Security Audit Report

Focus Areas: API endpoints, authentication, & authorization.

Executive Summary

This security audit aimed to analyze the vulnerabilities of the backend and API request system of WolfCafe. The biggest concern was that there was no limit to the number of API requests a hacker could make in guessing a password as well as in overloading the system with faulty API requests. Suggestions are made to prevent a breakthrough in each section.

Objectives:

Analyze how hackers could…

- Login with false credentials.
- Break the WolfCafe system by sending too many requests.
- Corrupt the database data.

Controller Endpoints

There are a total of 5 controller classes in WolfCafe: AuthController, ItemController, OrderController, RegistrationController, & UserController. Within these 5 controller classes, there are 25 different methods that you can send a REST API to. In other words, there are 25 different API endpoints that a hacker can use to attack WolfCafe.

Logging in as an Admin

Admins have all permissions that staff and customers have in addition to being able to manipulate user account information related to other users. A hacker gaining access to an admin account could result in corruption of data and leaking of sensitive info. Taking a look at the AuthController class, the login() method is responsible for logging in the 3 different types of users: admins, staff members, and customers. There is zero limit to the number of API requests sent to this method, increasing the likelihood that a hacker breaks through if the password is not complex enough.

The recommendation for this issue is to set a limit for the number of requests (perhaps 5) sent to this method by a particular IP address for each hour. This way, even if an admin password is simple, the likelihood of discovering it through brute force is unlikely.

API Request Overload

Spring Boot does not set a limit on the number of API requests sent to the controller methods. Because they use a fixed number of threads to handle requests, a hacker sending hundreds of requests in a short period of time could crash the system. This, in other words, would result in a successful DoS attack. To fix this, set a limit on the number of requests that can be sent from one IP address. If you wanted to prevent a crash resulting from attacks being sent from multiple IP addresses, you could set a limit on the max number of requests sent in from all sources. Using software integrated from Cloudflare would handle this.

Corruption of Database Data

All controllers except for RegistrationController have the ability to corrupt database data. However, all of the methods in these controllers capable of this have the PreAuthorize annotation that either requires an admin or staff user to use this method. This results in a correct authorization that won't get broken through unless a hacker finds the credentials for a staff or admin user.