Lab 6 Test Report | Introduction to Cybersecurity

Blake Raphael, Charlotte Gorgemans, Sanjana Kumpati, Benjamin Kohav,
Liam McChesney, Ian Stewart, Charlie Gau

Link to Website: http://recitation-15-team-05.eastus.cloudapp.azure.com:3000/

All Vulnerabilities Tested for:

1. Ordering/Placing in Cart Negative Quantity of Items **(Client Side Controls)**
   - **Test:** Attempted to edit the number of items to be able to add a negative number. Did this through the clicker as well as editing html code.
   - **Result:** There was a server-side check on this so the text failed.
2. Restrictions on Password Format/Content
   - **Test:** Try different types of passwords of varying lengths and content
   - **Result: Attack Succeeded;** See #1 in found vulnerabilities
3. Entering a single quote (‘) into text inputs **(SQL Injection)**
   - **Test:** Put ‘ into the various text inputs in the register, login, profile update, and order.
   - **Result: Partial Attack Success**; See #4 and #11 in found vulnerabilities
   - The register, login, and profile update pages successfully handled the SQL Injection
   - The attack consisted of “The List of Naughty Words,” and failed. The website handled it appropriately and gave the appropriate errors for all cases.
4. Checking if an address put into order can be a non-valid address.
   - **Test:** Put a non-valid email address in the registration and profile update
   - **Result: Attack Succeeded**; See #9 in found vulnerabilities
5. Can someone create an admin account?
   - **Test:** Create an account with “admin” (or other commonly used admin keywords, such as “test”, “root”, and “user”) as the username.
   - **Result:** We were able to successfully create an account with “admin” as the username. Also note that we were able to make duplicate admin accounts (i.e. there was no check that ensured only 1 admin account was created and used throughout the website’s functionality).
6. Does the website prevent making duplicate username accounts
   - **Test:** Register two users with the same register information
   - **Result: Test Failed**; See #5 in found vulnerabilities
7. Buying items with insufficient account funds.
   - **Test:** Added items to cart with a total greater than current funds.
   - **Result:** A server-side check kept us from doing this.
8. Accessing the ‘admin’ account…
   - **Test:** Brute force checking usernames and passwords in the login field.

    - **Result: Attack Succeeded;** We were able to guess ‘test’ for the username and ‘test’ for the password which is essentially their “admin” account. See vulnerability #4.

9. What happens when you add money to your account?
    - **Test:** I go into the profile and edit the profile to add more money.
    - **Result:** When I add more money I get logged out of my account and I have to sign in again
10. Does the registration check if an email is being passed in?
    - **Test:** I register with new information and for the email box I place an arbitrary string with no ‘@’ or ‘.com’
    - **Result:** We are able to log in successfully since the website is not checking the string I entered as an email
11. Attempt to steal a logged-in user’s session token via HTML cookie image link input **(XSS)**
    - **Test:** We set up a basic Python server with a nohup.out file to capture the requests to the server and send a malicious link with the HTML and script to the site.
    - **Result: Attack Failed;** The headers handled the sent HTML correctly and just displayed it in the header rather than sending a request to the site and stealing the user's cookies.
12. Steal the cookies of another user to submit orders as that user
    - **Test:** Used the inspect feature and tried to steal the cookies available in Applications
    - **Result: Attack Failed**; Cookies are not the main form of session verification, so cannot be stolen to be used by other users for multiple sessions
13. Attempting to change a user’s password by writing a post request for edit profile and changing the password **(CSRF)**
    - **Test:** We copied a skeleton of the form used to edit profiles and created a malicious form that would auto-submit when the link was clicked changing passwords and user info of the user that is logged in
    - **Result: Attack Succeeded;** The password and other user information was successfully changed for the logged-in user allowing us to log in to their account using our changed information. See vulnerability #10
14. Functionality Test: When you cancel a hat order, is your balance properly refunded?
    - **Test:** When inputting the proper credentials for ordering an item, shipping it, and checking shipped orders, there is an option to cancel the order. If you click this option, does it refund your account balance?
    - **Result: Attack Succeeded**. The account balance was not refunded upon canceling an order. See vulnerability #16.

Found Vulnerabilities:

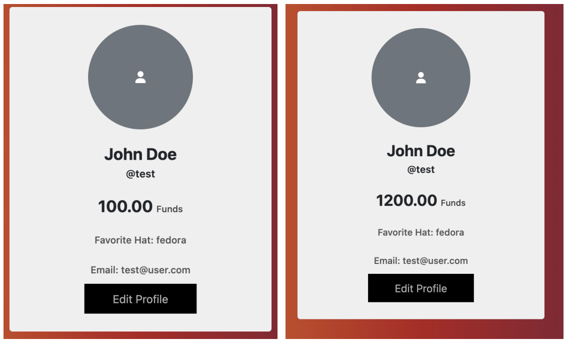
1. Passwords have no minimum restrictions (i.e. no minimum character count, special characters, numbers, capital letters, etc).
    - **Exploit:** Users can put very insecure passwords into the password field, allowing for easily guessed passwords
    - **Fix:** Add password generation verification such as a minimum character count, at least one special character, one uppercase letter, one lowercase letter, and one number.
2. For placing the order the information that is needed doesn't check if it's an address (order specifics)
    - **Exploit:** Users can put any information into the address and they will not get the packages of hats because it's not checking if the address they had placed is an actual address or just a string
    - **Fix:** Add address box to check if the address is correctly inputted by the user
3. Anyone can create an account with an admin username
    - **Exploit:** You can create an account with the username "test" which is the admin account for this website. This allows you to make an admin account effectively.
    - **Fix:** When registering new users, check that the username is unique by checking with the current database of users.
4. Found the information for the "admin" account
    - **Exploit:** Brute force guess and checking usernames and passwords for the test/ admin account
    - **Fix:** Make the password for the test/admin account way stronger and not 4 letters.
5. A single quote in the address field causes a complete website crash
    - **Exploit:** There was an error when checking out where if the requirements for the address fields were not met an error would be thrown. However, somehow the way this error was handled caused the entire website to crash.
    - **Fix:** Implement restrictions before submission of an order to ensure that the address field is a valid entry.
6. Can create account duplicates (username duplicates → ie can create an account with admin username when admin account has already been created)
    - **Exploit:** When two users create an account with the same username, the website does not log in either user and throws the error message: "An error occurred during login"
    - **Fix:** When registering new users, check that the username is unique by checking with the current database of users.
7. You can have the same password and username, you can register twice (and replace the original account).

- **Exploit:** Creating a new account with an identical username as someone else creates a new account. This effectively allows you to lock people out of their accounts by replacing their original ones.
- **Fix:** Check to see if the username for a new account is in use yet before allowing an account to be created.

8. It doesn't tell you that when adding more money to your website it will log you out for security reasons
    - **Exploit:** The user is not notified that the money was placed into their account and this will cause problems if the user didn't mean to put that much money into the account
    - **Fix:** Notify the user that they had placed money into their account before logging them out for security reasons.
9. Email Vulnerability
    - **Exploit:** The various email fields don't check if the input is an email with @ sign and .com.
    - **Fix:** Make the input field type="email" in the HTML to ensure the input must be an email to submit the form.
10. Vulnerable to CSRF attack by making a form on a separate site and submitting an edit profile change
    - **Exploit:** Clicking a malicious link with a pre-filled skeleton form of the edit profile form or other forms on the site that auto-submits will change the info in that form for that logged-in user, leading to account stealing or false orders.
    - **Fix:** Adding a CSRF token to all forms so that the CSRF attacks cannot be successfully deployed against the forms
11. When single characters are put in for every text box in the ordering field, it crashes the website
    - **Exploit:** There was an error when checking out where if the requirements for the address fields were not met an error would be thrown. However, somehow the way this error was handled caused
    - **Fix:** Implement restrictions before submission of an order to ensure that the address field is a valid entry.
12. When changing the password, the same instance of the website logged into another computer (on the same account) is not logged out
    - **Exploit:** When the user has 2 (or more) instances of their account open, and the user changes their password/modifies their account details, the user is not logged out of their account. Note that a user could try to change their password if their account has been compromised, and this vulnerability prevents a successful resolution (on the user's end) to this issue. *This vulnerability was tested on two different computers during the same time interval.

- **Fix:** When a user tries to change their password/username or make changes to their account information, end the user's session or log the user out of their account.

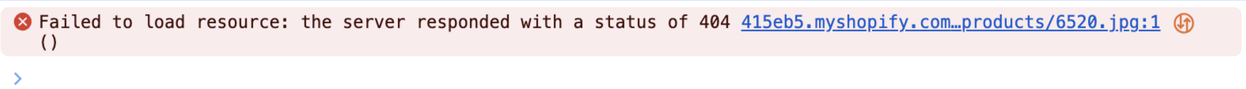
13. Broke into the admin account
    - **Exploit:** Brute forced commonly used admin account usernames and passwords (i.e. we tried "admin", "root", "test", and "user"). Using "test" as both the username and password, we were granted admin access to the website. *Please note that while the website didn't currently have special admin privileges when we tested, this exploit could be an issue in the future, as additional points of access are added.*
    - **Fix:** Add a check, ensuring that the username keywords specified above are not used when the user tries to create a new account.
14. Can add money to the admin account (without a dual-factor authentication):



John Doe
@test
100.00 Funds
Favorite Hat: fedora
Email: test@user.com
Edit Profile

John Doe
@test
1200.00 Funds
Favorite Hat: fedora
Email: test@user.com
Edit Profile

    - **Exploit:** The user is able to add any amount of money to their account, without any form of authentication/assurance that the user wants to add funds.
    - **Fix:** To ensure that the user actually wants to add funds to their account, implement a form of authentication (i.e. prompt the user to re-enter their password, ask for a code sent to their phone/email, etc). Implementing this additional functionality will ensure that the user's account and financial information stays safe.
15. Upon canceling the order for a hat, the user balance was not refunded. Test FAILED.
    - **Exploit:** This is more of a self-explot. You can screw yourself out of some money by ordering something and then canceling it.
    - **Fix:** Refund money to an account when an order has been canceled. Ensure that they are unable to cancel after the order has been delivered.
16. The following error appears when inspecting the website's HTML code:



```
Failed to load resource: the server responded with a status of 404 415eb5.myshopify.com…products/6520.jpg:1
()
```

- **Exploit:** Please note that while the error above is not a direct vulnerability, it's important to note that it could possibly be developed into a vulnerability later on.

Found while analyzing the website via inspect, the error conveys an image failing to load. Even though the image was not loaded onto the website correctly, the HTML markup successfully displayed a concise description reflective of the image.

- **Fix:** Properly upload the image to the site.