Data Protection Case Study: SQL Injection (SQLI) and Prevention

CS4HS Cybersecurity Virtual Workshop
August 2020 Dr. Gary Hu

Agenda

- What is SQL Injection Attack
  - Structure of Relational database
  - SQL (Structural Query Language)
  - SQL statements
  - Logical Expression

- SQLi Prevention Strategies
  - Parametrized queries
  - Input validation

- Demo & Lab session

Injection Attacks on the rise

- **Open Web Application Security Project** (https://owasp.org/)
  - Non-profit community dedicated to improve the security of software.
- Top-10 Web Application Security risks
  - https://owasp.org/www-project-top-ten/
    - Feb 21, 2020
- Injection
  - Injection flaws occur when untrusted data is sent to an system as part of a command or query.
    - The attacker can trick the system into executing unintended commands or accessing data without proper authorization
  - such as SQL, NoSQL, OS, and LDAP injection.

Top 10 Web Application Security Risks 2020
1. **Injection**
2. **Broken Authentication**
3. **Sensitive Data Exposure**
4. **XML External Entities (XXE)**
5. **Broken Access Control**
6. **Security Misconfiguration**
7. **Cross-Site Scripting XSS**
8. **Insecure Deserialization**
9. **Using Components with Known Vulnerabilities**
10. **Insufficient Logging & Monitoring**

News of SQL Injection attacks

- In February 2002,
  - Jeremiah Jacks discovered Guess.com was vulnerable to an SQL injection attack
  - Pull down 200,000+ names, credit card info in customer database
- On November 1, 2005
  - a teenage hacker broke into the site of a Taiwanese information security magazine
  - steal customers' information.

- On August 17, 2009
  - Department of Justice charged Albert Gonzalez for reportedly "the biggest case of identity theft in American history as of 2009"
  - 170 million credit card numbers using an SQL injection attack

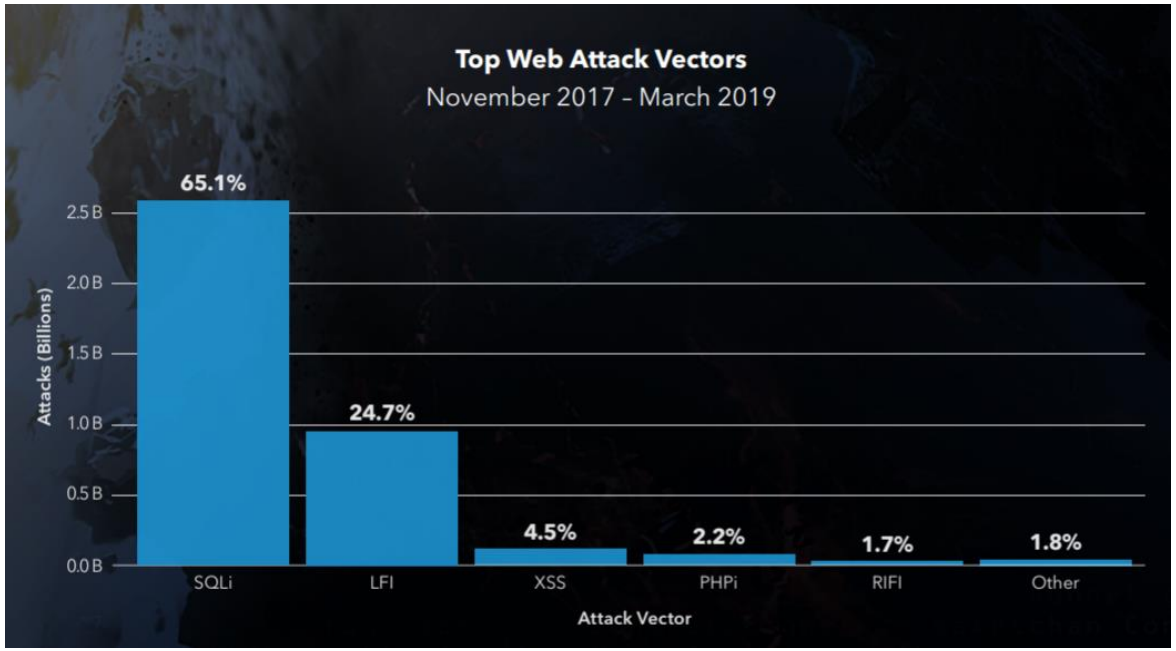
- On February 21, 2014
  - United Nations Internet Governance Forum, 3,215 account details leaked.
- On March 7, 2014
  - Johns Hopkins University Biomedical Engineering Servers, personal details of 878 students and staff
- In October 2015
  - British telecommunications company Talk Talk's servers
  - 156,959 customers personal details

...

- On **July 3,2020**
  - 7.5M customer records stolen from financial service Dave Inc.

Injection Attacks on the rise

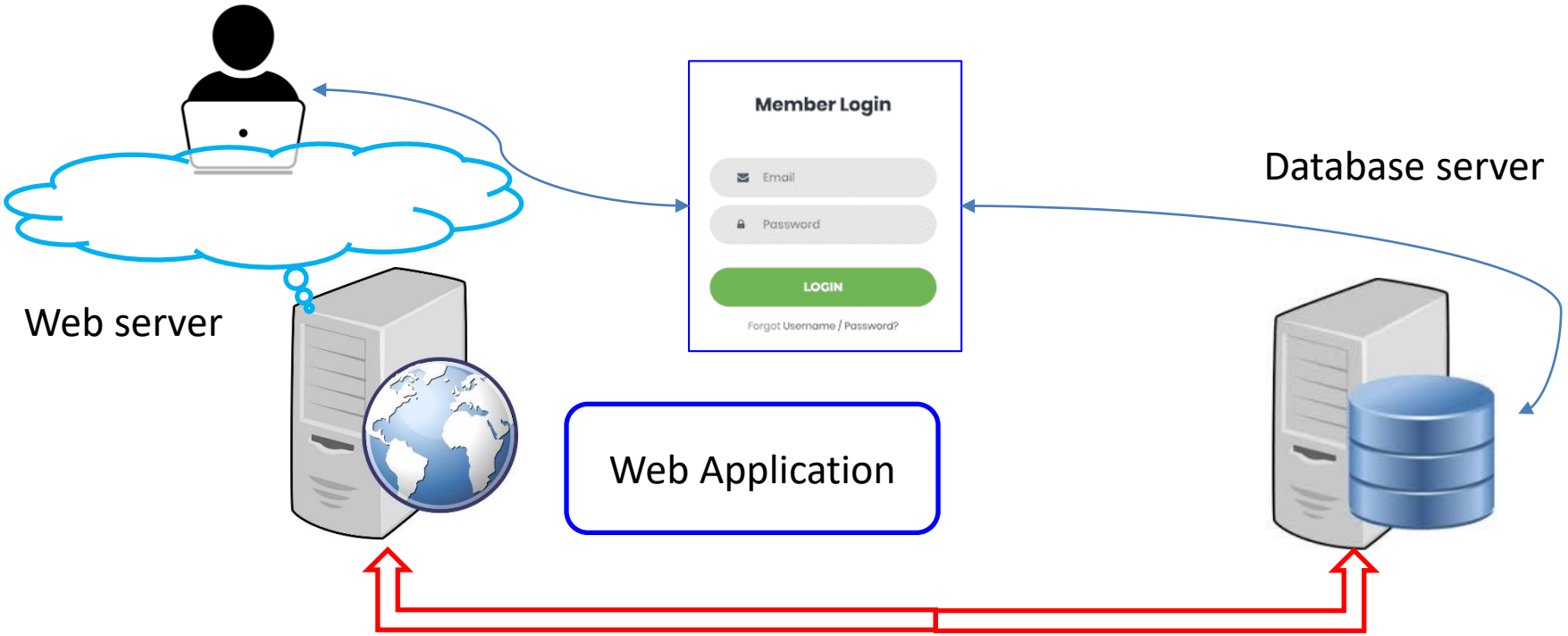
- Nov. 2017 ~ March 2019
  - 4 billion (3.993) web attack alerts
    - 1.23 billion of these occurring in the first quarter of 2019 alone.
  - SQL injection attacks accounted for 65% of web-based attacks



- Video
- https://www.youtube.com/watch?v=rWHvp7rUka8&t=220s

What is a SQL Injection Attack?

- Architecture of Web Applications



What is a SQL Injection Attack?

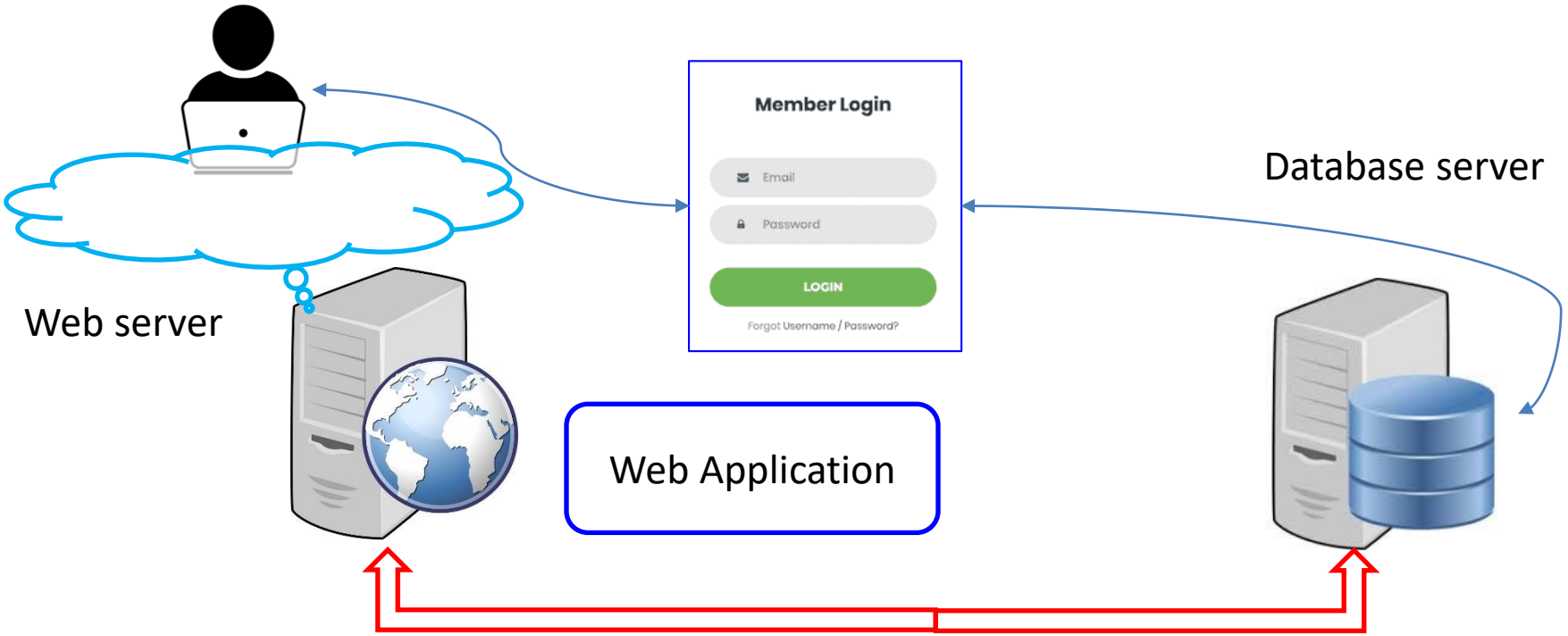
- Code injection technique

- Many web applications take user input from a form
  - Often this user input is used literally in the construction of a SQL query submitted to a database.

  - A SQL injection attack involves placing SQL statements in the user input

What is a SQL Injection Attack?

- Architecture of Web Applications

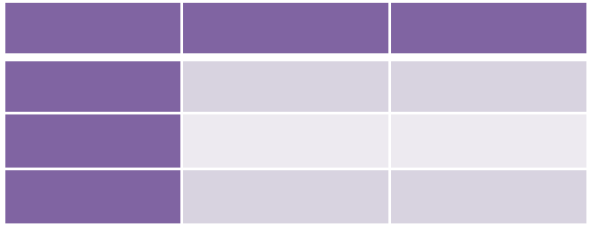


Relational Database

- Consists of many 2D tables

CustomerID	CustomerName	ContactName	Address	City	ZIP	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931



… …

Data Structure

Columns: Attributes

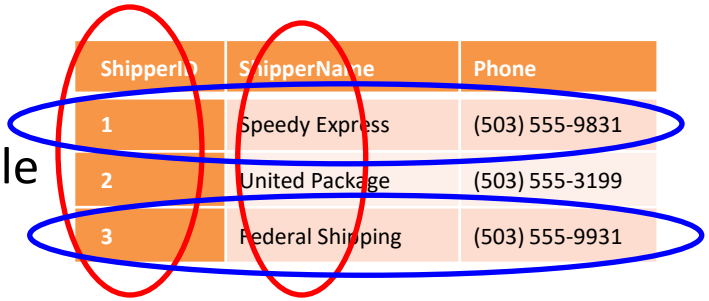
- Example: **Customer** Table

CID	CustName	ContactName	Address	City	ZIP	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany

Rows: Records/Tupples

Queries on a Single Table

- Get everything from a table
- Projection
  - Retrieve several attributes from a table



ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

- Selection
  - Retrieve some records from a table

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

- Combine both
  - Retrieve some attributes and some records from a table

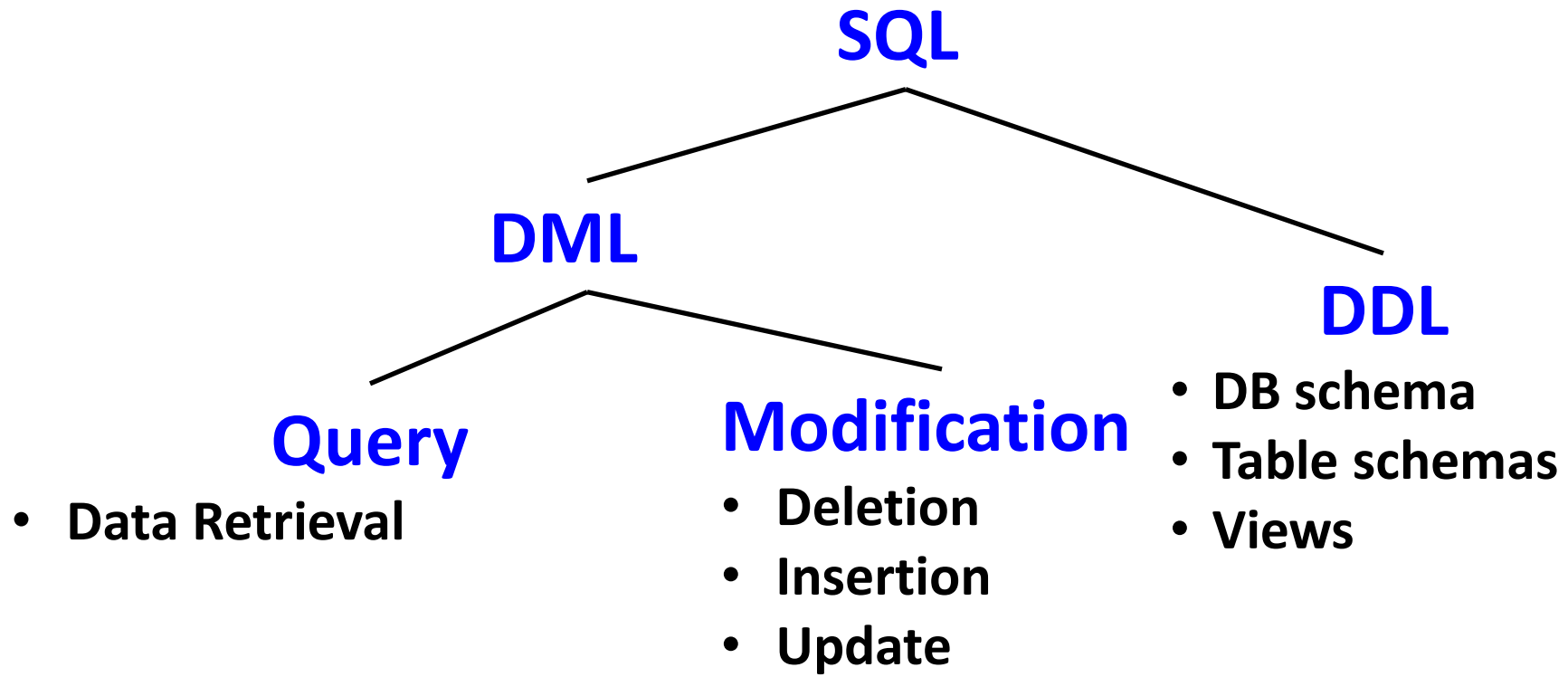
SQL Introduction

Standard language for querying and manipulating data

Structured Query Language

Many standards out there:

- ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3),
- Vendors support various subsets
- When starting to work with a specific database, make sure you are aware of the specific SQL form/syntax in use and the features supported


SQL
DML
DDL
Query
Modification
• Data Retrieval
• Deletion
• Insertion
• Update
• DB schema
• Table schemas
• Views

SQL Query

- Basic SELECT Statement :

```
Select A1,A2,…,An
From   T1,T2, …,Tm
Where  condition
```

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

Practice/LAB

https://www.w3schools.com/sql/trysql.asp?filename=trysql_op_in

- Get everything from the customers table

  Select * from customers
- Get everything from employees table
- Get everything from products table
- Projection operations

  Select productname, price from customers

Selection

```
Select A1,A2,…,An
From   Table
Where  condition
```

(Logical expression)

Find some cheap/expensive products

What is SQL Injection Attack

- Logical Expression
  - Boolean Algebra
    - Logical values (**true** and **false**)
    - Named after British mathematician George Boole

Three basic Boolean Operations

- NOT operation
  - Not T = F Not F = T
- AND operation
  - F and F = F F and T = F
  - T and F = F T and T = T
- OR operation
  - F or F = F F or T = T
  - T or F = T T or T = T

True => 1 **False =0**

and => * **or** => +

Do you agree?

- If **A** is False, anything and **A** is False.

- If **A** is True, anything or **A** is True.

- Questions:

  A or True = ?

  A and B or True=?

Practice

- Find all USA customers
- Find orders with large quantity
- Find some senior employees
- Find products its price is between 10~20 dollars
- …

Products

ProductID	ProductName	...	CategoryID	Price
1	Chais	..	1	18
2	Chang		1	19
3	Aniseed Syrup		2	10
4	Chef Anton's Cajun Seasoning		2	22
5	Chef Anton's Gumbo Mix		2	21.35
6	Grandma's Boysenberry Spread		2	25
7	Uncle Bob's Organic Dried Pears		7	30

Select * from products where price > 10 and price <20

- Malicious SQL Injection Statements
  - SELECT * FROM customers where PostalCode= ‘123456' or 1=1

Other injection possibilities

- Using SQL injections, attackers can:
  - Add new data to the database
    - Could be embarrassing to find yourself selling politically incorrect items on an eCommerce site
    - Perform an INSERT in the injected SQL
  - Modify data currently in the database
    - Could be very costly to have an expensive item suddenly be deeply ‘discounted’
    - Perform an UPDATE in the injected SQL
  - Perform dangerous operations on system tables
    - Eg. DROP TABLE users

Prevention

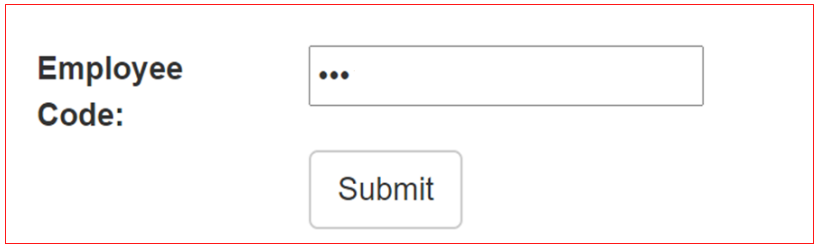
- Parametrized queries
- Input validation

PHP Code without Defense

```
…
$sqlQuery="SELECT e.FIRST_NAME,e.LAST_NAME,
                  d.DEPARTMENT_NAME, d.MANAGER_ID
                  from employees e,departments d
                  where d.DEPARTMENT_ID=e.DEPARTMENT_ID and
                  e.EMPLOYEE_ID=".$employeeCode." ";

$stid = oci_parse($sqlConnection, $sqlQuery);
oci_execute($stid);
…
```

SELECT e.FIRST_NAME,e.LAST_NAME,d.DEPARTMENT_NAME,d.MANAGER_ID from employees e,departments d where d.DEPARTMENT_ID=e.DEPARTMENT_ID and e.EMPLOYEE_ID=**101**



888 Or 1=1

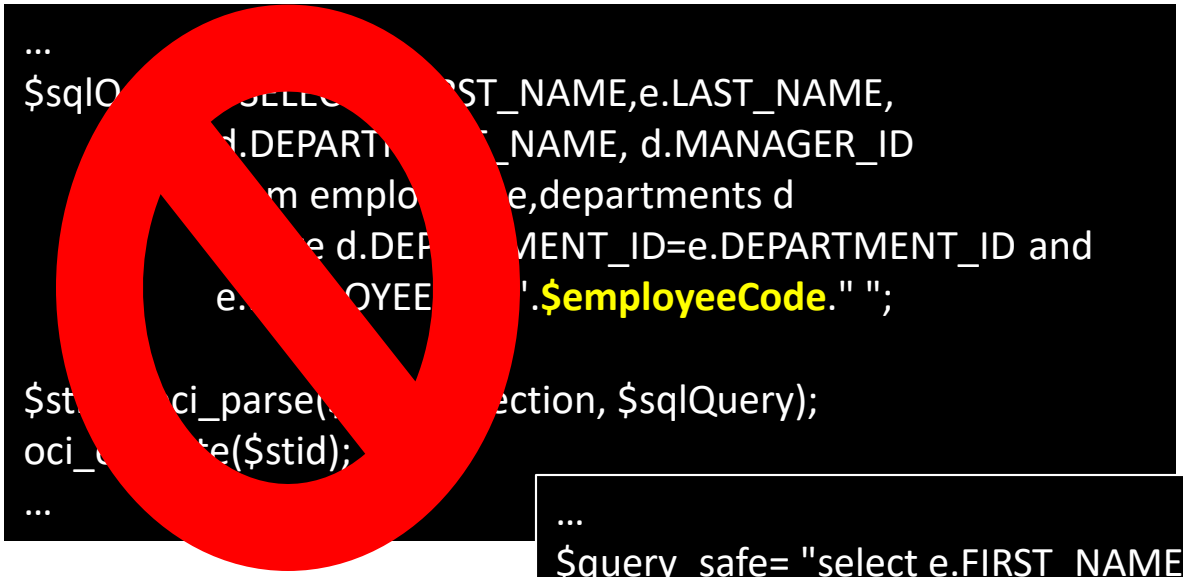
Code with Defense

- Queries with Parameter Binding
  - User inputs are not directly combined with the query statement
  - Define placeholders in queries
  - Bind the variables and placeholders by placeholder names
- Binding is important for database performance and also as a way to avoid SQL Injection security issues.

```
…
$query_safe= "select e.FIRST_NAME,e.LAST_NAME,
              d.DEPARTMENT_NAME,d.MANAGER_ID
              from employees e,departments d
              where d.DEPARTMENT_ID=e.DEPARTMENT_ID and
              e.EMPLOYEE_ID= :empCode ";

 $stid = oci_parse($sqlConnection, $sqlQuery);
  oci_bind_by_name($stid, ': empCode', $employeeCode);
  oci_execute($stid);
…
```

- Function oci_bind_by_name (…)
  - A bridge between variables and SQL statement
  - Able to block unwanted input values




```
...
$query_safe= "select e.FIRST_NAME,e.LAST_NAME,
              d.DEPARTMENT_NAME,d.MANAGER_ID
              from employees e,departments d
              where d.DEPARTMENT_ID=e.DEPARTMENT_ID and
              e.EMPLOYEE_ID= :empCode ";

 $stid = oci_parse($sqlConnection, $sqlQuery);
 oci_bind_by_name($stid, ': empCode', $employeeCode);
 oci_execute($stid);
...
```

More Defenses

- Input validation
  - Many classes of input have fixed formats
    - Email addresses, dates, part numbers, etc.
    - Verify that the input is a valid string in a certain format
    - Exclude problematic characters (eg. *, quotes, semicolons or #)
- Have length limits on input
  - Many SQL injection attacks depend on entering long strings

Even More Defenses

- Scan query string for undesirable word combinations that indicate SQL statements
  - INSERT, DROP, etc.
  - If you see these, can check against SQL syntax to see if they represent a statement or valid user input
- Limit database permissions and segregate users
  - If you're only reading the database, connect to database as a user that only has read permissions
  - Never connect as a database administrator in your web application

And Yet More Defenses

- Configure database error reporting
  - Default error reporting often gives away information that is valuable for attackers (table name, field name, etc.)
  - Configure so that this information is never exposed to users

Summary

- Injection Attack has been one of top data security risks for decads, and still on the rise
- It is related to user inputs and code technique
- SQL Injection is preventable
  - Good coding habits
  - Solid Input validation
  - Diligent server configuration