MySQL 101
Wie man einen MySQL-Server am besten absichert

Simon Bailey
simon.bailey@uibk.ac.at
Version 1.1

23. Februar 2003

Change History

21. Jänner 2003: Version 1.0

23. Februar 2002: Version 1.1

- Diverse Korrekturen
- Änderung der Firewall Anweisungen
- `DELETE FROM db WHERE Db LIKE 'test%';`

Inhaltsverzeichnis

1 Einführung

MySQL ist ein Freeware Datenbanksystem, das recht schnell, zuverlässig und sicher läuft. Damit es aber wirklich abgesichert ist, muss man das Datenbanksystem nicht nur vor fernen Benutzern sondern auch vor lokalen Benutzern absichern. Dieses Merkblatt enthält alle Schritte, die notwendig sind, um dies zu erreichen.

2 Netzwerk-Absicherung

MySQL ist ein Datenbankserver, deshalb muss dieser vor unerwünschten Zugriffen über das Netzwerk abgesichert werden. Auf einem Linux-System erfolgt das mittels `ipchains` oder (seit dem 2.4-Kernel) mit `iptables`. MySQL hört auf TCP-Port 3306.

Wenn man von einem restriktiven Ruleset auf der Firewall ausgeht, muss in der Datei `etc/sysconfig/iptables` folgendes eintragen, um MySQL über das Netzwerk erreichbar zu machen[1]:

```
-A INPUT -p tcp -s 138.232.0.0/32 --dport 3306 -j ACCEPT
-A INPUT -p tcp --dport 3306 -j REJECT
```

Danach muss man mit `$$ service iptables restart` die Firewall neu laden.

Bezüglich Firewall Policies, siehe das Dokument Linux Firewalling im ZID-Web[2].

Diese Befehlsfolge hat die Auswirkung, dass alle Zugriffe auf Port 3306 aus dem Universitäts-Netzwerk zugelassen werden, alles andere wird zurückgewiesen.

[1] Diese Befehle gelten für einen RedHat Linux Host im Netzwerk der Universität, der `iptables` in einer Basis-Konfiguration verwendet.
Wenn man in RedHat eine vorgegebene Firewall Policy gewählt hat, dann ist dieser Schritt hinfällig

[2] http://www2.uibk.ac.at/zid/software/unix/linux/firewall.html

3 Zugriffsrechte

MySQL ist jetzt vor unerwünschten Zugriffen aus dem Netz geschützt; jetzt müssen wir noch den Datenbankserver selber absichern. MySQL bietet dazu ein sehr ausgeklügeltes Zugriffsmanagement[3], womit in jedem Datenbankschema die Zugriffsrechte für einzelne Benutzer bis auf Spaltenebene geregelt werden können.

Bei der Erstinstallation von jedem MySQL-Server sind zwei Datenbanken[4] vorhanden, `mysql` und `test`. In `mysql` werden die Zugriffsrechte administriert, die Datenbank `test` ist anfangs leer.

Es ist anfangs auch kein Passwort für den Benutzer `root` gesetzt. Das erste was man daher machen muss, ist ein Passwort für diesen Benutzer zu setzen:

```
$$ mysql -u root mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor. Commands end with ; or  g.
Your MySQL connection id is 1 to server version: 3.23.49

Type 'help;' or ' h' for help. Type ' c' to clear the buffer.

mysql> UPDATE user SET password=PASSWORD('new.password')
    -> WHERE user='root';
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql>
```

Jetzt müssen wir die leeren Benutzer-Einträge aus der MySQL Datenbank entfernen. Ansonsten ist es für jeden möglich, sich ohne Benutzernamen und Passwort am MySQL-Server zu validieren.

```
mysql> DELETE FROM user WHERE user='';
Query OK, 2 rows affected (0.00 sec)

mysql>
```

Wenn wir unsere Einträge kurz überprüfen, schaut die Benutzertabelle ähnlich aus wie diese:

```
mysql> SELECT Host, User, Password FROM user;

+-----------+------+------------------+
| Host      | User | Password         |
+-----------+------+------------------+
| localhost | root | 5efb737e09967912 |
| rp-c102   | root | 5efb737e09967912 |
+-----------+------+------------------+
2 rows in set (0.00 sec)

mysql>
```

[3] Oracle-Admins mögen mir das verzeihen!

[4] Hinweis für Oracle Benutzer:In *MySQL-Speak* ist eine Datenbank gleichzusetzen mit einem Schema in Oracle

In der Tabelle `db` müssen wir als nächstes aufräumen. Wenn wir uns einen Auszug aus dieser Tabelle kurz ansehen, sehen wir, wieso:

```
mysql> SELECT Host, Db, User, Select_priv, Drop_priv FROM db;

+------+---------+------+-------------+-----------+
| Host | Db      | User | Select_priv | Drop_priv |
+------+---------+------+-------------+-----------+
| %    | test    |      | Y           | Y         |
| %    | test\_% |      | Y           | Y         |
+------+---------+------+-------------+-----------+
2 rows in set (0.00 sec)

mysql>
```

Wie man hier sieht, kann jeder User (`user=''`) die `test`-Datenbank verwenden, und diese auch löschen. Jeder User hat die selben Rechte auf alle Tabellen, die mit `test_` beginnen. Das ist eher unerwünscht.

```
mysql> DROP database test;
Query OK, 0 rows affected (0.02 sec)

mysql> DELETE FROM db WHERE Db LIKE 'test%';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Da wir jetzt alle unerwünschte Einträge entfernt haben, müssen wir MySQL nur noch anweisen, diese Zugriffsrechte anzunehmen:

```
mysql> FLUSH privileges;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

4 Laufende Administration

Im laufenden Betrieb empfiehlt es sich, für jede Datenbank einen eigenen Benutzer anzulegen. Die Schritte für das Anlegen einer Datenbank sind dann wie folgt:

1. Anlegen der Datenbank:

```
mysql> CREATE DATABASE new;
Query OK, 1 row affected (0.03 sec)

mysql>
```

2. Anlegen des Benutzers:

```
mysql> INSERT INTO user (Host, User, Password)
    -> VALUES ('localhost', 'newUser', PASSWORD('newPassword'));
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO user (Host, User, Password)
    -> VALUES ('host', 'newUser', PASSWORD('newPassword'));
Query OK, 1 row affected (0.05 sec)

mysql>
```

   Mit *host* ist der Kurzname des Rechners gemeint, also im Falle von `myhost.uibk.ac.at` wäre das `myhost`. In manchen Fällen empfiehlt es sich sogar die FQDN des Hosts einzutragen.

3. Zuweisen der Rechte für den Benutzer:

```
mysql> INSERT INTO db (Host, Db, User, Select_priv, Insert_priv, Update_priv,
    -> Delete_priv, Create_priv, Drop_priv, Index_priv, Alter_priv) VALUES
    -> ('%', 'new', 'newUser', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
    -> 'Y', 'Y');
Query OK, 1 row affected (0.00 sec)

mysql>
```

4. Neuinitialisierung der Rechtetabelle des Servers:

```
mysql> FLUSH privileges;
Query OK, 0 rows affected (0.01 sec)

mysql>
```