Access control vulnerabilities and privilege escalation.

I.Kiểm soát truy cập .

- Kiểm soát truy cập (hoặc ủy quyền) là việc áp dụng các ràng buộc đối với ai (hoặc cái gì) có thể thực hiện các hành động đã cố gắng hoặc truy cập tài nguyên mà họ đã yêu cầu. Trong ngữ cảnh của các ứng dụng web, kiểm soát truy cập phụ thuộc vào xác thực và quản lý phiên:

* **Xác thực** xác định người dùng và xác nhận rằng họ là chính họ.
* **Quản lý phiên** xác định các yêu cầu HTTP tiếp theo nào đang được thực hiện bởi chính người dùng đó.
* **Kiểm soát truy cập** xác định xem người dùng có được phép thực hiện hành động mà họ đang cố gắng thực hiện hay không.
* Phân loại :
* Vertical access control (dọc)
* Horizontal access control (ngang)
* Context-dependent access control (ngữ cảnh)

II.Vertical access control.

* Kiểm soát truy cập theo chiều dọc là cơ chế hạn chế quyền truy cập vào chức năng nhạy cảm không có sẵn cho các kiểu người dùng khác.
* Với các điều khiển truy cập dọc, người dùng khác nhau có quyền truy cập vào các chức năng ứng dụng khác nhau. Ví dụ: quản trị viên có thể sửa đổi hoặc xóa tài khoản của bất kỳ người dùng nào, trong khi người dùng bình thường không có quyền truy cập vào các hành động này.
* Kiểm soát truy cập theo chiều dọc có thể là cách triển khai chi tiết hơn của các mô hình bảo mật được thiết kế để thực thi các chính sách kinh doanh như tách biệt các nhiệm vụ và ít đặc quyền nhất.

III.Horizontal access control.

* Kiểm soát truy cập theo chiều ngang là cơ chế hạn chế quyền truy cập vào tài nguyên đối với những người dùng được phép truy cập cụ thể vào các tài nguyên đó.
* Với điều khiển truy cập ngang, những người dùng khác nhau có quyền truy cập vào một tập hợp con các tài nguyên cùng loại.

IV. Context-dependent access control.

* Kiểm soát truy cập phụ thuộc vào ngữ cảnh hạn chế quyền truy cập vào chức năng và tài nguyên dựa trên trạng thái của ứng dụng hoặc tương tác của người dùng với nó.
* Các kiểm soát truy cập phụ thuộc vào ngữ cảnh ngăn người dùng thực hiện các hành động không đúng thứ tự.

V.LAB .

*“Nếu người dùng có thể có quyền truy cập vào chức năng mà họ không được phép truy cập thì đây là sự leo thang đặc quyền theo chiều dọc. “*

Lab1 : Ở mức cơ bản nhất, sự leo thang đặc quyền theo chiều dọc phát sinh khi ứng dụng không thực thi bất kỳ biện pháp bảo vệ nào đối với chức năng nhạy cảm.

* Chức năng admin không được bảo vệ : robots.txt-> admin panel

Lab2 : Trong th trang admin không được bảo vệ nhưng được giấu đi .

* View source -> find admin panel

*“Một số ứng dụng xác định quyền truy cập của người dùng hoặc vai trò khi đăng nhập, sau đó lưu trữ thông tin này ở vị trí người dùng có thể kiểm soát, chẳng hạn như trường ẩn, cookie hoặc tham số chuỗi truy vấn đặt trước. Ứng dụng đưa ra các quyết định kiểm soát truy cập tiếp theo dựa trên giá trị đã gửi”*

Lab3 : Vai trò người dùng được kiểm soát qua tham số .

* Login ‘wiener : peter’ -> modify cookie ‘admin : true’ - > admin panel .

Lab 4 : Vai trò của người dùng được sửa đổi qua hồ sơ của họ .

* Login “wiener : peter”
* Update email -> response include roleid = 1 -> add roleid = 2 in request update email and resent .
* Access /admin , remove calos

***“****Một số ứng dụng thực thi các kiểm soát truy cập ở lớp nền tảng bằng cách hạn chế quyền truy cập vào các URL và phương thức HTTP cụ thể dựa trên vai trò của người dùng”*

*“Một số khuôn khổ ứng dụng hỗ trợ các tiêu đề HTTP không chuẩn khác nhau có thể được sử dụng để ghi đè URL trong yêu cầu ban đầu, chẳng hạn như X-Original-URL và  X-Rewrite-URL. Nếu một trang web sử dụng các kiểm soát giao diện người dùng nghiêm ngặt để hạn chế quyền truy cập dựa trên URL, nhưng ứng dụng cho phép ghi đè URL qua tiêu đề yêu cầu, thì có thể bỏ qua các kiểm soát truy cập bằng cách sử dụng một yêu cầu như sau:*

*POST / HTTP/1.1  
X-Original-URL: /admin/deleteUser”*

Lab 5 : Kiểm soát dựa trên URL có thể bị phá vỡ .

* Access GET /admin -> deny
* Add in header : X-Original-URL: /admin -> access adin panel
* Request : GET /delete?username=calos

X-Original-URL: /admin

*“Một cuộc tấn công thay thế có thể phát sinh liên quan đến phương thức HTTP được sử dụng trong yêu cầu. Các điều khiển giao diện người dùng ở trên hạn chế quyền truy cập dựa trên URL và phương thức HTTP. Một số trang web có thể chấp nhận các phương thức yêu cầu HTTP thay thế khi thực hiện một hành động. Nếu kẻ tấn công có thể sử dụng GET(hoặc một phương thức khác) để thực hiện các hành động trên một URL bị hạn chế, thì chúng có thể phá vỡ kiểm soát truy cập được triển khai ở lớp nền tảng.”*

Lab6 : Kiểm soát truy cập dựa trên phương thức có thể bị phá vỡ .

* Login “administrator : admin” -> upgrade role of carlos .
* Login “wiener : perter” -> replace cookie for admin request.
* Sent request upgrade -> "Unauthorized" -> change POST -> POSTX -> "Missing parameter 'username'"
* Change method to GET :

-> GET /adminroles?username=wiener&action=upgrade