

SQL Injection

SQL Injection 이란?

입력 값에 **SQL** 문을 주입하여 **비정상적인 동작**을 하도록 하는 행위

(Application에서 사용자의 입력 값을 받아 그대로 SQL 구문으로 사용하는 경우가 많기 때문에 이를 취약점으로 활용한 것)

SQL Injection 이란?

A [SQL injection](#) attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

SQL Injection 이란?

1. 유저로부터 Application에 들어오는 SQL 쿼리 입력 값을 통해 이루어 짐
2. SQL Injection을 통해 DB에 저장된 민감한 데이터를 읽을 수 있음
3. DB에 저장된 값을 수정, 삭제, 삽입 할 수 있음
4. DB를 종료시키는 등 관리자적인 동작을 할 수 있음

SQL Injection 심각도

SQL을 삽입한다

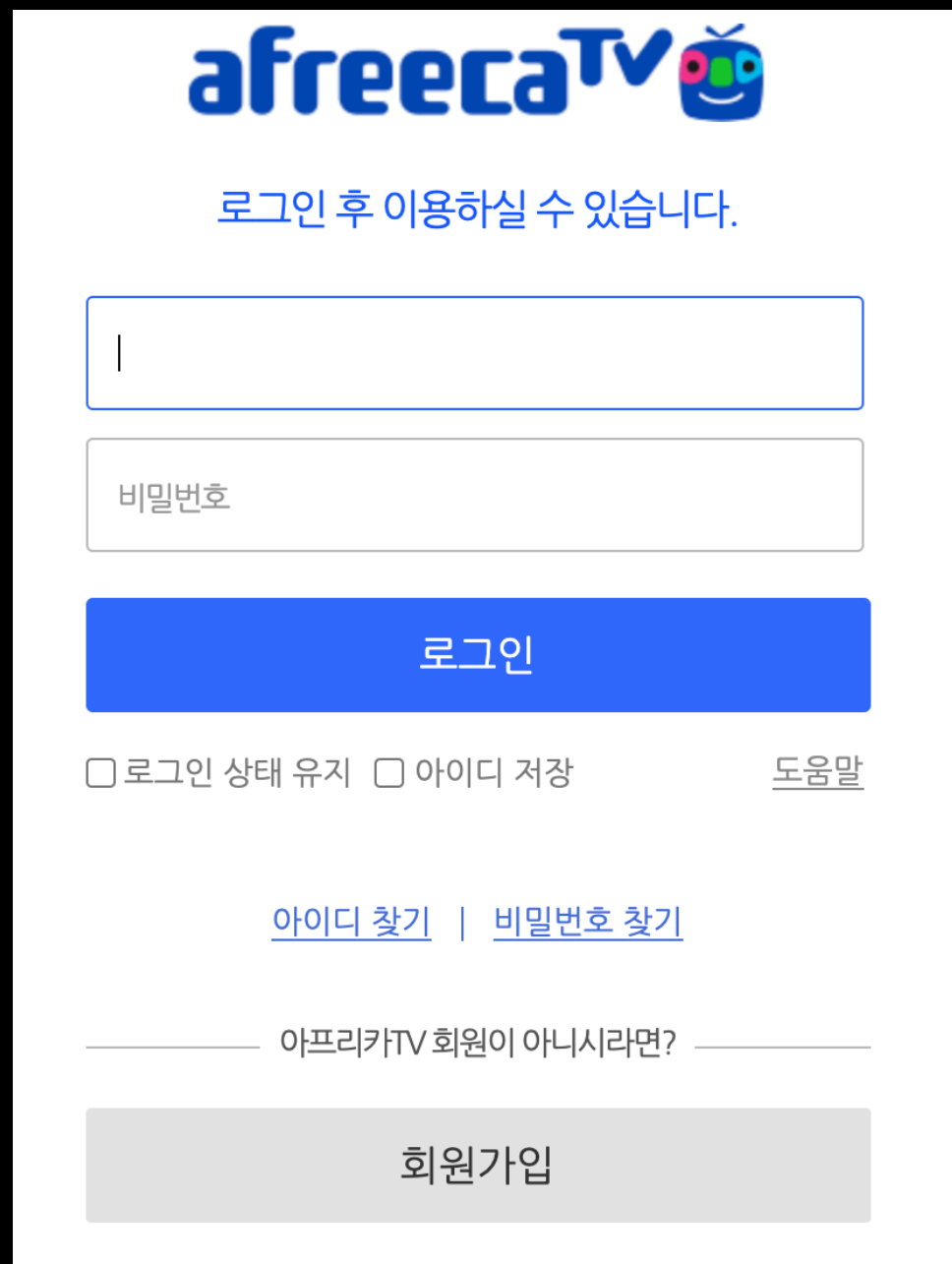


SQL로 할 수 있는건 다할 수 있다



SQL Injection Attacker의 실력 + 상상력에 따라 달라짐

SQL Injection 예시



The screenshot shows the afreecaTV login interface. At the top is the afreecaTV logo. Below it is a message: "로그인 후 이용하실 수 있습니다." (You can use it after login). There are two input fields: the first is for the ID, containing a single vertical bar character '|', and the second is for the password, labeled "비밀번호". Below these fields is a blue "로그인" (Login) button. Under the button are two checkboxes: "로그인 상태 유지" (Keep login state) and "아이디 저장" (Save ID), followed by a link "도움말" (Help). Below these are two links: "아이디 찾기" (Find ID) and "비밀번호 찾기" (Find password). At the bottom, there is a text "아프리카TV 회원이 아니시라면?" (If you are not an AfreecaTV member?) and a grey "회원가입" (Sign up) button.

```
SELECT
    nickname
    , starballoon
FROM user_tbl
WHERE
    user_id = ?
    AND user_pw = ?
```

SQL Injection 예시



로그인 후 이용하실 수 있습니다.

radi' OR '1'='1

.....

로그인

☐ 로그인 상태 유지 ☐ 아이디 저장

[도움말](#)

[아이디 찾기](#) | [비밀번호 찾기](#)

아프리카TV 회원이 아니시라면?

회원가입

```
SELECT
    nickname
    , starballoon
FROM user_tbl
WHERE
    user_id = 'radi' OR '1' = '1'
    AND user_pw = ?
```

SQL Injection 예시

보유중인 별풍선

▶ 선물받은 별풍선 내역 확인 및 환전신청은 선물받은 별풍선 / 선물받은 해외 별풍선 메뉴에서 확인할 수 있습니다.

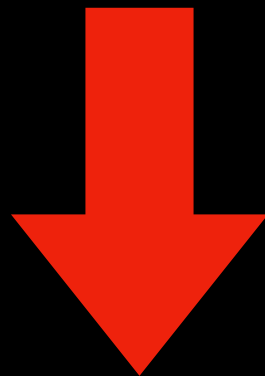
radi0603님이 보유중인 별풍선은 30개 입니다.

- ▶ 이미 선물한 별풍선: 0개
- ▶ 유효기간이 지난 별풍선: 0개 (구매한 별풍선의 유효기간은 5년입니다)

이제 이 별풍선은 제 것 입니다.

SQL Injection 성립 조건

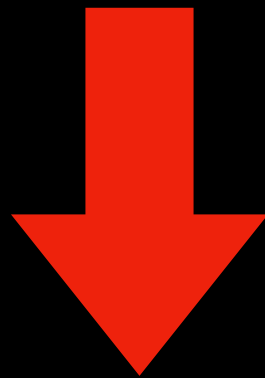
입력 값에 **SQL** 문을 주입하여 **비정상적인 동작**을 하도록 하는 행위



1. 입력 값을 SQL 구문으로 인식하는 경우만 가능
2. 해당 쿼리를 실행할 권한이 있어야 가능

SQL Injection 방어

입력 값의 **SQL** 문



값으로 취급

SQL Injection 방어 - EX 1

PreparedStatement

```
// This should REALLY be validated too
String custname = request.getParameter("customerName");
// Perform input validation to detect attacks
String query = "SELECT account_balance FROM user_data WHERE user_name = ? ";
PreparedStatement pstmt = connection.prepareStatement( query );
pstmt.setString( 1, custname);
ResultSet results = pstmt.executeQuery( );
```

컴파일 과정을 거치면서 문법적 의미를 잃게됨
-> 입력 값으로 들어온 SQL 구문이 조건절 값 자체로 사용 됨

SQL Injection 방어 - EX 2

Stored Procedure

```
// This should REALLY be validated
String custname = request.getParameter("customerName");
try {
    CallableStatement cs = connection.prepareCall("{call sp_getAccountBalance(?)}");
    cs.setString(1, custname);
    ResultSet results = cs.executeQuery();
    // ... result set handling
} catch (SQLException se) {
    // ... logging and error handling
}
```

SQL Injection 방어 - EX 3

Whitelist Input Validation

```
public String someMethod(boolean sortOrder) {  
    String SQLquery = "some SQL ... order by Salary " + (sortOrder ? "ASC" : "DESC");  
    ...  
}
```

사용자 입력을 쿼리를 동적 할 때 그대로 사용하지말라는 뜻
EX) 게시글 오름차순 / 내림차순 정렬할때
ASC / DESC 라는 값을 그대로 받아서 쿼리에 APPEND 하지마라
이렇게 짜놓은데가 있나싶다...

SQL Injection 방어 - EX 4

Escaping All User-Supplied Input

```
Codec ORACLE_CODEC = new OracleCodec();  
String query = "SELECT user_id FROM user_data WHERE user_name = '"  
+ ESAPI.encoder().encodeForSQL( ORACLE_CODEC, req.getParameter("userID"))  
+ "' and user_password = '"  
+ ESAPI.encoder().encodeForSQL( ORACLE_CODEC, req.getParameter("pwd")) + "'";
```

최후의 수단으로 하세요... 어지간하면 쓰지말래여...

End