**信息安全实验报告**

**Lab 10 Cross-Site Request Forgery (CSRF) Attack Lab**

**孙铁**

**SA20225414**

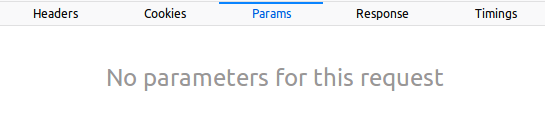
**Task 1**

打开http://www.csrflabelgg.com进入Elgg页面；打开HTTP Header Live插件，登录Boby账号。

捕获HTTP GET请求如下：



使用开发者工具查看字段：

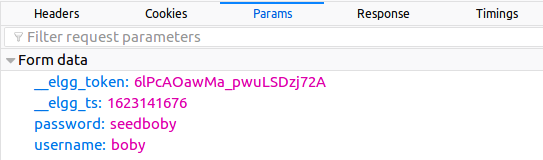


GET请求并无任何字段；

捕获HTTP POST请求如下：



使用开发者工具查看字段：



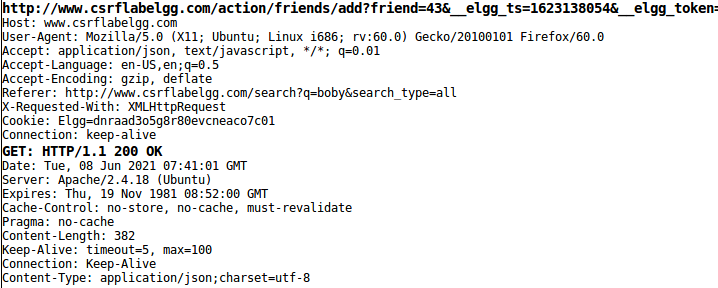
POST请求中出现了四个字段：

\_\_elgg\_token字段与\_\_elgg\_ts字段是Elgg针对CSRF攻击设置的防御机制，本实验中已经禁用，后续攻击中不需要考虑这两个参数；password字段与username字段则是传输的数据，对应Boby登录时的密码和用户名。

GET请求会将数据附加在请求的URL中；而POST请求则会将数据放在请求的数据字段中。

**Task 2**

让Alice向Boby发送好友请求，用HTTP Header Live捕获GET请求如下：



URL为“http://www.csrflabelgg.com/action/friends/add?friend=43”，即Boby的GUID为43，后面的elgg\_ts字段和elgg\_token字段是Elgg针对CSRF攻击设置的防御机制，本实验中已经禁用，后续攻击中不需要考虑这两个参数。

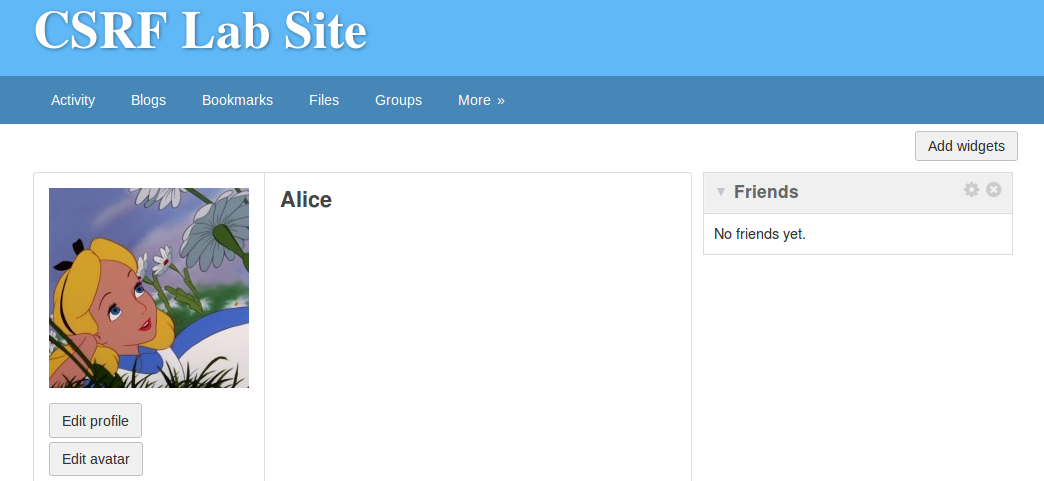
删除Boby的好友位置，接下来要通过针对GET请求的CSRF攻击来让Boby成为Alice的好友。

在/var/www/CSRF/Attacker路径中创建文件index.html：

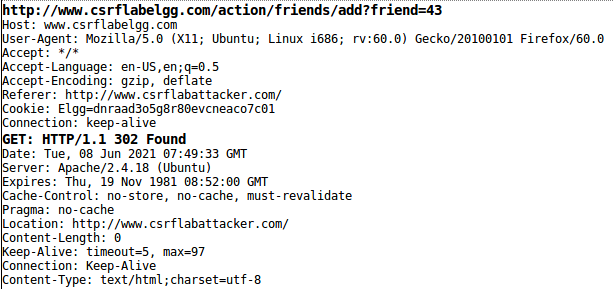


只要打开 www.csrflabattacker.com网站，<img>标签就会自动向src属性指定的URL发送一个HTTP GET请求。将src指向Alice向Boby添加好友时的URL，只要Alice账号登录时打开了恶意网站，就相当于主动添加Boby为好友。

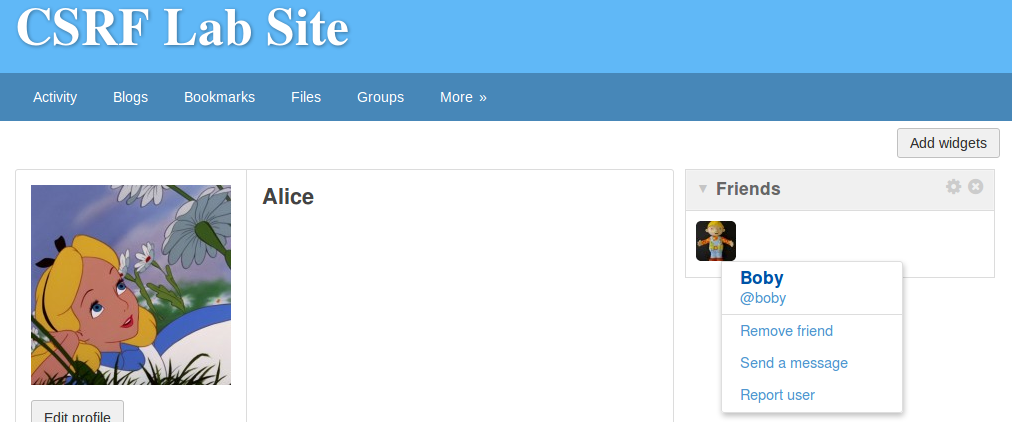
登录Alice账号，此时Boby不是Alice的好友：



打开恶意网站：www.csrflabattacker.com，此时HTTP Header Live捕获到了一个GET请求：



刷新页面，发现此时Boby成为了Alice的好友：



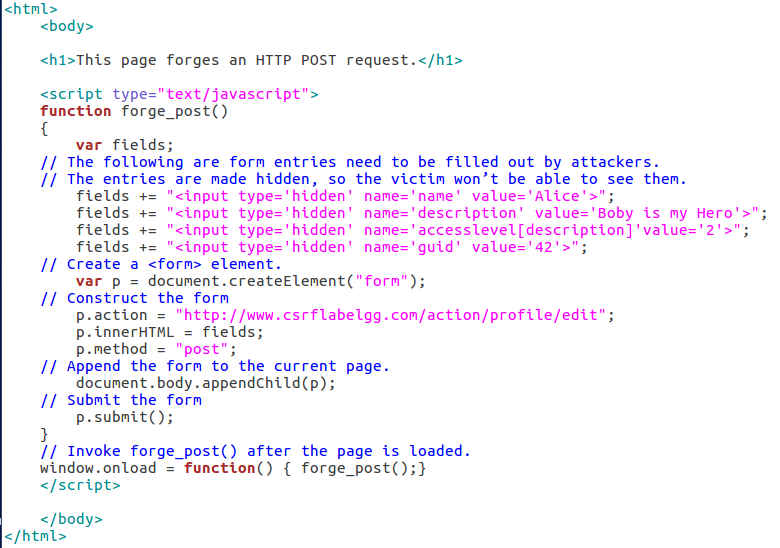
**Task 3**

登录Alice账号修改个人介绍，用HTTP Header Live捕获POST请求如下：



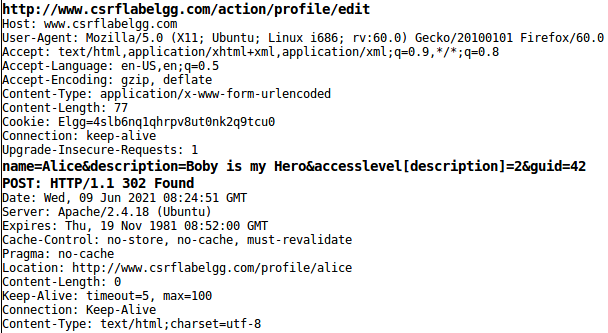
由此可以得到URL为“http://www.csrflabelgg.com/action/profile/edit”， Alice的GUID为42，descripiton字段的内容即为修改的个人介绍。

在/var/www/CSRF/Attacker路径中创建文件index2.html：

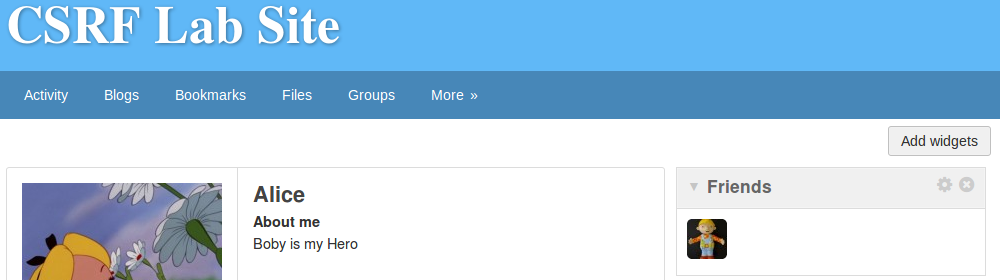


在<script>标签内动态创造一个表单，表单构造好之后会被添加到恶意网站上，当恶意网站被打开，forge\_post函数自动被调用，运行到p.submit函数将整个表单发送出去，就相当于向目标URL发送了一个包含表单内容的POST请求。

登录Alice的账号并打开恶意网站，此时HTTP Header Live捕获到了一个POST请求：

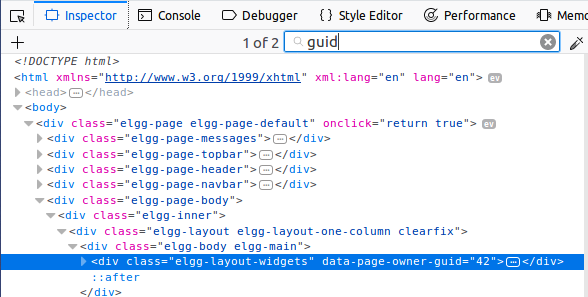


此时页面刷新，Alice的个人介绍被改变：



**Question1：**

访问Alice的主页，然后使用开发者工具查看网页源码即可找到Alice的GUID。



**Question2：**

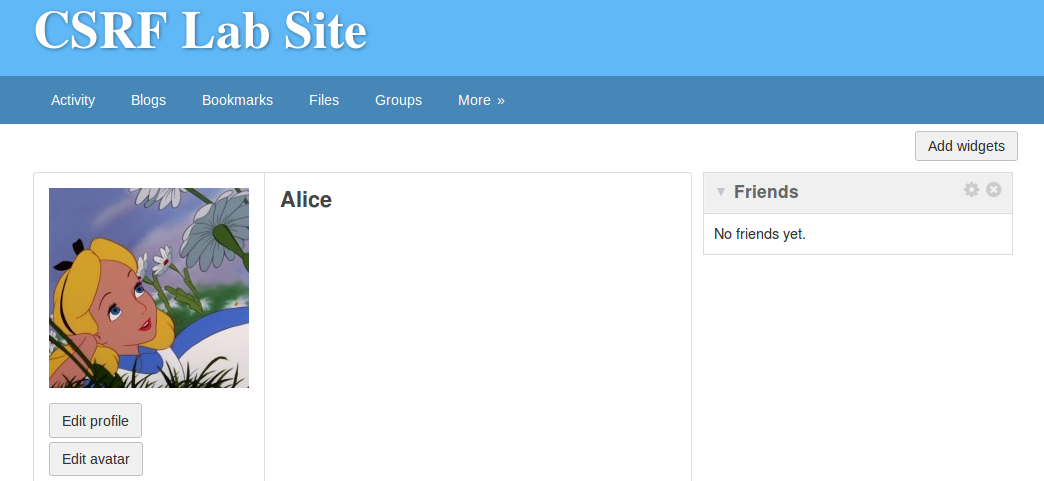
我觉得可以，在防御措施关闭的情况下，攻击成功关键在于能否动态获取访问网页的GUID及其用户名，而这些参数都可以在主页的源码中查找到。

**Task 4**

打开/var/www/CSRF/ Elgg/vendor/elgg/elgg/engine/classes/Elgg路径下的ActionsService.php文件，将gatekeeper函数第一行return true注释掉，让后面的代码得以执行：

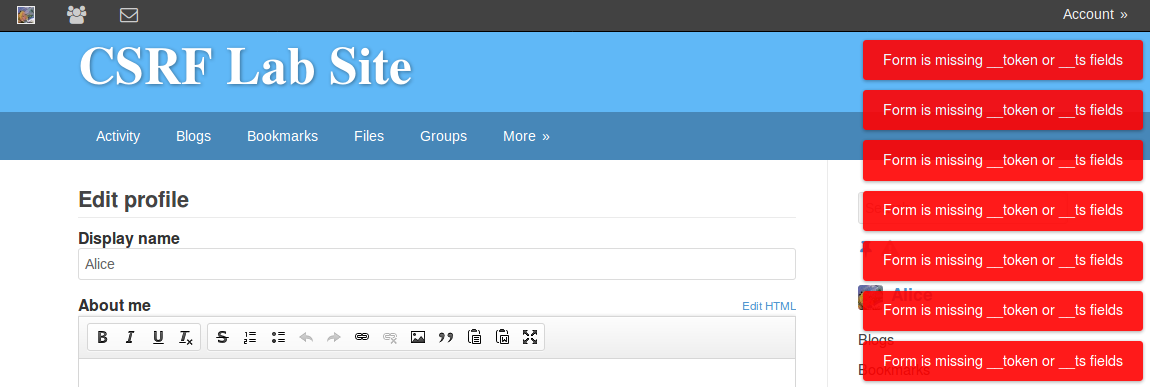


删除Boby的好友位置，重新进行Task2中的攻击：



发现攻击没有成功。

清空Alice的个人介绍，重新进行Task3的攻击：



Alice的个人介绍并未被修改，而且个人介绍的编辑页面中会不断弹出信息警告elgg\_ts字段和elgg\_token字段缺失，说明保护措施已经开启。

Elgg网站的机密令牌是两个机密值elgg\_ts字段和elgg\_token字段，它们由ActionsService.php文件生成并添加到每个网页中。

在HTTP Header Live中捕获的POST请求中的机密值：



将return true注释掉之后，gatekeeper函数就能够根据elgg\_ts和elgg\_token字段来验证请求是否跨站。由于同源策略，浏览器会阻止<script>中的代码访问Elgg页面的任何内容，请求无法获取elgg\_ts和elgg\_token字段，也就让这个跨站请求无法通过验证，导致攻击失败。