[wargame] ch4n3-world
Bug hunting report
(SQL Injection, Invalid code)

ch4n3 world 버그 헌팅 보고서

선린인터넷고등학교 문시우
2017.11.22

1. join_chk.php (Invalid code)

- mysql error

preg_match()의 정규식에서 필터 키워드를 잘못 표현해 발생한 에러다.
만약 '\'를 검사하고 싶다면 "/\\/"가 아닌 "/\\\\/" 이렇게 검사해야 한다.
하지만 join_chk.php의 59줄을 보면 "/\\/"로 검사하고 있다.

```
59  if(preg_match("/'|\"|\\|`|@|[*]|_|-|;|=/i", $id)) {

74  // 이미 있는 아이디, 닉네임인지 확인함
75  $query = "select id, nick from login where id=('{$id}')";
76  $result = mysqli_fetch_array($mysqli->query($query));
```

따라서 '\'를 넣어주게 되면 preg_match()에 걸리지 않고
mysqli_query는 false를 반환한다. 그리고 아래와 같은 에러가 발생한다.

Warning: mysqli_fetch_array() expects parameter 1 to be mysqli_result, boolean given in /host/home1/ch4n3/html/join_chk.php on line 76

insert into solvers(id,pw,nick,intro,ip) values ($id,$pw,$nick,$intro,ip); 부분에서
insert injection을 시도했으나 pw는 sha512, intro는 base64로 인코딩 하므로
insert injection으로 DB의 데이터를 뽑는 건 힘든 상황이었다.

- 중복 닉네임 가입

```
74  // 이미 있는 아이디, 닉네임인지 확인함
75  $query = "select id, nick from login where id=('{$id}')";
76  $result = mysqli_fetch_array($mysqli->query($query));
77
78  if($id === $result['id']) {
79      die("<script>alert('이미 존재하는 아이디입니다.'); history.go(-1);</script>");
80  } else if($nick === $result['nick']) {
81      die("<script>alert('이미 존재하는 닉네임입니다.'); history.go(-1); </script>");
82  }
```

75줄의 query를 보면 아이디만 가져와 비교하여 검사한다.
아이디와 닉네임을 검사하려면 id=('{$id}') or nick=('{$nick}'); 이렇게 해야 한다.
또한 ===으로 비교할 필요 없이 가져온 레코드가 있는지 없는지만 검사하면 된다.

---

2. problem.php (sql injection)

- SQL Injection (1)

```
6   if(!isset($_SESSION['nick'])) {
7       die("login plz..");          //로그인 했는지 확인함
8   } else if(!isset($_GET['no']) || $_GET['no']=="") {
9       die("query error..");        //no에 올바른 값이 제대로 넘어왔는지 확인함
10  } else if(preg_match("/#|-|_|;|\n| |\t/", $_GET['no'])) {
11      die("no hack ~_~");          //no에 hackable 단어가 있으면 die함
12  }
```

problem.php는 플래그 인증을 하는 페이지다.
$_GET['no']로 문제 번호를 받고 있고 필터링 키워드는 위와 같다.

```
25  $query = "select * from probs where no=(({$no}))";
26  $result = @mysqli_fetch_array($mysqli->query($query));
```

25줄을 보면 SQL Injection이 발생함을 알 수 있다. (그 외 17줄, 41줄, 51줄)
공격하는 여러 방법과 페이로드는 다음 페이지에서 확인하자.

- SQL Injection (2)

필터링 키워드가 별로 없기 때문에 쉽게 공격이 가능하다.
닷홈의 웹 방화벽 기능이 없다면 union based로 모두 뽑아올 수 있으며
웹 방화벽을 피해 time based 또는 insert injection도 가능하다.

union based) no=0)) union select 1,2,3,4,5,((6
time based) no=0))||sleep(1)&&no=((1

insert injection은 플래그를 인증하고 풀이자의 정보를 테이블에 저장할 때
query : insert into solvers values({$no}, '$id', '$nick', now());
위와 같이 solvers 테이블에 풀이자 정보를 넣는 부분에서 발생한다.

insert based) no=1,'my_id',(select%0a~~~~~),'2017-11-22')#

이렇게 union, time, insert based 등 다양한 기법으로 공격할 수 있다.

- SQL Injection (3) : 중복 인증하기

```
// 플래그 인증 부분
if (isset($_POST['flag'])) {
    if ($_POST['flag']===$flag) {
        $query = "select * from solvers where id=('{$_SESSION['id']}') and no=({$no})";
        $result = @mysqli_fetch_array($mysqli->query($query));
        if (isset($result['id'])) {
            die("Already solved (date : {$result['date']})");
        }

        //푼 사용자의 점수를 올려주는 쿼리
        $query = "update login set point=point+{$point} where id=('{$_SESSION['id']}')";
        $result = $mysqli->query($query);
```

POST : http://ch4n3.dothome.co.kr/problem.php?no=0))||no=((4
BODY : flag=(4번 문제의 플래그)

이미 풀었던 문제인지 solvers테이블에서 검사할 때 no 컬럼에
'no=(0))||no=((4)' 이렇게 들어가기 때문에 해당 query는 에러가 발생한다.
그러므로 중복 검사가 이루어지지 않고 계속해서 인증할 수 있는 것이다.

ch4n3-world
Bug hunting report
(SQL Injection, Invalid code)

선린인터넷고등학교 문시우