**SCENARIO**

This lab's password change functionality is susceptible to brute-force attacks. The objective is to utilize a list of candidate passwords to brute-force Carlos's account and then access his "My account" page.

Your provided credentials: wiener:peter

Victim's username: carlos

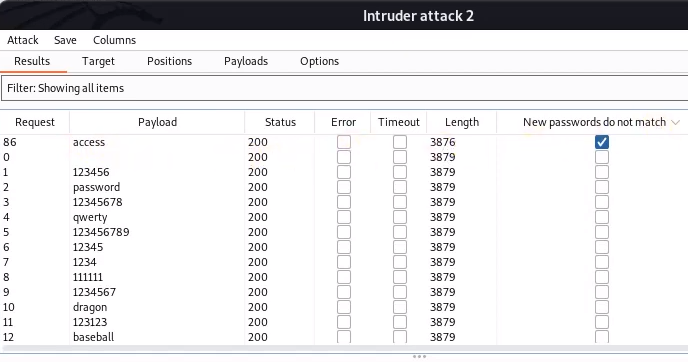
**PROCEDURE**

1. Initiate Burp and log into the application. Investigate the password change feature. Note that the username is presented as a hidden input in the POST request.
2. Observe the application behavior when an incorrect current password is entered. When the two new password entries match, the account gets locked. If they don’t match, an error message appears stating "Current password is incorrect". Using a correct current password but differing new passwords results in the message: "New passwords do not match." This discrepancy allows for password enumeration.
3. Use your valid current password but enter two different new passwords. Forward this POST /my-account/change-password request to Burp's Intruder feature.
4. Within Burp Intruder, modify the username parameter to carlos. Establish a payload marker for the current-password parameter. Ensure the new password parameters remain distinct, like:

sqlCopy code

1. username=carlos&current-password=§incorrect-password§&new-password-1=123&new-password-2=abc
2. Navigate to the Payloads tab, then insert the candidate password list as the payload set.
3. On the Settings tab, integrate a grep match rule to highlight responses that include "New passwords do not match". Launch the attack.
4. After the attack concludes, pinpoint the response with the "New passwords do not match" message. Document the identified password.
5. Logout from your account in the browser. Log back in with the carlos username using the recently discovered password.
6. Select "My account" to successfully complete the lab.

**PAYLOAD**

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Account Lockout:** Implement account lockout policies after several failed login or password change attempts.
2. **Rate Limiting:** Introduce rate limiting on the password change functionality to prevent rapid-fire attempts.
3. **Consistent Error Messages:** Use consistent error messages irrespective of the reason for the password change failure. For instance, "Password change unsuccessful" without specifying the exact reason.
4. **CAPTCHA:** Add a CAPTCHA system on the password change page to deter automated brute-force attempts.