CS327E: Elements of Databases

Cybersecurity and SQL Injection

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Last updated: October 31, 2016 at 12:21

What I'd Like to Discuss

- Why cyber security is Important
- Why cyber security is hard
- SQL Injection

From the Headlines

Silent War, Vanity Fair, July 2013

On the hidden battlefields of history's first known cyber-war, the casualties are piling up. In the U.S., many banks have been hit, and the telecommunications industry seriously damaged, likely in retaliation for several major attacks on Iran.

Washington and Tehran are ramping up their cyber-arsenals, built on a black-market digital arms bazaar, enmeshing such high-tech giants as Microsoft, Google, and Apple.

From the Headlines

U.S. Not Ready for Cyberwar Hostile Attackers Could Launch, The Daily Beast, 2/21/13

Leon Panetta says future attacks could plunge the U.S. into chaos. We're not prepared. If the nightmare scenario becomes suddenly real ... If hackers shut down much of the electrical grid and the rest of the critical infrastructure goes with it ...

If we are plunged into chaos and suffer more physical destruction than 50 monster hurricanes and economic damage that dwarfs the Great Depression ... Then we will wonder why we failed to guard against what outgoing Defense Secretary Leon Panetta has termed a "cyber-Pearl Harbor."

The U.S. at Risk?

Experts believe that U.S. is perhaps particularly vulnerable to cyberattack compared to many other countries. Why?

The U.S. at Risk?

Experts believe that U.S. is perhaps particularly vulnerable to cyberattack compared to many other countries. Why?

- The U.S. is highly dependent on technology.
- Sophisticated attack tools are easy to come by.
- A lot of critical information is available on-line.
- Critical infrastructure may be accessible remotely.
- Other nations exercise more control over information and resources.

How Bad Is It?

Cyberwarfare greater threat to US than terrorism, say security experts, Al Jazeera America, 1/7/14

Cyberwarfare is the greatest threat facing the United States — outstripping even terrorism — according to defense, military, and national security leaders in a Defense News poll.

45 percent of the 352 industry leaders polled said cyberwarfare is the gravest danger to the U.S., underlining the government's shift in priority—and resources—toward the burgeoning digital arena of warfare.

Is Cyber Security Particularly Hard?

Why would cybersecurity by any harder than other technological problems?

Is Cyber Security Particularly Hard?

Why would cybersecurity by any harder than other technological problems?

Partial answer: Most technological problems are concerned with ensuring that something good happens. Security is all about ensuring that *bad things never happen*.

To ensure that, you have to know what all the bad things are!

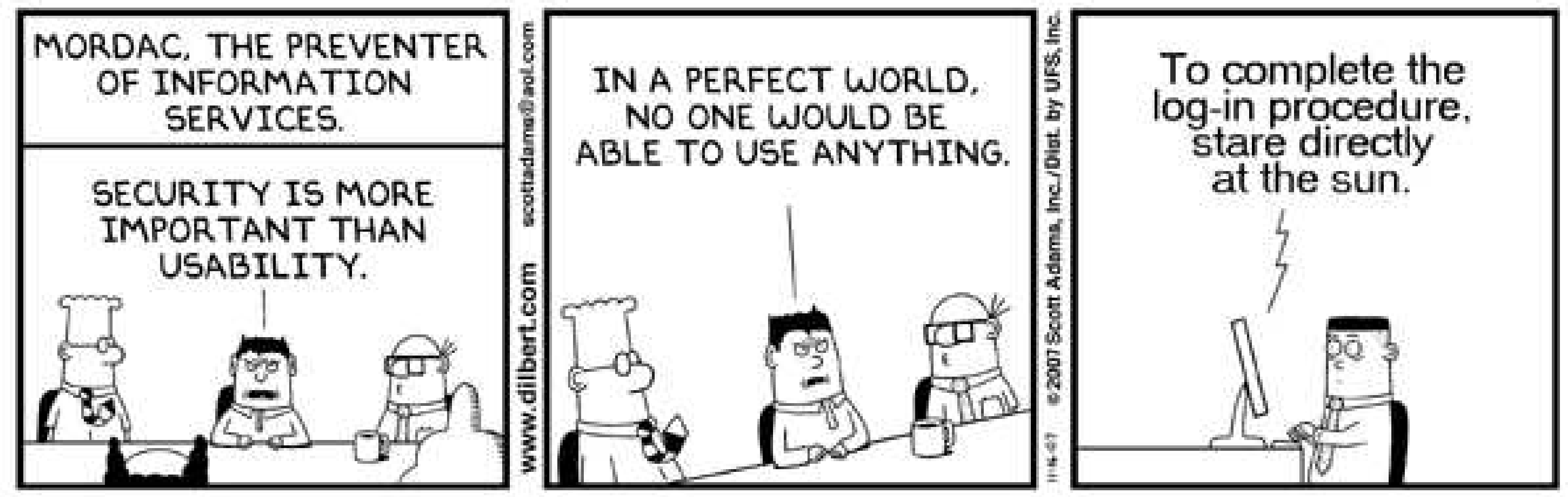
Cyber Defense is Asymmetric

In cybersecurity, you have to defeat an *actively malicious adversary*.

The defender has to find and eliminate *all* exploitable vulnerabilities; the attacker only needs to find *one*!

Cyber Security is Tough

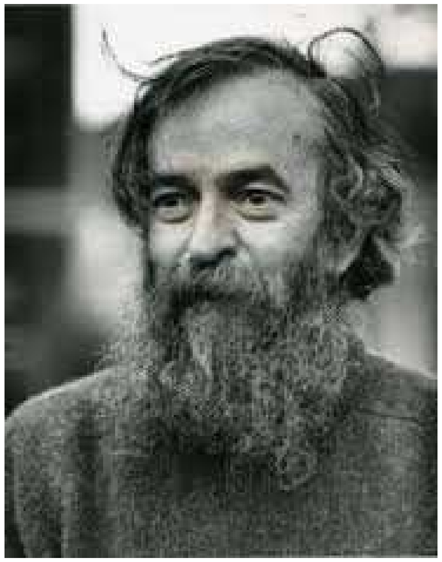
Perfect security is unachievable in any useful system. We trade-off security with other important goals: functionality, usability, efficiency, time-to-market, and simplicity.


MORDAC, THE PREVENTER OF INFORMATION SERVICES.
SECURITY IS MORE IMPORTANT THAN USABILITY.
www.dilbert.com
scottadams@aol.com
IN A PERFECT WORLD, NO ONE WOULD BE ABLE TO USE ANYTHING.
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.
To complete the log-in procedure, stare directly at the sun.

© Scott Adams, Inc./Dist. by UFS, Inc.

Is It Getting Better?

"The three golden rules to ensure computer security are: do not own a computer; do not power it on; and do not use it." –Robert H. Morris (mid 1980's), former chief scientist of the National Computer Security Center



"Unfortunately the only way to really protect [your computer] right now is to turn it off, disconnect it from the Internet, encase it in cement and bury it 100 feet below the ground." –Prof. Fred Chang (2009), former director of research at NSA

Some Sobering Facts

- There is no completely reliable way to tell whether a given piece of software contains malicious functionality.
- Once PCs are infected they tend to stay infected. The median length of infection is 300 days.

- “The number of detected information security incidents has risen 66% year over year since 2009. In the 2014 survey, the total number of security incidents detected by respondents grew to 42.8 million around the world, up 48% from 2013—an average of 117,339 per day.” (CGMA Magazine, 10/8/2014)

The Cost of Data Breaches

The Privacy Right's Clearinghouse's *Chronology of Data Breaches* (January, 2012) estimates that *more than half a billion sensitive records have been breached since 2005*. This is actually a very "conservative estimate."

The Ponemon Institute estimates that the approximate current cost per record compromised is around $318.

"A billion here, a billion there, and pretty soon you're talking real money" (attributed to Sen. Everett Dirksen)

How Bad Could it Be?

Some security experts warn that a successful possible widespread attack on U.S. computing infrastructure *could largely shut down the U.S. economy for up to 6 months.*

It is estimated that the destruction from a single wave of cyber attacks on U.S. critical infrastructures could exceed $700 billion USD—the equivalent of 50 major hurricanes hitting U.S. soil at once. (Source: US Cyber Consequences Unit)

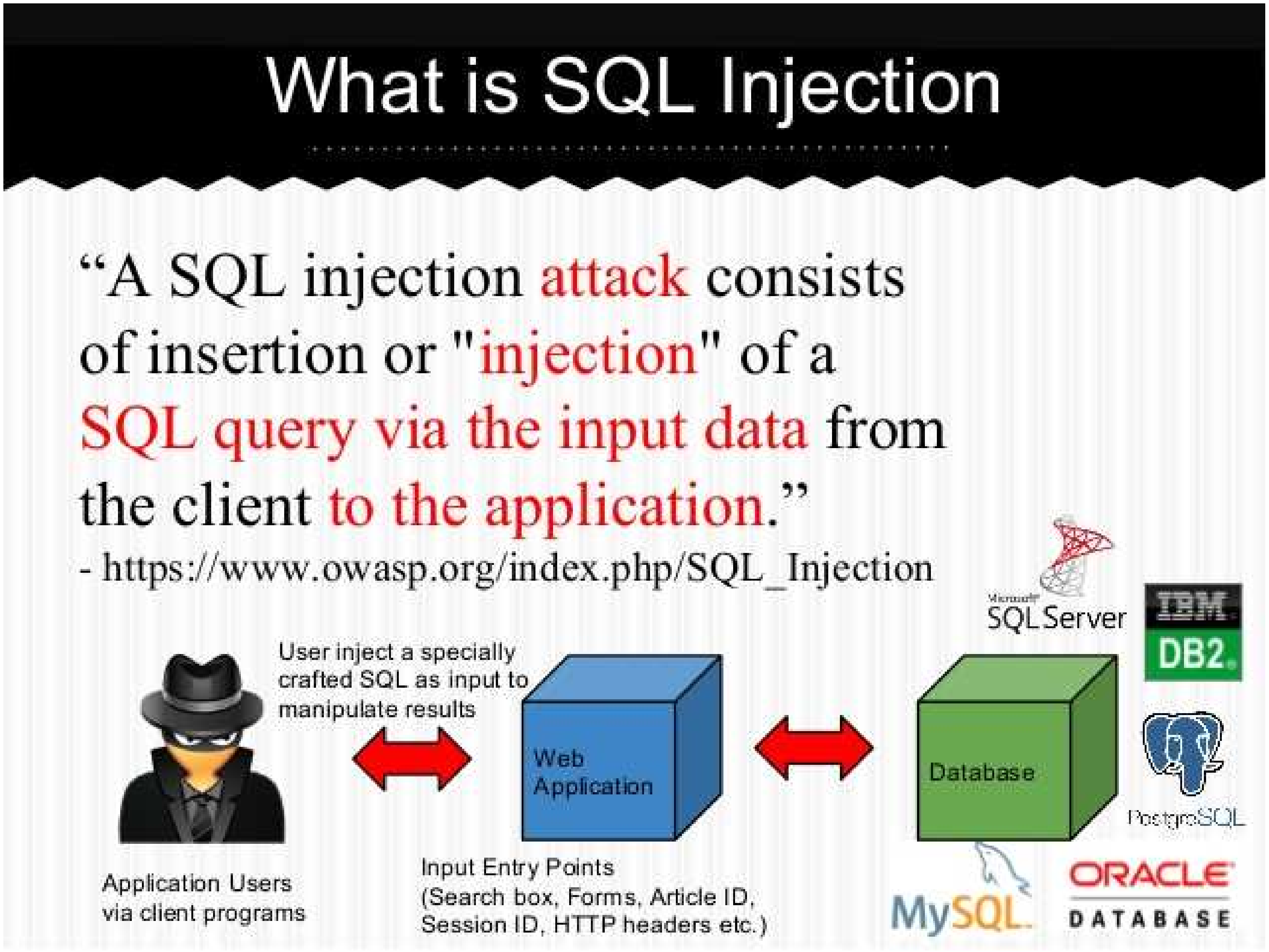
CyberAttacks: An Existential Threat?

Cyberattacks an 'Existential Threat' to U.S., FBI Says, Computerworld, 3/24/10

A top FBI official warned today that many cyber-adversaries of the U.S. have the ability to access virtually any computer system, posing a risk that's so great it could "challenge our country's very existence."

According to Steven Chabinsky, deputy assistant director of the FBI's cyber division: "The cyber threat can be an existential threat—meaning it can challenge our country's very existence, or significantly alter our nation's potential."

Structure of an SQL Injection?



What is SQL Injection

"A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application."

- https://www.owasp.org/index.php/SQL_Injection

User inject a specially crafted SQL as input to manipulate results
Web Application
Database
Application Users via client programs
Input Entry Points (Search box, Forms, Article ID, Session ID, HTTP headers etc.)
Microsoft SQL Server
IBM DB2
PostgreSQL
MySQL
ORACLE DATABASE

What is SQL Injection?

An *SQL Injection* is a vulnerability that results when you give an attacker the ability to influence the SQL queries that you pass to the database.

-: Administrator Login :-
Username : hi' or 1=1--
Password : ••••••••••••
login
SQL

They've been around a long time. In 1998, Rain Forest Puppy wrote an article for Phrack titled "NT Web Technology Vulnerabilities" that first highlighted SQL injection attacks.

Web Application Structure

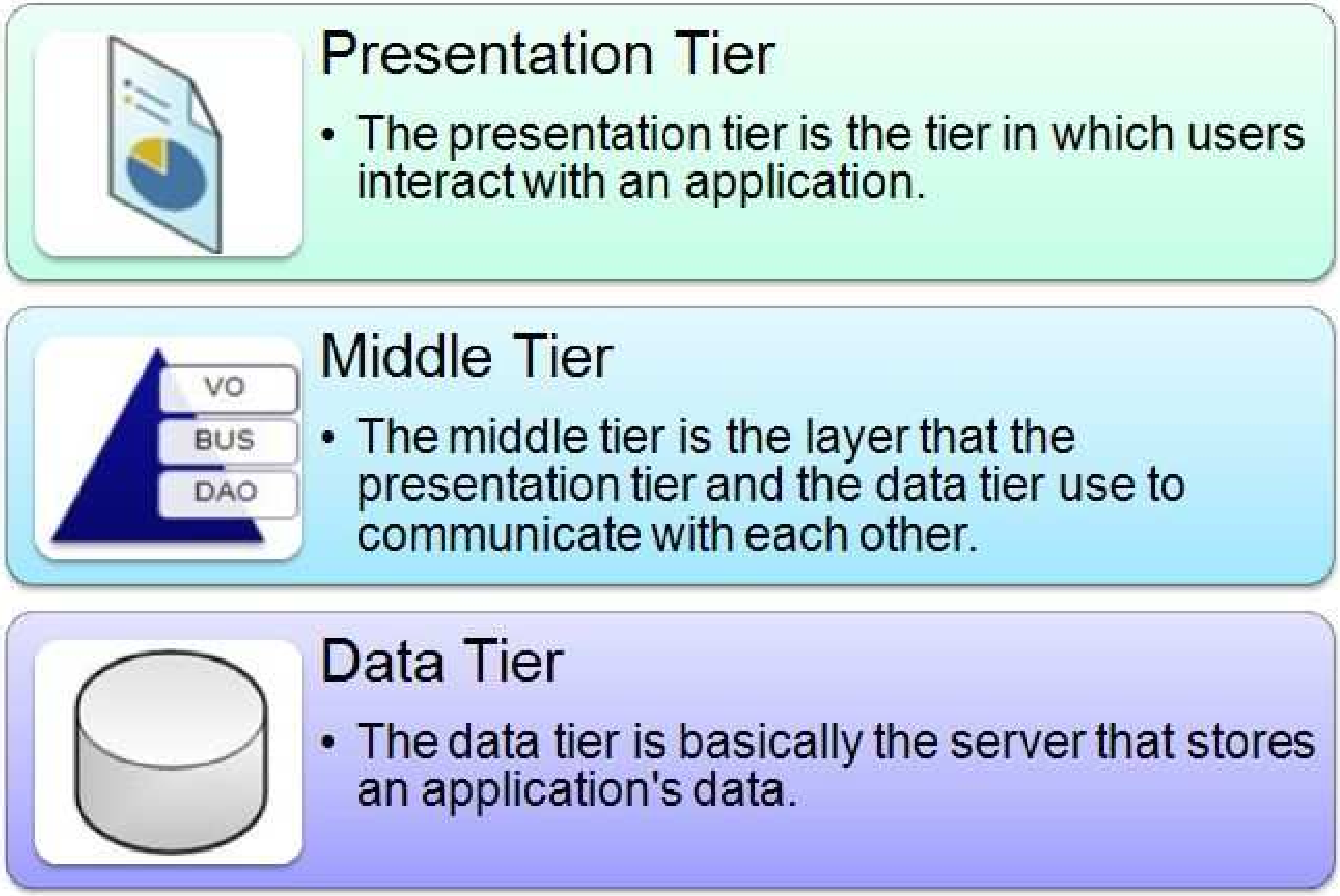
Most Web applications are *interactive*, accepting input from the user.

Many are also *database driven*, meaning that they query a database in response to user input.

Web applications often have three *tiers*:

1. **presentation tier**: interface (e.g. web browser) accepting user inputs;
2. **middle (logic) tier**: services user requests by presenting queries to the database;
3. **data tier**: database processing queries from the logic tier.

Web Application Structure



Presentation Tier

- The presentation tier is the tier in which users interact with an application.

Middle Tier

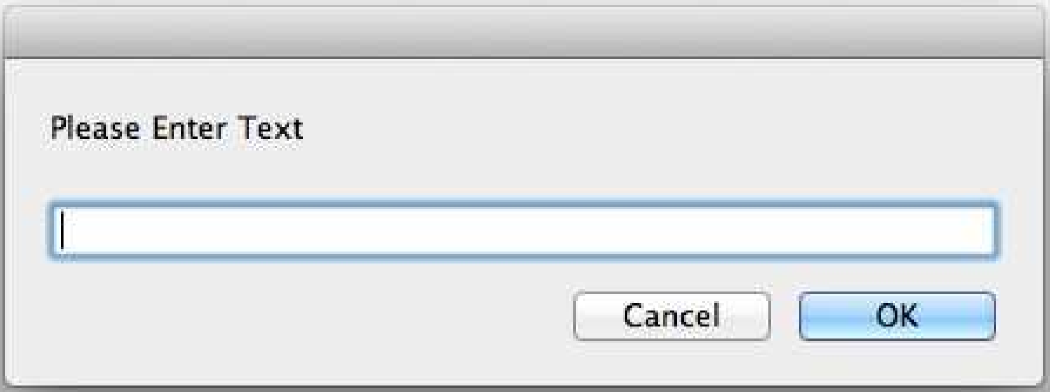
- The middle tier is the layer that the presentation tier and the data tier use to communicate with each other.

Data Tier

- The data tier is basically the server that stores an application's data.

Accepting User Input

Many web applications accept user input from online forms, search boxes, etc. The user is free to type in any ASCII text.



The application interprets that text to generate an appropriate response.

Simple SQLi Example

Scenario 1: an online retailer provides an option to search for products of interest, including those less than a given price.

E.g. to view all products of cost less than $100, the user inputs:

`Products:`	`all`
`Cost below:`	`100`

In response, the interface produces URL:

```
http://www.dupe.com/products.php?val=100
```

Simple SQLi Example

In response, to this http request

```
http://www.dupe.com/products.php?val=100
```

the middle layer code (`products.php`) generates a query to the data layer:

```
SELECT *
FROM Products
WHERE Price < '100'
ORDER BY ProductDescription;
```

Simple SQLi Example (Continued)

But suppose the attacker types:

`Products:`	`all`
`Cost below:`	`100' OR '1'='1`

The system generates the following http request:

```
http://www.dupe.com/products.php?val=100' OR '1'='1
```

Simple SQLi Example (Continued)

A careless middle layer might produce this query for the data layer:

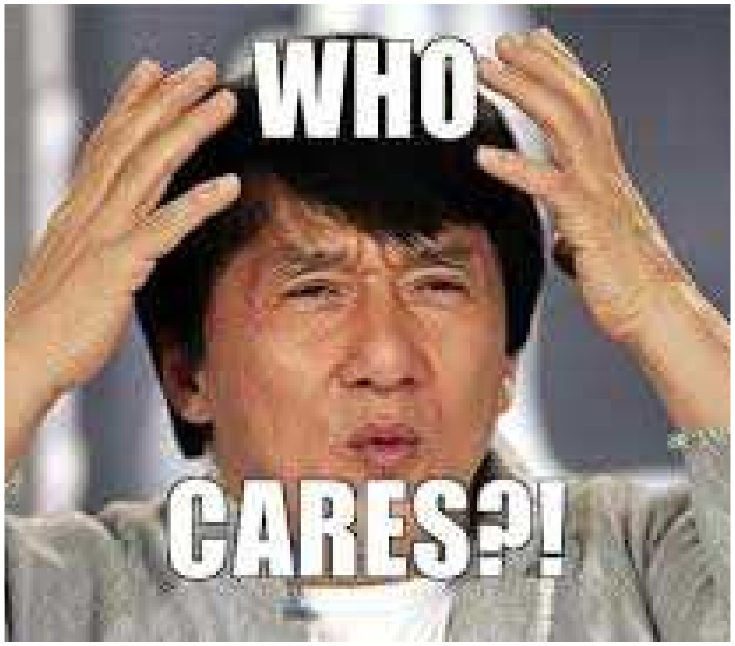
```
SELECT *
FROM Products
WHERE Price < '100' OR '1'='1'
ORDER BY ProductDescription;
```

Now the user sees all products, not just those under $100.

What Went Wrong?

The middle layer accepted user input and incorporated it directly into a database query, *without checking* for acceptability.

But who cares? So what if the user sees all products rather than a select set?



SQLi Example 2

Scenario 2: the retailer's login interface checks username / password against records in the database.

Username:	foo
Password:	bar

From this the presentation layer produces http request:

```
http://www.dupe.com/login.php?user=foo&passwd=bar
```

SQLi Example 2

```
http://www.dupe.com/login.php?user=foo&passwd=bar
```

From this the middle layer (`login.php`) generates:

```
SELECT userid
FROM CMSUsers
WHERE user = 'foo' AND passwd = 'bar';
```

The user is granted access only if the database returns more than zero records.

SQLi Example 2 (Continued)

The attacker types:

`Username:`	`foo`
`Password:`	`any' OR '1'='1`

where foo is any legitimate user.

From this the interface generates http request:

```
http://www.dupe.com/login.php?user=foo&passwd=any' OR '1'='1
```

A careless middle layer could send this query to the data layer:

```
SELECT userid
FROM CMSUsers
WHERE user = 'foo' AND passwd = 'any' OR '1'='1';
```

The attacker is granted access to the system.

SQLi Example 3

Scenario 3: a business allows a logged-in customer to access that customer's data:

```
Get Data for Customer: 17
```

```
http://www.dupe.com/customer.php?userid=17
```

This generates the database query:

```
SELECT *
FROM userinfo
WHERE userid = 17;
```

SQLi Example 3 (Continued)

Instead the customer types:

```
Get Data for Customer: | 17; DROP TABLE users |
```

which generates http request:

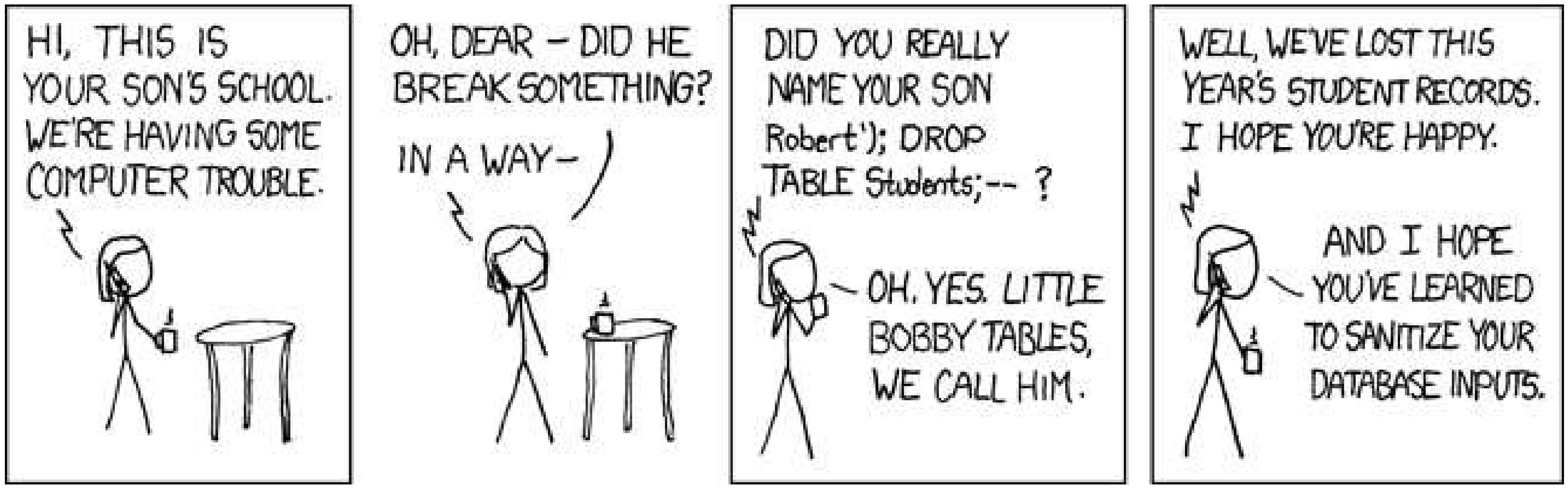
```
http://www.dupe.com/customer.php?userid=1; DROP TABLE users
```

A careless middle layer might send this query to the data layer:

```
SELECT *
FROM userinfo
WHERE userid = 17; DROP TABLE users;
```

Some database system allow multiple statements to be executed with one call in this way.

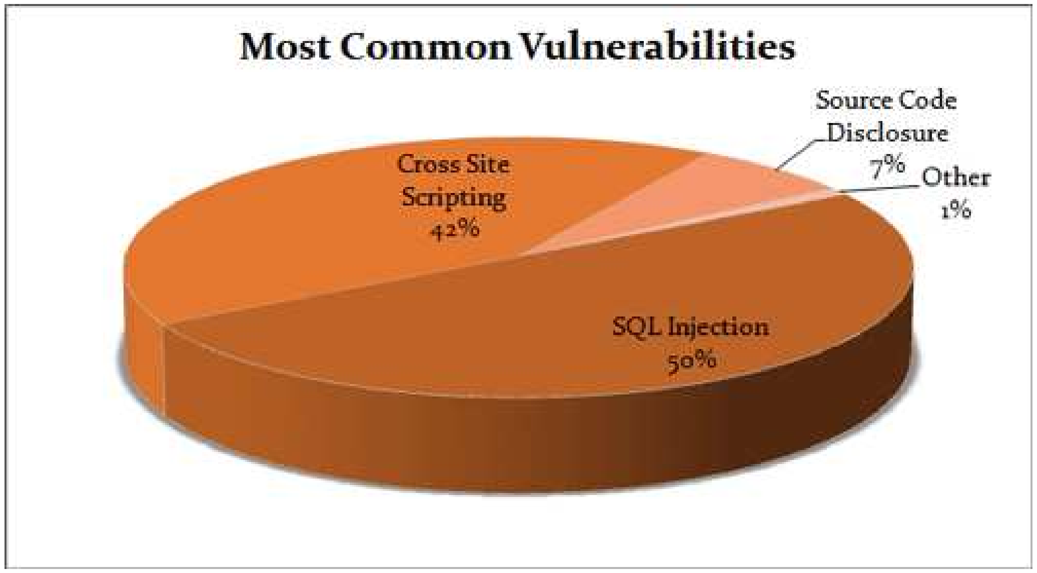
Do You Get It?


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.

What is this cartoon about? Would this attack work?

How Common are They?

The SANS Institute has consistently ranked SQL injection as the most dangerous software error.



Not everyone agrees! Whitehat Security founder Jeremiah Grossman said:

> *"SQL injection, for all the damage that it causes, is actually not in our top 10 when it comes to strict prevalence. It's number 14 at 7 percent of websites."*

Examples of Attacks

- March, 2008; Heartland Payment Systems: 134 million credit cards exposed
- February, 2014; AVS TV: 40,000 accounts
- February, 2014; United Nations Internet Governance Forum: 3,215 account details leaked
- February, 2014; Spirol International: 70,000 user accounts compromised

- August 2014, Hold Security: disclosed theft of confidential information from nearly 420,000 websites

How to Counter SQL Injection

Countering SQLi Flaws

There are two primary means of avoiding SQL injection attacks:

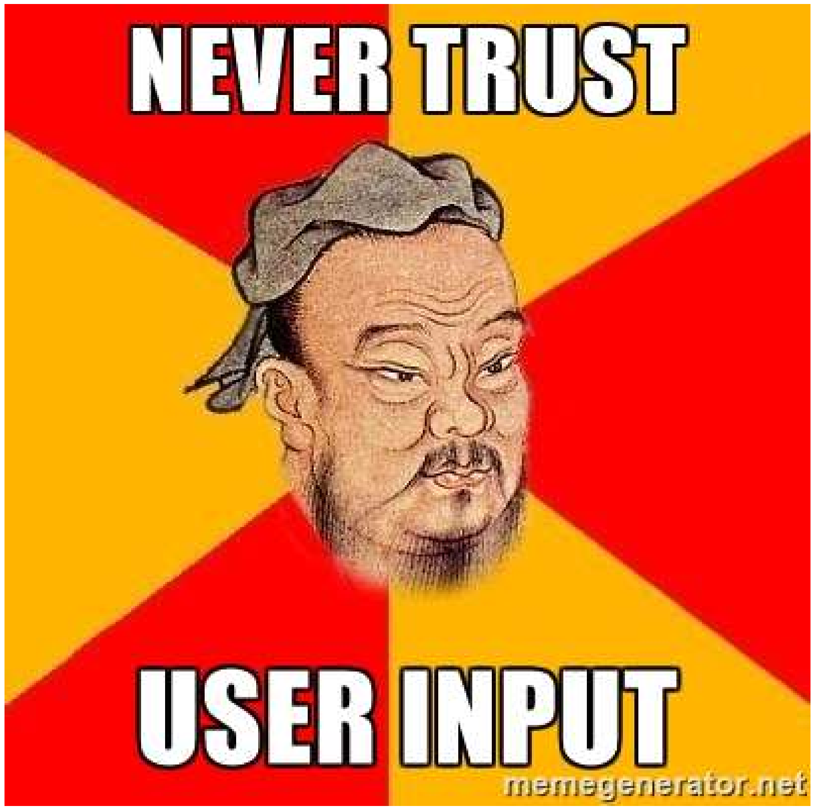
1. Avoid the use of dynamic SQL queries entirely. I.e., don't generate queries on the fly from user supplied inputs.
2. Prevent user-supplied input from affecting query logic by sanitizing it.

Prevent SQL Injection

- **Use parameterized statements:** instead of embedding user input in queries, parameters have associated types and can't be arbitrary text.
- **Enforce least privilege:** minimize the privileges assigned to database accounts. E.g., very few accounts require privilege to delete tables.

Prevent SQL Injection

- **Escape user supplied input:** characters with special meaning to SQL are "escaped" to ensure that they are not treated as code, but as data.
- **Perform input validation:** authenticate user input against a set of predefined rules for length, type, and syntax.



Conclusions

- Cyber attacks are a serious threat to the U.S. and other states.
- The nature of the Internet makes cyber attacks powerful, difficult to counter, and difficult to attribute.
- SQL Injection is one method of cyberattack that is pervasive and extremely dangerous.
- There are good technical solutions you can employ to protect your data.

- Treaties and legal frameworks have not kept pace with the threat.
- Promising theories and approaches are developing to help the international community cope.