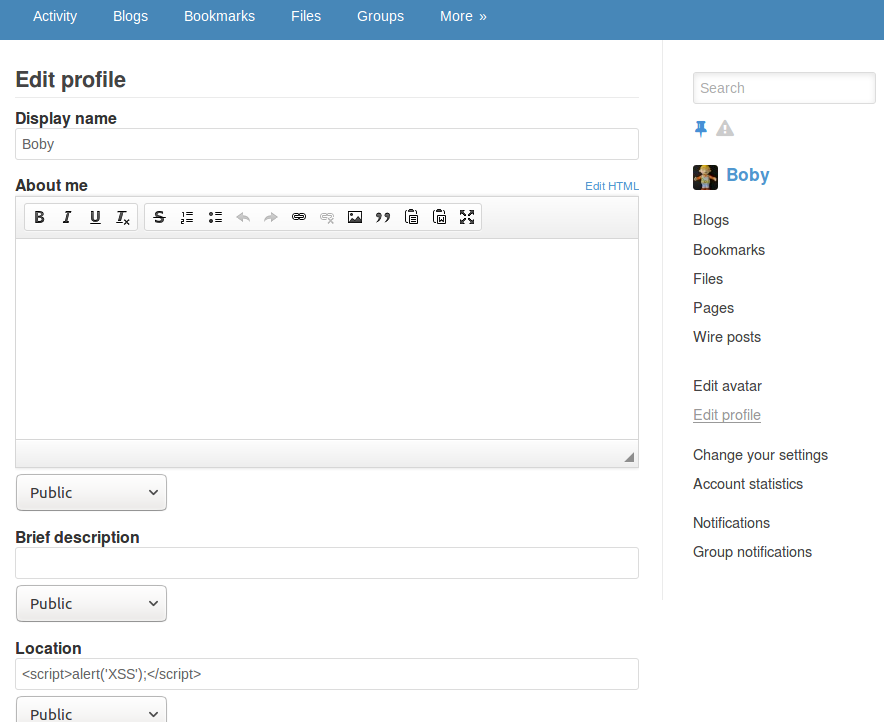
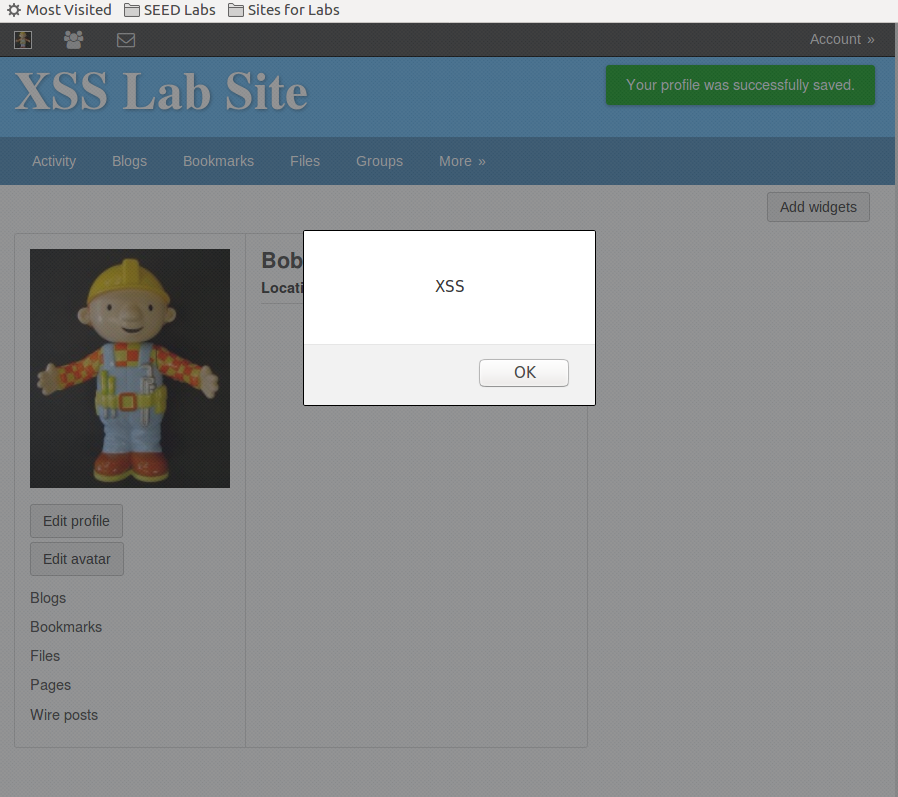
**Cross-Site Scripting Attack Lab**

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

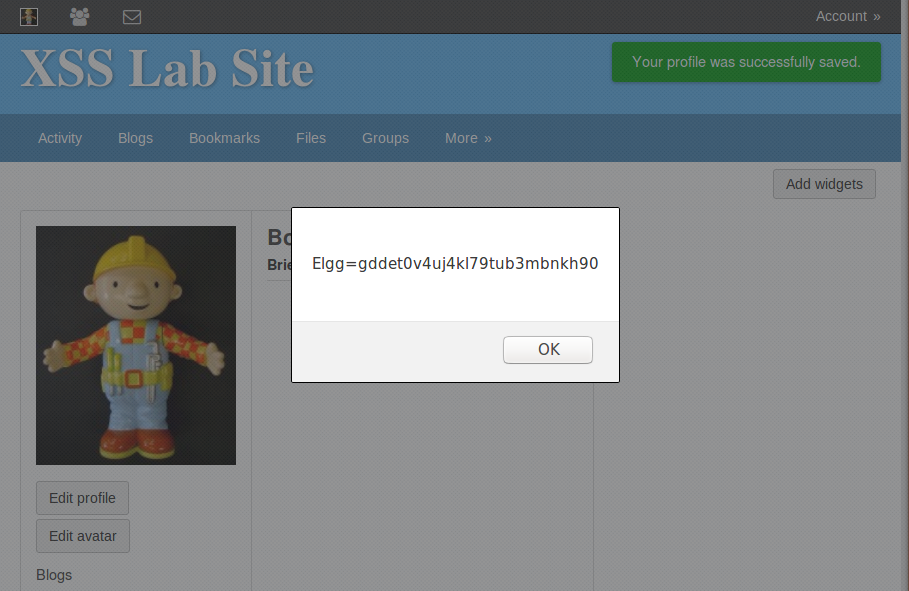
在个人资料栏中添加代码，即可在用户界面显示窗口





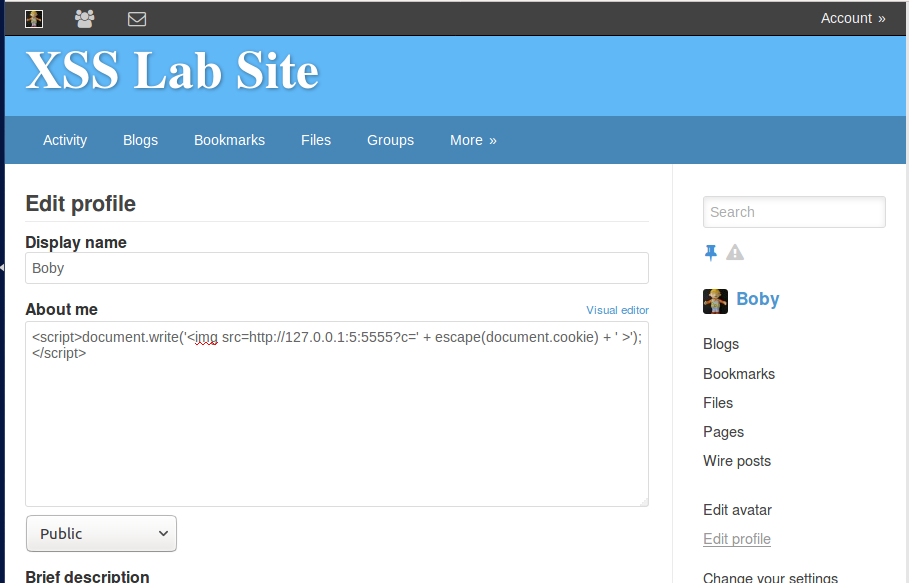
**Task 2: Posting a Malicious Message to Display Cookies**

同任务一，添加代码，出现显示cookie的弹窗

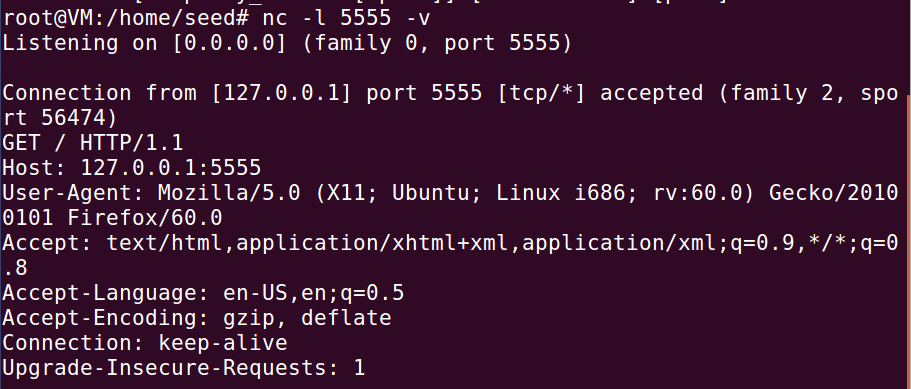


**Task 3: Stealing Cookies from the Victim’s Machine**

通过让恶意JS代码插入<img>标签并将其src属性设置为攻击者的机器



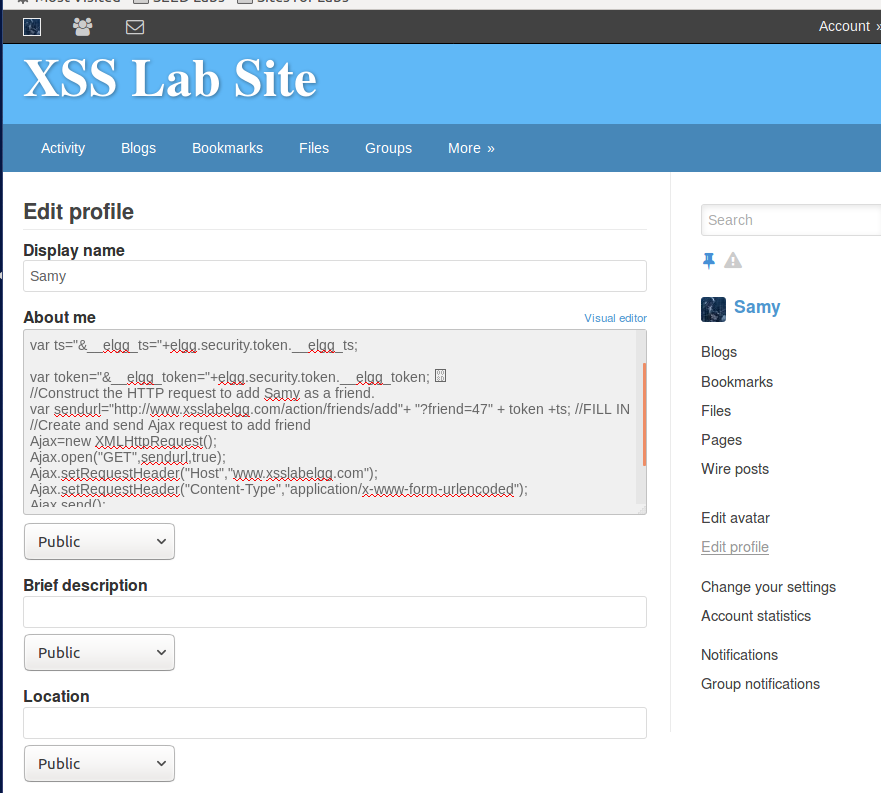
输入nc命令，在浏览器中访问127.0.0.1:5555端口，发现成功监听



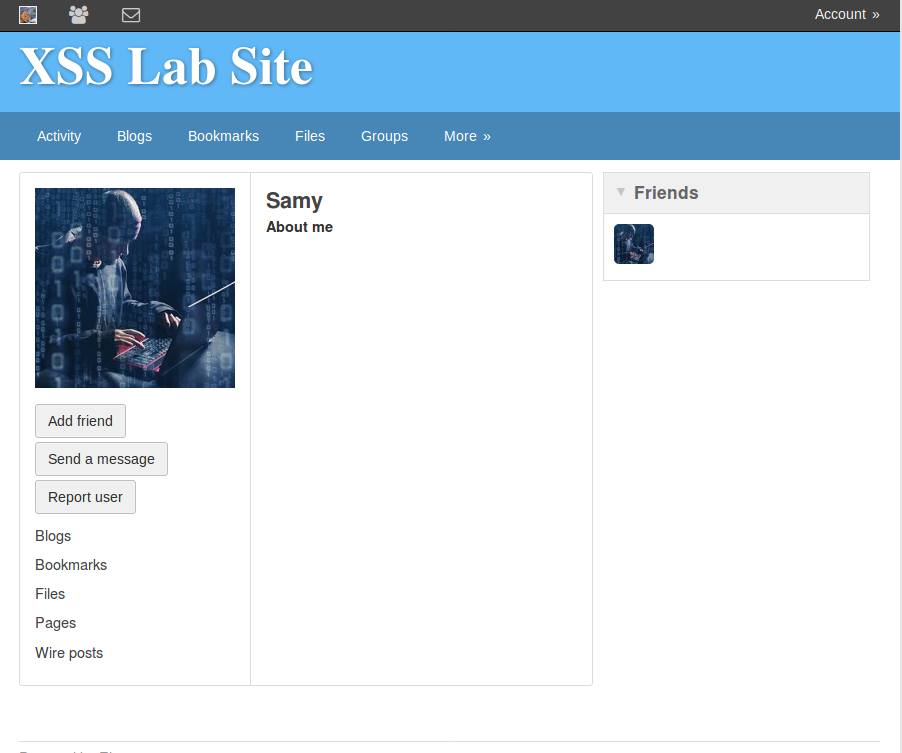
**Task 4: Becoming the Victim’s Friend**

在HTTP Header Live工具下进行添加好友功能，从而获取信息，Samy的guid为47





运行代码之后，发现Alice登录Samy主页后成功地添加了Samy的好友。



**Question 1:**

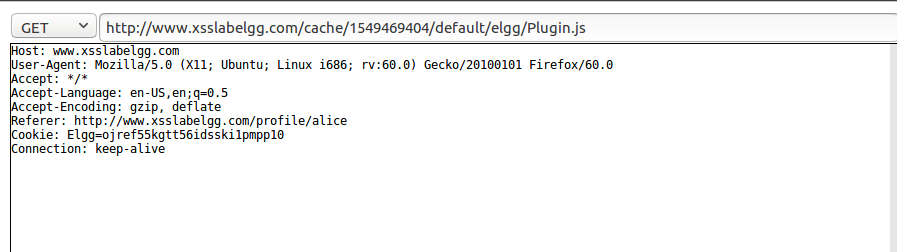
这两行代码的作用是自动从JS变量中获取时间戳和密钥，如果没有这两行代码，会被网站的系统拦截，导致攻击失败。

**Question 2:**

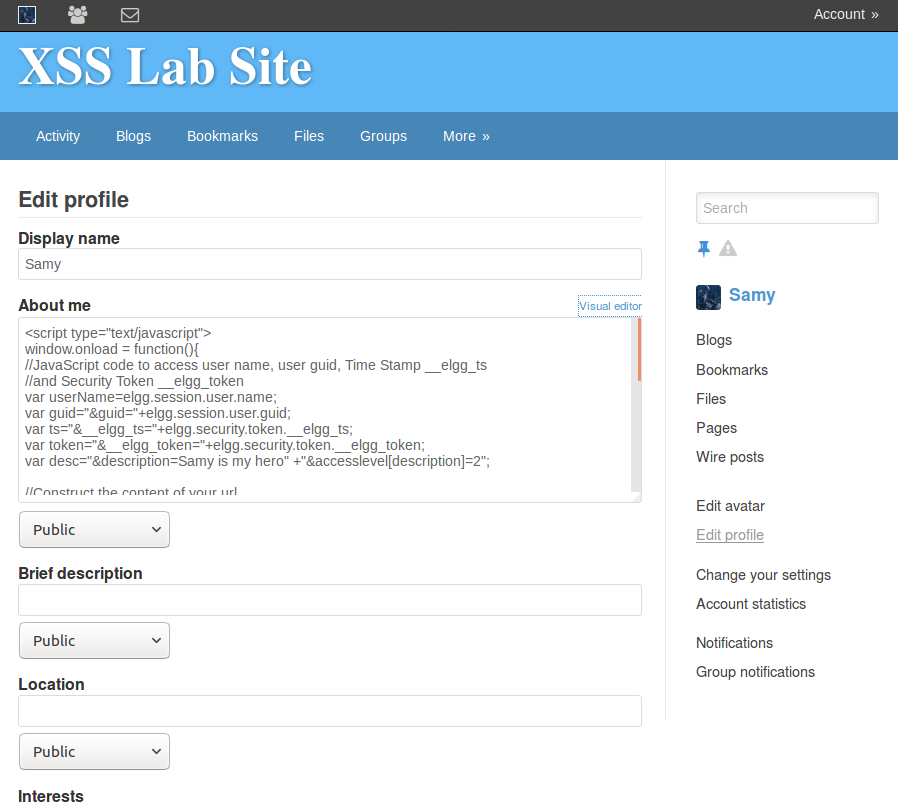
如果不能转为文本编译模式，那么攻击将无法进行，代码会直接以文本的方式展示在主页上。因为非文本模式，网站会自动添加额外代码，破坏攻击程序的完整性。

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

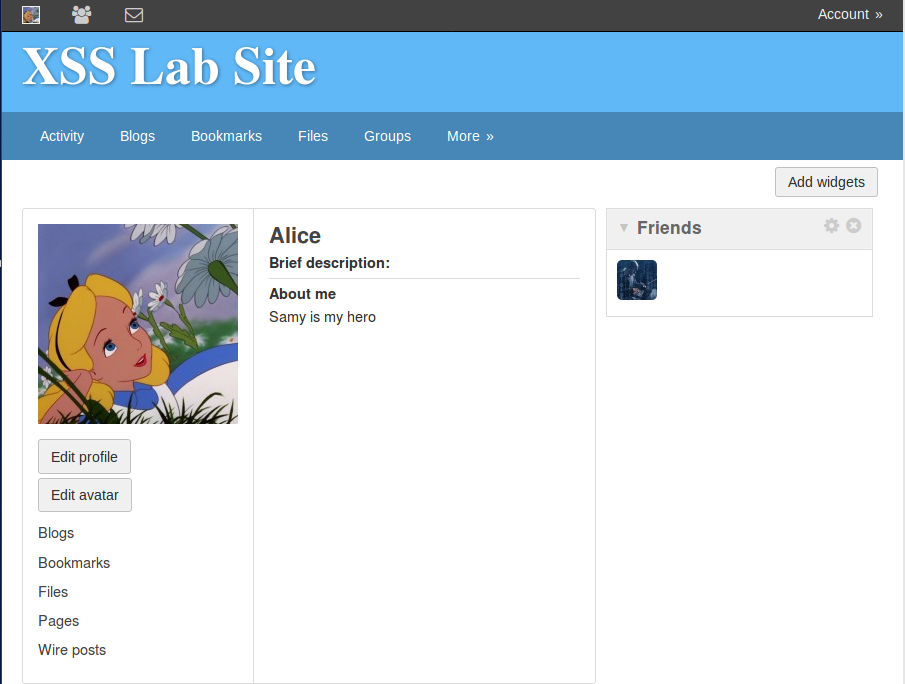
此次任务的目的是在受害者访问Samy的页面时修改受害者的个人资料。



将代码写入个人资料界面



可以看到，在访问了Samy的界面之后，Alice的个人资料更改了，说明XSS攻击成功。



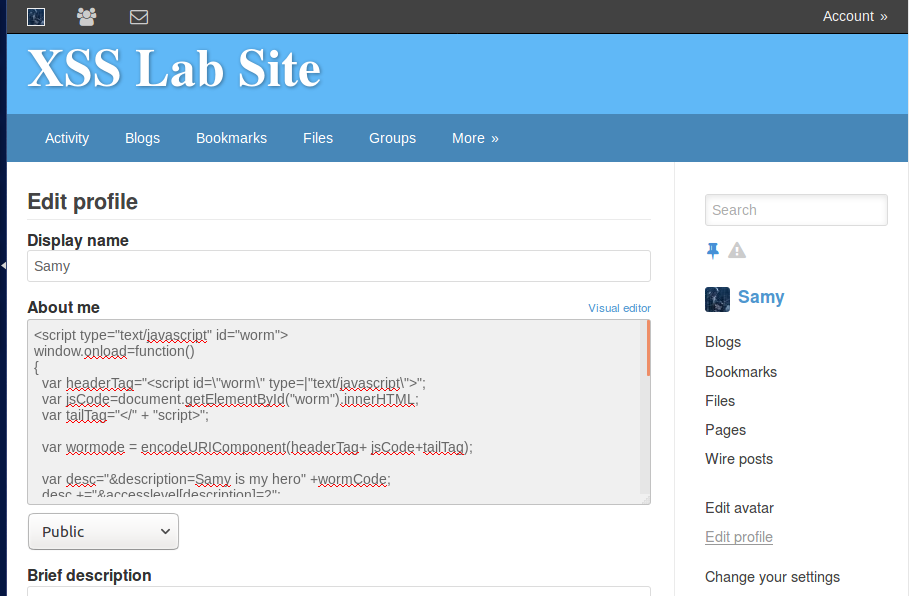
**Question 3:**

删除这句话，再次实行攻击，会发现XSS攻击无法进行传播。该行代码的作用为，不修改本身的资料。如果删除，就会导致在界面刚开始Samy的主页就会被攻击程序修改，原本的代码就会被”Samy is my hero“覆盖，这样恶意程序就无法传播。

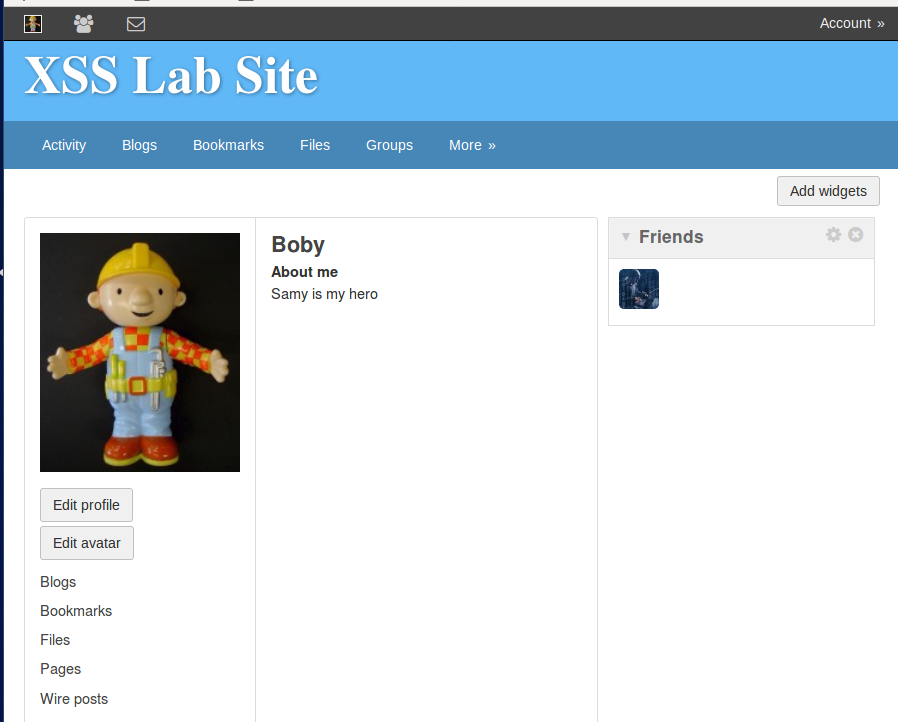
**Task 6: Writing a Self-Propagating XSS Worm**

**通过DOM实现自我传播的XSS蠕虫**

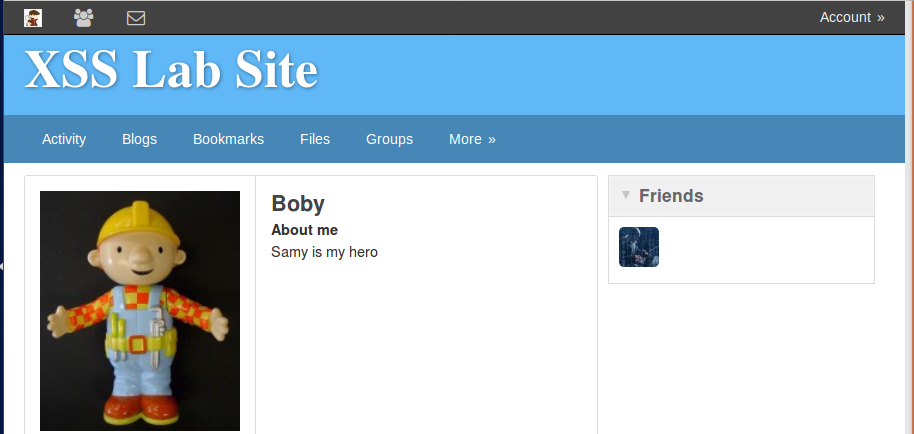
将相关js代码放置在Samy主页资料中



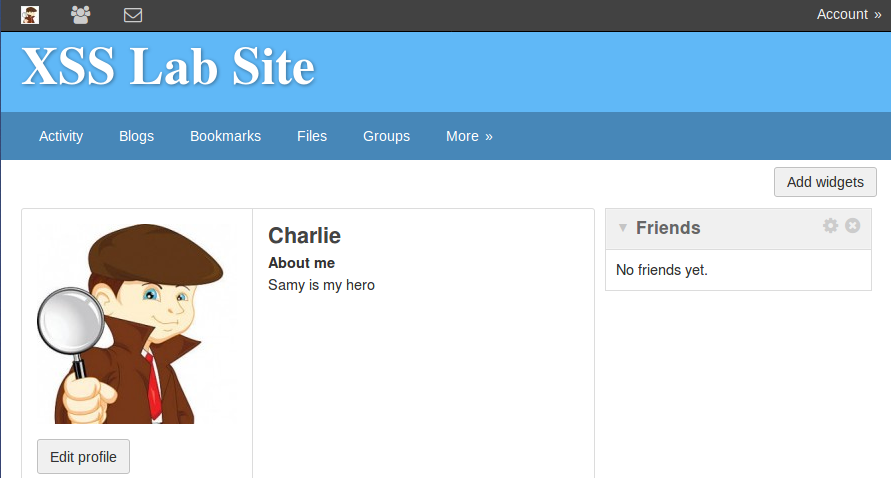
Boby在访问Samy主页之后，个人资料也被修改



接下来我们登录Charlie的账号去访问Boby的主页

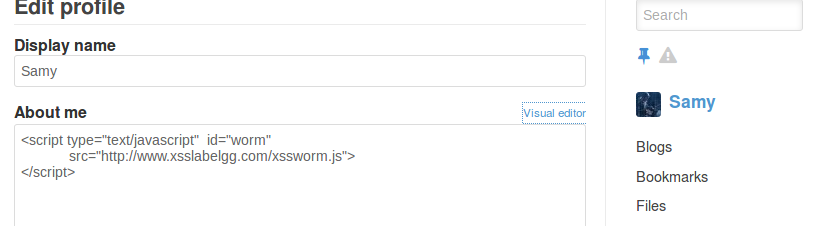


可以发现Charlie的主页资料也被修改，因此可以推断蠕虫病毒传播成功

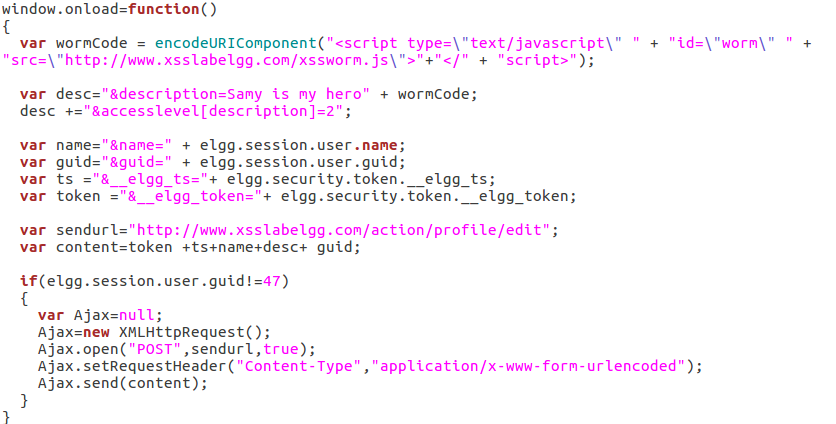


**通过link方式实现**

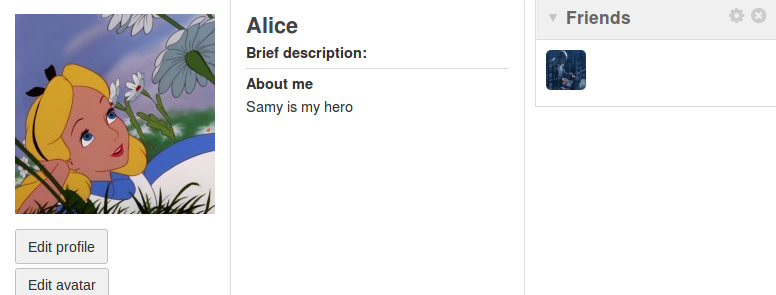
在Samy的主页放入Link链接

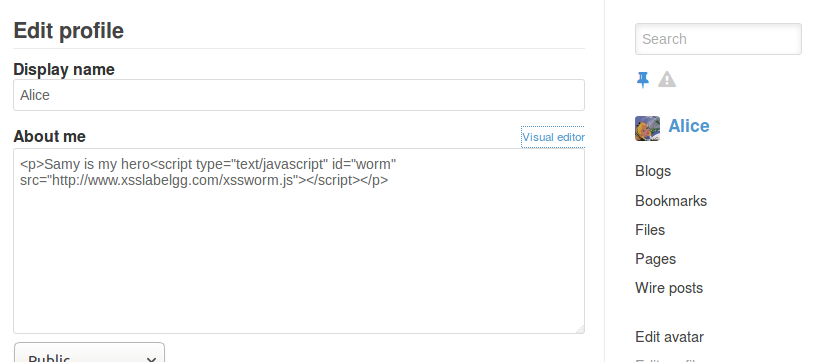


在文件夹中创建xssworm.js，注意不加<script>

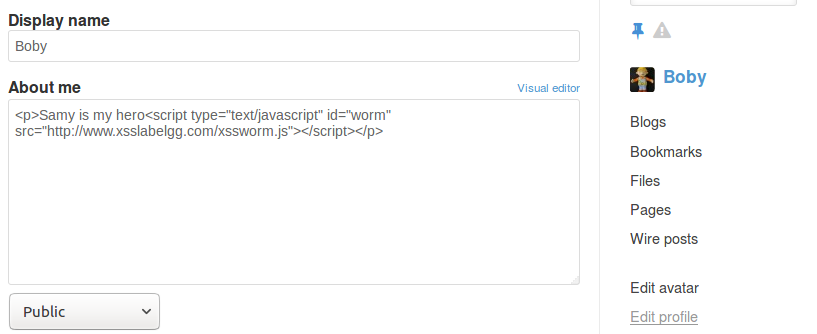


登录Alice的账号访问Samy主页，发现Alice主页也被代码覆盖





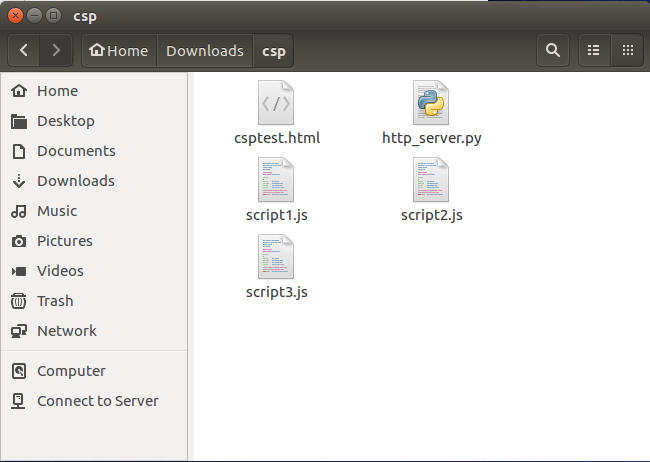
使用Boby的账号访问Alice，发现主页资料也被修改，说明XSS蠕虫攻击成功。

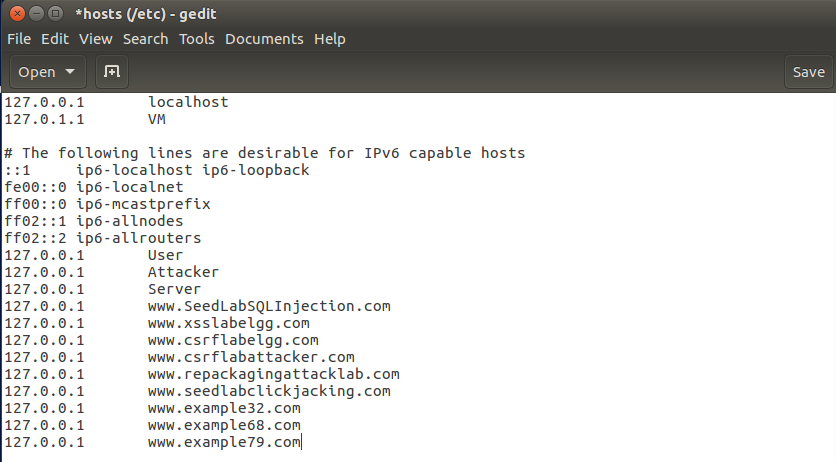


**Task 7: Defeating XSS Attacks Using CSP**

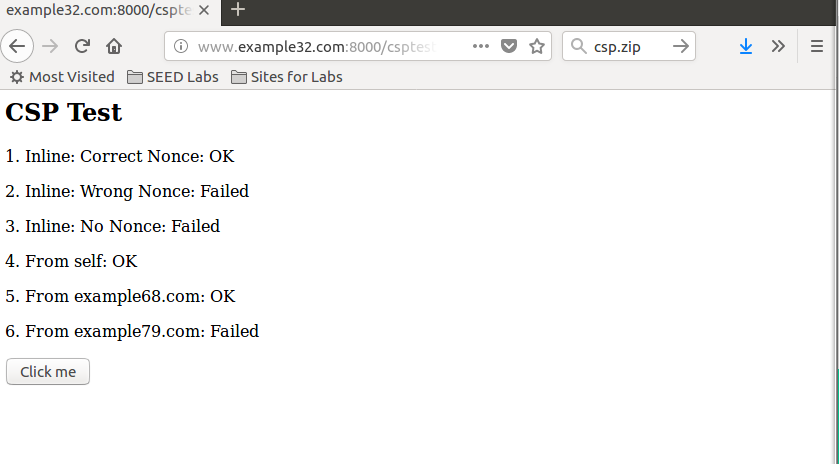
XSS漏洞的根本问题时HTML允许将JS代码与数据混合。要解决这个问题，我们需要将代码与数据分开。

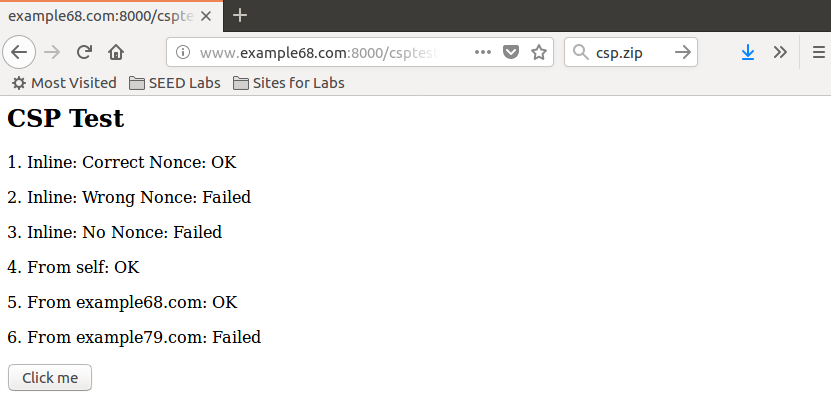
首先需要下载并解压csp.zip文件，修改其中的etc/hosts文件，增加需要实验的三个URL。

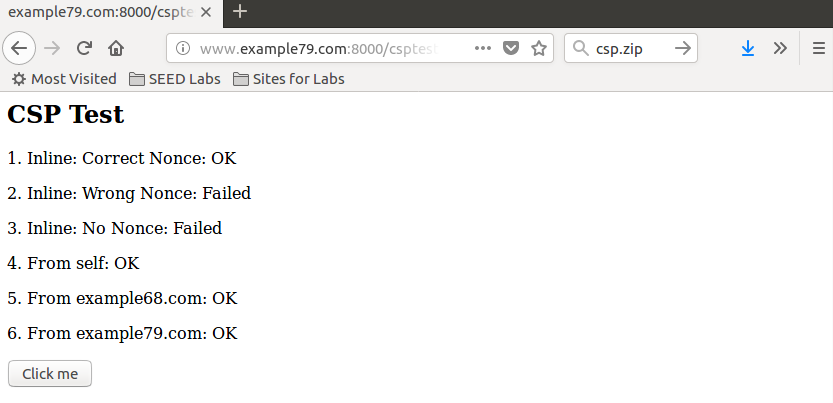




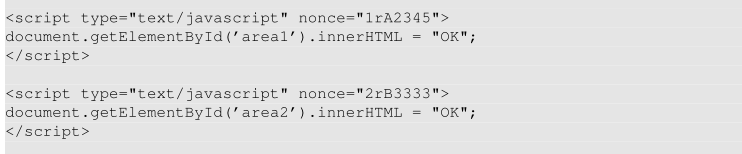
在浏览器中访问[www.example32.com](http://www.example32.com)等URL，运行结果如下：







通过对比可以发现，这几个文件的第二项都显示为failed，分析源代码发现是少添加了一个nounce



修改http\_server.py文件，添加相应的nounce-2rB3333，再次运行，发现成功使1，2，4，5，6显示OK。

