a80e0 [GstRTSPClient], message=0x5555558aa418) at rtsp-
client.c:4223
#11 0x00007ffff4e36186 in  () at /usr/lib/x86_64-linux-gnu/libgstrtsp-1.0.so.0
#12 0x00007ffff757b285 in g_main_dispatch (context=0x5555558a90d0) at ../../../../glib/gmain.c:3176
#13 0x00007ffff757b285 in g_main_context_dispatch (context=context@entry=0x5555558a90d0) at ../../../../glib/gmain.c:3829
#14 0x00007ffff757b650 in g_main_context_iterate (context=0x5555558a90d0, block=block@entry=1, dispatch=dispatch@entry=1, self=<optimized
out>) at ../../../../glib/gmain.c:3902
#15 0x00007ffff757b962 in g_main_loop_run (loop=0x5555558a9190) at ../../../../glib/gmain.c:4098
#16 0x00007ffff72ee48f in do_loop (thread=0x5555558a4800) at rtsp-thread-pool.c:331
#17 0x00007ffff75a3b60 in g_thread_pool_thread_proxy (data=<optimized out>) at ../../../../glib/gthreadpool.c:307
#18 0x00007ffff75a3195 in g_thread_proxy (data=0x555555810630) at ../../../../glib/gthread.c:784
#19 0x00007ffff5b716db in start_thread (arg=0x7ffff3b64700) at pthread_create.c:463
#20 0x00007ffff685888f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
#0  0x00007ffff756a580 in g_str_hash (v=0x0) at ../../../../glib/ghash.c:1894
    p = 0x0
    h = 5381
#1  0x00007ffff75699d4 in g_hash_table_lookup_node (hash_return=<synthetic pointer>, key=0x0, hash_table=0x555555898c60) at
../../../../glib/ghash.c:379
    node_hash = <optimized out>
    hash_value = <optimized out>
    have_tombstone = 0
    step = 0
    node_index = <optimized out>
    first_tombstone = 0
    node_hash = <optimized out>
    __func__ = "g_hash_table_lookup"
#2  0x00007ffff75699d4 in g_hash_table_lookup (hash_table=0x555555898c60, key=0x0) at ../../../../glib/ghash.c:1153
    node_hash = <optimized out>
    __func__ = "g_hash_table_lookup"
#3  0x00007ffff72ea43e in default_authenticate (auth=0x55555579eca0 [GstRTSPAuth], ctx=0x7ffff3b63a20) at rtsp-auth.c:766
    token = <optimized out>
    priv = 0x55555579ec50
    credentials = 0x7fffdc00c330
    credential = 0x7fffdc00c330
    __func__ = "default_authenticate"
#4  0x00007ffff72ea27d in ensure_authenticated (auth=auth@entry=0x55555579eca0 [GstRTSPAuth], ctx=ctx@entry=0x7ffff3b63a20) at rtsp-
auth.c:872
    klass = <optimized out>
    __func__ = "ensure_authenticated"
```

Timeline

2020-03-20 - Vendor disclosure

2020-03-23 - Public Release

CREDIT

Discovered by Peter Wang of Cisco ASIG
