Closed · Issue created 2 years ago by **Daniel Kahn Gillmor**

GTlsClientConnection warning about NULL server-identity property

Thanks for the release of balsa 2.6.0!

I tested it out today with a connection to imap://bob@protected-headers.cmrg.net, and found this warning on stderr:

```
(balsa:891359): GLib-Net-WARNING **: 12:43:11.244: GTlsClientConnection certificate verification will fail because its server-identi
```

The application worked as i expect it to, but i haven't tested it against (for example) an invalid x.509 certificate.

I haven't dug into it further yet, sorry! If balsa is doing its own certificate checking, then it should probably also do something to suppress this warning so that people don't freak out about it.

Drag your designs here or click to upload.

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Related merge requests 2

fix NULL server-identity TLS warning with recent gio
!26

Improve TLS certificate validation error message
!27

Activity

Albrecht Dreß @albrecht · 2 years ago · Developer

Thanks for the report!

Unfortunately, I cannot reproduce the issue, neither with the `master` (gmime 2.6) nor with the 'gmime3' branch, on my Ubuntu 18.04 LTS box (gio ver. 2.56.4), e.g.

```
albrecht@deneb:~/Balsa/git/balsa-gmime3$ G_MESSAGES_DEBUG=libnetclient src/balsa
[…]
(balsa:5041): libnetclient-DEBUG: 19:21:47.135: connected to protected-headers.cmrg.net
(balsa:5041): libnetclient-DEBUG: 19:21:47.586: R '* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITE
(balsa:5041): libnetclient-DEBUG: 19:21:47.586: W '1 StartTLS'
(balsa:5041): libnetclient-DEBUG: 19:21:47.686: R '1 OK Begin TLS negotiation now.'
(balsa:5041): libnetclient-DEBUG: 19:21:48.100: connection is encrypted
(balsa:5041): libnetclient-DEBUG: 19:21:48.100: W '2 CAPABILITY'
(balsa:5041): libnetclient-DEBUG: 19:21:48.199: R '* CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+
(balsa:5041): libnetclient-DEBUG: 19:21:48.338: R '2 OK Pre-login capabilities listed, post-login capabilities have more.
[…]
```

Actually, the `server_identity` argument in the call to `g_tls_client_connection_new()` is set to `NULL`, which is explicitly allowed at least for gio 2.56.

Which version of gio do you use?

Peter Bloomfield @peterb · 2 years ago · Developer

That warning is issued by `verify_peer_certificate()` (in glib-networking/tls/base/gtlsconnection-base.c). I set a break-point on it, but it doesn't get triggered. I'm connecting to my usual IMAP and POP3 servers, not @dkg 's test server, though.

Daniel Kahn Gillmor @dkg · 2 years ago · Author · Contributor

I'm using debian testing/unstable for this test at the moment, on a which means both `libglib2.0-dev` and `glib-networking` are at version 2.64.1-1 -- i think that's the same as the version of "gio".

If i can provide you with other information, please let me know!

Peter Bloomfield @peterb · 2 years ago · Developer

Could you run `balsa` in `gdb` with a break-point at `verify_peer_certificate`? Seeing the code path that results in that warning could be really helpful.

Peter Bloomfield @peterb · 2 years ago · Developer

Huh--I'm seeing the warning now. I upgraded to Fedora 32 yesterday, and merged `gmime3` into `master` today, and I don't know which brought up the warning.

Stack trace:

```
(balsa:63045): GLib-Net-WARNING **: 11:28:48.809: GTlsClientConnection certificate verification will fail because its serv

Thread 8 "balsa" received signal SIGTRAP, Trace/breakpoint trap.
[Switching to Thread 0x7fffd97b0700 (LWP 63068)]
(gdb) bt
#0  g_log_writer_default (log_level=<optimized out>, log_level@entry=G_LOG_LEVEL_WARNING, fields=fields@entry=0x7fffd97af
#1  0x00007ffff7ed3828 in g_log_structured_array (n_fields=6, fields=0x7fffd97af010, log_level=G_LOG_LEVEL_WARNING) at ..
#2  g_log_structured_array (log_level=G_LOG_LEVEL_WARNING, fields=0x7fffd97af010, n_fields=6) at ../glib/gmessages.c:1898
#3  0x00007ffff7ed4323 in g_log_structured_standard
    (log_domain=log_domain@entry=0x7fffd9ff207f "GLib-Net", log_level=log_level@entry=G_LOG_LEVEL_WARNING, file=file@entry
#4  0x00007fffd9feca1b in verify_peer_certificate (peer_certificate=0x7ffffbc0044a0 [GTlsCertificateGnutls], tls=0x7ffffbc0
#5  update_peer_certificate_and_compute_errors (tls=0x7ffffbc012a0 [GTlsClientConnectionGnutls]) at ../tls/base/gtlsconnec
#6  0x00007fffd9fee9ec in accept_or_reject_peer_certificate (user_data=user_data@entry=0x7ffffbc0121a0) at ../tls/base/gtls
#7  0x00007ffff7ec844b in g_idle_dispatch (source=source@entry=0x7ffffb800bbb0, callback=0x7fffd9fee970 <accept_or_reject_p
#8  0x00007ffff7ecc76f in g_main_dispatch (context=0x7ffffbc011c70) at ../glib/gmain.c:3309
#9  g_main_context_dispatch (context=0x7ffffbc011c70) at ../glib/gmain.c:3974
#10 0x00007ffff7eccaf8 in g_main_context_iterate (context=context@entry=0x7ffffbc011c70, block=block@entry=1, dispatch=disp
#11 0x00007ffff7eccbc3 in g_main_context_iteration (context=0x7ffffbc011c70, may_block=may_block@entry=1) at ../glib/gmain
#12 0x00007fffd9febc39 in crank_sync_handshake_context (tls=<optimized out>, cancellable=<optimized out>) at ../tls/base/g
#13 0x00007fffd9fee698 in g_tls_connection_base_handshake (conn=<optimized out>, cancellable=0x0, error=0x7fffd97af8f0) at
#14 0x00000000004f608c in net_client_start_tls (client=0xa061c0 [NetClientPop], error=error@entry=0x7fffd97af8f0) at ../ba
#15 0x00000000004f730b in net_client_pop_connect (client=client@entry=0xa061c0 [NetClientPop], greeting=greeting@entry=0x4
#16 0x00000000004ce115 in libbalsa_mailbox_pop3_startup (msg_list=0x7fffd97af8a8, name=0xb82e60 "ATT", mailbox_pop3=0x5b44
    at ../balsa/libbalsa/mailbox_pop3.c:506
#17 libbalsa_mailbox_pop3_check (mailbox=0x5b4470 [LibBalsaMailboxPOP3]) at ../balsa/libbalsa/mailbox_pop3.c:633
#18 0x00000000004b931d in libbalsa_mailbox_check (mailbox=0x5b4470 [LibBalsaMailboxPOP3]) at ../balsa/libbalsa/mailbox.h:3
#19 0x00000000004568a9 in bw_check_mailbox (mailbox=<optimized out>) at ../balsa/src/main-window.c:3294
#20 0x00007ffff7ef6652 in g_thread_proxy (data=0xd2f120) at ../glib/gthread.c:807
#21 0x00007ffff37e2432 in start_thread () at /lib64/libpthread.so.0
#22 0x00007ffff37109d3 in clone () at /lib64/libc.so.6
(gdb)
```

Since, as @albrecht noted, the `server-identity` parameter is *documented* as `nullable`, issuing this warning seems like an over-reaction.

Albrecht Dreß @albrecht · 2 years ago · Developer

> Since, as @albrecht noted, the `server-identity` parameter is *documented* as `nullable`, issuing this warning seems like an over-reaction.

Yes! And the git blame record indicates that this statement has been added only 21 days ago (compare with the previous blame record)...

Looks either a logical or a documentation bug in glib to me.

Please register or sign in to reply

Albrecht Dreß mentioned in commit e8952e3c 2 years ago

Albrecht Dreß mentioned in merge request !26 (merged) 2 years ago

Albrecht Dreß @albrecht · 2 years ago Developer

Giving some more thoughts to this issue, I think it is actually a good idea to set the identity anyway, even if the gio code needs to be modified and/or the documentation fixed. @dkq @peterb – could you please review the merge request !26 (merged) and check if it fixes the issue with the brand-new gio version?

Thanks, Albrecht.

Peter Bloomfield @peterb · 2 years ago Developer

It works for me! @dkq if it works for you, I'll go ahead and merge.

Michael Catanzaro @mcatanzaro · 2 years ago Developer

Did you have any other code in balsa to ensure that the hostname of the certificate matches the server you're connected to? Probably not?

It's possible that balsa is checking server identity manually, either by passing the identity to g_tls_certificate_verify() (unlikely, since that would be odd when GTlsClientConnection can do it for you), or by extracting the certificate PEM or DER and parsing it using another library like GnuTLS (unlikely, since it's a lot of work).

Anyway, if server identity isn't checked, then evil.com could present a certificate that is valid for evil.com, and balsa would trust it when connecting to honest-mailserver.com. So that would be bad.

Michael Catanzaro @mcatanzaro · 2 years ago Developer

BTW if it turns out that you don't have code for this, then you can fix by checking to see if G_TLS_CERTIFICATE_BAD_IDENTITY is set in the GTlsCertificateFlags passed to accept-certificate and rejecting it if so (now that you set server identity). That error would *always* have been set prior to this commit (due to lack of server identity), so balsa must be ignoring it currently and is therefore responsible for doing that verification on its own. (Right?) Basically it should be OK if you look up the hostname in a map to see if it has a pinned cert, but not OK if you just have a bunch of pinned certs not otherwise associated with particular hosts.

Albrecht Dreß @albrecht · 2 years ago Developer

> Did you have any other code in balsa to ensure that the hostname of the certificate matches the server you're connected to? Probably not?

No. The code just calls `g_tls_client_connection_new()` with the default validation flags (`G_TLS_CERTIFICATE_VALIDATE_ALL`), and a callback is displayed to the user on error. I must admit that I never tested if a `G_TLS_CERTIFICATE_BAD_IDENTITY` error is triggered, though; but at least the callback *is* called e.g. for the typical self-signed certificate of the local SMTP server.

The underlying `GSocketConnection` was created by calling `g_socket_client_connect_to_host()` , and my (obviously wrong) assumption was that *this* identity was also used for the certificate host name verification. I now changed the code to calling `g_network_address_parse()` , and the result is passed to both `g_socket_client_connect()` and `g_tls_client_connection_new()` .

It is somehow logical that the check cannot be performed from the underlying connection (e.g. if it is created by IP address), but it would be great if the documentation would be more explicit here. At least, the "nullable" property of the `server_identity` parameter is really confusing!

> That error would always have been set prior to this commit (due to lack of server identity), so balsa must be ignoring it currently and is therefore responsible for doing that verification on its own.

This is strange – as I noted above, the default `G_TLS_CERTIFICATE_VALIDATE_ALL` is *not* altered, i.e. the identity *should* have been checked, `G_TLS_CERTIFICATE_BAD_IDENTITY` is *not* ignored, but I *never* saw such a verification error!

Peter Bloomfield @peterb · 2 years ago Developer

Without @albrecht 's patch, the error that's seen in the `"accept-certificate"` handler is `G_TLS_CERTIFICATE_UNKNOWN_CA` , not `G_TLS_CERTIFICATE_BAD_IDENTITY` .

Michael Catanzaro @mcatanzaro · 2 years ago Developer

> I must admit that I never tested if a G_TLS_CERTIFICATE_BAD_IDENTITY error is triggered, though; but at least the callback is called e.g. for the typical self-signed certificate of the local SMTP server.

If you don't set server identity on the GTlsClientConnection, *every* certificate verification should fail with this error, every time. If you're not seeing that (before this patch), then there's something wrong.

> The underlying GSocketConnection was created by calling g_socket_client_connect_to_host(), and my (obviously wrong) assumption was that this identity was also used for the certificate host name verification.

Well... it really should be. GSocketClient should absolutely set the server identity of its GTlsClientConnection. Is it being *unset* somehow later on?

Albrecht Dreß @albrecht · 2 years ago Developer

Just to get the picture of the status on Debian Buster (gio 2.58.3), I used the attached simple code and dummy certs for a few tests. test-tls.tar.xz

As a first test, check "this" server, which does *not* throw an error, regardless of the `server_identity` :

```
test@buster:~/test-tls$ ./test_server_valid gitlab.gnome.org:443 1
** Message: 18:25:44.625: set remote server_identity: yes
** Message: 18:25:44.923: g_tls_connection_handshake(): OK (no error)
test@buster:~/test-tls$ ./test_server_valid gitlab.gnome.org:443 0
** Message: 18:25:49.539: set remote server_identity: no
** Message: 18:25:49.839: g_tls_connection_handshake(): OK (no error)
```

Now start a TLS dummy server, using a non-standard CA and a valid certificate:

```
/usr/bin/gnutls-serv -a --x509keyfile=cert_u.pem --x509certfile=cert_u.pem --echo -p 65001
```

As the CA is not known, `G_TLS_CERTIFICATE_UNKNOWN_CA` is emitted in both cases:

```
test@buster:~/test-tls$ ./test_server_valid localhost:65001 0
** Message: 18:26:33.285: set remote server_identity: no
** Message: 18:26:33.328: cert_accept_cb: conn=0x55806e36a150, cert=0x7f93040066b0 for 'localhost', error=0x1
** Message: 18:26:33.328: g_tls_connection_handshake(): ERROR (Unacceptable TLS certificate)
test@buster:~/test-tls$ ./test_server_valid localhost:65001 1
** Message: 18:26:37.002: set remote server_identity: yes
** Message: 18:26:37.045: cert_accept_cb: conn=0x5603d7017180, cert=0x7fb8e80066b0 for 'localhost', error=0x1
** Message: 18:26:37.045: g_tls_connection_handshake(): ERROR (Unacceptable TLS certificate)
```

Adding the CA certificate, again *no* error is thrown, regardless of the `server_identity` :

```
test@buster:~/test-tls$ ./test_server_valid localhost:65001 0 $(pwd)/ca_cert.pem
** Message: 18:31:51.363: set remote server_identity: no
** Message: 18:31:51.373: using CA certs /home/test/test-tls/ca_cert.pem
** Message: 18:31:51.391: g_tls_connection_handshake(): OK (no error)
test@buster:~/test-tls$ ./test_server_valid localhost:65001 1 $(pwd)/ca_cert.pem
** Message: 18:31:58.315: set remote server_identity: yes
** Message: 18:31:58.323: using CA certs /home/test/test-tls/ca_cert.pem
** Message: 18:31:58.340: g_tls_connection_handshake(): OK (no error)
```

Use the dummy server, but with a certificate with a wrong `subjectAltName` :

```
/usr/bin/gnutls-serv -a --x509keyfile=bad_dns.pem --x509certfile=bad_dns.pem --echo -p 65001
```

The bad cert is accepted silently (!) if the `server_identity` is `NULL` , otherwise the (expected) `G_TLS_CERTIFICATE_BAD_IDENTITY` is thrown:

```
test@buster:~/test-tls$ ./test_server_valid localhost:65001 0 $(pwd)/ca_cert.pem
** Message: 18:34:44.449: set remote server_identity: no
** Message: 18:34:44.456: using CA certs /home/test/test-tls/ca_cert.pem
** Message: 18:34:44.474: g_tls_connection_handshake(): OK (no error)
test@buster:~/test-tls$ ./test_server_valid localhost:65001 1 $(pwd)/ca_cert.pem
** Message: 18:34:47.861: set remote server_identity: yes
** Message: 18:34:47.870: using CA certs /home/test/test-tls/ca_cert.pem
```

```
** Message: 18:34:47.888: cert_accept_cb: conn=0x56429d998170, cert=0x7fec000066b0 for 'smtp.evil.com', error=0x2
** Message: 18:34:47.888: g_tls_connection_handshake(): ERROR (Unacceptable TLS certificate)
```

Conclusion: At least up to gio 2.58, the *library* seems to skip the identity check if `server_identity` is `NULL`, even if the respective flag *is* set – which is bad, as a MITM attack wouldn't be detected. At least, this should be documented.

> If you don't set server identity on the GTlsClientConnection, *every* certificate verification should fail with this error, every time. If you're not seeing that (before this patch), then there's something wrong.

@mcatanzaro – The above tests IMHO clearly demonstrate that this is *not* the case. So this means gio ≤ 2.58 (at least) is buggy? Or did I miss something in my POC code?

> Well... it really should be. GSocketClient should absolutely set the server identity of its GTlsClientConnection. Is it being unset somehow later on?

@mcatanzaro – No, never. And I don't think this would ever work, as the Connectable can be created using a multitude of methods, which may or may not include the proper identity.

Peter Bloomfield @peterb · 2 years ago — Developer

I get the same results with glib2-2.64.1-1.fc32.x86_64, but with the new warning for the cases where `server-identity` is not set.

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

OK, looks like problems, so this is on my TODO to investigate.

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

I'm not done investigating yet, but seems pretty clear this is going to turn into a GLib CVE. :/ Good findings, ~~Peter~~ Albrecht.

Documentation says:

> If the G_TLS_CERTIFICATE_BAD_IDENTITY flag is set in "validation-flags", this object will be used to determine the expected identify of the remote end of the connection; if "server-identity" is not set, or does not match the identity presented by the server, then the G_TLS_CERTIFICATE_BAD_IDENTITY validation will fail.

The documented behavior is the intended behavior. But indeed, you're right, the implemented behavior is that hostname verification is just skipped if server identity is not set. I'll probably request a CVE for GLib, because reasonable applications would expect the documentation to be correct.

Seems like a pretty serious hole in our otherwise decent testsuite. Created glib-networking#135 (closed).

Edited by Michael Catanzaro 2 years ago

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

> Well... it really should be. GSocketClient should absolutely set the server identity of its GTlsClientConnection. Is it being unset somehow later on?
>
> @mcatanzaro – No, never. And I don't think this would ever work, as the Connectable can be created using a multitude of methods, which may or may not include the proper identity.

BTW, connectable is the identity, it just gets passed along if using GSocketClient in TLS mode, i.e. if enabled via g_socket_client_set_tls(). Your testcases are manually creating the GTlsClientConnection since that's required to demonstrate the bug. Since GSocketClient passes along its connectable, it's immune to this issue.

It's probably a good idea to use GSocketClient in TLS mode and let it create the GTlsClientConnection for you, unless you have some specific reason to not do it that way. (It's perfectly fine to do manually if you wish, but that opened you up to this bug.) When TLS is enabled on GSocketClient, it returns a GTcpWrapperConnection and then g_tcp_wrapper_connection_get_base_io_stream() will return the GTlsClientConnection.

Peter Bloomfield @peterb · 2 years ago — Developer

Hi @mcatanzaro Glad to see that the issue will be addressed! Credit for the research on `Balsa`'s side is to @albrecht, not me!

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

Oops, yes, you do indeed seem to be different people. :)

Albrecht Dreß @albrecht · 2 years ago — Developer

Hi @mcatanzaro, thanks a lot for the clarification! I agree that the issue qualifies for a cve... feel free to use my POC code if you like!

> It's probably a good idea to use GSocketClient in TLS mode and let it create the GTlsClientConnection for you, unless you have some specific reason to not do it that way.

The reason I implemented it this way was to address STARTTLS for SMTP, POP3 and IMAP (RFC 2595, RFC 3207): open a plain connection, and iff enabled by the user *and* supported by the remote server *and* the initial protocol-specific plain handshake succeeds, switch *the open plain connection* to TLS. Or does GIO provide a better way to implement this?

There might be an other place for improvement in my code, as I never figured out how to assign a user certificate with an encrypted key for certificate-based authentication – I asked this on the mailing list back in 2016, but never got any reply. My implementation now uses GnuTLS directly. If there is a better way, I would highly appreciate if you could enlighten me!

> Oops, yes, you do indeed seem to be different people. :)

...actually... 😉

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

> The reason I implemented it this way was to address STARTTLS for SMTP, POP3 and IMAP (RFC 2595, RFC 3207): open a plain connection, and iff enabled by the user and supported by the remote server and the initial protocol-specific plain handshake succeeds, switch the open plain connection to TLS. Or does GIO provide a better way to implement this?

Nope, that's a good reason to do it this way.

(That said, I would think about the security user experience of STARTTLS and make sure it is not presented as a secure way to send or receive email.)

> There might be an other place for improvement in my code, as I never figured out how to assign a user certificate with an encrypted key for certificate-based authentication – I asked this on the mailing list back in 2016, but never got any reply. My implementation now uses GnuTLS directly. If there is a better way, I would highly appreciate if you could enlighten me!

I don't think it's currently possible. It's the sort of thing we could add if someone wanted to work on supporting it.

Albrecht Dreß @albrecht · 2 years ago — Developer

> Nope, that's a good reason to do it this way.

Ok, good to know that I'm not *too* dumb... 😃

> (That said, I would think about the security user experience of STARTTLS and make sure it is not presented as a secure way to send or receive email.)

Well, it's a secure way (ok, SMTPS/IMAPS/POP3S would be preferred, but not every ISP supports it) to log into the ISP's mail server. The secure way for exchanging messages is of course end-to-end encryption – which Balsa supports!

> I don't think it's currently possible. It's the sort of thing we could add if someone wanted to work on supporting it.

As I mentioned, I have a working solution in Balsa, so it's actually not important, and only a handful of ISP's support user certificate authentication anyway.

Please register or sign in to reply

Peter Bloomfield mentioned in issue glib-networking#130 (closed) 2 years ago

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

I agree that the warning is too aggressive. I'll probably change it to not print unless certificate verification actually fails due to G_TLS_CERTIFICATE_BAD_IDENTITY. If you don't pass the server identity, then GLib will reject the certificate rather than accept it (unless the application overrides that choice, like balsa does), so the API does fail safe.

Edited by Michael Catanzaro 2 years ago

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

BTW dkg, I remember years ago you had reported a bug for Epiphany loading web pages after certificate verification had failed... if nothing else, we're not *that* bad anymore. :P

Michael Catanzaro @mcatanzaro · 2 years ago — Developer

> I'll probably change it to not print unless certificate verification actually fails due to G_TLS_CERTIFICATE_BAD_IDENTITY.

(This plan failed. See glib-networking#130 (closed) for details.)

Peter Bloomfield @peterb · 2 years ago Developer

@mcatanzaro Thanks for your insights and discussion--much appreciated!

Albrecht Dreß mentioned in commit `0ae0fde1` 2 years ago

Albrecht Dreß mentioned in merge request !27 (merged) 2 years ago

Michael Catanzaro mentioned in issue glib-networking#135 (closed) 2 years ago

Michael Catanzaro @mcatanzaro · 2 years ago Developer

So my verdict on this is: balsa was wrong to not set server identity. That's a balsa bug. But GTlsClientConnection is supposed to be fail-safe here. It is a serious security bug affecting balsa, but only because GTlsClientConnection does not properly implement its intended, documented behavior. Accordingly, the CVE belongs to GLib, not to balsa. I will request one.

Michael Catanzaro closed 2 years ago

Albrecht Dreß mentioned in commit `bde37791` 2 years ago

Please register or sign in to reply