LAB EXERCISE

7

COMP3134

Introduction to Cyber Security

Week: 10

Objective(s):
Attacking the web applications and applying mitigation techniques

Learning Outcome(s):
Implement various tactics to attack a network or application
Critique and execute mitigation techniques

Table of Contents

Contents

Summary

Goal: Attacking the web applications and applying mitigation techniques

In Effort To: Implement various tactics to attack a network or application, as well as to critique and execute mitigation techniques

A. Clone GitHub Repo

Clone course GitHub repo to any location on your local machine
Navigate to the location above and create a folder named **wk10**
Use this local folder created above to create all the files necessary for this Lab Exercise

B. Cookies

What are Cookies & What is Their Danger?

A small text file (up to 4KB) created by a website that is stored in the user's computer either temporarily for that session only or permanently on the hard disk (persistent cookie). Cookies provide a way for the website to recognize you and keep track of your preferences. Cookies are dangerous because it stores personal information about the user which could be used to take their identity or spoof him or her.

How To View Cookie Files on a Machine

On a PC, follow the instructions here:
https://www.thewindowsclub.com/cookies-folder-location-windows

On a Mac, follow the instructions here:
https://www.fireebok.com/resource/how-to-open-a-cookie-file-on-mac-computer.html

On Linux machines, the cookies are stores in the home directory as hidden files (files starting with the dot

How To View Cookies From a Browser

The following link shows how to view cookies from the 5 most popular browsers:
https://www.wikihow.com/View-Cookies

Using a Browser Console

Navigate to the console of a browser and type in the code below to view a string of your cookie information

```
document.cookie
```

C. Sending Information to Application

Most interactive sites need to take input from the user. This is done via a Form and/or a query string

Production Site Example with Mitigation

Navigate to the following url: https://www.dictionary.com/browse/cybersecurity

Create a text file named **query_string_observations.txt**.
Write your observations for the scenarios below.

1) Navigate to the form and type a valid English word. Write your observations.
2) Navigate to the URL and type an invalid English word after the `/browse/` section of the URL. Write your observations
3) Navigate to the form and type the following text: <script>alert(%27hello%27)</script> Write your observations.
4) Navigate to the URL and type the following text: <script>alert(%27hello%27)</script> after the `term=` section of the URL. Write your observations.

Development Site Example without Mitigation

Navigate to the web root of your server and create a file named **sending_info_to_app_1.php**
Add the following code:

```
<form method="get">

  <input name="q" placeholder="Enter Text">

  <br/>

  <input type="submit" value="Go">

</form>
```

Add the code necessary to output without any filter any content that is passed to the script via the form or query string.

Open the URL: {Your droplet IP}/sending_info_to_app_1.php
Execute the 4 scenarios earlier in this section and take a screenshot of the result each time in the format of **sending_info_to_app_N.png** where N represents a scenario

Development Site Example without Mitigation

Create a PHP script named **sending_info_to_app_2.php**. Copy the contents of **sending_info_to_app_1.php** as your starting point.
Apply the necessary changes by either stripping tags or displaying the html entity value of any user input.
What is the difference between the two solutions? Write the answer in a file named **strip_tags_vs_html_entities.txt**

Open the URL: {Your droplet IP}/sending_info_to_app_2.php
Execute the 4 scenarios earlier in this section and take a screenshot of the result each time in the format of **sending_info_to_app_mit_N.png** where N represents a scenario

D. HTTP Requests Using CURL

To execute an HTTP request, use the built-in tool named curl

Manual & Useful Examples
A manual of curl can be found here: https://helpmanual.io/help/curl/
Useful CURL examples can be found here:
https://flaviocopes.com/http-curl/#perform-an-http-get-request

Create a file named **curl_commands.txt** and state the commands needed to

1) Send a GET request to dictionary.com to search for the word “curl”
2) Send a GET request to dictionary.com to search for the word “curlxyz”
3) Send a GET request to {Your droplet IP}/sending_info_to_app_1.php to submit the query “helloworld”
4) Send a GET request to {Your droplet IP}/sending_info_to_app_1.php to submit the query “<script>document.write(‘overwrite_everything’)</script>”
5) Send a GET request to {Your droplet IP}/sending_info_to_app_2.php to submit the query “<script>document.write(‘overwrite_everything’)</script>”

E. Commit and Upload Changes to GitHub repo

Commit the changes to your repo by:

1. Opening a GitBash window and ensure that it is connected to your local machine
2. Navigate to local repository directory location
3. Add all the files completed in this Lab Exercise
4. Commit the changes
5. Push the changes to your GitHub course repo

Please refer to the instructions in the last section of Lab Exercise 1