

Web Hacking

CONTENTS

Contents 1

- Database

Contents 2

- SQL

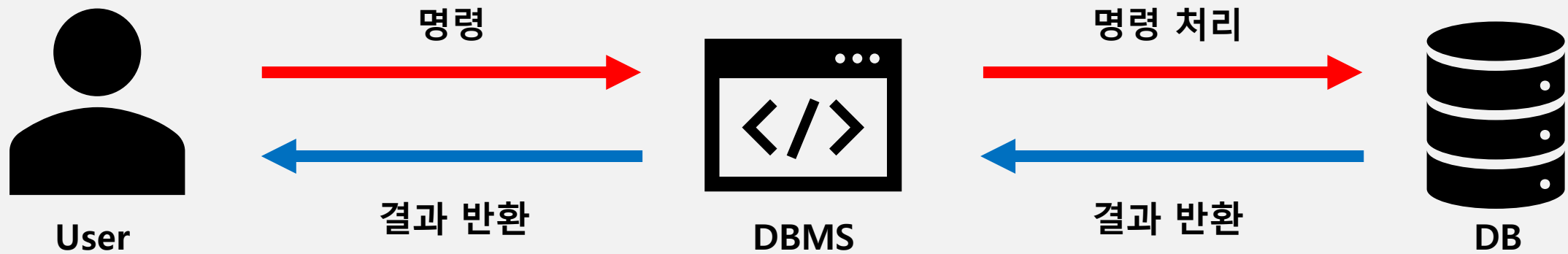
Contents 3

- SQL Injection

- 연관된 데이터들의 집합
- 추가 / 삭제 등으로 인한 변동
- 수 많은 데이터를 효율적으로 관리
- 사이트 내에서 회원 관리 / 게임 정보 관리



Database Management System (DBMS)



- 데이터베이스 관리 시스템

데이터베이스를 제어하는 소프트웨어

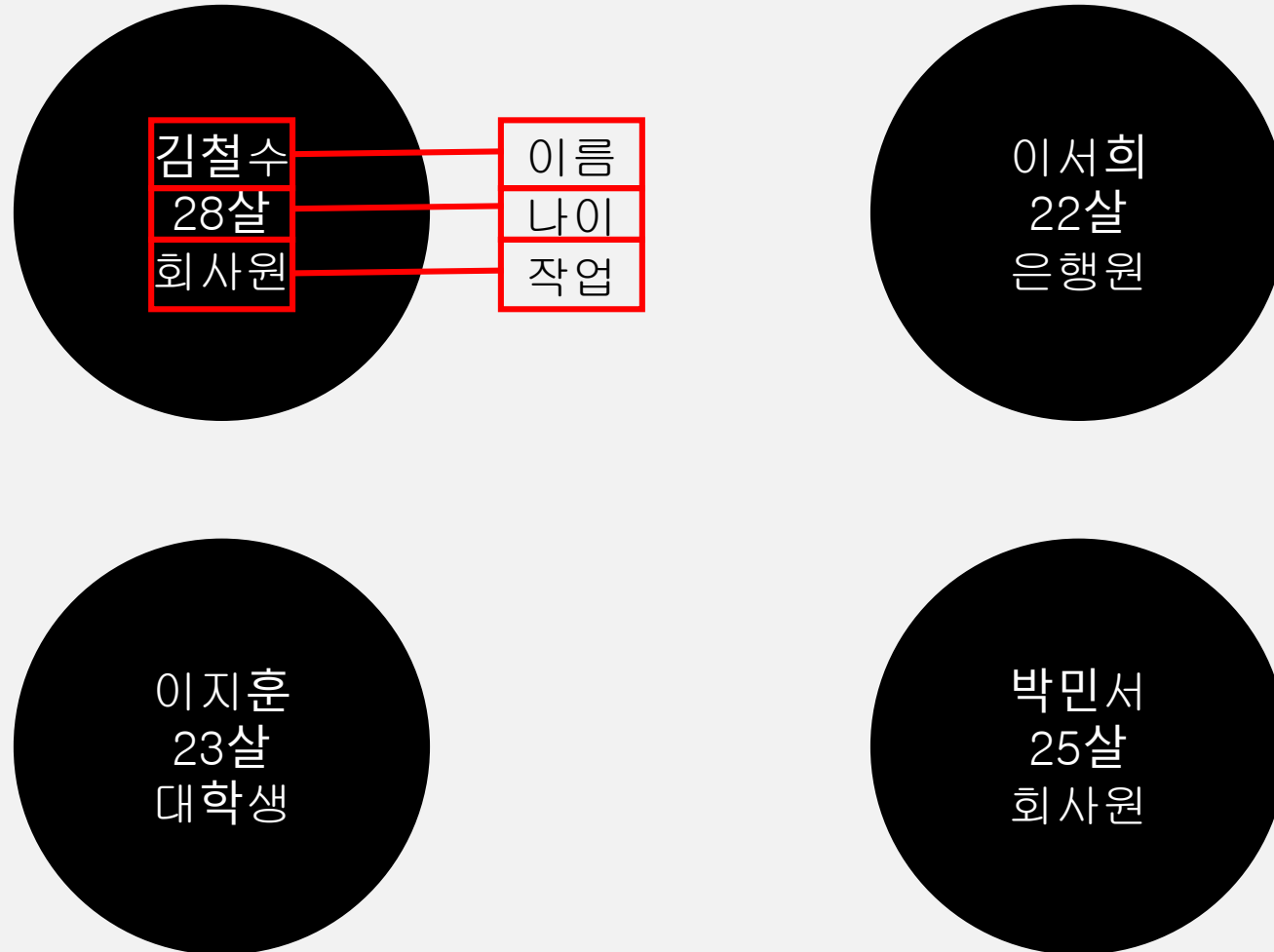
- 사용자와 데이터베이스 사이에 존재

사용자의 명령에 따라 데이터 생성, 삭제, 수정, 조회 등을 처리

Contents 1

Database

Database (DB)



테이블 (Table)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

행 (Row / Record)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

열 (Column / Field)

이름	나이	직업
김철수	28	회사원
이서희	22	은행원
이지훈	23	대학생
박민서	25	회사원

Contents 1
Database

Database (DB)

Database

Table

Table

Table

- **DBMS의 데이터를 관리하기 위해 설계된 프로그래밍 언어**
- **C, Python 같은 언어의 일종**
데이터베이스에 접근하는 언어
- **SQL을 기초로 하는 여러 DBMS가 존재**
예) MySQL, Oracle DB, SQLite, MSSQL 등



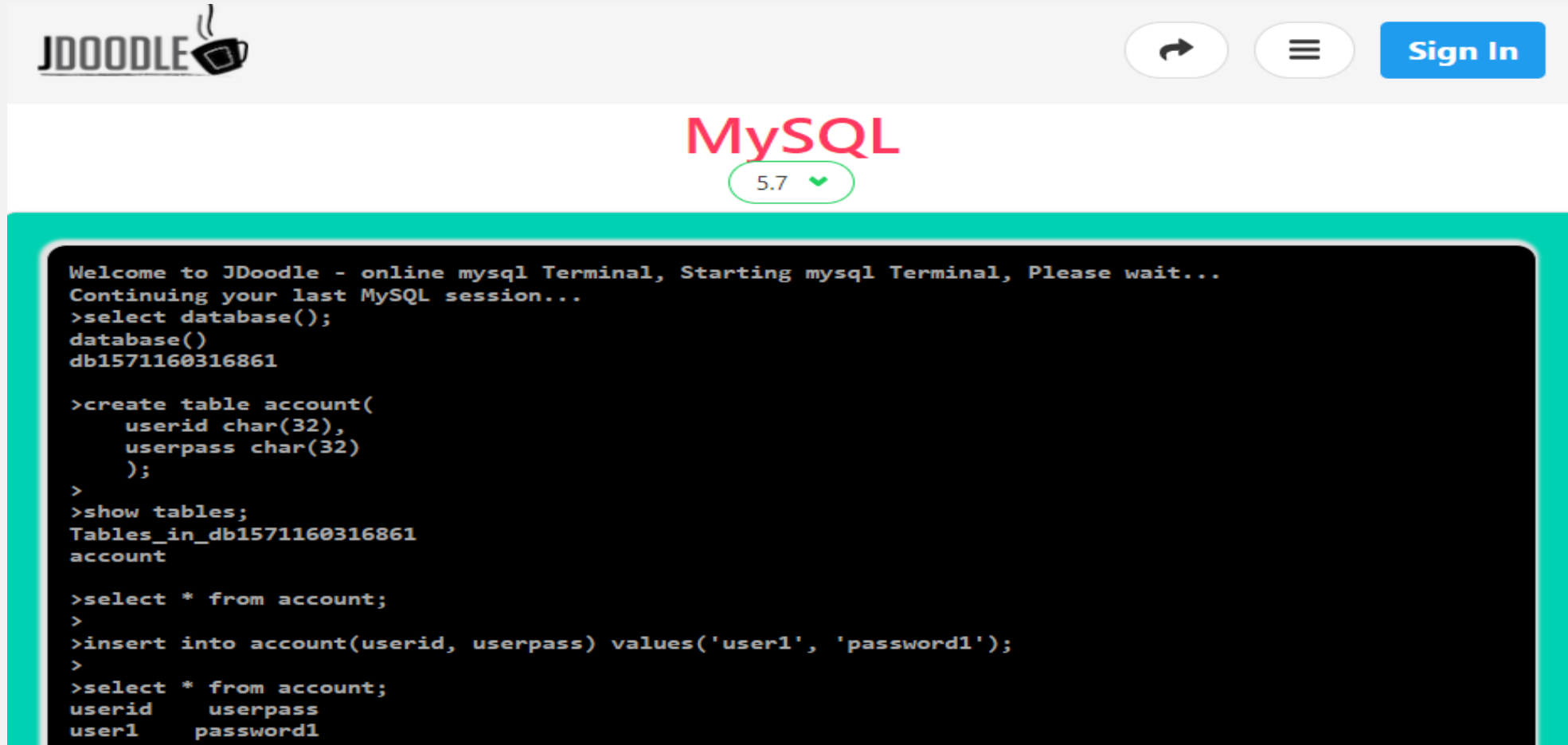
데이터 정의 언어 (DDL)

문법	기능
CREATE	테이블, 데이터베이스 등을 생성
DROP	테이블, 데이터베이스 등을 삭제
ALTER	기존의 테이블, 데이터베이스 등을 수정

데이터 조작 언어 (DML)

문법	기능
SELECT	데이터 조회
INSERT	데이터 삽입
UPDATE	데이터 수정
DELETE	데이터 삭제

<https://www.jdoodle.com/online-mysql-terminal/>



```
Welcome to JDoodle - online mysql Terminal, Starting mysql Terminal, Please wait...
Continuing your last MySQL session...
>select database();
database()
db1571160316861

>show tables;
>
```

- SELECT database()
디폴트(현재) 데이터베이스 이름을 질의
- SHOW TABLES
데이터베이스 안의 모든 테이블을 조회하는 명령
현재 데이터베이스 안에는 아무런 테이블도 존재하지 않음

CREATE TABLE

```
>create table account(  
    userid char(32),  
    userpass char(32),  
    age int  
);  
>show tables;  
Tables_in_db1571160316861  
account
```

- **CREATE TABLE** account (
 [컬럼이름, 자료형],
 [컬럼이름, 자료형]
)
 테이블을 만드는 명령
- show tables로 account 테이블이 생성된 것을
 볼 수 있음

userid	userpass	age

```
>insert into account(userid, userpass, age) values('user1', 'password1', 22);  
>insert into account(userid, userpass, age) values('user2', 'password2', 14);  
>  
>select * from account;  
userid      userpass      age  
user1       password1     22  
user2       password2     14
```

- **INSERT INTO** [테이블 명] ([컬럼], ...) **VALUES** (컬럼 값, ...)
테이블에 데이터(열)을 삽입하는 명령
- **SELECT * FROM** [테이블 명]
테이블 안의 데이터를 조회하는 명령
*는 모든 항목을 의미

SELECT

userid	userpass	age
user1	password1	22
user2	password2	14

- **SELECT * FROM** [테이블 명] **WHERE** [조건]
특정 조건에 해당하는 데이터를 찾을 때 WHERE을 이용

예를 들어, account테이블 내의 age가 20이상인 데이터의 열을 찾고 싶다면
SELECT * FROM account WHERE age >= 20

Contents 2 SQL

SELECT

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT * FROM account WHERE age >= 20

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT userpass FROM account WHERE userid = 'user1'

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT * FROM account
WHERE userid = 'user1' AND userpass='password1'

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT * FROM account
WHERE userid = 'user1' OR userpass = 'password2'

```
>delete from account where userid = 'user2';  
_____  
>select * from account;  
_____  
userid      userpass      age  
user1       password1     22
```

- **DELETE FROM** [테이블 명] **WHERE** [조건]
테이블에 데이터(열)을 삭제하는 명령
- **SELECT * FROM** [테이블 명]

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT * FROM account WHERE userid = '[아이디]' AND userpass = '[비밀번호]'

아이디 입력	user1';#
비밀번호 입력	12345
로그인	

userid	userpass	age
user1	password1	22
user2	password2	14

SELECT * FROM account WHERE userid = 'user1';#' AND userpass = '12345'

SELECT * FROM account WHERE userid = 'user1';#' AND userpass = '12345'

```
>select * from account where userid = 'user1';#' and userpass = '12345';  
userid      userpass    age  
user1       password1   22
```

- SQL Injection
공격자가 Input에 악의적인 내용을 삽입하여 의도하지 않은 Query문이 실행되는 것
위의 예시처럼 인증을 우회하거나 데이터베이스 내의 정보가 유출되거나 서버가 장악 당하는 피해 발생
- 주석(#, --)을 이용한 SQL Injection
#은 MySql에서의 주석
#뒤의 내용 ' and userpass = '12345';가 무시됨
따라서 12345라는 잘못된 비밀번호 입력에도 로그인 성공

```
>select * from account where userid = 'user1' or '1' and userpass='12345';  
userid      userpass    age  
user1       password1   22  
  
>select * from account where userid = 'aaaa' and userpass = ''or 1;#'  
userid      userpass    age  
user1       password1   22  
user2       password2   25
```

- AND 와 OR 연산자를 이용한 SQL Injection
SQL에서 AND 와 OR 중 AND의 우선순위가 더 높음
따라서 AND와 OR이 한문장에 있다면 AND가 먼저 진행됨
이를 통해 인증에 성공
- SELECT * from account where 1 은 SELECT * from account 와 동일

```
>select * from account where userid = 'user1' or '1' and userpass='12345';
userid      userpass    age
user1       password1   22
```

- userid, userpass 입력

`SELECT * FROM account WHERE userid = 'user1' OR '1' AND userpass = '12345'`

입력한 아이디 입력한 비밀번호

- AND 연산자 실행

SELECT * FROM account WHERE userid = 'user1' OR '1' AND userpass = '12345'

1은 True, Userpass = 12345에 해당하는 값이 없음
true(1) and Empty(0) = Empty

- OR 연산자 실행

SELECT * FROM account WHERE **userid = 'user1' OR '1'** AND userpass = '12345'

'user1' or 0 = 'user1' 반환 Empty(0)
Userid = 'user1'인 로그인 성공


```
>select * from account where userid = 'aaaa' and userpass = ''or 1;#'
```

userid	userpass	age
user1	password1	22
user2	password2	25

- userid, userpass 입력

`SELECT * FROM account WHERE userid = 'aaaa' AND userpass = ''OR 1; #'`

입력한 아이디 입력한 비밀번호

- AND 연산자 실행

SELECT * FROM account WHERE userid = 'aaaa' AND userpass = '' OR 1; #'

- OR 연산자 실행
 - userid = aaaa에 해당하는 값이 없음
 - userpass = 는 비어있음
 - AND 하면 Empty

- OR 연산자 실행

SELECT * FROM account WHERE userid = 'aaaa' AND userpass = " OR 1; #'









#뒤에 '는 주석처리
Empty OR 1 = 1
WHERE 1로 들어가서 모든 행 반환

Contents 3

SQL Injection

bee-box 실습

- bee-box 실행

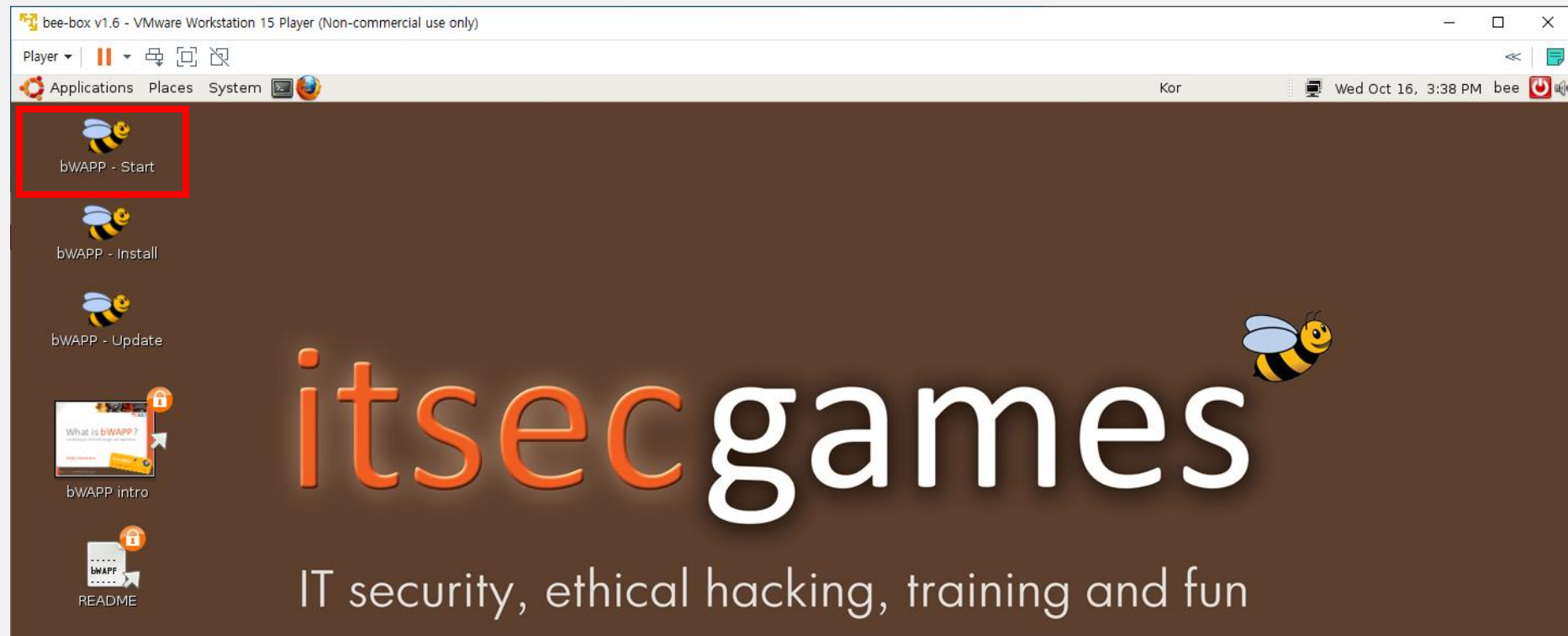
	bee-box.vmdk.lck	2019-10-16 오후 10:27	파일 폴더	
	bee-box.vmx.lck	2019-10-16 오후 10:27	파일 폴더	
	bee-box-c4429406.vmem.lck	2019-10-16 오후 10:27	파일 폴더	
	bee-box.nvram	2019-10-16 오전 12:27	NVRAM 파일	9KB
	bee-box	2019-10-16 오후 10:27	VMware virtual di...	1KB
	bee-box.vmsd	2014-11-03 오전 8:21	VMSD 파일	1KB
	bee-box	2019-10-16 오후 10:27	VMware virtual m...	3KB
	bee-box.vmx	2013-07-16 오전 6:58	VMXF 파일	4KB

Contents 3

SQL Injection

bee-box 실습

- bWAP 실행

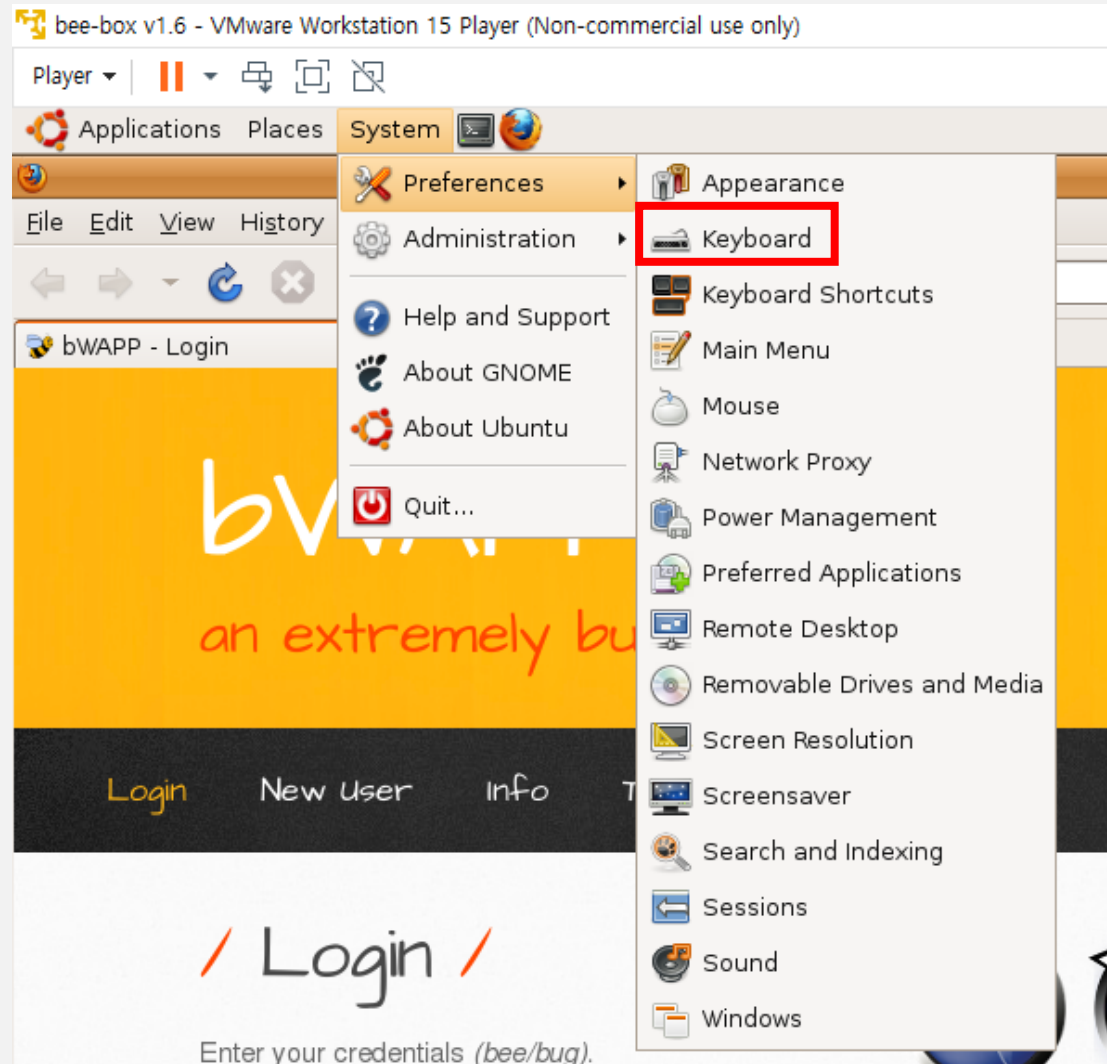


Contents 3

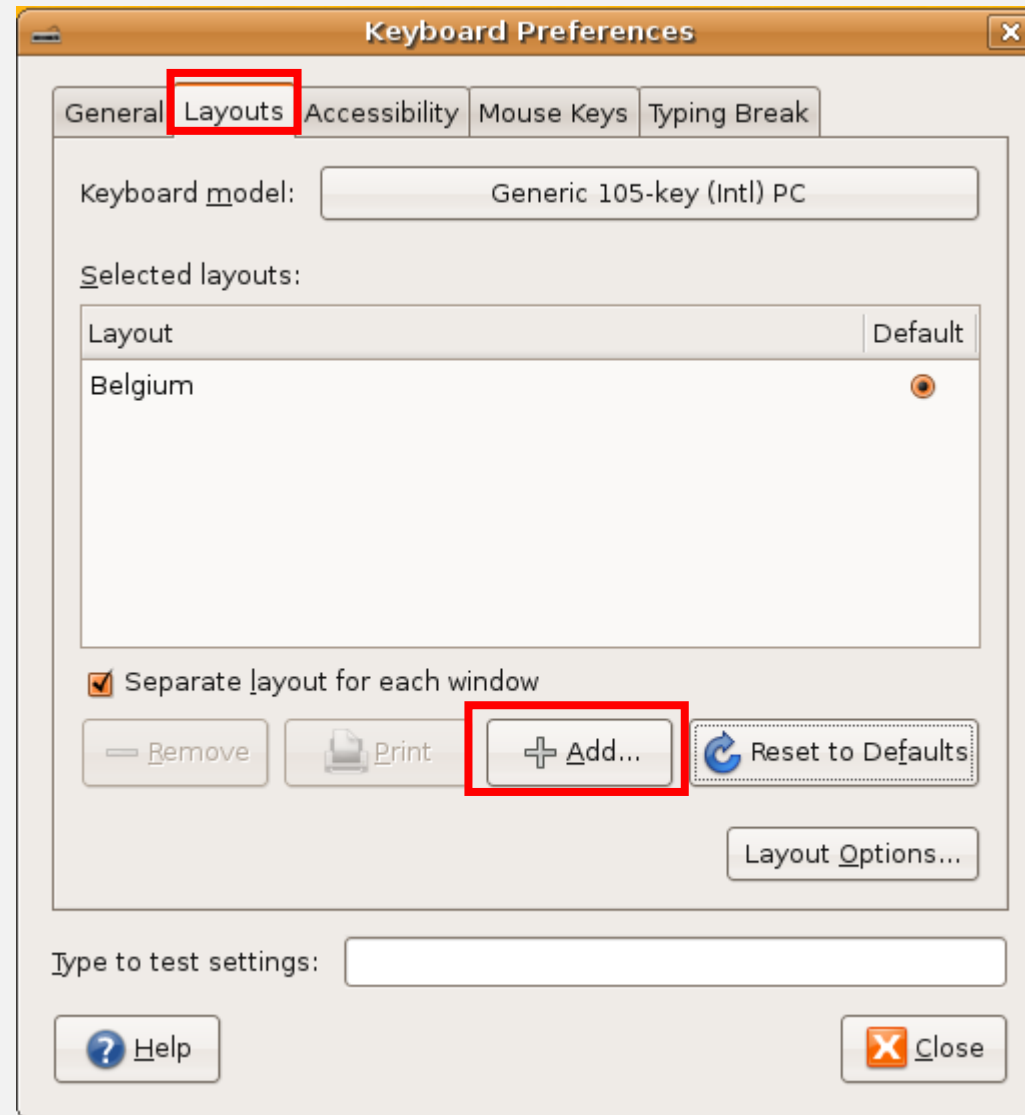
SQL Injection

bee-box 실습

- 키보드 설정



- 키보드 설정

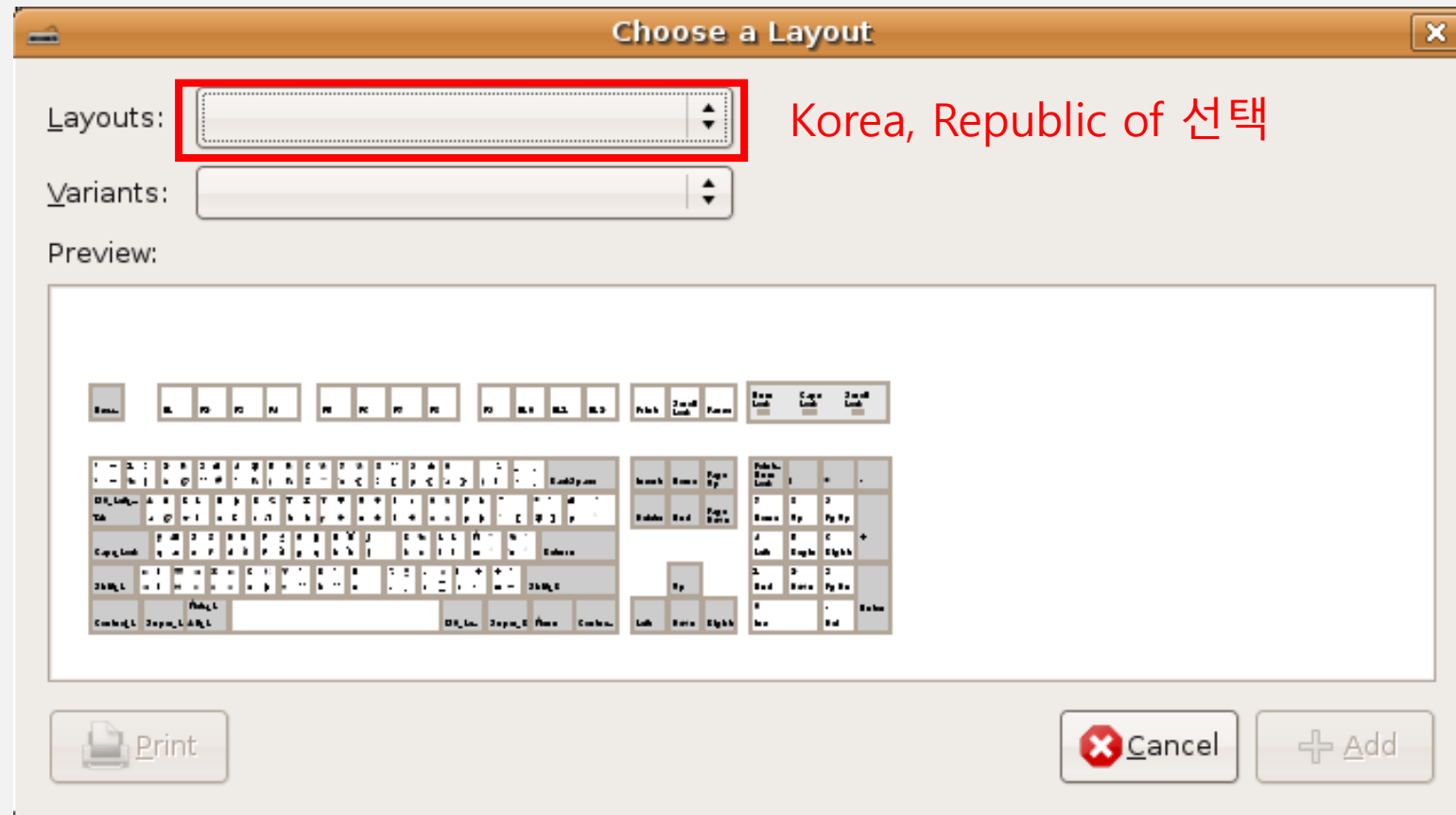


Contents 3

SQL Injection

bee-box 실습

- 키보드 설정

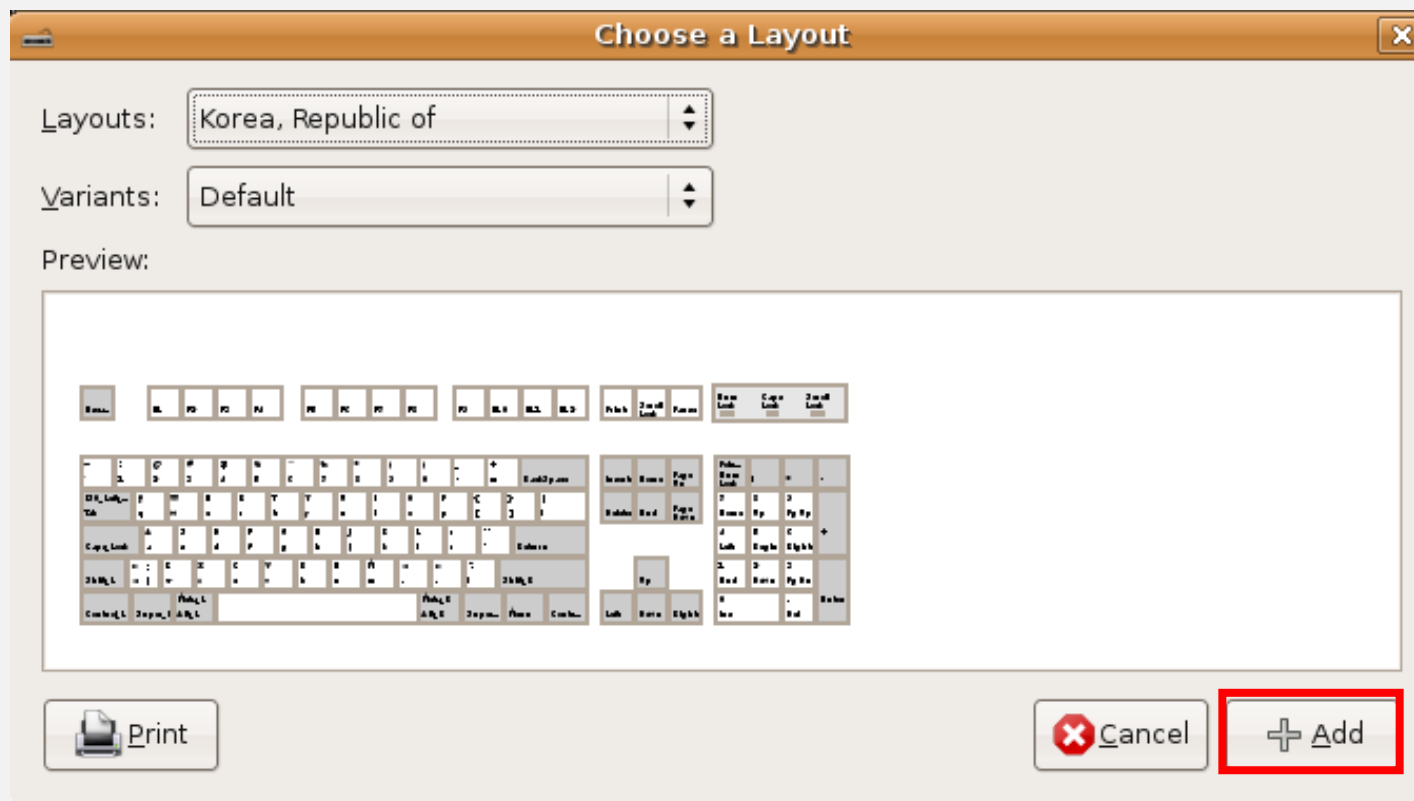


Contents 3

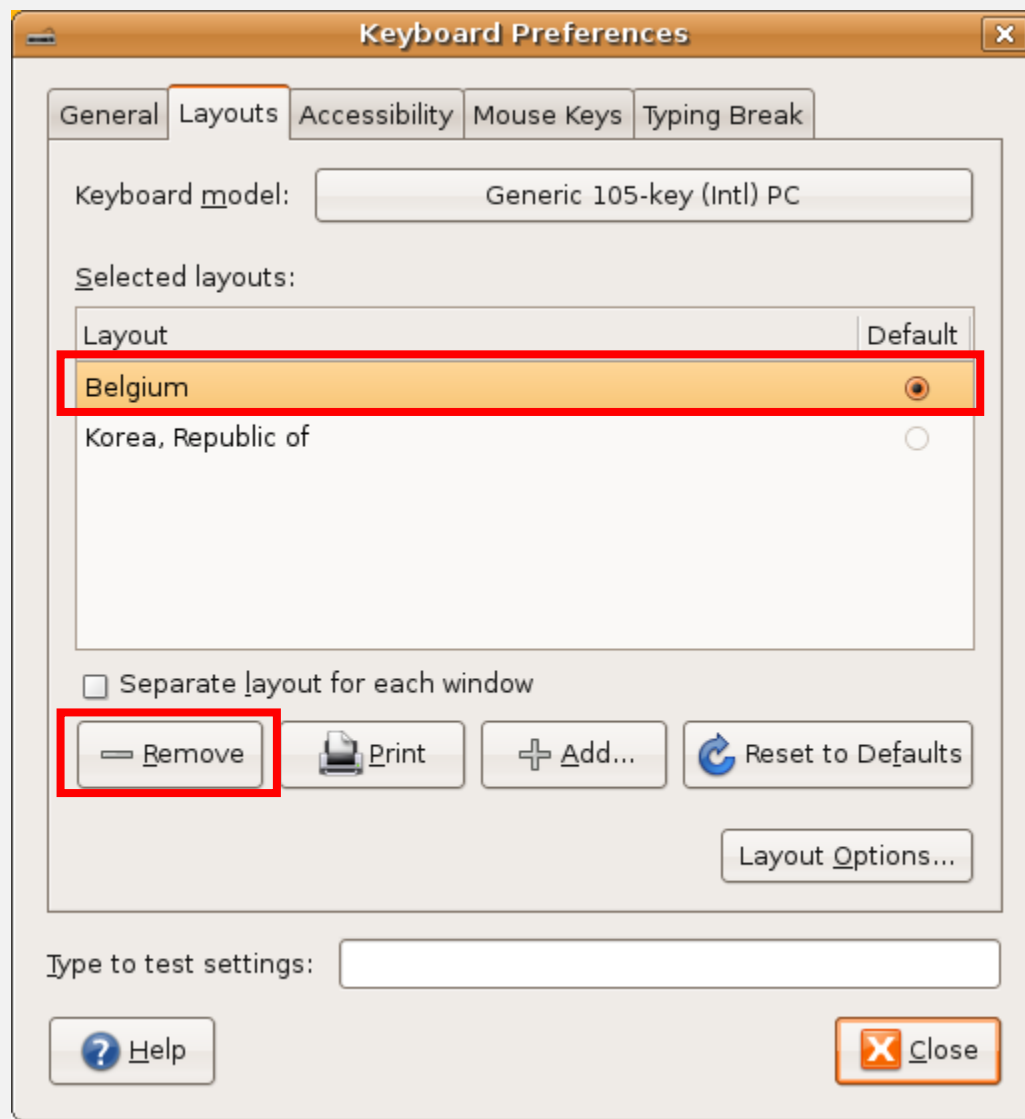
SQL Injection

bee-box 실습

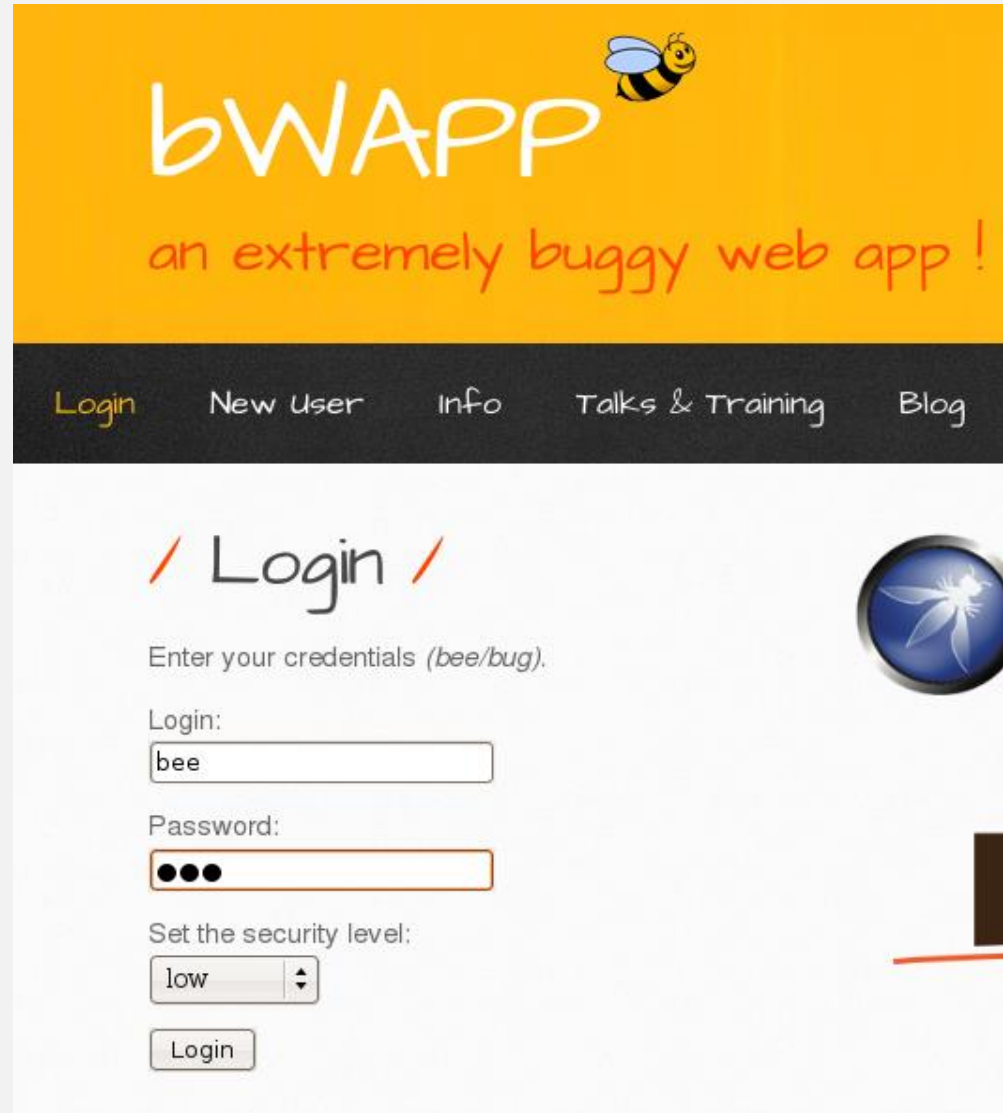
- 키보드 설정



- 키보드 설정



- bee / bug 로 로그인



The screenshot shows the bWAPP login interface. At the top, there's a yellow banner with the text "bWAPP" in white, a small bee icon, and the tagline "an extremely buggy web app !" in red. Below this is a dark navigation bar with links: "Login" (highlighted in yellow), "New User", "Info", "Talks & Training", and "Blog". The main content area is white and titled "Login" with red slashes. It prompts the user to "Enter your credentials (bee/bug)". There are two input fields: "Login:" with the text "bee" entered, and "Password:" with three black dots. Below these is a "Set the security level:" section with a dropdown menu currently set to "low". A "Login" button is at the bottom. On the right side, there's a circular blue icon with a white bug and a small black rectangular element with a red underline.

bWAPP
an extremely buggy web app !

Login New User Info Talks & Training Blog

/ Login /

Enter your credentials (bee/bug).

Login:

Password:

Set the security level:

Login

- SQL Injection 문제 선택

Choose your bug:

SQL Injection (Login Form/Hero)

Set your security level:

low Current: low

- ID 'Neo'로 로그인 하기

/ SQL Injection (Login Form/Hero) /

Enter your 'superhero' credentials.

Login:

Password:

Login

- 존재하는 모든 계정의 개수와 계정 명 알아내기 (Hint : LIMIT)



The image shows a web application interface for a login form. At the top, the title "SQL Injection (Login Form/Hero)" is displayed in a large, handwritten-style font, flanked by two orange diagonal slashes. Below the title, the instruction "Enter your 'superhero' credentials." is shown in a smaller, grey font. The form consists of two input fields: "Login:" followed by a text box, and "Password:" followed by a text box. Below these fields is a "Login" button with a grey gradient and rounded corners.

감사합니다