AWAS Project

The project consists of a dynamic web application with three different vulnerabilities. The vulnerabilities are inspired by the "Bypassing Client-side Controls" "Attacking Authentication" and "Code Injection Attacks" chapters from the AWAS course. The project violates some of the basic defense mechanisms, such as:

1. Handling authentication - "Bypassing Client-side Controls" chapter focuses mainly on bypassing restrictions on the client side to gain access to hidden forms or different kinds of functionalities. In this project, there is a hidden client side functionality during the registration process. If the attacker exploits this functionality, then the attacker will be able to privileged access. Part of the developer's responsibility is to ensure the safety of the users. By implementing the security mechanisms on the client-side, the developer endangers the users, by enabling the attacker to bypass the security mechanisms on the client-side.

2. Handling attackers - "Attacking Authentication" chapter provides different insights on security measures for handling authentication functionalities. Having a descriptive error message instead of a generic one causes more harm to the users as it might reveal important information to attackers. In this project, the error messages when registering new users are not generic. Therefore, an attacker is able to use the information revealed in the error messages during the registration process, to gain more information on the status of other user accounts. Another vulnerability is not stopping/delaying the attack when the leaked information gets used in an attack.

3. Handling user input - "Code Injection Attacks" chapter consists of user input and its risks. When the user visits the dashboard page, there is an input section that an attacker could use to exploit and extract more information about the details hidden in the database. When creating any kind of user input it is fundamental to sanitize the user input to avoid any kind of exploitation and information leak from the server-side using the user input.

App preview: https://www.youtube.com/watch?v=B73Hu7swBhQ

Introduction

Welcome to the very secure website. There are two tasks in this project.

1. Find the flag
2. Find Mr. King's first name, address and phone number

Find the flag

The flag only appears to "privileged" users. There are two ways to find it. Compromise an already existing account, or exploit a vulnerability to create a new privileged account.

If you choose the former, you can use brute force attacks to find the user name and the password. The wordlists are provided with the project files.

Data extraction

After logging in you will be able to make queries to the database via a form field. Here, the goal is to see if the input is vulnerable to injection attacks. If it is, you need to extract the personal information specified in the task.