

## Module 12: Web Development Overview and Agenda

Hey everyone, this module is focussed on understanding some of the fundamental tools and protocols associated with web development in relation to cyber security. The homework assignment is merely a question and answer worksheet, so we're going to spice things up a little for the study session.

### **Activity1 : Jeopardy+** (45 mins)

We're going to play a special review game like in our old days as students in the classroom. We'll make two teams, and everyone is a judge. That means, even when it's not your turn, you should still be paying attention because you can point out any error made by the other team, and steal their points. It's not about winning and losing though, we're playing to make sure we all are up to speed on these topics, and help each other out in getting there. Plus it's a good chance to try to have some fun with your co-workers!

*(The winning team doesn't get anything FYI, just bragging rights and the absolute joy and satisfaction of helping out your co-workers and students, isn't that wonderful enough? Who needs prizes? ^\_^)*

### **Game Rules:**

1. At least 2 teams, or free for all, depending on attendance
2. 30 seconds to pick a question (or loose your turn)
3. 1 minute to consult with your team, and answer a question (or loose your turn)
4. Players are also the judges, so listen to the answer given, and scrutinize it for accuracy
5. Answers must be both *necessary and sufficient* to count for points
6. The opposing team can steal points from the other team by pointing out any errors in the answer given, and correcting it.

**\*You cannot use anything to help your team answer, no internet, no cheat sheets, treat it like a test\***

You won't be able to access this link until after the study session to prevent anyone from getting an unfair advantage. [Jeopardy Game](#)

Remember our main goal is to check our understanding, and have fun! Each Jeopardy question is an opportunity for a deeper group discussion, and we will use this time to clarify the main points with one another.

### **Activity 2 : Bonus Challenge, The Cookie Jar** (45 mins)

The HW assignment for Module 12 ends with a bonus challenge. The goal is to login to a wordpress account just using cookies and the curl command. That said, ideally everyone will setup the wordpress site *before* the study session. If you need help with Docker though, we can help you out during the session! Many students misunderstand docker/containers, so we will make sure everyone is clear on the concept and execution.

Lab Setup Directions can be found here: [Wordpress Docker Setup](#)

## **Bonus Challenge Instructions**

First, using Docker Compose, navigate to the Day 1 WordPress activity directory and bring up the container set:

- `/home/sysadmin/Documents/docker_files`

Using curl, you will do the following for the Ryan user:

- Log into WordPress and save the user's cookies to a cookie jar.
- Test a WordPress page by using a cookie from the cookie jar.
- Pipe the output from the cookie with grep to check for authenticated page access.
- Attempt to access a privileged WordPress admin page.

## **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: Baselineing**

For these "baselineing" 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, log out.

### **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.

```
curl --form "log=Ryan" --form "pwd=123456" http://localhost:8080/wp-login.php
```

- Question: Did you see any obvious confirmation of a login? (Y/N)
  - *There shouldn't be any confirmation.*

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
```

3. Read the contents of the ryancookies.txt file.
  - Question: How many items exist in this file?
    - *There should be four cookies in the file.*

### **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>.

```
curl --cookie ./ryancookies.txt http://localhost:8080/wp-admin/index.php
```

- Question: Is it obvious that we can access the Dashboard? (Y/N)
    - *It shouldn't be obvious.*
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: 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.
      - *There should instances of the word Dashboard, meaning we know we have successfully created an index.php session as Ryan.*

### **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>.
  - *You should get the same message as before, the page is inaccessible.*