CONFIGURATION GUIDES › ADMINISTRATION & SECURITY

SQL injection detection in Applicare

Applicare 9.2 · Updated February 24, 2025 · **Pending technical review**

SQL injection is a type of cyberattack where malicious SQL code is inserted into a web application's input fields, which then interferes with the application's database queries. This can allow attackers to view, modify, or delete data that they are not authorized to access.

How it works:

1. **Vulnerable Application:** A web application is vulnerable to SQL injection if it directly incorporates user-supplied input into SQL queries without proper sanitization or parameterization.
2. **Malicious Input:** An attacker crafts malicious SQL code and inserts it into an input field, such as a search box or login form.
3. **Query Manipulation:** When the application processes the input, the malicious SQL code becomes part of the database query, altering its intended logic.
4. **Unauthorized Access:** The modified query can grant the attacker access to sensitive data, allow them to modify or delete data, or even take control of the database server.

Example:

Imagine a web application with a login form. The application constructs an SQL query like this:

SQL

```
TEXT
SELECT * FROM users WHERE username = '$username' AND password = '$password';
```

If an attacker enters the following username:

```
TEXT
' OR '1'='1
```

The resulting query becomes:

SQL

```
TEXT
SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '$password';
```

Since `'1'='1'` is always true, this query will return all users in the database, effectively bypassing authentication.

Impact:

SQL injection attacks can have severe consequences, including:

- **Data breaches:** Attackers can steal sensitive data, such as customer information, financial records, or intellectual property.
- **Data manipulation:** Attackers can modify or delete data, leading to data corruption or loss.
- **Denial of service:** Attackers can disrupt the availability of the application or the database server.
- **Complete system compromise:** In some cases, attackers can gain complete control of the database server and potentially other systems on the network.

Prevention:

The most effective way to prevent SQL injection is to use parameterized queries or prepared statements. These techniques treat user input as data rather than executable code, which prevents attackers from injecting malicious SQL code. Other prevention measures include input validation, least privilege, and web application firewalls.