dovecot (IMAP) und postfix (SMTP) holen ihre Authentifikations-Infos aus der Datenbank sqlmail. In der Datenbank steht nicht das Passwort selbst, sondern sein Hash-Wert.

User anlegen

Verfügbare Password Hashes testen

```
doveadm pw -l
```

Empfohlener hash ist Blowfish (BLF-CRYPT). Die glibc in Ubuntu Server bis V. 18 konnte das nicht nicht (siehe https://bugs.launchpad.net/ubuntu/+source/linux/+bug/1349252). Das ist in Ubuntu 20 und GLIBC 2.31 gelöst.

Beispiele zum Erzeugen eines Hashes:

mit blowfish (blf-crypt)

```
doveadm pw -s BLF-CRYPT -u thomas.rother@miteinander-esslingen.de -p
XXXXXxxxxxx
```

mit SHA512

```
doveadm pw -s SHA512-CRYPT -u thomas.rother@miteinander-esslingen.de -p
XXXXXxxxxxx
```

Vom Ergebnis-String wird alles ab 6 bzw. $2y$ in die Datenbank geschrieben

Login-Test

```
doveadm pw -t '{SHA512-CRYPT}$hashash' -p "xxxxxxxxxxx" (verified)

doveadm pw -t '{BLF-CRYPT}$2y$hashash' -p "xxxxxxxxxxx" (verified)
```

Oder bei Fehler

```
Fatal: reverse password verification check failed: Password mismatch
```

Siehe auch
https://kaworu.ch/blog/2016/04/20/strong-crypt-scheme-with-dovecot-postfixadmin-and-roundcube/

Grundkonfiguration

Konfigurations-Optionen:

https://docs.kolab.org/administrator-guide/roundcube-settings-reference.html

Konfigurations-Empfehlungen: https://thomas-leister.de/mailserver-debian-stretch/

Authentifizierung

/etc/dovecot/conf.d/10-auth.conf definiert die zulässigen Authentifizierungs-Mechanismen Mit diesem Wert wird der Login über unverschlüsselte Verbindungen verhindert. Ausnahme: Client und Server mit gleicher IP.

```
disable_plaintext_auth = yes
```

Login Mechanismen:

```
auth_mechanisms = plain login
```

Hier wird definiert, ob über lokale userdb authentifiziert wird oder über SQL:

```
!include auth-sql.conf.ext #!include auth-ldap.conf.ext ## Hier
Authentifizierung über lokale userdb ## !include auth-passwdfile.conf.ext
```

über MySQL

Quelle: https://thomas-leister.de/internet/mailserver-ubuntu-server-dovecot-postfix-mysql/ (für Server 14.04) und https://thomas-leister.de/sicherer-mailserver-dovecot-postfix-virtuellen-benutzern-mysql-ubuntu-server-xenial/ (für Server 16.04)

In dovecot-sql.conf.ext wird der SQL query definiert.

Verbindungsaufbau

```
connect = host=127.0.0.1 dbname=userauth user=userauth password=XXXXX
```

Welcher Hash wird benutzt? Achtung: blowfish (BLF-CRYPT)) wäre theoretisch möglich, wird zur Zeit aber von glibc unter Ubunut nicht unterstützt.

```
default_pass_scheme = SHA512-CRYPT
```

Und hier der SELECT. Pro Login werden alle Varianblen über denselben SQL Request ermittelt ("userdb prefetch"):

```
# password query including userdb info in one request (prefetch)
password_query = \ SELECT userid AS user, password, \ home AS userdb_home,
uid AS userdb_uid, gid AS userdb_gid \ FROM users WHERE userid = '%n' AND
domain = 'netzwissen.de'
```

Wieso Port 587 und 143 mit STARTTLS?

(Zitat Thomas Leister)

Postfix verwendet Port 587 für die Verbindung zu MUAs (Mail User Agents - “Mailprogramme”), weil Port 25 nur für den Transfer der E-Mails zwischen den Servern zuständig sein soll. Während auf Port 25 von jedem Sender E-Mails ohne Authentifizierung empfangen werden können, wird auf Port 587

eine vorherige Authentifizierung des Endnutzers erzwungen. Port 587 wird daher auch als "Submission"-Port bezeichnet und ist üblicherweise in Firewalls freigeschaltet (siehe auch: RFC 6409 Submission). Port 25 hingegen wird beispielsweise in Unternehmensfirewalls und von manchen DSL-Routern blockiert, um das Spam-Problem einzudämmen. Aus einem solchen Netz können dann nur noch E-Mails zum zuständigen Mailserver gesendet werden - nicht mehr an jeden beliebigen Mailserver direkt. Da 587 als "universeller" Port (mit und ohne TLS-Verschlüsselung) definiert wurde, wird hier STARTTLS eingesetzt. Der ehem. "nur-TLS"-Port 465 ist nicht mehr als Standard festgelegt!

Für Dovecot verwende ich ebenfalls einen STARTTLS-Port. Dieser ist als Port 143 in RFC 3501 definiert. Prinzipiell ließe sich zwar als "TLS only" Port der weiterhin spezifizierte "imaps"-Port 993 verwenden, aber aus Gründen der Einheitlichkeit (und weil auch mit STARTTLS Verschlüsselung erzwungen werden kann) habe ich mich für 143 entschieden.

Kurz: Die Ports habe ich aus Abwägungen bezüglich der geltenden IANA-Standards und der Einheitlichkeit gewählt. Durch den Einsatz von STARTTLS entstehen (bei meiner Konfiguration) keine Nachteile.