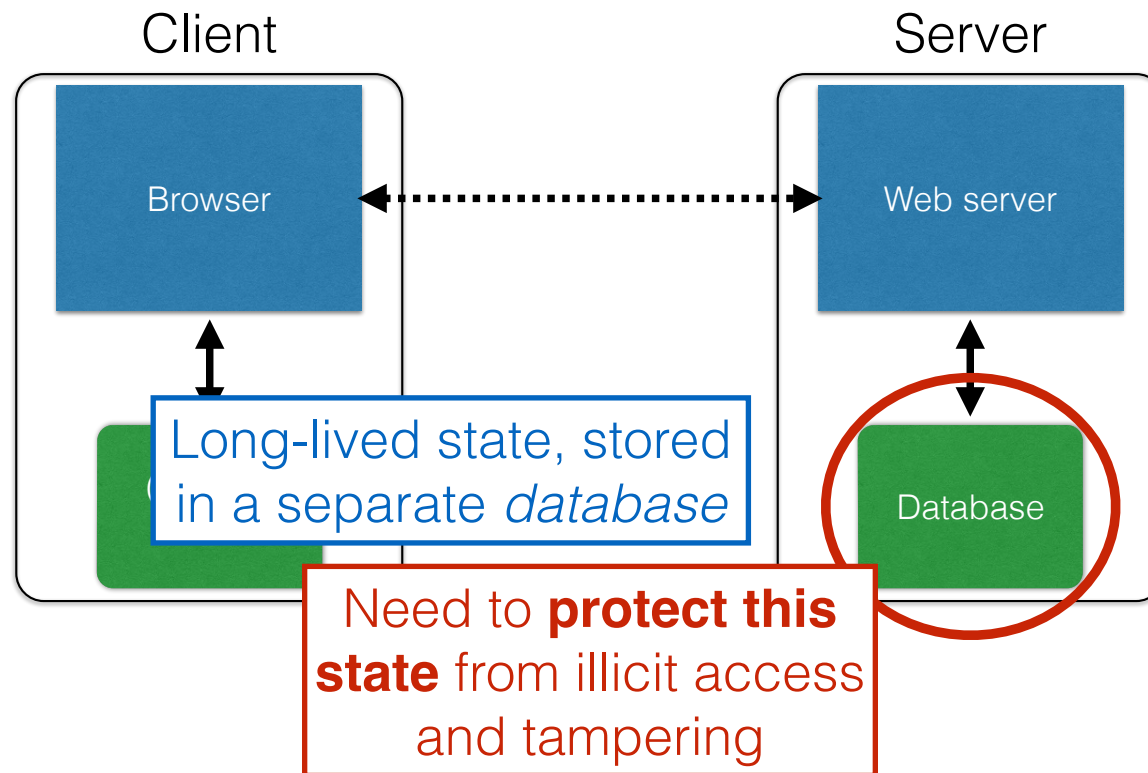


SQL injection

Server-side data



Server-side data

- Typically want **ACID** transactions
 - **Atomicity**
 - Transactions complete entirely or not at all
 - **Consistency**
 - The database is always in a valid state
 - **Isolation**
 - Results from a transaction aren't visible until it is complete
 - **Durability**
 - Once a transaction is committed, its effects persist despite, e.g., power failures
- **Database Management Systems** (DBMSes) provide these properties (and then some)

SQL (Standard Query Language)

Table

Users Table name				
Name	Gender	Age	Email	Password
Dee	F	28	dee@pp.com	j3i8g8ha
Mac	M	7	bouncer@pp.com	a0u23bt
Charlie	M	32	readgood@pp.com	0aergja
Dennis	M	28	imagod@pp.com	1bjb9a93

**Row
(Record)**

Column

```
SELECT Age FROM Users WHERE Name='Dee';      28
UPDATE Users SET email='readgood@pp.com'
WHERE Age=32; -- this is a comment
INSERT INTO Users Values('Frank', 'M', 57, ...);
DROP TABLE Users;
```

Server-side code

Website



Username: Password: Log me on automatically each visit ☐

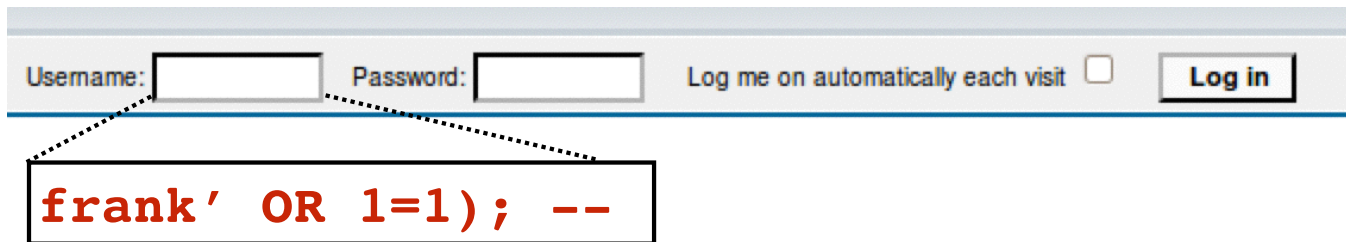
“Login code” (PHP)

```
$result = mysql_query("select * from Users  
                        where(name='$user' and password='$pass')");
```

Suppose you successfully log in as \$user
if this returns any results

How could you exploit this?

SQL injection



Username: Password: Log me on automatically each visit ☐

frank' OR 1=1); --

```
$result = mysql_query("select * from Users  
where(name='$user' and password='$pass')");
```

```
$result = mysql_query("select * from Users  
where(name='frank' OR 1=1); --  
and password='whocares')");
```

SQL injection



Username: Password: Log me on automatically each visit ☐

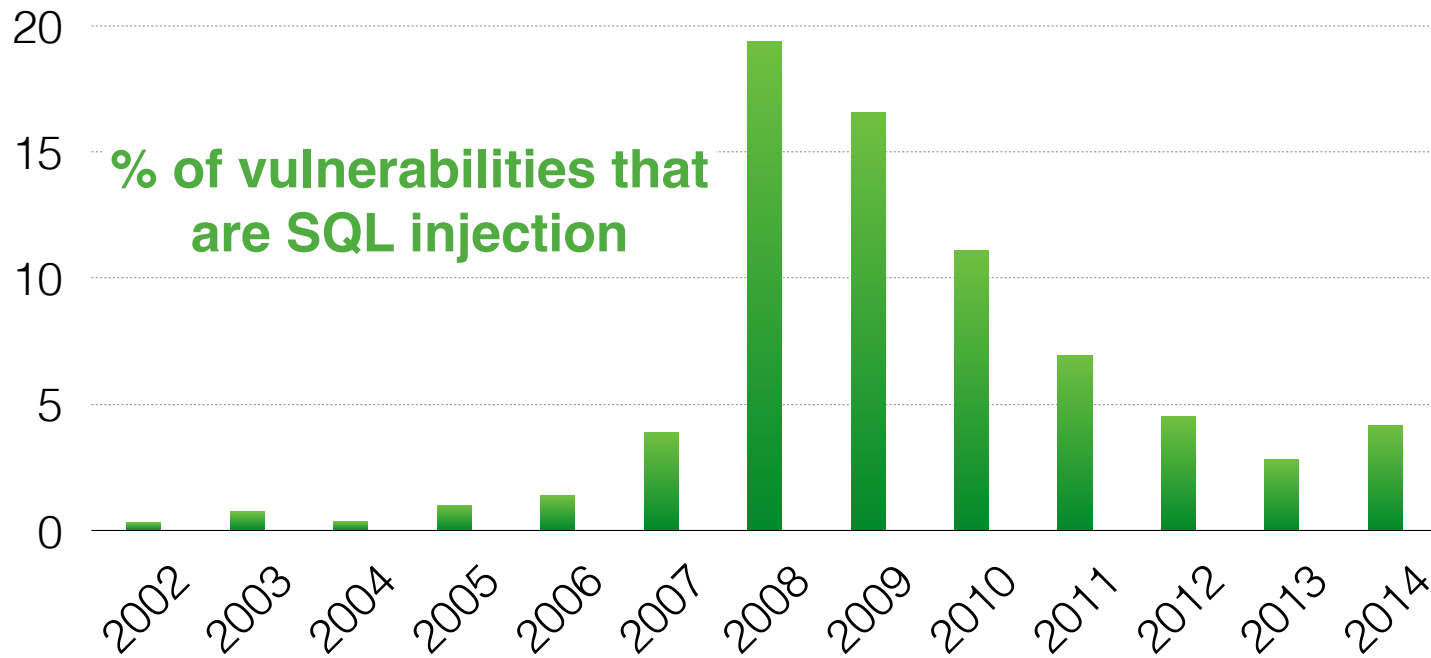
frank' OR 1=1); DROP TABLE Users; --

```
$result = mysql_query("select * from Users  
where(name='$user' and password='$pass')");
```

```
$result = mysql_query("select * from Users  
where(name='frank' OR 1=1);  
DROP TABLE Users; --  
and password='whocares')");
```

**Can chain together statements with semicolon:
STATEMENT 1 ; STATEMENT 2**

SQL injection attacks are common



<http://web.nvd.nist.gov/view/vuln/statistics>



<http://xkcd.com/327/>

