BÁO CÁO THỰC HÀNH

**Môn học: An toàn mạng máy tính nâng cao**

**Lab 4: Security with Snyk in DevSecOps**

*GVHD: Đỗ Thị Phương Uyên*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT534.O21.ATTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hà Thị Thu Hiền | 21522056 | 21522056@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Tất cả các bài tập | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. **Task: Tạo tài khoản Snyk và GitHub**

A screenshot of a computer

Description automatically generated

1. **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?**

A diagram of a product

Description automatically generated with medium confidence

Dựa vào thông tin về các công cụ của Snyk, ta có thể dự đoán các công cụ này hỗ trợ kiểm tra, đánh giá và khắc phục các vấn đề bảo mật ở các giai đoạn sau trong quá trình phát triển phần mềm:

1. **Snyk Code (SAST) và Snyk Open Source (SCA)**: Các công cụ này hỗ trợ kiểm tra mã nguồn và các gói phần mềm mã nguồn mở của bên thứ ba. Chúng có thể được sử dụng trong **giai đoạn phát triển và kiểm thử** để phát hiện các lỗ hổng bảo mật ngay từ khi mã nguồn được viết ra và sử dụng các gói phần mềm.
2. **Snyk Container**: Công cụ này tập trung vào kiểm tra cấu hình của các hình ảnh container và các lỗ hổng trên nền tảng Linux. Nó hỗ trợ phát hiện các lỗ hổng bảo mật trong **quá trình triển khai ứng dụng** thông qua containerization.
3. **Snyk Infrastructure as Code**: Công cụ này cung cấp đánh giá cho các cấu hình cơ sở hạ tầng đám mây. Nó có thể được sử dụng trong **giai đoạn triển khai và quản lý hạ tầng** để phát hiện và khắc phục các lỗ hổng bảo mật liên quan đến cấu hình hạ tầng.

Tóm lại, các công cụ của Snyk có thể hỗ trợ kiểm tra, đánh giá và khắc phục các vấn đề bảo mật ở cả giai đoạn phát triển và triển khai phần mềm, từ việc phát hiện lỗ hổng từ mã nguồn đến cấu hình hạ tầng đám mây.

* 1. **Fork sample webapp vào GitHub repository**

A screenshot of a computer

Description automatically generated

* 1. **Cấu hình GitHub Intergration**
* Đăng nhập vào <http://app.snyk.io>

A screenshot of a computer

Description automatically generated

* Tại trang chủ, chọn Intergrations ➔ Source Control ➔ GitHub

A screenshot of a computer

Description automatically generated

* Điền các thông tin kết nối github và Snyk

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. **Import Repository và enable Snyk Code**
* Synk đã được kết nối với GitHub account.
* Tiếp theo, chúng ta sẽ tiến hành enable Snyk Code và import Repository vào Snyk.
  + Kiểm tra Synk Code đã được enable chưa bằng cách truy cập vào Settings ➔ Snyk Code. Tiến hành Enable và lưu các thay đổi.
  + Thực hiện tương tự với Snyk IaC

A screenshot of a computer

Description automatically generated

* Chọn Project ➔ Add project ➔ GitHub
* Chọn Repo đã được fork ở bước trước và chọn Add selected repositories.
  1. **Phân tích kết quả của Snyk**
* Sau khi tiến hành scan, Snyk trả về kết quả các lỗ hổng, mối đe doạ được tìm thấy

A screenshot of a computer

Description automatically generated

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

Dựa trên kết quả của quá trình quét trên các môi trường khác nhau như mã nguồn ứng dụng, container và cấu hình cơ sở hạ tầng như Infrastructure as Code (IaC), chúng ta có thể rút ra một số nhận định sau:

1. **Mã nguồn ứng dụng (Code analysis):**

* Đầu tiên, ta có thể nhận thấy rằng code analysis phát hiện 1 lỗ hổng High và 1 lỗ hổng medium, click vào để xem rõ hơn.
* Đầu tiên, các lỗ hổng sẽ được hiển thị một cách ngắn gọn, lỗ hổng sẽ được sắp xếp theo priority score, thuộc CWE nào, vulnerablity types, và dưới các lỗ hổng sẽ có các lời cảnh báo về việc code không được an toàn.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* Khi click vào detail thì ta có thể thấy rõ hơn về data flow của lỗ hổng và các fix chúng để tránh khỏi các cuộc tấn công, hiểu rõ được các cuộc tấn công diễn ra như thế nào và cách chống lại chúng.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Container (Dockerfile):**

* Nhìn qua thì ta thấy lỗ hổng ở Dockerfile khá nhiều.

****

* Bao gồm các lỗ hổng ở mức độ: 11 Critical, 23 High, 19 Medium, 80 Low.
* Click vào để xem trong Dockerfile có gì

**A screenshot of a computer

Description automatically generated**

* Ta thấy tools phân tích các lỗ hổng ở các base image với current image và các phiên bản upgrades, bên cạnh đó ta có thể thấy phần Open a fix PR thì nó sẽ hiển thị như sau:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* Click vào Issue ta sẽ thấy [Snyk Vulnerability Database](https://security.snyk.io/vuln) , xem được những rủi ro, cảnh báo và cách fix

**A screenshot of a computer

Description automatically generated**

* Lướt xuống dưới phần Dockerfile thì cũng sẽ thấy các vấn đề cụ thể được liệt kê ra:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* Sẽ có những thông tin về lỗ hổng chi tiết và thông tin bảo mật, nhìn cũng có phần giống với code analysis.

1. **Infrastructure as Code (IaC) (argocd/snyk-iac-scan.yaml, terraform/main.tf):**

* Xem qua argocd/snyk-iac-scan.yaml

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

* Tiếp theo, cùng xem qua terraform/main.tf

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

* Cả 2 loại đều liệt kê ra các lỗ hổng ở các mức độ, chi tiết ở code và cách fix chúng, nhìn sơ bộ thì nó khá đơn giản hơn so với code analysis và dockerfile. Và các lỗ hổng ở phần này cũng ít hơn nhiều và tính ảnh hưởng cũng thấp hơn nhiều.

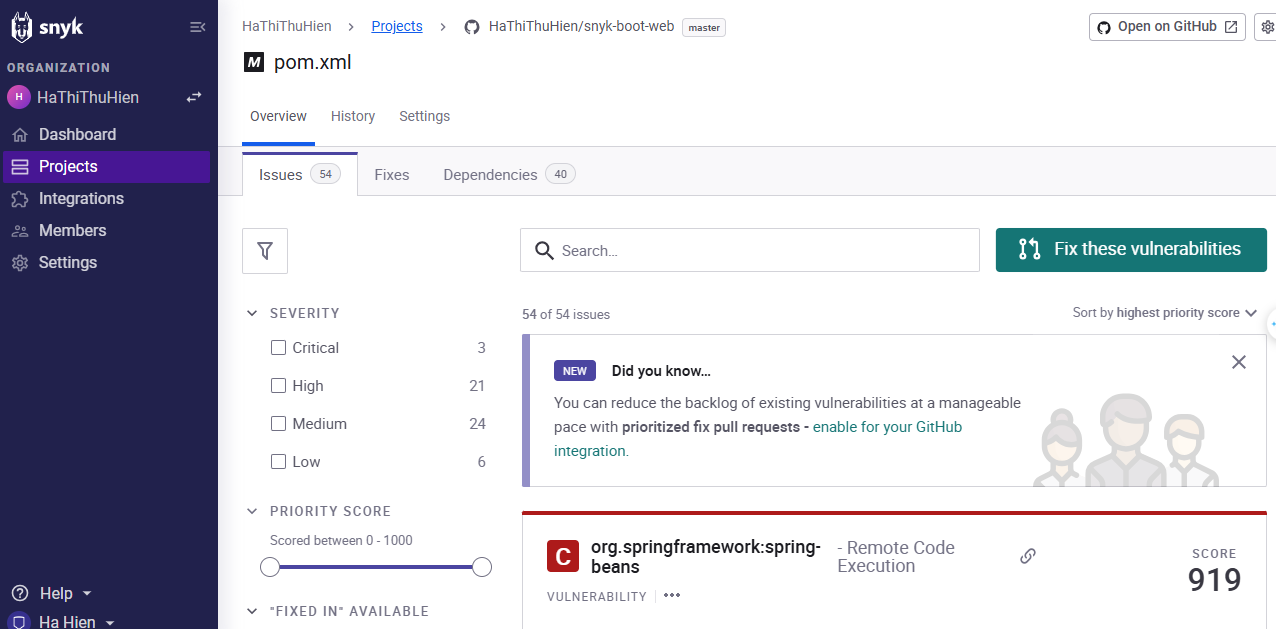
1. **Tổng kết:**

* Các vấn đề bảo mật được phát hiện chủ yếu tập trung ở Dockerfile và các tệp IaC, cho thấy cần phải tập trung vào việc cải thiện bảo mật cho các phần này.
* Đặc biệt, các vấn đề cấp cao (Critical) cần được xử lý ưu tiên để đảm bảo an toàn và bảo mật của hệ thống.
  1. **Fix các lỗ hổng bảo mật bằng tính năng Snyk Pull Request**
* Ở bước này, chúng ta đã có thử tìm hiểu và trình bày ở trên, nhưng để tìm hiểu rõ hơn về vấn đề này trong task sau

**4. 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.

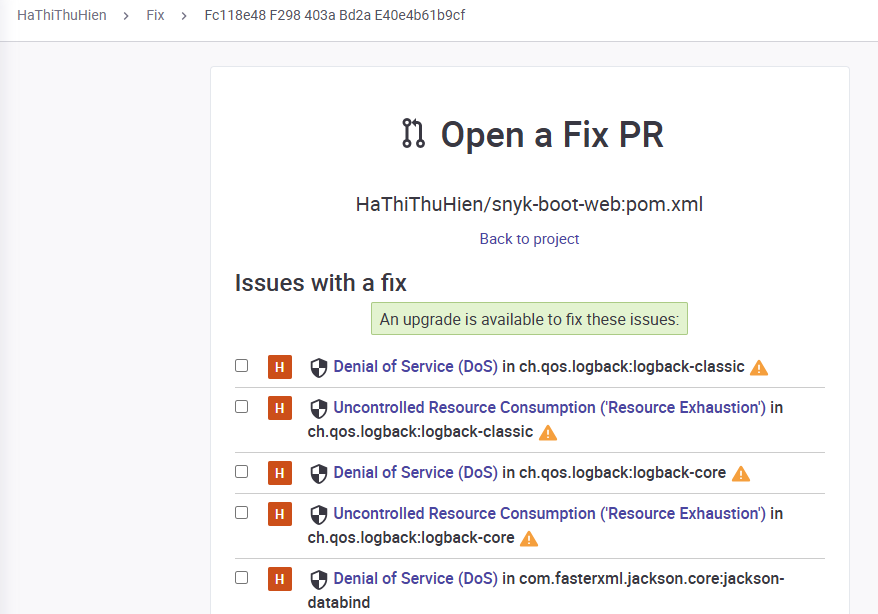


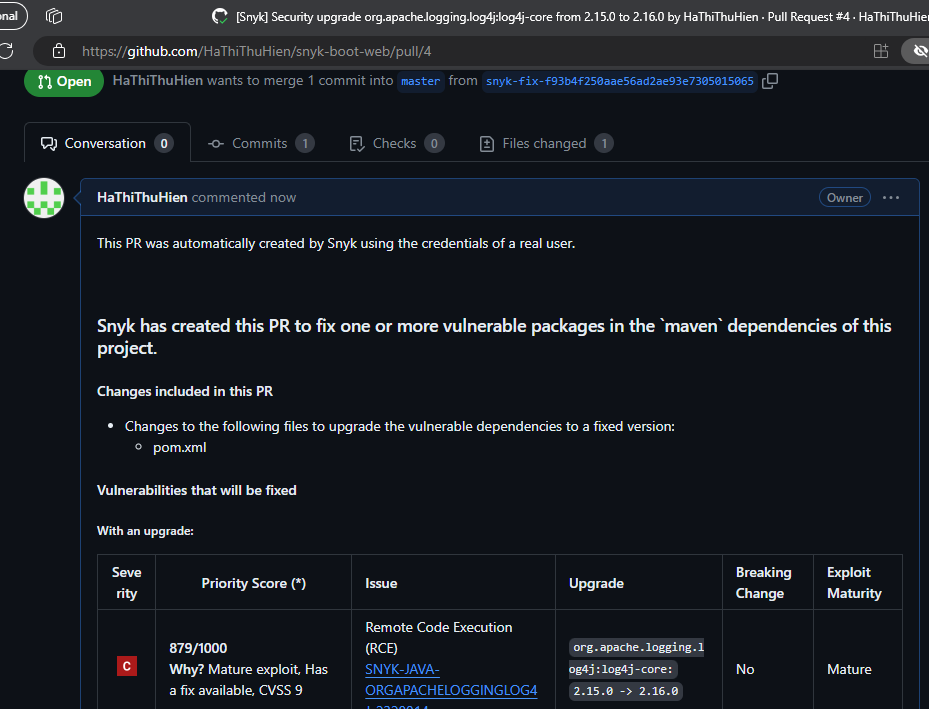


A screenshot of a computer

Description automatically generated

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





* Lúc này, một pull request mới đã được tạo, chúng ta có thể chọn các tab Conversation, Commits, Checks, Files changed để xem thông tin chi tiết về Pull Request này.
* 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

- Quay lại Snyk, kiểm tra và thấy rằng số lượng cảnh báo trên tập tin pom.xml đã giảm từ 3 Critical thành 2 Critical.



#### Snyk CLI & Snyk IDE

##### Snyk CLI

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

* Cài đặt Snyk CLI theo hướng dẫn sau: [https://docs.snyk.io/snyk-cli/install-or-update the-snyk-cli](https://docs.snyk.io/snyk-cli/install-or-update%20the-snyk-cli%20)

**A screen shot of a computer

Description automatically generated**

**A screen shot of a video game

Description automatically generated**

* Uỷ quyền cho Snyk CLI bằng cách chạy câu lệnh sau ở Terminal/CMD.

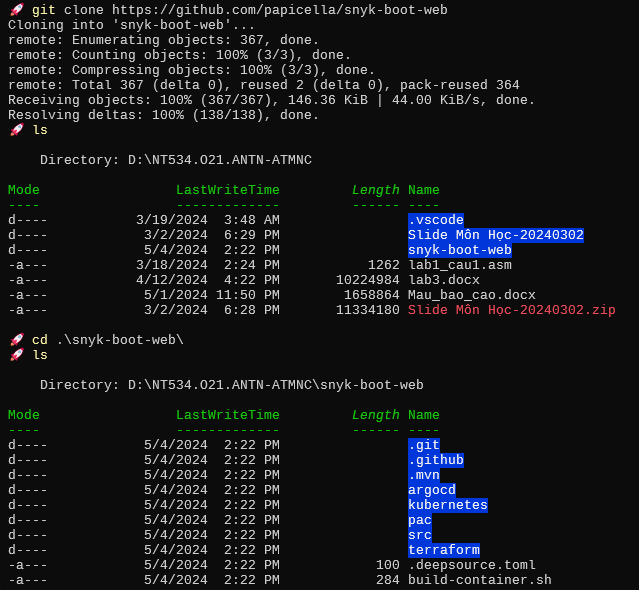
**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer error

Description automatically generated**

* Clone nội dung Webapp về máy



* Sử dụng Synk Open Source để scan manifest file

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

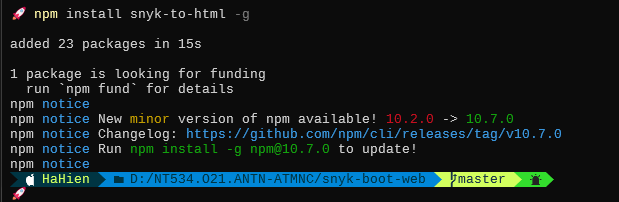
* Sử dụng Synk Code để scan source code

A computer screen shot of a program

Description automatically generatedA computer screen shot of a program

Description automatically generated

* Xuất kết quả thành file HTML. Để xuất được kết quả thành file HTML, cần cài đặt một plugin snyk-to-html ([https://docs.snyk.io/snyk-cli/scan-and-maintain-projects using-the-cli/cli-tools/snyk-to-html](https://docs.snyk.io/snyk-cli/scan-and-maintain-projects%20using-the-cli/cli-tools/snyk-to-html))



A black background with white text

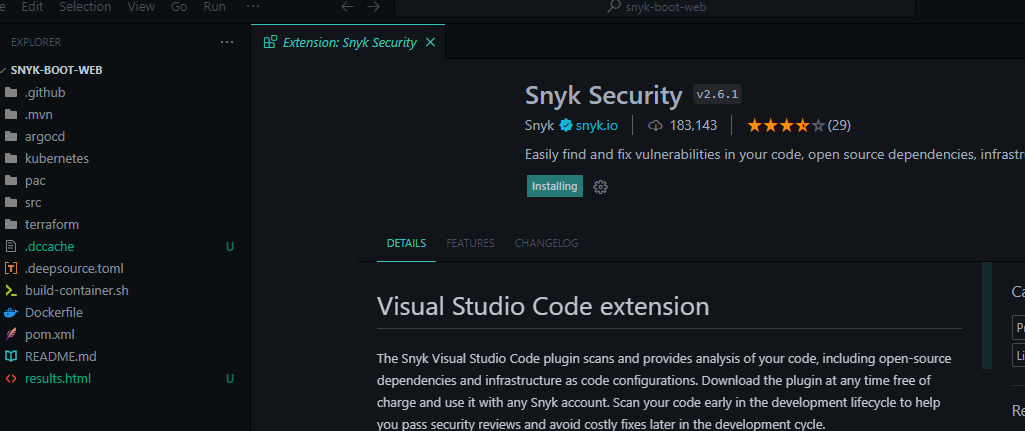
Description automatically generated

A screenshot of a computer

Description automatically generated

##### Snyk IDE

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

****

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

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

* Tạo repository mới để thực hiện task này

A computer screen shot of a computer

Description automatically generated

* Nhiều file sample ta có thể thấy ở đây.

A screenshot of a computer

Description automatically generated

* Tiến hành chỉnh code ở pre-commit.sample và đổi tên

A screen shot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

* Chạy lệnh git commit và ta có được kết quả

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

*Ví dụ: [*NT101.K11.ANTT*]-Exe01\_Group03.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)