rsyslog parent startup failure: error reading "fork pipe"

RSyslog

I’ve been configuring my OpenMediaVault NAS server for RSyslog shipping logs to my centralised RSyslog setver, when I experienced a cryptic error:

rsyslog startup failure: error reading "fork pipe": No such file or directory

rsyslog didn’t start, so it took me a bit to investigate.

Turns out, the issue was mismatch of RSyslog config syntax: OMV used one version, my templates used another.

Specifically, I’m using old-syntax multi-line way of
describing global TLS settings for configuring client side of RSyslog:

```plaintext
global(  
    DefaultNetstreamDriver="gtls"  
    DefaultNetstreamDriverCAFile="/etc/rsyslog.d/ca.crt"  
    DefaultNetstreamDriverCertFile="/etc/rsyslog.d/helios4.crt"  
    DefaultNetstreamDriverKeyFile="/etc/rsyslog.d/helios4.key"  
)
```

But earlier in the file I used a more recent way of configuring RSyslog modules:

```plaintext
module(load="imtcp")
input(type="imtcp" port="514")
```

It seems RSyslog doesn’t support this kind of mixing config styles – so one of these config stanzas needs rewriting. In my case, I actually only needed imtcp for debug purposes – so I just commented it out and RSyslog restarted just fine.

See Also

- [Centralised RSyslog](#)
- [RSyslog by Hostname](#)
How To Check SSL Connection with OpenSSL

OpenSSL

I’m tidying up Centralised RSyslog setup on the newly reinstalled becky Raspberry Pi system. One of the tasks at hand was to configure TLS based encrypted log shipping from my dedicated servers to home RSyslog server, this post shows the command and technique I use.

How To Check SSL Connection

What we do is run openssl command with the s_client option and specify remote server we’re testing connection to. It can be an HTTPS connection (port 443) to a website (will do a post about it some other time), but in my case I’m connecting to home office server becky.ts.fm with port 6514 (TLS encrypted port for Syslog):

root@s2:/ # openssl s_client -connect becky.ts.fm:6514
CONNECTED(00000003)
depth=0 CN = becky.ts.fm, O = Tech Stack Solutions, L = Dublin, C = IE
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 CN = becky.ts.fm, O = Tech Stack Solutions, L = Dublin, C = IE
verify error:num=21:unable to verify the first certificate
verify return:1
Certificate chain
  0 s:/CN=becky.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
  i:/CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
Server certificate
-----BEGIN CERTIFICATE-----
MIIEJDCCAoygAwIBAgIUJGqZcuyXa7ekrK+U8yfB2Cu54FYwDQYJKoZIhvNAQ
EL
jMNHiZ0zdolHWzkV6iKc20Mx0v3ftQ1TsE7vg+/Z2fTSv2f0uirPZUpSewr
75
ABRIDGED
9n7UHknn7/mV+lLclo0A8oyXB5zeVf+lxLufVRYhEIpLFTvRiu0Go6PWOgjwMo
PM
QB/0E6WgtSDMf43f9qzSdtKNgHFw1MpxVQdULSabanI6n0gpfuUIvKDBmBazgh6
lR
RztqUqz09pE=
-----END CERTIFICATE-----
subject=/CN=becky.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
issuer=/CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
Acceptable client certificate CA names
/ CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
Client Certificate Types: RSA sign, DSA sign, ECDSA sign
  Requested Signature Algorithms:
  RSA+SHA256:0x09+0x08:0x04+0x08:ECDSA+SHA256:0x07+0x08:
  RSA+SHA384:0x0A+0x08:0x05+0x08:ECDSA+SHA384:RSA+SHA512:
  0x0B+0x08:0x06+0x08:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
Shared Requested Signature Algorithms:
  RSA+SHA256:ECDSA+SHA256:RSA+SHA384:ECDSA+SHA384:RSA+SHA512:
  ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1
Peer signing digest: SHA512
Server Temp Key: ECDH, P-256, 256 bits
SSL handshake has read 1704 bytes and written 427 bytes
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
  Protocol : TLSv1.2
  Cipher    : ECDHE-RSA-AES256-GCM-SHA384
  Session-ID: 224B0D3C5183426D7DDAD5A5FB361BC9C5175EC9CB0AA6A2F396DA7E7178080
  Session-ID-ctx:
  Master-Key: 6BE67A8AD4E22029DE1B3D0DE1F4351FD0488AB1D8ABC7E25187
  Key-Arg    : None
  Krb5 Principal: None
  PSK identity: None
  PSK identity hint: None
  Start Time: 1583790681
  Timeout    : 300 (sec)
  Verify return code: 21 (unable to verify the first certificate)

As you can see, there is a problem. I’m checking the last line in the output, which should return code ok, but it tells me that my s2 server can’t verify the first (only) certificate presented by my RSyslog server becky.ts.fm.

From experience, I know that’s because s2 somehow needs to acknowledge the certificate authority (CA) that issued certificate for becky.ts.fm.

How Successful TLS Connection Looks in OpenSSL
So if I specify this CA cert as a command line option, openssl will establish TLS connection and confirm code ok:

Now let’s specify the CA certificate I used for RSyslog, the connection and certificates verification should work just fine now:

```
root@s2:/ # openssl s_client -CAfile /etc/rsyslog.d/ca.crt -connect becky.ts.fm:6514
CONNECTED(00000003)
depth=1 CN = syslog.ts.fm, O = Tech Stack Solutions, L = Dublin, C = IE
  verify return:1
depth=0 CN = becky.ts.fm, O = Tech Stack Solutions, L = Dublin, C = IE
  verify return:1
Certificate chain
  0 s:/CN=becky.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
  i:/CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
Server certificate
-----BEGIN CERTIFICATE-----
MIIEJDCCAoygAwIBAgIUJGqZcuyXa7ekrK+U8yfB2Cu54FYwDQYJKoZIhvcNAQEL
BQAwVDEVMBMGAGAYCAMEExGyYBQgIQQQgAAABAD
jMNHiz0zdzo1HWzkV6iKc20Mx0v3ftQ1TsE7vg+/Z2fTSv2f0uirPZUPegSzwr
759n7UHknn7/mV+llclo0A8oyX85zeVf+lxLufVRyhEIpLFVtRiu0Go6PW0gwMo
PMQB/0E6WgtSDMf43f9qzSdtKNgHFw1MpxVQdULSabnI6n0gpfuUIvKDBmBazgh6
lR
RtZqUqz09pE=
-----END CERTIFICATE-----
subject=/CN=becky.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
issuer=/CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
Acceptable client certificate CA names
/CN=syslog.ts.fm/O=Tech Stack Solutions/L=Dublin/C=IE
```
Client Certificate Types: RSA sign, DSA sign, ECDSA sign

Requested Signature Algorithms:
- RSA+SHA256:0x09+0x08:0x04+0x08:ECDSA+SHA256:0x07+0x08:RSA+SHA384:
  0x0A+0x08:0x05+0x08:ECDSA+SHA384:RSA+SHA512:0x0B+0x08:
- 0x06+0x08:ECDSA+SHA512:RSA+SHA1:ECDSA+SHA1

Shared Requested Signature Algorithms:
- RSA+SHA256:ECDSA+SHA256:
- RSA+SHA384:ECDSA+SHA384:
- RSA+SHA512:ECDSA+SHA512:
- RSA+SHA1:ECDSA+SHA1

Peer signing digest: SHA512

Server Temp Key: ECDH, P-256, 256 bits

SSL handshake has read 1704 bytes and written 427 bytes
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
  Protocol: TLSv1.2
  Cipher: ECDHE-RSA-AES256-GCM-SHA384

Session-ID: C6797515EEA312D7A9EC6685F895AE004798550FF70619E85F24AB5ACF80F0A9

Session-ID-ctx: Master-Key: 4B84DF3CFE9697EEC634DC271B2A490D94B7A7AB1CA218F016B1ED141FA1479C
  Key-Arg: None
  Krb5 Principal: None
  PSK identity: None
  PSK identity hint: None
  Start Time: 1583790782
  Timeout: 300 (sec)
  Verify return code: 0 (ok)

That’s it – this means secure connection establishes successfully, so I can enjoy secure log shipping from s2 to becky.ts.fm.
I’m converting one of my Raspberry Pi systems, becky, into an **RSyslog-based log collector**, and while there’s not enough knowledge for a complete tutorial yet, I think I’ll start making short notes here in case someone comes looking for them.
host name

One of the most common tasks after you configure your remote servers to ship logs into your new RSyslog collector is to start logging events into separate log files.

Specifically, you may want to have one log per each server, perhaps with the hostname in the filename.

Here’s how you do this. Add the following lines to /etc/rsyslog.conf, taking into account that your logs location may not be the /logs filesystem but some other path:

$template DynaFile,"/logs/%HOSTNAME%.log"
.* ?DynaFile

... and restart rsyslogd:

root@becky:/logs # systemctl restart rsyslog

That’s it, I can now see that there are separate logs created in /logs for my home office systems: becky, s7 and server are the hostnames.

root@becky:/logs # ls -al
total 284
  drwxr-xr-x  4 root root  4096 Mar  1 12:40 .
  drwxr-xr-x 26 root root  4096 Feb 28 16:51 ..
-rw-r-----  1 root adm  9554 Mar  1 12:59 becky.log
-rw-r-----  1 root adm   370 Mar  1 12:55 s7.log
-rw-r-----  1 root adm 12786 Mar  1 12:41 server.log

See Also

- Advanced Unix Commands
- Centralised RSyslog
- RSyslog
- linux ls