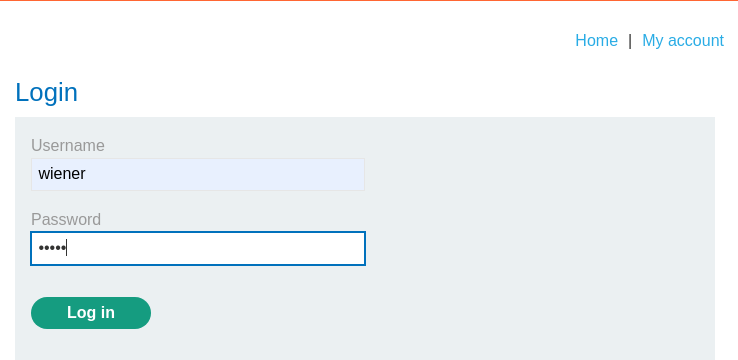
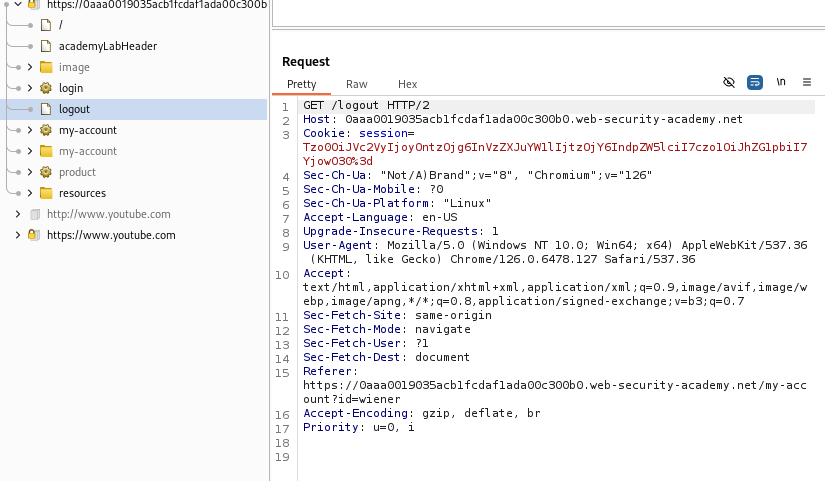
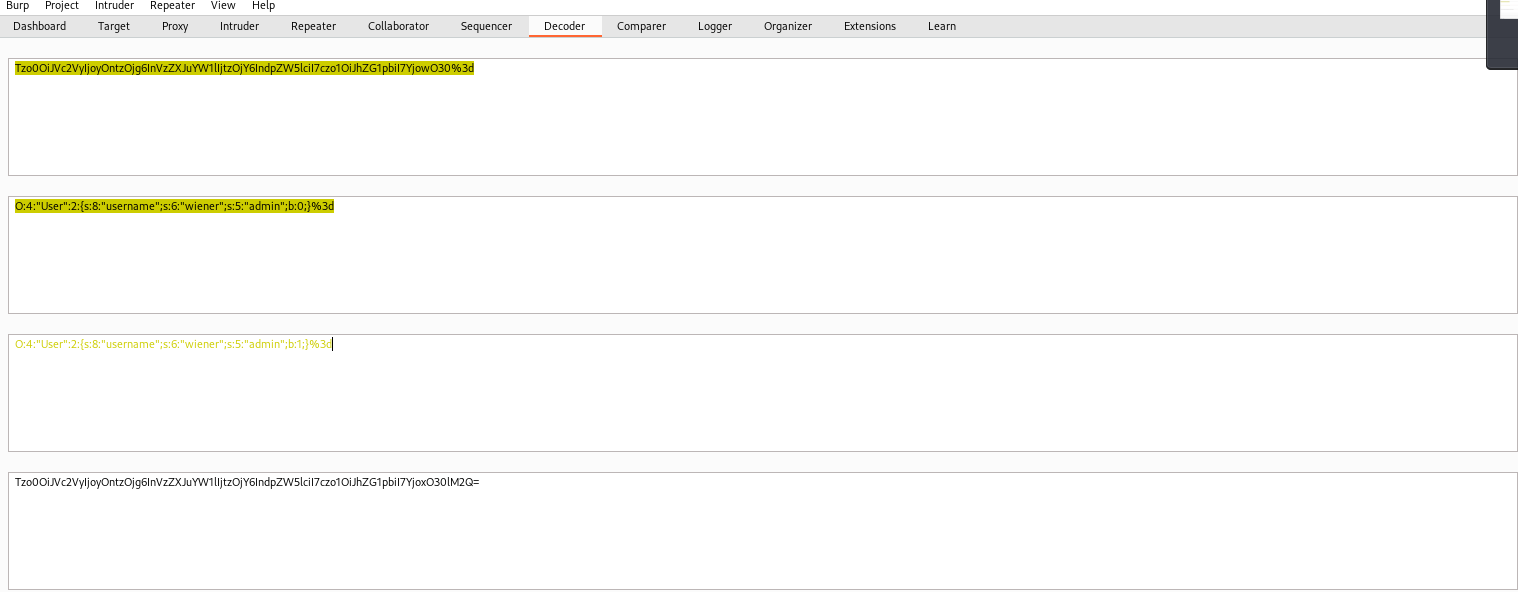
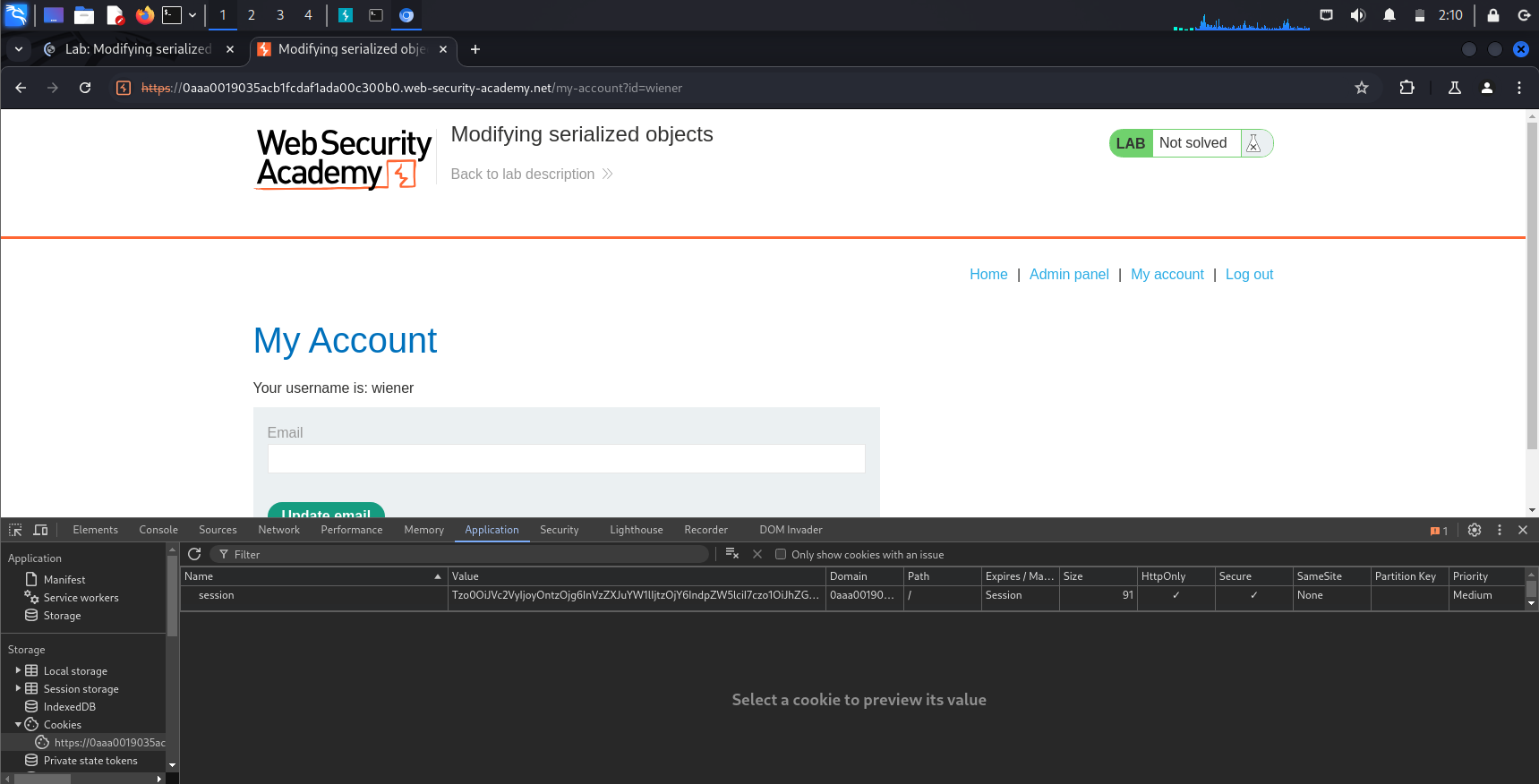
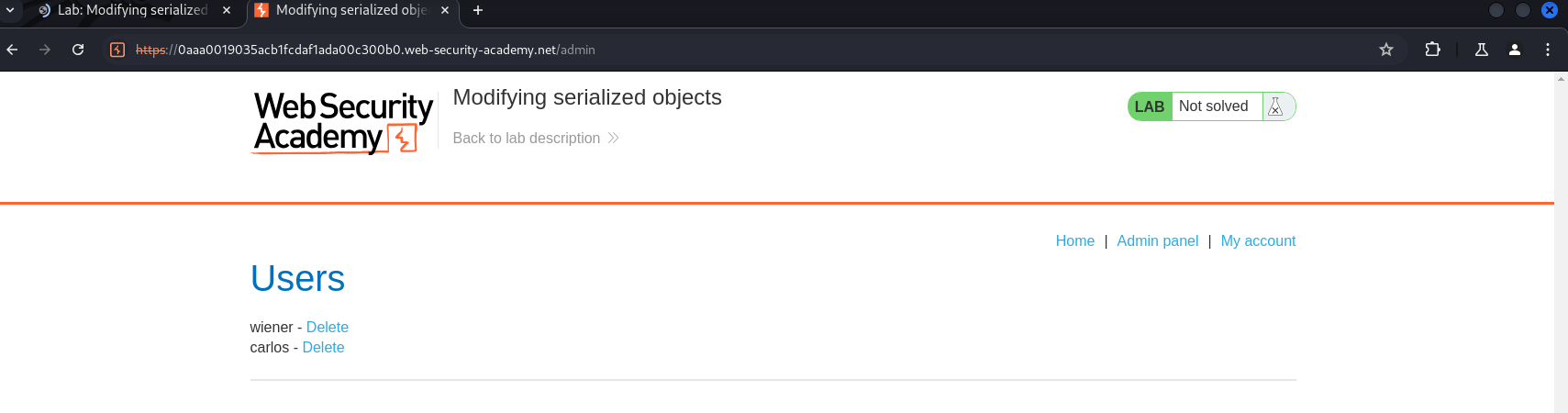
# Lab1: Modifying serialized objects

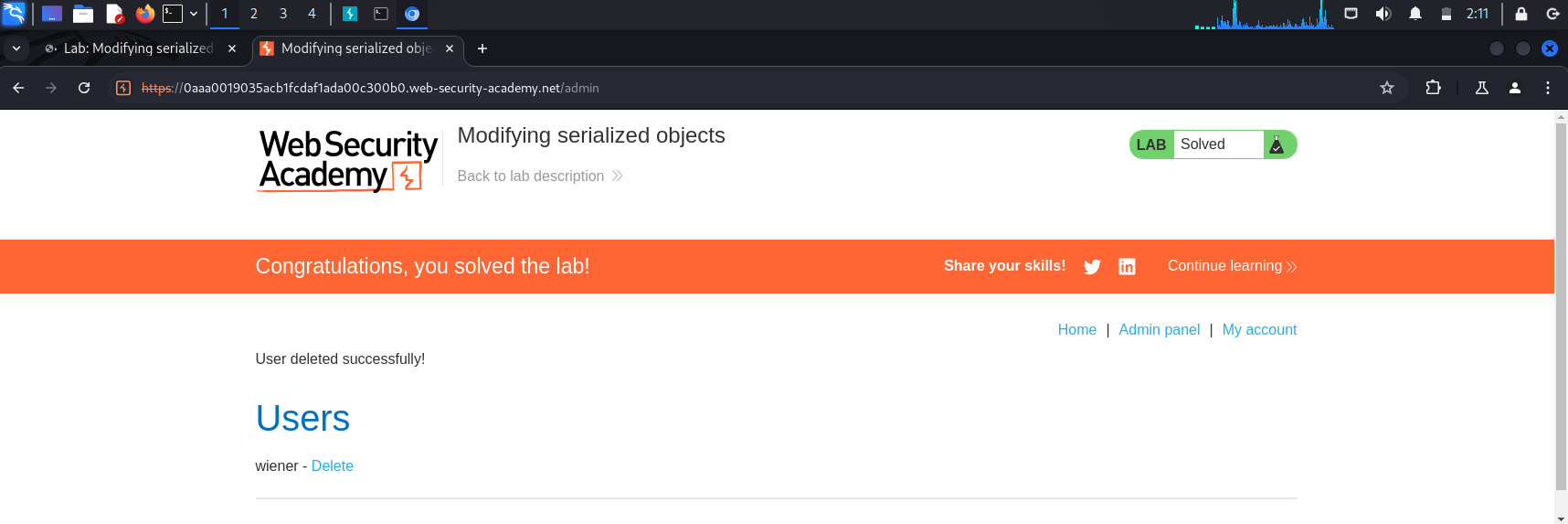
1. **NỘI DUNG CHỦ ĐỀ**
   1. **Đặc điểm và nguyên tắc hoạt động**

* Tấn công này lợi dụng cơ chế session dựa trên serialization để nâng cao quyền truy cập.
* Tin tặc có thể chỉnh sửa đối tượng đã được serialized trong cookie để thay đổi quyền truy cập.
* Ví dụ, một thuộc tính admin có thể bị đổi từ giá trị false thành true để truy cập vào các trang quản trị.
  1. **Phương pháp ngăn chặn**
* Kiểm tra và xác thực giá trị cookie trước khi sử dụng.
* Sử dụng cơ chế mã hóa và chữ ký số để đảm bảo tính toàn vẹn của dữ liệu.
* Hạn chế quyền truy cập của người dùng và thực hiện kiểm tra quyền truy cập ở phía máy chủ.

1. **Các bước thực hiện bài Lab**
   1. **Mục tiêu**

Khai thác lỗ hổng trong cơ chế session của ứng dụng để nâng quyền truy cập lên admin và xóa người dùng carlos.

* 1. **Thực hiện**
  + Đăng nhập bằng tài khoản: **wiener:peter.**
  + ****
  + Lấy session cookie sau khi đăng nhặp
  + ****
  + Sử dụng công cụ decoder để giải mã cookie, nhận ra đó là đối tượng PHP được serialized.
  + ****
  + Thay đổi giá trị thuộc tính admin từ b:0 thành b:1.
  + ****
  + Copy đoạn mã gán vào session của trang hiện tại nó sẽ hiện ra Admin panel ta sẽ vào đó để xóa người dùng Carlos
  + ****
  + ****
  1. **Kết quả**
* Đã truy cập thành công vào trang quản trị với quyền admin và xóa người dùng carlos.

****

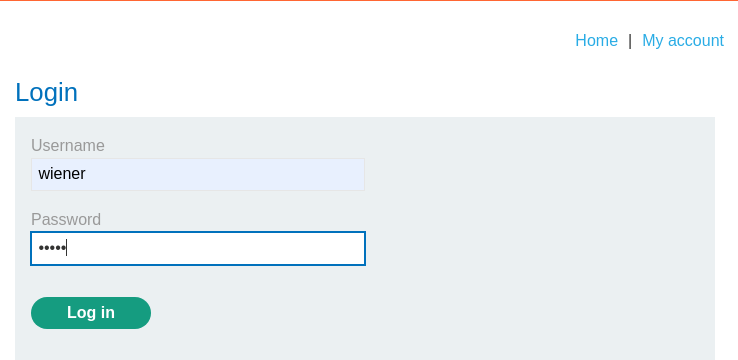
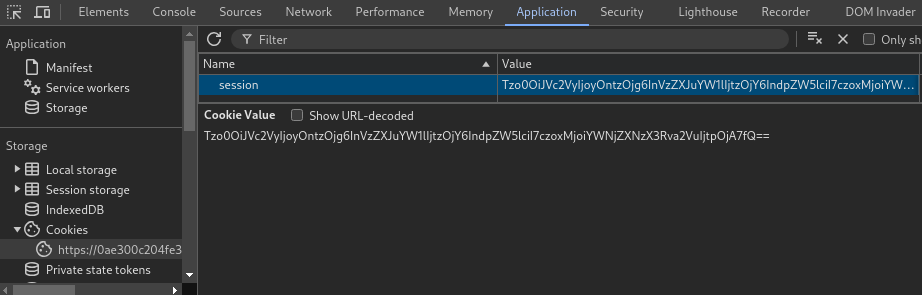
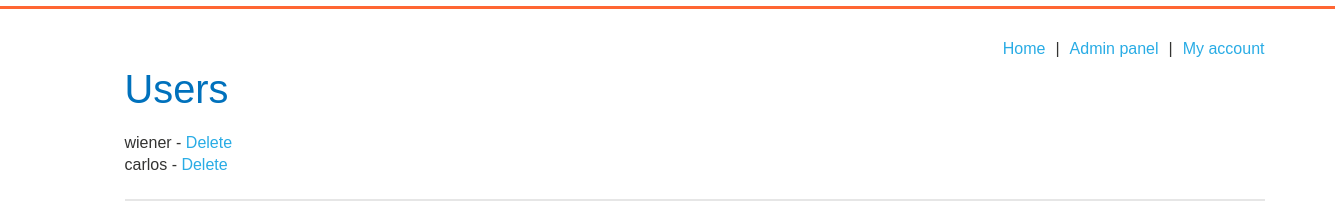
# Lab2: Modifying serialized data types

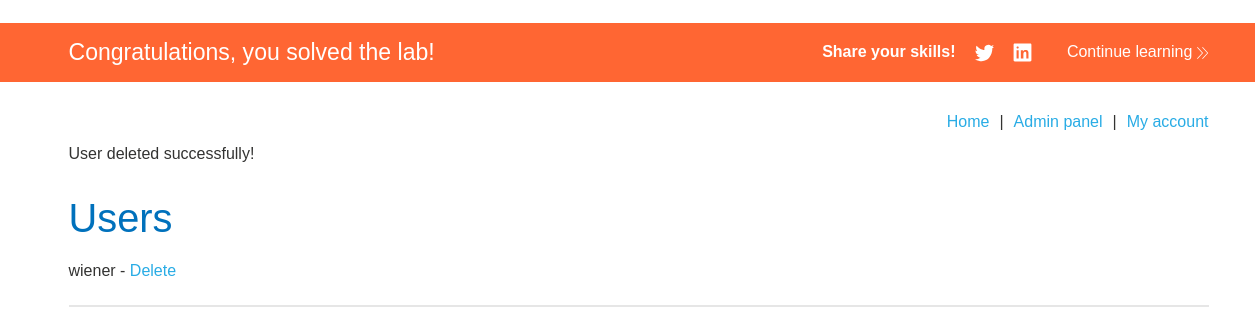
1. **NỘI DUNG CHỦ ĐỀ**
   1. **Đặc điểm và nguyên tắc hoạt động**
   * Tấn công này lợi dụng cơ chế session dựa trên serialization để bypass xác thực người dùng.
   * Tin tặc có thể chỉnh sửa đối tượng đã được serialized trong cookie để truy cập vào tài khoản khác.
   * Ví dụ, việc so sánh dữ liệu khác loại trong PHP có thể bị lợi dụng để chỉnh sửa thuộc tính của đối tượng serialized.
   1. **Phương pháp ngăn chặn**

* Kiểm tra và xác thực giá trị cookie trước khi sử dụng.
* Sử dụng cơ chế mã hóa và chữ ký số để đảm bảo tính toàn vẹn của dữ liệu.
* Hạn chế quyền truy cập của người dùng và thực hiện kiểm tra quyền truy cập ở phía máy chủ.

1. **Các bước thực hiện bài Lab**
   1. **Mục tiêu**

Khai thác lỗ hổng trong cơ chế session của ứng dụng để truy cập vào tài khoản quản trị viên và xóa người dùng carlos.

* 1. **Thực hiện**
  + Đăng nhập bằng tài khoản: **wiener:peter.**
  + ****
  + Lấy session cookie sau khi đăng nhập
  + ****
  + Chỉnh sửa session cookie: Đổi access token thành số nguyên 0 và loại bỏ dấu ngoặc kép quanh giá trị, Cập nhật nhãn loại dữ liệu cho access token từ s thành i.
  + ****
  + Copy đoạn mã gán vào session của trang hiện tại nó sẽ hiện ra Admin panel ta sẽ vào đó để xóa người dùng Carlos
  + ****
  + ****
  1. **Kết quả**
* Đã truy cập thành công vào trang quản trị với quyền admin và xóa người dùng carlos.

****