

Plaudertasche

DANIEL GULTSCH

Chatprogramme, die nur ein kleines Subset des Jabber-Protokolls unterstützen, gibt es viele. Möchte man den Client außerdem auf der Konsole benutzen und legt man Wert auf geringen Ressourcenverbrauch, wird die Auswahl schon deutlich geringer. *mcabber* ist ein Jabber-Client, der diese Aufgaben bewältigt und auch für zahlreiche Plattformen verfügbar ist. Seine Konfiguration ist nicht einfach, die Bedienung gewöhnungsbedürftig, es lohnt sich aber trotzdem, ihn zu installieren.

Der Jabber-Client *mcabber* wird auf <http://mcabber.com/> gehostet, wobei dies nur ein Link auf <http://www.lilotux.net/~mikael/mcabber/> ist, unter http://wiki.mcabber.com/index.php/DE_Main_Page ist sogar ein deutschsprachiges Wiki eingerichtet. Hier findet man unter <http://wiki.mcabber.com/index.php/Packages> den Hinweis auf Pakete für das eigene Betriebssystem. Aber die Installation ist natürlich auch aus den Quelltexten möglich. Wie üblich wird dazu das Quellarchiv ausgepackt und mit dem GNU-Dreisatz aus `./configure`, `make` und `make install` das Programm eingespielt. Das Zielverzeichnis läßt sich – wie immer – mit der Option `--prefix` beim Konfigurationslauf angeben. Diese Installation, bei der natürlich der C-Compiler installiert sein muß, ist immer dann interessant, wenn die vorgefertigte Version für das Betriebssystem veraltet ist.

Pakete gibt es für eine ganze Reihe von Betriebssystemen, natürlich diverse Linux-Distributionen, aber auch für FreeBSD, MacOS X und sogar für die Cygwin-Umgebung. Nachfolgend geht es um die nicht ganz triviale Konfiguration auf Basis der Versionen 0.9.6 und 0.9.7. Weil *mcabber* stetig weiterentwickelt wird, gibt es

bei Erscheinen dieser freeX aber sicher bereits eine neuere Veröffentlichung. Die grundlegende Konfiguration und Bedienung ändert sich damit aber nicht.

Konfiguration

Ist das Programm – auf welche Weise auch immer – installiert, muß vor dem ersten Start im Heimatverzeichnis

des Benutzers ein Ordner `.mcabber` angelegt werden. Hier muß sich eine Textdatei `mcabberrc` befinden, die als Basis die folgenden Zeilen enthält:

```
set server = *Server Name*
set username = *Benutzername*
set password = *Passwort*
set resource = mcabber
set priority = 3
set ssl = 1
set ssl_verify = 0
set lang = de
```

```
mcabber - Konsole
[status]
...
[ ] JiQiRen
[ ] Stephan
[ ] Udo
[0] @jabber.ccc
[?] @jabbe
[ ] @
[?] @jabber.c
[x] @conf
[ ] @jabber.
[?] @mabb
[ ] @debianforum
[?] inputmice@jabber.
[?] @jabber.
[?] @get-privacy.
[?] michiSPO
[?] @j4a.info
[ ] @jabber
Special buffer: [status]
[14:53:00] Received an IQ version request from <@jabber.ccc.de/X5-d23a>
[14:53:00] Received an IQ last time request from <@jabber.ccc.de/X5-d23a>
[14:53:00] Received an IQ version request from <@jabber.ccc.de/Adium>
[14:53:00] Received an IQ last time request from <@jabber.ccc.de/Adium>
[0]
```

Bild 1: *mcabber* gleich nach dem Start mit überfülltem Roster

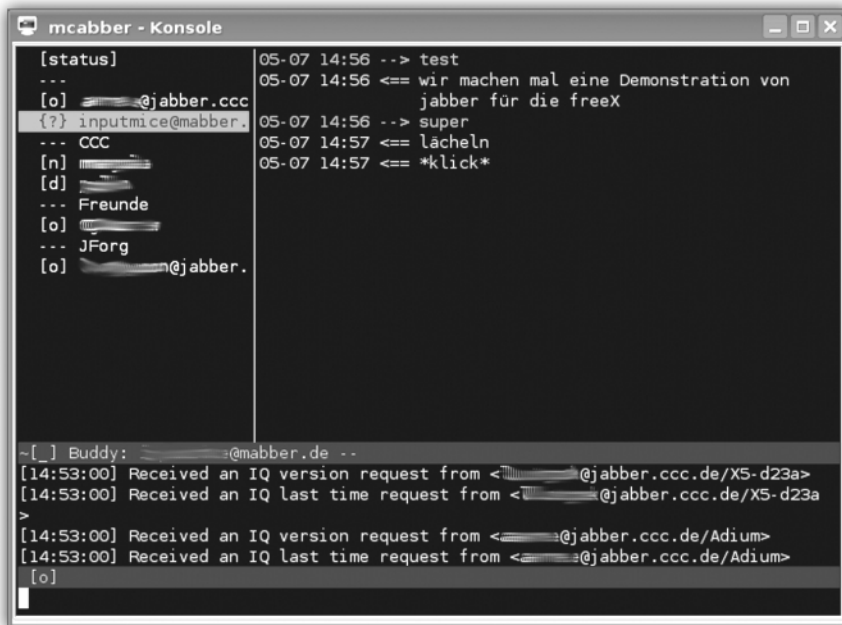


Bild 2: Ein Chat in mcabber mit versteckten Offline-Kontakten im Roster

Wirklich wichtig sind hier nur die Angaben *set server* und *set username*. Wird das Paßwort nicht angegeben, erfragt es *mcabber* bei jedem Start. Hat man mehr als einen Jabber-Client im Einsatz (zum Beispiel auf einem Notebook oder Handy), sollte mit *set resource* dem Client ein von außen ansprechbarer Namen gegeben werden. *set priority* definiert die Reihenfolge der verschiedenen Clients. Sollte der Jabber-Server SSL unterstützen – und die meisten Server tun dies – sollte mit *set ssl = 1* die Verbindung zum Server verschlüsselt werden. Die eingebauten Hilfetexte in mcabber sind fast vollständig übersetzt. Die Sprache mit *set lang = de* auf Deutsch zu setzen lohnt sich also.

Der erste Start

Nach dem Start präsentiert sich mcabber mit einer vierteiligen Ansicht. Auf der linken Seite befindet sich die im Jabber-Jargon »Roster« genannte Kontaktliste. Die größte Fläche nimmt das eigentliche Chatfenster auf der rechten Seite ein. Unter Roster und Chatfenster ist das Logfenster, in dem die mcabber- und auch die Servernachrichten erscheinen. Ganz unten befindet sich eine Eingabezeile, mit der mcabber gesteuert und in dem auch gechattet wird.

Die wichtigsten mcabber-Befehle sind */quit* und */help*. */quit* in der Eingabezeile eingegeben, beendet das Programm. */help* listet alle möglichen Befehle auf. Gibt man */help* als Parameter einen weiteren Befehlsnamen mit, zeigt das Hilfesystem von mcabber weitere Information zu diesem Befehl an. */help quit* gibt beispielsweise eine Erklärung des */quit*-Befehls. Angezeigt wird die Hilfe an zwei verschiedenen Stellen, im Log erscheinen nur die jeweils fünf letzten Zeilen. Der komplette Text verbirgt sich hinter dem obersten Eintrag im Roster, *[status]* genannt, der erst mit einem Druck auf [Enter] aktiviert werden muß. Man braucht übrigens nicht immer den kompletten Befehl einzugeben, meistens reichen die ersten paar Zeichen gefolgt von einem Druck auf die [Tab]-Taste. Genau wie in der Shell kann man mit den Pfeiltasten durch vorherige Befehle navigieren. Ist der Account schon etwas länger in Benutzung und hat man sehr viele Kontakte, möchte man die angezeigten Kontakte eventuell reduzieren. Mit */roster hide_offline* versteckt man alle Kontakte die nicht online sind, */roster show_offline* macht das wieder rückgängig. */add nick@server.com Name* fügt einen Kontakt hinzu. *Name* entspricht dabei dem Namen, der im Roster angezeigt wird.

Um eine Konversation zu starten, muß man zunächst den gewünschten Kontakt auswählen, indem man im Roster mit den Befehlen */roster up* und */roster down* oder den Tasten [Bild auf] und [Bild ab] nach oben und unten navigiert. Der aktuelle Kontakt wird von mcabber hervorgehoben. Nachrichten werden in die Eingabezeile getippt. Kommen Nachrichten von anderen Benutzern an, markiert mcabber den Kontakt rot. Wenn eine Nachricht von einem Benutzer eintrifft, mit dem man vorher noch nicht kommuniziert hat, muß man eventuell die Anzeige noch mit einem Druck auf [Enter] aktivieren. Bei zuviel Text im Chatfenster scrollt man mit [Strg]-[P] und [Strg]-[N], die Alternative sind die Befehle */buffer up 10* und */buffer down 10*. Die Zahl entspricht dabei der Anzahl der Zeilen, um die mcabber scrollen soll. Der eigene Status wird mit dem Befehl */status* gesetzt. Der erste Parameter ist der Status selber, möglich sind *online*, *avail*, *invisible*, *free*, *dnd*, *notavail* und *away*. Als zweiten Parameter gibt man bei Bedarf eine Statusmeldung an. Ein Beispiel:

```
/status notavail Ich bin nicht da
```

Zwar ist bei Jabber in den meisten Fällen die Verbindung zum Server verschlüsselt, theoretisch könnte jedoch der Betreiber des Servers die Nachrichten mitlesen. Außerdem ist nicht gewährleistet, daß die Verbindung zwischen Server und Chatpartner ebenfalls verschlüsselt ist.

Verschlüsselung

Deshalb bietet mcabber gleich zwei Möglichkeiten, eine End-to-End-Verschlüsselung aufzubauen, wobei die Nachricht beim Sender verschlüsselt und erst vom Empfänger wieder entschlüsselt wird. Die erste Möglichkeit ist eine Erweiterung des Jabber-Protokolls auf Basis des populären GnuPG. Falls man GnuPG sowieso schon für E-Mails anwendet und bereits im Besitz eines GnuPG-Schlüssels ist, ist das die bequemste Variante. Man erweitert lediglich die

```

mcabber - Konsole
[status] 05-07 14:14 <=> <Nebuk> even works with putty
...      05-07 14:14 <=> <Nebuk> is there a reason that this is
[C] mcabber@conf.lilotux.net not in man mcabber?
... CCC 05-07 14:16 <=> <franky> It's in the default rc file.
[n]      05-07 14:16 <=> <Nebuk> hmm
[o]      05-07 14:16 <=> <Nebuk> then i'm blind, sorry
        05-07 14:16 <=> <franky> No problem. :]
        05-08 07:46 *** The topic has been set to: MCabber 0.9.7
        released
        05-08 07:46 *** ~ The topic has been set to: MCabber
        0.9.7 released
        05-08 07:48 *** Room members:
        05-08 07:48 *** [o] McBot
        05-08 07:48 *** [o] ts52
        05-08 07:48 *** [o] Exxter
        05-08 07:48 *** [o] iNPUTmice
        05-08 07:48 *** [n] McKael_away
        05-08 07:48 *** Status message: AFK -- holiday
        05-08 07:48 *** [o] franky
-[C] Buddy: mcabber@conf.lilotux.net -- MCabber 0.9.7 released
[07:45:40] Error presence packet from <diozaka@jabber.org>
[07:45:40] Received error packet [remote-server-not-found]
[07:45:40] Error code from server: 404 Not Found
[07:45:56] Sent a join request to <mcabber@conf.lilotux.net>...
[07:46:08] You have joined mcabber@conf.lilotux.net as "iNPUTmice"
[o]
/room names

```

Bild 3: Multiuser-Chat

mcabber-Konfigurationsdatei um die folgenden Zeilen:

```
set pgp = 1
set pgp_private_key =
```

Als privater Schlüssel (Private Key) wird die Schlüssel-Kennung des eigenen GnuPG-Schlüssels eingetragen. In Zukunft fragt mcabber bei jedem Start nach dem GnuPG-Paßwort. Besitzt man keinen GnuPG-Schlüssel, kann man sich mit dem Programm `gpg --gen-key` einen Schlüssel erzeugen. Diese komplexere Angelegenheit soll aber nicht Teil dieses Beitrags sein, sie wurde schon ausführlich besprochen (siehe beispielsweise in der freeX 1'2004 ab Seite 44).

Mit `/pgp enable` beziehungsweise `/pgp disable` schaltet man nun GnuPG für jeden Kontakt an oder aus. Falls der jeweilige Kontakt seine Schlüssel-ID nicht in seinen Kontaktinformationen stehen hat, weist man mit `/pgp setkey .Schlüssel-ID` dem Kontakt einen Schlüssel zu.

Die zweite Möglichkeit für eine Verschlüsselung von Gesprächen mit mcabber ist OTR von <http://www.cypherpunks.ca/otr/>. Die Abkürzung »OTR« steht für Off the Record Messaging. Sie macht die Verschlüsselung unabhängig vom Protokoll. OTR kann

mit jedem Chatprotokoll benutzt werden, zum Beispiel ICQ, MSN und natürlich Jabber, nur der jeweilige Client muß OTR unterstützen. Indem man die Zeile

```
set otr = 1
```

zur Konfigurationsdatei hinzufügt, bringt man mcabber die Unterstützung für OTR bei. Außerdem muß noch der Ordner `.mcabber/otr/` angelegt werden, damit mcabber seinen automatisch generierten OTR-Schlüssel abspeichern kann. Mit dem Befehl `/otrpolicy default opportunistic` wird OTR in Zukunft automatisch eingeschaltet, wenn der Client des Chatpartners OTR versteht.

Alternativ kann man mit `/otrpolicy default manual` dieses Verhalten ausschalten und dann mit `/otr start` die Verschlüsselung für jeden Kontakt separat aktivieren.

Multiuser-Chat

Eines der besten Jabber-Features sind IRC-ähnliche Räume, in denen sich mehrere Teilnehmer treffen. mcabber enthält volle Unterstützung für diesen Multiuser-Chat (MUC). Jabber-Räume sind separate Dienste, die meistens unter der Domain `conference.jabber-server.tld` erreichbar sind.

Mit `/room join raumName@conference.jabber-server.tld` betritt man einen Raum. Existiert der Raum noch nicht, wird er aufgebaut. Als dritten Parameter gibt man optional einen Nick an. Wird keiner angegeben, nimmt mcabber den Usernamen aus der Konfigurationsdatei. Als vierter Parameter ist noch ein Paßwort für den Raum möglich. Der Raum wird als Eintrag im Roster dargestellt. Hat man den Raum betreten, fordert man mit `/room names` eine Liste aller Chatteilnehmer an. Mit `/room topic Neuer Text` ändert man das Topic, also die Kurzbeschreibung des Raums. Möchte man ein zuvor gesetztes Paßwort wieder entfernen, ruft man dafür den Befehl `/room unlock` auf. Ähnlich wie im IRC wirft man im MUC unerwünschte Personen mit `/room kick nickname` kurzfristig aus dem Raum. `/room ban nickname` sperrt die entsprechende Person endgültig aus. Jeden einzelnen Befehl von mcabber in hier zu erwähnen, macht keinen Sinn, wegen der ausführlichen deutschen Hilfe ist das auch nicht notwendig. Mehr Details offenbart außerdem die Manpage. Es lohnt sich auch, die Entwicklung vom mcabber im Auge zu behalten. Die momentan aktuelle Version 0.9.7 hat beispielsweise eine Möglichkeit implementiert, Befehle von außen an das laufende Programm zu schicken. Weitere Funktionen werden sicher folgen. ◆

