SQL Injection Attacks: A Quick Primer

hassan.abudu@owasp.org

OWASP

The Open Web Application Security Project

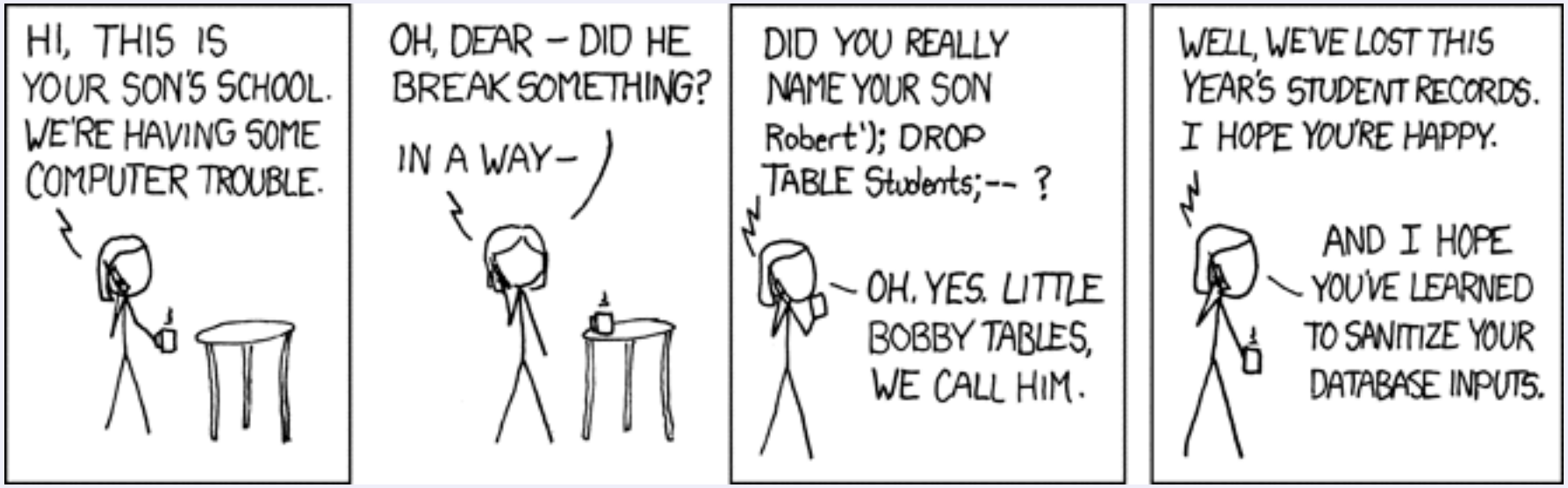
OWASP
The Open Web Application Security Project

OWASP Top 10 Vulnerabilities - 2017

Rank	Name
1	Injection
2	Broken Authentication
3	Sensitive Data Exposure
4	XML External Entities
5	Broken Access Control
6	Security Misconfiguration
7	Cross-Site Scripting
8	Insecure Deserialization
9	Using Components with Known Vulnerabilities
10	Insufficient Logging & Monitoring

Injection Attacks

An important lesson: Trust nobody



Explanation

Suppose user makes a modified HTTP request
› https://www.store.com/orders?year=0%20OR%201%3D1

SELECT date, item FROM orders

WHERE user=126 AND year=0 ***OR 1=1***

Effect
› sets year variable to 0 OR 1=1
› shows all orders in the database

OWASP
The Open Web Application Security Project

Price List for Stolen Data

Address	*$0.50*
Phone number	*$0.25*
Unpublished phone	*$17.50*
Cell phone number	*$10*
Date of birth	*$2*
Social Security number	*$8*
Drivers's License	*$3*
Education	*$12*
Credit History	*$9*
Bankruptcy details	*$26.50*
Lawsuit information	*$2.95*

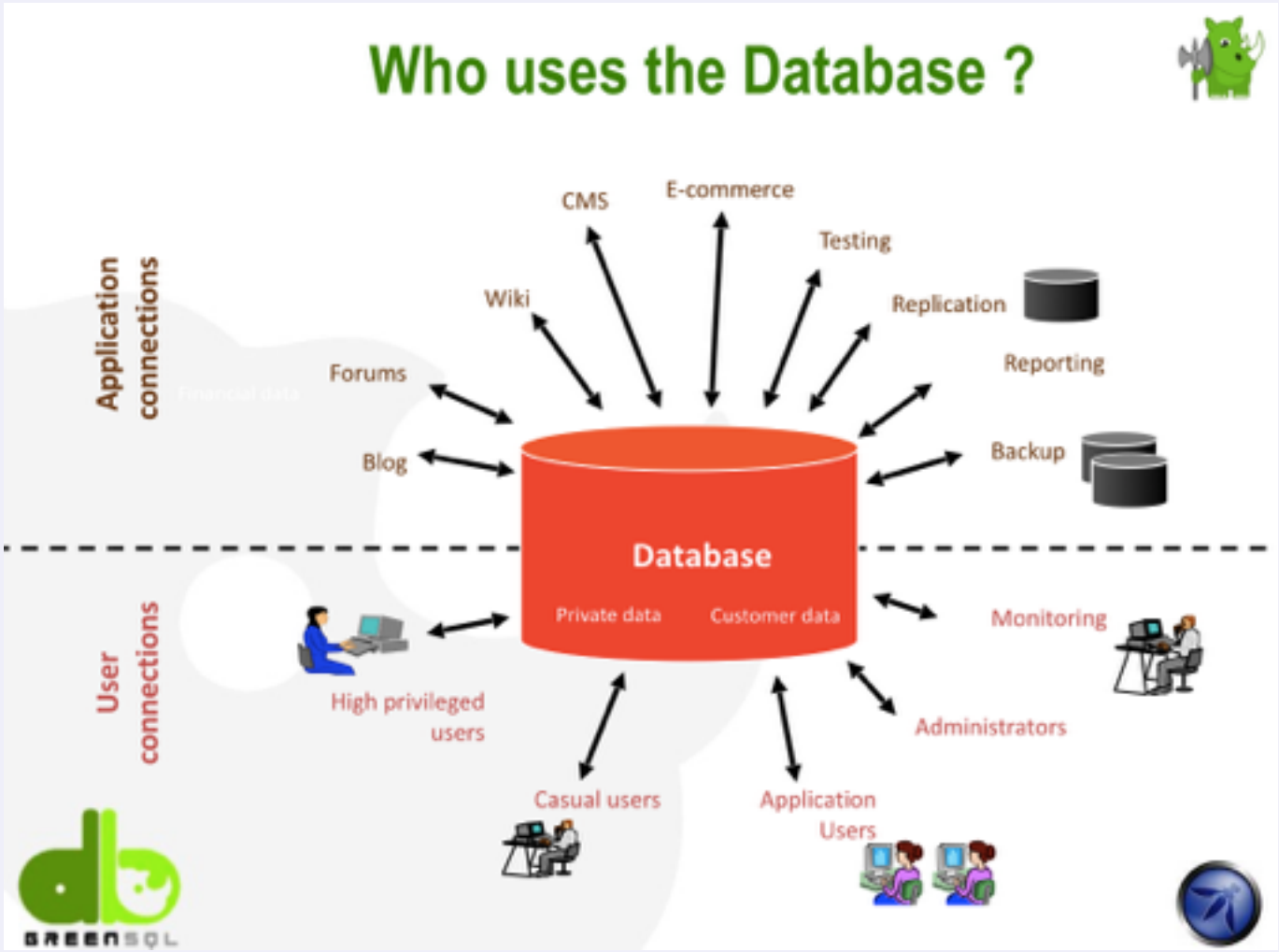
OWASP
The Open Web Application Security Project

Solution

Use parameterized queries, and don’t sweat it!

```
// PHP - PDO

$stmt = $dbh->prepare("INSERT INTO REGISTRY (name, value) VALUES (:name, :value)");
$stmt->bindParam(':name', $name);
$stmt->bindParam(':value', $value);
```


Who uses the Database ?
Application connections
CMS
E-commerce
Testing
Wiki
Replication
Forums
Reporting
Blog
Backup
Database
Private data
Customer data
User connections
Monitoring
High privileged users
Administrators
Casual users
Application Users

OWASP
The Open Web Application Security Project

Thanks for your attention! :-)

(Easy) Questions?