SQL Injection

SQL Injection

- What is it?
  - A code injection technique used to attack a web app’s database server

- SQLi works by injecting SQL commands into an SQL query via web page input i.e. login form

SQL Injection

SQL Injection

- An attacker can execute malicious SQL statements to:
  - See entire database including sensitive information
  - Insert new data or change existing data or even delete data

- How does it work?

Comment Syntax

- SELECT 1+1;   # This comment continues to the end of line
- SELECT 1+1;   -- This comment continues to the end of line
- SELECT 1 /* this is an in-line comment */ + 1;

SQL in Web Page

- **Login code in the Web Page**
  - uName = getRequestString("username")
  - uPass = getRequestString("userpassword")

  - sql = 'SELECT * FROM Users WHERE Name = ' + uName + ' AND Pass = ' + uPass

- **Example**
  - Username: Michelle
  - Password: Firstlady
  - SELECT * FROM Users WHERE Name ='Michelle' AND Pass ='Firstlady'

SQL in Web Page

Injection

- sql = 'SELECT * FROM Users
  WHERE Name = ' + uName + ' AND Pass = ' + uPass

- Username: ' or 1=1; /*
- Password: */--

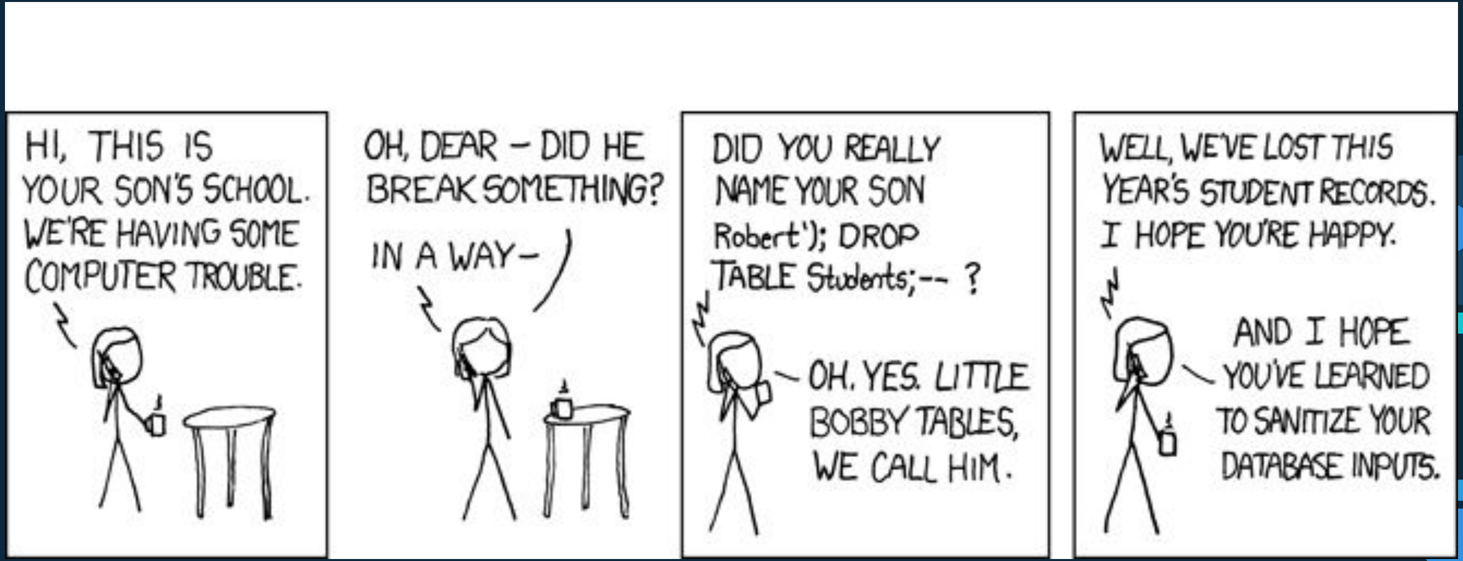
- SELECT * FROM Users
  WHERE Name = ' ' or 1=1; /* ' AND Pass = ' */--

Always TRUE **Become Comments**

SQL in Web Page

Other Injections

- Username: 105; DROP TABLE users

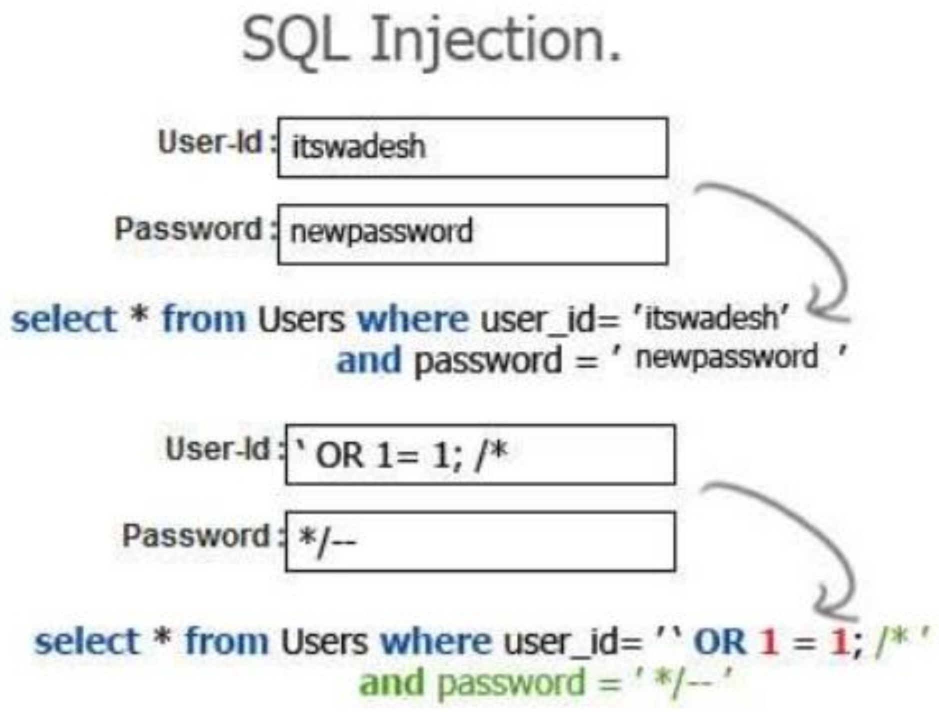

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL Injection

Try it

- https://www.hacksplaining.com/exercises/sql-injection

CONCLUSION



SQL Injection.

User-Id : itswadesh

Password : newpassword

```
select * from Users where user_id= 'itswadesh'
                    and password = ' newpassword '
```

User-Id : ' OR 1= 1; /*

Password : */--

```
select * from Users where user_id= '' OR 1 = 1; /* '
                    and password = ' */-- '
```

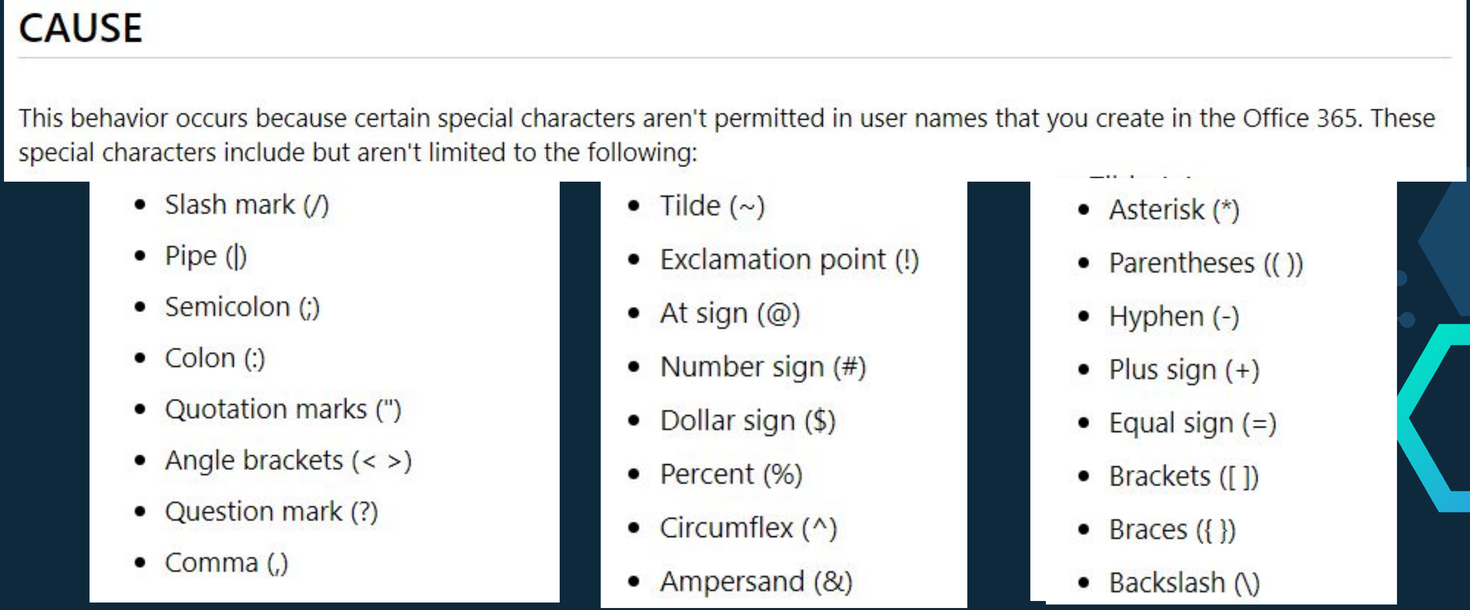
Prevention

- Input validation
  - Control the data types and numbers of characters accepted
  - Determines if a user’s input matches the expected format

Prevention

Example:

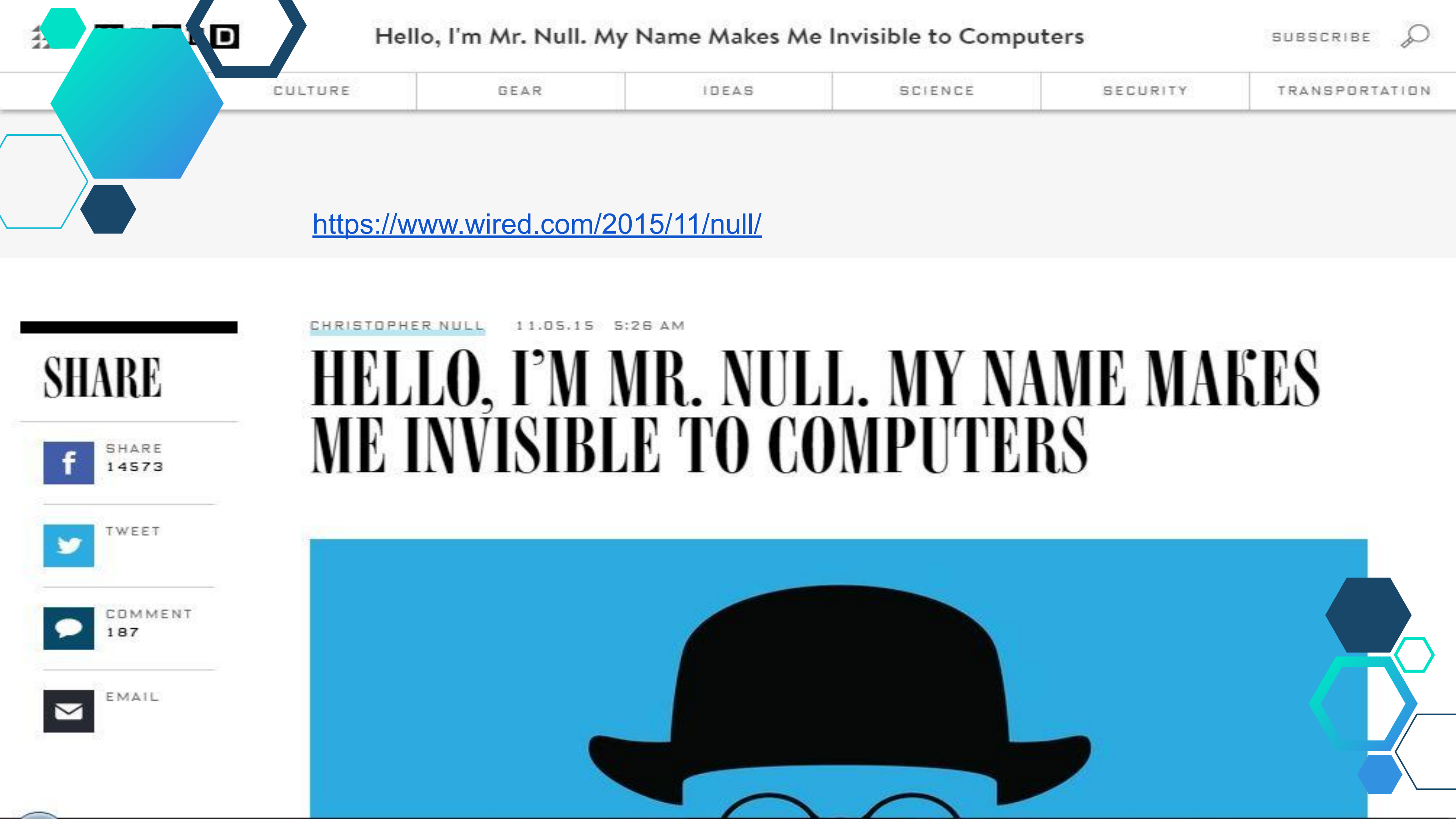
- Error message when you try to create a user name that contains a special character in Office 365: **"Invalid user name"**



CAUSE

This behavior occurs because certain special characters aren't permitted in user names that you create in the Office 365. These special characters include but aren't limited to the following:

- Slash mark (/)
- Pipe (|)
- Semicolon (;)
- Colon (:)
- Quotation marks (")
- Angle brackets (< >)
- Question mark (?)
- Comma (,)
- Tilde (~)
- Exclamation point (!)
- At sign (@)
- Number sign (#)
- Dollar sign ($)
- Percent (%)
- Circumflex (^)
- Ampersand (&)
- Asterisk (*)
- Parentheses (( ))
- Hyphen (-)
- Plus sign (+)
- Equal sign (=)
- Brackets ([ ])
- Braces ({ })
- Backslash (\)



https://www.wired.com/2015/11/null/

CHRISTOPHER NULL 11.05.15 5:26 AM

HELLO, I'M MR. NULL. MY NAME MAKES ME INVISIBLE TO COMPUTERS

More Challenges

- https://redtiger.labs.overthewire.org/