Web-Application Security: XSS and XSRF

Week of January 24, 2017

Goals:

- Gain familiarity with basic web application security.
- Implement successful XSS and XSRF attacks on the Mutillidae and DVWA web applications.
- Learn the basics of Burp Suite.

Required:

- Take the relevant XSS and XSRF lessons on hacksplaining.com.
- Read the 2013 OWASP Top 10 List http://bit.ly/2jOs4ba.
- Install Burp Suite Free Edition from https://portswigger.net/burp/freedownload (if necessary).
- If you are using Firefox or Chrome, install the FoxyProxy Basic plugin for convenience.

Approach:

- We will be attacking the web applications included with the Metasploitable 2 virtual machine, namely Mutillidae and DVWA. If you wish to follow this lab in the future using a fresh copy of Metasploitable 2, there is an error in one of Mutillidae's config files:

```
root@metasploitable:~# cat /var/www/mutillidae/config.inc
<?php
        /* NOTE: On Samurai, the $dbpass password is "samurai" rather than blank
 */

        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '';
        $dbname = 'metasploit';
?>
```

  In /var/www/mutillidae/config.inc, please change the line that reads:

  `$dbname = 'metasploit';`

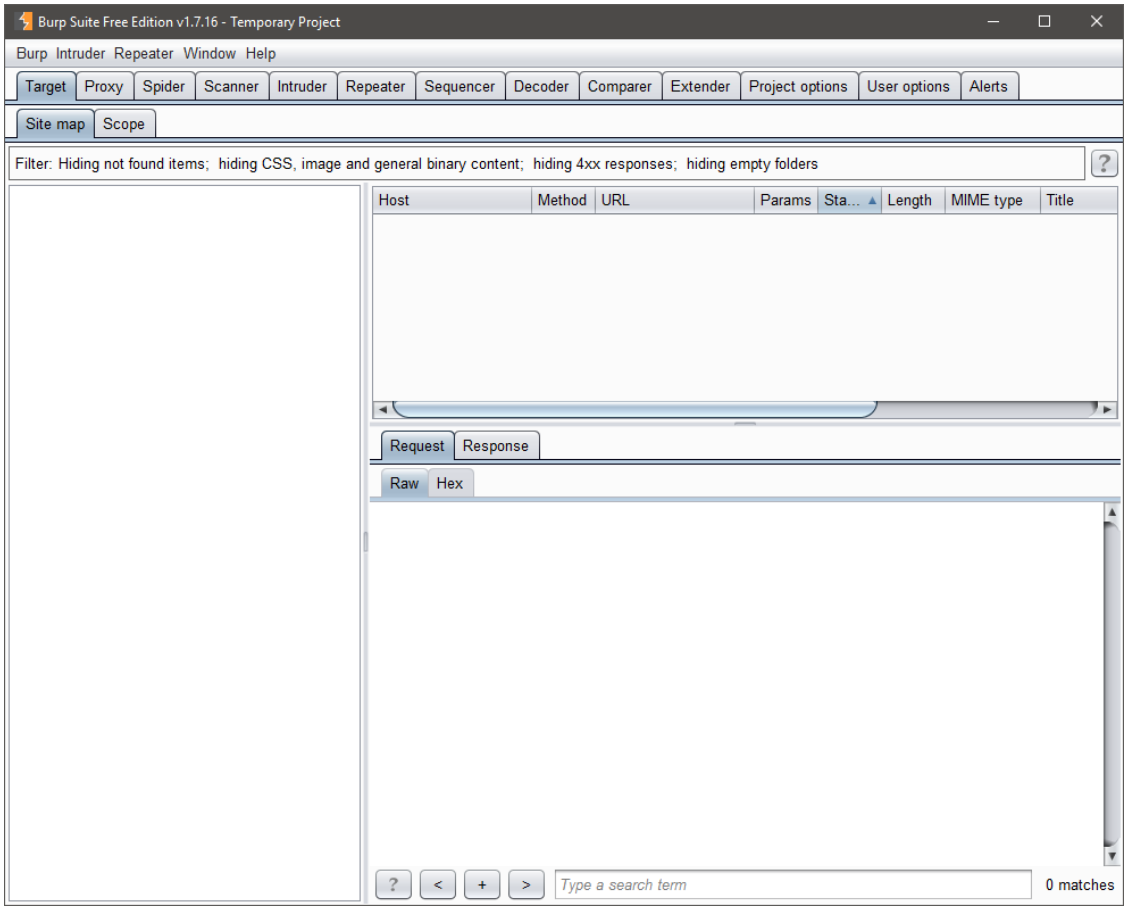
  To:

  `$dbname = 'owasp10';`

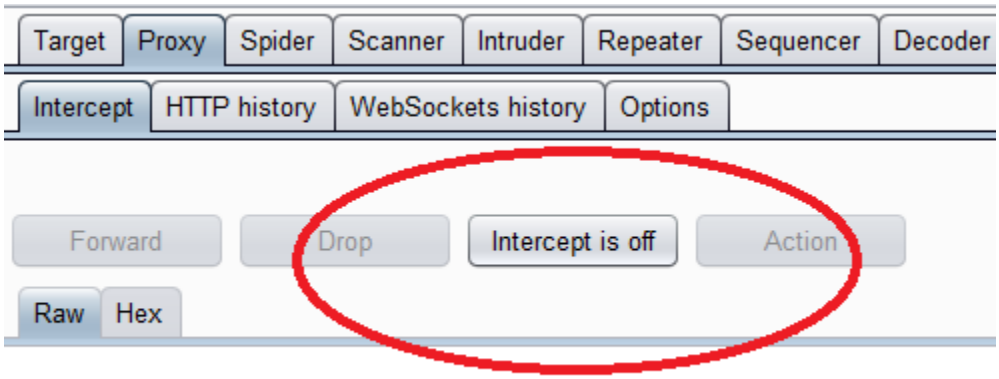
  Note: The ACM VMs have already had this issue fixed.

1. We have several Metasploitable 2 VMs running on the ACM network. Make sure you are connected and when you assemble your group, write the IP Address of the VM that was assigned to you here.

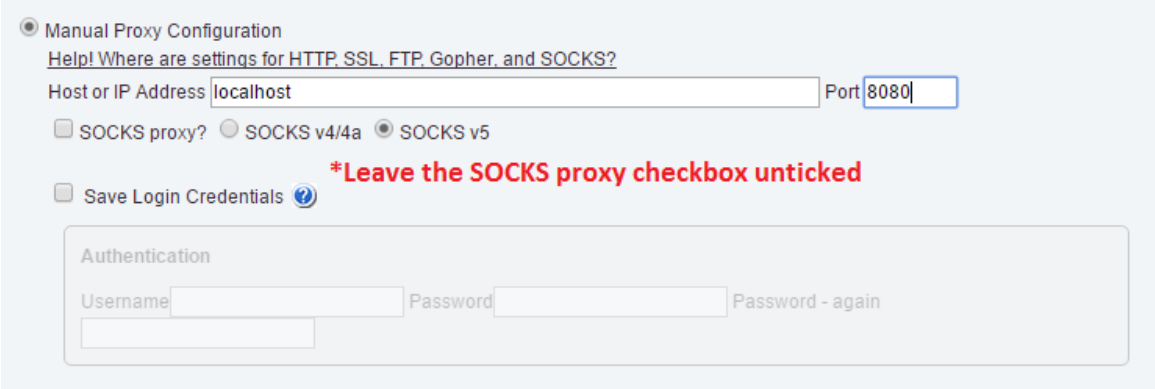
2. Go ahead and fire up Burp Suite. Click through the windows that pop up; all of the default configurations are fine. You'll be brought to a window that looks like this:

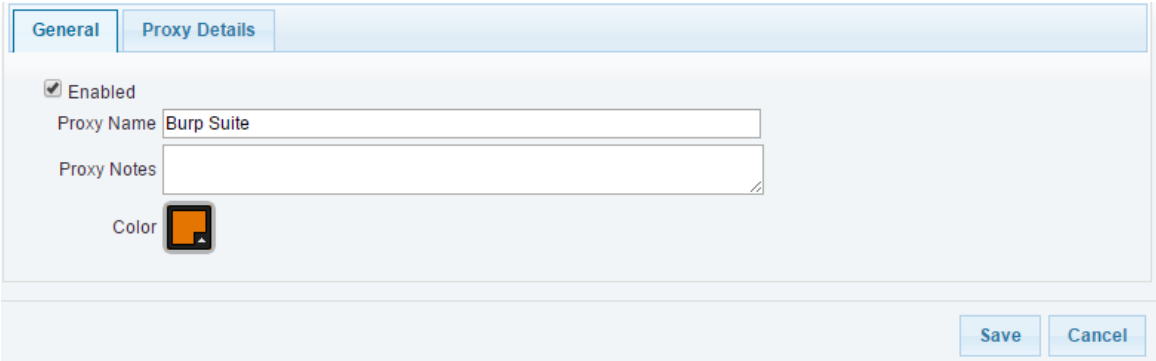


3. Click on the "Proxy" tab near the top and toggle intercept to off.



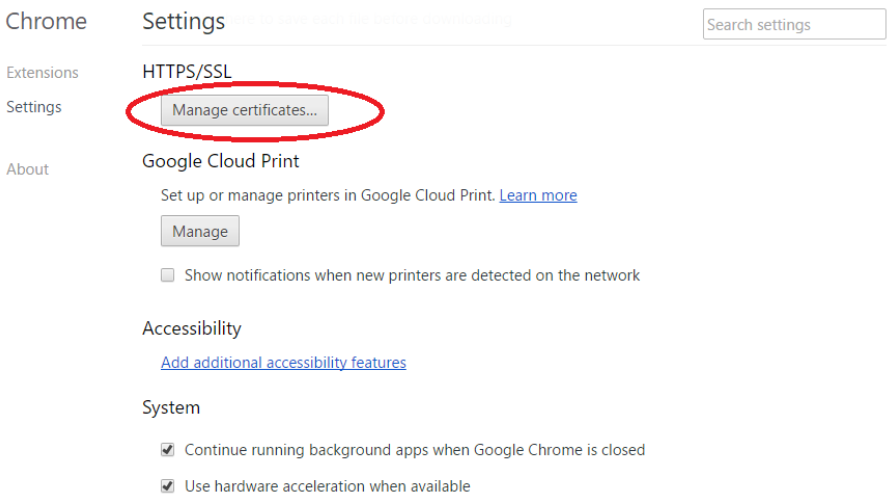
4. By default, Burp Suite binds to localhost on port 8080. We will leave it as is for this lab. However, we will need to add Burp Suite as a proxy for FoxyProxy in our browser and install its certificate in the root store. Go ahead and bring up the options menu for FoxyProxy and add Burp Suite as shown below:

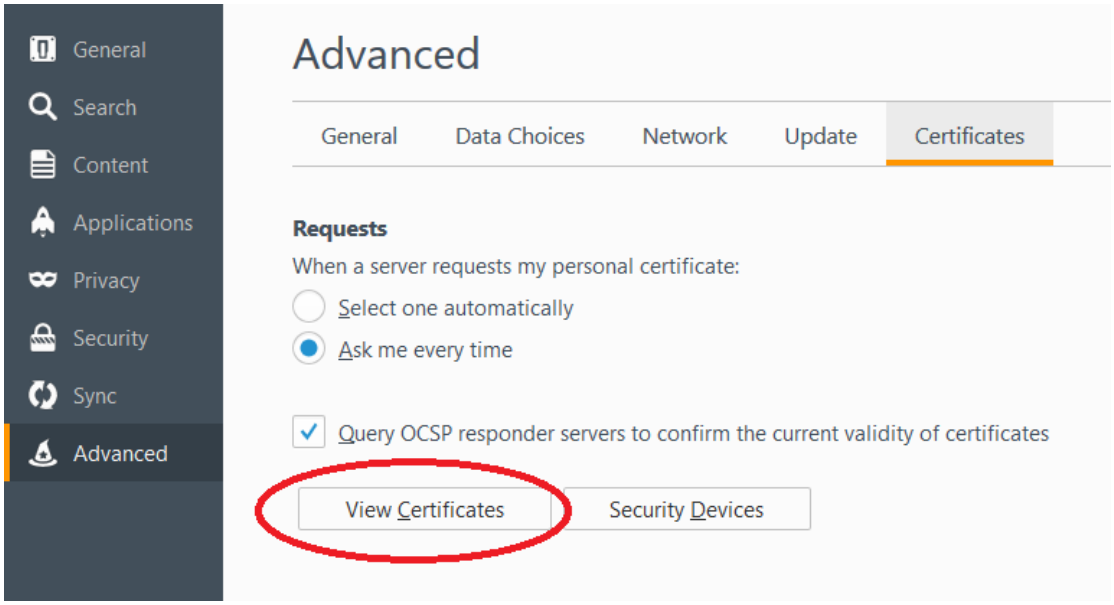




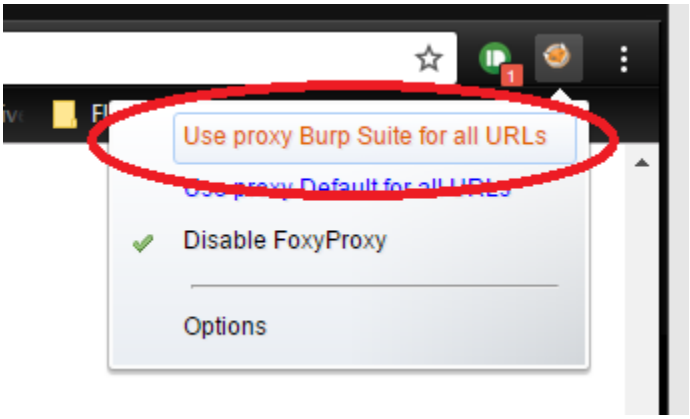
5. Export the Burp Suite certificate by clicking on the Proxy->Options tab and clicking Import/Export CA certificate. Select the first option to export the cert in DER format and save it as burp.der on your Desktop. You can also visit http://localhost:8080/cert to achieve the same effect.

6. Follow your browser's instructions for importing a certificate:





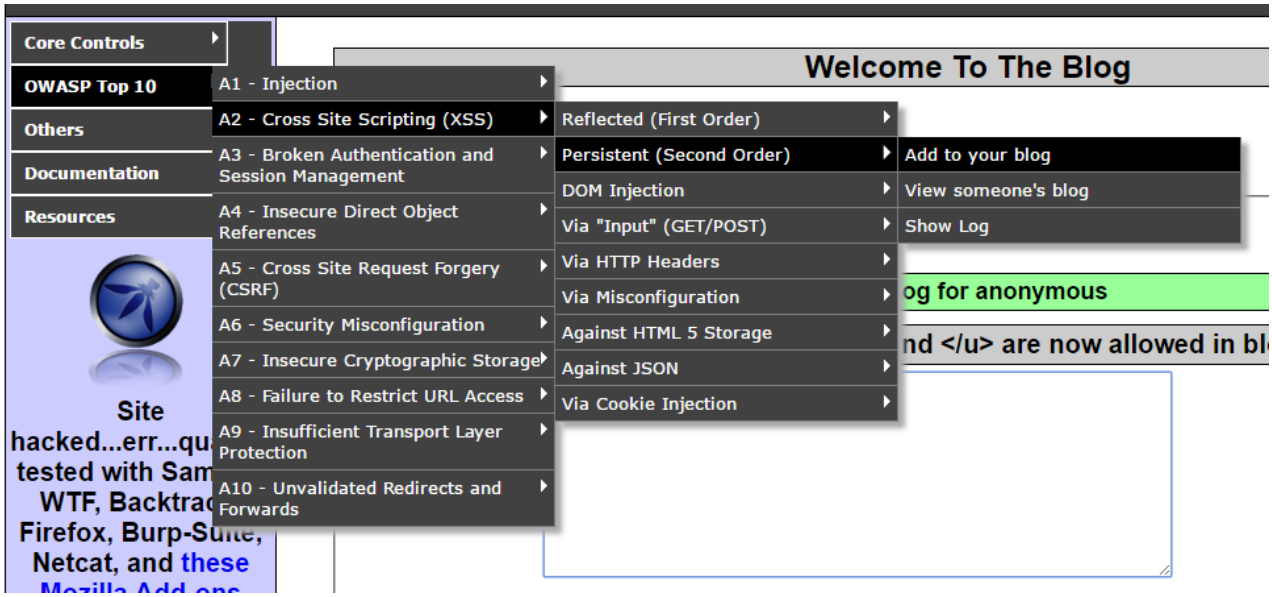
7. Now we're ready to begin actual testing! Make sure to set Burp Suite as your active proxy in FoxyProxy:



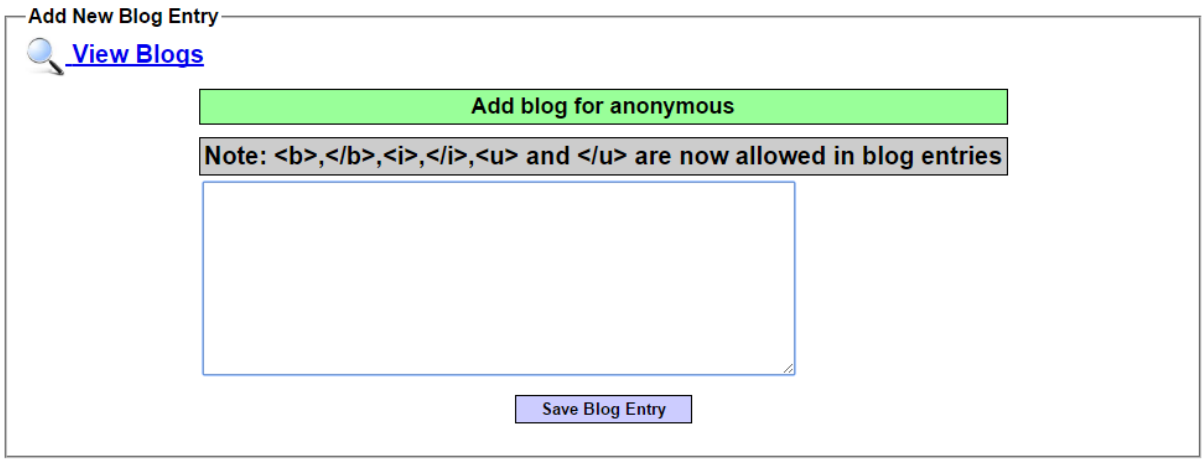
8. We will first begin with Mutillidae. Navigate to the IP address that was assigned to you in part 1. You will be presented with a page that looks like this:

Click on Mutillidae to be presented with the main menu.

9. **Stored XSS (second order XSS)** – Let's start with the most damaging form of XSS. Stored XSS creates a persistent threat that will attack anyone that happens to visit the vulnerable page. First, navigate to the blog page:



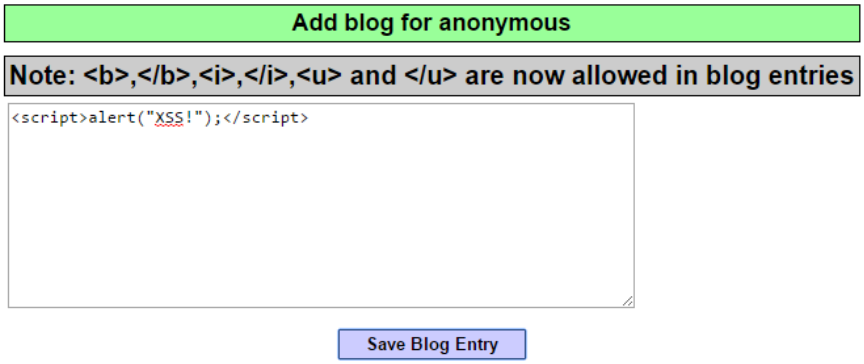
10. Go ahead and add something to your blog. It can be whatever you like. Notice that when you submit it, the results are immediately displayed back to you in the page:



View Blogs

2 Current Blog Entries			
	Name	Date	Comment
1	anonymous	2017-01-22 21:52:18	It can be anything you like.
2	anonymous	2009-03-01 22:27:11	An anonymous blog? Huh?

11. If the input is not properly escaped, we could inject malicious JavaScript code into the victim's browser. Let's try the canonical example. Enter `<script>alert("XSS!");</script>` into the blog and submit:



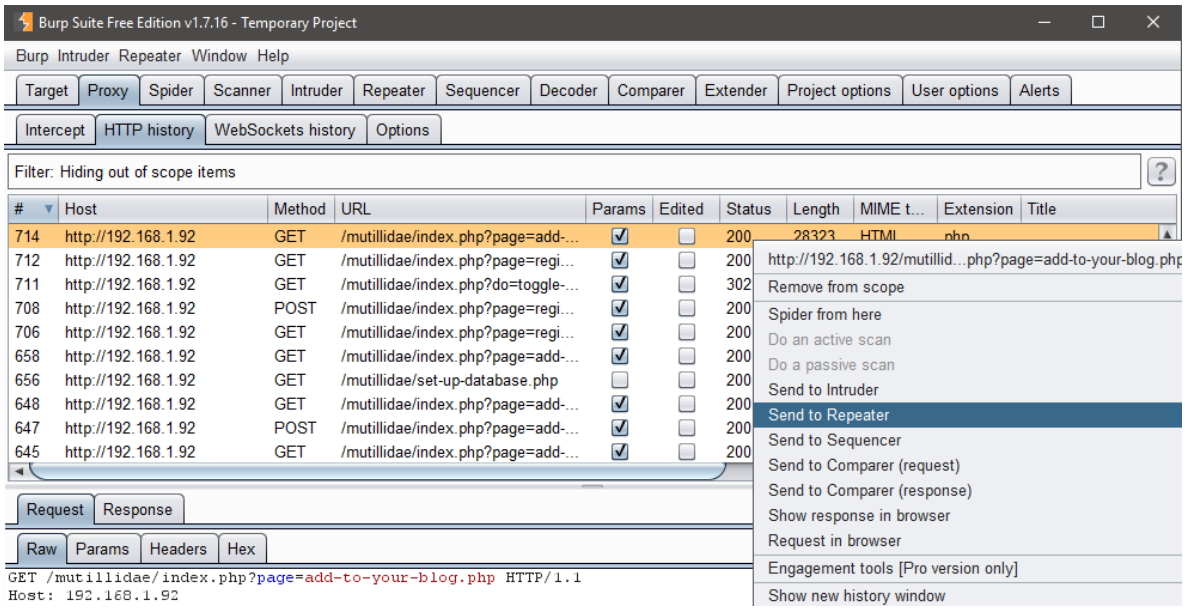
12. After you submit and reload the page, an odd popup appears:

13. Not surprisingly, Mutillidae seems to be vulnerable to Stored XSS. This example is relatively benign, but an attacker could steal all of your cookies associated with that domain or redirect you towards a malicious site:

I didn't want to get any work done anyways…

14. Go ahead and try to exploit the other XSS vulnerabilities on the site. If you get stuck, there is a button located at the top to toggle hints:

15. After you have played around a little bit, you may have noticed that the Proxy->HTTP history tab in Burp has been monitoring all the traffic between you and the server. You can view the request or response by clicking on the appropriate tab, and even selectively view header info or parameters. Go ahead and right click one of these requests and click on “Send to repeater” to bring us to the next tool:

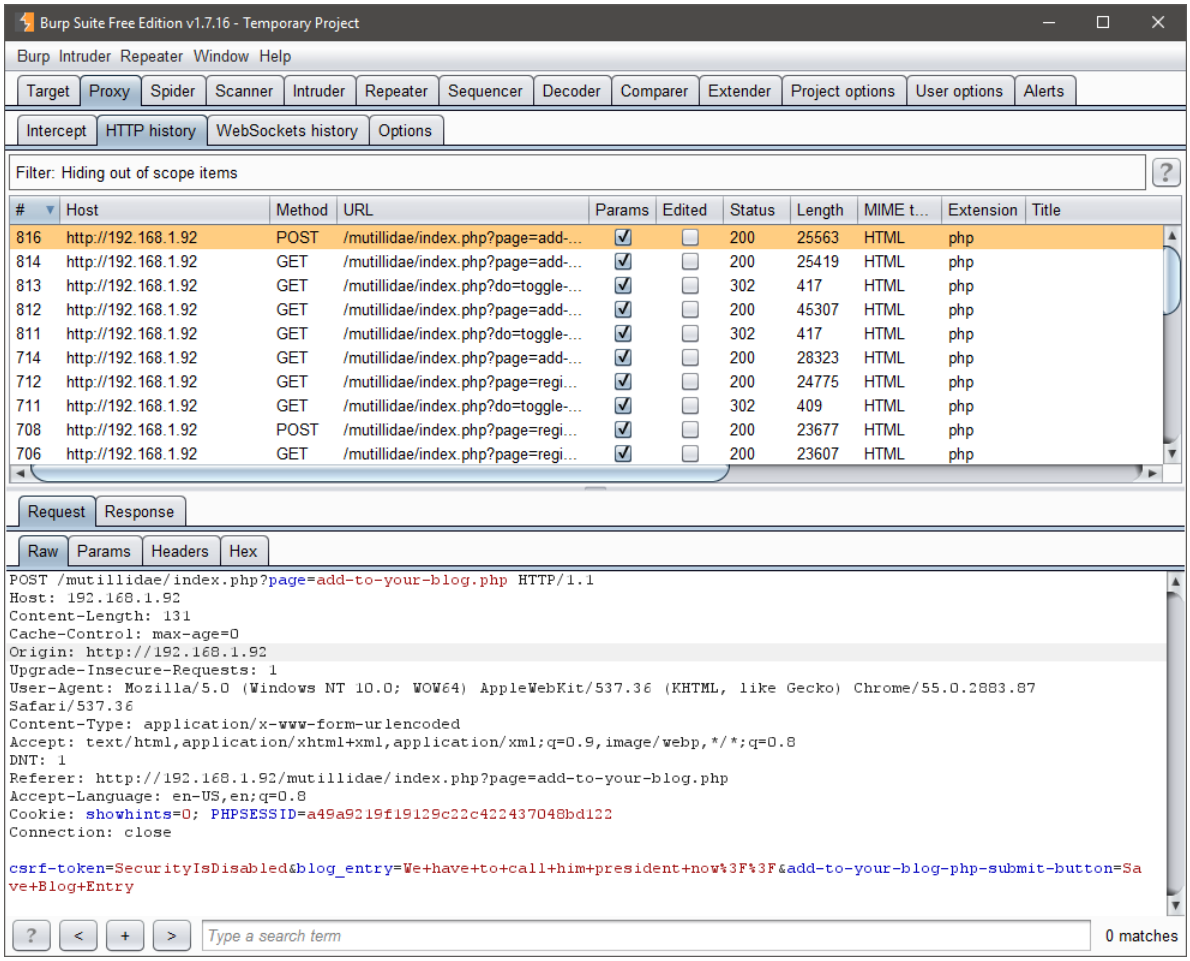


16. Repeater allows us to alter the contents of a specific request, resend it and look at the response. You can either edit the request in its raw format or click on one of the other tabs to have Burp help you by encoding your input. Click “Go” near the top to send the altered request, and explore the response received.

17. You may have noticed earlier that one of the pages vulnerable to Stored XSS was the log page. Navigate to that page (click the “View Log” button at the top) and examine the page. Examine the page for any input that we might have control over. Once you’ve identified the vulnerability, try exploiting it like we did with the blog:

    ***HINT: You will need to use repeater here.**

18. **XSRF** – Coming in at number 8 in OWASPS top 10, XSRF allows an attacker to make illegitimate requests on the victim's behalf. It turns out that the page we visited earlier, the blog, is also vulnerable to XSRF. Go ahead and visit the Proxy->HTTP history tab and look at the request sent when you post to your blog:



19. There are two things that I want to point out: notice that our message was passed as the parameter named "blog_entry" and that there is an additional parameter labeled "csrf-token". We will address the csrf token later. For now, let's create a malicious website to forge a blog post in the victim's name. To do this, we will use our personal webpages on the UIC ACM's website. To visit yours, point your browser at **acm.cs.uic.edu/~<username>** (without the brackets). Simply upload your files to your home directory under **public_html**. To make it easier for you guys, I've created a template under my webpage located at https://acm.cs.uic.edu/~mbaccia. Just visit my page and look at the response in Burp.

20. If everything was done correctly, when you visit your webpage, you should be redirected back to Mutillidae with the message posted:

View Blogs

2 Current Blog Entries			
	Name	Date	Comment
1	anonymous	2017-01-22 23:44:56	Oh no! XSRF!
2	anonymous	2009-03-01 22:27:11	An anonymous blog? Huh?

21. **Finish** – Try the same exercises again only this time, increase the security level by pressing the button at the top. How does Mutillidae try to protect itself?