Activity 4 Guide

Ryan Coon

CST-407

Professor Ty Howard

January 14, 2024

Step By Step Screenshots:

A screenshot of a computer

Description automatically generated

Here you can see the ping results of the two newly created hosts.

A screenshot of a computer

Description automatically generated

Here is a shot of the victimsite successfully running.

A screen shot of a computer

Description automatically generatedHere is a shot of the Hackersite successfully running.

A screenshot of a computer screen

Description automatically generated

Here is the victimsite when comment added with JavaScript injected.

A screen shot of a computer program

Description automatically generated

When inspecting page elements, you can see the keylogger script that was injected in with the comment.

A screenshot of a computer

Description automatically generated

Here we submitted a new post, now we will check the log file to see if the comment has been logged.

A screenshot of a computer

Description automatically generated

Here is the keylog file that has the comment we just submitted via the webpage.

Cross Site Request Forgery:

A screenshot of a computer

Description automatically generated

Here you can see that we entered a post on the hackersite and not victimsite.

A screenshot of a computer

Description automatically generated

Upon submission you can see that it takes us straight to the victimsite with the post processed.

Cross Site Forgery Token:

A screenshot of a computer

Description automatically generated

Here is an attempt to post from another site that was intercepted.

Cookie Stealing:

A screenshot of a computer

Description automatically generated

Here you can see that we are successfully logged in as victor.

A screenshot of a computer

Description automatically generated

We logged in as john and tried to post. Was not allowed.

Steal the Session Id:

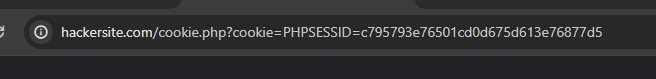


Here we are adding a post with the script to intercept the users cookie.

A screen shot of a computer

Description automatically generated

Here is the submitted post. You can see the script inserted and running on the right hand console.



Here you can see the cookie or session id in the url.

A computer screen shot of a black screen

Description automatically generated

Here is that same Session id in the log.

Find the Session ID Value:

A screenshot of a computer

Description automatically generated

Here is the session id found through inspect in the coogle chrome dev tools.

Use the Stolen Session ID:

A screenshot of a computer

Description automatically generated

Here I used Chrome as the initial browser, then Edge to plug in the stolen session ID.

A screenshot of a computer

Description automatically generated

Made a post without logging in.

Solutions:

A screenshot of a computer

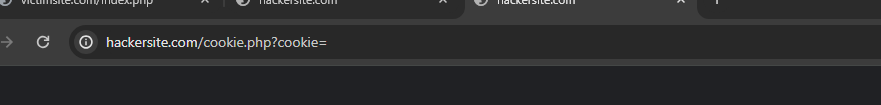
Description automatically generated

Here you can see that we successfully have a set login token now.

A screenshot of a computer

Description automatically generated

Here we did steal victors login id, but was blocked from posting because the tokens did not match.



Here you can see that we were blocked from getting the session ID after setting it to http only.

A screenshot of a computer

Description automatically generated

With the addition to the code on the index.php, we were able to not allow the JavaScript to run, but instead posting it to read.

Summary Questions:

1.Explain the meaning of CSRF and how it was demonstrated in this exercise. How did we correct this vulnerability?

Cross-Site Request Forgery (CSRF) is an attack that forces authenticated users to submit a request to a web application against which they are currently authenticated. This attack occurs when a malicious website tricks a user's browser into making a request to another website on which the user is authenticated.

We demonstrated this by matching the form on the victim site almost identically, and then using that on the hackersite. We then entered the information on the form and have it submit to the users victimsite.com/index.php.

We initially fixed this by requiring the victimsite to use session id and not allowing anything to post without matching session id’s.

2.Explain the process of using JavaScript code to attack an unsuspecting user of an insecure web application. How did we correct this vulnerability?

This is also known as a JavaScript injection attack. In this the attacker uses vulnerabilities to inject malicious code into the client side JavaScript directly. This is done by injecting the code into a comment or input form field.

We corrected this by sanitizing the form output using PHP’s built in commands to remove unwanted or dangerous code from being able to process through form inputs.

References:

【Javascript Injection】Definition, Examples, and Prevention. (2022, October 4). https://crashtest-security.com/js-injection-attack/

Cybersecurity Series: What is cross-site request forgery (CSRF) and how to defend against it. (n.d.). TeamPassword. Retrieved January 15, 2024, from https://teampassword.com/blog/cybersecurity-series-what-is-cross-site-request-forgery-csrf-and-how-to-defend-against-it?utm\_campaign=XSS%20Performance%20Max%202023&utm\_medium=cpc&utm\_source=google&utm\_term=&gad\_source=1&gclid=EAIaIQobChMIztOXpL7egwMVaZpQBh0l4gfWEAAYAiAAEgJSYvD\_BwE