Quy trình Pentest Web:

1. Khái niệm và mục đích của pentest web, sản phẩm thiết bị, ứng dụng: Pentest web (kiểm thử xâm nhập web) là quá trình đánh giá bảo mật của các ứng dụng web, trang web, hệ thống web, hay các ứng dụng dựa trên web nhằm xác định các lỗ hổng bảo mật có thể bị tấn công và tiềm năng dẫn đến việc xâm nhập hoặc phá hoại. Mục đích chính của pentest web là tìm ra các lỗ hổng bảo mật để giúp ngăn chặn các tấn công và cải thiện mức độ bảo mật của ứng dụng web.
2. Các bước trong quy trình pentest web:

a. Thu thập thông tin: Thu thập thông tin về mục tiêu, công nghệ sử dụng, cấu trúc trang web, tìm kiếm các thông tin công khai, cơ sở dữ liệu, tên miền, địa chỉ IP, v.v.

b. Phân tích và lập kế hoạch: Đánh giá rủi ro, xác định các kỹ thuật pentest phù hợp, thiết kế kế hoạch kiểm thử.

c. Quét lỗ hổng: Sử dụng các công cụ tự động để quét lỗ hổng bảo mật, ví dụ: OWASP ZAP, Burp Suite, Nikto, v.v.

d. Phân tích lỗ hổng: Kiểm tra và đánh giá kết quả quét để xác định các lỗ hổng thực sự.

e. Kỹ thuật thủ công: Tiến hành kiểm tra thủ công, tìm kiếm các lỗ hổng phức tạp mà các công cụ tự động không phát hiện được.

f. Khai thác lỗ hổng: Nếu tìm thấy lỗ hổng có thể khai thác, thử thực hiện tấn công để xác minh tính khả thi và mức độ tổn thất.

g. Truy cập và giữ lâu dài: Nếu có thể, xác minh tính năng truy cập không ủy quyền và giữ quyền truy cập lâu dài vào hệ thống.

h. Báo cáo kết quả: Tổng hợp kết quả kiểm thử và lỗ hổng tìm thấy, đưa ra đánh giá rủi ro và gợi ý biện pháp khắc phục.

3. Các công cụ và kỹ thuật sử dụng trong pentest web, thiết bị, sản phẩm:

a. Công cụ quét lỗ hổng tự động: OWASP ZAP, Burp Suite, Nikto, Acunetix, Nessus, OpenVAS, v.v.

b. Công cụ kiểm tra lỗ hổng thủ công: Kali Linux (cung cấp nhiều công cụ bảo mật), Metasploit Framework, Nmap, SQLMap, v.v.

c. Proxy: Burp Suite, OWASP ZAP (cho kiểm tra các yêu cầu và phản hồi giữa máy khách và máy chủ).

d. Công cụ kiểm tra mã nguồn: SonarQube, Checkmarx, Fortify, v.v.

e. Công cụ quản lý và phân tích kết quả: JIRA, Mantis, Redmine, v.v.

4. Các kỹ thuật khai thác lỗ hổng phổ biến:

a. SQL Injection: Tấn công vào cơ sở dữ liệu bằng cách nhập các câu truy vấn SQL độc hại thông qua các trường dữ liệu đầu vào.

b. Cross-Site Scripting (XSS): Chèn mã độc JavaScript vào trang web để thực hiện các tấn công đánh cắp cookie và thông tin người dùng.

c. Cross-Site Request Forgery (CSRF): Thực hiện các hành động không được ủy quyền từ người dùng bằng cách lừa họ kích hoạt các yêu cầu HTTP giả mạo.

d. Remote Code Execution (RCE): Tấn công vào hệ thống từ xa thông qua lỗ hổng trong phần mềm hoặc ứng dụng.

e. File Inclusion: Thực hiện việc chèn các tập tin độc hại hoặc kiểm soát các tệp cần thiết trên máy chủ.

5. Quy trình kiểm tra và báo cáo kết quả:

a. Tổng hợp kết quả: Tạo báo cáo chi tiết về các lỗ hổng đã tìm thấy, bao gồm cả thông tin về cách tấn công và khả năng tác động.

b. Đánh giá mức độ nghiêm trọng: Xác định mức độ nghiêm trọng của từng lỗ hổng để ưu tiên biện pháp khắc phục.

c. Gợi ý biện pháp khắc phục: Đưa ra các gợi ý và hướng dẫn để cải thiện bảo mật của ứng dụng web.

d. Đối chiếu tiêu chuẩn bảo mật: So sánh kết quả với các tiêu chuẩn bảo mật như OWASP Top 10 để đánh giá mức độ tuân thủ và sự phù hợp.

e. Trình bày báo cáo: Trình bày báo cáo kết quả cho nhóm phát triển hoặc quản lý để họ có thể triển khai các biện pháp khắc phục.