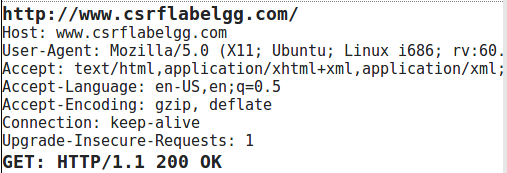
Lab4-Report

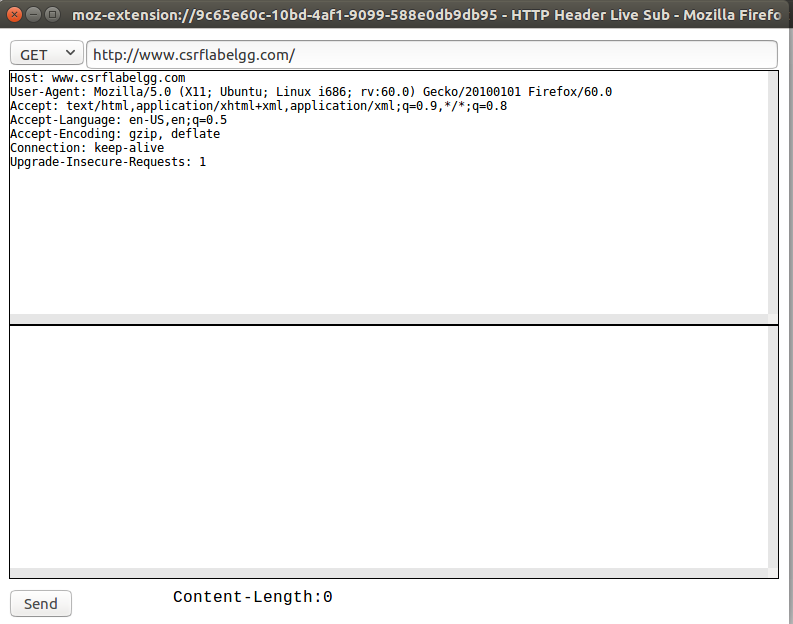
57118103郭欣然

**Task1：Observing HTTP Request**

用HTTP Header Live在Elgg网站抓取GET和POST请求报文，并对参数进行分析，如下图：

**GET报文：**





﻿Host: www.csrflabelgg.com

请求的主机名，允许多个域名同处一个IP地址，即虚拟主机。

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

用户代理，产生请求的浏览器类型，这里是Ubuntu系统的火狐浏览器

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

客户端可识别的内容类型列表，这里表示可以识别以上类型

Accept-Language: en-US,en;q=0.5

接受语言，即通知服务端可以发送的语言

Accept-Encoding: gzip, deflate

接受编码，通知服务端可以发送的数据压缩格式

Connection: keep-alive

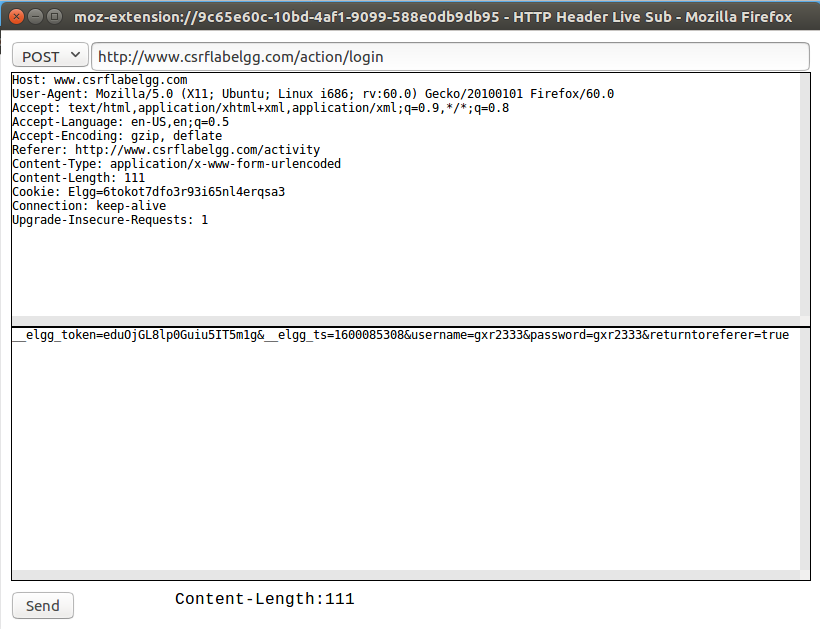
﻿指定与连接相关的属性，这里是连接：保持活动

Upgrade-Insecure-Requests: 1

用于让浏览器自动升级请求从http到https,用于大量包含http资源的http网页直接升级到https，相当于在http和https之间起的一个过渡作用.

**POST报文：**





分析上图：

Host: [www.csrflabelgg.com](http://www.csrflabelgg.com)

请求的主机名，允许多个域名同处一个IP地址，即虚拟主机。

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

用户代理，产生请求的浏览器类型，这里是Ubuntu系统的火狐浏览器

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

客户端可识别的内容类型列表。

Accept-Language: en-US,en;q=0.5

接受语言，即通知服务端可以发送的语言

Accept-Encoding: gzip, deflate

接受编码，通知服务端可以发送的数据压缩格式

Referer: http://www.csrflabelgg.com/activity

请求的来源网址

Content-Type: application/x-www-form-urlencoded

请求行（post专有），具体请求中的媒体类型信息，这里的application/x-www-form-urlencoded 表示：<form encType=””>中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）

Content-Length: 111

请求行（post专有），描述HTTP消息实体的传输长度，the transfer-length of the message-body，这里http消息实体的传输长度为111

Cookie: Elgg=6tokot7dfo3r93i65nl4erqsa3

﻿将客户端的cookie传给服务器

Connection: keep-alive

﻿指定与连接相关的属性，这里是连接：保持活动

Upgrade-Insecure-Requests: 1

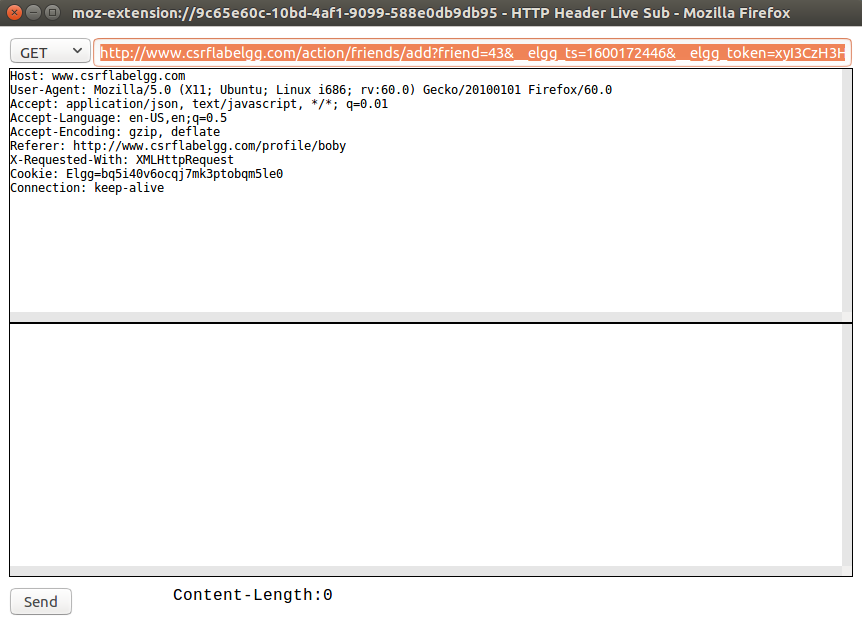
用于让浏览器自动升级请求从http到https,用于大量包含http资源的http网页直接升级到https，相当于在http和https之间起的一个过渡作用.

\_\_elgg\_token=eduOjGL8lp0Guiu5IT5m1g&\_\_elgg\_ts=1600085308&username=gxr2333&password=gxr2333&returntoreferer=true

请求正文内容，头部请求的求最后会有一个空行，表示请求头部结束，接下来为请求正文。

**Task2：CSRF Attack using GET Request**

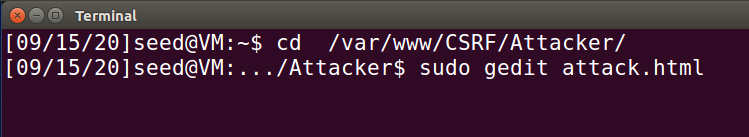
为了得到添加好友的GET请求细节，首先登录Alice账号并手动添加Boby为好友，然后用Live HTTP Header工具观察HTTP报头：

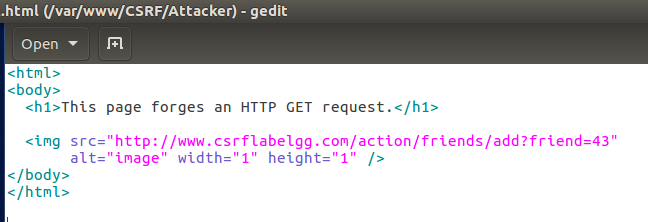


GET请求的报头，附带信息有friend的id号（friend=43），即Alice的guid是43，后面elgg\_ts,egg\_token是个时间戳，防伪冒用。

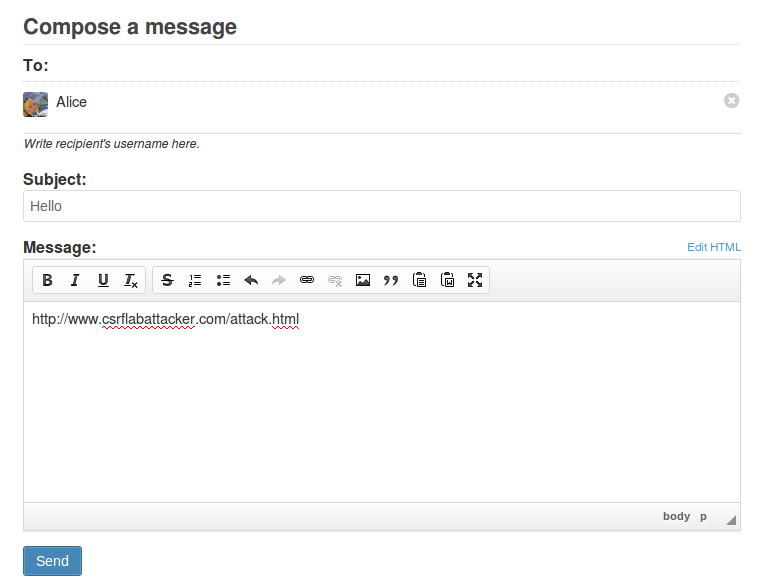
接下来进入路径修改攻击者网页的内容，根据要求添加一个<img>标签，src使用添加好友的url链接，即当Alice访问了这个恶意网页之后，因为会自动GET<img>标签的内容，所以url自动就发出去了，浏览器根据url发送给服务器添加好友的命令，也就是Alice主动添加Boby为好友。

网页源代码如下：

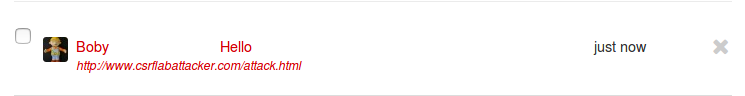




登录Boby向Alice发送一封邮件。



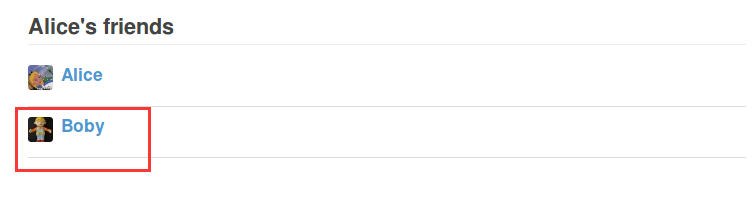
Alice收到邮件，并点击进去链接。



进入攻击者网站，用抓包工具可以看到后台自动发送了一个GET命令。

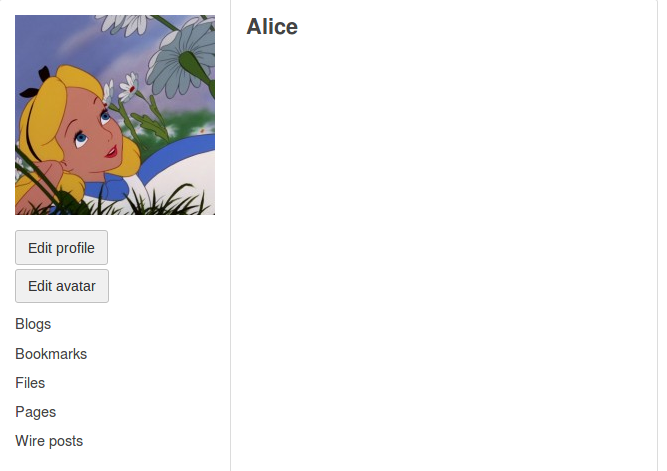


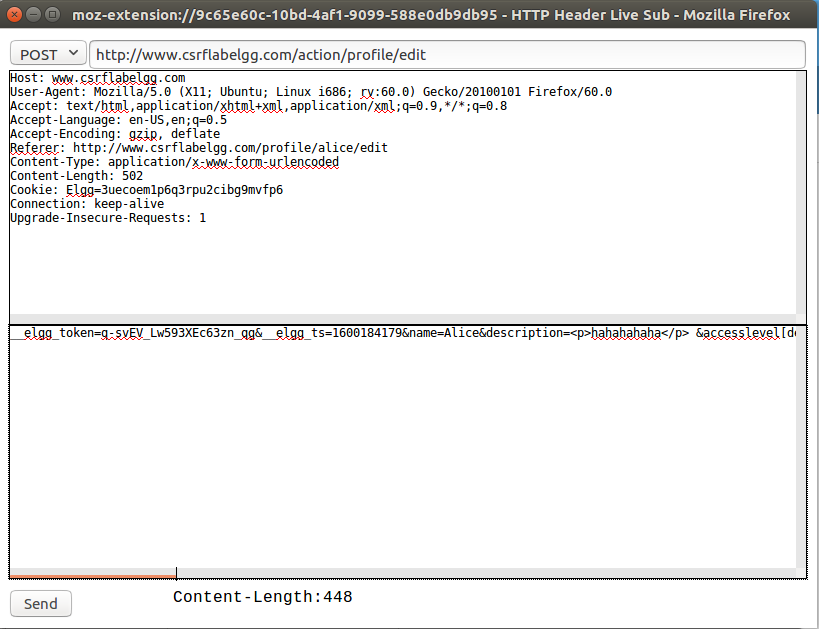
查看Alice好友圈，发现Boby已经被自动添加为Alice好友。攻击成功。



**Task3: CSRF Attack using POST Request**

为了得到POST报文的具体细节，先登录Alice手动修改自己的Profile里面的签名并保存，随后删除签名



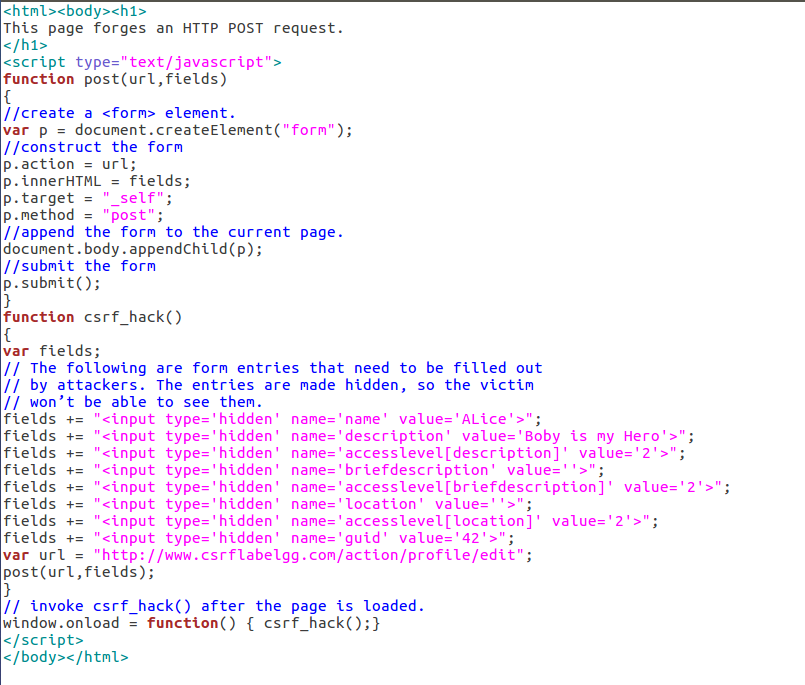


\_\_elgg\_token=q-svEV\_Lw593XEc63zn\_qg&\_\_elgg\_ts=1600184179&name=Alice&description=<p>hahahahaha</p>

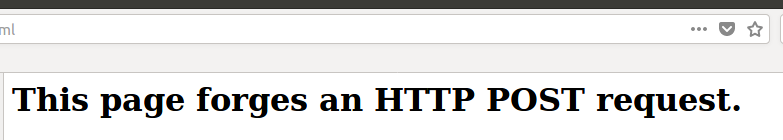
&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&accesslevel[location]=2&interests=&accesslevel[interests]=2&skills=&accesslevel[skills]=2&contactemail=&accesslevel[contactemail]=2&phone=&accesslevel[phone]=2&mobile=&accesslevel[mobile]=2&website=&accesslevel[website]=2&twitter=&accesslevel[twitter]=2&guid=42

利用抓包工具抓取POST命令，观察内容，对应修改攻击网站代码如下：

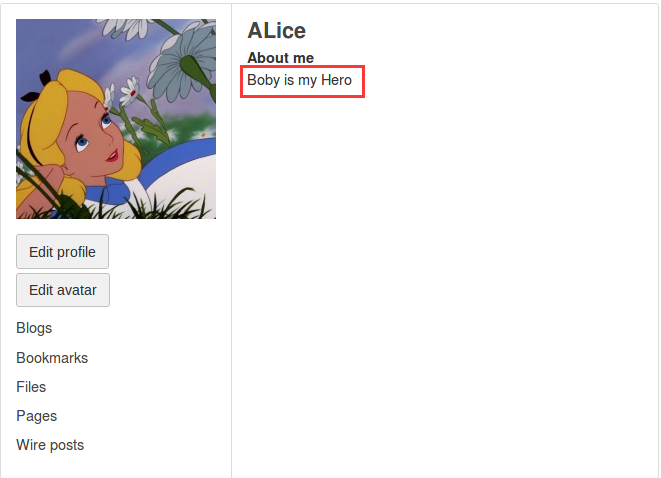
其中单引号应手动修改为英文字符，name=’accesslevel[briefdescription]’ value=’2’>"中value修改为2，即公开权限。



再次让Alice访问Boby写的攻击网页，显示出代码中添加的如下内容。



自动跳转至Alice个人简介，发现已经被修改为“Boby is my hero”



思考问题：

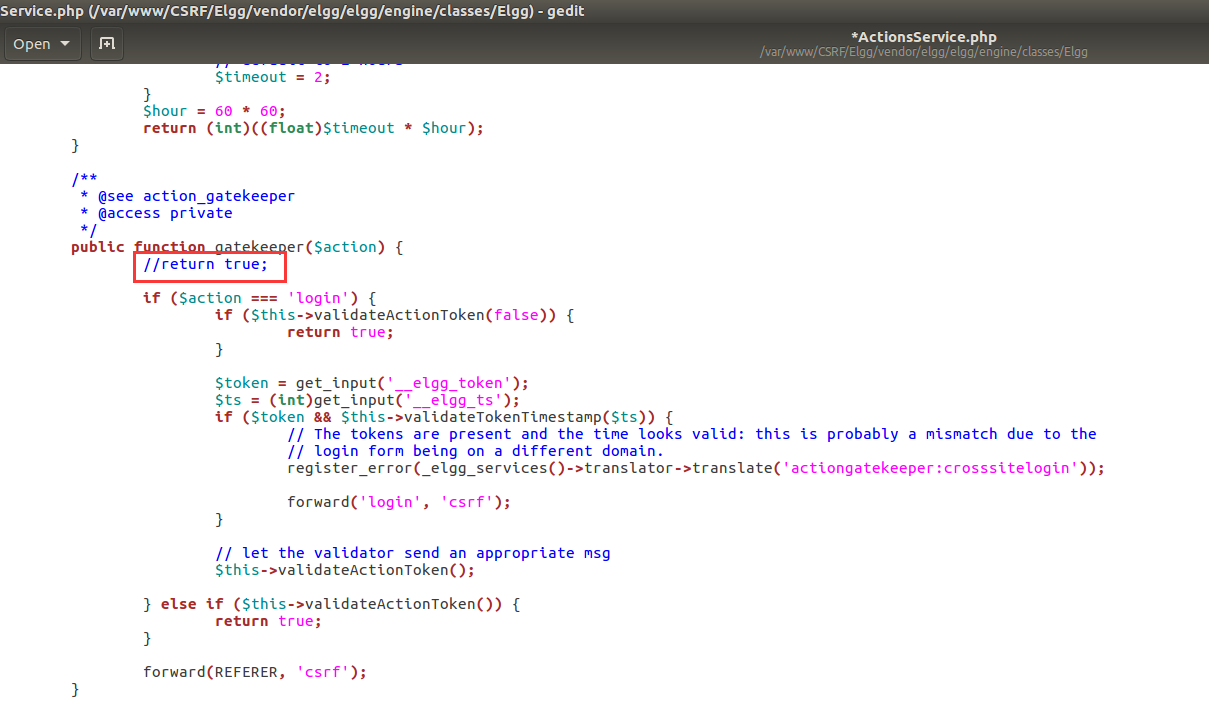
1、Boby无法登陆Alice账号前提下如何得到Alice的guid？

答：Boby主动添加Alice为好友，用live工具观察报头，可以看到其中会有Alice的guid=42

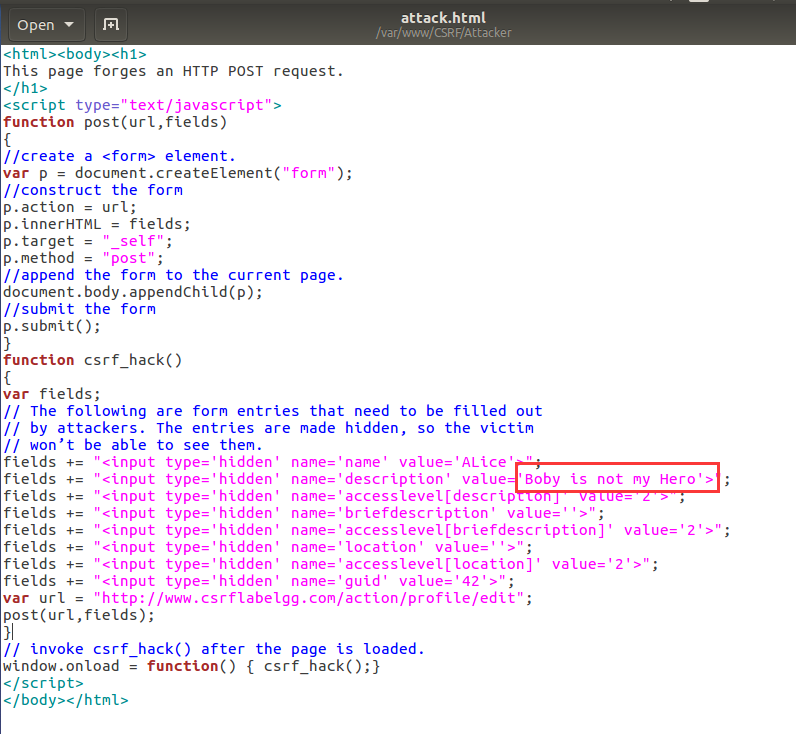
2、如果需要让任意访问恶意网站的人都可以修改掉其个人简介。但是，又不能事先知道谁会访问这个恶意网站。因此，还能通过CSRF来完成修改个人简介的操作么？为什么？  
答：可以，如果能够利用函数动态获取当前访问该攻击网站用户的guid，即可修改对方的个人简介。本题中我们是观察http的报头手动获取访问用户的guid。

**Task4: Implementing a countermeasure for Elgg**

按照路径打开php文件注释掉actioin.php文件里actioin\_gatekeeper函数最前面的return true



进行Task3中相同的攻击，修改个人简介内容。

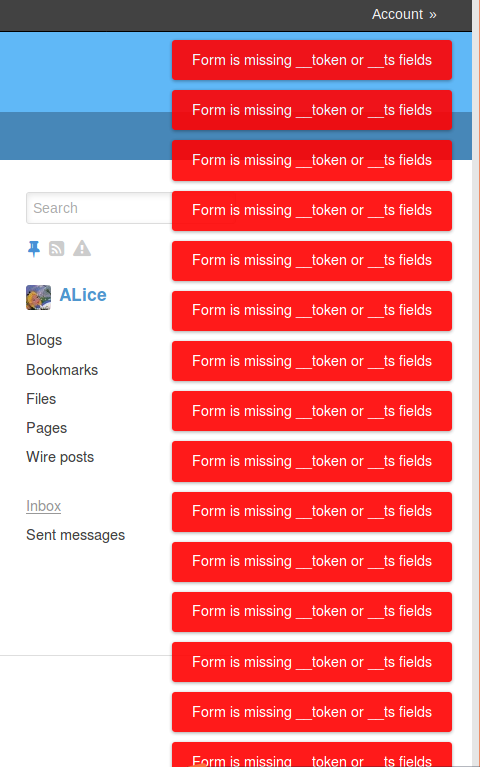


Alice访问攻击网站，会发现一直停留在如下页面，不会自动跳转。

抓包工具不断捕获到数据包，如下图：



返回Boby的账号，发现很多提示内容，意思是token验证未通过，说明防止CSRF攻击成功。



分析原因：

攻击失败的原因是token的缺失，这是因为注释掉return true之后会执行后面的代码，也就是validate\_action\_token验证时间戳函数。代码中有一个MD5加密函数，所以即便攻击者能够知道guid，只要没有md5加密用的密钥，同样不能伪造出正确的token出来，也就不能进行CSRF攻击。

