SCENARIO

The application a serialization-based session mechanism and is vulnerable to authentication bypass 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:12:"access\_token";s:32:"y4esqtkxwo718kbom0duosnt6nvln9f5";}**

1. So, we inject the payload in place of the above value.
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:13:"administrator";s:12:"access\_token";i:0;}

**REMEDIATION**

1. **Avoid Storing Sensitive Information in Cookies:** As a general rule, sensitive information such as access tokens, roles, or permissions should not be stored in cookies, even if they are encrypted. Instead, use server-side sessions that are referenced by a session ID.
2. **Implement Strong Session Management:** Ensure a strong session management system that checks the validity of session tokens, employs strong session identifiers that can't be easily guessed, and keeps track of valid sessions server-side.
3. **Signed Sessions:** The session cookies should be signed to prevent any tampering. A digital signature can help ensure that the session data hasn't been altered during transmission.
4. **Encrypt Serialized Data:** If you must store serialized data, make sure it's encrypted using a strong encryption algorithm. This ensures that even if an attacker is able to obtain the serialized data, they won't be able to understand or modify it without the encryption key.
5. **Regularly Update Libraries:** Serialization vulnerabilities often come from outdated libraries. Keep all libraries updated, especially those responsible for serialization.