

[New issue](#)[Jump to bottom](#)

Having to re-set the weechat.network.gnutls_ca_user env variable every single session. #1763

✓ Closed

kittywritescode opened this issue on Mar 12 · 6 comments

Assignees



Labels

question

Milestone

🔗 3.5

kittywritescode commented on Mar 12 • edited ▼

Question

Everytime I start a session of Weechat and attempt to connect to a server, if the `weechat.network.gnutls_ca_user` config option has already been set to anything, or even nothing at all, and even when it has been correctly set to `"/etc/ssl/certs/ca-certificates.crt"` from previous sessions, I always encounter certificate issues with gnutls and thus TLS handshake will fail, disallowing me from making a secure connection to any server (non-SSL connections still work, of course)

```
gnutls: peer's certificate is NOT trusted
gnutls: peer's certificate issuer is unknown
irc: TLS handshake failed
irc: error: Error in the certificate.
```

However, if I then proceed to `/set weechat.network.gnutls_ca_user` to something **else**, and then *after* that, set it to the correct value with `/set weechat.network.gnutls_ca_user "/etc/ssl/certs/ca-certificates.crt"`, Weechat will handle SSL certificates and connect just fine like it is supposed to. But when I quit and the next time I start Weechat, the same issue shows up again, and then I have to manually set that configuration to something else wrong on purpose, then set it back right again, even when it is already correctly set from the get go!

I have already tried to find a way to circumvent this issue by starting weechat with a new config `weechat --dir /tmp/weechat`, and even there, the same issue happens after I quit and restart.

Right now, my only option to get around this issue is to probably use a script that can `/set weechat.network.gnutls_ca_user` to something different each time I quit the client, and then set the weechat client to run the correct `/set` command option on startup. Still though, I'd rather seek help to see why exactly my WeeChat is behaving this way.

- WeeChat version: 3.4
- OS, distribution and version: EndeavourOS Linux x86_64

 **kittywritescode** added the **question** label on Mar 12

flashcode commented on Mar 12

Member

Hi,

The option `weechat.network.gnutls_ca_user` must be used only to set extra certificates, for example some custom certs, not the system ones.

The system certificates are automatically loaded when `weechat.network.gnutls_ca_system` is on (default value), the path is automatically found, you don't have to set a path any more (since WeeChat 3.2).


Can you just connect fine when you run WeeChat with default config?

If yes, why changing the option `weechat.network.gnutls_ca_user` ?

Anyway there's maybe a bug, could you please give the steps to reproduce the problem, starting with the default config and providing commands to cause the issue?

Thanks.

 **flashcode** added the **waiting info** label on Mar 12

 **flashcode** self-assigned this on Mar 12

kittywritescode commented on Mar 12

Author

I did not realize that the option `weechat.network.gnutls_ca_user` was only used to set additional certs, I thought it was renamed from a deprecated `weechat.network.gnutls_ca_file` option and that it served the same function.

Starting from the default config:

1. Add example server: `server add blackaster gimel.blackaster.xyz/6697 -ssl`

2. Connect to that server: `connect blackaster`

Encounter cert issues:

```
gnutls: peer's certificate is NOT trusted
gnutls: peer's certificate issuer is unknown
irc: TLS handshake failed
irc: error: Error in the certificate.
```

3. Disconnect from server, then `/set weechat.network.gnutls_ca_user` to random gibberish, putting anything inside the quotes, then set it back to the way it was before.

4. Now it is possible to connect to servers without any certificate issues.

It's starting to look like a bug to me.

trygveaa commented on Mar 13

Contributor

I can reproduce it with those steps. `gimel.blackaster.xyz` doesn't have a valid certificate though (it's self signed), so it seems like the bug here is that if you set `weechat.network.gnutls_ca_user` and then reconnect to a server you previously got a certificate error on, WeeChat will not verify if the certificate is valid and just connect to it.

kittywritescode commented on Mar 13 • edited ▼

Author

I can reproduce it with those steps. `gimel.blackaster.xyz` doesn't have a valid certificate though (it's self signed), so it seems like the bug here is that if you set `weechat.network.gnutls_ca_user` and then reconnect to a server you previously got a certificate error on, WeeChat will not verify if the certificate is valid and just connect to it.

It's quite interesting that when WeeChat successfully reconnects to `gimel.blackaster.xyz` and I check myself with `/whois`, it does say that I have a secure connection, which makes sense since I use port 6697 for it, but somehow still strange, since obviously it was not verified.

I wonder if there is a way to make it so that for servers without a valid certificate only, WeeChat will just go ahead and connect to it instead of attempting to verify again? I'd still prefer to keep my `weechat.network.gnutls_ca_system` on, if possible.

flashcode commented on Mar 13

Member

If the certificate is not valid, WeeChat is supposed to **NEVER** connect to it, unless you explicitly force the connection to bypass the certificate checking (with IRC options).

If playing with option `weechat.network.gnutls_ca_user` lets you connect, this is indeed a bug. I'll try to reproduce and then fix the problem.

  **flashcode** removed the `waiting info` label on Mar 13

trygveaa commented on Mar 13

Contributor

It's quite interesting that when WeeChat successfully reconnects to `gimel.blackaster.xyz` and I check myself with `/whois`, it does say that I have a secure connection, which makes sense since I use port 6697 for it, but somehow still strange, since obviously it was not verified.

You're still using an encrypted TLS connection even though the certificate is not verified. The server has no way of knowing if you verified the certificate or not, so `whois` shows it as secure.

I wonder if there is a way to make it so that for servers without a valid certificate only, WeeChat will just go ahead and connect to it instead of attempting to verify again?

I assume you mean disable verification for specific servers, since just connecting to all servers without a valid certificate would be the same as just not doing any verification.

There's a couple of options here:

1. Set `irc.server.blackaster.ssl_fingerprint` to the fingerprint of the certificate. This means that instead of verifying that it's valid, WeeChat will verify that it's this specific certificate. If you know it's not compromised that's just as secure, but it means you have to update this option if the certificate changes.
2. Set the blackaster CA in `weechat.network.gnutls_ca_user`. This means any certificate they create will be accepted by WeeChat. This affects all servers though, not just your blackaster server.
3. Set `irc.server.blackaster.ssl_verify` to off. This disables verification so if they are compromised you won't notice it. So that's not recommended.



1

 **flashcode** closed this as completed in [6004139](#) on Mar 13

  **flashcode** added this to the **3.5** milestone on Mar 13

 **flashcode** added a commit that referenced this issue on Mar 13

 **core:** set again TLS verification functions after GnuTLS options are c... ...

✗ 7102478



MingcongBai mentioned this issue on May 24

weechat: security update to ^3.4.1 AOSC-Dev/aosc-os-abbs#3996

✓ Closed

Assignees



flashcode

Labels

question

Projects

None yet

Milestone

3.5

Development

No branches or pull requests

3 participants

