SCENARIO

In this lab environment, the implemented login mechanism employs rate limiting as a defense against brute-force attempts. Nevertheless, due to an inherent race condition, this defensive measure is not entirely foolproof. The objective of this exercise is to exploit the race condition flaw, employ brute-force to crack the password of user 'carlos', log in with this user, and subsequently delete the said user.

# PROCEDURE

1. Start by purposefully entering incorrect passwords for our account. After three consecutive failed login attempts, we notice a temporary blockage. Trying a different username, however, still returns the standard "Invalid username or password" message. This suggests rate limiting is based on the username and not the session.
2. There could be a delay between when a login attempt is submitted and when the server increments the failed attempt count for a username.
3. Using Burp Suite, make several POST requests to the /login endpoint. If more than three requests return the "Invalid username or password" message before getting a blockage message, it confirms our hypothesis.
4. Send multiple login requests simultaneously using Burp's Turbo Intruder. Use the given list of potential passwords for 'carlos'. If done right and fast enough, one of these should succeed before the server can react and block the attempts.
5. Among the responses, identify any with a 302 status. This is likely a successful login attempt. Make a note of the password that caused this.
6. Once the temporary block has been lifted, use the obtained password to log in as 'carlos'. Access the admin panel.
7. Delete the 'carlos' user account to complete the lab challenge.

## PAYLOAD

The key payload involves rapidly sending POST requests with varying passwords for the carlos user. This can be exemplified as multiple HTTP POST requests to /login, each carrying a different password from the provided list.

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Enhance Rate Limiting:** Instead of rate limiting based solely on the username, consider implementing IP-based or session-based rate limiting.
2. **Implement Account Lockouts:** After a certain number of failed attempts, lock out the account for a prolonged period, making brute-force attacks infeasible.
3. **CAPTCHAs:** Introduce CAPTCHAs after a couple of failed login attempts to deter automated login attempts.
4. **Monitor and Alert:** Implement systems that can detect and alert on abnormal behaviors, such as rapid-fire login attempts, even if they fall within rate limits.