**Những lỗ hổng kiểm soát truy cập**

**và leo thang đặc quyền**

**Kiểm soát truy cập là gì?**

Kiểm soát truy cập là áp dụng những ràng buộc về việc ai (hoặc cái gì) có thể thực hiện các hành động hay truy cập những mã nguồn họ yêu cầu. Trong bối cảnh của ứng dụng web, phân quyền dựa trên xác thực và quản lí phiên.

* **Xác thực**: xác định người dùng và xác nhận xem họ có đúng như họ nói
* **Quản lí phiên**: xác định những yêu cầu http tiếp theo có được làm từ cùng 1 người hay không
* **Kiểm soát truy cập**: xác định xem người dùng đó được phép thực hiện những hành động nào.

Kiểm soát truy cập bị hỏng là 1 lỗ hổng bảo mật thường gặp và rất nghiêm trọng. Thiết kế và quản lí kiểm soát truy cập là 1 vấn đề phức tạp và năng động áp dụng các ràng buộc về kinh doanh, tổ chức và pháp lý đối với việc triển khai kĩ thuật. Những quyết định kiểm soát truy cập phải được làm bởi con người chứ không phải công nghệ và khả năng xảy ra lỗi rất cao.

Từ khía cạnh của người dùng, kiểm soát truy cập được chia ra thành những loại sau:

* Kiểm soát truy cập theo chiều dọc
* Kiểm soát truy cập theo chiều ngang
* Kiểm soát truy cập dựa trên bối cảnh

**Kiểm soát truy cập theo chiều dọc**

Là các kĩ thuật hạn chế truy cập chức năng nhạy cảm không khả thi cho các loại người dùng khác.

Với kiểm soát truy cập theo chiều dọc, nhiều loại người dùng khác nhau sẽ truy cập các chức năng khác nhau. Ví dụ, quản trị viên có thể điều chỉnh hoặc xóa tài khoản người dùng, trong khi người dùng lại không.

**Kiểm soát truy cập theo chiều ngang**

Là các kĩ thuật hạn chế truy cập tài nguyên mà chỉ những người dùng đặc biệt mới được phép truy cập nó.

Với kiểm soát truy cập theo chiều ngang, những người dùng khác nhau sẽ truy cập vào 1 tập hợp con các tài nguyên cùng loại. Ví dụ, 1 ứng dụng ngân hàng sẽ cho phép người dùng xem giao dịch và thanh toán từ chính tài khoản của họ chứ không phải tài khoản của người khác.

**Kiểm soát truy cập dựa trên bối cảnh**

Là các kĩ thuật hạn chế truy cập chức năng và tài nguyên dựa vào trạng thái của ứng dụng hay sự tương tác của người dùng với nó.

Kiểm soát truy cập dựa trên bối cảnh ngăn người dùng thực hiện các hành động sai thứ tự. Ví dụ, 1 trang web bán lẻ có thể ngăn chặn người dùng không điều chỉnh nội dung của giỏ hàng sau khi đã thanh toán.

**Những ví dụ về kiểm soát truy cập bị hỏng**

Những lỗ hổng liên quan đến kiểm soát truy cập bị hỏng tồn tại khi người dùng có thể truy cập tài nguyên và thực hiện những hành động không được phép.

**Leo thang đặc quyền theo chiều dọc**

Nếu 1 người dùng có thể truy cập chức năng mà họ không được phép truy cập thì đó là leo thang đặc quyền theo chiều dọc. Ví dụ, nếu bạn không phải là quản trị viên mà có thể truy cập trang admin để xóa, chỉnh sửa tài khoản người dùng.

**Chức năng không được bảo vệ**

Leo thang đặc quyền theo chiều dọc sinh ra khi 1 ứng dụng không thực thi bảo vệ cho chức năng nhạy cảm. Ví dụ, những chức năng quản trị có thể được liên kết từ 1 trang chào mừng quản trị chứ không phải người dùng. Tuy nhiên, người dùng có thể truy cập các chức năng quản trị bằng cách duyệt 1 Url liên quan đến admin

Ví dụ, 1 trang web có thể lưu trữ chức năng nhạy cảm ở Url sau:

https://insecure-website.com/admin

Bất cứ ai có đường dẫn đều có thể truy cập chức năng của admin. Trong một vài trường hợp, Url quản trị có thể được tiết lộ ở vị trí khác như robots.txt

https://insecure-website.com/robots.txt

Thậm chí nếu url này không được tiết lộ, kẻ tấn công có thể dùng wordlists để tấn công dồn dập vị trí của các chức năng nhạy cảm.

Trong một vài trường hợp, chức năng nhạy cảm không được bảo vệ 1 cách nghiêm ngặt nhưng được che giấu bằng cách cho nó 1 url khó đoán hơn. Nếu chỉ ẩn chức năng nhạy cảm sẽ không kiểm soát truy cập hiệu quả vì người dùng vẫn có thể khám phá Url bị xáo trộn theo nhiều cách khác nhau.

Ví dụ, xem xét 1 ứng dụng lưu trữ chức năng quản trị ở url sau:

https://insecure-website.com/administrator-panel-yb556

Kẻ tấn công có thể không đoán được. Tuy nhiên, ứng dụng có thể rò rỉ Url cho người dùng. Ví dụ, url có thể bị tiết lộ trong Javascript để xây dựng giao diện người dùng dựa trên vai trò của họ

<script>

var isAdmin = false;

if (isAdmin) {

...

var adminPanelTag = document.createElement('a');

adminPanelTag.setAttribute('https://insecure-website.com/administrator-panel-yb556');

adminPanelTag.innerText = 'Admin panel';

...

}

</script>

Đoạn mã này thêm đường dẫn vào giao diên người dùng nếu họ là admin. Tuy nhiên, nó chứa 1 url có thể nhìn thấy bởi tất cả người dùng.

**Kiểm soát truy cập dựa trên tham số**

Một vài ứng dụng xác định vai trò của người dùng ở trang đăng nhập và sau đó lưu trữ thông tin này ở vị trí người dùng có thể kiểm soát được như mục ẩn, cookie hoặc tham số. Ứng dụng kiểm soát truy cập dựa trên giá trị được gửi đi. Ví dụ:

https://insecure-website.com/login/home.jsp?admin=true

https://insecure-website.com/login/home.jsp?role=1

Cách tiếp cận này không an toàn vì người dùng có thể điều chỉnh giá trị và giành quyền truy cập vào chức năng nhạy cảm.

**Kiểm soát truy cập bị hỏng vì cấu hình sai**

Một vài ứng dụng kiểm soát truy cập ở lớp nền tảng bằng cách hạn chế truy cập những urls và phương thức https xác định. Ví dụ, một ứng dụng có thể cấu hình như sau:

DENY: POST, /admin/deleteUser, managers

Cấu hình này từ chối truy cập phương thức POST trên url /admin/deleteUser, rất nhiều thứ có thể sai trong tình huống này dẫn đến kiểm soát truy cập bị bỏ qua.

Một vài thư viện ứng dụng hỗ trợ những headers http có thể được dùng để ghi đè url trong yêu cầu trước như X-Original-URL và X-Rewrite-URL. Nếu 1 trang web kiểm soát truy cập nghiêm ngặt nhưng ứng dụng cho phép Url được ghi đè qua header của yêu cầu, sau đó nó có thể bỏ qua kiểm soát truy cập.

POST / HTTP/1.1

X-Original-URL: /admin/deleteUser

...

Một cuộc tấn công khác có thể liên quan đến phương thức http trong yêu cầu. Giao diện người dùng kiểm soát truy cập dựa trên url và phương thức http. Một vài trang web cho phép những phương thức http khác khi thực hiện 1 hành động. Nếu 1 kẻ tấn công có thể dùng GET (hay 1 phương thức khác) để thực hiện những hành động trên url bị giới hạn, sau đó họ có thể phá vỡ kiểm soát truy cập được thực thi trên lớp nền tảng.

**Kiểm soát truy cập bị hỏng do sự khác biệt về Url**

Khi chuyển hướng những yêu cầu, các trang web có sự khác nhau về sự nghiêm ngặt của đường dẫn. Ví dụ, nó có thể chấp nhận viết hoa, vì vậy 1 yêu cầu /ADMIN/DELETEUSER có thể được ánh xạ đến điểm vào /admin/deleteUser. Nếu kiểm soát truy cập nghiêm ngặt, nó sẽ xem chúng là 2 điểm vào khác nhau và không thực thi những hạn chế thích hợp.

Sự khác biệt tương tự cũng xảy ra khi nhà phát triển dùng thư viện Spring để thực thi useSuffixPatternMatch. Nó cho phép đường dẫn có phần mở rộng tập tin được ánh xạ với điểm vào không có. Nói cách khác, yêu cầu đến /admin/deleteUser.anything

sẽ giống với /admin/deleteUser.

Ở 1 hệ thống khác, bạn có thể gặp sự khác biệt ở /admin/deleteUser và /admin/deleteUser/. Trong trường hợp đó, bạn có thể bỏ qua kiểm soát truy cập dễ dàng bằng việc thêm / sau đường dẫn.

**Leo thang đặc quyền theo chiều ngang**

Leo thang đặc quyền theo chiều ngang sinh ra khi người dùng có thể truy cập tài nguyên của người khác. Ví dụ, 1 nhân viên chỉ có thể truy cập hồ sơ việc làm và bảng lương của chính họ, nhưng nếu cũng truy cập được của nhân viên khác thì nó là leo thang đặc quyền theo chiều ngang.

Leo thang đặc quyền theo chiều ngang có thể dùng những phương pháp khai thác giống với leo thang đặc quyền theo chiều dọc. Ví dụ, 1 người dùng có thể truy cập vào trang cá nhân của họ bằng url sau:

https://insecure-website.com/myaccount?id=123

Bây giờ nếu kẻ tấn công điều chỉnh tham số id thành 1 người dùng khác, sau đó hắn có thể truy cập trang cá nhân của họ.

Trong một vài ứng dụng, tham số khai thác có giá trị không dự đoán được. Ví dụ, thay vì số tăng dần, 1 ứng dụng có thể dùng mã định danh duy nhất (GUIDs) để xác minh người dùng. Ở đây, kẻ tấn công không thể đoán được mã định danh của họ. Tuy nhiên, GUIDs thuộc về người dùng khác có thể bị tiết lộ ở nơi mà người dùng có thể truy cập được như tin nhắn hay bình luận.

Trong một vài trường hợp, khi 1 ứng dụng phát hiện người dùng không được phép truy cập mã nguồn sẽ chuyển hướng đến trang đăng nhập. Tuy nhiên, phản hồi để chuyển hướng có thể chứa dữ liệu nhạy cảm nên vẫn tấn công được.

**Leo thang đặc quyền từ chiều ngang sang chiều dọc**

Leo thang đặc quyền theo chiều ngang có thể chuyển sang chiều dọc khi thâm nhập được người dùng có đặc quyền cao. Ví dụ, leo thang đặc quyền theo chiều ngang cho phép kẻ tấn công thu thập mật khẩu của người dùng khác. Nếu kẻ tấn công nhắm đến quản trị viên và thâm nhập tài khoản của họ, sau đó hắn có thể thêm, xóa tài khoản của người dùng khác.

Ví dụ, kẻ tấn công có thể truy cập trang cá nhân của người khác bằng url sau:

https://insecure-website.com/myaccount?id=456

Nếu người dùng là quản trị viên, hắn sẽ giành quyền truy cập trang quản trị. Trang đó có thể tiết lộ mật khẩu hay cách để truy cập chức năng nhạy cảm.

**Insecure direct object references (IDOR)**

**Khái niệm:**

IDOR là 1 loại lỗ hổng kiểm soát truy cập sinh ra khi ứng dụng sử dụng đầu vào do người dùng cung cấp để truy suất 1 đối tượng trực tiếp. Thuật ngữ IDOR phổ biến bởi sự xuất hiện của nó trong OWASP 2007 Top Ten. Tuy nhiên, nó chỉ là một trong rất nhiều lỗi thực thi của kiểm soát truy cập. Lỗ hổng IDOR thường gắn với leo thang đặc quyền theo chiều ngang, nhưng chúng cũng có thể sinh ra trong leo thang đặc quyền theo chiều dọc.

**Những ví dụ về IDOR**

Có rất nhiều lỗ hổng kiểm soát truy cập mà tham số do người dùng kiểm soát được dùng để truy suất tài nguyên hay chức năng trực tiếp.

**Lỗ hổng IDOR với truy suất trực tiếp đến các đối tượng trong cơ sở dữ liệu**

Xem xét 1 trang web sử dụng Url sau để truy cập trang cá nhân của khách hàng bằng việc truy suất thông tin từ cơ sở dữ liệu:

https://insecure-website.com/customer\_account?customer\_number=132355

Ở đây, customer\_number là 1 tham số trong câu truy vấn được thực hiện trong cơ sở dữ liệu. Nếu không được kiểm soát nghiêm ngặt, kẻ tấn công có thể điều chỉnh giá trị này để xem những bản ghi của khách hàng khác. Đây là 1 ví dụ về lỗ hổng IDOR dẫn đến leo thang đặc quyền theo chiều ngang.

**Lỗ hổng IDOR với truy suất trực tiếp tập tin tĩnh**

Lổ hổng IDOR thường xảy ra khi dữ liệu nhạy cảm được lưu ở tập tin tĩnh. Ví dụ, 1 trang web có thể lưu trữ tin nhắn trò chuyện vào đĩa bằng 1 tập tin tăng dần và cho phép người dùng truy suất nó:

https://insecure-website.com/static/12144.txt

Trong tình huống đó, kẻ tấn công có thể điều chỉnh tên tập tin để truy suất tin nhắn được tạo bởi người dùng khác.

**Lỗ hổng kiểm soát truy cập trong quy trình nhiều bước**

Nhiều trang web thực hiện các chức năng quan trọng qua một loạt các bước. Điều này thường được thực hiện khi cần thu thập nhiều đầu vào hay khi người dùng cần xem xét và xác nhận thông tin trước khi hành động được thực hiện. Ví dụ chức năng chỉnh sửa thông tin của quản trị có thể gồm những bước sau:

1. Tải biểu mẫu chứa thông tin người dùng chi tiết
2. Gửi những thay đổi
3. Xem xét sự thay đổi và xác nhận

Đôi khi, 1 trang web sẽ kiểm soát truy cập nghiêm ngặt đối với một số bước nhưng lại bỏ qua các bước khác. Ví dụ, kiểm soát truy cập được áp dụng đúng với bước 1 và 2, nhưng không áp dụng cho bước 3. Thực tế, trang web giả định rằng người dùng sẽ chỉ đến bước 3 nếu họ đã hoàn thành bước 1, được kiểm soát nghiêm ngặt. Ở đây, kẻ tấn công có thể truy cập trái phép đến chức năng bằng cách bỏ qua 2 bước đầu tiên và trực tiếp gửi yêu cầu đến bước 3 với những tham số cần thiết.

**Kiểm soát truy cập dựa trên Referer:**

Một số trang web kiểm soát truy cập dựa trên header Referer. Header Referer thường được thêm vào các yêu cầu của trình duyệt để chỉ ra trang mà từ đó yêu cầu được bắt đầu.

Giả sử 1 ứng dụng kiểm soát truy cập nghiêm ngặt ở trang quản trị /admin, nhưng đối với các trang phụ như /admin/deleteUser chỉ kiểm tra header Referer. Nếu header chứa url chính là /admin thì yêu cầu được cho phép.

Trong tình huống đó, vì header Referer có thể được kiểm soát bởi kẻ tấn công, họ có thể giả mạo những yêu cầu đến các trang phụ và giành quyền kiểm soát.

**Kiểm soát truy cập dựa trên vị trí**

Một số trang web kiểm soát truy cập dựa trên vị trí địa lý của người dùng. Điều này có thể áp dụng cho những ứng dụng ngân hàng hay dịch vụ truyền thông khi áp dụng luật pháp tiểu bang hay hạn chế kinh doanh. Kiểm soát truy cập này có thể bị phá vỡ bằng web proxies, VPNs hay tận dụng những kĩ thuật định vị phía máy khách.

**Cách ngăn chặn những lỗ hổng kiểm soát truy cập**

Những lỗ hổng kiểm soát truy cập có thể bị ngăn chặn bằng cách thực hiện phòng thủ chuyên sâu và áp dụng các nguyên tắc sau:

* Không bao giờ chỉ dựa vào sự che giấu để kiểm soát truy cập
* Ngoại trừ những tài nguyên được phép truy cập công khai, nên để mặc định là từ chối
* Nếu có thể, hãy dùng 1 cơ chế duy nhất trên toàn ứng dụng để kiểm soát truy cập.
* Bắt buộc các nhà phát triển phải khai báo quyền truy cập cho từng tài nguyên và nên để mặc định là từ chối.
* Thử xem kiểm soát truy cập có đang hoạt động như thiết kế không.