



## 8 Simple Rules to Design Secure Apps



Nick Westerlund  
Senior MySQL DBA  
westerlund@pythian.com

Augusto Bott  
Senior MySQL DBA, Team Lead  
bott@pythian.com



# Who we are

**DBAs**

**Paranoid Architects**

**Don't patch around something**

**Design so it does not need patching**



# Why are we doing this

**Apps get visibility  
DBs grow**

**The App becomes a target  
data gets interesting to be owned**

**You're in trouble**



# How do we do it

**Pay attention to some basic rules**  
**Develop your own**

**Be suspicious, always**

**Nothing is unbreakable**  
**Design for resiliency**



## The basic rules

- Don't trust anything that comes from outside the Firewall**
- Use and abuse of stored procedures and views to isolate**
- Isolate raw data from the App user**
- Make sure you have proper credentials management**
- Do not send data in plain text (ever!)**
- Do not store passwords anywhere**
- Make sure your application is Auditable**
- Make sure it's recoverable**



# Outside of Firewall

- **Your enemy is outside of the firewall**
- **But might be inside already**
- **Double-check and validate**
- **Sanitize your data**
- **Prevent execution privileges**



# (Ab)use Stored Procedures

- Mask your data
- `add_to_cart(session_id, product_id, qty)`
- `see_cart_content(session_id)`
- `! SELECT * FROM cart;`



## Isolate raw data

```
mysql> CREATE DEFINER='root'@'localhost' SQL SECURITY  
DEFINER VIEW t3 AS SELECT a FROM t1 WHERE b = 5 AND  
active=1;  
Query OK, 0 rows affected (0.42 sec)
```

```
mysql> GRANT SELECT (a) ON example.t3 to 'app'@'172.16.1.%'  
identified by 'secret';  
Query OK, 0 rows affected (0.04 sec)
```



# Credential Management

- **Users, app users**
- **Passwords**
  - **Change regularly**
  - **Make them complex**
- **Minimal credentials**
- **Restrict access**



# Don't send plain-text data

- **Encrypt everything**
- **Use asymmetrical encryption**
- **Privacy should be considered**
- **Use certificates**
- **Do not store plain-text sensitive data**



# Don't store passwords

- Use an App server
- Do not store passwords on the filesystem



# Audit?

- **Logging**
- **Historical tables**
- **Keep track of who did what**
- **Access to the system**



# Make sure you can recover

- Backups
- Replication
- Use a DR site
- Retrace the steps



Questions?