

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

Giới thiệu môn học

- Mã môn học: **NT521**
- Tên môn học:
 - Tiếng Việt: **Lập trình an toàn và khai thác lỗ hổng phần mềm**
 - Tiếng Anh: **Secure Programming and Exploiting Vulnerabilities**
- Số tín chỉ: **04**
- Thời gian: **15 buổi x 3 tiết**

Giới thiệu GV










- **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** - *Information Security Researcher @ UIT InSecLab*
 - Email: quyennh@uit.edu.vn



Giới thiệu môn học

Tại sao lại học môn học này?

Các công việc liên quan tới nội dung môn học

 <p>Kỹ sư an ninh phần mềm (Software Security Engineer)</p>	 <p>Chuyên gia Phân tích Lỗ hổng (Vulnerability Analyst)</p>	 <p>Penetration Tester (Pentester)</p>
 <p>Chuyên gia Phân tích Mã Độc (Malware Analyst)</p>	 <p>Kỹ sư Phát triển Phần mềm An toàn (Secure Software Developer)</p>	 <p>Chuyên gia Đánh giá Bảo mật (Security Auditor)</p>
 <p>Kỹ sư Phát triển Công cụ Bảo mật (Security Tools Developer)</p>	 <p>Tư vấn Bảo mật (Security Consultant)</p>	 <p>Nghiên cứu học thuật (Academic Research)</p>

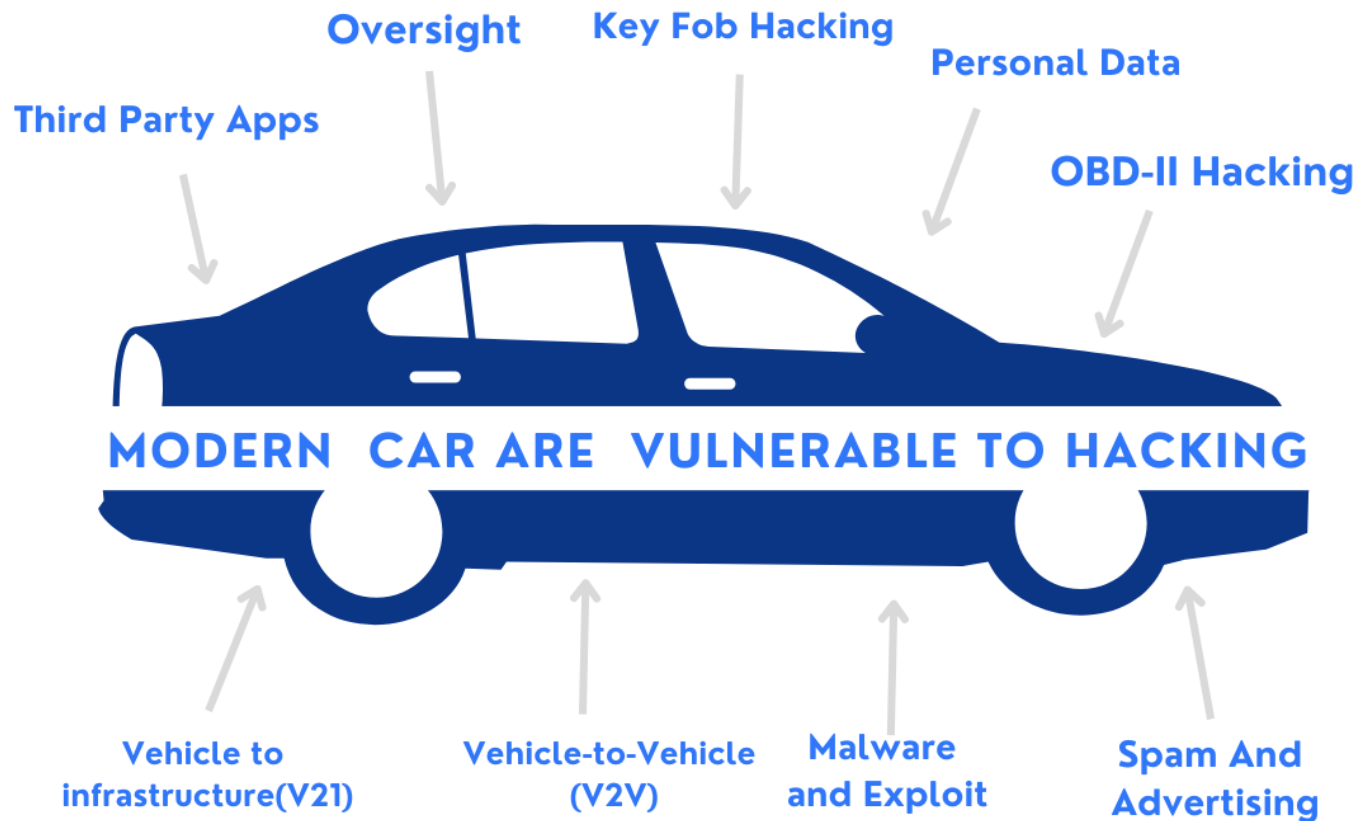
Bắt đầu

- Một chiếc xe ô tô có thể **không an toàn** do những lí do gì?



Bắt đầu

- Một chiếc xe ô tô có thể **bị tin tặc tấn công?**



Bắt đầu

- Một chiếc xe ô tô có thể **bị tin tặc tấn công?**

Ví Dụ Mẫu Thiết Kế Ô Tô Dễ Bị Tấn Công:

- **Hệ thống mở khóa không dây (Keyless Entry System):**
 - Thiết kế hệ thống này không bao gồm mã hóa đủ mạnh để bảo vệ tín hiệu giữa chìa khóa và xe. Tin tặc có thể dễ dàng thực hiện các cuộc tấn công relay attack, bằng cách chặn và phát lại tín hiệu, mở khóa xe mà không cần chìa khóa thật.
- **Phần mềm điều khiển động cơ (Engine Control Software):**
 - Thiết kế phần mềm điều khiển động cơ không có kiểm tra tính toàn vẹn và xác thực mã nguồn. Điều này cho phép kẻ tấn công tiêm mã độc vào hệ thống, gây ra các lỗi trong điều khiển động cơ hoặc thậm chí tắt động cơ khi xe đang chạy.
- **Hệ thống thông tin giải trí (Infotainment System):**
 - Hệ thống này được kết nối với mạng internet và mạng nội bộ của xe nhưng thiếu các biện pháp bảo mật cần thiết, như tường lửa hoặc phân đoạn mạng. Kẻ tấn công có thể khai thác các lỗ hổng trong hệ thống thông tin giải trí để truy cập vào các hệ thống quan trọng khác của xe, như hệ thống phanh hoặc lái.

Khảo sát kiến thức: CWE là gì?

Common Weakness Enumeration (CWE) là danh sách các loại điểm yếu phổ biến trong phần mềm và phần cứng có liên quan đến bảo mật. Một “**điểm yếu**” là một **điều kiện** trong phần mềm, firmware, phần cứng, hoặc thành phần dịch vụ mà, dưới một số hoàn cảnh nhất định, có thể **góp phần tạo ra các lỗ hổng bảo mật**

CWE Top 25 Most Dangerous Software Weaknesses



Welcome to the 2023 Common Weakness Enumeration (CWE™) Top 25 Most Dangerous Software Weaknesses list (CWE™ Top 25). This list demonstrates the currently most common and impactful software weaknesses.

Often easy to find and exploit, these can lead to exploitable vulnerabilities that allow adversaries to completely take over a system, steal data, or prevent applications from working.

[2023 Top 25 List](#)

[Key Insights](#)

[Methodology](#)

Common Weakness Enumeration (CWE) is a list of common software and hardware **weakness types** that have security ramifications. A “weakness” is **a condition** in a software, firmware, hardware, or service component that, **under certain circumstances**, could **contribute to the introduction of vulnerabilities**.

Khảo sát kiến thức: Lỗ hổng (vulnerability) là gì?

Một lỗ hổng (vulnerability) là một khiếm khuyết (flaw) hoặc điểm yếu (weakness) trong hệ thống máy tính, phần mềm, các quy trình bảo mật, các kiểm soát nội bộ, hoặc thiết kế và triển khai, có thể bị khai thác để vi phạm chính sách bảo mật của hệ thống.

Vulnerability is a weak point

Weak point of a process or an assets which increases a likelihood of a risk



A vulnerability is a flaw or weakness in a computer system, software, its security procedures, internal controls, or design and implementation, which could be exploited to violate the system security policy.

Khảo sát kiến thức: CVE là gì?

The screenshot displays a grid of eight vulnerability entries from a CVE database. Each entry is a card with the following structure:

- VULNERABILITY NAME:** CVE-XXXX-XXXX
- Score:** X.X/10 (e.g., 9.8/10, 7.5/10, 10/10)
- Description:** A brief text explaining the vulnerability and its impact.
- Tags:** A set of buttons indicating the severity (All, Unauthenticated, Privilege Escalation, Critical, High, Progress) and affected products (e.g., Barracuda Networks, Citrix, Cisco, Atlassian).

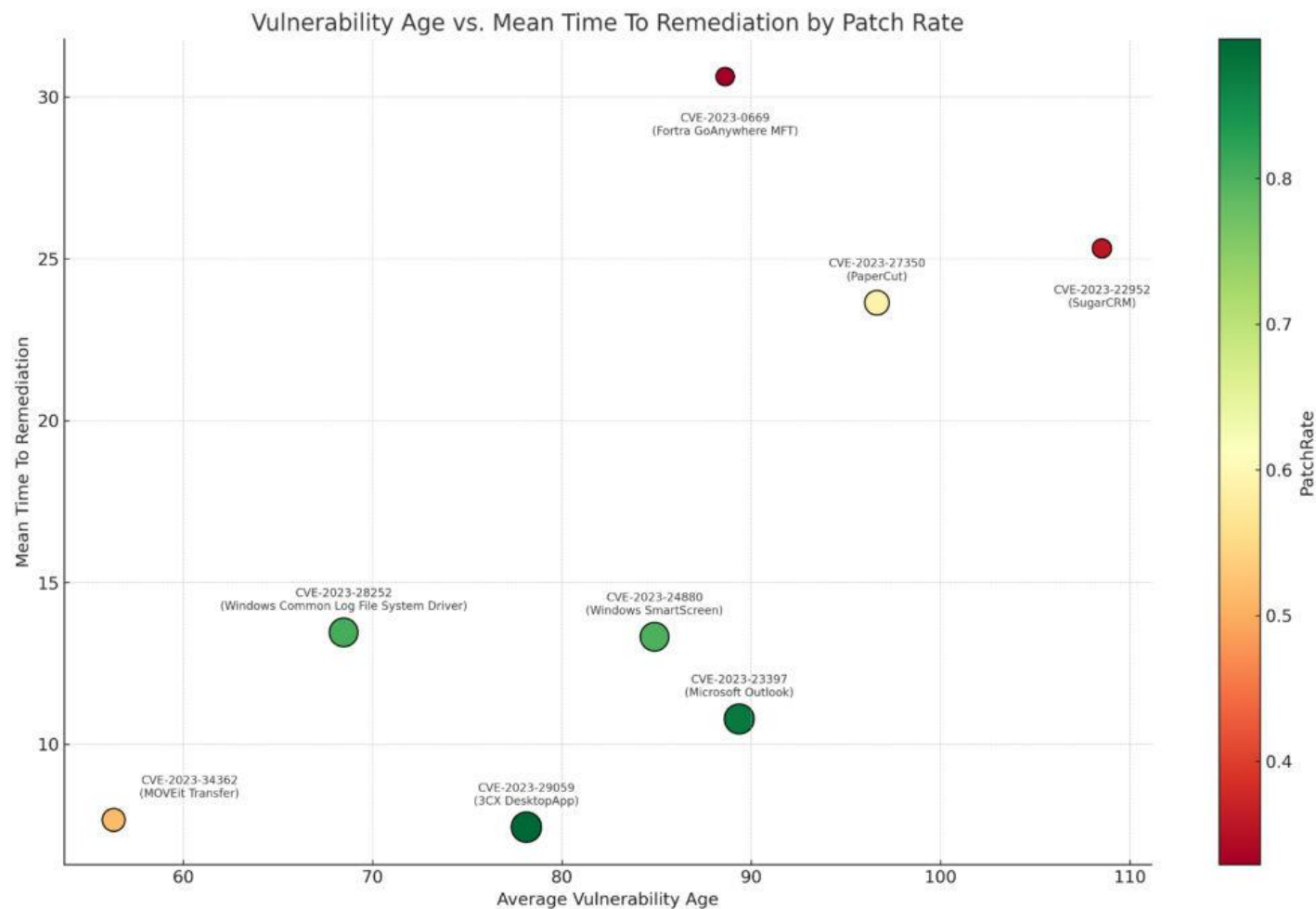
The eight entries shown are:

- CVE-2023-34362** (9.8/10): "The MOVEit Transfer vulnerability" - Exploited by the ClOp ransomware group.
- CVE-2023-4966** (7.5/10): "Citrix bleed" - Exploited by LockBit 3.0 affiliates.
- CVE-2023-20198** (10/10): A maximum severity privilege escalation vulnerability on Cisco Devices.
- CVE-2023-22518** (9.8/10): A critical improper authorization vulnerability used by Cerber ransomware.
- CVE-2023-2868** (9.8/10): Replace your compromised ESG appliances - UNC4841 exploited this vulnerability for espionage.
- CVE-2023-20269** (9.1/10): Information disclosure vulnerability exploited by LockBit and Akira ransomware groups.
- CVE-2023-27350** (9.8/10): Bl00dy - This vulnerability was exploited by Bl00dy, ClOp and Lockbit.
- CVE-2023-22515** (9.8/10): Broken access control flaw that allowed threat actors to become Confluence administrators.

CVE, viết tắt của **Common Vulnerabilities and Exposures**, là danh sách các lỗ hổng bảo mật máy tính đã được công bố công khai.

→ Khi ai đó đề cập đến **một CVE**, họ đang nói về một lỗ hổng bảo mật đã được gán một số ID CVE.

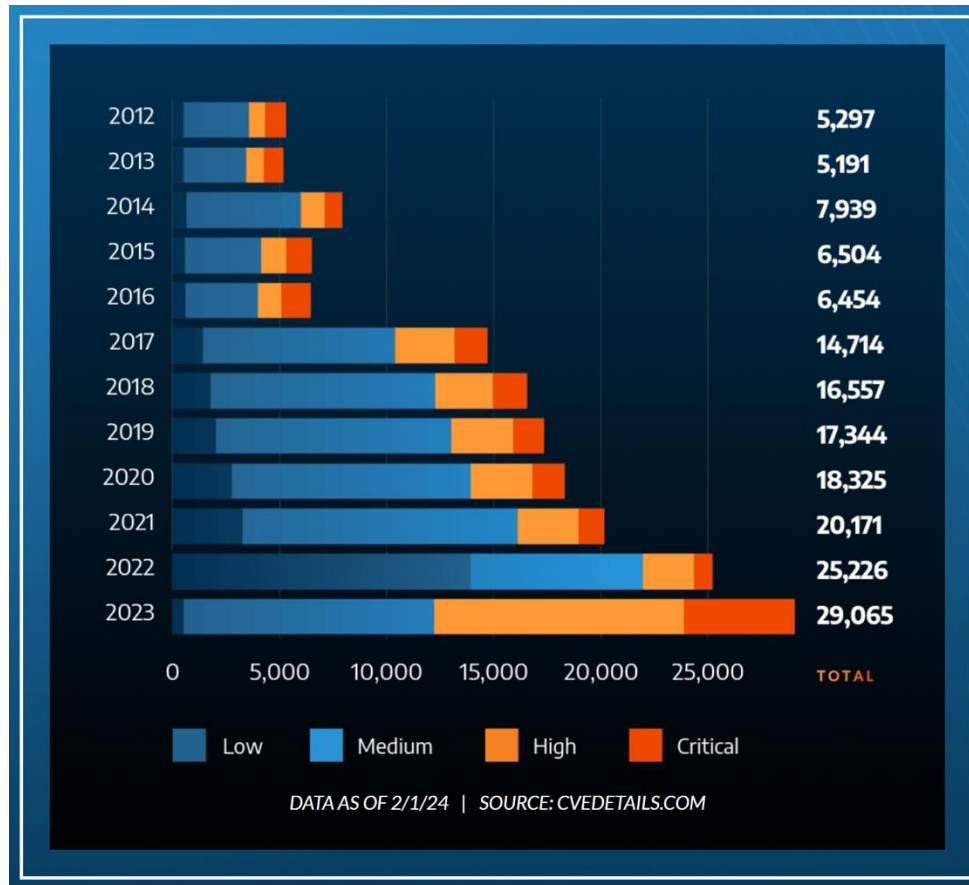
Khảo sát kiến thức: Top 10 lỗ hổng được khai thác 2023?



Analysis of MTTR, Patch Rate, and Vulnerability Age for Top 10 Vulnerabilities in 2023 (Qualys Threat Research Unit)

Khảo sát kiến thức:

Các CVE được ghi nhận trong 10 năm?



- Có **hơn 29,000 lỗ hổng bảo mật (vulnerabilities)** được công bố trong năm 2023, tăng thêm hơn 3,800 lỗ hổng bảo mật và phơi nhiễm thông tin (CVEs) so với năm 2022.
- Điều đáng lo ngại hơn so với số lượng lớn các lỗ hổng trong năm 2023 là **hơn một nửa trong số đó** đã được đánh giá với điểm CVSS cho thấy mức độ nghiêm trọng cao hoặc cực kỳ nghiêm trọng.

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.

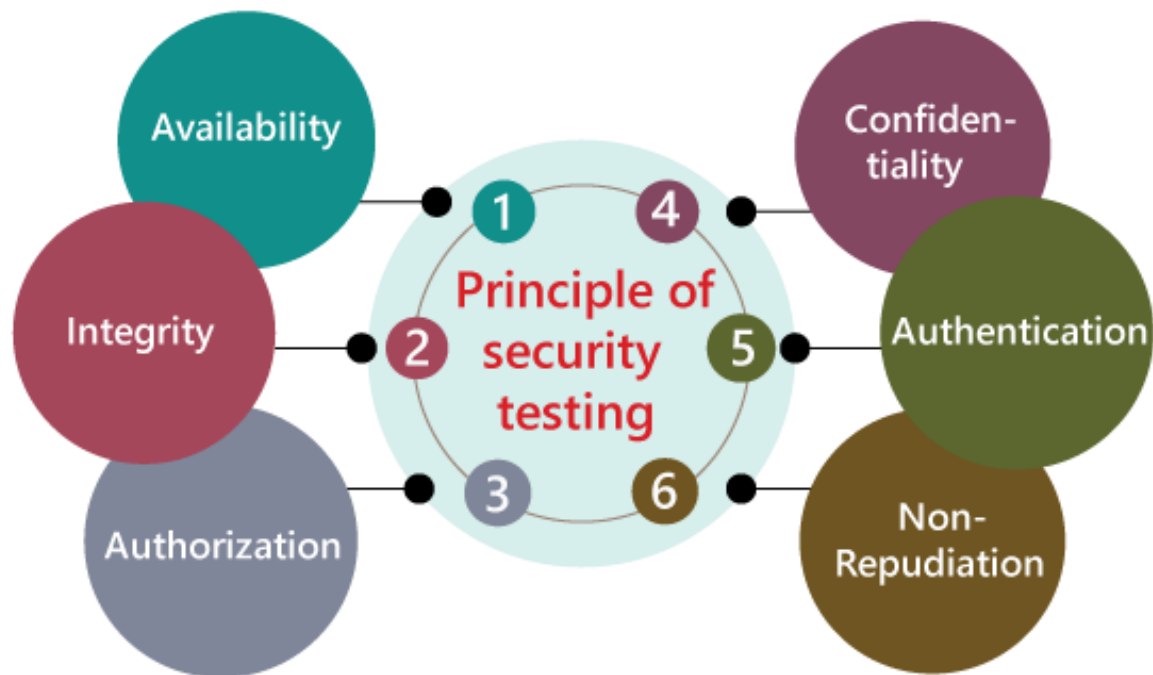
“Software security is an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks”.

Khảo sát kiến thức



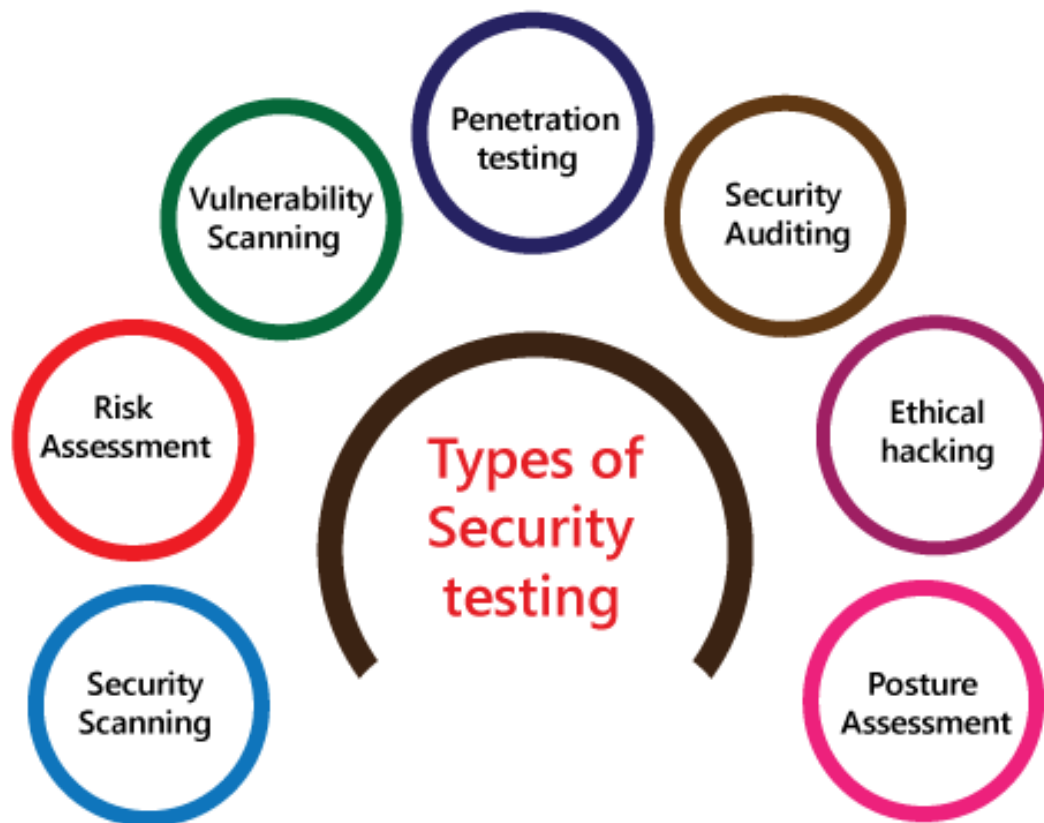
Quy trình phát triển phần mềm - Software Development Life Cycle (SDLC)

Khảo sát kiến thức



Các nguyên tắc/khía cạnh trong kiểm thử phần mềm

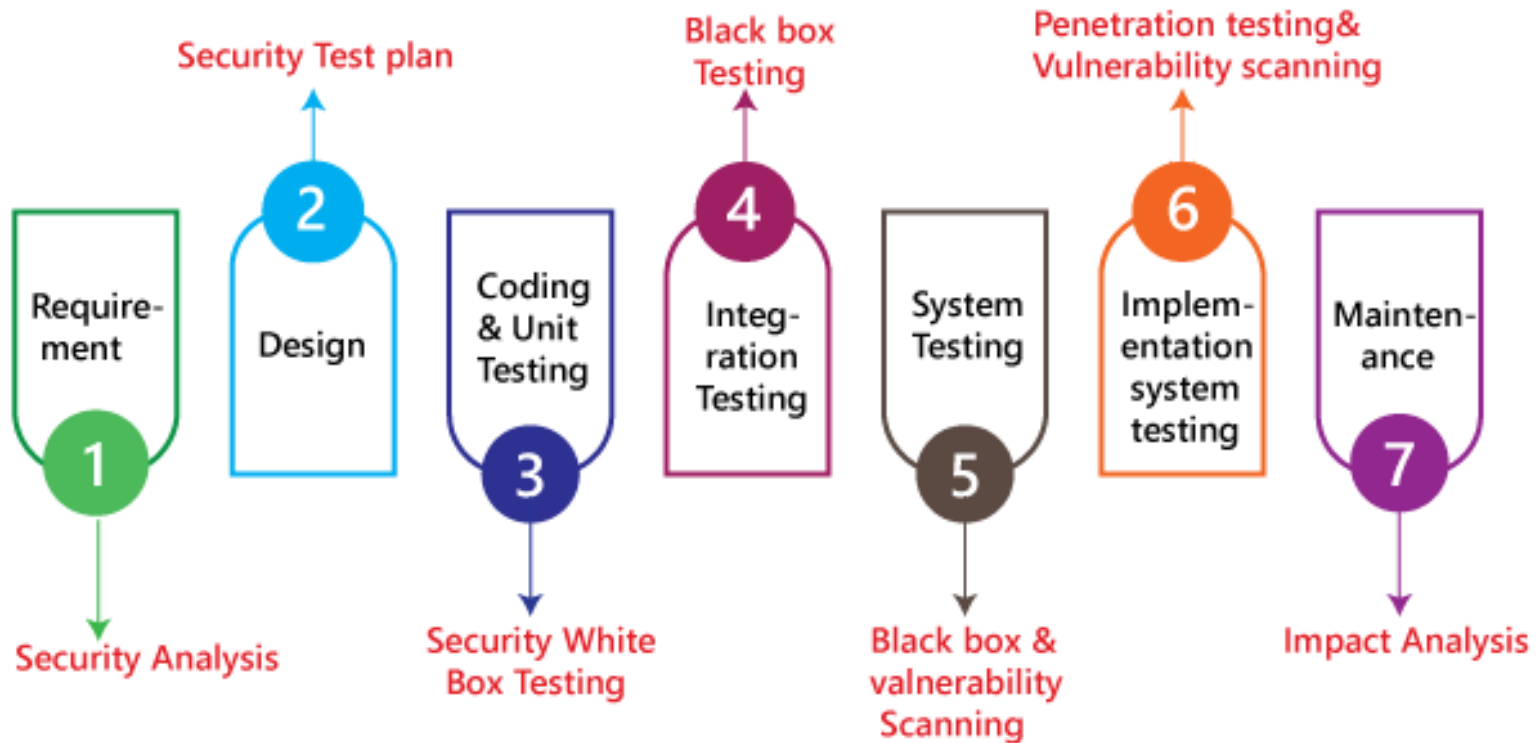
Khảo sát kiến thức



Các dạng kiểm thử trong kiểm thử phần mềm

Khảo sát kiến thức

Security Testing along with SDLC



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

Khảo sát kiến thức



Chu kỳ SSDLC được thực hiện lặp đi lặp lại

Khảo sát kiến thức

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

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

- **Quy trình phát triển phần mềm - Software Development Lifecycle (SDLC)**
 - Các mô hình SDLC truyền thống
 - Các mô hình Agile
- **Quy 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, Yêu cầu và qui định của môn học
- SDLC – Tổng quan về qui trình phát triển phần mềm (Phần 1)

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

- **Buổi 02:** SDLC – Tổng quan về qui trình phát triển phần mềm (Phần 2)

- **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 chaining attack
- **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 thiết kế, phân tích + 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 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.
- Tham gia buổi học:
 - Được vắng **tối đa 03/15 buổi học**.
 - Có lí do chính đáng cho các buổi vắng học khác (nếu có).
 - Cấm thi cuối kì với các trường hợp không thực hiện theo qui định tham gia buổi học tối thiểu

CHƯƠNG 3. KIỂM TRA VÀ THI HỌC PHẦN

Điều 20. Tổ chức đánh giá môn học

1. Điều kiện dự thi kết thúc học phần

Sinh viên được dự thi kết thúc học phần khi có mặt từ 80% trở lên số buổi học theo thời khóa biểu của học phần đó, trừ những sinh viên được Hiệu trưởng cho phép học chương trình song ngành.

Giảng viên có thể đề nghị P.ĐTĐH không cho sinh viên dự thi kết thúc học phần theo quy định riêng của môn học đã công bố cho sinh viên.

(Trích QĐ số 790/QĐ-ĐHCNTT ngày 28/9/2022 về quy chế đào tạo theo học chế tín chỉ)

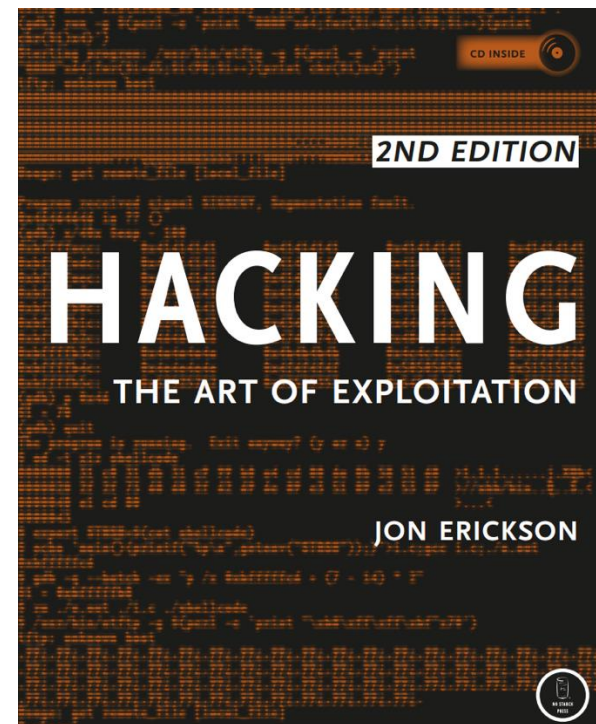
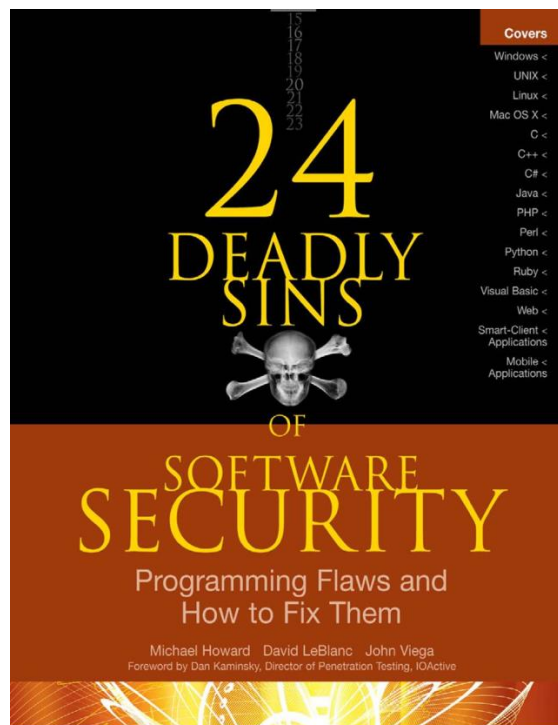
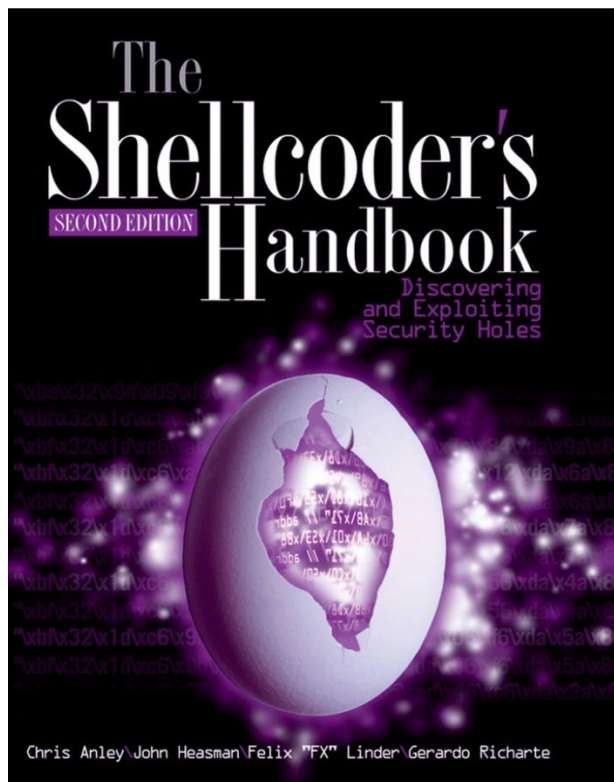
Qui định học tập

- Nhóm: **03** thành viên/nhóm
- Qui định khi làm việc nhóm:
 - Không ghi đầy đủ thông tin nhóm \rightarrow 0đ
 - Sao chép bài nhóm khác \rightarrow 0đ
 - Điểm của thành viên trong nhóm không phải là điểm chung của nhóm

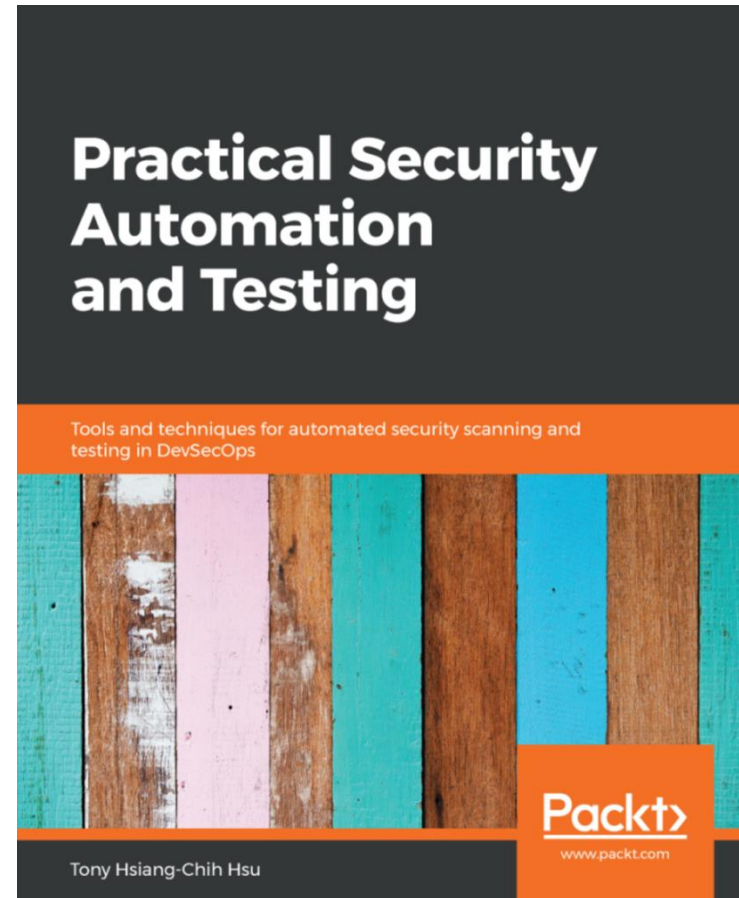
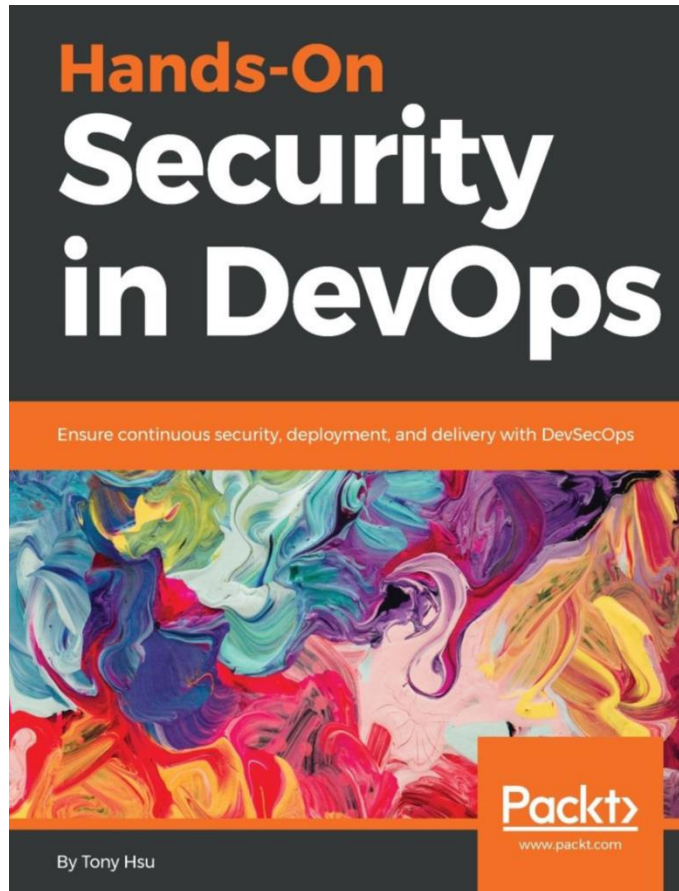
Qui định học tập

- **Đồ án môn học:** tổng hợp các kiến thức môn học
 - Giao đề tài Đồ án vào **buổi học số 03**
 - **Không đăng kí đồ án theo qui định, hoặc không báo cáo tiến độ đồ án giữa kì → NHẬN ĐIỂM KHÔNG (0) CHO ĐIỂM QUÁ TRÌNH**
 - Báo cáo đề tài Đồ án + (Bài tập) bắt đầu từ Buổi 12.

Tài liệu tham khảo



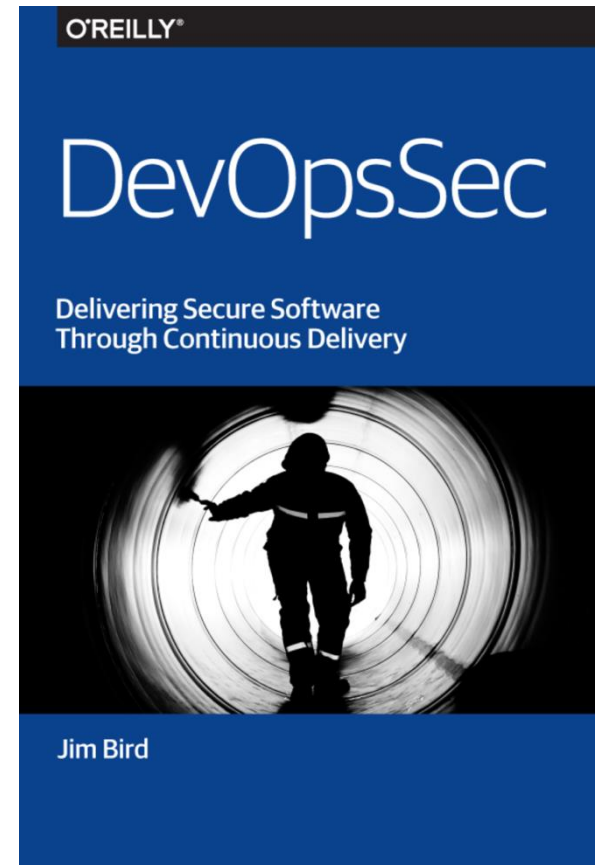
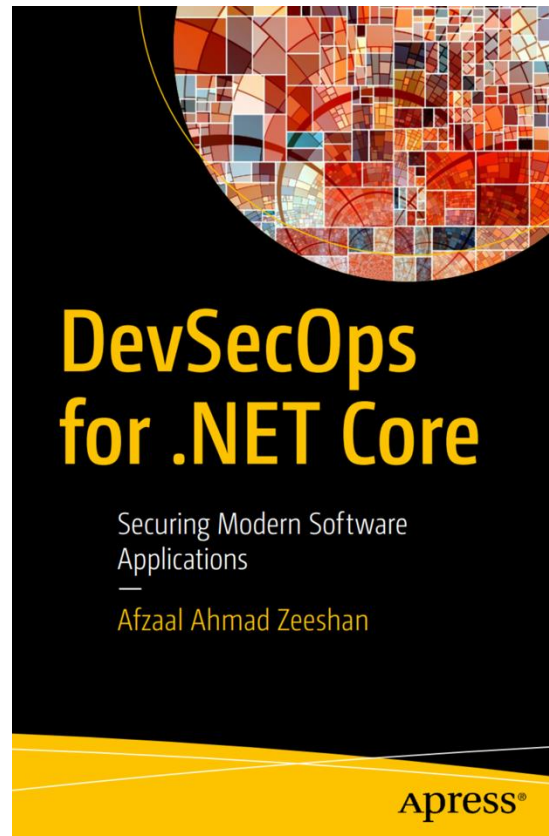
Tài liệu tham khảo



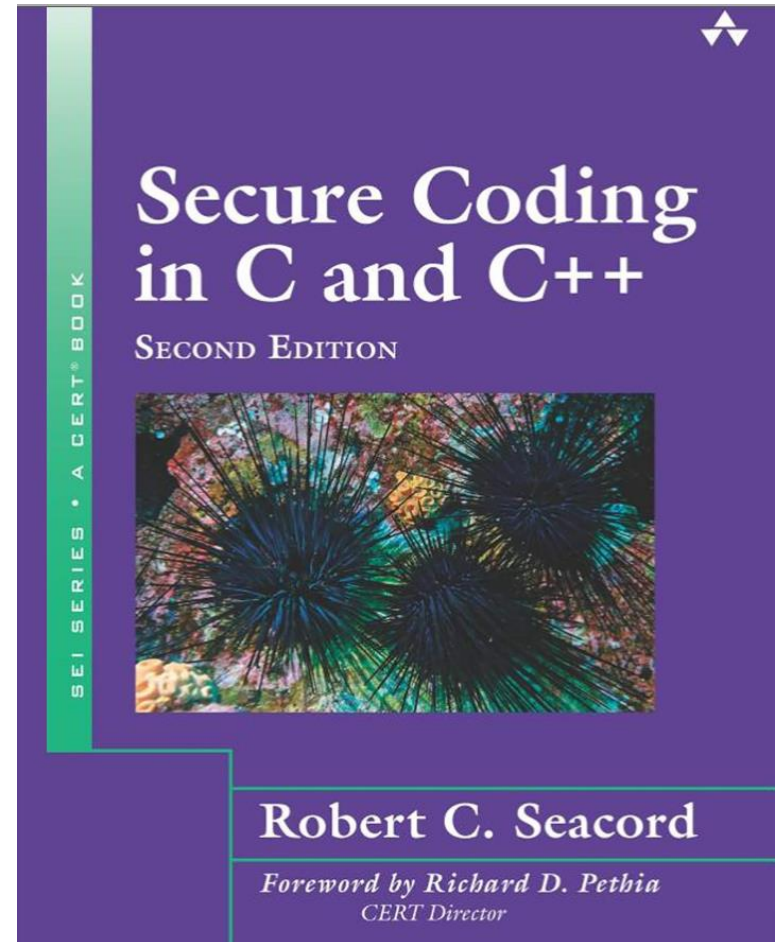
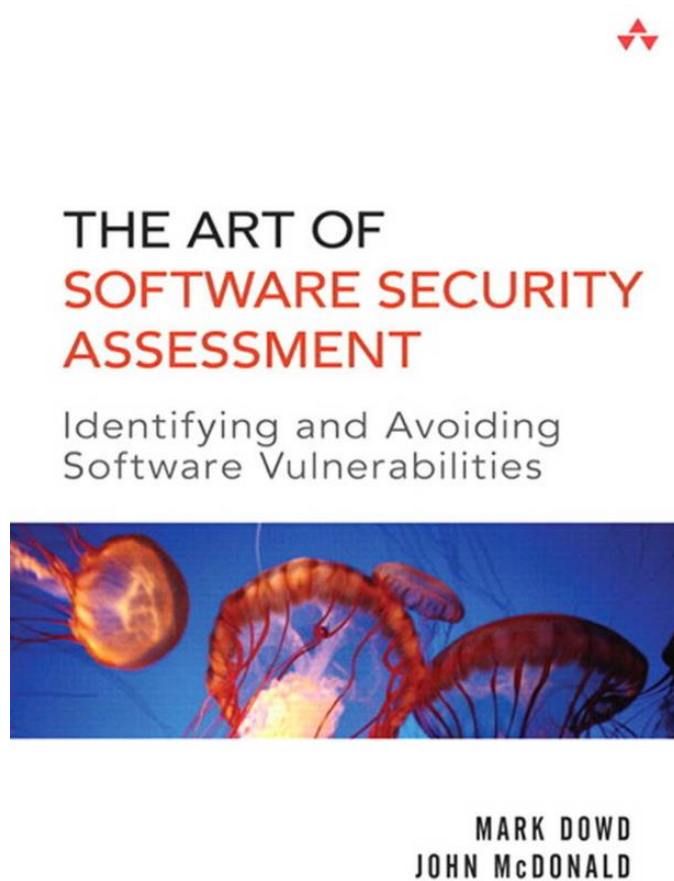
Tài liệu tham khảo



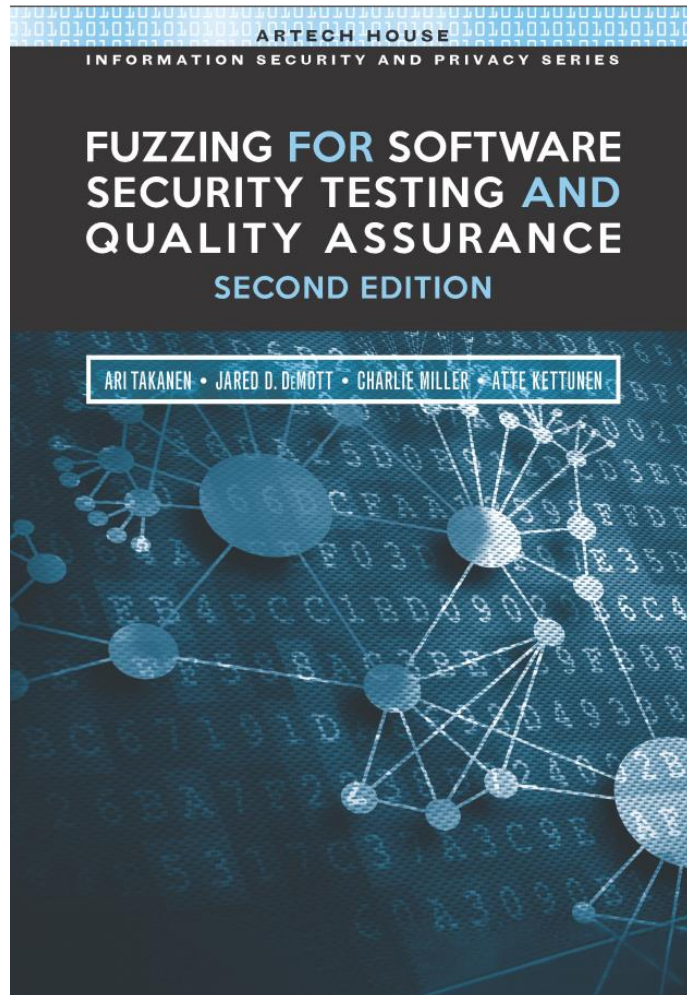
Laura Bell, Michael Brunton-Spall,
Rich Smith & Jim Bird



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/sec+code/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/>

A person is seen from behind, working on several laptops. The screens display lines of code in a dark-themed editor. The person is wearing a striped long-sleeved shirt and glasses. The scene is set in a modern office or lab environment with wooden desks.

Lập trình an toàn & Khai thác lỗ hổng phần mềm



Trường ĐH CNTT - ĐHQG TP. HCM