**BÁO CÁO LAB 4**

*Môn: An toàn mạng máy tính nâng cao*

*GVTH: Đỗ Thị Phương Uyên*

|  |  |
| --- | --- |
| Sinh viên thực hiện | **Sinh viên 1**  MSSV: 21521520  Họ tên: Huỳnh Minh Tân Tiến  **Sinh viên 2**  MSSV: 21521817  Họ tên: Bùi Hoàng Trúc Anh  **Sinh viên 3**  MSSV: 21521253  Họ tên: Lê Hoàng Oanh  **Sinh viên 4**  MSSV: 21520353  Họ tên: Nguyễn Ngọc Trà My |
| Lớp | **NT534.O21.ATCL** |
| Tổng thời gian thực hiện Lab trung bình |  |
| Phân chia công việc  *(nếu là nhóm)* | **[Sinh viên 1]**:  **[Sinh viên 2]**: |
| Link Video thực hiện *(nếu có yêu cầu)* |  |
| Ý kiến *(nếu có)*  *+ Khó khăn gặp phải + Đề xuất, góp ý…* |  |
| Điểm tự đánh giá *(bắt buộc)* | 10/**10** |

*[Nội dung báo cáo chi tiết – Trình bày tùy sinh viên, Xuất file .PDF khi nộp]*

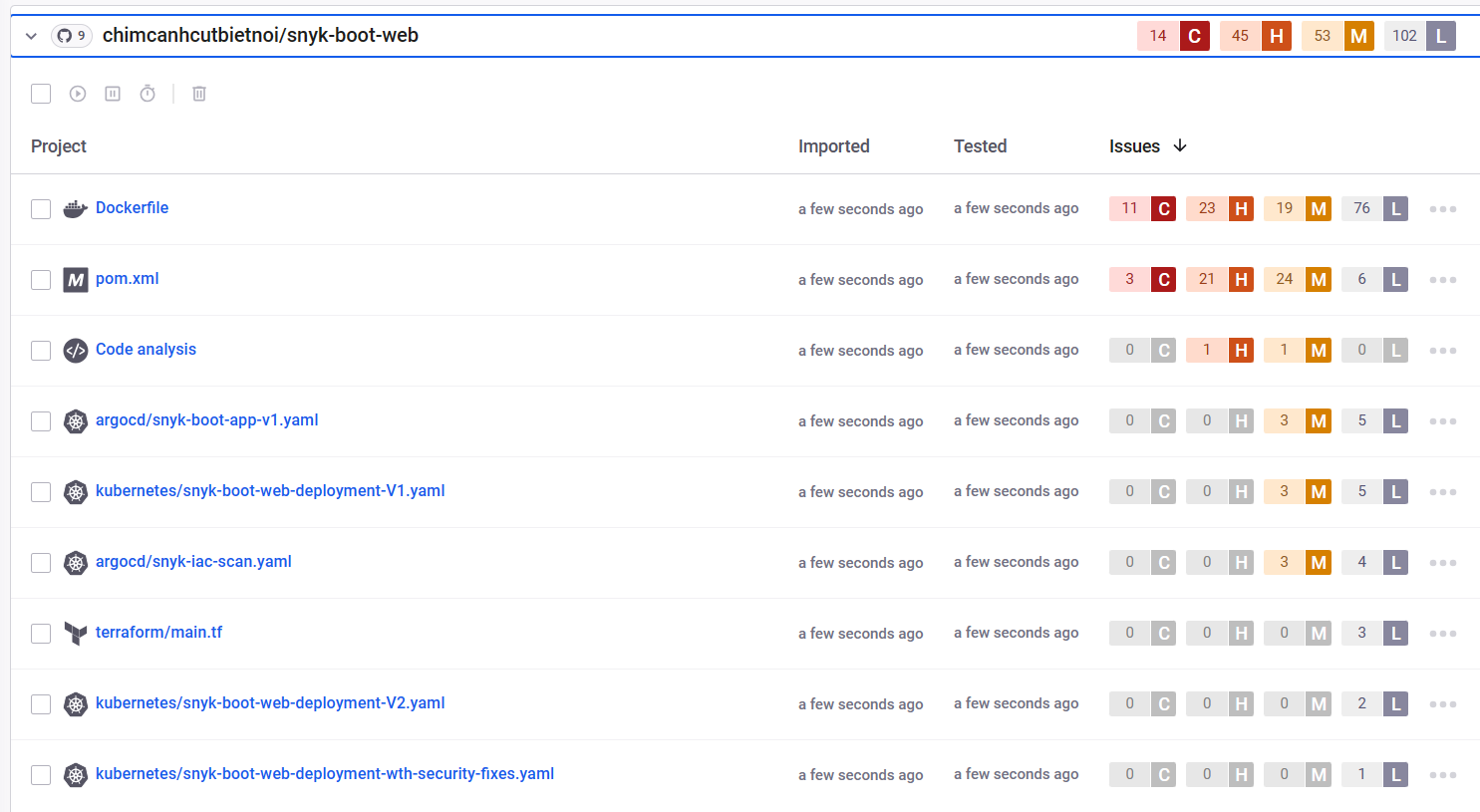
**Question: Dựa vào thông tin về các công cụ của Snyk, hãy dự đoán các công cụ này của Snyk hỗ trợ kiểm tra, đánh giá và khắc phục các vấn đề bảo mật ở những giai đoạn nào trong quá trình phát triển phần mềm?**

Công cụ của Snyk có thể được sử dụng trong các giai đoạn khác nhau của quy trình phát triển phần mềm để kiểm tra, đánh giá và khắc phục các vấn đề bảo mật. Dưới đây là các giai đoạn chính mà các công cụ này có thể hỗ trợ:

1. **Phát triển và thiết kế:** Công cụ có thể kiểm tra mã nguồn và các thiết kế sơ bộ để phát hiện và cảnh báo về các lỗ hổng bảo mật tiềm ẩn từ các quy trình thiết kế ban đầu.
2. **Phát triển mã:** Trong quá trình viết mã, công cụ có thể tích hợp với môi trường phát triển tích hợp (IDE) để cung cấp cảnh báo và gợi ý sửa lỗi trực tiếp cho các lỗ hổng bảo mật.
3. **Kiểm thử và triển khai:** Trước khi triển khai, công cụ có thể tự động kiểm tra mã nguồn và các thành phần phần mềm thứ ba để đảm bảo rằng không có lỗ hổng bảo mật nào được triển khai cùng với ứng dụng.
4. **Triển khai và vận hành:** Công cụ có thể cung cấp giám sát liên tục và phát hiện các vấn đề bảo mật mới sau khi ứng dụng đã được triển khai.
5. **Quản lý và cập nhật:** Công cụ cung cấp cập nhật liên tục về các lỗ hổng bảo mật mới được phát hiện và hỗ trợ quản lý quy trình cập nhật phần mềm để khắc phục chúng.

Tóm lại, công cụ của Snyk có thể hỗ trợ trong toàn bộ quy trình phát triển phần mềm, từ thiết kế ban đầu đến triển khai và vận hành, giúp đảm bảo rằng ứng dụng được phát triển và vận hành một cách an toàn về mặt bảo mật.

**Task: Quan sát và phân tích kết quả của việc scan trên các môi trường khác nhau: code application, container, IaC.**



1. **Code application:**

Dockerfile: Tìm thấy 11 lỗ hổng với mức độ nghiêm trọng trung bình (M) và cao (H). Các lỗ hổng này có thể ảnh hưởng đến bảo mật của container được xây dựng từ Dockerfile.

Mpom.xml: Tìm thấy 3 lỗ hổng với mức độ nghiêm trọng trung bình (M). Các lỗ hổng này có thể ảnh hưởng đến bảo mật của các ứng dụng Java được triển khai bằng Maven.

Code analysis: Không tìm thấy lỗ hổng nào.

1. **Container:**

argocd/snyk-boot-app-v1.yaml: Không tìm thấy lỗ hổng nào.

kubernetes/snyk-boot-web-deployment-V1.yaml: Tìm thấy 3 lỗ hổng với mức độ nghiêm trọng trung bình (M). Các lỗ hổng này có thể ảnh hưởng đến bảo mật của triển khai Kubernetes.

kubernetes/snyk-boot-web-deployment-V2.yaml: Tìm thấy 3 lỗ hổng với mức độ nghiêm trọng trung bình (M). Các lỗ hổng này có thể ảnh hưởng đến bảo mật của triển khai Kubernetes.

kubernetes/snyk-boot-web-deployment-wth-security-fixes.yaml: Tìm thấy 1 lỗ hổng với mức độ nghiêm trọng cao (OH). Lỗ hổng này có thể ảnh hưởng đến bảo mật của triển khai Kubernetes.

1. **IaC:**

argocd/snyk-lac-scan.yaml: Tìm thấy 1 lỗ hổng với mức độ nghiêm trọng cao (C). Lỗ hổng này có thể ảnh hưởng đến bảo mật của cấu hình ArgoCD.

terraform/main.tf: Tìm thấy 1 lỗ hổng với mức độ nghiêm trọng cao (C). Lỗ hổng này có thể ảnh hưởng đến bảo mật của cấu hình Terraform.

**Kết luận:**

Môi trường code application: Có nhiều lỗ hổng bảo mật cần được khắc phục.

Môi trường container: Có một số lỗ hổng bảo mật cần được khắc phục, đặc biệt là trong các triển khai Kubernetes.

Môi trường IaC: Có một số lỗ hổng bảo mật nghiêm trọng cần được khắc phục trong cấu hình ArgoCD và Terraform.

**Task: Dùng tính năng Snyk Pull Request để fix các lỗ hổng được tìm thấy**

Mở file pom.xml để quan sát lại các lỗ hổng bảo mật đã được tìm thấy.

Chọn một lỗ hổng bảo mật và chọn Fix this vulnerability

Chọn những lỗ hổng cần khắc phục và chọn Open PR Fix để tạo một pull request mới

A screenshot of a computer

Description automatically generated

•Sau khi kiểm tra và xác nhận không có xung đột gì, tiến hành merge pull request

A screenshot of a computer

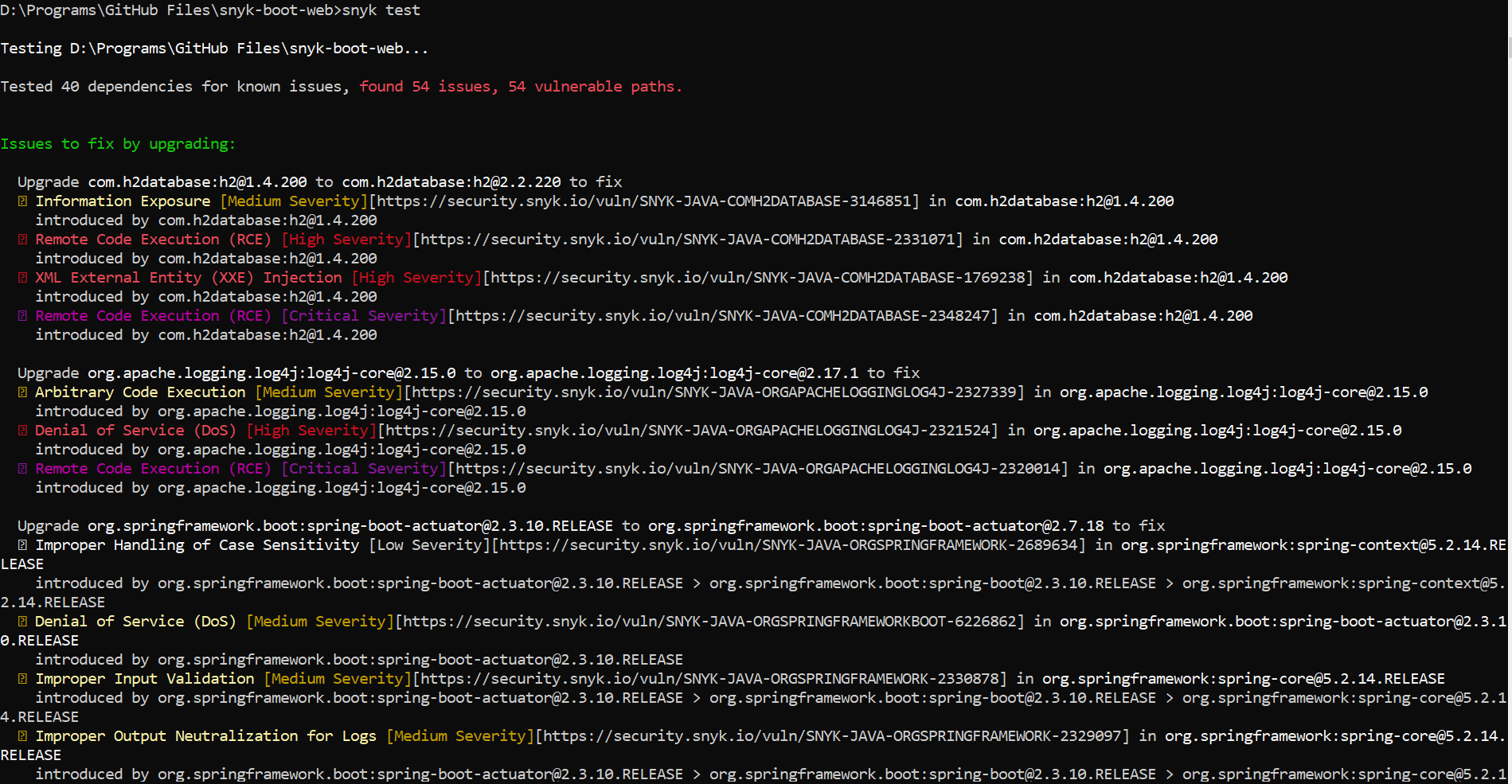
Description automatically generated

A screenshot of a computer

Description automatically generated

Số lượng lỗ hổng đã giảm từ 54 còn 53

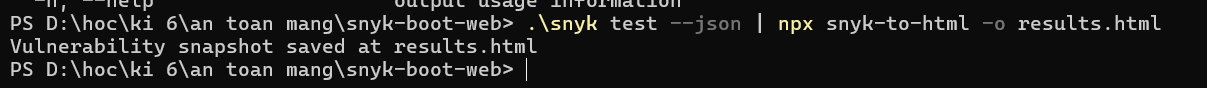
**Task: Cài đặt Snyk CLI, sử dụng các công cụ của Snyk để scan và xuất report thành file HTML**

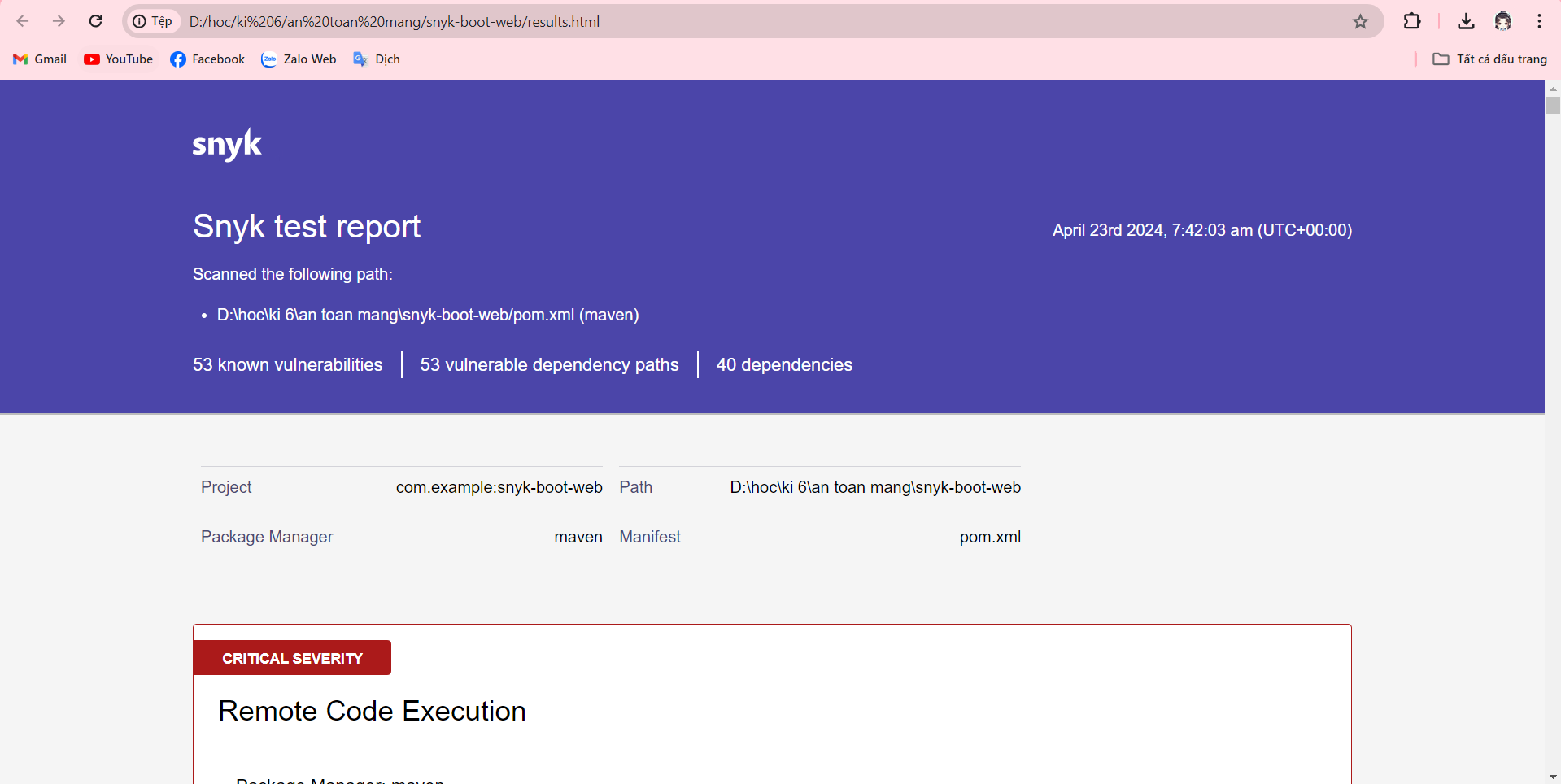
Sử dụng Synk Open Source để scan manifest file 

Sử dụng Synk Code để scan source codeA computer screen with white text

Description automatically generated

Xuất kết quả thành file HTML.





**Task: Cài đặt Snyk plugin/extension vào IDE đang sử dụng và quan sát kết quả scan**

Sau khi cài đặt Snyk plugin vào Visual Studio Code, ta thấy nó scan được những lỗ hổng sau:

A screenshot of a computer

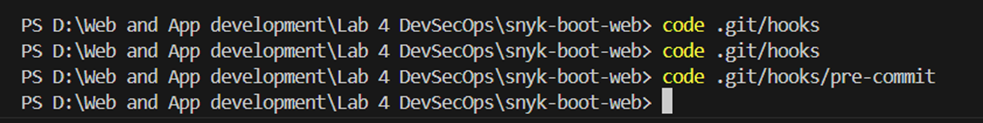
Description automatically generated

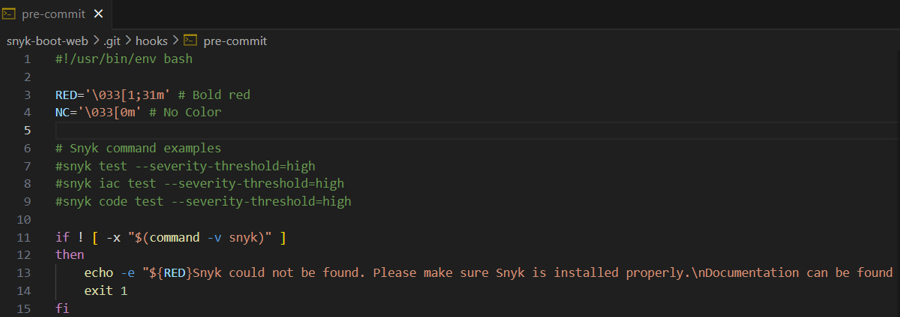
A screenshot of a computer

Description automatically generated

**Bonus: Tạo một pre-commit hook gọi Snyk CLI để scan repository**

Vào mục **.git/hooks** và tạo file pre-commit.sh

Nội dung file pre-commit

Kết quả

