SSSD Errors

Updated February 14 2025 at 11:28 PM - English

TABLE OF CONTENTS

Overview

This article explains the common issues related to sssd and how to troubleshoot them. There can be multiple reasons due to which these issues can occur and hence this article covers the issues in detail and the probable solutions/workaround available to fix those issues.

Issue

- SSSD process is terminated by own WATCHDOG
- sdap_async_sys_connect request failed / sdap_async_sys_connect request failed
- krb5_auth_timeout or krb5_child_timeout reached
- TSIG error with server: tsig verify failure
- s2n_exop_request failed or ldap_extended_operation failed
- ldap_install__tls failed
- ldap_sasl_interactive_binds_failed
- GPO-based access control failed.

SSSD process is terminated by own WATCHDOG

```
(Fri Apr 14 15:07:19 2023) system1 sssd[sssd]: Child [1277] ('SSSDdomain':'%BE_SSSDdomain') was terminated by
own WATCHDOG. Consult corresponding logs to figure out the reason.
```

The above log indicates that the sssd_be process was blocked too long on something that was longer than 3*10 seconds(the default value of timeout is 10 sec). In order to find which operation is blocking sssd_be, enable sssd debugging by adding debug_level = 9 under all sections of the `/etc/sssd/sssd.conf` file, especially under the [$domain] section, and wait for the issue to reoccur. Once the issue is observed, take the timestamp of the ... was terminated by own WATCHDOG message and then spot the last operation before the timestamp in `/var/log/sssd/sssd_$domain.log` . In the above example, sssd_be is getting killed, but there could be some other situation where sssd_nss or sssd_pam processes could get killed by watchdog. In this case, the troubleshooting process will be the same, but the respective sssd log needs to be checked instead of the `/var/log/sssd/sssd_$domain.log` file. Increasing the timeout values may serve as a workaround, however, finding the root cause of the watchdog termination may be important. On a side note if the processing of group membership is very slow then also we can observe sssd_be killed by watchdog.

The same has been documented in 6155072

sdap async sys connect request failed or sdap sync sys connect request failed

```
(Fri Apr 14 16:07:19 2023) [sssd[be[example.com]]] [sssd_async_socket_init_done] (0x0020):
sdap_async_sys_connect request failed: [110]: Connection timed out.
```

sdap_async_sys_connect request failed occurs if sssd is not able to connect to the LDAP server within 6 seconds. This could be an issue with DNS or the network. Validate the DNS SRV records; if SRV records are not working, hardcoding the AD/LDAP server may help here. For example, if id_provider = ad is being used then hardcoding of AD servers can be done as: add ad_server = ad1.example.com, ad2.example.com under the `[$domain]` section of the `/etc/sssd/sssd.conf` . If the network is slow or ldap_network_timeout is reached, then consider increasing the value of ldap_network_timeout which is set to 6 seconds by default.

```
(2023-04-20 21:37:10): [be[example.com]] [generic_ext_search_handler] (0x0020): [RID#1434]
sdap_get_generic_ext_recv failed: [110]: Connection timed out [ldap_search_timeout]
(2023-04-20 21:37:16): [be[example.com]] [sssd_async_connect_done] (0x0020): [RID#1434] connect failed [110]
[Connection timed out].
(2023-04-20 21:37:16): [be[example.com]] [sssd_async_connect_done] (0x0020): [RID#1434] connect failed [110]
[Connection timed out].
(2023-04-20 21:37:16): [be[example.com]] [sssd_async_socket_init_done] (0x0020): [RID#1434]
sdap_async_sys_connect request failed: [110]: Connection timed out [ldap_network_timeout].
```

In the above example, `sdap_async_sys_connect request failed` is getting generated due to a network issue. There are two timeouts `Connection timed out [ldap_network_timeout]` and a `connection timed out [ldap_search_timeout]` . As a workaround consider increasing the value of `ldap_network_timeout` and `ldap_search_timeout` . However, it is necessary to identify the underlying network issue.

The same can be observed in the case of id_provider = ldap `sdap_get_generic_ext_recv failed: [110]: Connection timed out [ldap_search_timeout]` in such situation it’s necessary to check if the TLS handshake is successful.

krb5 auth timeout or krb5 child timeout reached

```
(Fri Apr 14 16:37:19 2023) [sssd[be[example.com]]] [krb5_child_timeout] (0x0040): Timeout for child [23514]
reached. In case KDC is distant or network is slow you may consider increasing value of krb5_auth_timeout.
```

The above error is self-explanatory when `SSSD` is trying to connect the KDC server it's getting timed out. Mostly the KDC server is responding very slowly due to some reason. One of them could be an issue with the firewall or a slow network. As a workaround, consider increasing the value of `krb5_auth_timeout` which is 6 seconds by default.
For details refer to 6155072

TSIG error with server: tsig verify failure

```
(Fri Apr 14 16:37:19 2023) example.com sssd[87108]: ; TSIG error with server: tsig verify failure
```

When `SSSD` is configured with `id_provider = ad`, by default, `SSSD` will try to update the DNS record using the `nsupdate` command. If `tsig/nsupdate` is failing, sssd will return TSIG error with server: tsig verify failure. This error can be safely ignored if Dynamic DNS update is not being used. Dynamic DNS update can be disabled using `dyndns_update = false` under the `[$domain]` section of the `/etc/sssd/sssd.conf` file. If Dynamic DNS update is being used, then in order to identify the issue, enable sssd debugging using troubleshooting sssd

s2n exop request failed / ldap extended operation failed

```
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [sdap_call_op_callback] (0x20000): [RID#9] Handling LDAP
operation [26][server: [10.23.x.x:389] IPA EXOP] took [10083.267] milliseconds.
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [ipa_s2n_exop_done] (0x0040): [RID#9] ldap_extended_operation
result: Time limit exceeded(3), (null).
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [ipa_s2n_exop_done] (0x0040): [RID#9] ldap_extended_operation
failed, server logs might contain more details.
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [sdap_op_destructor] (0x2000): [RID#9] Operation 26 finished
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [ipa_s2n_get_user_done] (0x0040): [RID#9] s2n exop request
failed.
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [sdap_id_op_done] (0x4000): [RID#9] releasing operation
connection
(Sat Apr 15 12:07:19 2023): [be[ipa.example.com]] [ipa_subdomain_account_done] (0x0040): [RID#9] ipa_get_*_acct
request failed: [1432158230]: Network I/O Error.
```

`ldap_extended_operation failed` or `s2n exop request failed` indicates that the IPA client has sent the request to the IPA server, but the connection with the LDAP server due to some reason the request has failed. The IPA server's sssd log will contain more information. Compare the logs between both client and server from the same timeframe. The IPA client s2n request also hits the IPA directory server `extdom` plugin, investigating IPA 389 directory server access/error logs may be useful here too. For more details, refer to the IPA common issues.

```
(Sat Apr 15 01:37:19 2023): [be[ipa.example.com]] [ipa_s2n_exop_done] (0x0040): ldap_extended_operation result:
Server is busy(51), Too many extdom instances running.
```

The above log indicates that on the server, the client is currently using 80% of the worker threads of the directory server are busy with requests coming from SSSD clients. To avoid having other important tasks blocked, no further requests will be accepted until more workers are free again. So increasing the number of worker threads on the Directory/IPA server should be a workaround. Refer to the performance tuning documentation for more details.

ldap install tls failed/ ldap*install*tls failed: [Connect error] [unknown error]

```
(Sat Apr 15 01:57:19 2023): [be[example.com]] [sss_ldap_init_sys_connect_done] (0x0020): ldap_install_tls
failed: [Connect error]
```

If sssd is not able to create a TLS/SSL connection with the LDAP server due to some reason, then ldap_install_tls failed is observed. There may be an issue with the certificates or LDAP server. The above error indicates that, the hostname is not matching with subjectAltName in the certificate.

```
(Sat Apr 15 02:11:19 2023): [be[default]] [sss_ldap_init_sys_connect_done] (0x0020): ldap_install_tls failed:
[Connect error] [unknown error]
....
  (Sat Apr 15 02:11:19 2023): [be[default]] [sdap_sys_connect_done] (0x0020): sdap_async_connect_call request
failed: [1432158304]: TLS handshake was interrupted.
```

The above error message is a generic message that simply indicates that ldap_install_tls is failing. In order to find the exact reason, enable SSSD debugging using troubleshooting sssd and add `ldap_library_debug_level = -1` under the `[$domain]` section of the `/etc/sssd/sssd.conf` .
Note from the SSSD side it is hard to tell, because the SSSD logs can only show what libldap is returning. libldap itself depends on OpenSSL to setup the TLS connection.

ldap sasl interactive bind failed

ldap_sasl_interactive_bind_s failed as the name suggests the bind is failing here sssd is not able to create SASL bind with the LDAP server.

```
(2023-04-07 12:41:21): [be[example.local]] [ad_sasl_log] (0x0040): SASL: GSSAPI Error: Unspecified GSS failure.
Minor code may provide more information (KDC has no support for encryption type)
(2023-04-07 12:41:21): [be[example.local]] [sasl_bind_send] (0x0020): ldap_sasl_interactive_bind_s failed (-2)
[Local error]
```

The above log indicates the bind is failing since the KDC’s supported encryption type does not match the RHEL encryption type. If this is specific to `RHEL8.x` then investigate the enabled crypto policy. By default, `SSSD` supports `RC4, AES-128` and `AES-256` Kerberos encryption types. In RHEL8 `RC4` encryption has been deprecated and disabled by default, as it is considered less secure than the newer `AES-128` and `AES-256` encryption types. Without any common encryption types, communication between RHEL hosts and AD domains might not work, or some AD accounts might not be able to authenticate. To remedy this situation, enable AES encryption support in Active Directory (recommended option) or enable `RC4` support in RHEL.

```
(2023-03-20 15:32:07): [be[example.com]] [sasl_bind_send] (0x0100): [RID#4] Executing sasl bind mech: GSS-
SPNEGO, user: RHEL_BOX$
...
(2023-03-20 15:32:07): [be[example.com]] [sasl_bind_send] (0x0020): [RID#4] ldap_sasl_interactive_bind_s failed
(-2)[Local error]
(2023-03-20 15:32:07): [be[example.com]] [sasl_bind_send] (0x0080): [RID#4] Extended failure message:
[SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information
(Cannot contact any KDC for realm 'root.example.com')
```

The above log indicates that the bind is failing since `SSSD` is not able to contact the KDC server.

GPO access check failed: [1432158235](Host Access Denied)

```
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400): RESULTANT POLICY:
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400): gpo_map_type: Remote
Interactive
....
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400): CURRENT USER:
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):        user_sid = S-1-5-21-
3431094107-1054041429-3049316506-1162
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   group_sids[0] = S-1-5-21-
3431094107-1054041429-3049316506-1149
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   group_sids[1] = S-1-5-21-
3431094107-1054041429-3049316506-513
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   group_sids[2] = S-1-5-21-
3431094107-1054041429-3049316506-1189
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   group_sids[3] = S-1-5-21-
3431094107-1054041429-3049316506-1123
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   group_sids[4] = S-1-5-11
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400): POLICY DECISION:
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):  access_granted = 0
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_check] (0x0400):   access_denied = 0
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_perform_hbac_processing]
(0x0040): GPO access check failed: [1432158235](Host Access Denied) <---
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_cse_done] (0x0040): HBAC processing failed:
[1432158235](Host Access Denied} <-------
(Thu Aug  6 13:41:42 2023) [sssd[be[example.com]]] [ad_gpo_access_done] (0x0040): GPO-
based access control failed. <–
```

The above log indicates the authentication has been denied by GPO. We can by pass the GPO by setting `ad_gpo_access_control = disabled` under the `[$domain]` section of the `/etc/sssd/sssd.conf` .
In order to find the exact reason why the access has been denied by GPO focus on the `RESULTANT POLICY` it will show the `allowed_sids` and `denied_sids` on the basis of this the access will be granted or denied. In the below example current `user_sid` is present on the `allowed_sids[0]` list and as a result the access is granted if the `user_sid` is not present is `allowed_sid` then the access will be denied.

```
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] RESULTANT POLICY:
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] gpo_map_type: Network
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_size = 5
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_sids[0] = S-1-5-21-1454471165-2077806209-1801674531-9013140 <---
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_sids[1] = S-1-5-21-1454471165-2077806209-1801674531-9019798
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_sids[2] = S-1-5-21-1454471165-2077806209-1801674531-8285000
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_sids[3] = S-1-5-21-1454471165-2077806209-1801674531-5236220
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] allowed_sids[4] = S-1-5-21-1454471165-2077806209-1801674531-5197589
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] denied_size = 5
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  denied_sids[0] = S-1-5-21-1454471165-2077806209-1801674531-9013141
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  denied_sids[1] = S-1-5-21-1454471165-2077806209-1801674531-9019799
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  denied_sids[2] = S-1-5-21-1454471165-2077806209-1801674531-8285001
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  denied_sids[3] = S-1-5-21-1454471165-2077806209-1801674531-5236227
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  denied_sids[4] = S-1-5-21-1454471165-2077806209-1801674531-5197599
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] CURRENT USER:
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]        user_sid = S-1-5-21-1454471165-2077806209-1801674531-8996978
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]   group_sids[0] = S-1-5-21-1454471165-2077806209-1801674531-9013140<-- matched
.....
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491] POLICY DECISION:
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]  access_granted = 1
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_check] (0x0400): [RID#24491]   access_denied = 0
(2023-07-03  8:11:17): [be[example.com]] [ad_gpo_access_done] (0x0400): [RID#24491] GPO-based access control successful.
```

SBR Identity Management **Product(s)** Red Hat Enterprise Linux **Category** Troubleshoot **Component** sssd

Tags troubleshooting **Article Type** General

Was this helpful?

YES NO

Get notified when this content is updated FOLLOWING

People who viewed this article also viewed

Continues PartialGroupNameException errors with sssd

Discussion - May 14, 2018

SSSD is showing "sss_pac_make_request failed" error.

Solution - Jun 14, 2024

SSSD

Discussion - Apr 19, 2023

Case Links (Red Hat Internal)

04066518 - Salesforce / CaseView+ - rhn-support-abroy

04080760 - Salesforce / CaseView+ - rhn-support-jabsher

04078259 - Salesforce / CaseView+ - rhn-support-abroy

04079208 - Salesforce / CaseView+ - rhn-support-tharring

04074748 - Salesforce / CaseView+ - rhn-support-abroy

04060350 - Salesforce / CaseView+ - rhn-support-abroy

04059193 - Salesforce / CaseView+ - rhn-support-abroy

04024297 - Salesforce / CaseView+ - rhn-support-shas

03982465 - Salesforce / CaseView+ - rhn-support-shas

03988606 - Salesforce / CaseView+ - rhn-support-asakure

Show More

Comments

RED HAT

ACTIVE CONTRIB...

Add comment

Formatting Help

☑ Send notifications to content followers □ Mark comment as private

Submit

Quick Links

Help

Site Info

Related Sites

About

Red Hat Subscription Value

About Red Hat

Red Hat Jobs

About Red Hat

Jobs

Events

Locations

Contact Red Hat

Red Hat Blog

Diversity, equity, and inclusion

Cool Stuff Store

Red Hat Summit

Privacy statement

Terms of use

All policies and guidelines

Digital accessibility

Cookie preferences