**Lab4实验**

**57118111 王雯**

1. **实验目的：**

本实验的目的是理解跨站请求伪造（CSRF）攻击。CSRF攻击涉及受害者用户，受信任的站点和恶意站点。受害用户在访问恶意站点时会与受信任的站点进行活动会话。恶意站点将对受信任站点的HTTP请求注入受害用户会话，从而造成损害。在本实验中，将使用CSRF攻击来攻击社交网络应用程序。

**二、实验环境准备**

The ElggWebApplicationElgg应用程序

DNS Conﬁguration DNS配置

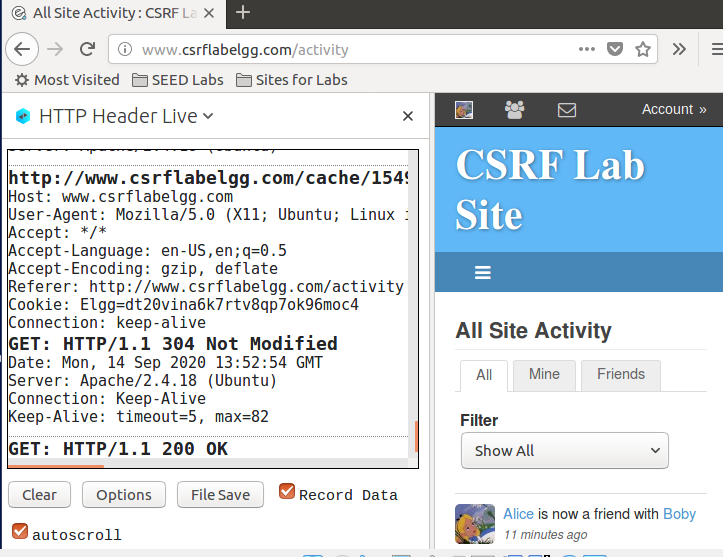
ApacheConﬁgurationApache配置



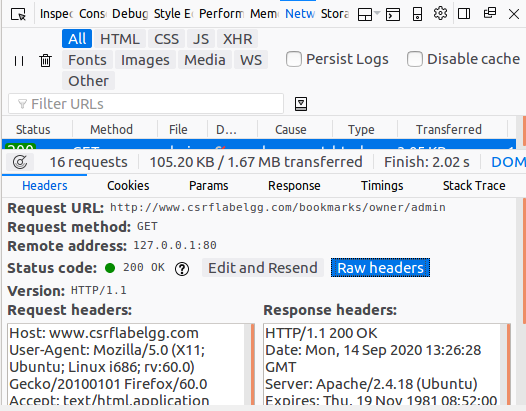
**三、实验任务**

**Task1: ObservingHTTPRequest**

首先我们根据实验要求，添加HTTPHeaderLive的插件，来获取web的一个信息。

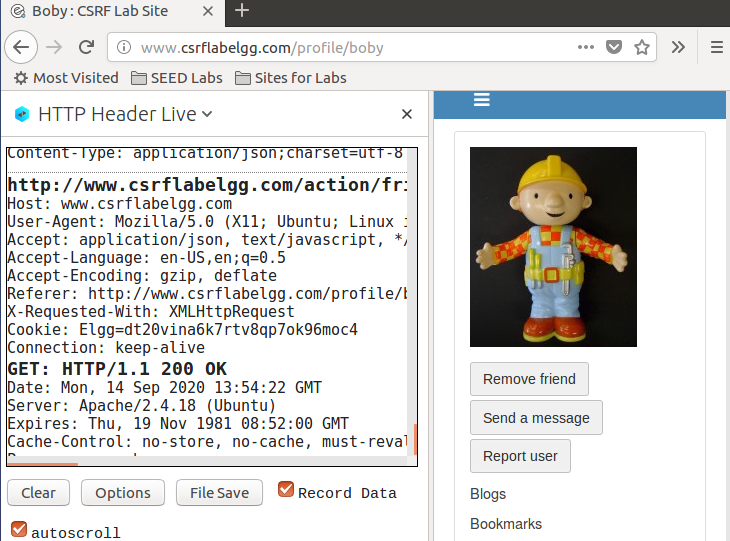


使用浏览器的工具栏得Web-console来跟踪HTTP



**Task2: CSRF Attack using GET Request**

首先我们通过添加boby，并通过HTTP Header Live捕捉添加好友的HTTP请求，可以发现添加boby为好友的序号为43，之后我们通过已知的好友序号，构建恶意网站。

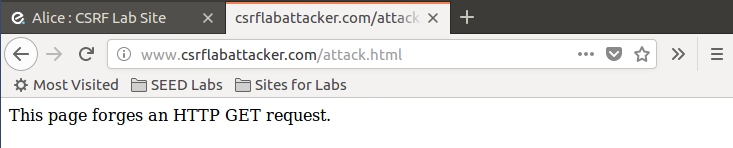




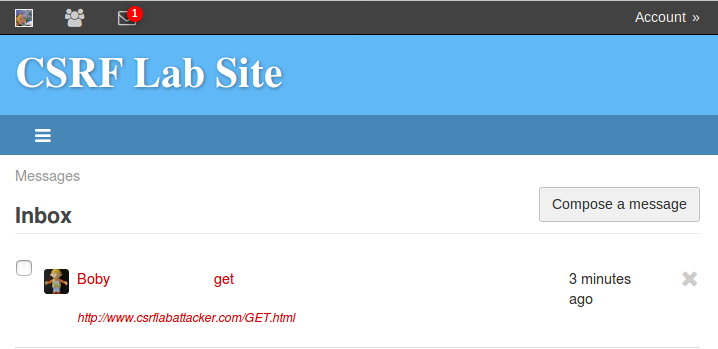


在恶意网站[www.csrflabattacker.com](http://www.csrflabattacker.com)中上传制作好的网页，把这个网页放在/var/www/CSRF/Attacker文件夹中





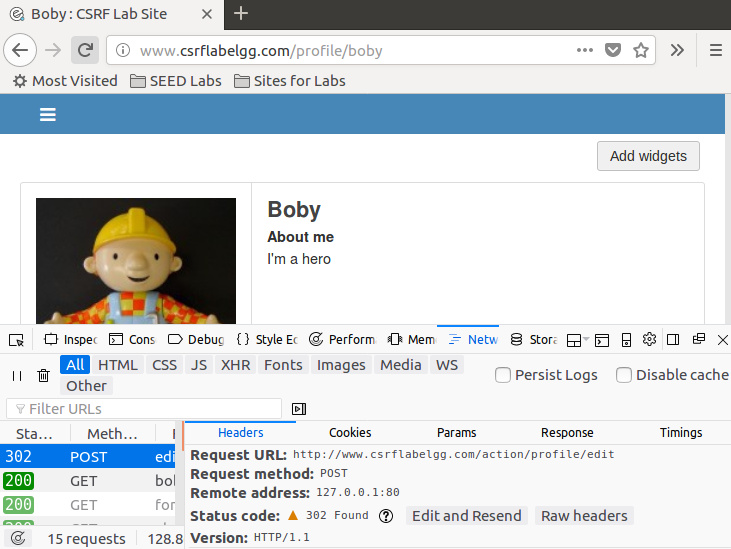
之后boby为了吸引alice访问恶意网站，先登录上boby的网站，给alice发送邮件，然后alice收取boby的邮件，并点击链接，发现加了boby

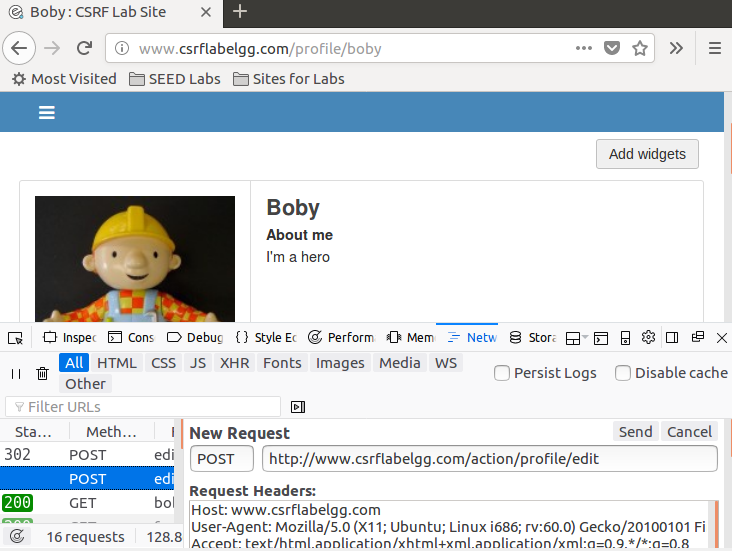


**Task3: CSRFAttackusingPOSTRequest**

目标是修改Alice的个人简介内容为“Bob is my hero！”，Elgg接受修改个人简介的服务器脚本是/pfofile/edit.php，这个脚本接受GET或者POST的请求。

为了使用CSRF跨站请求攻击，首先我们需要观察post请求的格式。





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: zh,en-US,en;q=0.7,en;q=0.3

Accept-Encoding: gzip, deflate

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

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

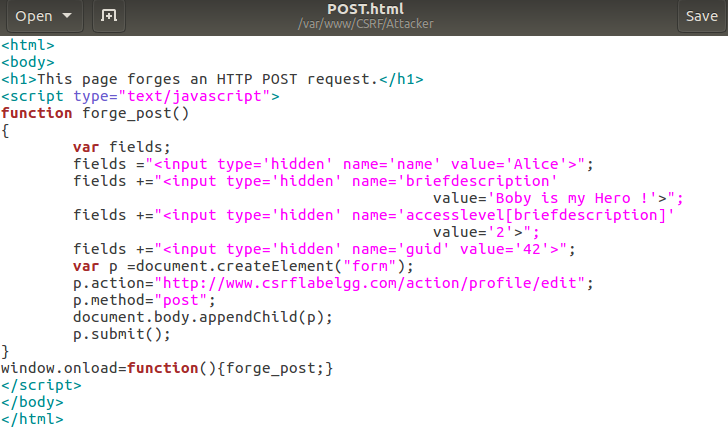
Content-Length: 550

Cookie: Elgg=vimdrt98nn0s93rvn418kpdei2

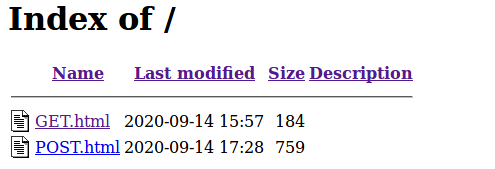
Connection: keep-alive

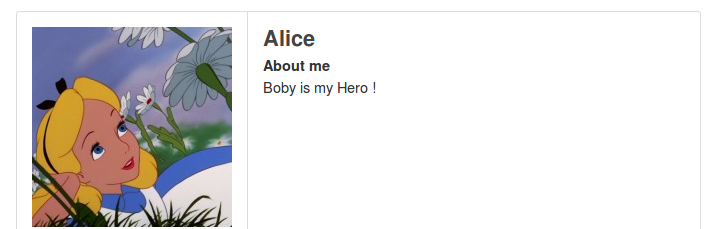
Upgrade-Insecure-Requests: 1

之后构建恶意网站









通过task3我们成功的将alice的个人简介内容变为了boby is my Hero!

实验手册中问题：

问题1:伪造的HTTP请求需要Alice的用户id (guid)才能正常工作。如果波比目标

特别是Alice，在攻击之前，他可以找到获取Alice的用户id的方法。Boby不知道Alice的Elgg密码，所以他无法登录Alice的账户获取信息。请描述如何解决这个问题。

答：在Alice登录自己的网站页面时，网站产生Cookie信息返回给浏览器，此时可以通过正常请求到网站登录，在未退出登录是点开攻击网页时，浏览器在接受攻击代码后网站会携带用户的Cookie信息，向网站发起请求，以至于会被恶意执行。

问题2:Boby是否想对访问其恶意网页的任何人发起攻击。

在这种情况下，他事先不知道谁正在访问web页面。他还能启动CSRF吗

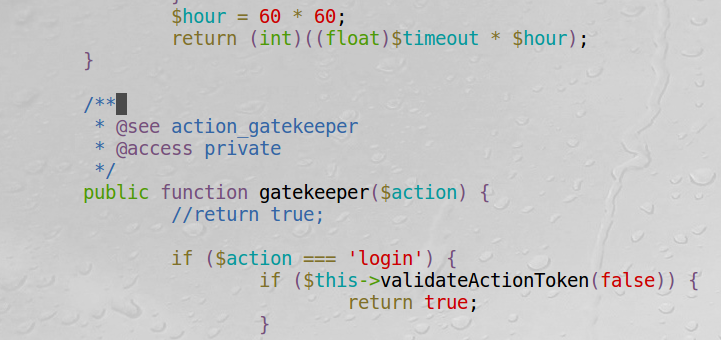
修改受害者elgg档案的攻击?请解释一下。

答：可以。因为CSRF攻击是在受害者不知情的情况下，以受害者的名义伪造请求发送受攻击站点，不需要权限的自动操作，boby作为攻击者，然而他并不会获得Cookie的信息，他无法自主启动CSRF来修改受害者的信息。

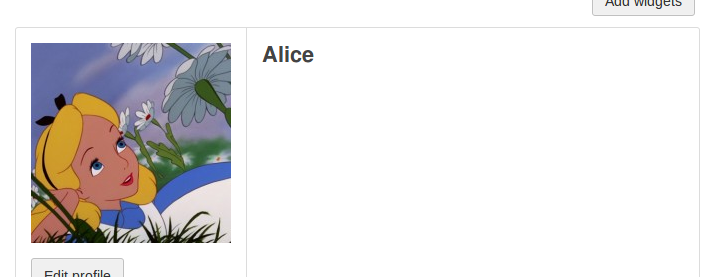
**Task4: Implementingacountermeasurefor Elgg**

CSRF攻击的主要原因是服务器无法区别跨站请求。而在防御 CSRF 攻击主要有三种策略：验证 HTTP Referer字段；在请求地址中添加 token 并验证；在 HTTP 头中自定义属性并验证。使用refer需要报告浏览的隐私，一般使用一个秘密token。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。

打开防护措施**：**到目录/ var / www / CSRF / Elgg / vendor / elgg / elgg / engine / classes / Elgg，然后在ActionsService.php文件中找到功能Gatekeeper。注释掉第一句“return true”，攻击会失败。



再按照task3的步骤来一遍，我们会发现，攻击失败



**四、实验总结**

通过这次实验，我们对CSRF攻击有了更多的了解，同时也对HTTPHeaderlive所抓包的网络信息进行分析，对web-console对HTTP的追踪信息，对GET，POST报文有了更为生动的理解。在实验过程中，在对添加boby为好友时，要注意的是boby的好友号使我们使用http header live所获取的43。