MySQL security best practices

A 101 talk presented by Dimitri Vanoverbeke

MySQL Security

- Having a security mindset.
  - Infrastructure
  - Operating system
  - Applications
- MySQL privileges
- SSL communication
- Handling ransomware
- Encryption options

Have the correct mindset

- Applications should be written with security from the ground up.
- Work together with your **sysadmins** and **devteam** to make the correct choices.
- Disable and restrict remote access
- Understand the **cloud** means working on **other peoples comput**ers.
- **Restrictive** mindset

Infrastructure: Network

- Separate your network
  - Only application servers should be able to connect to the DB remotely.
  - Dev access/general access should be limited by using a bastion/jumphost
  - DO NOT OPEN IT UP TO THE INTERNET!!! (or use strict firewall rules)
- IPS/IDS appliance/software can be handy

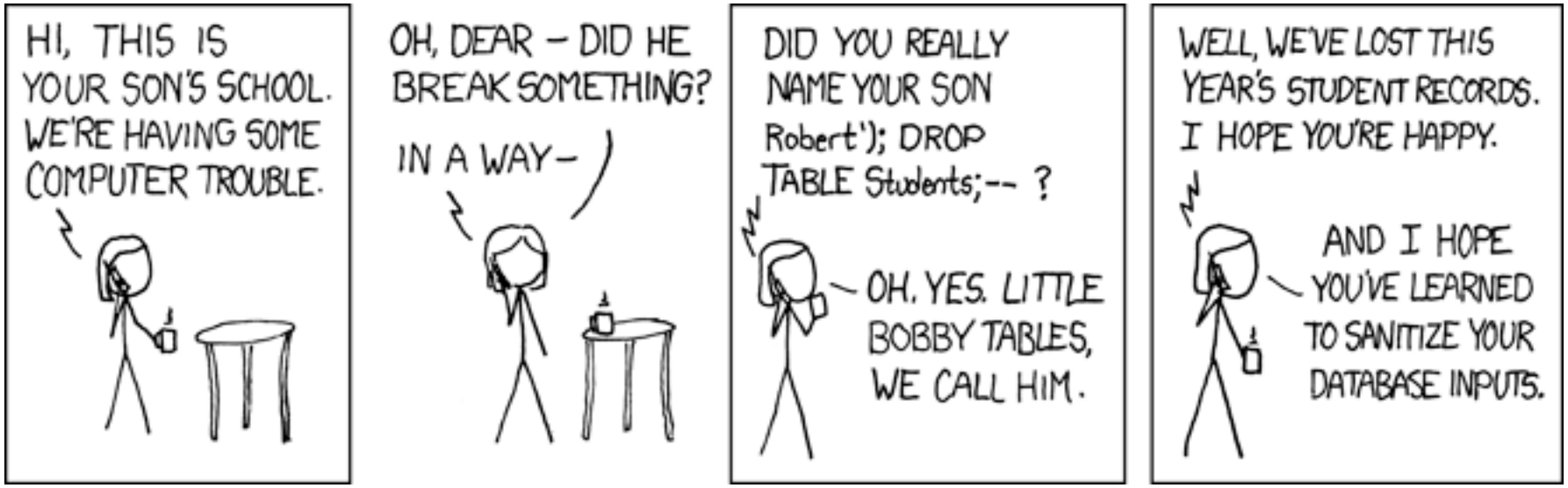
Be friends with your network engineer

Operating system security

- Deploy security patches as soon as possible.
- Make sure permissions are correct:
  - mysql should be the owner
  - Don’t use chmod 777 :-)
- selinux setenforce 1
- If PCI compliant?
  - ecryptfs
- Use trusted package sources!

Applications

- Perform penetration tests on staging environments.
- Validate user inputs
- Watch out for SQL injections.



Use configuration management

- Use your favourite configuration management solution. Tools like puppet and chef are excellent tools to ensure compliancy:

```
$users = {
  'dim0@localhost' => {
    ensure                   => 'present',
    max_connections_per_hour => '0',
    max_queries_per_hour     => '0',
    max_updates_per_hour     => '0',
    max_user_connections     => '0',
    password_hash            => '*T5D3A5831A93829BE2468926B4132313728C250DBF',
  },
}
```

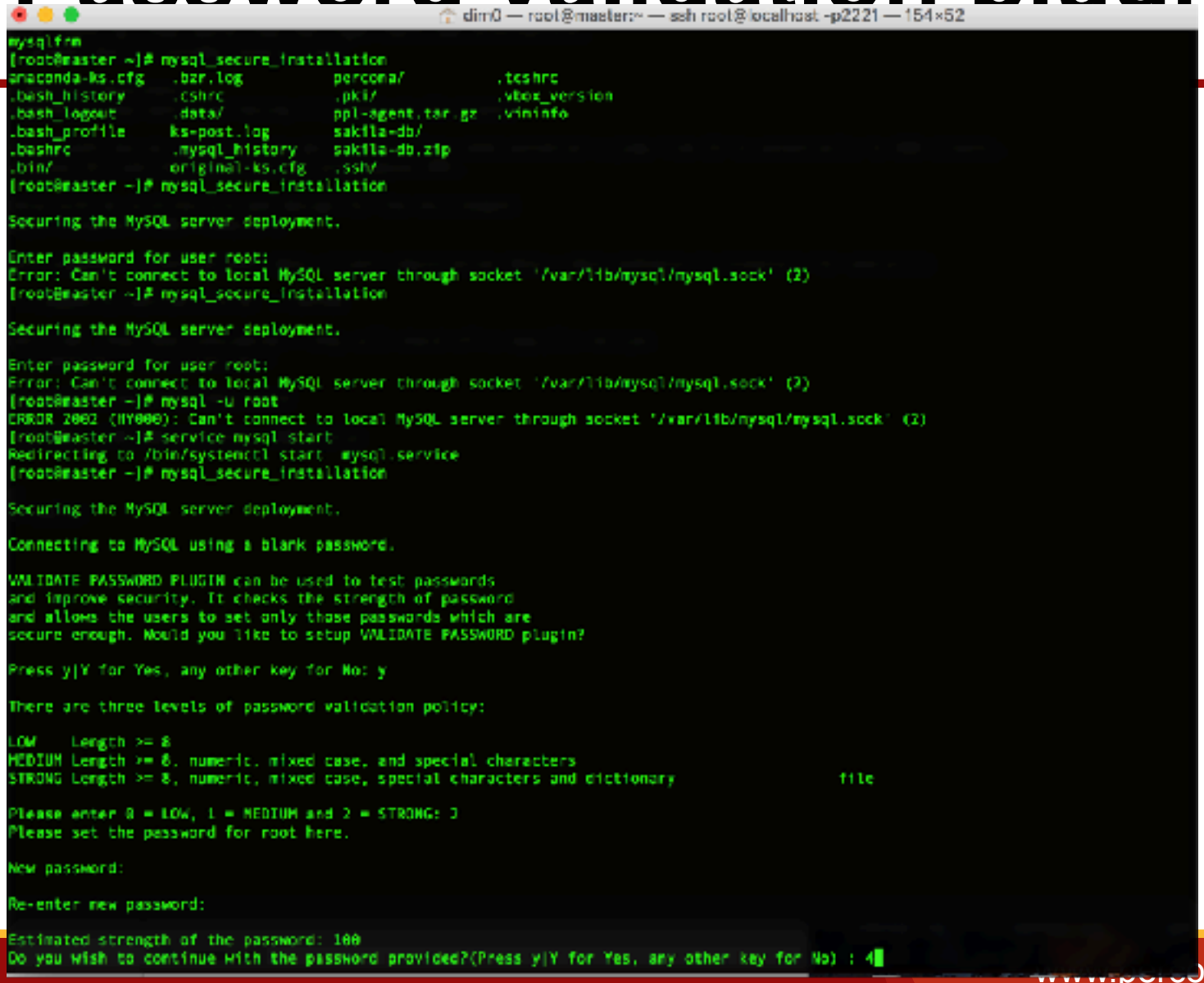
(Again) use configuration management

- Configuration management will help you with:
  - Consistent and effective rollout of your configuration files
  - MySQL database version (security patches, feature updates, etc)
  - OS security updates
  - User management
  - Resource limitations
  - Documents environments
  - Ensures the correct packages are installed
  - Less manual work

MySQL privileges

- Limit your user privileges to key application servers.
- Be restrictive for your meat-ware
- Use complex passwords
  - Use the password validation plugin

Password validation plugin



MySQL Grants

- Identify users based on: **user@host**
  - user: **username**
  - host: **hostname/ip**/network of the client that connects
  - different host, different user, different ‘grants’
  - use of **wildcards**
- Examples:

‘dim0’@‘localhost’, ‘root’@‘localhost’
‘tommeketoch’@‘app0001’, ‘kenju’@‘192.168.%’
‘ledijkske’@‘192.168.1.212’, ‘fredjen’@‘app.fq.dn’

- Creating A User:

```
>CREATE USER 'dim0'@'app0001';
```

- Drop user: change CREATE into DROP

MySQL Grants (2)

- Grant the user some kind of privilege
- Grant ... to:

server,
database,
table,
column,
trigger,
stored procedure,
view,
index

- Example: INSERT, SELECT, UPDATE, DELETE
  - SQL Command:

> GRANT SELECT ON db.* TO ‘dim0’@‘app0001’;
> GRANT INSERT ON *.* TO ‘dim0’@‘app0001’;

- Revoking privileges: change GRANT into REVOKE

MySQL grants (3)

- Password Expiration Policy
- User Account Locking

MySQL supports locking and unlocking user accounts using the ACCOUNT LOCK and ACCOUNT UNLOCK clauses

Grants (Limit your resources)

- For every user: max_user_connections

mysql> GRANT USAGE ON db.* TO ‘dim0'@'localhost' WITH **MAX_QUERIES_PER_HOUR** 1000 **MAX_UPDATES_PER_HOUR** 999 **MAX_CONNECTIONS_PER_HOUR** 100 **MAX_USER_CONNECTIONS** 5; FLUSH USER_RESOURCES;

It’s however not really popular… :-D

SSL connection

- SSL encryption to ensure in transit encryption.
- Requirement for PCI and other security compliance.
- Can give a slight performance penalty
- AWS/RDS users should definitely have a look at this

Handling ransomware

- Again limit access to trusted services and users.
- Make sure you have backups - offsite
- Sanitise user input fields in your application.

Encryption

- Encrypting your filesystem is still the most popular option.
- Since MySQL 5.7 table level encryption is included.

Closing remarks

- Disable the use of the “LOAD DATA LOCAL INFILE”
- Audit plugin
- pam authentication

Remember be restrictive!!!