

Datenbank-Server: MySQL für externen Zugriff konfigurieren

Problem:

Auf die MySQL-Datenbank soll nicht nur vom localhost per Script oder phpMyAdmin zugegriffen werden, sondern auch von extern. Z.B. weil man einen eigenen MySQL-Client nutzt, oder die Verbindung mit einem anderen Server haben will, etc.

Die Lösung ist zwei geteilt:

- 1.) Öffnen von MySQL für externen Zugriff.
- 2.) User-Berechtigungen so setzen, daß MySQL auch deren externe Logins annimmt.

Lösung:

1. Teil (Konfiguration von MySQL):

Es gibt inzwischen zwei Fälle. Beide werden in der `my.cnf` in der Sektion `[mysqld]` konfiguriert.

a) Bei älteren MySQL-Versionen war es der Parameter `SkipNetworking`. Wenn dieser aktiviert ist, lauscht MySQL lediglich an den lokalen Sockets. Also muß diese auskommentiert werden:

```
#SkipNetworking
```

b) Neuere Versionen bringen den Parameter `bind-address` mit. Damit kann man eine IP angeben, an der gelauscht wird, bzw. den Parameter deaktivieren. Also eine der folgenden Einstellungen wählen:

```
bind-address 0.0.0.0      # deaktiviert
bind-address 127.0.0.1    # nur localhost
bind-address 192.168.0.100 # nur über die locale IP
                        # (sinnvoll innerhalb einer DMZ)
```

Sollte die `my.cnf` verändert worden sein, so muß auch der MySQL-Daemon davon in Kenntnis gesetzt werden:

```
/etc/init.d/mysql restart
```

Testen ob der Zugriff von Extern funktioniert:

Datenbank-Server: MySQL für externen Zugriff konfigurieren

In eine DOS-Box oder beliebig anderen externen Server folgendes eintippen:

```
telnet meine-domain.tld 3306
```

Sollte ein [Verbinden fehlgeschlagen](#) kommen, so stimmt hier noch etwas nicht. (Z.B. Firewall, Router, etc.). Bei Erfolg kommt ein häßliches Zeichenwirrwar. Einfach paarmal auf die Enter-Taste tippen und Ihr seid wieder draussen.

2. Teil (User-Berechtigung):

Nun müssen die User dafür frei geschaltet werden. Anwender von Confixx oder Plesk können dies einfach in der Admin-Oberfläche erledigen. Per Hand sieht es (anhand des Beispiel-DB-Users 'huschi') so aus:

```
#In der Shell/Putty:
mysql -uroot -p mysql
#root-Passwort eingeben
#nun sind wir im MySQL-Client, der nur noch SQL-Befehle versteht:
update user set host='%' where user='huschi';
update db set host='%' where user='huschi';
flush privileges;
exit;
```

Dies kann man natürlich auch per phpMyAdmin erledigen.

Wichtig zu wissen:

Die Tabelle `mysql.user` ist zuständig für die allgemeinen Benutzerrechte.

Und `mysql.db` regelt den Zugriff auf spezielle Datenbanken.

Als weiterer Tip: nimmt niemals den User `root` oder unter Plesk `admin` für den externen Zugang!

Warum: a) Sie sind leicht bei BruteForce-Attacken zu erraten und b) wer diesen Zugang knackt hat gleich alle Rechte die man unter MySQL nur haben kann.

Fehlersuche:

An die Fehlersuche geht man einfach logisch ran:

- Lauscht `mysqld` auch extern am richtigen Port?

Datenbank-Server: MySQL für externen Zugriff konfigurieren

Dazu nutzt man `netstat` auf dem Server und/oder `telnet MEINE.IP 3306` von einem externen Computer.

- Kann sich der neu erstellte User einloggen?

Testen auf dem Server mit `mysql -u USER -p DATENBANK.`

Das Selbe von einem externen Rechner mit installiertem MySQL(-Client).

Eindeutige ID: #1122
huschi
2008-01-02 08:50