**Concept Review**

**Which attacks were easiest to execute? Which were the most difficult?**

* I believe that the moderately difficult ones to find were Cross-Site Scripting and SQL Injection, since I was not sure where to insert the script or the Injection. The ones that I believe to be the easiest to execute were IDOR, Session-Hijacking, and User Enumeration. I used examples from the CTF/Shepherd challenges to help me with IDOR and Session-Hijacking. I was randomly probing the log in field using random characters, as well as the username provided by code path (jmonroe99) as a test example, to see the results of trying to log in. The difficult one that I could not identify, or execute, was the remaining Cross-Site Request Forgery. For CSRF, I was not sure how to forge a form and where to implement the form to get results.

**What is a good rule of thumb which would prevent accidentally username enumeration vulnerabilities like the one created here?**

* Programmers can eliminate what is displayed after a user attempts to log in, or they can add something like “username/password not correct.” This explains to someone who failed to log in that something was wrong when they typed in the username or the password and does not determine which was correct or incorrect.

**Since you should be somewhat familiar with the CMS and how it was coded, can you think of another resource which could be made vulnerable to an Insecure Direct Object Reference? What code could be removed which would expose it? (Hint: It was also the answer to the first bonus objective to the Weekly Assignment for week 3.)**

**Many SQL Injections use OR as part of the injected code. (For example: ' OR 1=1 --'.) Could AND work just as well in place of OR? (For example: ' AND 1=1 --'.) Why or why not?**

* I do not believe that the ‘AND’ operator would be able to work as effectively as the ‘OR’ operator. When using the ‘OR’ operator, it is to set a condition that would result into a true statement. For example, using ‘OR 1=1’ determines that the condition is always true, which could then return information because the condition is true. The ‘AND’ operator is used when two pieces of code are joined to create a longer statement. For example, this code will return “x” AND this code will return “y”, thus both items will return.

**A stored XSS attack requires patience because it could be stored for months before being triggered. Because of this, what important ingredient would an attacker most likely include in a stored XSS attack script?**

* Attackers can mask the script as a link to make the attack more appealing to people. For example, an attacker can leave a comment that talks about some article and include a link that points to the article. But in reality, it is a script that can activate once it is clicked, which it can then perform its intended purpose.

**Imagine that one of your classmates is an authorized admin for the site's CMS and you are not. How would you get them to visit the self-submitting, hidden form page you created in Objective #5 (CSRF)?**

**Compare session hijacking and session fixation. Which attack do you think is easier for an attacker to execute? Why? One of them is much easier to defend against than the other. Which one and why?**

* I believe it would be easier for an attacker to execute Session Hijacking since there are a lot of ways to retrieve a victim’s session ID; one of them being an XSS attack. With Session fixation, the attacker would need to get the victim to use a session that the attacker has set up, possibly by visiting a link, which can take some time. I believe that, while being the easiest to execute, session hijacking can also be easily defended against. Programmers can set up filters to prevent the possibility of an XSS from happening, and navigating through HTTPS enabled websites can also reduce the possible of a session hijacking since HTTPS is more secure than HTTP.