SQL Injection

Database Basics

- Relational databases store information in tables like this one
- They have columns for each category and rows for each entry
- Databases are collections of many of these tables

Name	FName	City	Age	Salary
Smith	John	3	35	$280
Doe	Jane	1	28	$325
Brown	Scott	3	41	$265
Howard	Shemp	4	48	$359
Taylor	Tom	2	22	$250

Database Basics

Table: users

id	username	password
1	prof_bailey	mypassword123
2	grandma	snickerdoodles1957
3	ThomasQ	.{5%^72fhslej\

SQL Basics

- Stands for Structured Query Language! There's a lot to learn about it (take 411 to really dive in), but you just need to know basic SELECT queries:

SELECT [column name(s)] FROM [table name]
WHERE [some condition]

SQL Basics

Example: **SELECT password FROM users WHERE username = ‘grandma’;**

1) Go to table ‘users’

2) Select all rows where username is ‘grandma’ (only 1 row here)

3) Return all items in the password column for those rows

id	username	password
1	prof_bailey	mypassword123
2	grandma	snickerdoodles1957
3	ThomasQ	.{5%^72fhslej\

SQL Basics

- Get creative with your SELECT and WHERE clauses! SELECT can choose any combination of columsn. WHERE can do boolean basic operations like **AND**, **OR**, **NOT**, etc., arithmetic, and checking values from multiple columns:

SELECT password FROM users WHERE username = ‘grandma’ **OR** username = ‘ThomasQ’;

SELECT username, password FROM users WHERE password = username **AND** id >= 1;

SELECT * FROM users WHERE password = ‘grandma’ **OR** 12 **<>** 3;

Injection

- Poorly made websites use simple string concatenation to build query statements. If they want to use your input (i.e. for a login), they might do something like this:

db.query(“**SELECT COUNT(*) FROM users WHERE username = ‘**“ + given_username + “**‘ AND password = ‘**“ + given_password + “**;**”)

Once concatenated, this string turns into:

db.query(“**SELECT COUNT(*) FROM users WHERE username = ‘grandma’ AND password = ‘snickerdoodles1957’**”**);**

Injection

- Let’s say you throw some extra quotation marks in (by accident, of course)...
- Instead of inputting **grandma**, you type **grandma’ OR username = ‘grandpa**

Now your query becomes:

db.query(“SELECT COUNT(*) FROM users WHERE username =

‘sn0wden’

AND password = ‘**xxx’ OR ‘’=’**””);

Injection

- This often means you can add your own SQL code to the query by finding a way to break out of the string.
- SQL Injection can be used to log in without a correct password, get passwords or other user information, or wreak havoc (re: “DROP table…”);

Protecting Against SQL Injection

1) Use given resources for sanitizing user input (i.e. Prepared/Parameterized Statements). Don’t try doing it yourself.
2) Hash your passwords!! (*screams*)
3) Third-party authentication
4) Quit the internet