**Questions**

**HTTP Requests and Responses**

Answer the following questions about the HTTP request and response process.

1. What type of architecture does the HTTP request and response process occur in?

* HTTP request and response process occurs in **Client/Server based architecture**.

1. What are the different parts of an HTTP request?

* HTTP request consists of following three parts.

**Request Line, Request Header, Request body**

1. Which part of an HTTP request is optional?

* **Request body** is the optional part of an HTTP request.

1. What are the three parts of an HTTP response?

* The three parts of an HTTP response are :

**Status Line, Headers, Body**

1. Which number class of status codes represents errors?

* **400** status codes represent errors.

1. What are the two most common request methods that a security professional will encounter?

* **GET and POST** are two most common request methods that a security professional will encounter.

1. Which type of HTTP request method is used for sending data?

* **POST** request method is used for sending data for HTTP request.

1. Which part of an HTTP request contains the data being sent to the server?

* The **Request Body** part of an HTTP request contains the data being sent to the server.

1. In which part of an HTTP response does the browser receive the web code to generate and style a web page?

* The **Response Body** part of an HTTP response does the browser receive the web code to generate and style a web page.

**Using curl**

Answer the following questions about curl:

1. What are the advantages of using curl over the browser?

* The advantages of using curl over the browser are :
  + It transfers data to or from the server using different protocols.
  + It gives access to response headers.
  + With curl command HTTP headers, cookies and authentication information can be passed in as parameters.

1. Which curl option is used to change the request method?

* The curl option that is used to change the request method is **–X.**

1. Which curl option is used to set request headers?

* The curl option that is used to set request headers is **–H.**

1. Which curl option is used to view the response header?

* The curl option is used to view the response header is **-I**

1. Which request method might an attacker use to figure out which HTTP requests an HTTP server will accept?

* The **OPTIONS** request method that an attacker might use to figure out which HTTP requests an HTTP server will accept.

**Sessions and Cookies**

Recall that HTTP servers need to be able to recognize clients from one another. They do this through sessions and cookies.

Answer the following questions about sessions and cookies:

1. Which response header sends a cookie to the client?

HTTP/1.1 200 OK

Content-type: text/html

Set-Cookie: cart=Bob

* The response header that sends a cookie to the client is **Set-Cookie.**

1. Which request header will continue the client's session?

GET /cart HTTP/1.1

Host: www.example.org

Cookie: cart=Bob

* The request header that will continue the client's session is **Cookie**

**Example HTTP Requests and Responses**

Look through the following example HTTP request and response and answer the following questions:

**HTTP Request**

POST /login.php HTTP/1.1

Host: example.com

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: 34

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Mobile Safari/537.36

username=Barbara&password=password

1. What is the request method?

POST Request.

1. Which header expresses the client's preference for an encrypted response?

Upgrade-Insecure-Requests: 1

1. Does the request have a user session associated with it?

No, since there is no cookie or session id.

1. What kind of data is being sent from this request body?

The user login credentials are being sent from this request body.

**HTTP Response**

HTTP/1.1 200 OK

Date: Mon, 16 Mar 2020 17:05:43 GMT

Last-Modified: Sat, 01 Feb 2020 00:00:00 GMT

Content-Encoding: gzip

Expires: Fri, 01 May 2020 00:00:00 GMT

Server: Apache

Set-Cookie: SessionID=5

Content-Type: text/html; charset=UTF-8

Strict-Transport-Security: max-age=31536000; includeSubDomains

X-Content-Type: NoSniff

X-Frame-Options: DENY

X-XSS-Protection: 1; mode=block

[page content]

1. What is the response status code?

* **200 is the response status code.**

1. What web server is handling this HTTP response?

* **Server: Apache**

1. Does this response have a user session associated to it?

* Yes**, Set-Cookie: SessionID=5**

1. What kind of content is likely to be in the [page content] response body?

* **The code of website,** **Content-Type: text/html; charset=UTF-8,**

1. If your class covered security headers, what security request headers have been included?

* **Upgrade-Insecure-Requests**. This request header tells the HTTP server that it prefers to communicate through HTTPS over HTTP.
* The **Strict-Transport-Security** header tells the HTTP client that it should use HTTPS over HTTP.
* The **Content Security Policy (CSP)** header recommends way to protect websites and applications against XSS attacks.
* **X-frame** option provide protection against cross-site scripting attacks involving HTML iframes.

**Monoliths and Microservices**

Answer the following questions about monoliths and microservices.

1. What are the individual components of microservices called?

* The individual components of microservices are called **Services.**

1. What is a service that writes to a database and communicates to other services?

* A service that writes to a database and communicates to other services is an **API.**

1. What type of underlying technology allows for microservices to become scalable and have redundancy?

* The type of underlying technology allows for microservices to become scalable and have redundancy is **Containers.**

**Deploying and Testing a Container Set**

Answer the following questions about multi-container deployment:

1. What tool can be used to deploy multiple containers at once?

* The tool can be used to deploy multiple containers at once is **Docker-Compose.**

1. What kind of file format is required for us to deploy a container set?

* The kind of file format is required for us to deploy a container set is **.yml.**

**Databases**

1. Which type of SQL query would we use to see all of the information within a table called customers?

* **Select \* customers;**

1. Which type of SQL query would we use to enter new data into a table? (You don't need a full query, just the first part of the statement.)

* **Insert i**nto table name field 1, filed2… value1, value2….

1. Why would we never run DELETE FROM <table-name>; by itself?

* We would never run DELETE FROM <table-name>; by itself because this query will delete all the records.

**Bonus Challenge Instructions: The Cookie Jar**

**Step 1: Set Up**

Create two new users: Amanda and Ryan.

1. Navigate to localhost:8080/wp-admin/
2. On the left-hand toolbar, hover over **Users** and click **Add New**.
3. Enter the following information to create the new user named Amanda.
   * Username: Amanda
   * Email: amanda@email.com
4. Skip down to password:
   * Password: password
   * Confirm Password: Check the box to confirm use of weak password.
   * Role: Administrator
5. Create another user named Ryan.
   * Username: Ryan
   * Email: ryan@email.com
6. Skip down to password:
   * Password: 123456
   * Confirm Password: Check the box to confirm use of weak password.
   * Role: Editor
7. Log out and log in with the following credentials:
   * Username: Amanda
   * Password: password

**Step 2: Baselining**

For these "baselining" steps, you'll want to log into two different types of accounts to see how the WordPress site looks at the localhost:8080/wp-admin/users.php page. We want to see how the Users page looks from the perspective of an administrator, vs. a regular user.

1. Using your browser, log into your WordPress site as your sysadmin account and navigate to localhost:8080/wp-admin/users.php, where we previously created the user Ryan. Examine this page briefly. Log out.
2. Using your browser, log into your Ryan account and attempt to navigate to localhost:8080/wp-admin/index.php. Note the wording on your Dashboard.
3. Attempt to navigate to localhost:8080/wp-admin/users.php. Note what you see now.

Log out in the browser.

**Step 3: Using Forms and a Cookie Jar**

Navigate to ~/Documents in a terminal to save your cookies.

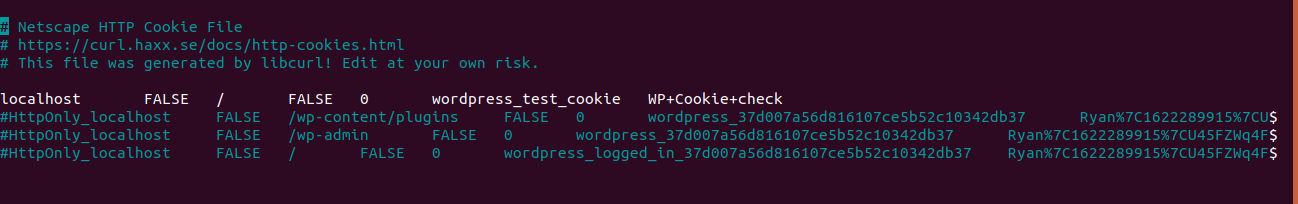
1. Construct a curl request that enters two forms: "log={username}" and "pwd={password}" and goes to http://localhost:8080/wp-login.php. Enter Ryan's credentials where there are placeholders.
   * **Question:** Did you see any obvious confirmation of a login? (Y/N)
     1. ***curl --form "log=Ryan" --form "pwd=123456" http://localhost:8080/wp-login.php --verbose***
     2. ***NO***
2. Construct the same curl request, but this time add the option and path to save your cookie: --cookie-jar ./ryancookies.txt. This option tells curl to save the cookies to the ryancookies.txt text file.

* ***curl --cookie-jar ./ryancookies.txt --form "log=Ryan" --form "pwd=123456" http://localhost:8080/wp-login.php --verbose***

1. Read the contents of the ryancookies.txt file.

**Question:** How many items exist in this file?

* + - ***nano ryancookies.txt***

******

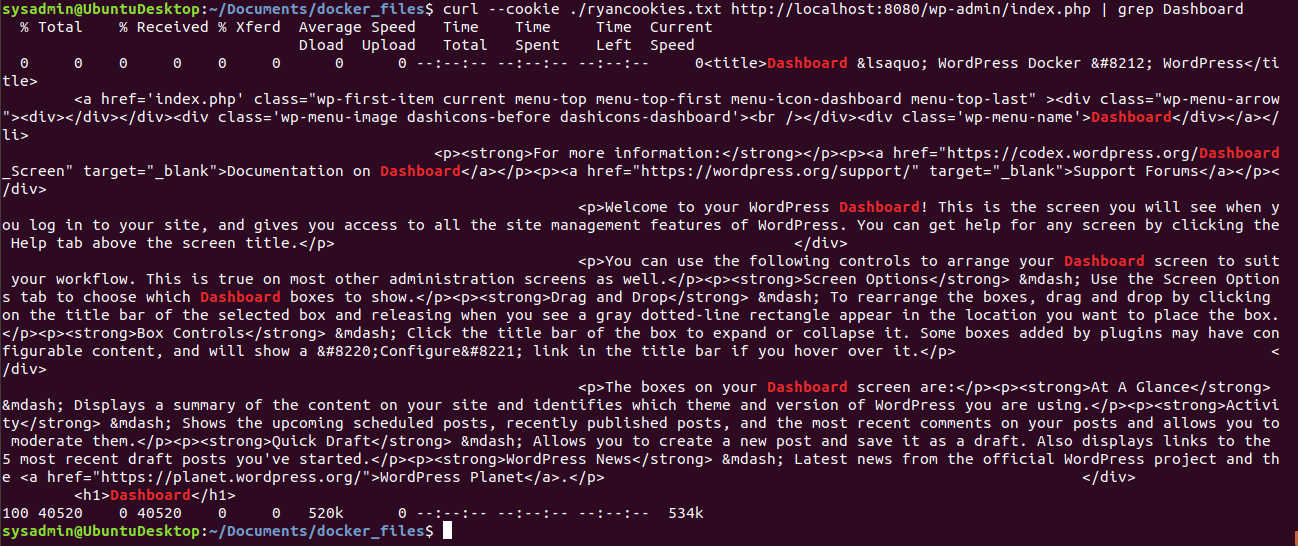
* + ***It has three items.***

Note that each one of these is a cookie that was granted to Ryan after logging in.

**Step 4: Log in Using Cookies**

1. Craft a new curl command that now uses the --cookie option, followed by the path to your cookies file. For the URL, use http://localhost:8080/wp-admin/index.php.
   * **Question:** Is it obvious that we can access the Dashboard? (Y/N)
     + ***curl --cookie ./ryancookies.txt***[***http://localhost:8080/wp-admin/index.php***](http://localhost:8080/wp-admin/index.php)
2. Press the up arrow on your keyboard to run the same command, but this time, pipe | grep Dashboard to the end of your command to return all instances of the word Dashboard on the page.
   * **Question:** Look through the output where Dashboard is highlighted. Does any of the wording on this page seem familiar? (Y/N) If so, you should be successfully logged in to your Editor's dashboard.

***curl --cookie ./ryancookies.txt***[***http://localhost:8080/wp-admin/index.php***](http://localhost:8080/wp-admin/index.php)***| grep Dashboard***

******

* ***Yes***

**Step 5: Test the Users.php Page**

1. Finally, write a curl command using the same --cookie ryancookies.txt option, but attempt to access http://localhost:8080/wp-admin/users.php.
   * **Question:** What happens this time?
     + ***curl --cookie ./ryancookies.txt***[***http://localhost:8080/wp-admin/users.php***](http://localhost:8080/wp-admin/users.php)

