NT521 - Lập trình an toàn & Khai thác lỗ hổng phần mềm

Giới thiệu môn học

- Mã môn học: **NT521**
- Tên môn học: **Lập trình an toàn và khai thác lỗ hổng phần mềm**
- Số tín chỉ: **04**
- Thời gian: **15 buổi x 3 tiết**

Giới thiệu môn học

- **Pham Van Hau** – *Head of Information Security Department - Faculty of Computer Networks and Communication*
  - **Email: haupv@uit.edu.vn**
- **Phan The Duy** – *Information Security Researcher @ UIT InSecLab*
  - **Email: duypt@uit.edu.vn**
- **Do Thi Thu Hien -** *Information Security Researcher @ UIT InSecLab*
  - **Email: hiendtt@uit.edu.vn**
- **Nguyen Huu Quyen** (Teaching Assistant) - *Information Security Researcher @ UIT InSecLab*
  - **Email: quyennh@uit.edu.vn**




Khảo sát kiến thức

- SDLC
- SSDLC
- Shift Left
- Secure Design
- Threat model
- DevOps/DevSecOps
- Software Programming
- Secure Coding
- Software Vulnerabilities
- Software Testing
- Static application security testing (SAST)
- Dynamic application security testing (DAST)
- Symbolic Execution
- Fuzzing Testing
- Exploitation
- Zero-day
- PoC

Khảo sát kiến thức:
An toàn phần mềm là gì?

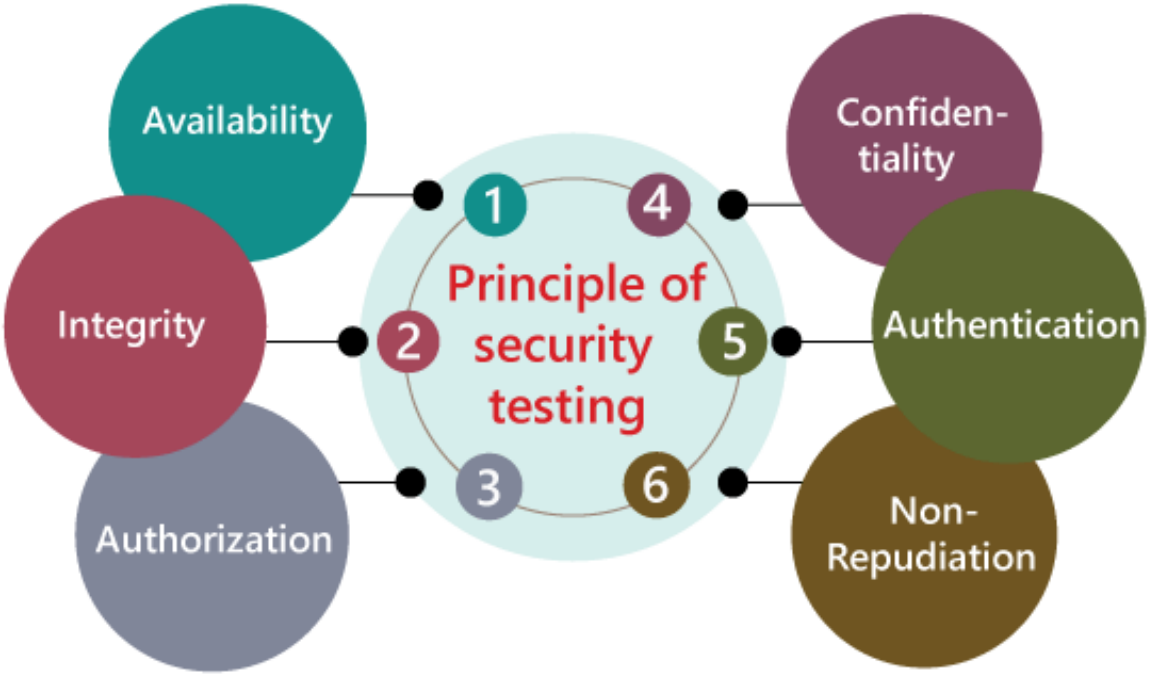
- **An toàn phần mềm** là một trong những khía cạnh chính để **xây dựng hệ thống/phần mềm tin cậy (trustworthy/reliability).**
- An toàn phần mềm là nguyên lý triển khai **các cơ chế bảo mật trong việc thiết kế, xây dựng, và kiểm thử** để *giúp phần mềm duy trì chức năng (hoặc khả năng) chống lại các cuộc tấn công*.
  - Để đảm bảo khả năng chống lại các cuộc tấn công nguy hiểm của phần mềm trước khi đưa ra thị trường, một phần mềm phải trải qua quá trình **thiết kế, lập trình, kiểm tra an toàn - bảo mật** theo tiêu chuẩn.

*"Software security is **an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks**".*

Khảo sát kiến thức

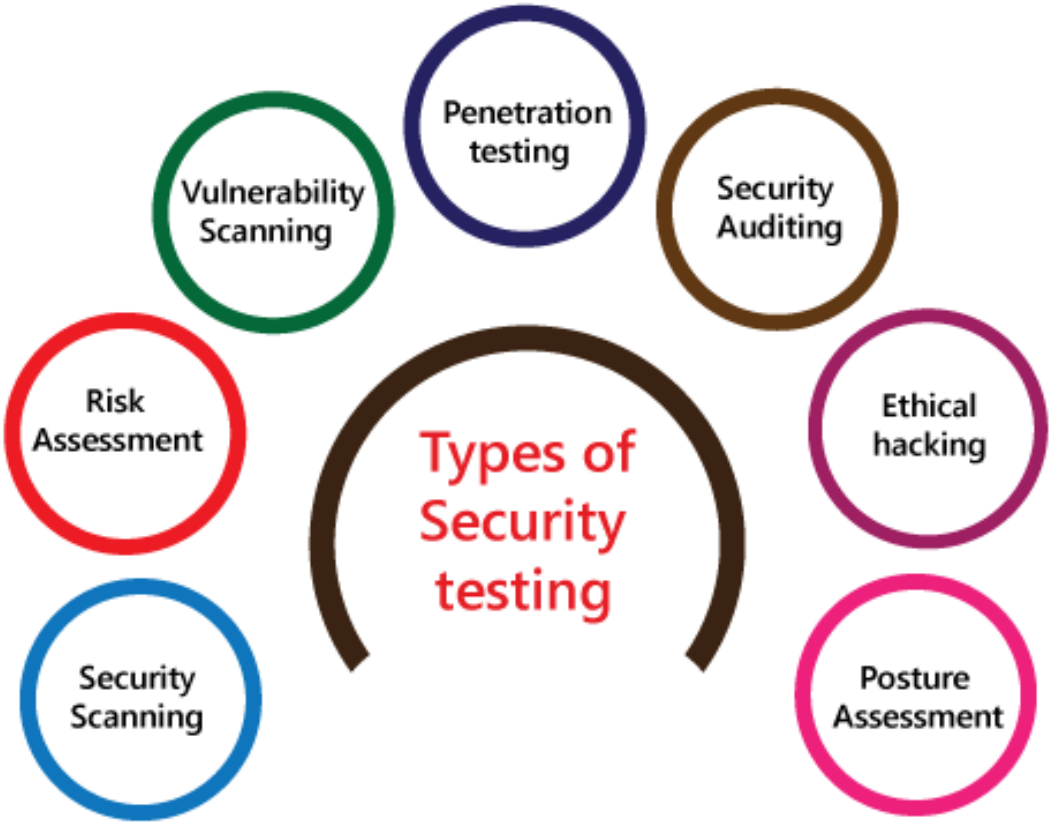
Quy trình phát triển phần mềm - Software Development Life Cycle (SDLC)

Khảo sát kiến thức



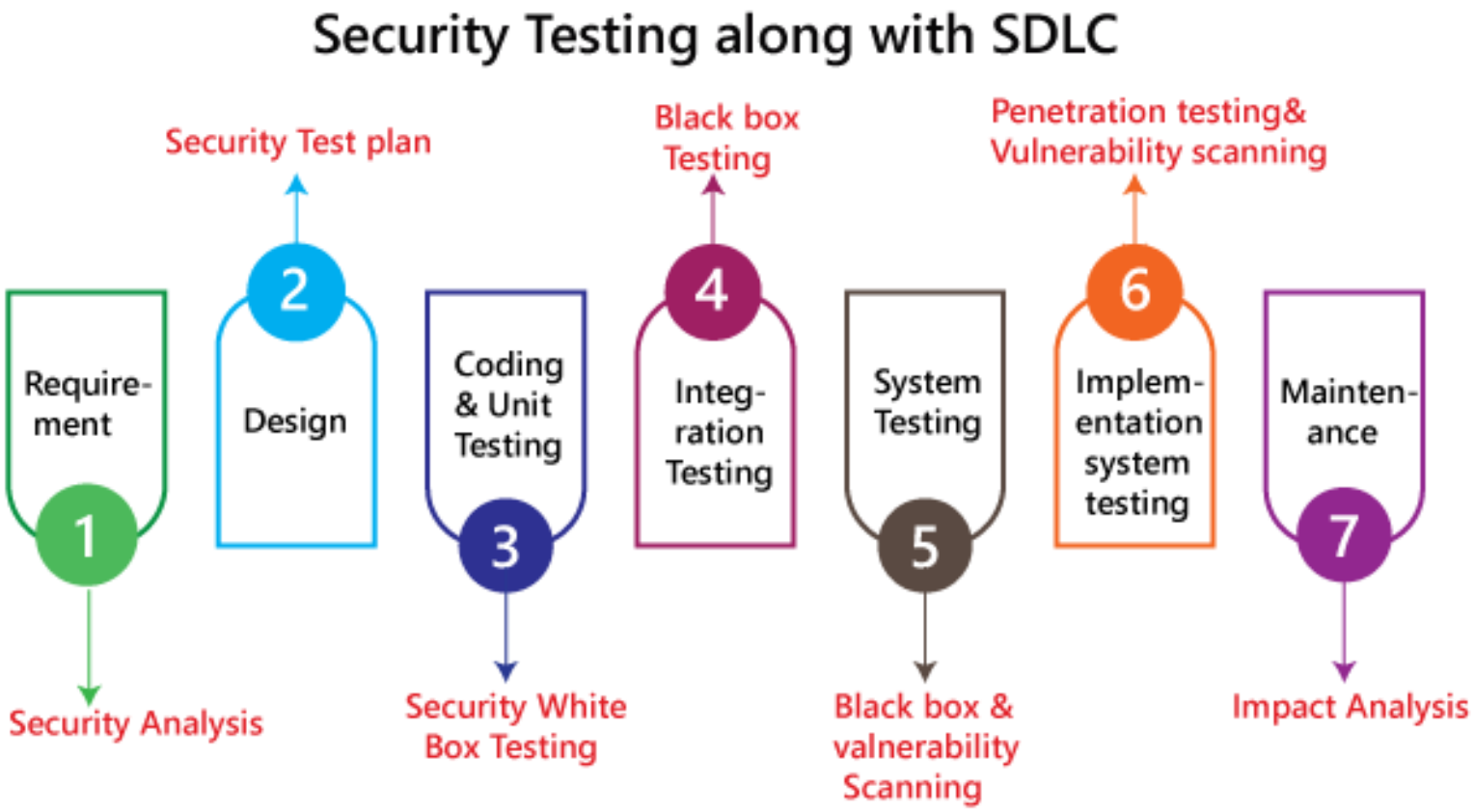
Các nguyên tắc/khía cạnh trong kiểm thử phần mềm

Khảo sát kiến thức



Các dạng kiểm thử trong kiểm thử phần mềm

Khảo sát kiến thức



SSDLC với nguyên tắc “Dịch Trái” (Shift Left)

Khảo sát kiến thức

Chu kỳ SSDLC được thực hiện lặp đi lặp lại

Mục tiêu

- Mục tiêu:
  - Trang bị những kiến thức cơ bản về bảo mật phần mềm, quy trình thiết kế, lập trình, kiểm thử phần mềm an toàn.
  - Trang bị kiến thức về các lỗ hổng phần mềm phổ biến: cách khai thác và phòng tránh

Nội dung chính

- **Qui trình phát triển phần mềm - Software Development Lifecycle (SDLC)**
  - Các mô hình SDLC truyền thống
  - Các mô hình Agile
- **Qui trình phát triển phần mềm an toàn - Secure SDLC (SSDLC):**
  - Designing and Building Secure Software
  - Shift Left
  - Secure Programming & Program Analysis
  - DevSecOps
- **Khai thác lỗ hổng phần mềm (Exploiting Software Vulnerabilities):**
  - Common Vulnerabilities
  - Exploitation
  - Repairing
- **Hướng nghiên cứu trong lĩnh vực An toàn phần mềm (Software Security)**

Khung chương trình

- ***Buổi 01: Giới thiệu môn học***
  - Giới thiệu môn học, Yêu cầu và qui định của môn học

Phần I: Thiết kế và phát triển phần mềm an toàn

- ***Buổi 02:*** SDLC – Tổng quan về qui trình phát triển phần mềm
- ***Buổi 03:*** *SSDLC* – Qui trình thiết kế và phát triển phần mềm an toàn:
  - ✓ Threat model & Security requirements
  - ✓ Secure Design
  - ✓ Common Flaws in Programming
- ***Buổi 04:*** DevSecOps
- ***Buổi 05:*** Phân tích và kiểm thử chương trình phần mềm
  - Program Analysis: Static Code Analysis
  - Testing/Coverage Testing
- ***Buổi 06:*** Kỹ thuật Fuzzing trong Kiểm thử phần mềm
  - Cơ bản về Fuzzing
  - Ứng dụng fuzzing trong kiểm thử

Khung chương trình

Phần II - Khai thác lỗ hổng phần mềm

- **Buổi 07: Cơ bản về khai thác lỗ hổng phần mềm**
  - *Portable Execution* + *Compiler* + Shellcode + Code Injection
  - OverFlow
  - Format String
- **Buổi 08:** Arc-injection Attack, Off-by-one
- **Buổi 09:**
  - Address Space Layout Randomization (ASLR);
  - DEP + ROP
- **Buổi 10:** Heap Exploitation
- **Buổi 11:** Future Direction of Secure Software & Exploitation
  - *Một số hướng nghiên cứu về An toàn phần mềm (Software Security); tự động hoá Khai thác lỗ hổng phần mềm*
- **Buổi 12-13-14:** Báo cáo Đồ án môn học
- **Buổi 15:** Ôn tập

Đánh giá

25% quá trình:

I. Bài tập + Các bài tập CTF (40%)

II. Đồ án môn học (60%)

25% thực hành:

I. 06 bài Lab

II. Các bài thực hành CTF

50% cuối kì:

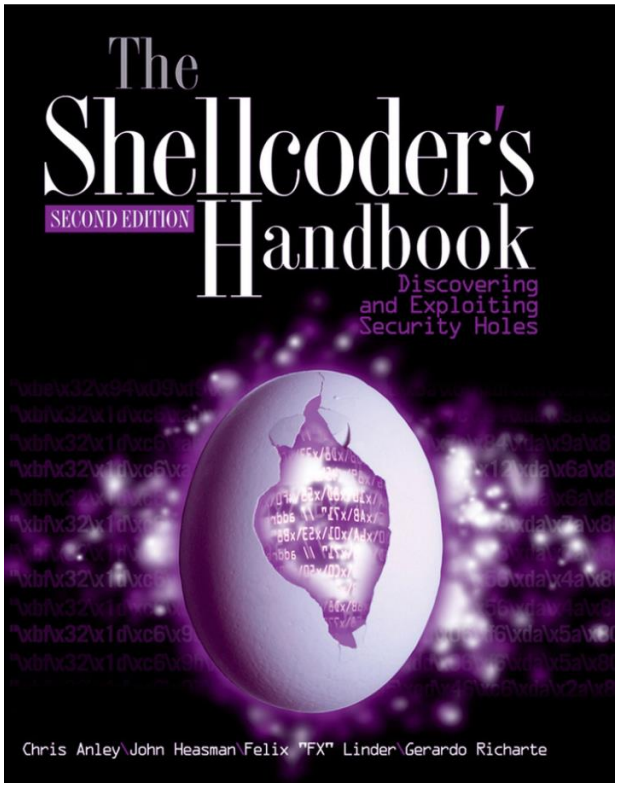
I. Đồ án: câu hỏi về Bài tập, đề tài Đồ án môn học (30%)

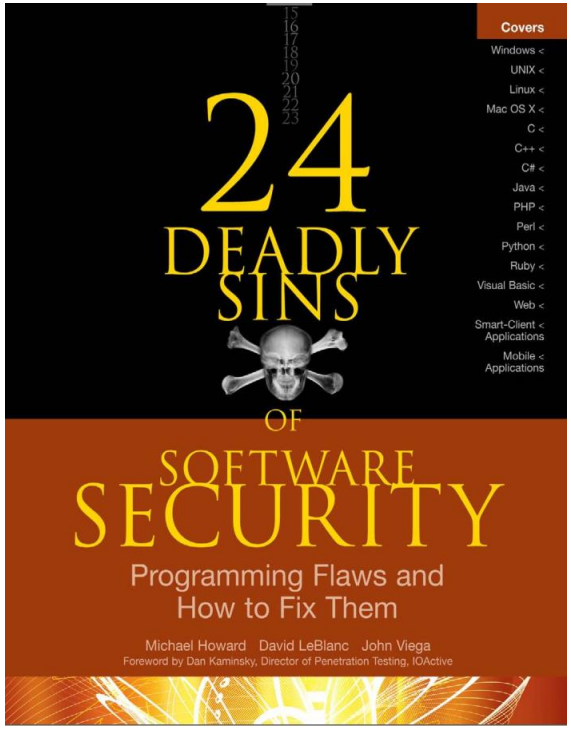
II. Thi lý thuyết (trắc nghiệm và tự luận): 70%

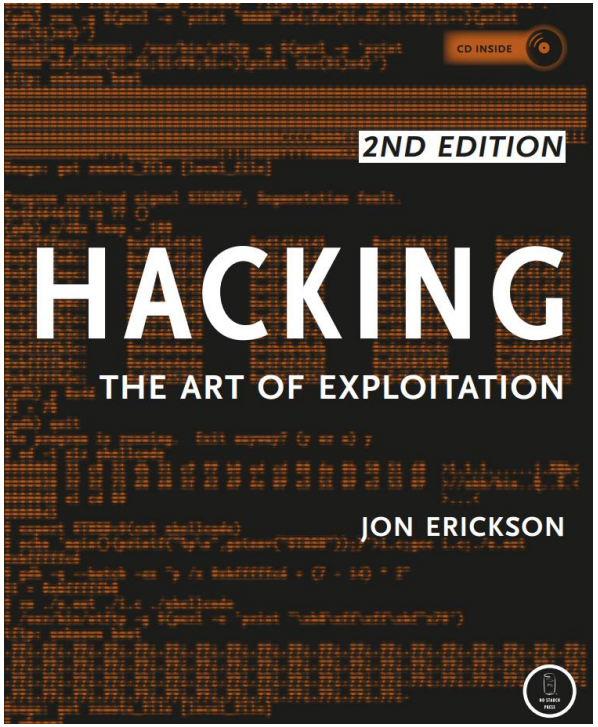
Qui định học tập

- Tìm hiểu, thảo luận về nội dung bài giảng, bài tập
- Đến lớp đúng giờ, không làm việc riêng trong giờ học.
- **Đồ án môn học:** tổng hợp các kiến thức môn học
  - Giao đề tài Đồ án vào **buổi học số 03**
  - Báo cáo đề tài Đồ án + (Bài tập) bắt đầu từ Buổi 12.
- Nhóm: **03** thành viên/nhóm
- Qui định khi làm việc nhóm:
  - Không ghi đầy đủ thông tin nhóm → 0đ
  - Sao chép bài → 0đ
  - Điểm của thành viên trong nhóm không phải là điểm chung của nhóm

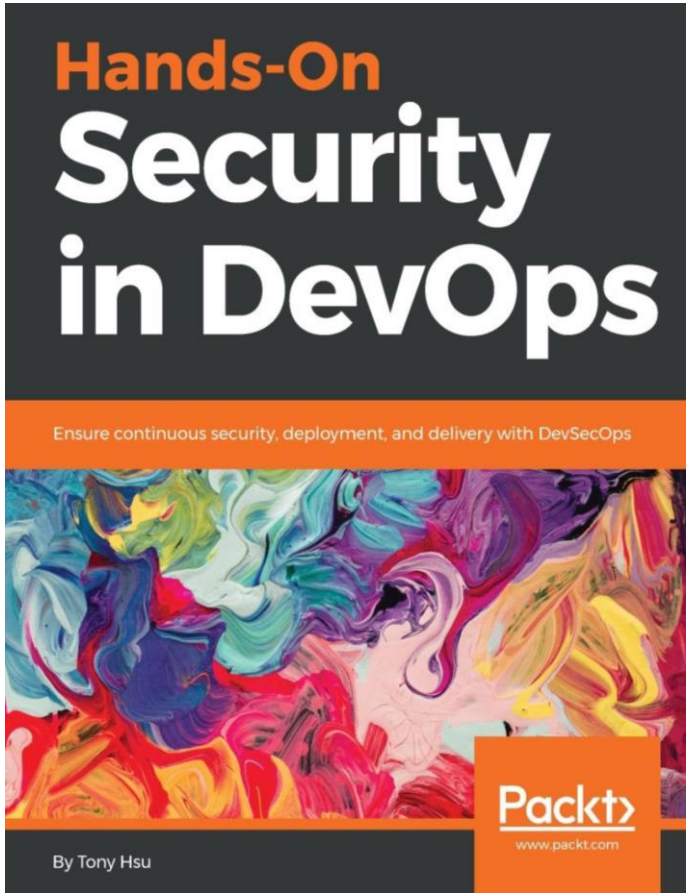
Tài liệu tham khảo

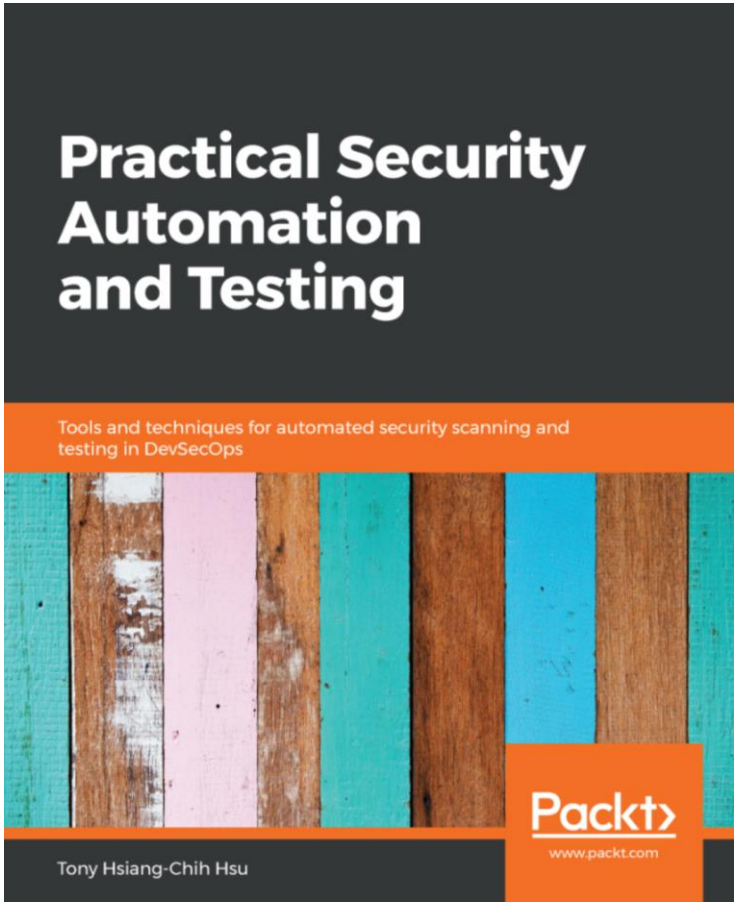




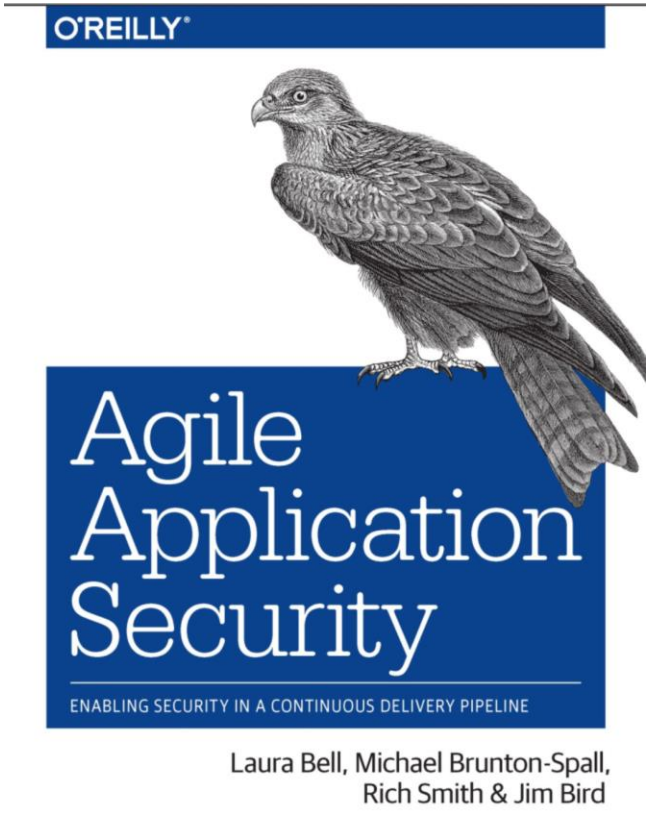


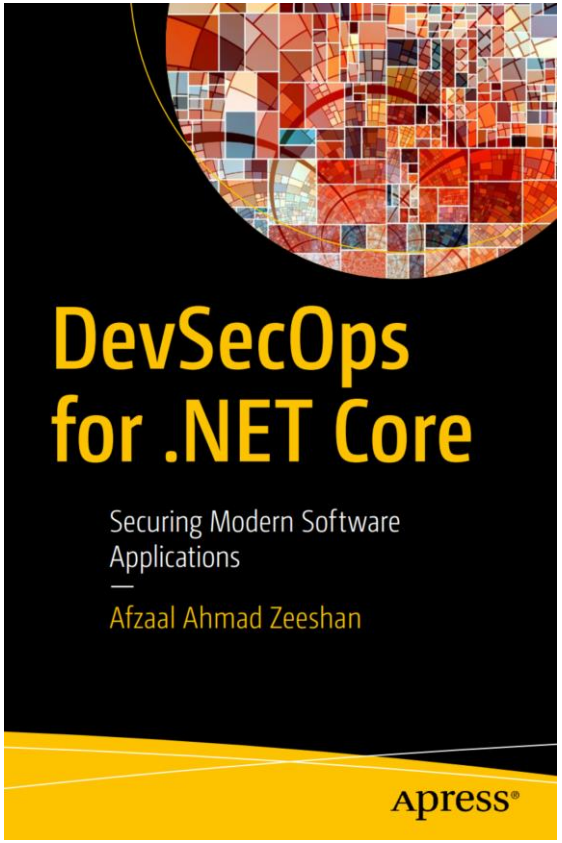
Tài liệu tham khảo

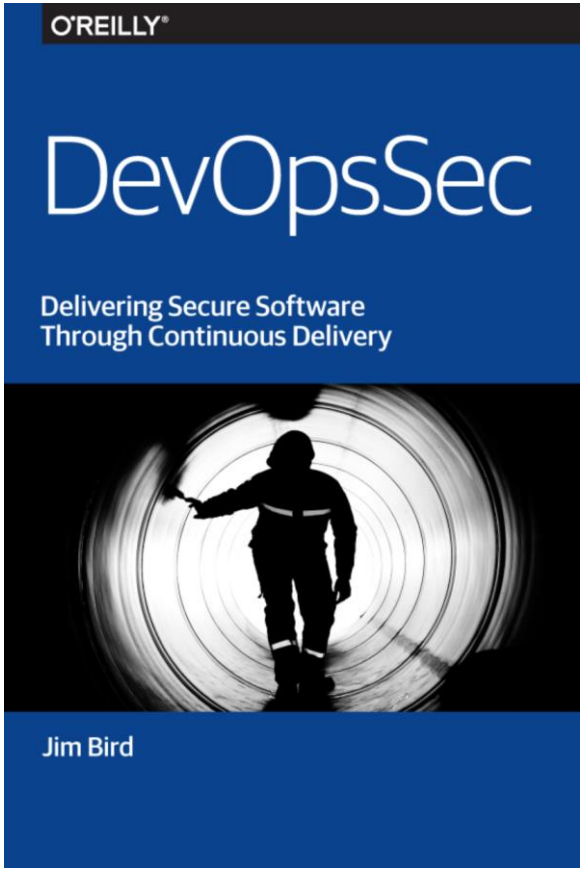




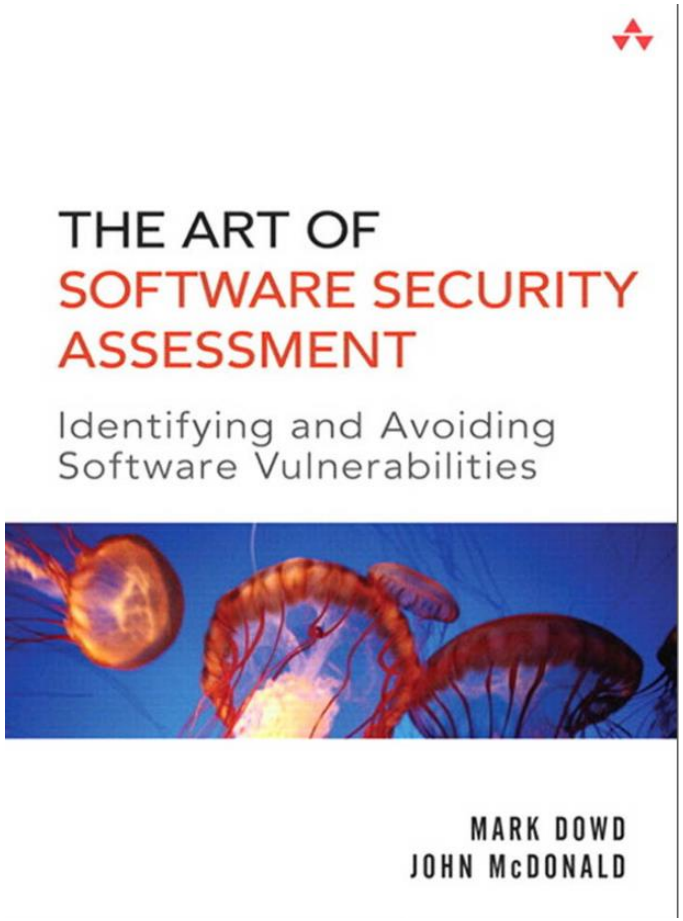
Tài liệu tham khảo

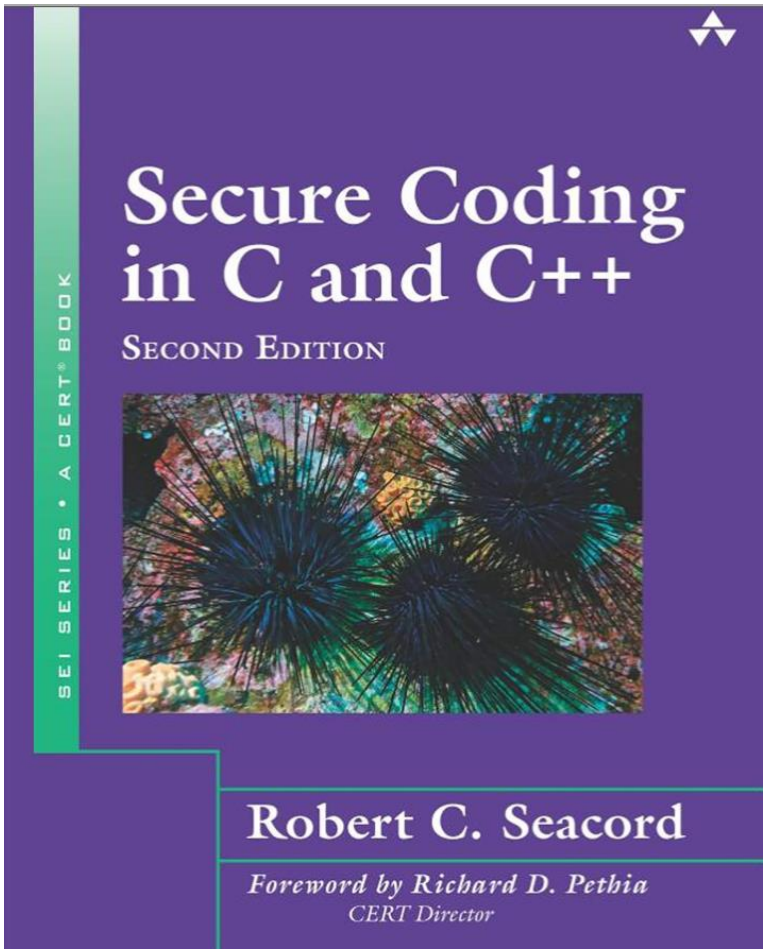




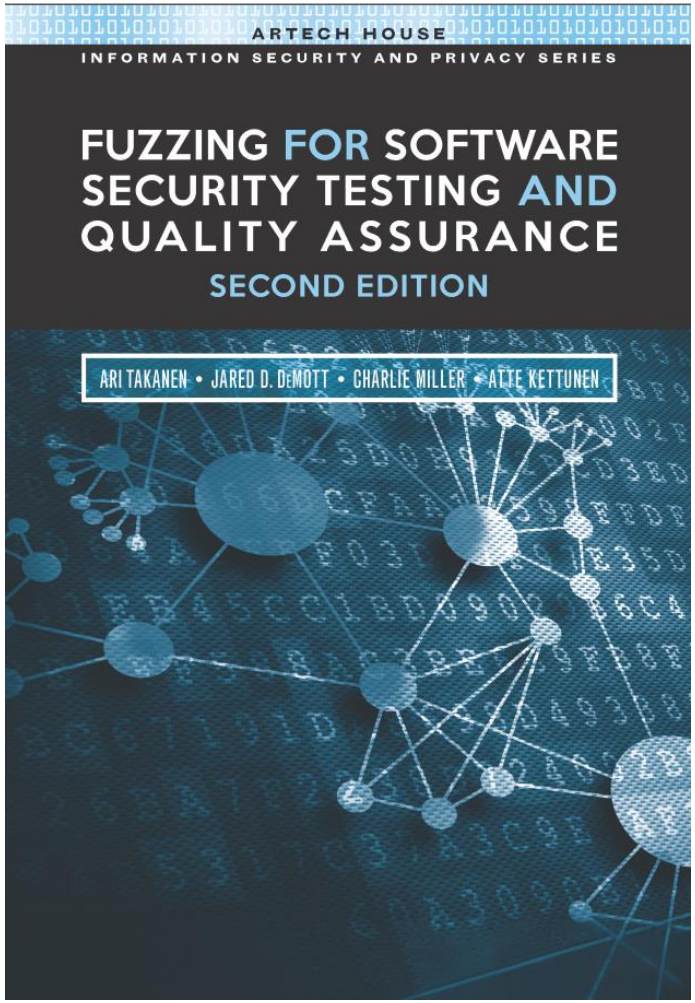


Tài liệu tham khảo

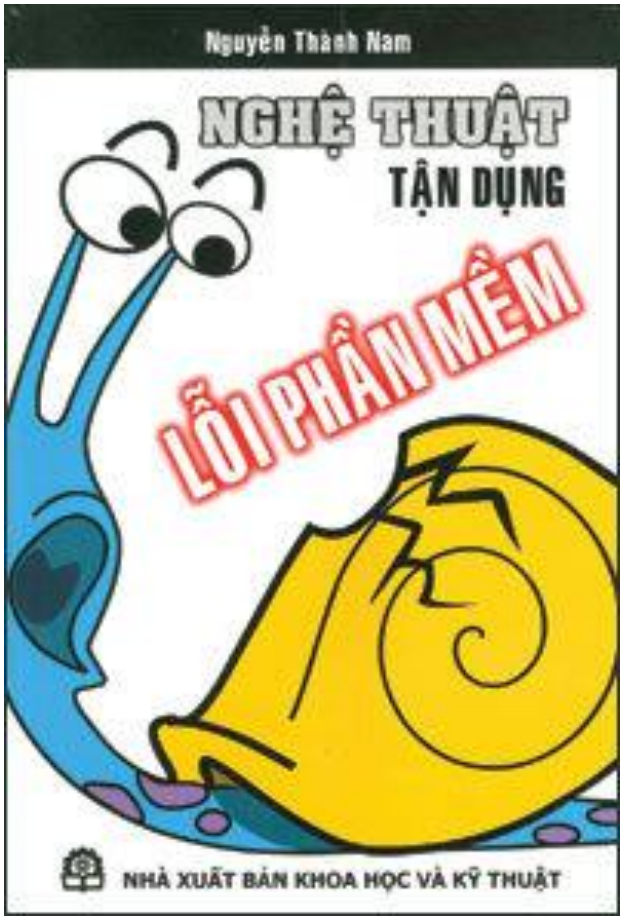




Tài liệu tham khảo



Tài liệu tham khảo



Tài liệu tham khảo

- https://security.berkeley.edu/secure-coding-practice-guidelines
- https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices
- https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf
- https://www.softwaretestinghelp.com/guidelines-for-secure-coding/
- http://security.cs.rpi.edu/courses/binexp-spring2015/
- https://www.ired.team/

Lập trình an toàn & Khai thác lỗ hổng phần mềm

Trường ĐH CNTT - ĐHQG TP. HCM