**Concept Review**

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

The attacks that was the easiest to execute was the username enumeration! The attacks that was the most difficult was the Cross-Site Request Forgery!

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

A good rule of thumb which would prevent accidentally username enumeration vulnerabilities like the one above is to create a request for a valid user and also a request for an invalid user and then compare the output for each of these requests to make sure they are the same.

* 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.)

Another resource that could be made vulnerable to an Insecure Direct Object Reference is the url. The code that could be removed to expose it would be the url already in it.

* 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?

No, AND would work differently because you have to be sure you have both tables being implemented in order for AND 1=1 to work.

* 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?

The important ingredient that an attacker would most likely include in a stored attack script would be by crowdsourcing!

* 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)?

I would get them to visit the self-submitting hidden form page I created in object #5 by attempting to email them a link for them to click.

* 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 think session fixation is easier for the attacker to execute because once they have the users session ID they’re able to access anything with the user involved. Session hijacking is easier to defend against because it lets the hackers bypass your verification system!