Quassel IRC - Feature #32

Implement identd support

05/21/2007 05:20 PM - Sputnick

Status:	Resolved	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	al	**% Done:**	0%
Category:	Quassel Core	**Estimated time:**	0.00 hour
Target version:	0.8.0		
OS:	Any		

Description

Quassel should have an integrated identd server. This will allow people to use identd without having to install and/or run a separate daemon on their system.

Related issues:

Related to Quassel IRC - Feature #1137: oidentd support	**Resolved**	**02/13/2012**

History

#1 - 05/23/2007 12:42 AM - EgS

Ident servers are expected to listen on Port 113 which is a privileged port.
Though this sounds like a nice feature - and obviously one that could easily use the cores backend - I don't think that this is a good Idea.

Perhaps include it as an advanced and optional feature when there are some free resources...

#2 - 05/23/2007 02:07 AM - Sputnick

That's a very valid point, and one I did not think of. One would probably need to have Quassel run as root, which is not a good idea for an IRC client, I guess. I don't know (but doubt) if there are ways to spawn a process with root privileges from an unprivileged process, or if one could allow an unprivileged process to open a certain privileged port.

Of course, most Linux users wouldn't mind to start a separate daemon, and I wouldn't be surprised if Windows allowed any process to open privileged ports, so having an included identd for such cases might still be a nice feature.

#3 - 07/08/2007 04:48 PM - Sputnick

Probably not going to happen. identd is overrated anyway.

#4 - 04/03/2008 11:17 AM - Sputnick

Some networks (such as Quakenet) require for special features (such as Trust) an identd server that allows to identify actual users/clients. AFAIK, for Quassel this cannot be done with a separate identd server running, since that couldn't know which connection belongs to which Quassel User.

For this reason, implementing identd support that reports a core user correctly should be taken on the agenda again. There still is the question how to handle the fact that 113 is priviledged - maybe provide a sort of relay for an external service? How do other daemons do that - most don't run as root even if they use a priv'd port?

#5 - 07/01/2008 03:47 PM - seezer

They switch their uid/gid after setting up the privileged tcpsocket.
unistd.h provides setuid/setgid for any posix compatible OS.

#6 - 08/20/2008 02:47 AM - krf

What about using oidentd? oidentd is a seperate service which runs as Identd and handles $HOME/.oidentd.conf files. All you have to do is something like `echo $ID > $HOME/.oidentd.conf` and notify the daemon that you are about to claim the identd service for now (i really dont know the internals). Eggdrop uses it, psyBNC, sBNC, too.
It's packaged and available on all major distros.

#7 - 08/20/2008 12:09 PM - Sputnick

This is a very hackish solution, and one that is bound to fail if multiple users connect to IRC at roughly the same time (which is quite common upon a quasselcore restart, for example).

The way oidentd works: Upon connect, the BNC writes the ID in the config file. If the identd request arrives after a short while, oidentd will use that ID. Now imagine what happens if 10 quassel users connect at the same time... all identd requests will get the same ID of the quassel user who initiated the connect last. This can be a major problem in particular in those scenarios where one actually needs identd, for example for a Quakenet trust.

In order to properly support Quassel, an identd daemon needs a way to map a port number to the actual user. If we don't want to do this from within quassel itself, we could have an external (custom) daemon that reads a config file / queries the db / uses dbus to access that information.

#8 - 08/20/2008 01:43 PM - krf

In either way you have to delay the reconnects. If all the users connect to one Network you have to do that anyway. A normal IRCd throttles connections from the same host. I don't think its very hackish because there are no other (efficient) ways to do that. You have to wait for each single user to receive an identd reply. Again, all the bouncer daemons i know of delay their reconnects upon startup (yes, I like bouncers, heh).

#9 - 08/20/2008 04:27 PM - Sputnick

I'd rather not mix those two issues; throttling should be done automatically by the ircd (otherwise, I'd implement a connect delay into Quassel), but I don't want to depend on the ircd to actually ask for an identd reply. Also I wouldn't know how to figure out from within quassel if the request went through such that I could continue connecting.

Seriously, depending on the behavior of a race condition to perform a basic task like connecting to IRC does not sound like something I would want to support.

I'd much rather implement my own identd implementation tied to quassel via some sync mechanism (i.e. conf file or dbus) or even built into quassel than depend on something like that. After all, the protocol is damn simple. Queries not related to the running quasselcore could be relayed to a proper identd then that listens locally on a different port, this would avoid having to implement the full functionality of identd for the given host.

Another way would be to extend an existing identd (such as oidentd) to query the quasselcore for the correct information, but I think that's even more work and requires upstream cooperation.

#10 - 03/22/2009 12:33 PM - seezer

- Category changed from General / Unspecified to Quassel Core

- Assignee changed from Sputnick to seezer

- % Done changed from 0 to 20

- OS set to Any

#11 - 11/12/2011 07:20 AM - TerrorBite

Sputnick wrote:

> The way oidentd works: Upon connect, the BNC writes the ID in the config file. If the identd request arrives after a short while, oidentd will use that ID. Now imagine what happens if 10 quassel users connect at the same time... all identd requests will get the same ID of the quassel user who initiated the connect last. This can be a major problem in particular in those scenarios where one actually needs identd, for example for a Quakenet trust.

This is not actually the case for oidentd... the config file supports entries of the following format, where localport and destport are port ranges:

```
to <host> lport <localport> from <boundip> fport <destport> {

<capability directive>
}
```

It should be easy for Quassel to write an entry of the following format for each identity a user has configured, and then ensure that outgoing connections using that identity use a source port in the given port range (in fact, the oidentd config file would only need to be changed when a user created, deleted or edited an identity):

```
lport 44000:44020 {
    reply "userident"
}
```

#12 - 02/13/2012 08:59 PM - johu

- Assignee changed from seezer to al

- Target version set to 0.8.0

- % Done changed from 20 to 100

#13 - 02/13/2012 09:01 PM - johu

- Target version changed from 0.8.0 to Some future release

#14 - 02/13/2012 09:02 PM - johu

- Assignee changed from al to seezer

- % Done changed from 100 to 20

#15 - 12/16/2013 10:25 AM - Anonymous

- Status changed from Assigned to Resolved

- Assignee changed from seezer to al

- Target version changed from Some future release to 33

- % Done changed from 20 to 0

The oident integration since f9a73 resolves this issue imo.

#16 - 01/14/2014 01:24 AM - Sputnick

- Target version changed from 33 to 0.8.0