

# Async API in OpenSIPS

2 February 2020

- Peter Lemenkov
- <https://github.com/lemenkov> (<https://github.com/lemenkov>)
- <https://github.com/lemenkov/brussels-fosdem-2020> (<https://github.com/lemenkov/brussels-fosdem-2020>)

# Introduction

- Introduced in OpenSIPS 2.1 (<https://www.opensips.org/Documentation/Script-Async-2-1>)
- Not widely used
- Not really well documented
- I'll talk about ver. 3.0 (<https://www.opensips.org/Documentation/Script-Async-3-0>)
- More at OpenSIPS Summit 2020

## async in c code

```
static pv_elem_t *extra_id_pv = NULL;

static acmd_export_t acmds[] = {
    {"rtpengine_offer", (acmd_function)rtpengine_offer_af, {
        {CMD_PARAM_STR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_VAR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_VAR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_STR | CMD_PARAM_OPT, 0, 0}, {0,0,0}}},
    {"rtpengine_answer", (acmd_function)rtpengine_answer_af, {
        {CMD_PARAM_STR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_VAR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_VAR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_STR | CMD_PARAM_OPT, 0, 0}, {0,0,0}}},
    {"rtpengine_delete", (acmd_function)rtpengine_delete_af, {
        {CMD_PARAM_STR | CMD_PARAM_OPT, 0, 0},
        {CMD_PARAM_VAR | CMD_PARAM_OPT, 0, 0}, {0,0,0}}},
    {0, 0, {{0, 0, 0}}}
};
```

## async in c code (cont.)

```
}  
  
static int rtpe_function_call_async(struct sip_msg *msg, async_ctx *ctx, str *flags_str,  
    pv_spec_t *spvar, pv_spec_t *bpvar, str *body, enum rtpe_operation op)  
{  
    struct ng_flags_parse ng_flags;  
    str oldbody;  
    struct rtpe_node *node;  
    struct rtpe_set *set;  
    int ret, read_fd;  
    --- 63 строки: rtpe_async_param *param;-----  
  
    param->bencbuf = bencbuf;  
    param->op = op;  
    param->node = node;  
    param->cookie = strdup(cookie);  
    param->bpvar = bpvar;  
    param->spvar = spvar;  
  
    ctx->resume_f = resume_async_send_rtpe_command;  
    ctx->resume_param = param;  
  
    /* async started with success */  
    async_status = read_fd;  
    return 1;  
  
error:  
    bencode_buffer_free(bencbuf);  
    return -1;  
}
```

## async in c code (cont. next one)

```
enum async_ret_code resume_async_send_rtpe_command(int fd, struct sip_msg *msg, void *_param)
{
+-- 12 строк: int len = 0, cookie_len = 0;-----

    rtpe_async_param *param = (rtpe_async_param *)_param;

    if (param->node->rn_umode == 0) {
        do {
            len = read(fd, buf, sizeof(buf) - 1);
        } while (len == -1 && errno == EINTR);
        close(fd);
        if (len <= 0) {
            LM_ERR("can't read reply from a RTP proxy\n");
            goto error;
        }
    }
    else {
        fds[0].fd = fd;
        fds[0].events = POLLIN;
        fds[0].revents = 0;
        while ((poll(fds, 1, rtpeengine_tout * 1000) == 1) && (fds[0].revents & POLLIN) != 0)
            do {
                len = recv(fd, buf, sizeof(buf)-1, 0);
            } while (len == -1 && errno == EINTR);
            if (len <= 0) {
                LM_ERR("can't read reply from a RTP proxy\n");
                RTPE_IO_ERROR_CLOSE(param->node->idx);
                goto error;
            }
        cookie_len = strlen(param->cookie);
    }
}
```

# async in config

```
        $acc_extra(rtpstat) = $rtpquery;
    }

    xlog("L_INFO", "$ci ($cfg_file) RTP UNFORCE - ASYNC STARTING $ci");
    async(rtpengine_delete(), rtpengine_post_delete);
    break;
}

xlog("L_INFO", "$ci ($cfg_file) RTP Proxy Reply [$avp(rtp_proxy_engaged)] (Res: $var(rtpengine_res) )");

if ($var(rtpengine_res) == -1) {
    xlog("L_ERROR", "$ci ($cfg_file) RTP Response was -1 clearing rtp_proxy_engaged");
    $avp(rtp_proxy_engaged) := 0;
}

return($var(rtpengine_res)); # Return retcode from rtpproxy_offer
}

route[rtpengine_post_delete]{
    xlog("L_INFO", "$ci ($cfg_file) RTP UNFORCE - ASYNC COMPLETE 1 $ci");
    xlog("L_INFO", "$ci ($cfg_file) RTP Server IP: $avp(rtp_server_ip)");
    if (isbflagset("SRC_NAT_BFLAG") && (($rm=="BYE") || ($rm=="CANCEL"))) fix_nated_contact();
    xlog("L_INFO", "$ci ($cfg_file) RTP UNFORCE - ASYNC COMPLETE 2 $ci");
    #if (isbflagset("SRC_NAT_BFLAG") && $rm) fix_nated_contact();
    #if (isbflagset("SRC_NAT_BFLAG")) fix_nated_contact();
    if (!t_relay()) {
        xlog("L_ERROR", "$ci ($cfg_file) Unable to r_relay");
        if ($avp(branch_no_advance)){ # WebRTC Registrations
            xlog("L_ERROR", "$ci ($cfg_file) Sleeping for 3 seconds");
            async( sleep(3), egress);
        } else {
            send_reply(500,"Internal Error");
        }
    }
};
xlog("L_INFO", "$ci ($cfg_file) RTP UNFORCE - ASYNC COMPLETE 3 $ci");
}
```

## async in config (cont.)

```
xlog("L_INFO", "REMOVE HDRS 3 $ci");

# RTP Responses can sometime stop the processing (bug) of the rest to the script route, keep at the end.
if (is_method("BYE")){
    route(set_bye_acc_fields);
    xlog("L_INFO", "$ci ($cfg_file) BYE MESSAGE IN REPLY");
    route(rtpproxy, "unforce");
    if (is_present_hf("Reason")) insert_hf("Reason: SIP ;cause=Timeout ;text=\"$DLG_end_reason\"\r\n", "
    xlog("L_INFO", "$ci ($cfg_file) $DLG_end_reason");
}

if (has_body("application/sdp")){
    if (isflagset("LATE_CODEC")){
        xlog("L_INFO", "RTP OFFER $ci");
        route(rtpproxy, "offer");
    } else {
        route(rtpproxy, "update_answer");
        drop();
        exit;
    }
    if ($retcode == -1 && $avp(rtp_mode) == 'strict'){
        xlog("L_WARN", "$ci ($cfg_file) RTP Engagement Error (Late Codec Negotiation)");
        $acc_extra(x_sip_reason) = "RTP Engagement Error (R)";
        drop();
        exit;
    }
}
}
```

# Issues

## [CRASH] called on a not free-able lump when using async API #1948

Edit

New issue

Open

lemenkov opened this issue 8 days ago · 4 comments



lemenkov commented 8 days ago · edited ▼

Contributor + @ ...

Hello,

I'm still working on my module which is using async API and got another one crash. Unfortunately I could not create a small config reproducer so all I have is this backtrace (which looks very similar to #1683).

```
#0 __GI_raise (sig=6) at ../sysdeps/unix/sysv/linux/raise.c:51
#1 0x00007fd78c8cf801 in __GI_abort () at abort.c:79
#2 0x0000559720552be6 in free_lump (lump=0x7fd78a08aa80) at data_lump.c:441
#3 0x0000559720552ceb in free_lump_list (l=0x7fd78a08aa48) at data_lump.c:470
#4 0x00005597205531a8 in del_flagged_lumps (lump_list=0x7fd78946ecb8 <faked_req+2040>, f
#5 0x00007fd78924cbf8 in post_print_uac_request (request=0x7fd78946e4c0 <faked_req>, or
#6 0x00007fd78924d617 in add_uac (t=0x7fd78a100350, request=0x7fd78946e4c0 <faked_req>,
#7 0x00007fd78924e502 in t_forward_nonack (t=0x7fd78a100350, p_msg=0x7fd78946e4c0 <fake
#8 0x00007fd7892305f0 in w_t_relay (p_msg=0x7fd78946e4c0 <faked_req>, flags=0x0, proxy=
#9 0x000055972056cfc0 in do_action (a=0x7fd78ba1f6e0, msg=0x7fd78946e4c0 <faked_req>) a
#10 0x000055972056a2b8 in run_action_list (a=0x7fd78ba1f6e0, msg=0x7fd78946e4c0 <faked_r
#11 0x00005597205b16b5 in eval_elem (e=0x7fd78ba1f838, msg=0x7fd78946e4c0 <faked_req>, v
#12 0x00005597205b2896 in eval_expr (e=0x7fd78ba1f838, msg=0x7fd78946e4c0 <faked_req>, v
#13 0x00005597205b2990 in eval_expr (e=0x7fd78ba1f8d0, msg=0x7fd78946e4c0 <faked_req>, v
#14 0x00005597205b29ca in eval_expr (e=0x7fd78ba1f968, msg=0x7fd78946e4c0 <faked_req>, v
#15 0x00005597205b6f7e in do_action (a=0x7fd78ba20be8, msg=0x7fd78946e4c0 <faked_req>) a
#16 0x000055972056a2b8 in run_action_list (a=0x7fd78ba1dd68, msg=0x7fd78946e4c0 <faked_r
#17 0x000055972056a172 in run_actions (a=0x7fd78ba1dd68, msg=0x7fd78946e4c0 <faked_req>)
#18 0x000055972056bed1 in do_action (a=0x7fd78ba1bf28, msg=0x7fd78946e4c0 <faked_req>) a
#19 0x000055972056a2b8 in run_action_list (a=0x7fd78ba1bab8, msg=0x7fd78946e4c0 <faked_r
#20 0x000055972056c0e4 in do_action (a=0x7fd78ba1c808, msg=0x7fd78946e4c0 <faked_req>) a
#21 0x000055972056a2b8 in run_action_list (a=0x7fd78ba1b358, msg=0x7fd78946e4c0 <faked_r
#22 0x000055972056cc66 in do_action (a=0x7fd78ba1cc10, msg=0x7fd78946e4c0 <faked_req>) a
#23 0x000055972056a2b8 in run_action_list (a=0x7fd78ba00958, msg=0x7fd78946e4c0 <faked_r
#24 0x000055972056a172 in run_actions (a=0x7fd78ba00958, msg=0x7fd78946e4c0 <faked_req>)
#25 0x000055972056be7c in do_action (a=0x7fd78b99efb0, msg=0x7fd78946e4c0 <faked_req>) a
#26 0x000055972056a2b8 in run_action_list (a=0x7fd78b99eeb48, msg=0x7fd78946e4c0 <faked_r
#27 0x000055972056a172 in run_actions (a=0x7fd78b99eeb48, msg=0x7fd78946e4c0 <faked_req>)
#28 0x000055972056a44f in run_top_route (a=0x7fd78b99eeb48, msg=0x7fd78946e4c0 <faked_req
#29 0x00007fd78920977f in run_failure_handlers (t=0x7fd78a100350) at t_reply.c:604
#30 0x00007fd78920a99b in t_should_relay_response (Trans=0x7fd78a100350, new_code=400, b
#31 0x00007fd78920b2eb in relay_reply (t=0x7fd78a100350, p_msg=0xffffffffffffff, branci
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants





# Issues (cont.)

The screenshot shows a GitHub issue page for the repository `OpenSIPS / opensips`. The issue title is `[CRASH] msg_translator.c: len mismatch : calculated 1096, written 1397 #1939`. It has 142 issues, 14 pull requests, and 0 projects. The issue was opened by `lemenkov` 18 days ago and has 7 comments. The issue is labeled `stale` and has no assignees or milestones. The comments section shows two comments from `lemenkov` and `razvancrainea`. The first comment from `lemenkov` describes a crash and includes a link to a core dump. The second comment from `razvancrainea` asks for the OS being used. The right sidebar shows the issue's metadata, including labels, projects, milestones, and notifications.

Search or jump to... Pull requests Issues Marketplace Explore

OpenSIPS / opensips Unwatch 90 Unstar 676 Fork 354

<> Code Issues 142 Pull requests 14 Actions Projects 0 Wiki Security Insights

**[CRASH] msg\_translator.c: len mismatch : calculated 1096, written 1397 #1939** Edit New Issue

lemenkov opened this issue 18 days ago · 7 comments

**lemenkov** commented 18 days ago · edited Contributor + @ ...

Hello All,  
I've just got hit by this strange message followed by abort() call (see msg\_translator.c:2323).

I am testing a custom opensips module but I still cannot find a clue on how it can affect it. I am using opensips [b660a21](#) as a master for my development.

Several invites are passing OK but after a 20-40 ones it fails.

Here is a core dump:  
<https://send.firefox.com/download/3db383781e7a099a/#kOKTGDPNgTeGeQk6ES0ziQ>

Apparently lumps\_len(...) gives us a different size than one calculated during apply\_msg\_changes(...) function.

**razvancrainea** commented 17 days ago Member + @ ...

Could you please tell us the OS you are using?

**razvancrainea** commented 17 days ago Member + @ ...

**Assignees**  
No one assigned

**Labels**  
`stale`

**Projects**  
None yet

**Milestone**  
No milestone

**Notifications** Customize  
[Unsubscribe](#)  
You're receiving notifications because you're watching this repository.

**2 participants**

# Issues

- Sometimes very fragile
- Read logs **carefully** (I mean it)

```
opensips[16039]: 4-4689@10.6.195.85 (/etc/opensips/route.cfg) [91.183.239.28] To Tag Found
opensips[16039]: 4-4689@10.6.195.85 (/etc/opensips/sequential-requests.cfg) [<null>] Sequential Request: ACK
opensips[16039]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002Y
opensips[16039]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002Y
opensips[16039]: 4-4689@10.6.195.85 (/etc/opensips/sequential-requests.cfg) [<null>] Dialog Matched
opensips[16040]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002#010
opensips[16039]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002#007
opensips[16040]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002#010
opensips[16039]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#002#007
opensips[16040]: 1-4689@10.6.195.85 (/etc/opensips/rtpproxy.cfg) RTP UNFORCE - ASYNC COMPLETE 3 1-4689@10.6.195.85
opensips[16040]: 1-4689@10.6.195.85 (/etc/opensips/rtpproxy.cfg) RTP Proxy Reply [3] (Res: <null> )
opensips[16040]: ERROR:nathelper:fix_nated_contact_f: SCRIPT BUG - second attempt to change URI Contact
opensips[16039]: 19-4689@10.6.195.85 (/etc/opensips/route.cfg) UAC NAT Detected
opensips[16039]: 19-4689@10.6.195.85 (/etc/opensips/route.cfg) [91.183.239.28] To Tag Found
```

## Issues (cont.)

- Read logs really really carefully

```
ensips[16362]: 1-5025@10.6.195.85 (/etc/opensips/rtpproxy.cfg) RTP OFFER - ASYNC COMPLETE 4 1-5025@10.6.195.85
ensips[16362]: 1-5025@10.6.195.85 (/etc/opensips/rtpproxy.cfg) NON-ASYNC DATA RTP Server IP: udp:178.62.29.244:122
ensips[16362]: 1-5025@10.6.195.85 (/etc/opensips/rtpproxy.cfg) RTP Proxy Reply [3] (Res: <null> )
ensips[16362]: 1-5025@10.6.195.85 (/etc/opensips/per-branch-ops.cfg) RTP Engagement Error
ensips[16362]: ERROR:tm:t_forward_nonack: failure to add branches
ensips[16362]: ERROR:tm:w_t_relay: t_forward_nonack failed
ensips[16362]: 1-5025@10.6.195.85 (/etc/opensips/relay.cfg) Unable to r_relay
ensips[16362]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#001f
ensips[16362]: ERROR:topology_hiding:topo_callid_post_raw: could not parse resulted sip message: HEP3#001f
ensips[16363]: 1-5025@10.6.195.85 (/etc/opensips/route.cfg) UAC NAT Detected
ensips[16363]: 1-5025@10.6.195.85 (/etc/opensips/route.cfg) [91.183.239.28] To Tag Found
```

## Conclusion

- Go and use it!
- Very careful testing required

# Thank you

- Peter Lemenkov
- <https://github.com/lemenkov> (<https://github.com/lemenkov>)
- <https://github.com/lemenkov/brussels-fosdem-2020> (<https://github.com/lemenkov/brussels-fosdem-2020>)