Chương 1

GIỚI THIỆU BẢO MẬT WEBSITE

TS. Phan Thuong Cang
CIT - CTU
2019

Bảo mật website

- Bảo mật website là hoạt động phòng ngừa, phát hiện, bảo vệ và ngăn chặn trước nguy cơ website bị hacker tấn công nhằm duy trì một website ổn định và an toàn. Đây là hoạt động quan trọng, cần thiết và thường xuyên trong quá trình sử dụng và vận hành website.
- Bảo mật website là bất kỳ hành động hoặc ứng dụng nào thực hiện việc đảm bảo dữ liệu website không bị phơi bày trước tội phạm mạng hoặc để ngăn chặn việc khai thác website dưới bất kỳ hình thức nào.

Ứng dụng web

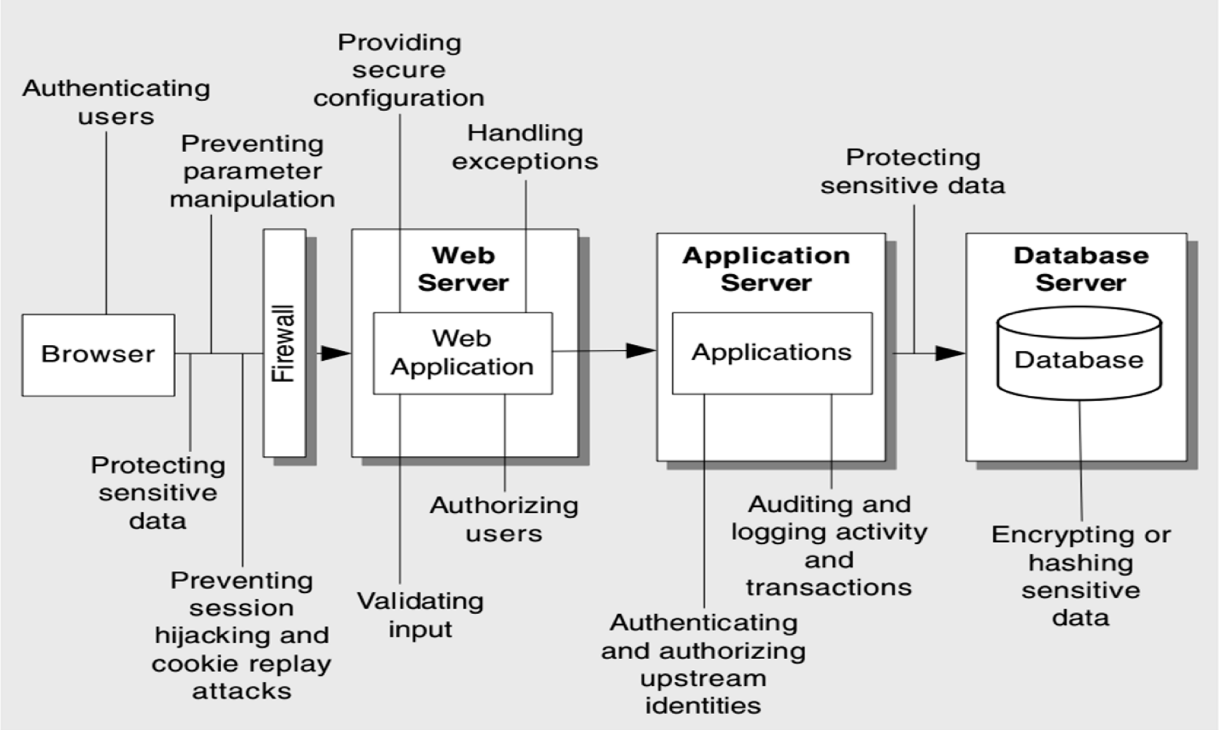


Figure 4.1
Web application design issues

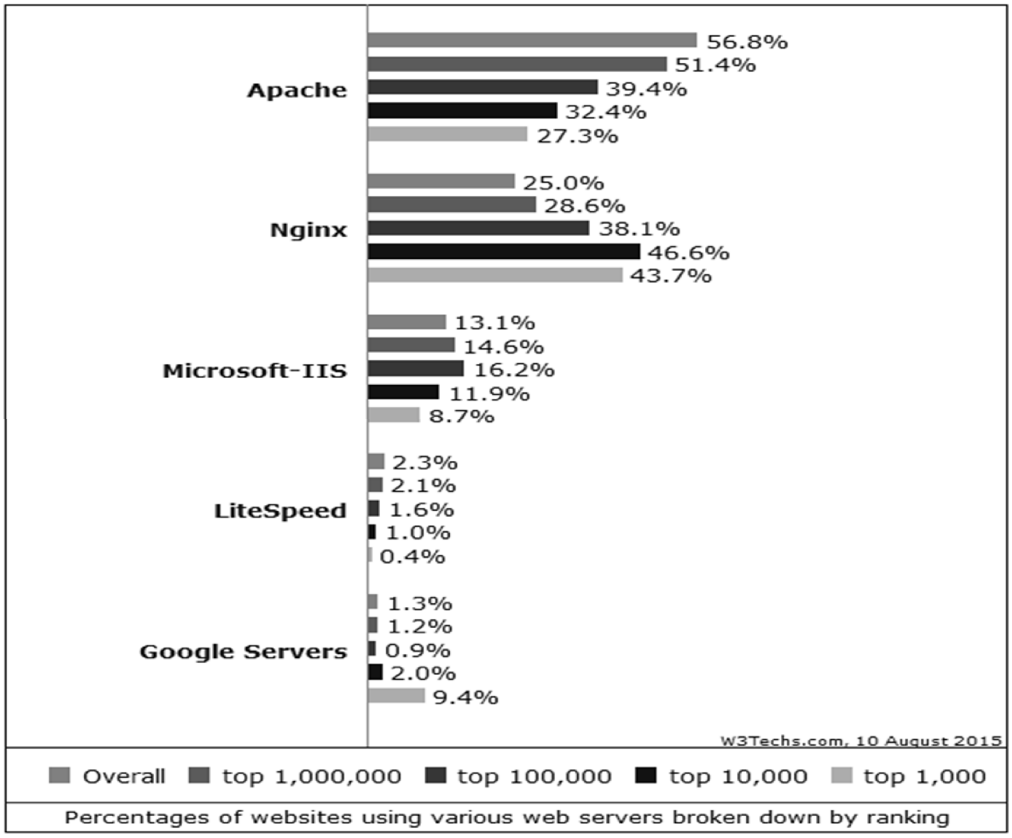
Các vấn đề bảo mật ứng dụng web

- Phía trình duyệt web:
  - Hacker có thể giả mạo với bất kỳ dữ liệu nào được lưu trữ trên trang Web của client.
  - Dữ liệu gửi đi từ client đều có thể bị đọc và giả mạo cùng với nó.
  - Hiển thị các thông báo lỗi trả về từ server có thể được dùng để tìm hiểu về cấu trúc và cấu hình của máy chủ.

- Phía server:
  - Cấu hình sai (cả web server và hệ điều hành)
  - Các phần mềm chứa mã độc, lỗ hổng, phiên bản cũ
  - Ứng dụng web có các xử lý chưa quan tâm bảo mật

Các vấn đề bảo mật ứng dụng web

- Phía server:



Các vấn đề bảo mật ứng dụng web

- Phía đường truyền mạng:
  - Mọi dữ liệu đến server bằng HTTP đều có thể bị xem và bị giả mạo bởi bên thứ ba.
  - SSL cung cấp khả năng mật mã ngăn cản xem hoặc giả mạo với dữ liệu giữa nguồn chuyển dữ liệu và máy chủ, nhưng không thể ngăn cản sự giả mạo dữ liệu tại nguồn của nó.

Đánh giá độ an toàn website

- Bốn giai đoạn để đánh giá mức bảo mật:
  - **Phân loại tài sản:** là cơ sở cho tất cả các công việc, điều này sẽ làm rõ mục tiêu mà bạn muốn bảo vệ.
  - **Phân tích mối đe doạ:** xác định được mối nguy hiểm đến từ đâu.
  - **Phân tích rủi ro:** Rủi ro bao gồm các yếu tố:

    Nguy cơ = Xác suất * Tiềm năng thiệt hại
  - **Thiết kế các chương trình an ninh:** các chương trình bảo mật phải có khả năng chống lại các mối đe dọa hiệu quả nhưng không nên can thiệp vào các qui trình nghiệp vụ bình thường.

Một số giải pháp bảo mật website

- Malware scan
- Malware removal
- Manual malware and hack removal
- File change monitoring
- Blacklist/spam monitoring
- Blacklist removal
- Security monitoring
- Advanced DDoS mitigation
- Web Application Firewall (WAF)
- Content Delivery Network (CDN)
- Site Seal

OWASP - Công bố 10 lỗ hổng hàng đầu trong các ứng dụng web (2017-nay)

OWASP Top 10 – 2013 (Previous)	OWASP Top 10 – 2017 (New)
A1 – Injection	A1 – Injection
A2 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References - Merged with A7	A4 – Broken Access Control (Original category in 2003/2004)
A5 – Security Misconfiguration	A5 – Security Misconfiguration
A6 – Sensitive Data Exposure	A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control - Merged with A4	A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities	A9 – Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards - Dropped	A10 – Underprotected APIs (NEW)

A1 – Injection (Lỗi nhúng mã/chèn mã)

- Lỗi nhúng mã là tập hợp các lỗ hổng bảo mật xảy ra khi dữ liệu đáng ngờ được chèn vào ứng dụng dưới dạng lệnh hay truy vấn.

- Các cuộc tấn công mã nhúng như: SQL, OS, XXE và LDAP. Trong đó, phổ biến nhất là tấn công SQL, còn được biết đến là SQLi. Một cuộc tấn công SQLi thành công khi đoạn mã sai được gửi đến dữ liệu máy chủ và tiếp cận với hệ thống dữ liệu.

A2 – Broken Authentication and Session Management

- Khi các chức năng của ứng dụng thực hiện không chính xác và bảo mật kém, tin tặc có thể dễ dàng xâm nhập, ăn cắp thông tin tài khoản, mật khẩu và khai thác các lỗ hổng khác bằng cách sử dụng các tài khoản/chứng chỉ đã đánh cắp.

A3 – Cross-site Scripting (XSS)

- XSS cho phép hacker đưa các kịch bản phía máy khách (javascript) vào các trang web và trong nhiều trường hợp, hacker có thể sử dụng các công cụ kiểm soát truy cập của họ.
- Chúng thực hiện bằng cách đánh lừa trình duyệt chấp nhận dữ liệu từ một nguồn không đáng tin cậy.

A4 – Broken Access Control

- Khi người dùng bị hạn chế kiểm soát truy cập, tin tặc có thể khai thác và truy cập các chức năng hoặc dữ liệu trái phép.
- Nguyên nhân lỗi kiểm soát truy cập xảy ra do các nhà phát triển thường bị sai (hoặc không quan tâm) trong việc kiểm soát truy cập phù hợp với các quy tắc đặt ra.

A5 – Security Misconfiguration (cấu hình bảo mật sai)

- Do cấu hình an ninh lỏng lẻo tại các tầng kiến trúc của web như nền tảng, framework, máy chủ, cơ sở dữ liệu và mã tùy chỉnh nên tin tặc có thể khai thác tấn công và có quyền truy cập dữ liệu.

A6 – Sensitive Data Exposure (Lộ dữ liệu nhạy cảm)

- Việc tiếp xúc dữ liệu nhạy cảm xảy ra khi các kiểm soát bảo mật, chẳng hạn như HTTPS không được thực hiện chính xác và để lại lỗ hổng, giúp tin tặc có thể ăn cắp thông tin tài khoản, mật khẩu, địa chỉ hay bất cứ thông tin có giá trị nào khác.

- Các ứng dụng cần đảm bảo truy cập được xác thực và dữ liệu đã được mã hóa. Nếu không sẽ dẫn đến việc vi phạm quyền riêng tư ở quy mô lớn.

A7 – Insufficient Attack Protection

- Các ứng dụng và API có thể chứa đựng lỗ hỏng,
- Việc ngăn chặn và phản hồi các cuộc tấn công, các chuyên gia sử dụng các phương pháp kiểm thử bảo mật, đánh giá tính dễ tổn thương và sử dụng WAF hay RASP để phát hiện và vá lỗi nhanh.

A8 – Cross-Site Request Forgery (CSRF)

- Giả mạo yêu cầu chéo trang web (CSRF)

- Một cuộc tấn công CSRF sẽ yêu cầu người dùng xác thực trên một ứng dụng web, sau đó, tin tặc lợi dụng điều đó để ăn cắp thông tin tài khoản người dùng và cướp quyền kiểm soát trình duyệt của người dùng.

A9 – Using Components with Known Vulnerabilities

- Việc sử dụng mà không kiểm duyệt các thư viện, plug-in, ứng dụng… đã tồn tại lỗ hổng (thường là mã nguồn mở). Tin tặc sẽ lợi dụng từ đây để tấn công hệ thống thông qua phương pháp SQLi và XSS.

A10 – Underprotected APIs

- Việc sử dụng mà không kiểm duyệt các thư viện, plug-in, ứng dụng… đã tồn tại lỗ hổng (thường là mã nguồn mở). Tin tặc sẽ lợi dụng từ đây để tấn công hệ thống thông qua phương pháp SQLi và XSS.

- Các API chứa các lỗ hổng khiến ứng dụng rất dễ bị tấn công. API cũng chứa nhiều giao thức phức tạp như SOAP/XML, REST/JSON, RPC và GWT mà kiểm thử bảo mật không thể kiểm tra thành công, khiến các API trở thành điểm mù quan trọng trong các tổ chức đang sử dụng chúng.