Homework 5 – Content Security Policy

# Lab Information

## Due Date:

Homework 5 Dropbox Deadline

## Objectives/Goal:

In this homework we will be implementing one of the newest client side security controls –content security policy. You will find why many major organizations tend not to employ the control on existing pages. All work will be done using provided source code.

## Deliverables:

* The updated source code
* Completed signoff sheet.
* Table of supported Referrer-Policy options

## Table of Contents:

[Lab Information 1](#_Toc470029507)

[Activity 1: Setup Armbook! 2](#_Toc470029508)

[Activity 2: Feel for your developer 2](#_Toc470029509)

[Activity 3: Setup Reporting 2](#_Toc470029510)

[Activity 4: Referrer-Policy HSTS 3](#_Toc470029511)

[Signoffs 4](#_Toc470029512)

# Activity 1: Setup Armbook!

You’ve done it before now you’ll do it again, but with a twist. One of the most important aspects we’ve talked about is the use of HTTPS (TLS) when setting up a webpage. Setup your webserver to serve Armbook over HTTP/2

## Step 1: Get Armbook working!!!!

Alright this step should be easy. Setup a webserver with PHP and database. Use the source provided to get the application up and running.

## Step 2: Add a pinch of TLS

Now that you have Armbook working properly it’s time to add TLS. Because you’re hosting this locally you won’t be able to use a public Certificate Authority (CA) like let’s encrypt. Instead create your own Certificate Authority and roll your own certificate for Armbook. Make sure to also add your CA to the list of trusted root CAs, as you can’t get this signoff if your browser is throwing a warning.

## Step 3: oooohhhh shiny!

We’ve learned in class that HTTP/2 will not work on modern browsers without HTTPS support -fortunately, we’ve just added HTTPS support. Configure your webserver to serve Armbook of HTTP/2!

# Activity 2: Feel for your developer

## Often penetration testers will make blanket suggestions or offer broad findings such as ‘Missing CSP Headers’. Your goal for this activity is to implement these controls on a small application – Armbook. The source code for this application has been provided as part of the dropbox.

## Step 1: Help a script out

CSP has many controls, some of them are specific to scripts. You must modify the Armbook application to properly/safely support the script-src. You may not use unsafe-inline or unsafe-eval while achieving this goal.

## Step 2: Moving onwards

Now that you’ve taken care of the bulk of the problems add support for font-src, child-src, connect-src, img-src, media-src, font-src, worker-src, object-src, and style-src. If desired you may default-src for directives that aren’t required.

## Step 3: Other Directives

Now that your application is really cooking with fire, implement form-action, frame-ancestors, and block-all-mixed-content.

# Activity 3: Setup Reporting

CSPs biggest issue is that it tends to be very annoying to tell if any of your changes falsely triggered, particularly on dynamic sites. In order to solve this CSP added a reporting feature.

## Step 1: Setup Reporting

Now that you’ve setup CSP it’s time to configure it to detect issues in the future. CSP supports reporting to a third party site in case of a violation. This can be specified using report-to (or report-uri prior to version 3). Use both of these to configure the where your violation reports will go to.

## Step 2: Create a listener

Now that you’ve got the configuration down create a page at the location you specified that listens for the policy violation reports and adds it’s to a MySQL database. Demonstrate this to your instructor to get the signoff.

# Activity 4: Referrer-Policy HSTS

Now that we’ve got CSP working it’s time to add a few more headers to get the most out of our modern browser.

## Step 1: Referrer-Policy

CSP used to have a policy called referrer but that has been obsoleted in favor of the Referrer-Policy header. Test the new header and see if it functions correctly on both IE/Edge and Chrome. Note that the Referrer-Policy has many different options, you should be able to discuss what each of the do and have a table explaining which are supported on the respective browsers.

Referrer-Policy Headers: "no-referrer" / "no-referrer-when-downgrade" / "strict-origin" / "strict-origin-when-cross-origin" / "same-origin" / "origin" / "origin-when-cross-origin" / "unsafe-url"

## Step 2: One step closer to the edge

The last step of this lab is to implement HSTS. This should be pretty simple, configure your webserver to use HSTS and then demonstrate it to your instructor by trying to intercept an HTTPS request using Burp.

# Signoffs

## Activity 1.3 – Demonstrate that Armbook is working over HTTPS with HTTP/2 enabled

## Activity 2.1 – Show the instructor your CSP headers and explain why you chose the options you did. Subsequently walk the instructor through the site and ensure that no violations appear in the console

## Activity 2.3 – Show the instructor your CSP headers and explain why you chose the options you did. Subsequently walk the instructor through the site and ensure that no violations appear in the console

## Activity 3.2 – Demonstrate that you are able to save CSP violations to a database

## Activity 4.1 – Referrer-Policy header is implemented and options can be explained.

## Activity 4.2 – HSTS is enabled and can be demonstrated as functional