MySQL Security für Oracle DBAs

DOAG Konferenz 2016, Nürnberg

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com

Über FromDual GmbH

www.fromdual.com

Support

Beratung

remote-DBA

Schulung

Inhalt

MySQL Security

- MySQL User Konzept
- Port und Socket
- Sicherheitsprobleme
- MySQL Härten
- Passwort Validierung
- Passwort im File
- Client/Server Verschlüsselung (SSL)
- Upgrade
- Account Management
- Sicherheitsrelevante Variablen
- Verschlüsselung

Grundlegende Massnahmen

- Quelle: Kapitel 7 Security
  - http://dev.mysql.com/doc/refman/5.7/en/security.html
- MySQL 5.7 „secure by default“ !!!
- Grundlegende Massnahmen:
  - MySQL `root` Password (`sys as sysdba`)
  - Root von remote (`root@%`)
  - Anonymous user (`''@localhost`)
  - Test Schema
  - Passwort-Stärken Prüfung

MySQL User - Konzept

- Account = User @ {Host | Domain}
- Beispiele:
  - `'root'@'localhost'`
  - `'root'@'127.0.0.1'`
  - `'root'@'192.168.1.1'`
  - `'root'@'%'`
- Host: `127.0.0.1` != `localhost`
  - `127.0.0.1` – lokales TCP/IP Interface
  - `localhost` – lokaler UNIX File-Socket
- Host: Quell von der ich kommen darf
- `root` in MySQL = `sys as sysdba` in Oracle

Port und Socket

- Port und Socket („Listener“)

```
shell> lsof -p 12890
COMMAND   PID  USER FD   TYPE DEVICE  NODE NAME
...
mysqld  12890 mysql 15u IPv4  31258   TCP *:3306 (LISTEN)
mysqld  12890 mysql 17u unix 0x0000 31259 /tmp/mysql.sock
...
```

- Wir haben keine „Listener“ Prozess wie Oracle.

User und aktueller User I

- Über UNIX File-Socket

```
shell> mysql --user=root # --socket=/tmp/mysql.sock
mysql> status;
...
Current user:                   root@localhost
Connection:                     Localhost via UNIX socket
UNIX socket:                    /tmp/mysql.sock
...
mysql> SELECT USER(), CURRENT_USER();
+----------------+----------------+
| USER()         | CURRENT_USER() |
+----------------+----------------+
| root@localhost | root@localhost |
+----------------+----------------+
```

- Ohne „langsamen“ TCP/IP Stack

User und aktueller User II

- Über lokalen TCP/IP-Socket

```
shell> mysql --user=root --host=127.0.0.1
mysql> status;
...
Current user:                   r
Connection:                     1
TCP port:                       3
...
mysql> SELECT USER(), CURR      _USER();
+----------------+---------------+
| USER()         | CURRENT_USER() |
+----------------+---------------+
| root@localhost | root@localhost | --> falsch! DNS lookup
+----------------+---------------+
```

"falsch" oder zumindest verwirrend!
Müsste sein: root@127.0.0.1
DNS Lookup

- Ohne „langsames" TCP/IP Netzwerk

User und aktueller User III

- Über remote TCP/IP-Socket

```
shell> mysql --user=root --host=192.168.1.35
mysql> status;
...
Current user:           root@chef.rebenweg
Connection:             192.168.1.35 via TCP/IP
TCP port:               3306
...
mysql> SELECT USER(), CURRENT_USER();
+--------------------+----------------+
| USER()             | CURRENT_USER() |
+--------------------+----------------+
| root@chef.rebenweg | root@%         |
+--------------------+----------------+
```

- Mit „langsamem“ TCP/IP Netzwerk

Zurück zur Sicherheit

- Root Password
- Root von remote
- Anonymous user

```
mysql> SELECT user, host, password FROM mysql.user;
mysql> -- 5.7: SELECT user, host, authentication_string
                  FROM mysql.user;
+-------------+-----------+-------------------------------+
| user        | host      | password                      |
+-------------+-----------+-------------------------------+
| root        | localhost | ______________________________ |
| root        | master    | ______________________________ |
| root        | 127.0.0.1 | ______________________________ |
| root        | ::1       | ______________________________ |
| ____________ | localhost | ______________________________ |
| ____________ | master    | ______________________________ |
+-------------+-----------+-------------------------------+
```

Anonymous User + Test Schema

- **Warum ist der Anonymous User heikel?**

```
mysql> SHOW GRANTS FOR ''@localhost;
+---------------------------------------+
| Grants for @localhost                 |
+---------------------------------------+
| GRANT USAGE ON *.* TO ''@'localhost'  |
+---------------------------------------+
mysql> SELECT * FROM mysql.db\G
                  Host: %
                    Db: test\_%
                  User:
           Select_priv: Y
           Insert_priv: Y
           Update_priv: Y
           Delete_priv: Y
           Create_priv: Y
...
```

MySQL 5.7

- MySQL 5.7 ist „secure by default“
  - d.h. Härtung ist nach Installation bereits gelaufen:
    - `shell> mysqld --initizialize ...`
  - Altes („unsicheres“) intialisieren:
    - `shell> mysqld --initizialize-insecure ...`
  - Root Password im MySQL error log (oder `STDOUT`):

```
shell> grep password error.log
[Note] A temporary password is generated for \
       root@localhost: pTJ(YNrrf4od
```

MySQL härten (5.6 und 5.7)

- Entweder
  - Von Hand (`DROP USER`, `ALTER USER`, etc...)
  - Oder `mysql_secure_installation`

```
shell> mysql_secure_installation --user=root
Enter current password for root (enter for none):
OK, successfully used password, moving on...
Set root password? [Y/n] Y
Remove anonymous users? [Y/n] y
 ... Success!
Disallow root login remotely? [Y/n] y
 ... Success!
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
Reload privilege tables now? [Y/n] y
 ... Success!
```

Weitere Massnahmen

- Keine `Shutdown` oder `Super` Privilegien (~~`ALL`~~ ausser `root`)

```
mysql> SELECT user, host, super_priv, shutdown_priv
  FROM mysql.user WHERE super_priv = 'Y' OR shutdown_priv ='Y';
+-------------+-----------+------------+---------------+
| user        | host      | super_priv | shutdown_priv |
+-------------+-----------+------------+---------------+
| root        | localhost | Y          | Y             |
| replication | master    | Y          | Y             |
+-------------+-----------+------------+---------------+
```

- Kein Zugriff auf `mysql` Schema (ausser `root`)

```
mysql> SELECT user, host, db, table_name, table_priv FROM mysql.tables_priv;
mysql> -- Check also: mysql.db, mysql.columns.priv!
+------+------+-------+-------------+------------+
| user | host | db    | table_name  | table_priv |
+------+------+-------+-------------+------------+
| spy  | %    | mysql | tables_priv | Select     |
+------+------+-------+-------------+------------+
```

Password-Stärken-Validierung

- Installation
  - Plug-in seit MySQL 5.6
  - Default in MySQL 5.7
  - Mittels `mysql_secure_installation`

```
mysql> SELECT plugin_name, plugin_status, plugin_license
  FROM information_schema.plugins
 WHERE plugin_name LIKE 'validate_password%';
+-------------------+--------+---------+
| plugin_name       | status | license |
+-------------------+--------+---------+
| validate_password | ACTIVE | GPL     |
+-------------------+--------+---------+

mysql> INSTALL PLUGIN validate_password
 SONAME 'validate_password.so';
```

Konfiguration

- `LOW` – Länge ≥ 8
- `MEDIUM` – Länge ≥ 8, Numerisch, Gross/Klein und Sonderzeichen
- `STRONG` – Länge ≥ 8, Numerisch, Gross/Klein, Sonderzeichen und Dictionary

```
mysql> SHOW GLOBAL VARIABLES LIKE 'validate_password%';
+--------------------------------------+--------+
| Variable_name                        | Value  |
+--------------------------------------+--------+
| validate_password_dictionary_file    |        |
| validate_password_length             | 8      |
| validate_password_mixed_case_count   | 1      |
| validate_password_number_count       | 1      |
| validate_password_policy             | MEDIUM |
| validate_password_special_char_count | 1      |
+--------------------------------------+--------+
```

Passwort in File

- Unverschlüsselt
  - `~/.my.cnf`
  - aber bitte mit `chmod 600`!!!

```
#
# ~/.my.cnf
#
[client]
user     = root
password = secret

shell> mysql   # ohne nix!
```

Passwort in File (verschlüsselt)

- Verschlüsselt (aber knackbar!)
  - `~/.mylogin.cnf`

```
shell> mysql_config_editor set --login-path=mysqld1 \
       --host=localhost --user=root --password

shell> mysql_config_editor print --all
[mysqld1]
user = root
password = *****
host = localhost

shell> mysql --login-path=mysqld1   # ohne nix!
```

Client/Server Verschlüsselung

- Früher: Passwort im Klartext!
- Dann: Passwort gehasht, SSL optional
- MySQL 5.7 SSL per default

```
shell> mysql --user=root --host=127.0.0.1 --ssl-mode=REQUIRED

mysql> SHOW SESSION STATUS LIKE 'Ssl_%';
+---------------+---------------------+
| Variable_name | Value               |
+---------------+---------------------+
| Ssl_cipher    | DHE-RSA-AES256-SHA  |
| Ssl_version   | TLSv1.2             |
+---------------+---------------------+

mysql> SHOW GLOBAL VARIABLES LIKE 'tls_version';
+---------------+-----------------------+
| Variable_name | Value                 |
+---------------+-----------------------+
| tls_version   | TLSv1,TLSv1.1,TLSv1.2 | (EE)
+---------------+-----------------------+
```

Schlüssel erstellen (5.7)


```
shell> mysql_ssl_rsa_setup --datadir=/var/lib/mysql
shell> ll *.pem
-rw------- 1 mysql mysql 1675 Nov  7 17:46 ca-key.pem
-rw-r--r-- 1 mysql mysql 1074 Nov  7 17:46 ca.pem
-rw-r--r-- 1 mysql mysql 1078 Nov  7 17:46 client-cert.pem
-rw------- 1 mysql mysql 1675 Nov  7 17:46 client-key.pem
-rw------- 1 mysql mysql 1679 Nov  7 17:46 private_key.pem
-rw-r--r-- 1 mysql mysql  451 Nov  7 17:46 public_key.pem
-rw-r--r-- 1 mysql mysql 1078 Nov  7 17:46 server-cert.pem
-rw------- 1 mysql mysql 1675 Nov  7 17:46 server-key.pem

mysql> SHOW GLOBAL VARIABLES LIKE 'ssl%';
+---------------+-----------------+
| Variable_name | Value           |
+---------------+-----------------+
| ssl_ca        | ca.pem          | Eingenes CA Zertifikat
| ssl_capath    |                 |
| ssl_cert      | server-cert.pem | Server Zertifikat
| ssl_key       | server-key.pem  | Server Priave Key
+---------------+-----------------+
```

Upgrade

- Minor Version 5.7.12 -> 5.7.15 (Patch)
- Major Version 5.6.31 -> 5.7.16 (Upgrade)
- In MySQL beides gleich (aufwändig):
  - DB Backupen und Stoppen
  - Binaries austauschen
  - DB Starten
  - `shell> mysql_upgrade`
  - Alles (ohne Backup) dauert ca. 15 Minuten

Upgrade Info

- Oracle CPU (4 x pro Jahr: Jan, Apr, Jul, Okt)

Oracle MySQL Risk Matrix

CVE#	Component	Sub-component	Protocol	Remote Exploit without Auth.?	CVSS VERSION 3.0 RISK (see Risk Matrix Definitions)									Supported Versions Affected
					Base Score	Attack Vector	Attack Complex	Privs Req'd	User Interact	Scope	Confid-entiality	Inte-grity	Avail-ability	
CVE-2016-630	MySQL Server	Server: Security: Encryption	MySQL Protocol	Yes	7.5	Network	Low	None	None	Un-changed	None	None	High	5.6.33 and earlier, 5.7.15 and earlier
CVE-2016-6662	MySQL Server	Server: Logging	None	No	7.2	Local	High	High	Required	Changed	High	High	High	5.5.52 and earlier, 5.6.33 and earlier, 5.7.15 and earlier
CVE-2016-5617	MySQL Server	Server: Error Handling	None	No	7.0	Local	High	Low	None	Un-changed	High	High	High	5.5.51 and earlier, 5.6.32 and earlier, 5.7.14 and earlier

- MySQL Change History
  - http://dev.mysql.com/doc/relnotes/mysql/5.7/en/
- What is new in MySQL 5.7
  - http://dev.mysql.com/doc/refman/5.7/en/mysql-nutshell.html
- Übersicht FromDual:
  - http://fromdual.com/security

MySQL Account Management

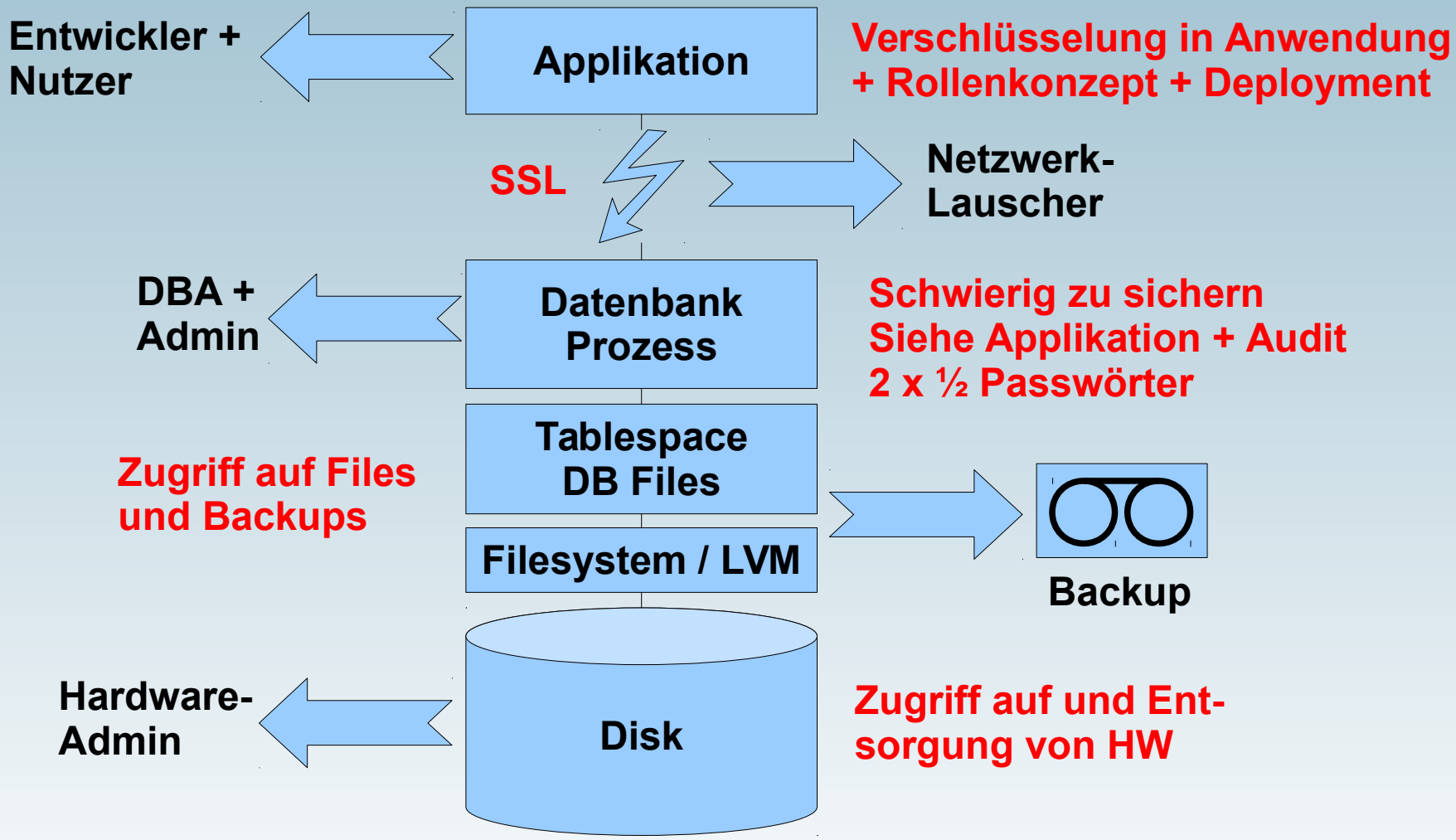
- Seit MySQL 5.7:
  - Account sperren
  - Password Verfallszeit

```
mysql> ALTER USER 'oli'@'localhost' ACCOUNT LOCK;

mysql> ALTER USER 'quack'@'%' PASSWORD EXPIRE INTERVAL 15 DAY;

mysql> SELECT user, host, password_expired AS expired
     , LEFT(password_last_changed, 10) AS last_changed
     , password_lifetime AS lifetime, account_locked AS locked
  FROM mysql.user;
+-------------+-----------+---------+--------------+----------+--------+
| user        | host      | expired | last_changed | lifetime | locked |
+-------------+-----------+---------+--------------+----------+--------+
| root        | localhost | N       | 2015-12-16   |     NULL | N      |
| focmm       | master    | Y       | 2015-12-18   |     NULL | N      |
| oli         | localhost | N       | 2016-08-05   |     NULL | Y      |
| quack       | %         | N       | 2016-09-04   |       15 | N      |
+-------------+-----------+---------+--------------+----------+--------+
```

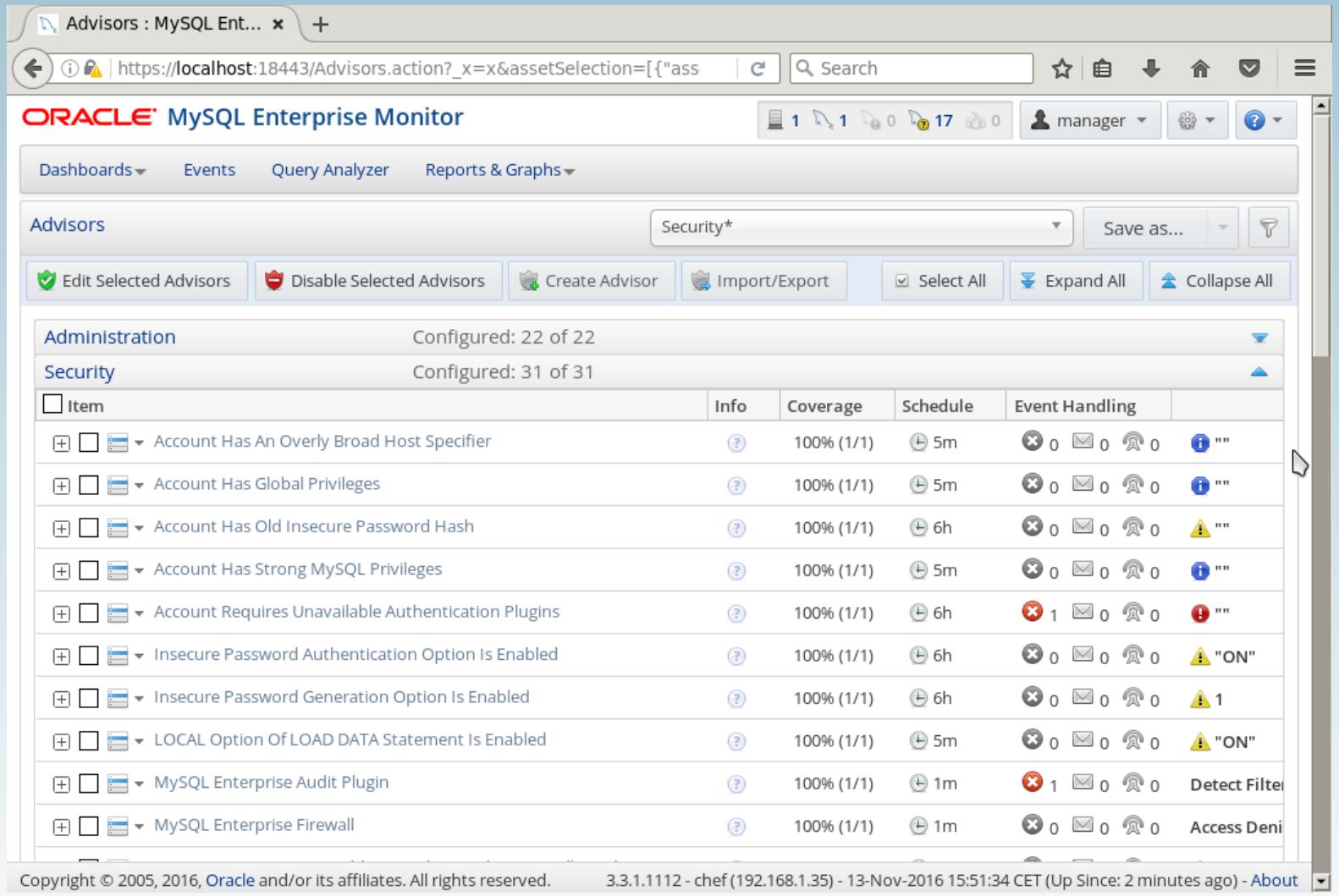
Sicherheitsrelevante Variablen

- Netzwerk Zugriff deaktivieren (nur noch lokaler Zugriff)
  - `skip_networking`
- Namensauflösung deaktivieren (DNS)
  - `skip_name_resolve`
- Authentisierung deaktivieren (!!!)
  - `skip_grant_tables`
  - Braucht man um `root` zurückzusetzen
- Dateien auf Server lesen/schreiben (`LOAD DATA INFILE`)
  - `local_infile`
- Dateien nur aus „sicheren“ Ordnern auf Server lesen
  - `secure_file_priv`
- http://dev.mysql.com/doc/refman/5.7/en/security-options.html

Verschlüsselung



MySQL Enterprise Monitor



Weitere Features

- MySQL Enterprise Firewall
  - SQL Firewall gegen SQL Injections
- MySQL Enterprise Audit
  - Auditing gegen Audit Vault (PCI, HIPAA, FERPA, SOX)
- MySQL Enterprise Authentication
  - LDAP/AD, Kerberos, Windows Authentication
- MySQL Enterprise Encryption
  - TDE, sha256, etc.
- MySQL Enterprise Backup
  - Physikalisches Backup für grosse Datenbanken

Q & A

Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
  - **Beratung**
  - **Remote-DBA**
  - **Support für MySQL, Galera, Percona Server und MariaDB**
  - **Schulung**

www.fromdual.com/presentations