MySQL sicher aufsetzen und betreiben

SLAC 2015, Berlin

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com

FromDual GmbH

www.fromdual.com

Support

Beratung

remote-DBA

Schulung

Inhalt

MySQL sicher aufsetzen und betreiben

- Vorbereitungen
- Installation
- Härten
- Upgrade
- Konfiguration
- User Management
- Verschlüsselung
- Backup / Restore
- Hochverfügbarkeit
- Monitoring
- Wo lauert das Böse?
- Angriffsvektoren

Vorbereitung Installation

- Welches O/S, welche Distro?
  - Windows kann heute auch sicher sein...
  - Was könnt Ihr am besten?
- Welcher Branch/Fork?
  - MySQL, MariaDB, Percona, Galera
- Packet, Binary Tarball, oder ~~Source~~
- DEB/RPM von Distribution oder Hersteller
  - Distribution oft veraltet
  - und wie gut gepatched?
  - und wenn gepachted, wie gut getestet?
- Version
  - MySQL: 5.5, 5.6, (5.7), MariaDB 5.5, 10.0, (10.1)

Installation

- Durch wen?
  - Pakete
  - selber (`mysql_install_db`)
- Wohin?
  - `/var/lib/mysql`
  - `/mount/mysql/data`
- Advanced Security macht ärger!
  - AppArmor
  - SElinux
- weitere potentielle Problemchen
  - `/etc/mysql/conf.d/debian.cnf` (`root` äquivalent)
  - Syslog auf Debian/Ubuntu: die Log-Informationen sind auf nimmer wiedersehen weg, nicht theoretisch sonder praktisch!

Was passiert bei Installation?

- InnoDB Tablespace und Log Files
- MySQL Data Dictionary (`mysql` Schema)
  - Kreiert User (`root`, `anonymous`)
  - Kreiert `test` Schema

slac1

- Was ist das Problem?
- Härtet jemand von Euch nach Installation?

Härten von MySQL

- **Wie:**
  - **`mysql_secure_installation`**
  - **leider „kaputt“**
- **Was:**
  - **`root` Passwort**
  - **`root` von remote**
  - **`test` Schema**
  - **anonymous User (`''@localhost`)**

slac1

Warum härten?

- Root Passwort setzen ist klar?
  - Nein mir nicht!
  - Wenn niemand lokal zugriff hat ausser `root` braucht es dort auch kein Passwort.
- Root von remote ist klar?
  - Sogar mir!
- Warum ist der anonymous User (`''@server`) böse?

slac2

- Warum ist das `test` Schema böse?

- Insbesondere Hoster/SaaS und Ähnliche aufpassen!!!

Neues Mätzchen mit 5.6

- Seit MySQL 5.6 gilt:
  - Passwort auf Kommandozeile ist böse, warum?

```
shell> mysql --user=sicher --password=secret

Warning: Using a password on the command line
Interface can be insecure.
```

  - Kann mir jemand sagen warum?

- Früher `~/.my.cnf` (`chmod 0600`)
  - Scheint heute immer noch sicher!

Heute: MySQL Config Editor

- **Seit MySQL 5.6 gilt:**

```
shell> mysql_config_editor set
--login-path=slac2
--host=localhost --user=sicher –-password

shell> mysql --login-path=slac2
```

  - **Wieso geht das jetzt???**

  - **Fortsetzung folgt...**

Und wie cracken?

The encryption used by mysql_config_editor **prevents passwords from appearing in .mylogin.cnf as cleartext** and **provides a measure of security by preventing inadvertent password exposure**. For example, if you display a regular unencrypted my.cnf option file on the screen, any passwords it contains are visible for anyone to see. With .mylogin.cnf, that is not true.
But the encryption used will not deter a determined attacker and you should not consider it unbreakable. A user who can gain system administration privileges on your machine to access your files could decrypt the .mylogin.cnf file with some effort.

- **Security by Obscurity???**
  - **Sicherheitsgewinn?**
- **Und jetzt wie cracken? :-)**

Weitere Sicherheitsfeatures

- MySQL CLI: Filter auf password in History
  - OK. Seh ich ja ein, ist aber mühsam:
  - `grep password ~/.mysql_history`
- Installation in 5.7 automatisch sicher
  - Kein `test` Schema mehr
  - Kein anonymous User mehr
  - Kein `root` von Remote mehr
  - UND: `root@localhost` hat default Passwort, welches geändert werden muss...

Upgrade / Release Cycles?

- Major Releases:
  - MySQL 5.1, 5.5, 5.6, 5.7 (ca. alle 2 Jahre)
  - MariaDB 5.1, 5.2, 5.3, 5.5, 10.0, 10.1 (ca. alle 18 Monate)
- Minor Releases:
  - 5.6.x
  - MySQL ca. 6 pro Jahr (also alle 2 Monate)
  - MariaDB ca. 6 pro Jahr (also alle 2 Monate)
- Und jetzt, alle 2 Monate ein Upgrade?
  - Ja, warum nicht?
  - Wir haben Testautomation und
  - automatisierte Deployments (siehe andere Vorträge...)

Upgrade in der Praxis

- Oracle liefert Critical Patch Updates (CPU)
  - (4 pro Quartal, alle 3 Monate)
- Distributionen ähnliche Zyklen?
  - ziehen die wirklich alles nach?
  - Ubuntu Security Notes (USN): alle 3 Monate
  - RHEL/CentOS?
- potentielle Probleme:
  - Wer testet wie gut (Distribution / Betreiber)?
  - Upgrade erfolgt automatisch
  - Restart der DB-Instanz (Red Hat/CentOS)
- http://fromdual.com/security

Upgrade wie tun:

- Major Release NICHT überspringen!
  - daher mindestens alle 2 Jahre ein upgrade
- Offizielle Methode:
  - Dump/Restore (aber wie mit meinen 5 Tbyte?)
- Geht meist auch:
  - Binary Upgrade
  - ab 5.7 offiziell supportet!!!
- Anschliessend NICHT vergessen:
  - `mysql_upgrade`

Warum kein Upgrade böse?

- **Darum:**  slac4

```
ERROR 2013 (HY000): Lost connection to MySQL server
during query
```

- **Oracle hat Security relevante Bugs heute versteckt...**
- **Aber, der Findige lässt sich nicht unterkriegen:** https://bugzilla.redhat.com/show_bug.cgi?id=919247

  - **Ich habe keine 15 Minuten gebraucht...**

MySQL Crash


```
150620 14:59:27 [ERROR] mysqld got signal 11 ;
This could be because you hit a bug...

Thread pointer: 0x47e10e0
Attempting backtrace. You can use the following information to find out
where mysqld died. If you see no messages after this, something went
terribly wrong...
stack_bottom = 0x7f8949817ea8 thread_stack 0x48000
mysqld(my_print_stacktrace+0x2e) [0x9c493e]
mysqld(handle_segfault+0x3b3) [0x611eb3]
/lib/x86_64-linux-gnu/libpthread.so.0(+0x10340) [0x7f8948a70340]
mysqld() [0x75aaf0]
mysqld(Geometry::append_points(String*, unsigned int, char const*, unsigned int) const+0x4e)
mysqld(Gis_polygon::get_data_as_wkt(String*, char const**) const+0xbb) [0x75c68b]
mysqld(Item_func_as_wkt::val_str(String*)+0x145) [0x5d6f55]
...
mysqld(mysql_execute_command(THD*)+0x2d4e) [0x62066e]
mysqld(mysql_parse(THD*, char*, unsigned int, char const**)+0x299) [0x623ff9]
mysqld(dispatch_command(enum_server_command, THD*, char*, unsigned int)+0xb9b) [0x624b9b]
mysqld(do_command(THD*)+0x101) [0x6255b1]
mysqld(handle_one_connection+0xdf) [0x61690f]
/lib/x86_64-linux-gnu/libpthread.so.0(+0x8182) [0x7f8948a68182]
/lib/x86_64-linux-gnu/libc.so.6(clone+0x6d) [0x7f8947f8347d]

Trying to get some variables.
Some pointers may be invalid and cause the dump to abort.
Query (0x7f88cc004b98): select astext(0x010000000003000000010000000000000010)
Connection ID (thread ID): 1
Status: NOT_KILLED
```

- **Crash ist IMMER böse = IMMER ein Bug**

Jeder hat seine Leidenschaft

- Es gibt Leute, die sammeln Briefmarken...
- Andere sammeln so was:

```
#
# Bug #68591: Geometry query crashes mysqld
# http://bugs.mysql.com/bug.php?id=68591
# https://mariadb.atlassian.net/browse/MDEV-4252
# https://bugzilla.redhat.com/show_bug.cgi?id=919247
#
# Fixed in: MySQL 5.6.12, 5.5.32, 5.1.70
#           MariaDB 5.5.30, 5.3.13, 5.2.15, 5.1.73
# Happens in: before
# Does not happen in: -
# Public since 2013-03
#

SELECT ASTEXT(0x0100000000030000001000000000000010);
```

Konfiguration Sicherheit

Name	Scope	Dynamic
allow_suspicious_udfs	Global	No
automatic_sp_privileges	Global	Yes
chroot	Global	No
des_key_file	Global	No
local_infile	Global	Yes
old_passwords	Both	Yes
safe_user_create	Global	Yes
secure_auth	Global	Yes
secure_file_priv	Global	No
skip_grant_tables	Global	No
skip_name_resolve	Global	No
skip_networking	Global	No
skip_show_database	Global	No

Konfiguration Ressourcen

- Schlechte Performance
  - DoS
- Unterallokation
  - Schlechte Performance (I/O Sättigung)
- Überallokation
  - Swapping (slow)
  - kill/crash (32-bit)
  - oom-killer (service outage)
  - Einfluss auf andere (VM, noisy neighbours)
- Kosten
  - Zu fette Hardware

User Management

- Übersicht

```
SELECT user, host, password FROM mysql.user;

SHOW GRANTS FOR ''@localhost;
```

- Host:
  - `skip_name_resolve` wenn DNS nicht getraut wird oder instabil ist
  - IP address spoofing (soll nicht ganz einfach sein)?
- `CREATE USER, DROP USER`
- Abgelegt unter: `mysql.user`
- Anonymous User kann immer verbinden (`USAGE`), in Kombination mit `test` Schema, siehe oben.

Objekt Privilegien

- Was sind Objekte:
  - Tabellen, Indices, Views, Procedures, Functions, Triggers, Events, Temporäre Tabellen
- Wer sollte was dürfen?
  - read-only User:
    - `SELECT, SHOW DATABASES`
  - read-write User:
    - `UPDATE INSERT DELETE CREATE TEMPORARY TABLES, LOCK TABLES`
  - Schema Owner:
    - `ALTER ROUTINE, CREATE ROUTINE, EXECUTE, CREATE VIEW, SHOW VIEW, TRIGGER, INDEX, ALTER, EVENT, REFERENCES, DROP, CREATE`
- Privilegien-Hierarchie: Global, Schema, Tabelle/Routine, Spalte
- `GRANT ... ON schema.table`
- `GRANT ... ON mysql.xxx` vermeiden!
- Liegen unter `mysql.{user, db, table_privs, procs_priv, column_privs)`

Globale/System Privilegien

- Globale Privilegien:
  - `ALL, USAGE, SUPER, SHUTDOWN, REPLICATION SLAVE, REPLICATION CLIENT, RELOAD, PROXY, PROCESS, CREATE USER, CREATE TABLESPACE, FILE, GRANT OPTION`
- Liegen in `mysql.user`
- Was lässt sich so anstellen mit:
  - `ALL` --> alles :-)
  - `USAGE` --> verbinden, siehe oben
  - `SUPER` --> `CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL`, Daten änderungen (`read_only`) Replikation start/stop, Log ein/ausschalten, Definer in Stored Programs ändern (Möglichkeit Privilegien aufzubohren?)
  - REPLICATION SLAVE --> Binary Logs remote lesen
  - PROCESS --> Anderer User Prozesse/Statements einsehen
  - GRANT OPTION --> Anderen Usern eigene Rechte geben
  - FILE --> Jedes beliebige File auf dem DB-Server lesen (welches `mysql` User lesen kann)
- Beispiel `FILE`

Rollen / externe Authentifizierung

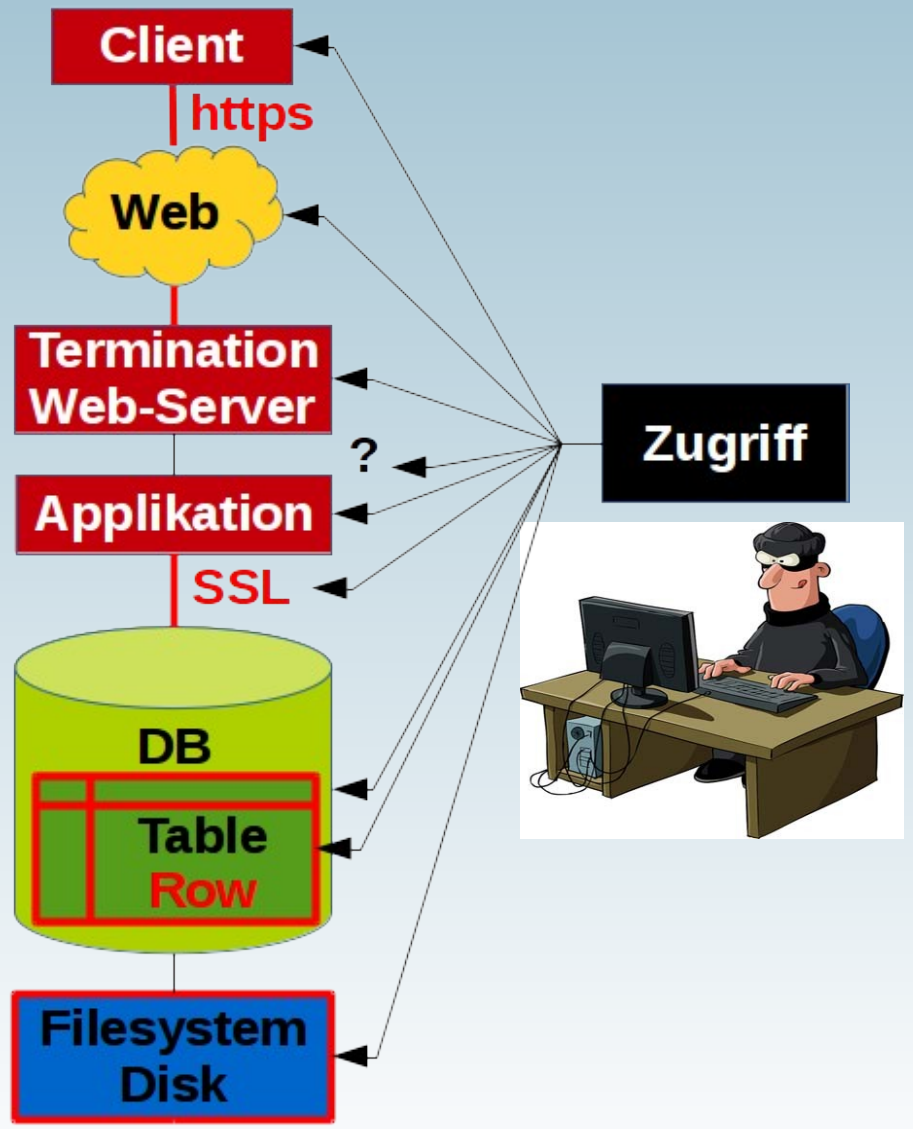
- MySQL kann native keine Rollen
  - Nicht so tragisch, Rollen heute meist in Applikation abgebildet
- Rolen ab MySQL 5.7.7 mit "expanded Proxy User Support"
  - http://mysqlblog.fivefarmers.com/2015/04/08/emulating-roles-with-expanded-proxy-user-support-in-5-7-7/
- MariaDB (10.0.5):
  - `CREATE ROLE myrole;`
  - `GRANT ... ON *.* to myrole;`
  - `GRANT myrole to oli;`
- MySQL Rollen mit PAM plugin
  - https://www.percona.com/blog/2015/03/02/emulating-roles-percona-pam-plugin-proxy-users/
- PAM Authentication (Unix, LDAP)
  - Percona, MariaDB und MySQL Enterprise

Verschlüsselung

- Wer ist der Angreifer?

- Client/Server SSL
- App: Record Encryption
- MariaDB 10.1.4
  Table(space) Encryption
- Filesystem Encryption

- Keys aus Memory hacken
- Performance Einfluss
- Operations/Fehlerbehebung



Zutritt verschaffen

- `~/.my.cnf` --> user / password
- `~/.mylogin.cnf` --> Entschlüsseln
- `/etc/mysql/conf.d/debian.cnf` (= `root`)
- `shell> history`
- `~/.mysql_history`
- `$datadir/master.info`
- `skip_grant_tables`
- So fertig Security?

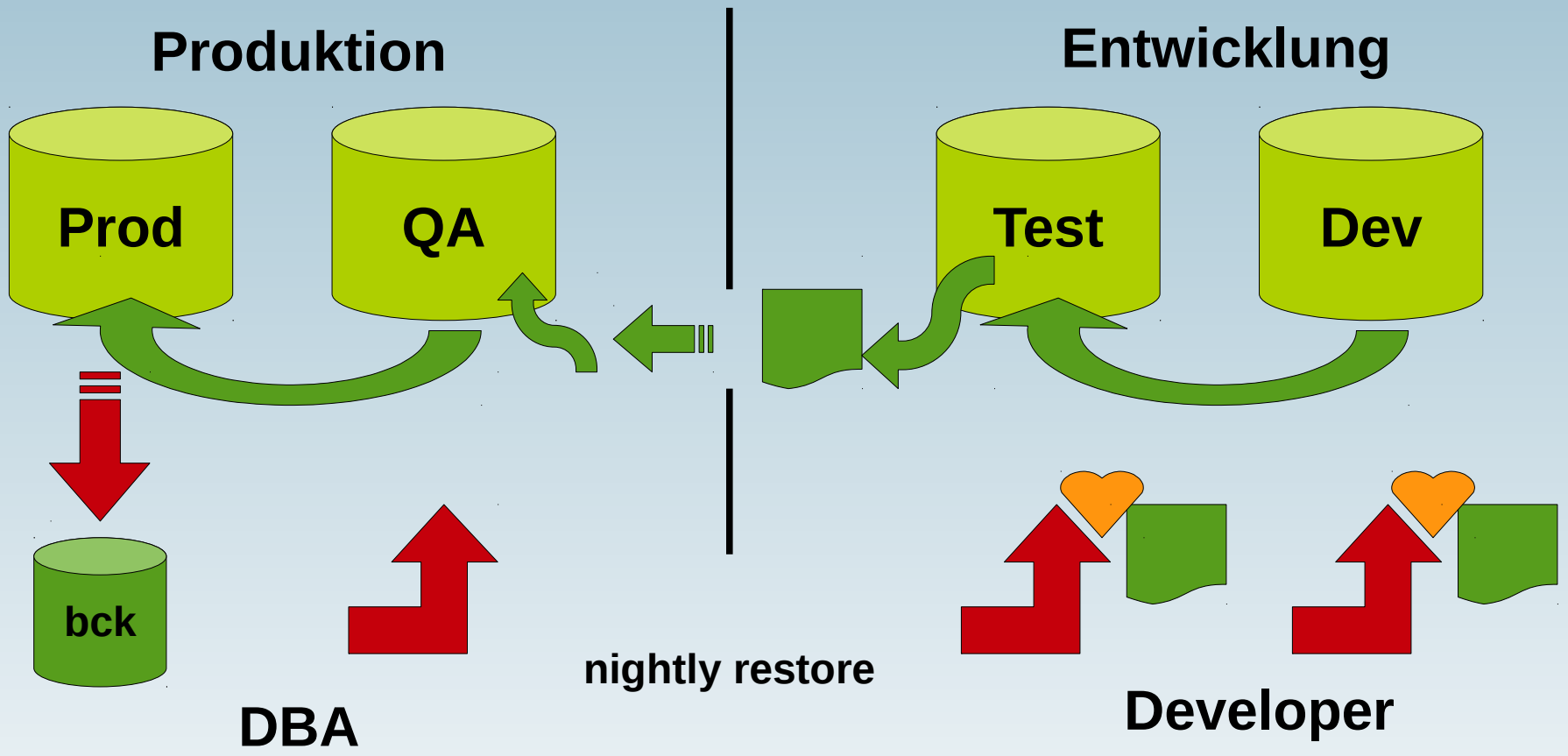
Ups!

Backup / Restore

- Backup? Nur für Mädchen...
- Bei logischen Fehlern
  - Ups Query!
  - Hardware-Ausfall --> HA Lösung
- Typen von Backup
  - Logische Backups (mysqldump)
  - Physische Backups (LVM, xtrabackup)
  - Delayed Replikation
- Restore
  - MTTR (Zeit)
- Erlaubter Datenverlust?
  - Point-in-Time-Recovery (Binary Logs)
- Automatisieren, Testen (regelmässig)
  - Warum?
  - Unser Konzept...

"I back up my files religiously. I pray nothing happens to them."

Restore regelmässig testen...



Und jetzt das...

Hochverfügbarkeit (HA)

- Ja, wir haben Sicherheit!
- Ja, wir haben Backup!
- Was passiert bei Hardwareausfall???
  - MTTR bei Hardwareausfall?
  - Redundanz
    - Hilft aber nicht bei logischen Fehlern!
- Master/Slave, Galera
  - Delayed Replikation für logische Fehler

Automatisches Failover

- KISS
  - Ihr müsst mit Eurem HA umgehen können...!
- Umschalten
  - voll manuell
  - halb-automatisch
  - voll-automatisch
- Wer entscheidet ob failover oder nicht?
  - Voll-automatisch kling verlockend!
  - Oft falsche Failovers, da schwierig zu entscheiden
- Wir empfehlen: halb-automatisch...

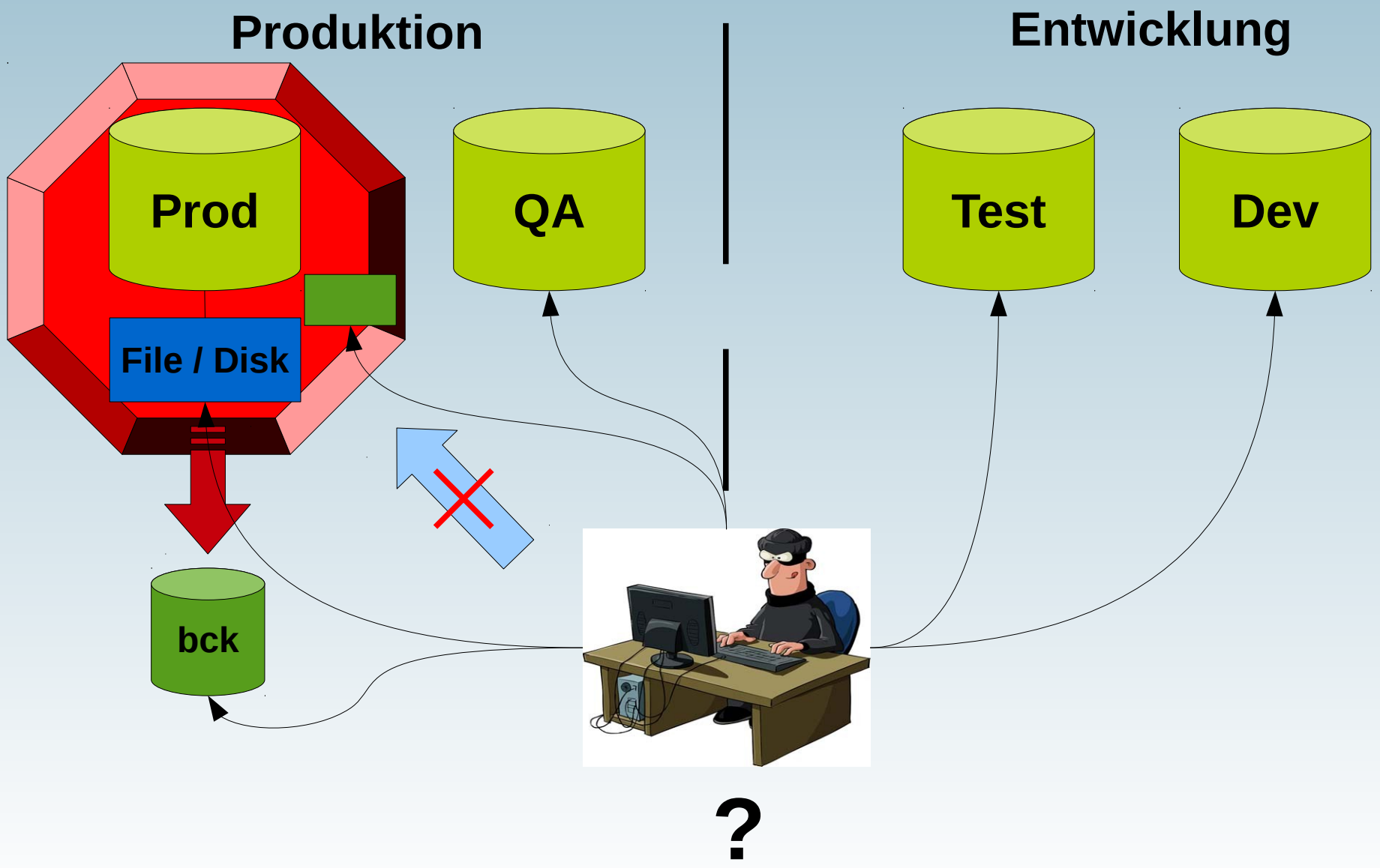
Monitoring

- Wie stellen wir fest, dass was schief läuft/lief?
- Überwachen (Monitoring)
  - Notfall
  - Trends
  - MySQL Error log.
- Lösungen:
  - FromDual Plugins für Nagios
  - FromDual Performance Monitor für MySQL und MariaDB
  - MySQL Enterprise Monitor
- Critical:
  - DB up/down
  - Filesystem full
  - Replikation läuft
- Alles andere:
  - Nice to have (Performance Graphen)

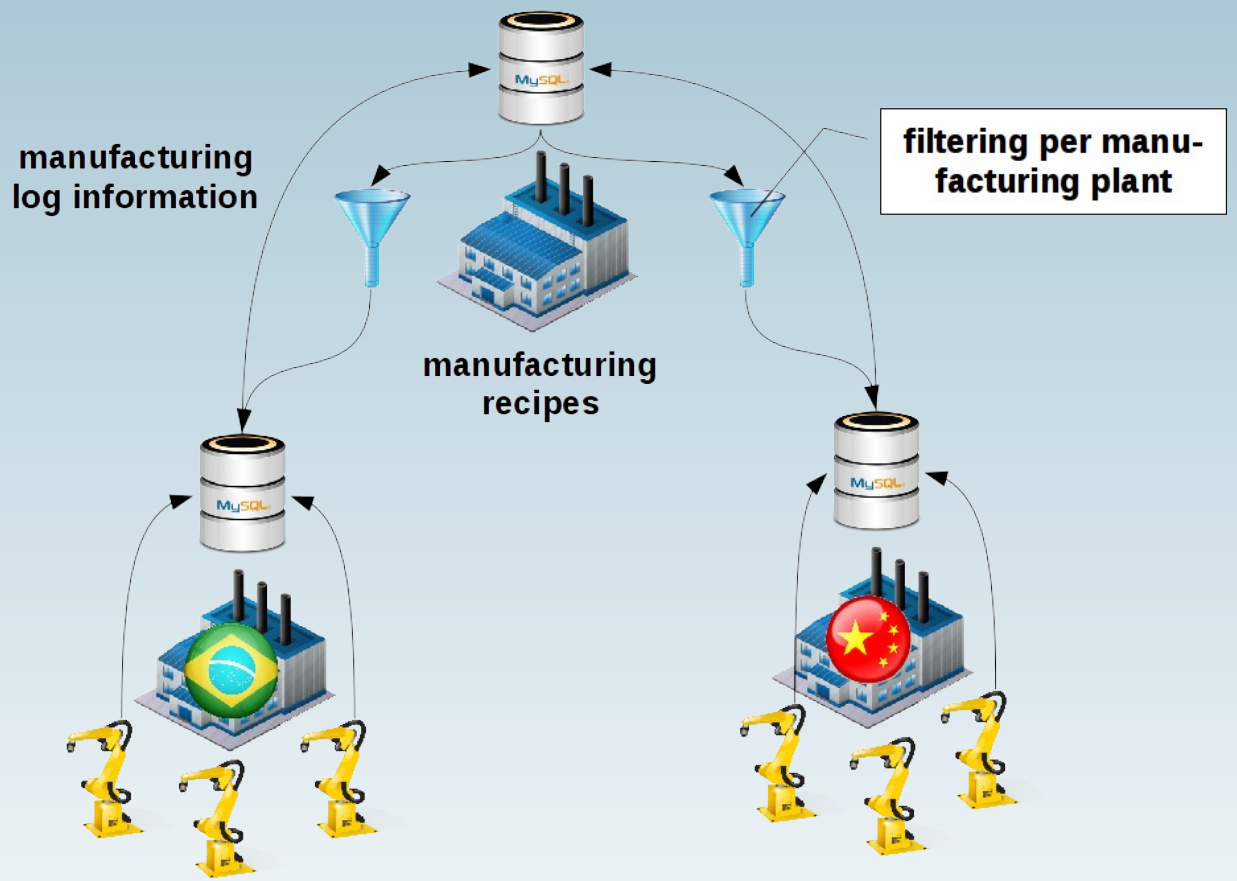
Wo lauert das Böse?

- Personen (eher Datenklau als Zerstörung)
  - Hardware-Schrauber
  - Unix-Admin (root)
  - DBA (kein root?)
  - Entwickler?
  - Endnutzer (Sales?)
  - Sonstige (Putzfrau, Chef, ...)
- Hintermänner:
  - Ehemalige Mitarbeiter
  - Konkurrenz / Geheimdienste
  - Kriminelle Organisationen, Erpressung, DoS
- Äussere Einflüsse (eher Zerstörung als Datenklau)
  - Wasser (Stausee, Hochwasser, Flutwelle, Lawine)
  - Feuer, Blitz, Sturm, KKW
  - Erbeben, Erdrutsche
  - Hochseefrachter :-)
  - Terror- und/oder Bombenanschläge (Philipinen)
  - Kriegerische Handlungen (Ukraine, „nuclear bomb blast on the city of London“)
  - Sonstige Sabotage-Akte

Angriffsvektoren



Beispiele aus der Praxis



Literatur

- FromDual Security: http://www.fromdual.com/security
- MySQL Docu: http://dev.mysql.com/doc/refman/5.6/en/security.html
- MySQL Enterprise Security: http://dev.mysql.com/doc/refman/5.6/en/mysql-enterprise-security.html
- Security in MySQL: http://dev.mysql.com/doc/mysql-security-excerpt/5.6/en/index.html
- What is new in MySQL 5.7: http://dev.mysql.com/doc/refman/5.7/en/mysql-nutshell.html
- Security Vulnerabilities Fixed in MariaDB: https://mariadb.com/kb/en/mariadb/security/

Wir suchen noch:

www.fromdual.com

MySQL Datenbank Enthusiast/in für Support / remote-DBA / Beratung

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