VULNERABILITY DECODER

SQL INJECTION

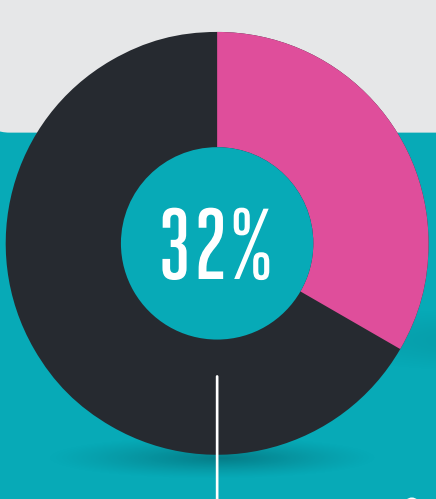
SQL injection (SQLi) is a high-severity vulnerability. Attackers can exploit SQLi vulnerabilities to access or delete data from the database and do other undesirable things.

WHAT IS SQL INJECTION?

A SQL query is one way an application talks to the database.

SQL injection occurs when an application fails to sanitize untrusted data (such as data in web form fields) in a database query.

An attacker can use specially-crafted SQL commands to trick the application into asking the database to execute unexpected commands.



One-third of web applications have at least one SQL injection vulnerability, according to Veracode's *State of Software Security Report.*

ATTACKERS CAN EXPLOIT SQL INJECTION VULNERABILITIES TO:

Control an application's data-driven behavior.
For example, tricking an application into allowing a login without a valid password.

Alter data in the database without authorization.
For example, creating fraudulent records, adding users or promoting users to higher access levels, or deleting data.

Access data without authorization.
For example, tricking the database into providing too many results for a query.

ANATOMY OF A SQL INJECTION ATTACK

A SQL query includes an *argument,* which tells the database to return only the desired records. The *value* for that argument can be provided by a user (in a form field, URL parameter, web cookie, etc.).

A SQL INJECTION ATTACK HAS TWO STAGES:

INJECTION ATTACK STAGE 1

Reconnaissance
Attacker tries submitting various unexpected values for the argument and observes how the application responds.

INJECTION ATTACK STAGE 2

Attack
Attacker provides a carefully-crafted input value that will be interpreted as part of a SQL command rather than merely data; the database then executes the SQL command as modified by the attacker.

ATTACKER

Automation
Reconnaissance and attack stages can be automated by readily-available tools.

THE RISK: DATA LEAKAGE

Some very large and devastating data breaches have been the result of SQL injection attacks. Here are a few recent examples and their consequences.

MOSSACK FONSECA

WHAT	HOW	RESULT
"The Panama Papers" — 11.5 million files and 2.6 TB of secret data — stolen from Panamanian law firm and leaked to world media.	Attacker may have exploited a customer web portal running a version of Drupal with a SQL injection flaw.	Many of the world's rich and powerful are implicated in tax avoidance schemes.

WORLD ANTI-DOPING AGENCY (WADA)

WHAT	HOW	RESULT
International anti-doping group targeted by Russia-linked espionage group.	In a two-pronged attack, attackers used SQL injection to steal email addresses and passwords from WADA's servers, then used spearphishing to steal staff credentials to a system containing private medical records.	American athletes exposed for taking banned substances for approved medical reasons.

PHILIPPINES COMMISSION ON ELECTIONS (COMELEC)

VOTE

WHAT	HOW	RESULT
Personal information on every registered voter in the Philippines — 55 million people — leaked online.	Hackers affiliated with the Anonymous hacktivist group used SQL injection to query data from a MySQL database.	Leaked data included detailed biometric and statistical information that could be used for impersonation and fraud.

QATAR NATIONAL BANK

WHAT	HOW	RESULT
1.4 GB-worth of information leaked on members of Qatari royal family, government and military officials and prominent journalists.	Hackers used the sqlmap pen-testing tool to steal data from the Oracle back-end database.	Criminals attempted to use leaked credentials to access bank and social media accounts.

SAMPLE SQL INJECTION: BREAKING THE BANK

The following hypothetical example shows how a SQL injection vulnerability could be exploited by an attacker to access all bank account numbers and balances from a database.

LOOKING UP AN ACCOUNT BALANCE

When you access your bank account online, the database query might look like this (in Java):

```
String accountBalanceQuery =
    "SELECT accountNumber, balance FROM accounts WHERE account_owner_id = "
    + request.getParameter("user_id");

try {
    Statement statement = connection.createStatement();
    ResultSet rs = statement.executeQuery(accountBalanceQuery);
    while (rs.next()) {
        page.addTableRow(rs.getInt("accountNumber"), rs.getFloat("balance"));
    }
} catch (SQLException e) { ... }
```

EXAMPLE QUERY:

If you have the user ID 984, when you're logged in you might visit the URL:
bankingwebsite/show_balances?user_id=984

The accountBalanceQuery passed to the database would end up being:

```
SELECT accountNumber, balance FROM accounts WHERE account_owner_id = 984
```

RESULT: The database returns any account numbers and balances for user ID 984.

SQL INJECTION ATTACK ON THE BANK WEBSITE

The attacker could change the parameter "user_id" to be interpreted as:

```
0 OR 1=1
```

And this results in accountBalanceQuery being:

```
SELECT accountNumber, balance FROM accounts WHERE account_owner_id = 0 OR 1=1
```

Because 1=1 in all cases, when this query is passed to the database, it will return all the account numbers and balances it has stored.

RESULT: The attacker now knows every user's account numbers and balances.

HOW TO REPAIR THE VULNERABLE CODE

A developer could easily repair this vulnerability by using a prepared statement to create a parameterized query as below:

```
String accountBalanceQuery =
    "SELECT accountNumber, balance FROM accounts WHERE account_owner_id = ?";

try {
    PreparedStatement statement = connection.prepareStatement(accountBalanceQuery);
    statement.setInt(1, request.getParameter("user_id"));
    ResultSet rs = statement.executeQuery();
    while (rs.next()) {
    page.addTableRow(rs.getInt("accountNumber"), rs.getFloat("balance"));
    }
} catch (SQLException e) { ... }
```

RESULT: If an attacker attempts to supply a value that's not a simple integer, then statement.setInt() will throw a SQLException error rather than permitting the query to complete.

PREVENTING SQL INJECTION ATTACKS

SQL injection is a common but avoidable vulnerability. Developers can follow these best practices to avoid SQLi vulnerabilities and limit the damage they can cause.

1 **Discover**
Discover SQLi vulnerabilities by routinely testing your applications using both static and dynamic testing.

2 **Repair**
Avoid and repair SQLi vulnerabilities by using parameterized queries.
These types of queries specify placeholders for parameters, so the database treats them as data rather than part of a SQL command.
Prepared statements and object-relational mappers (ORMs) make this easy for developers.

3 **Remediate**
Remediate SQLi vulnerabilities by escaping inputs before adding them to the query.
Use this technique only where prepared statements are unavailable.

4 **Mitigate**
Mitigate the impact of SQLi vulnerabilities by enforcing least privilege for accessing the database.

SMART DEVELOPERS, SECURE DEVELOPMENT

See how to build security into every stage of your software development lifecycle.

Five Principles for Securing DevOps
DOWNLOAD THE FREE GUIDE AT VERACODE.COM/DEVOPS5

VERACODE
SECURING THE SOFTWARE THAT POWERS YOUR WORLD.

Sources

Mossack Fonseca: "The security flaws at the heart of the Panama Papers," Wired, April 6, 2016, www.wired.co.uk/article/panama-papers-mossack-fonseca-website-security-problems.

WADA: "WADA confirms attack by Russian cyber-espionage group," WADA, September 13, 2016, www.wada-ama.org/en/media/news/2016-09/wada-confirms-attack-by-russian-cyber-espionage-group; "World Anti-Doping Agency site hacked, thousands of accounts leaked," Hackread, August 12, 2016, www.hackread.com/world-anti-doping-agency-site-hacked/.

COMELEC: "When a nation is hacked: Understanding the ginormous Philippines data breach," Troy Hunt, April 13, 2016, www.troyhunt.com/when-nation-is-hacked-understanding/, "Int'l web security expert slams Comelec...," GMA News, April 22, 2016, www.gmanetwork.com/news/story/563633/scitech/technology/int-l-web-security-expert-slams-comelec-for-slow-acknowledgment-of-data-hack.

Qatar National Bank: "Qatar National Bank leak: Security experts hint 'SQL injection' used in database hack," International Business Times, April 27, 2016, www.ibtimes.co.uk/qatar-national-bank-leak-security-experts-hint-sql-injection-used-database-hack-1557069.

LEARN MORE AT
VERACODE.COM