SQL Exploitation

Database Connectivity and Access

© 2020 Zilora

SQL Exploitation

- A big newsmaker
- Hacking is easy, given basic SQL knowledge
- Can cause major havoc
- The result:
  - EVERYBODY PANIC!! (not really)

© 2020 Zilora

SQL Exploitation

- Two types:
  - Malformed queries reveal underlying structure
  - SQL injection reveals data

© 2020 Zilora

Malformed Queries

- In your code, watch out for…
  - Carriage returns or line feeds
  - Null characters
  - “Ext’r”a quotes
  - # Comments (or -- in mysql)
- May be intentional or unintentional
- Can reveal underlying structure
- Can be protected against…

© 2020 Zilora

Malformed Queries Defenses

- Proactive methods
  - Whitelist—Only accept certain statement types
  - Sanitization—Strip query of inappropriate characters
- Reactive methods
  - **Catch the exception!**

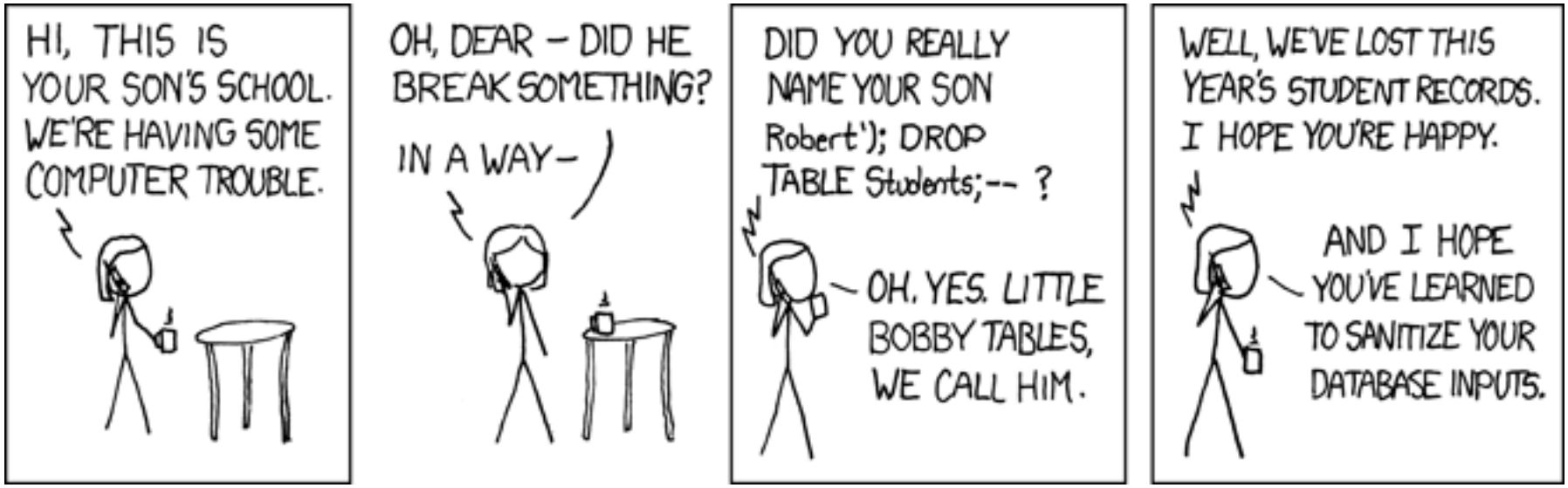
© 2020 Zilora

SQL Injection

- Changing the meaning or behavior of a statement
- Typically used in sequence to gain information
- Usually don't cause syntax errors

© 2020 Zilora

Detailed Example



http://xkcd.com/327/

© 2020 Zilora

Injection Example

- Using the xkcd suggestion…

"SELECT * FROM Students WHERE name IN ('" + stuName + "');"

and

stuName = "**Robert'); DROP TABLE Students; --** "

becomes, and passed to the database

SELECT * FROM Students WHERE name IN ('**Robert'); DROP TABLE Students; -- ');**

© 2020 Zilora

Another Example

This statement:

"SELECT username, password, name FROM Users WHERE username = '" + username + "' AND password = '" + password + "';"

when user enters

username: **someUser**

password: **somePass' OR 1 = 1; --**

becomes

SELECT username, password, name FROM Users WHERE username = '**someUser**' AND password = '**somePass' OR 1 = 1; --** **';**

© 2020 Zilora

SQL Injection Prevention

- **Prepared statements!**

  If you try to use an alternate access or commands, the prepared statements will throw an exception.

© 2020 Zilora

SQL Exploitation Prevention Summary

- Data cleansing
  - String testing libraries, query pre-processing
- Exception Handling
  - Don't allow structural information to flow to the user
- Prepared statements
  - Restrict input, and improve performance
- Stored procedures
  - Restrict input, effect thereof, without complex application logic

© 2020 Zilora

Language and DBMS Specific Exploits

- DBMS bugs are fixed all the time
- Zero-day exploits mean vulnerability *now*
- Language workarounds might introduce bugs
- Initial release or Maintenance coding might introduce bugs
  - Maintenance: Working on code may not be understood fully

© 2020 Zilora

Bottom Line

- Hackers are constantly trying new things
- As a professional, stay abreast of security developments
- Don't trust anyone
- Only expose what must be exposed
- Only expose what can't hurt you if stolen

© 2020 Zilora