Issue narrowed down a bit

Thursday, November 8, 2018 2:01 PM

I have narrowed down the scope of this crash to the following (I don't believe this is a heavy use of the AMI in manipulating channels):

User Scenario (In a nut shell)

I use most of the queue and agent functionality as intended, as I believe others would. The differences are that instead of using channel features to transfer and connect (i.e. *2 to transfer, * to hang up and connect the caller and 3rd party), I am using ATXFER and BRIDGEKICK via the AMI.

User Scenario (Details)

An agent logs in with this context:

[proxy-incoming] -- login in the agent -exten => _[127]XXX,1,NoOp() same => n,Set(CHANNEL(dtmf_features)=TH) same => n,Set(CHANNEL(musicclass)=silence) same => n,Answer() same => n,NoOp(\${CALLERID(num)}) same => n,NoOp(caller id name is \${CALLERID(name)}) same => n,Set(sipaccount=\${GET_CHECK_AMP_LOGIN(\${EXTEN}))}) same => n,NoOp(sip account is \${sipaccount}) same => n,Gotoif(\$[\${LEN(\${sipaccount})}>0]?:emptysipaccount) same => n,Gotolf(\$[\${sipaccount}=\${CALLERID(name)}]?:invalidsipaccount) same => n,Set(CDR_PROP(disable)=1) same => n,AgentLogin(\${EXTEN}) same => n,Hangup() same => n(invalidsipaccount),NoOp(Sip Account is Invalid) same => n,Playback(im-sorry&tt-somethingwrong&vm-and&is-in-use&goodbye) same => n,Hangup() same => n(emptysipaccount),NoOp(Sip Account is Empty) same => n,Playback(im-sorry&tt-somethingwrong&goodbye) same => n,Hangup()

They receive a call and need to transfer to someone. The application calls the Asterisk Action ATXFER with the agents channel ID.

action: atxfer\r\nchannel: PJSIP/hq-ast-v06-user-0000006\r\nexten:123456789888888888 \r\ncontext: atxfer\r\nactionid: NA[ARG4]\r\nr\n"

The agent channel subsequently gets routed to this context

[atxfer]

- exten => _NXXNXXXXXXXXXXXXX,1,NoCDR()
- exten => _NXXNXXXXXXXXXXXXXXX,n,Set(CALLERID(num)=\${EXTEN:10})
 exten => NXXNXXXXXXXXXXXXXX,n,NoOp(\${CALLERID(num)})
- exten => NXXNXXXXXXXXXXXXXXX,n,Set(MONITOR EXEC=soxmix)
- exten => NXXNXXXXXXXXXXXXXX,n,Set(BEEPID=\${PERIODIC HOOK(hooks,99,\${BEEPFREQ}})})
- exten =>_NXXNXXXXXXXXXXXXXXX,n,Dial(PJSIP/\${EXTEN:0:10}@\${CALLROUTER},,,tT)

exten => _NXXNXXXXXXXXXXXXXXX,n,Hangup()

The Dial() command connects to the 3rd party and the agent talks to the 3rd party. When the agent wants to connect the 3rd party to the original caller, the application will send a BRIDGEKICK action on the agent channel.

action: BridgeKick\r\nactionid:\r\nchannel: PJSIP/hq-ast-v06-user-00000006\r\n\r\n

After this action is sent, Asterisk connects the caller and 3rd party and attempts to put the agent back in the agent pool and then the following *segmentation fault* (in green) occurs about once every 100 (guess) successes.

Program terminated with signal 11, Segmentation fault.

#0 0x0000000004899a6 in ast_bridge_channel_establish_roles (bridge_channel=0x7f2adc09cd50) at bridge_roles.c:486

486 if (setup bridge role option(this role copy, role option->option, role option->value)) {

Missing separate debuginfos, use: debuginfo-install gsm-1.0.13-4.el6.x86_64 libedit-2.11-4.20080712cvs.1.el6.x86_64 libidn-1.18-2.el6.x86_64 libogg-1.1.4-2.1.el6.x86_64 libtool-ltdl-2.2.6-15.5.el6.x86_64 lua-5.1.4-4.1.el6.x86_64 pakchois-0.4-3.2.el6.x86_64 pot-1.13-7.el6.x86_64 speex-1.2-0.12.rc1.1.el6.x86_64

- ^[[?1034h(gdb) bt
- #0 0x0000000004899a6 in ast_bridge_channel_establish_roles (bridge_channel=0x7f2adc09cd50) at bridge_roles.c:486
- #1 0x00000000486582 in bridge_channel_internal_push_full (bridge_channel=0x7f2adc09cd50, optimized=0) at bridge_channel.c:2146
- #2 0x000000000486905 in bridge_channel_internal_push (bridge_channel=0x7f2adc09cd50) at bridge_channel.c:2199
- #3 0x000000000487a4e in bridge_channel_internal_join (bridge_channel=0x7f2adc09cd50) at bridge_channel.c:2697
- #4 0x00000000046da4b in ast_bridge_join (bridge=0x7f2ac8004020, chan=0x7f2aa8bacdb0, swap=0x0, features=0x7f2a7054c1d0, tech_args=0x0, flags=AST_BRIDGE_JOIN_PASS_REFERENCE) at bridge.c:1713
- #5 0x00007f2a953b769b in agent_run (agent=0x357ecb0, logged=0x7f2aa8bacdb0) at app_agent_pool.c:1550
- #6 0x00007f2a953b9afd in agent_login_exec (chan=0x7f2aa8bacdb0, data=0x7f2a7054c430 "1713,s") at app_agent_pool.c:2181
- #7 0x00000000058a523 in pbx_exec (c=0x7f2aa8bacdb0, app=0x3549cb0, data=0x7f2a7054c430 "1713,s") at pbx_app.c:491
- #8 0x000000000576e79 in pbx_extension_helper (c=0x7f2aa8bacdb0, con=0x0, context=0x7f2aa8bad768 "agentrelogin", exten=0x7f2aa8bad7b8 "11", priority=2, label=0x0, callerid=0x7f2ac0bcdcc0 "8554204494", action=E_SPAWN,
- found=0x7f2a7054eb10, combined_find_spawn=1) at pbx.c:2886
- #9 0x00000000057a546 in ast_spawn_extension (c=0x7f2aa8bacdb0, context=0x7f2aa8bad768 "agentrelogin", exten=0x7f2aa8bad7b8 "11", priority=2, callerid=0x7f2ac0bcdcc0 "8554204494", found= 0x7f2a7054eb10, combined_find_spawn=1)

at pbx.c:4111

- #10 0x00000000057b352 in __ast_pbx_run (c=0x7f2aa8bacdb0, args=0x0) at pbx.c:4288
- #11 0x000000000057cd76 in pbx_thread (data=0x7f2aa8bacdb0) at pbx.c:4610
- #12 0x00000000000604421 in dummy_start (data=0x7f2aa8236da0) at utils.c:1238
- #13 0x0000003802607aa1 in start_thread (arg=0x7f2a7054f700) at pthread_create.c:301
- #14 0x00000038022e8bcd in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:115

My research and thoughts and additional information

Looking at the code, it seems that 1 or more roles do exist and that role_options for those roles do exist in the roles datastore of the channel. I am basing this on code entering the two AST_LIST_TRAVERSE code blocks in the ast_bridge_channel_establish_roles function of the bridge_roles.c module.

I have put debug in the ast_bridge_channel_establish_roles function of bridge_roles.c module as follows, in order to track down what I can. When a crash happened in this scenario, it crashed at the highlighted section below. I am assuming that Role_option->option or Role_option->value no longer are accessible.

```
AST_LIST_TRAVERSE(&roles_datastore->role_list, role, list) {
    struct bridge_role *this_role_copy;
    if (setup bridge role(bridge channel->bridge roles, role->role)) {
         /* We need to abandon the copy because we couldn't setup a role */
        ast_bridge_channel_clear_roles(bridge_channel);
        return -1;
    this_role_copy = AST_LIST_LAST(&bridge_channel->bridge_roles->role_list);
    if (!this_role_copy)
        ast_debug(5, "JEFFBAD - copy of bridge role is null or empty. Channel: %s\n", ast_channel_name(bridge_channel->chan));
    if (role)
        ast_debug(5, "JEFFINFORM - Inside outside interation of datastore. Role name: %s Channel: %s\n",role->role, ast_channel_name(bridge_channel->chan);
    else
        ast_debug(5, "JEFFBAD - Rile is null or empty. Channel: %s\n",ast_channel_name(bridge_channel->chan));
    AST_LIST_TRAVERSE(&role->options, role_option, list) {
         if (!role_option)
             ast_debug(5, "JEFFBAD - Role option is null or empty and the code following is expecting it not to be empty. Chanel: %s \n", ast_channel_name(bridge_channel->chan));
        else
             Crash happens here (Role option object exists, but the values option and value must be gone)
             ast_debug(5, "JEFFGOOD - Role option is %s Channel: %s \n", role_option->option, ast_channel_name(bridge_channel->chan));
             ast_debug(5, "JEFFGOOD - Role option value is %s Channel: %s \n", role_option->value, ast_channel_name(bridge_channel->chan));
        if (setup_bridge_role_option(this_role_copy, role_option->option, role_option->value)) {
             /* We need to abandon the copy because we couldn't setup a role option */
             ast_bridge_channel_clear_roles(bridge_channel);
             return -1;
        }
   }
}
```

I have put my own debug in the Asterisk near the point of failure but it does not log the information in the debug file right at the crash time. I suspect that it crashed before it could process the calls to the debug log file.