

ourphp 站内信存储型xss漏洞

漏洞简介：

OurPHP（傲派建站系统）是一款使用PHP语言开发的网站内容管理系统，开发商为哈尔滨伟成科技有限公司。

OurPHP 1.7.1版本存在存储型xss漏洞，可获取任意用户cookie

漏洞分析

ourphp 站内信模块 过滤不严导致存储型xss漏洞。

漏洞文件:www/client/user/ourphp_play.class.php 276行

```
//处理站内邮件
}elseif($_GET["ourphp_cms"] == 'mail'){

    $query = $db -> select("id","`ourphp_user`","WHERE `OP_Useremail` =
'".dowith_sql($_POST["OP_Usercollect"])."' || `OP_Usertel` =
'".dowith_sql($_POST["OP_Usercollect"]).'");
    if (!$query){

        exit("<script language=javascript>
alert('".$usernameo."');history.go(-1);</script>");

    }elseif

        if (dowith_sql($_POST["OP_Usercollect"]) == $_SESSION['username']){
            exit("<script language=javascript>
alert('".$accessno."');history.go(-1);</script>");
        }

        $add = $db -> insert("`ourphp_usermessage`","`OP_Usersend` =
'".$_SESSION['username']."'`,`OP_Usercollect` =
'".dowith_sql($_POST["OP_Usercollect"])."`,`OP_Usercontent` =
'".dowith_sql($_POST["OP_Usercontent"])."`,`time` = '".date("Y-m-d
H:i:s")."',");
```

这里我们可以看到 `OP_Usercontent` 参数用 `dowith_sql` 进行了过滤。继续跟进 `dowith_sql` 参数。在文件 `www/function/ourphp_function.class.php`

```
function dowith_sql($ourphpstr){
    $ourphpstr = addslashes($ourphpstr);
    $ourphpstr = str_ireplace(" and ", "", $ourphpstr);
    $ourphpstr = str_ireplace(" or ", "", $ourphpstr);
    $ourphpstr = str_ireplace("execute", "", $ourphpstr);
    $ourphpstr = str_ireplace("update", "", $ourphpstr);
    $ourphpstr = str_ireplace("count", "", $ourphpstr);
    $ourphpstr = str_ireplace("chr", "", $ourphpstr);
    $ourphpstr = str_ireplace("truncate", "", $ourphpstr);
    $ourphpstr = str_ireplace("char", "", $ourphpstr);
```

```

$ourphpstr = str_ireplace("declare","", $ourphpstr);
$ourphpstr = str_ireplace("select","", $ourphpstr);
$ourphpstr = str_ireplace("create","", $ourphpstr);
$ourphpstr = str_ireplace("delete","", $ourphpstr);
$ourphpstr = str_ireplace("insert","", $ourphpstr);
$ourphpstr = str_ireplace("limit","", $ourphpstr);
$ourphpstr = str_ireplace("extractvalue","", $ourphpstr);
$ourphpstr = str_ireplace("concat","", $ourphpstr);
$ourphpstr = str_ireplace("&&", "", $ourphpstr);
$ourphpstr = str_ireplace("||", "", $ourphpstr);
$ourphpstr = str_ireplace("alert","", $ourphpstr);
$ourphpstr = str_ireplace("script","", $ourphpstr);
$ourphpstr = str_ireplace("iframe","", $ourphpstr);
$ourphpstr = str_ireplace("embed","", $ourphpstr);
$ourphpstr = str_ireplace("*", "", $ourphpstr);
$ourphpstr = str_ireplace("#", "", $ourphpstr);
$ourphpstr = str_ireplace("'", "\'", $ourphpstr);
return $ourphpstr;
}

```

可以看到这里只过滤了 `alert` , `script` 。那么可以用其他标签触发xss。继续看模板页面代码。在 `www/client/user/ourphp_mail.php` 28行

```

$ourphp_rs = $db ->
select("OP_Usersend,OP_Usercollect,OP_Usercontent,time","`ourphp_usermessage`","
where id = ".$id);
if($ourphp_rs[0] == $_SESSION['username'] || $ourphp_rs[1] ==
$_SESSION['username']){
?>

<table width="90%" border="0" cellpadding="10" style="font-size:12px;">
  <tr>
    <td width="150"><div align="right">发件人: </div></td>
    <td>&nbsp;<?php if($ourphp_rs[0] == $_SESSION['username']){ echo '我';}else{
echo $ourphp_rs[0];} ?></td>
  </tr>
  <tr>
    <td><div align="right">收件人: </div></td>
    <td>&nbsp;<?php echo $ourphp_rs[1]; ?></td>
  </tr>
  <tr>
    <td valign="top"><div align="right">收件内容: </div></td>
    <td>&nbsp;<?php echo $ourphp_rs[2]; ?></td>
  </tr>
  <tr>
    <td><div align="right">时间: </div></td>
    <td>&nbsp;<?php echo $ourphp_rs[3]; ?></td>
  </tr>
</table>

```

从上述代码我们可以看到 邮件内容 从sql语句 读取之后, 没有任何过滤, 直接用 `echo` 函数输出。

漏洞复现

在会员中心>站内信件 发送一封站内信, 前提是我们知道其他用户的收件账号。构造payload

```
<svg/onload=prompt(0)>。
```

🏠 会员中心

🛒 购物车 0

📋 我的订单 0

📍 收货地址

收件账号： *

信件内容：


<svg/onload=prompt(0)>

验证码： * 

发送完毕触发xss漏洞。

收件账号：

信件内容：

验证码： * 

0

信件列表：

发送人	接收人	内容	时间
我	test@qq.com		2017-06-07 17:05:26