**Tìm hiểu về OWASP A8-A10 (2021) và kế hoạch phát triển demo**

**1. Sơ lược về các lỗ hổng OWASP A8-A10 (2021):**

* **A8: 2021 - Software and Data Integrity Failures (Lỗi toàn vẹn dữ liệu và phần mềm) - Insecure Deserialization:**
  + Mô tả: Lỗ hổng này liên quan đến việc không đảm bảo tính toàn vẹn của phần mềm và dữ liệu liên quan đến code và cơ sở hạ tầng, có thể do các vấn đề trong quá trình phát triển, triển khai hoặc cập nhật:
* Ứng dụng dựa vào các plugin, thư viện hoặc mô-đun không đáng tin cậy, không an toàn. Dẫn đến truy cập trái phép, thực thi các mã độc hại hoặc xâm nhập hệ thống.
* Tự động cập nhật các bản cập nhật mà không xác minh tính toàn vẹn đầy đủ và được áp dụng cho phiên bản trước đó.
  + Cách thức tấn công: Tấn công giả mạo các bản cập nhật, thay đổi cấu hình, hoặc tiêm mã độc vào các thành phần phần mềm.
  + Ví dụ: Cập nhật mà không cần kí, người dùng sẽ vô tình tải về những bản cập nhật chứa mã độc mà kẻ tấn công cố tình phát tán trên mạng để đánh cắp thông tin hay khai thác dữ liệu trong máy nạn nhân.
  + Cách ngăn chặn:
* Sử dụng chữ kí số để xác minh phần mềm.
* Đảm bảo các thư viện và phần phụ thuộc.
* Có công cụ bảo mật để kiểm tra độ an toàn phần mềm.
* Đảm bảo những dữ liệu chưa được ký hoặc chưa được mã hóa không gửi đến các máy khách không đáng tin cậy.
* **A9: 2021 - Security Logging and Monitoring Failures (Các lỗi theo dõi và ghi nhật ký bảo mật):**
  + Mô tả: Thiếu hoặc không hiệu quả trong việc ghi log và giám sát các hoạt động của ứng dụng, khiến việc phát hiện và điều tra các sự cố bảo mật trở nên khó khăn.
  + Cách thức tấn công: Tiêm mã độc (Injection), đánh cắp thông tin xác thực (phishing, brute-force, dictionary attacks), tấn công mạng…
  + Ví dụ: Một hãng hàng không lớn của Ấn Độ (Air India) đã bị vi phạm dữ liệu liên quan đến dữ liệu cá nhân của hàng triệu hành khách trong hơn mười năm, bao gồm cả dữ liệu hộ chiếu và thẻ tín dụng. Các tin tặc đã đột nhập vào hệ thống của hãng hàng không Ấn Độ và lấy đi dữ liệu cá nhân của khách hàng được lưu trữ từ tháng 8/2011 tới tháng 2/2021, bao gồm tên, ngày sinh, thông tin liên lạc, hộ chiếu và thông tin về vé. Ngoài ra, kẻ xấu còn tiếp cận được với dữ liệu thẻ khách hàng thân thiết, và thẻ tín dụng của hành khách.
  + Cách ngăn chặn:
* Đảm bảo các lỗi đăng nhập, kiểm soát truy cập và xác thực đầu vào phía máy chủ được ghi lại đủ để xác định các tài khoản đáng ngờ.
* Đảm bảo nhật kí được mã hóa chính xác tránh việc tiêm hoặc tấn công vào hệ thống ghi nhật kí hoặc giám sát.
* Đảm bảo các hành động tác động lớn được kiểm tra với các biện pháp kiểm soát tính toàn vẹn để ngăn chặn việc giả mạo hoặc xóa, chẳng hạn như bảng cơ sở dữ liệu chỉ được thêm vào.
* Các nhóm DevSecOps nên thiết lập giám sát và cảnh báo hiệu quả để các hoạt động đáng ngờ được phát hiện và phản hồi nhanh chóng.
* **A10: 2021 - Server-side Request Forgery (SSRF - Giả mạo yêu cầu phía máy chủ):**
  + Mô tả: SSRF xảy ra bất cứ khi nào khi ứng dụng web đang tìm nạp tài nguyên từ xa mà không xác thực URL do người dùng cung cấp. Nó cho phép kẻ tấn công ép ứng dụng gửi một yêu cầu đến một điểm đích không mong muốn, ngay cả khi được bảo vệ bởi tường lửa.
  + Cách thức tấn công: Kẻ tấn công cung cấp một URL hoặc địa chỉ IP đặc biệt, khiến ứng dụng web gửi yêu cầu đến một máy chủ nội bộ hoặc một dịch vụ không được phép truy cập. Khi đó máy chủ bị tấn công sẽ thực hiện yêu cầu này, cho phép kẻ tấn công truy cập vào các hệ thống nội bộ hoặc thực hiện các hành động khác.
  + Ví dụ: Ứng dụng tải xuống hình ảnh **-** Nếu ứng dụng cho phép người dùng nhập URL hình ảnh tùy ý, kẻ tấn công có thể nhập một URL chỉ vào một dịch vụ nội bộ, chẳng hạn như một máy chủ lưu trữ mã nguồn.
  + Cách ngăn chặn:
* Lớp mạng:
* Phân đoạn chức năng truy cập tài nguyên từ xa trong các mạng riêng biệt để giảm tác động của SSRF.
* Thực thi các chính sách tường lửa “từ chối theo mặc định” hoặc các quy tắc kiểm soát truy cập mạng để chặn tất cả trừ lưu lượng mạng nội bộ thiết yếu.
* Lớp ứng dụng:
* Làm sạch và xác thực tất cả dữ liệu đầu vào do người dùng cung cấp.
* Thực thi lược đồ URL, cổng là điểm đến với danh sách cho phép xác thực.
* Tắt chuyển hướng HTTP.

**2. Lập kế hoạch phát triển demo code:**

**Công nghệ và ngôn ngữ:**

* PHP: Laravel, Symfony
* Python: Django, Flask
* Node.js: Express.js
* Java: Spring Boot

**Ứng dụng web đơn giản:**

* Ứng dụng quản lý tài liệu: Minh họa lỗ hổng A8 (tải xuống file từ nguồn không đáng tin cậy)
* Ứng dụng blog: Minh họa lỗ hổng A9 (thiếu log khi đăng nhập)
* Ứng dụng thanh toán: Minh họa lỗ hổng A10 (SSRF để truy cập các dịch vụ nội bộ)

**Môi trường phát triển:**

* IDE: Visual Studio Code, PyCharm, IntelliJ IDEA
* Máy chủ web: Apache, Nginx
* Cơ sở dữ liệu: MySQL (PHP), PostgreSQL
* Công cụ bảo mật: OWASP ZAP, Burp Suite

**3. Chuẩn bị môi trường phát triển:**

* Cài đặt: Cài đặt các phần mềm cần thiết (ngôn ngữ lập trình, framework, cơ sở dữ liệu, công cụ bảo mật).
* Cấu hình: Cấu hình môi trường làm việc, tạo các dự án mới.
* Tìm hiểu: Đọc tài liệu, xem hướng dẫn về các công cụ và framework.
* Thực hành: Bắt đầu xây dựng các ứng dụng demo đơn giản.

**Các bước thực hiện chi tiết:**

1. Chọn một ứng dụng demo: Ví dụ, chọn ứng dụng quản lý tài liệu để minh họa lỗ hổng A8.
2. Xây dựng chức năng cốt lõi: Tạo chức năng cho phép người dùng tải lên và tải xuống các tệp.
3. Tạo lỗ hổng: Không kiểm tra tính toàn vẹn của tệp tải xuống, cho phép người dùng nhập bất kỳ URL nào để tải xuống.
4. Kiểm tra lỗ hổng: Sử dụng OWASP ZAP hoặc Burp Suite để tìm kiếm và khai thác lỗ hổng.
5. Vá lỗ hổng: Thực hiện kiểm tra tính toàn vẹn của tệp tải xuống trước khi lưu trữ.
6. Lặp lại: Thực hiện tương tự với các ứng dụng demo khác và các lỗ hổng còn lại.

**Tài liệu tham khảo:**

* OWASP Top 10 2021:<https://owasp.org/Top10/>
* Tài liệu chính thức của các framework: Laravel, Django, Express.js, ...
* Các bài viết, blog về bảo mật ứng dụng web: Medium, HackerOne, ...