目录

一、题目

1、源码

```php
class RealSecureLoginManager {
    private $em;
    private $user;
    private $password;

    public function __construct($user, $password) {
        $this->em = DoctrineManager::getEntityManager();
        $this->user = $user;
        $this->password = $password;
    }

    public function isValid() {
        $pass = md5($this->password, true);
        $user = $this->sanitizeInput($this->user);

        $queryBuilder = $this->em->createQueryBuilder()
            ->select("COUNT(p)")
            ->from("User", "u")
            ->where("password = '$pass' AND user = '$user'");
        $query = $queryBuilder->getQuery();
        return boolval($query->getSingleScalarResult());
    }

    public function sanitizeInput($input) {
        return addslashes($input);
    }
}

$auth = new RealSecureLoginManager(
    $_POST['user'],
    $_POST['passwd']
);
if (!$auth->isValid()) {
    exit;
}

```

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。

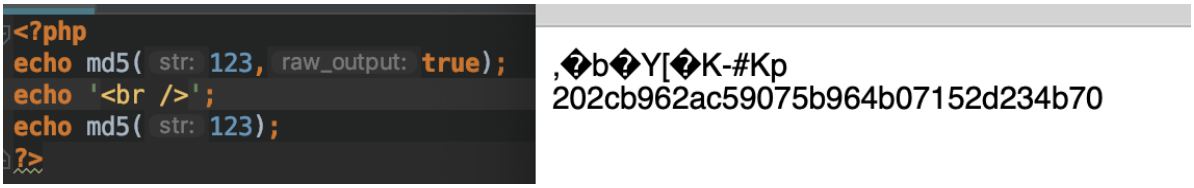
3）第33行，如果对象中的函数isValid()为假，就退出。

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）



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。

4、利用

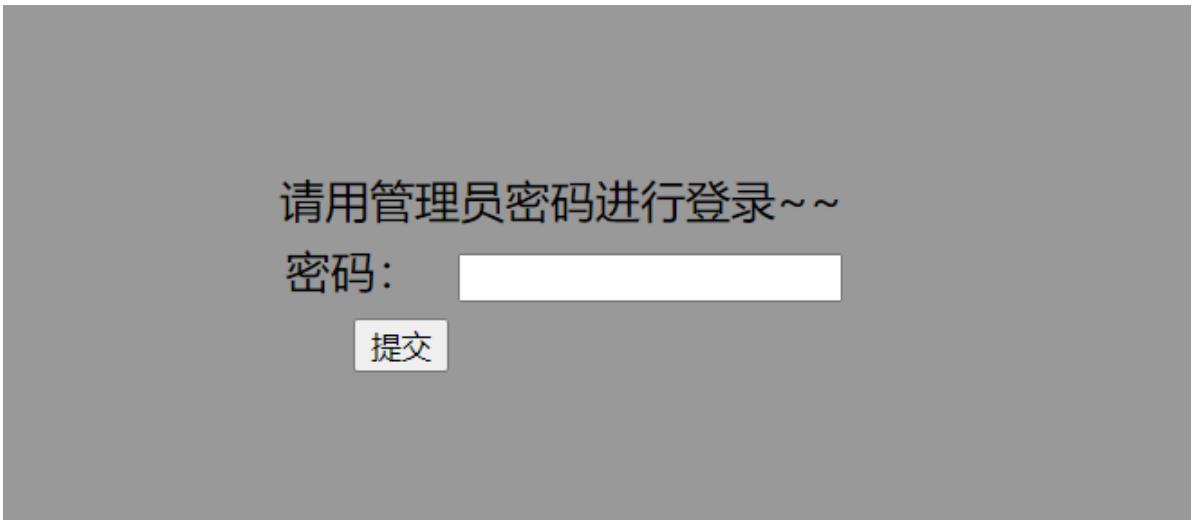
Payload：user= OR 1=1#&passwd=128

拼接后：select count(p) from user s where password='v a n l q \' and user=' OR 1=1#'

二、CMS-CTF题

1、分析

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



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

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body style="background-color: #999">
    <div style="position:relative;margin:0 auto;width:300px;height:200px;padding-top:100px;font-size:20px;">
    <form action="" method="post">
        <table>
            <tr>
                请用管理员密码进行登录~~
            </tr>
            <tr>
                <td>密码：</td><td><input type="text" name='password'></td>
            </tr>
            <tr>
                <td><input type="submit" name='submit' style="margin-left:30px;"></td>
            </tr>
        </table>
    </form>
        </div>
    <!-- $password=$_POST['password'];
    $sql = "SELECT * FROM admin WHERE username = 'admin' and password = '".md5($password,true)."'";
    $result=mysqli_query($link,$sql);
        if(mysqli_num_rows($result)>0){
            echo 'flag is :'.$flag;
        }
        else{
            echo '密码错误!';
        } -->
</body>
</html>

```

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