

# SQL INJECTION

NHÓM 1



Trong tất cả các cuộc tấn công nhằm vào website, tấn công SQL Injection là một trong những loại nguy hiểm và phổ biến nhất, nó đã gây ra những thiệt hại đáng kể cho nhiều doanh nghiệp và tổ chức trong những năm qua.



# BKAV

Là một công ty công nghệ hoạt động trong các lĩnh vực an ninh mạng, phần mềm, chính phủ điện tử, sản xuất điện thoại thông minh ...



August 22, 2021 at 02:38 AM This post was last modified: August 22, 2021 at 03:05 AM by chunx

👑 chunx



Tay to

GOD

Posts 42  
Threads 1  
Joined Aug 2021  
Reputation 89



Hi there,  
I'm still here for selling.

\$20k for BKAV Pro source + \$30k for server side code  
\$10k for BKAV Mobile AV source + \$10k for server side code.  
\$20 for BKAV Endpoint Security. + \$10k for server side code  
\$50k for GD5 source code.  
\$100k for AI source code.

So far, 3 copies of Bkav Pro source code has been sold. So, no one could buy this product exclusively anymore.

If you want to buy anything above, please, send the number of XMR appropriate to their price to the address:

47SonhEFSikNjwcmGAnKG6frXzkCkA2aZ7yBe8ourjnnNPDgQnjhwCa89NUyew62AxP7b8Uqqmh8dS2RCzg2E

Then, I will send the source code for the sender.

Because too much of people just ask the price for their curiousness, I will not make a trade for anyone if there's

Note: if u want to buy anything exclusively, the price will be double

Important:

If you have a facebook account, please report this facebook account (belongs to Nguyen Tu Quang) as he co



### Danh sách hội thoại



2h



3h



7h



20h



H qua



Hous



H qua



H qua



H qua

ΣΟΠ VIN

anh em kiểm tra lại thì mã nguồn đấy là module bé, từ 2020, khả năng là anh em gửi qua chat cho nhau nên đang nghi lấy từ vala



ANM đang lập taskforce điều tra rồi



QuangNT



Nó nói ở đâu

SonVN



nó nói trên whitehat

QuangNT



Cụ thể như này

SonVN



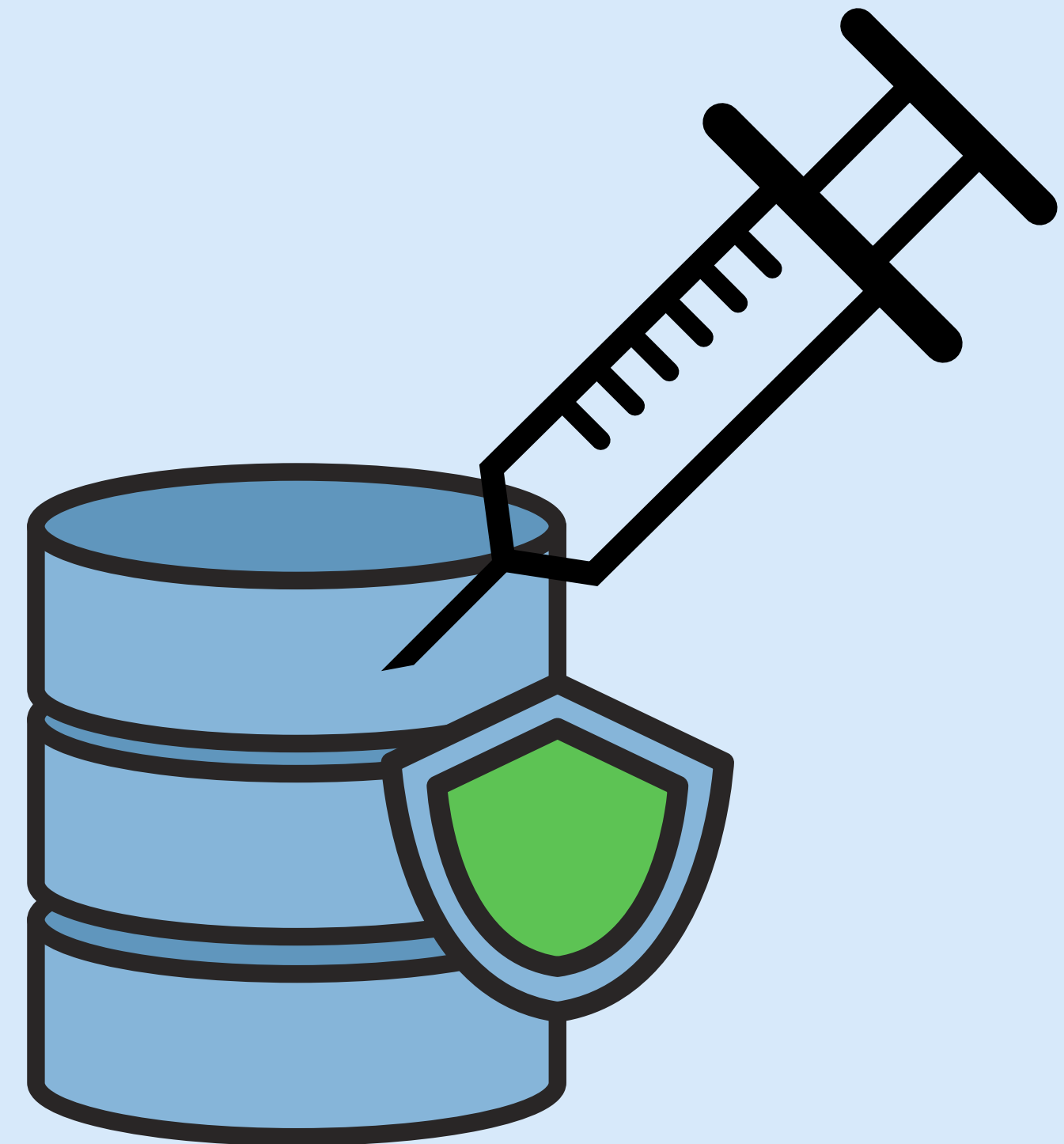


SQL



Injection??

- Là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp



SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update, v.v. trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy.

```
SELECT * FROM items  
WHERE owner = 'wiley'  
AND itemname = 'name' OR 'a'='a';
```

```
SELECT * FROM items  
WHERE owner = 'hacker'  
AND itemname = 'name';
```

```
DELETE FROM items;
```

```
--'
```



---

# Các lỗi thường gặp

---

ERROR



# 1. Không kiểm tra ký tự thoát truy vấn

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

+

```
a' or 't'='t
```

=

```
SELECT * FROM users WHERE name = 'a' or 't'='t';
```

## 2. Xử lý không đúng kiểu

Lỗi SQL injection dạng này thường xảy ra do lập trình viên hay người dùng định nghĩa đầu vào dữ liệu không rõ ràng hoặc thiếu bước kiểm tra và lọc kiểu dữ liệu đầu vào.

```
statement:= "SELECT * FROM data WHERE id = " + a_variable + ";"
```

```
statement:= "SELECT * FROM data WHERE id = " + a_variable + ";"
```

```
1;DROP TABLE users
```

```
SELECT * FROM data WHERE id=1;DROP TABLE users;
```

### 3. Blind SQL Injection

Là một kiểu tấn công SQL injection truy vấn cơ sở dữ liệu sử dụng các mệnh đề để đoán biết. Cách tấn công này thường được sử dụng khi mà một ứng dụng (web, apps) được cấu hình để chỉ hiển thị những thông báo lỗi chung chung, không hiển thị ra lỗi của SQL.

1. Thay đổi giá trị điều kiện truy vấn
2. Điều kiện lỗi
3. Thời gian trễ

```
http://www.shop-online.com/product_detail.php?id=1
```

```
SELECT * FROM products WHERE id = 1
```

```
http://www.shop-online.com/product_detail.php?id=1 and 1 = 2
```

```
http://www.shop-online.com/product_detail.php?id=1 and 1 = 1
```



---

# Các cách phòng chống SQL injection

---

- CÁCH 1 Validation Input
- CÁCH 2 Parameterized Statements
- CÁCH 3 Stored Procedures
- CÁCH 4 Escaping

---

# Validation Inputs

Kiểm tra thông tin và các giá trị người dùng nhập trước khi cập nhật vào database. Bao gồm các hàm sau:

- + `is_email($email)`: Kiểm tra địa chỉ email có hợp lệ hay không.
- + `intval($input)`: Lấy giá trị số nguyên của biến nhập vào.
- + `esc_url($input)`: Hàm này sẽ kiểm tra đường dẫn người dùng nhập vào có đúng không

```
<?php

if(isset($_POST["selRating"]))

{

    $number = $_POST["selRating"];

    if((is_numeric($number)) && ($number > 0) && ($number < 6))

    {

        echo "Selected rating: " . $number;

    }

    else


        echo "The rating has to be a number between 1 and 5!";

}
```

---

# Parameterized Statements

Đảm bảo rằng các tham số (tức là đầu vào) được chuyển vào các câu lệnh SQL được xử lý một cách an toàn.



```
// Connect to the database.
Connection conn = DriverManager.getConnection(URL, USER, PASS);

// Construct the SQL statement we want to run, specifying the parameter.
String sql = "SELECT * FROM users WHERE email = ?";

// Generate a prepared statement with the placeholder parameter.
PreparedStatement stmt = conn.prepareStatement(sql);

// Bind email value into the statement at parameter index 1.
stmt.setString(1, email);

// Run the query...
ResultSet results = stmt.executeQuery(sql);

while (results.next())
{
    // ...do something with the data returned.
}
```

```
// prepare and bind
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

// set parameters and execute
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();
```

---

# Stored Procedure

Nhóm một hoặc nhiều câu lệnh SQL thành một đơn vị logic. Các lần thực thi tiếp theo cho phép các câu lệnh được tham số hóa tự động (lưu trữ để sử dụng sau này và sử dụng nhiều lần)

```
CREATE PROCEDURE `avg_sal`(out avg_sal decimal)
BEGIN
    select avg(sal) into avg_sal from salary;
END
```



# Escaping

```
<?php

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$mysqli = new mysqli("localhost", "my_user", "my_password", "world");

$city = "'s-Hertogenbosch";

/* this query with escaped $city will work */
$query = sprintf("SELECT CountryCode FROM City WHERE name='%s'",
    $mysqli->real_escape_string($city));
$result = $mysqli->query($query);
printf("Select returned %d rows.\n", $result->num_rows);

/* this query will fail, because we didn't escape $city */
$query = sprintf("SELECT CountryCode FROM City WHERE name='%s'", $city);
$result = $mysqli->query($query);
```

Select returned 1 rows.

Fatal error: Uncaught mysqli\_sql\_exception: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 's-Hertogenbosch'' at line 1 in...

---

# Password Hashing

## PASSWORD HASHING

\*\*\*\*\*



\$12\$dfssa.dsf3848sd

```
password_hash(string $password, string|int|null $algo, array $options = []): string
```

# Một số dạng tấn công thường gặp với ứng dụng web

1. Dạng tấn công vượt qua kiểm tra lúc đăng nhập
2. Dạng tấn công sử dụng câu lệnh SELECT
3. Dạng tấn công sử dụng câu lệnh INSERT
4. Dạng tấn công sử dụng stored-procedures