**SCENARIO**

This lab's password reset functionality is vulnerable. The objective is to exploit this vulnerability, reset Carlos's password, then log in to his account and access his "My account" page. Credentials provided for testing:

Username: wiener

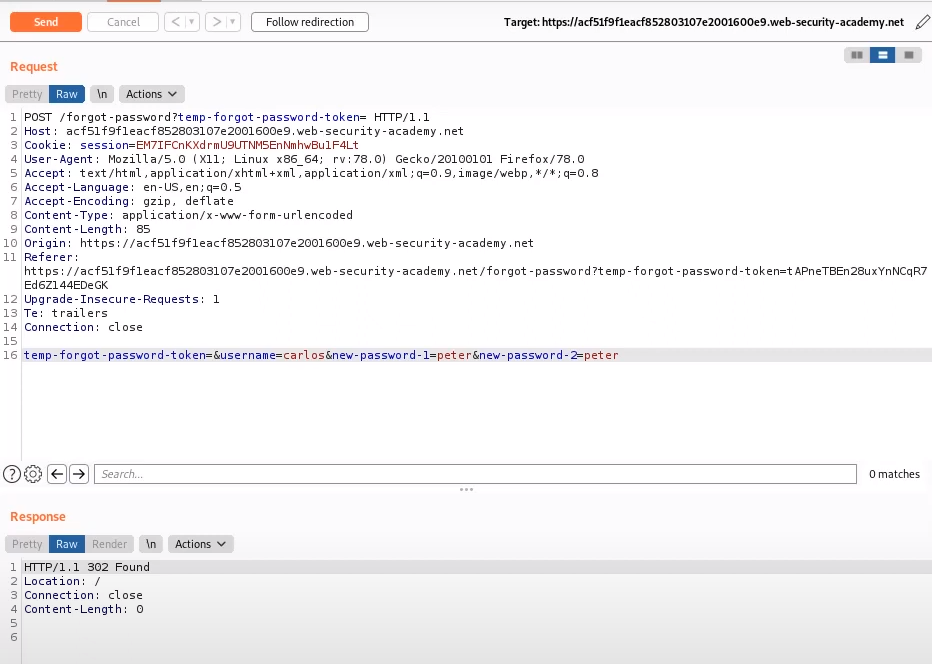
Password: peter

Victim's username: Carlos

**PROCEDURE**

1. Start with Burp running and click the "Forgot your password?" link. Enter your own username: wiener.
2. Use the "Email client" button to view the password reset email. Follow the link in the email and reset your password to any desired value.
3. In Burp, navigate to Proxy > HTTP history. Examine the requests and responses tied to the password reset functionality. Take note of the reset token present as a URL query parameter in the email.
4. Observe that in the POST /forgot-password?temp-forgot-password-token request, the username is given as a hidden input. Forward this request to the Burp Repeater.
5. Within Burp Repeater, realize that the password reset feature remains operational even when the value of the temp-forgot-password-token parameter is removed from both the URL and request body. This indicates the token isn't verified upon new password submission.
6. Request another password reset for your account and alter your password once more. Again, send the POST /forgot-password?temp-forgot-password-token request to Burp Repeater.
7. In Burp Repeater, eliminate the value for the temp-forgot-password-token parameter from both the URL and the body of the request. Modify the username parameter to "carlos" and set the new password to any value of your choosing. Execute the request.
8. In your browser, log in using Carlos's username and the newly set password. Click on the "My account" link to finalize the lab.

**PAYLOAD**

**  
PROOF OF CONCEPT  
  
REMEDIATION**

1. **Token Verification:** Always verify the validity and ownership of reset tokens before allowing password changes.
2. **Token Uniqueness:** Ensure that tokens are unique and tied to specific users and actions.
3. **Token Expiry:** Implement an expiry time for tokens, ensuring they cannot be used indefinitely.
4. **Limit Reset Requests:** Implement a rate limit for password reset requests to prevent abuse.
5. **Logs and Monitoring:** Monitor and log password reset activities to detect and react to suspicious activities promptly.