**SCENARIO**

This lab is vulnerable to username enumeration and password brute-force attacks. It features an account with a predictable username and password, which are hinted in the provided wordlists: 'Candidate usernames' and 'Candidate passwords'. The objective is to enumerate a valid username, brute-force this user's password, and subsequently access the user's account page.

**PROCEDURE**

1. With Burp Suite active, navigate to the login page and input an arbitrary username and password.
2. In Burp Suite, access Proxy > HTTP history and locate the POST /login request.
3. Highlight the username parameter value in the request and dispatch it to Burp's Intruder.
4. Within Burp Intruder, visit the Positions tab. The username parameter should be automatically tagged as a payload position, marked by two § symbols. Keep the password parameter constant.
5. Ensure that the "Sniper" attack type is active.
6. On the Payloads tab, select the "Simple list" payload type.
7. Under Payload settings, input the list of potential usernames. Click "Start attack".
8. Upon completion, analyze the Length column in the Results tab. Identify the distinct payload that returns "Incorrect password" rather than "Invalid username". Document this username.
9. Return to the Positions tab. Reset positions, replace the username with the one identified, and set a payload position for the password.
10. Within the Payloads tab, replace the usernames with candidate passwords. Initiate the attack.
11. After completion, scrutinize the Status column. Identify the lone request with a 302 response, signifying a successful login. Note down the corresponding password.
12. Log into the lab application using the discovered username and password, and navigate to the user's account page to complete the lab.

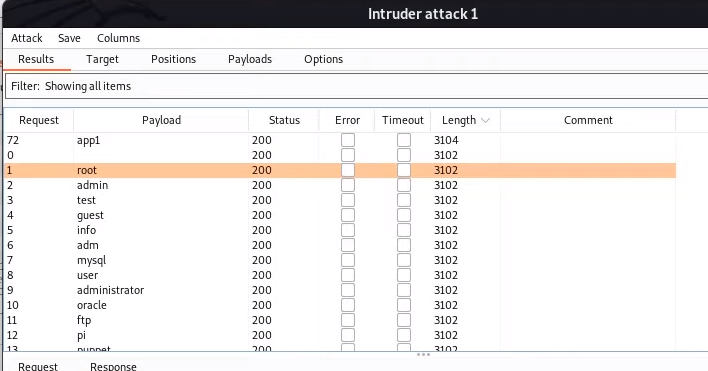
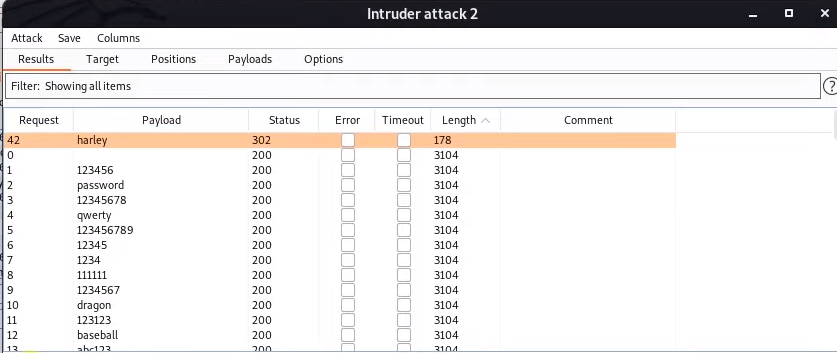
**PAYLOAD**

1. For username enumeration:

username=§candidate-username§&password=static-password

1. For password brute-forcing:

username=identified-username&password=§candidate-password§

**  
PROOF OF CONCEPT  
  
REMEDIATION**

1. **Limit Login Attempts:** Implement a mechanism to restrict the number of login attempts, introducing delays or temporary locks after a certain threshold.
2. **Unified Error Messages:** Always return a generalized error message, such as "Invalid credentials", to avoid giving away clues on valid usernames.
3. **Implement CAPTCHA:** Incorporate CAPTCHA challenges in the login form to deter automated brute-force attempts.
4. **Account Lockouts:** Implement a lockout mechanism after a predefined number of unsuccessful attempts to login.
5. **Multi-factor Authentication:** Implement multi-factor authentication as an additional security layer.