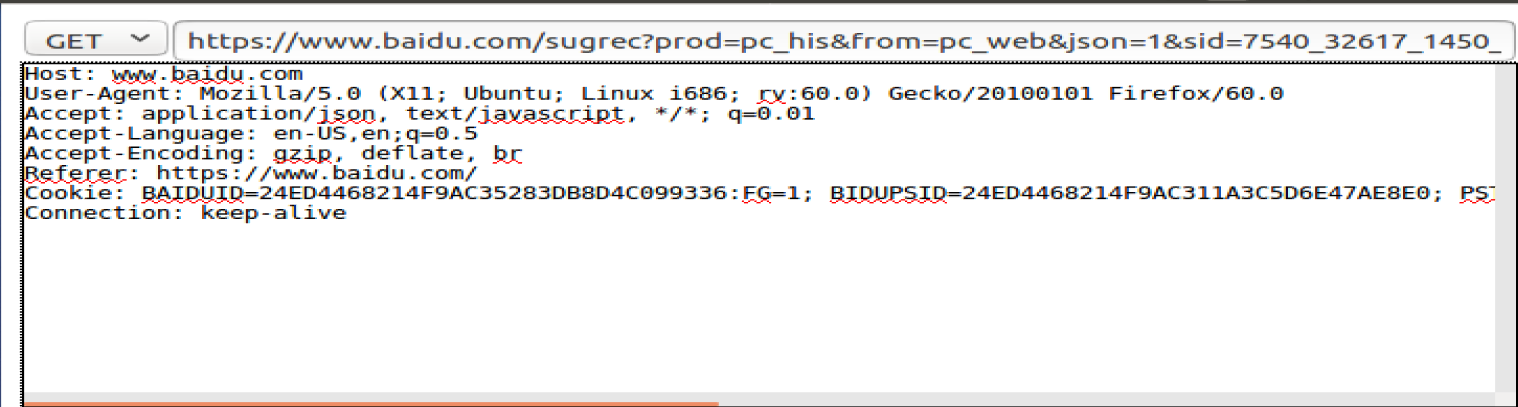
Lab 4

姓名：郭雅琪 学号：57118104

Task1:Observing HTTP Request

在火狐浏览器上安装Http Header Live插件后，进行抓包。

以下是抓到的一个GET请求：



具体内容如下：

Host: www.baidu.com

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

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

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

Accept-Encoding: gzip, deflate, br

Referer: https://www.baidu.com/

Cookie: BAIDUID=24ED4468214F9AC35283DB8D4C099336:FG=1; BIDUPSID=24ED4468214F9AC311A3C5D6E47AE8E0; PSTM=1599579077; BD\_UPN=133352; BD\_HOME=1; H\_PS\_PSSID=7540\_32617\_1450\_31253\_7605\_32117\_32691

Connection: keep-alive

对具体内容的参数进行分析：

Host：表示初始URL中的主机和端口。这里显示的百度的网址。

User-agent：浏览器类型，这里可以看出使用的是火狐浏览器。

Accept:表示浏览器可以接受的MIME类型。

Accpet-Language：浏览器所希望的语言类型。

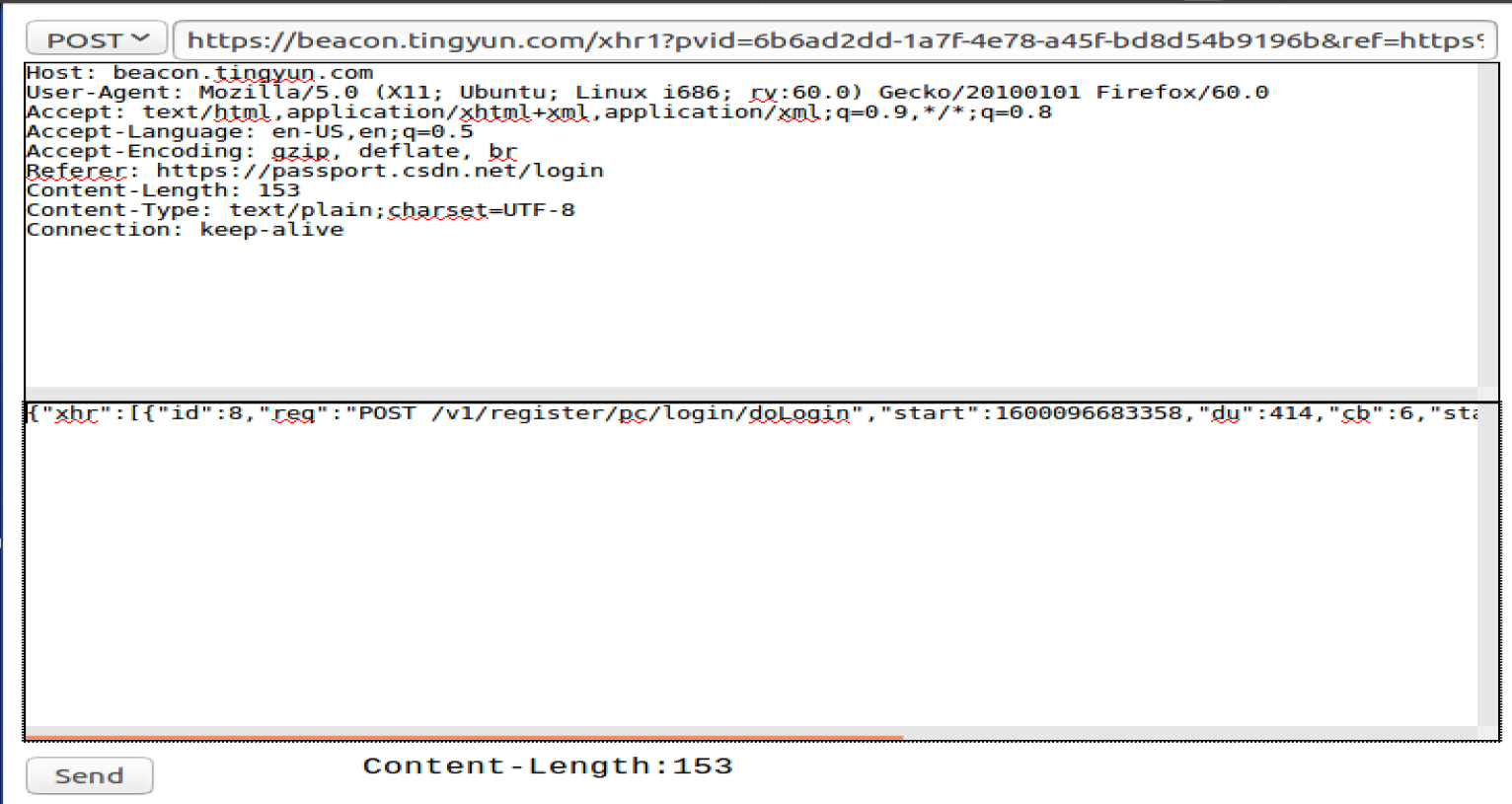
Accpet-Encoding：浏览器能够进行解码的数据编码方式。

Referer：这里的URL代表，用户从该用户出发访问当前请求的页面。这里仍旧是百度，说明应该是刷新页面时得到的http请求。

Cookie：相当于网站发给当前用户的一个身份验证，可以克服http无状态连接带来的使用不便。

Connection：表示是否需要持久连接。keep-alive表示需要持久连接。

以下是抓到的一个POST请求：



和GET最大的区别是，POST请求内容在content中，并且报头中多出了两个参数Content-Length和Content-Type。其中Content-Length指明内容的长度为153，Content-Type指明内容的类型为文字，以及编码方式为UTF-8。

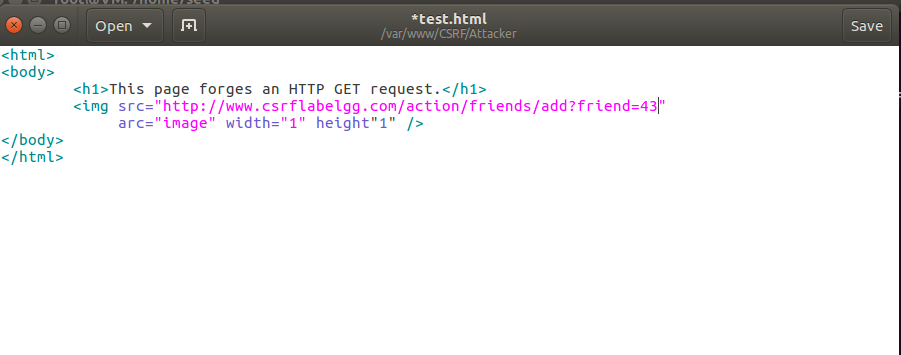
Task 2:CSRF Attacking using GET Request

首先登录一个boby以外的账号，这里登录了Charlie的账号，添加Boby为好友，通过Http Header Live看到Boby的id是43。



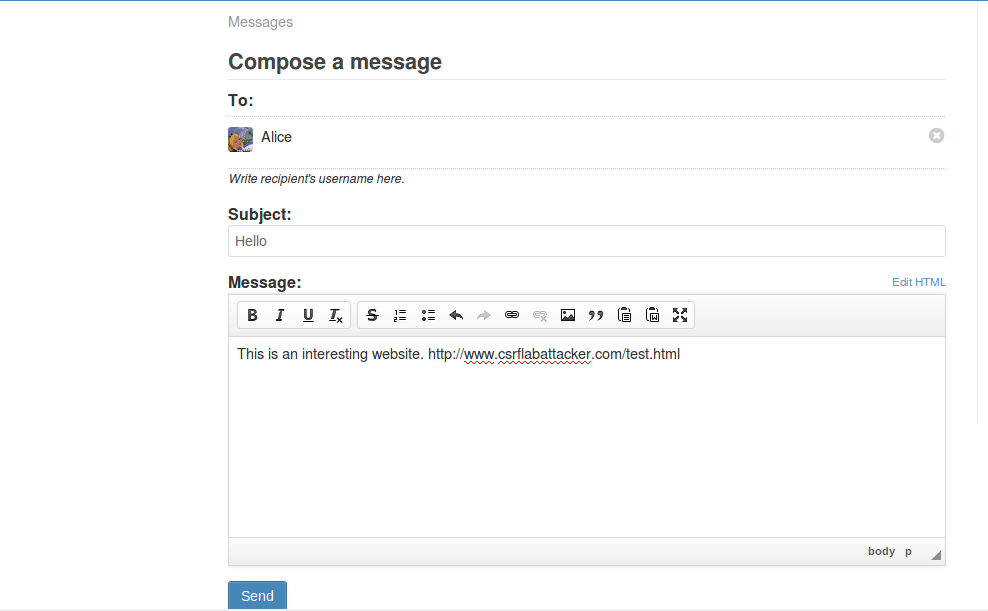
<http://www.csrflabelgg.com/action/friends/add?friend=43&__elgg_ts=1600076050&__elgg_token=bIY6EjWnYgFrKpFd1IajFQ&__elgg_ts=1600076050&__elgg_token=bIY6EjWnYgFrKpFd1IajFQ>

之后编写恶意网页，代码如下：

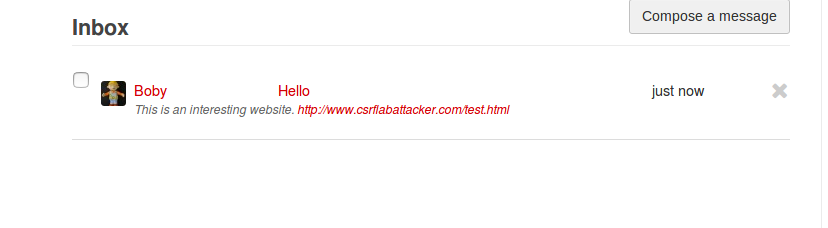


编写好之后放在/var/www/CSRF/Attacker/目录下。

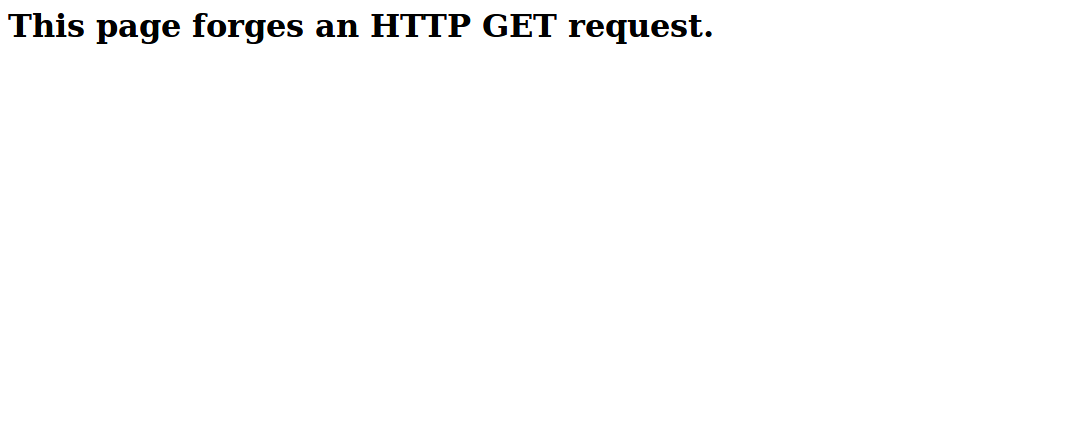
接着登录Boby的账号，给Alice发送邮件，邮件要包含恶意网页的网址，具体内容如下：



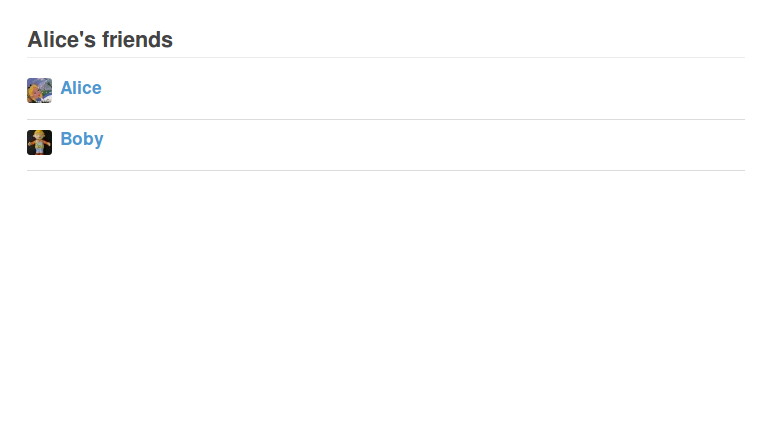
发送成功后退出Boby的账号，登录Alice的账号，发现收到邮件。



打开邮件后点击链接，跳转到恶意网页。



返回后发现已经添加了Boby为好友，说明攻击成功。

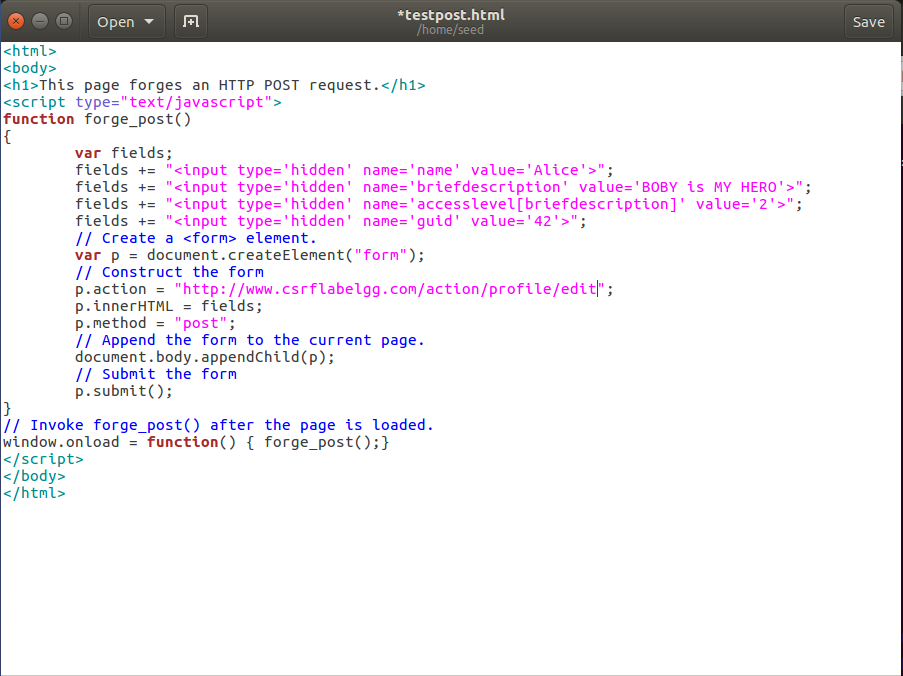


Task3:CSRF Attack using POST Request

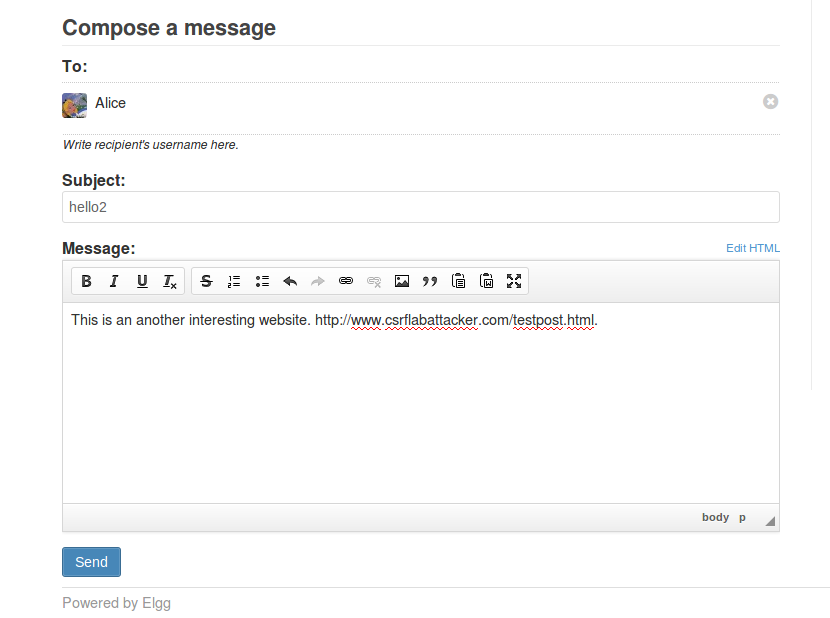
登录Alice账号，右键查看网页源码，找到Alide的guid为42。



然后开始编写恶意网页，具体代码如下：



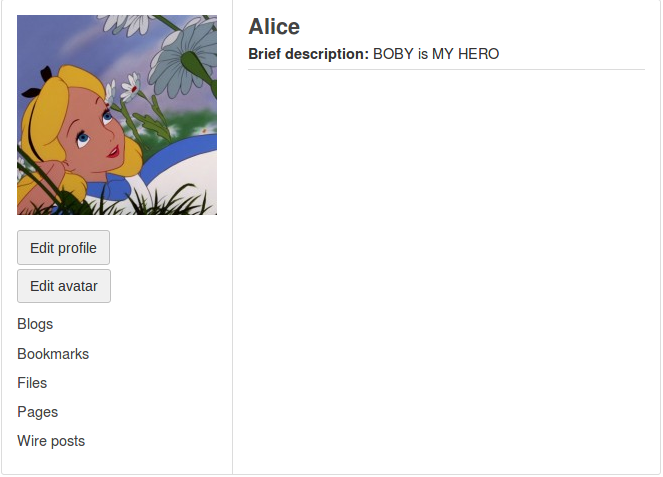
将该文件复制到/var/www/CSRF/Attacker/目录下，再次登录Boby的账号向Alice发送邮件，邮件中包含恶意网页的网址：



发送成功后登录Alice的账号，点击新收到邮件中的网址，跳转到恶意网页。



回到主页后，发现Alice的简介被修改，攻击成功。

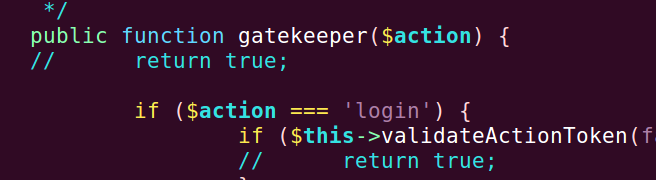


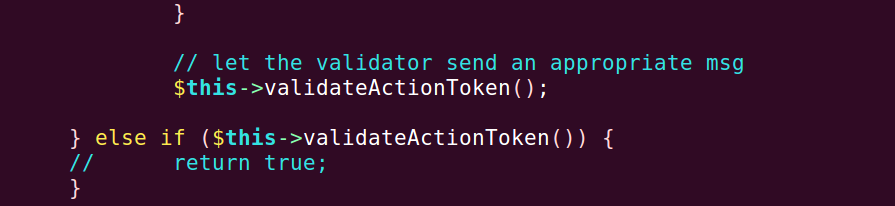
Question 1:可以开一个小号先去加Alice,通过Http Header Live抓包，就可以在URL中看到Alice的id为42。

Question 2:不可以，boby作为攻击者，他并不会获得cookie的信息，他无法自动启动CSFR来修改受害者的信息。

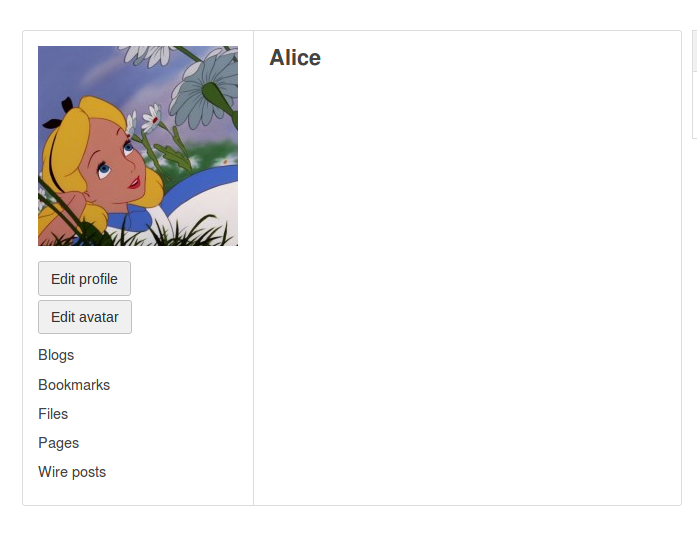
Task4:Implementing a countermeasure for Elgg

修改/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classess/Elgg/ActionsService.php中的函数gatekeeper,将其中的return true全部注释掉。





再次进行之前的攻击，发现攻击不成功。



攻击不成功的原因是此时客户端发送请求的时候附加一个token，服务器端同样产生一个，两个做比对。这个token可以通过Http Header Live监听获得，具体如下：

<http://www.csrflabelgg.com/action/friends/add?friend=45&__elgg_ts=1600089854&__elgg_token=_wouU6rf72fP5gtTOcsEvw&__elgg_ts=1600089854&__elgg_token=_wouU6rf72fP5gtTOcsEvw>

或者可以直接在控制台中输入命令elgg.security.token\_\_elgg\_token获得。