SCENARIO

The application uses a serialization-based session mechanism and is vulnerable to privilege escalation as a result. We will try to edit the serialized object in the session cookie to exploit this vulnerability and gain administrative privileges.

**PROCEDURE**

1. Go the application and login using the given credentials to act as an user.
2. Navigate to the **My Account** page.
3. Open the BurpSuite’s Proxy’s HTTP History and study the request carefully.
4. We notice that the session cookie is encoded in Base64 and then in URL encoding.
5. We decode it in BurpSuite and we can see that it appears in the format below:

**O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}**

1. So, we change the value of **b:0 to b:1** to send the request as an admin.
2. Now we access the **Admin Panel** using the response we got and delete the user by hitting the endpoint we found in the page.

**PAYLOAD**

O:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:1;}

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. **Avoid Direct Object References:** Do not serialize sensitive data directly in the session. Any direct references to objects or other server-side internals could provide an attacker with too much insight into the application's inner workings.
2. **Signed Sessions:** Ensure that the session cookies are signed to prevent any tampering. A digital signature can be used to ensure that the session data hasn't been altered during transmission.
3. **Encryption:** Even if it's just a session ID or non-sensitive data, it's a good practice to encrypt the serialized data to make it unreadable to anyone who might intercept it.
4. **Strong Session Management:** Implement a robust session management system that checks the validity of session tokens and employs strong session identifiers that can't be easily guessed.
5. **Regular Patching:** Ensure that all libraries related to serialization are kept updated. Often, security patches are released to address known vulnerabilities.