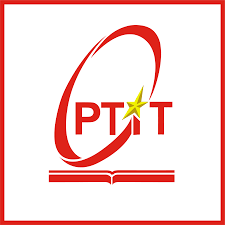
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**



**MÔN HỌC:** PHÂN TÍCH MÃ ĐỘC

**BÁO CÁO THỰC HÀNH BÀI 3**

**Giảng viên:** PGS.TS. Đỗ Xuân Chợ

**Sinh viên:** Hoàng Trung Kiên – B20DCAT098

Hà Nội – 5/2023

Mục lục

[**1. Tìm hiểu về Ghidra.** 3](#_Toc147500128)

[a, Mục đích. 3](#_Toc147500129)

[b, Lý thuyết. 3](#_Toc147500130)

[**2. Thực hành.** 4](#_Toc147500131)

[Nhiệm vụ 1: Sử dụng công cụ Ghidra. 4](#_Toc147500132)

[Nhiệm vụ 2: Tạo project và nhập chương trình cadet01. 5](#_Toc147500133)

[Nhiệm vụ 3: Tìm cổng mạng của dịch vụ. 9](#_Toc147500134)

[Nhiệm vụ 4: Tìm “Easter egg”. 11](#_Toc147500135)

[Nhiệm vụ 5:  Làm gián đoạn dịch vụ. 12](#_Toc147500136)

[**3. Checkwork.** 13](#_Toc147500137)

# **1. Tìm hiểu về Ghidra.**

## a, Mục đích.

Bài thực hành này giới thiệu công cụ Ghidra của ghidra-sre.org. Sinh viên sẽ sử dụng Ghidra để phân tích một tập tin thực thi nhằm xác định một số thuộc tính của nó.

## b, Lý thuyết.

Ghidra là một công cụ phân tích ngược mã máy mã nguồn mở được phát triển bởi Cơ quan An ninh Quốc gia Hoa Kỳ (NSA) và được công bố công khai vào tháng 3 năm 2019. Nó cung cấp một môi trường phân tích mạnh mẽ cho việc phân tích và giải mã các chương trình máy tính.

Ghidra có các tính năng chính sau:

1. Dịch mã máy: Ghidra cho phép người dùng dịch mã máy từ các tệp nhị phân và hiển thị mã hợp ngữ tương ứng. Nó có khả năng hỗ trợ nhiều kiến trúc vi xử lý khác nhau, bao gồm x86, ARM, MIPS, PowerPC và nhiều kiến trúc khác.

2. Phân tích động và tĩnh: Ghidra cung cấp các công cụ phân tích động và tĩnh để giúp người dùng hiểu cấu trúc và hoạt động của chương trình. Bằng cách theo dõi luồng điều khiển, các hàm và biến, người dùng có thể tìm hiểu cách chương trình hoạt động và cách các thành phần tương tác với nhau.

3. Hiển thị biểu đồ và tương tác: Ghidra cho phép người dùng xem và tương tác với các biểu đồ lưu đồ luồng điều khiển, biểu đồ phụ thuộc dữ liệu và các biểu đồ liên quan khác. Điều này giúp người dùng có cái nhìn tổng quan về cấu trúc của chương trình và tìm hiểu cách các thành phần tương tác với nhau.

4. Phân tích mã độc: Ghidra cung cấp các công cụ hỗ trợ phân tích mã độc và tìm kiếm lỗ hổng bảo mật. Nó cho phép người dùng tìm kiếm các mẫu và dấu hiệu của mã độc, như các lỗi tràn bộ đệm, lỗi truy cập không hợp lệ và các hành vi không an toàn khác.

5. Tích hợp scripting: Ghidra hỗ trợ việc viết và thực thi các script bằng nhiều ngôn ngữ, bao gồm Python và Java. Điều này cho phép người dùng mở rộng và tùy chỉnh các tính năng của Ghidra để phù hợp với nhu cầu của họ.

Ghidra là một công cụ mạnh mẽ và linh hoạt được sử dụng rộng rãi trong cộng đồng phân tích ngược và nghiên cứu bảo mật. Nó cung cấp một môi trường phân tích toàn diện và hỗ trợ cho việc tìm hiểu và phân tích mã máy.

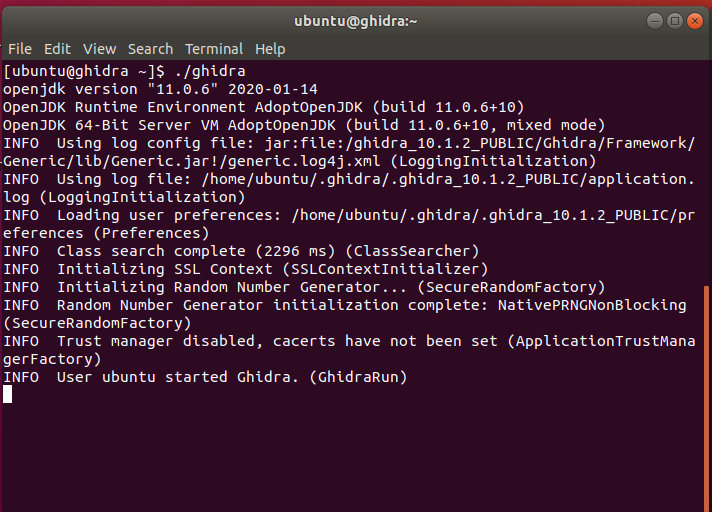
# **2. Thực hành.**

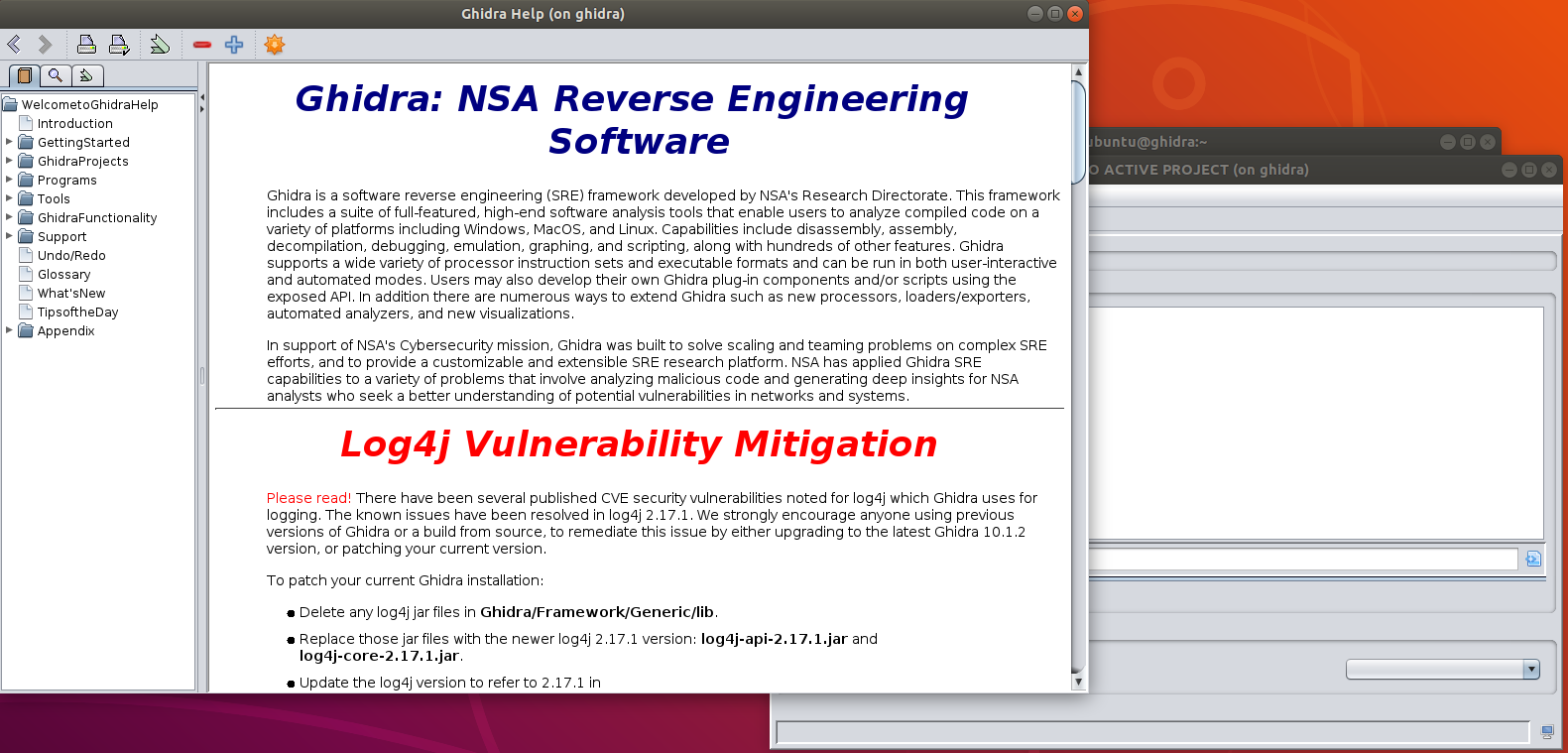
- Nhiệm vụ: Một bản sao của phần mềm dịch vụ có trong máy ghidra có tên là cadet011 trong thư mục Home. Chương trình này đang chạy trên máy chủ. Mục tiêu của bạn là kết nối đến dịch vụ; khiến nó hiển thị “easter egg”; và sau đó làm chương trình bị lỗi. Sinh viên sử dụng Ghidra để phân tích chương trình cadet01 nhằm đạt được các mục tiêu trên.

## Nhiệm vụ 1: Sử dụng công cụ Ghidra.

Khởi động Ghidra

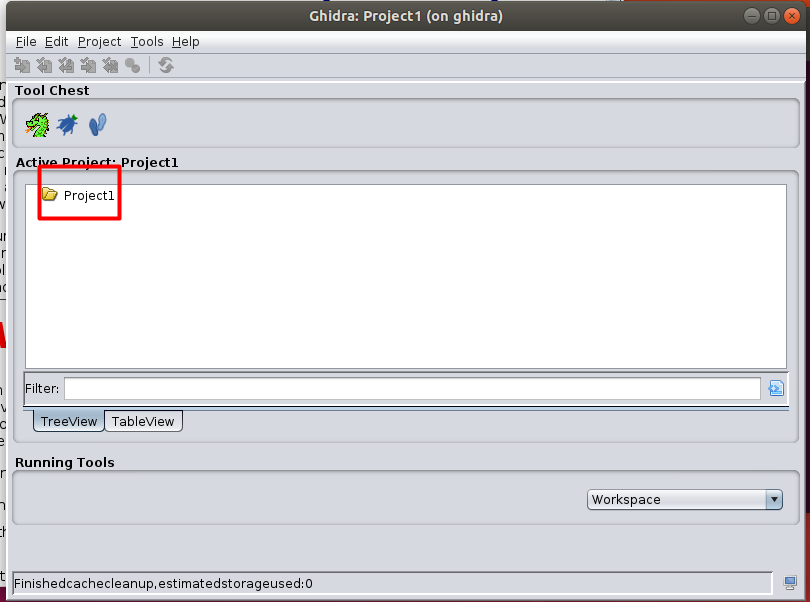
Chạy lệnh ./ghidra để khởi động chương trình



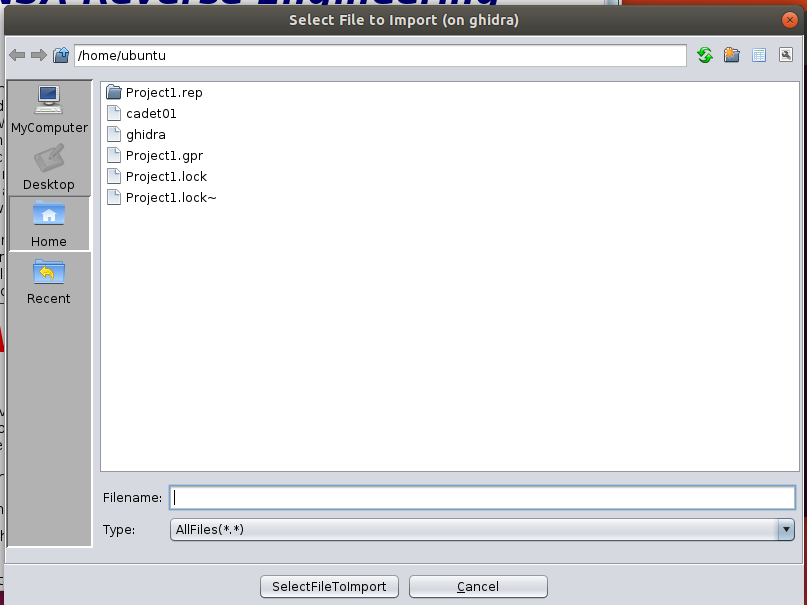


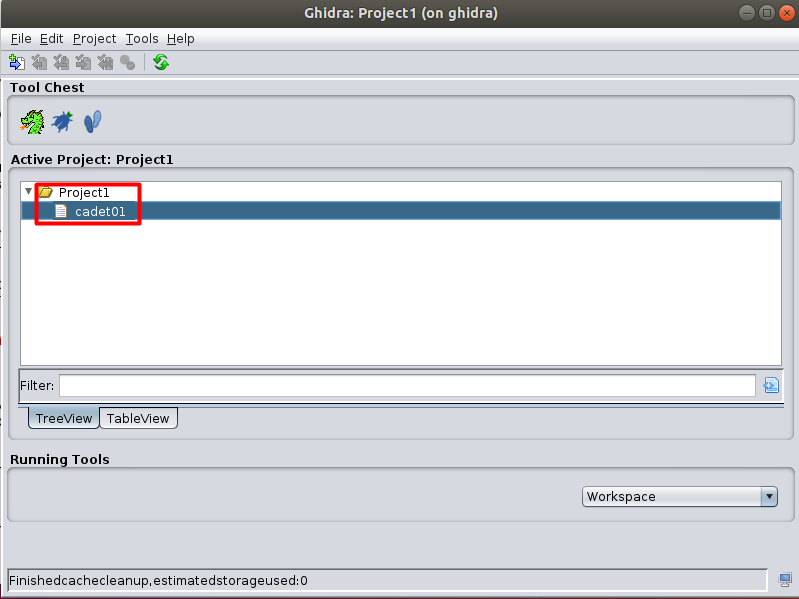
## Nhiệm vụ 2: ****Tạo project và nhập chương trình cadet01.****

Sử dụng menu File/New project trên cửa sổ chính của Ghidra để tạo một dự án mới.

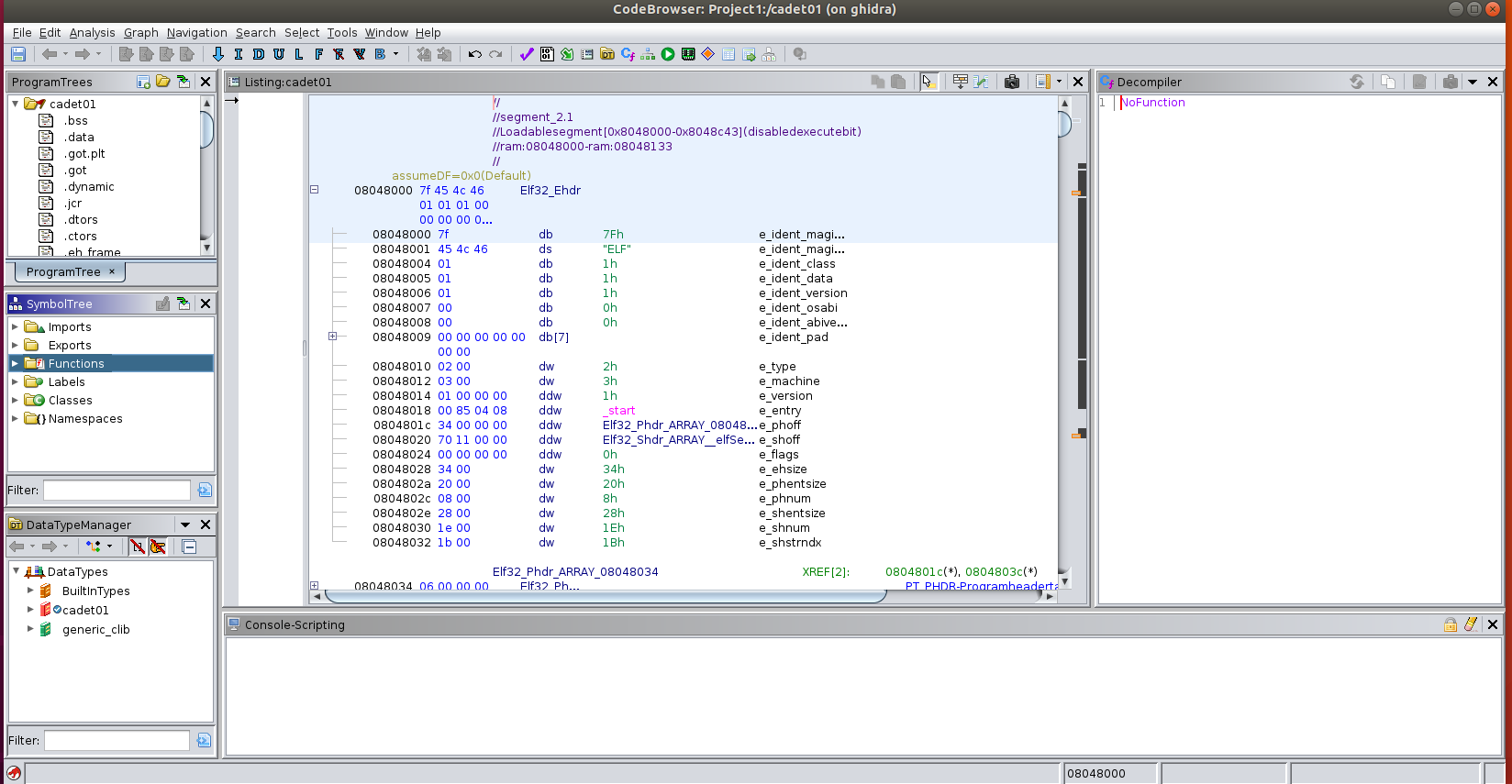


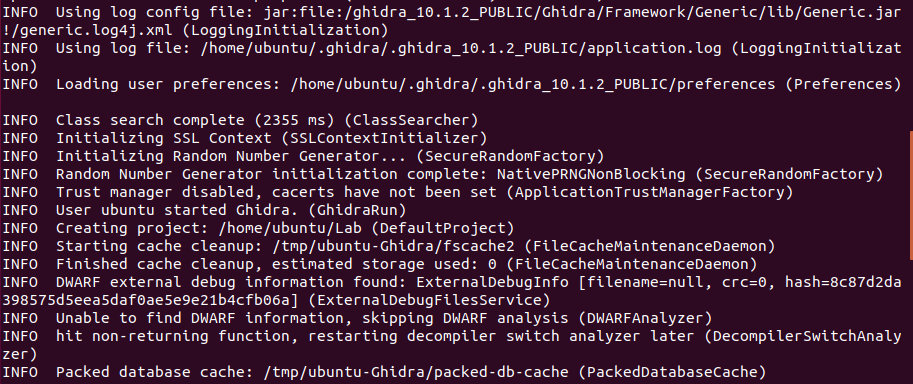
Sau đó sử dụng menu File/Import file để nhập chương trình cadet01.

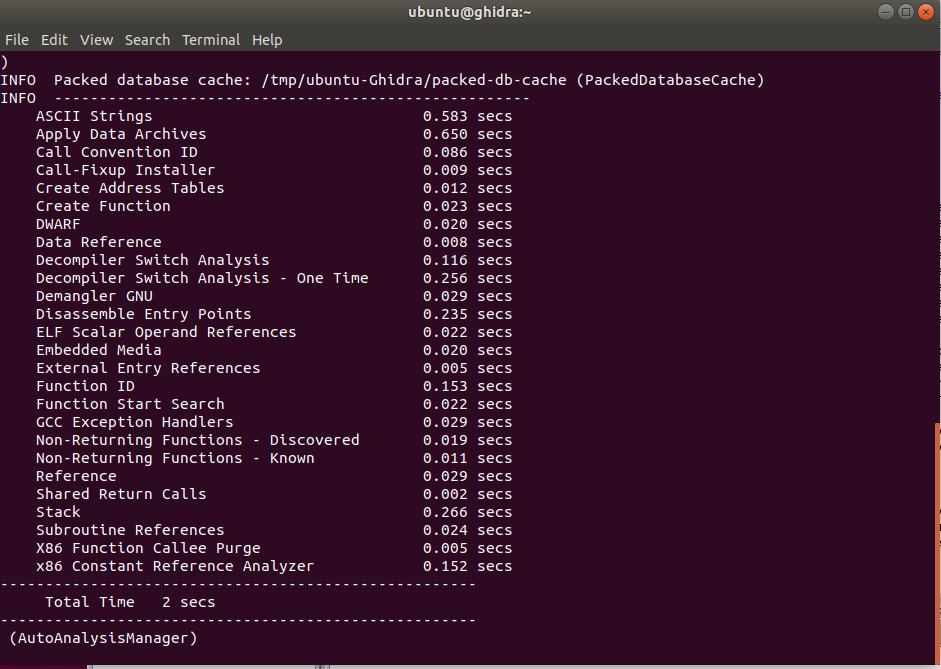




Khi yêu cầu phân tích, hãy chọn Yes và chấp nhận các phân tích mặc định.







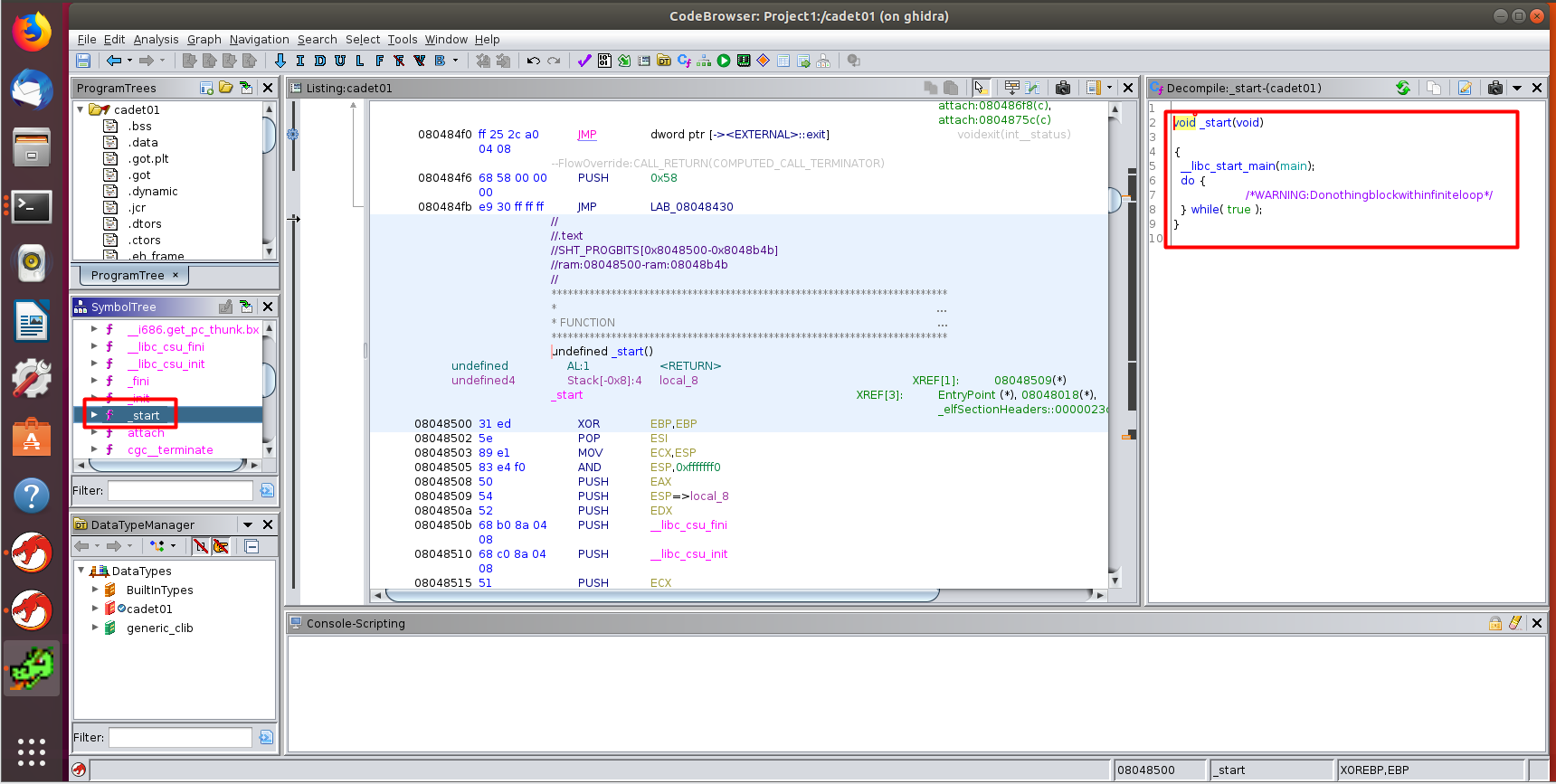
Các hàm gọi bởi hàm chính

Hàm được gọi bởi hàm main là: cgc\_terminate(), cgc\_transmit\_all(),cgc\_check(),

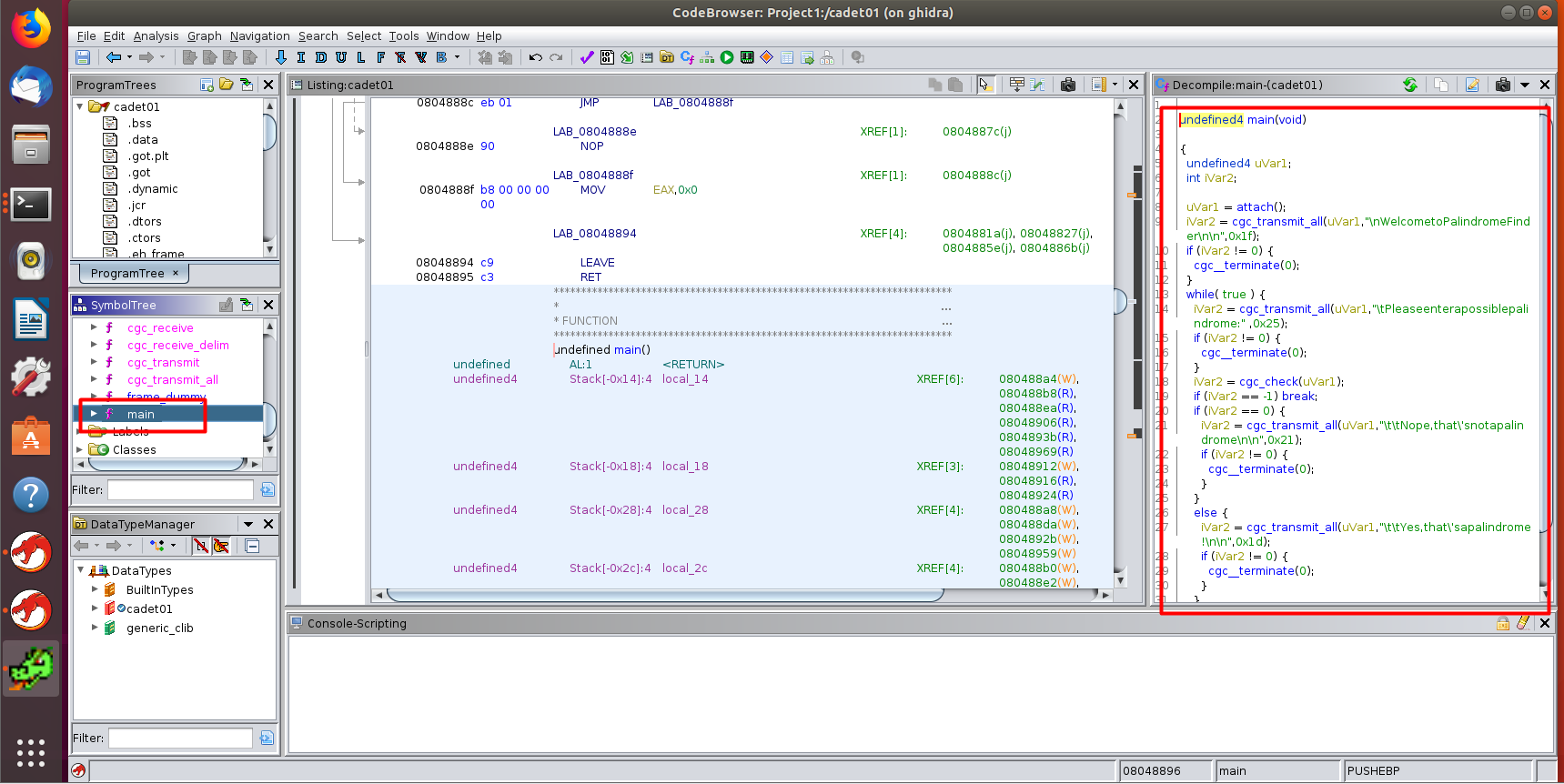
Hàm cgc\_transmit\_all() gọi hàm cgc\_transmit()

Hàm cgc\_terminate() gọi hàm exit(1)

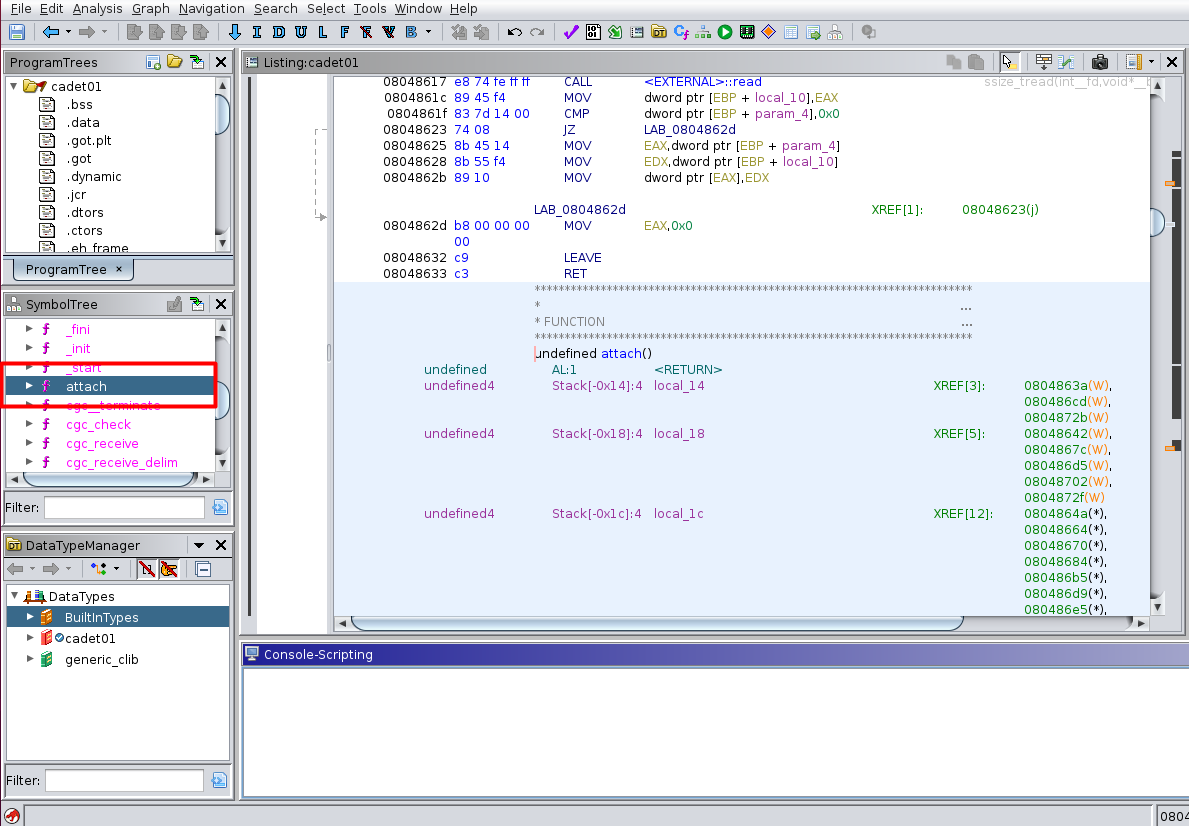
Hàm cgc\_check() gọi hàm cgc\_receive\_delim(), cgc\_transmit\_all(), cgc\_terminate()



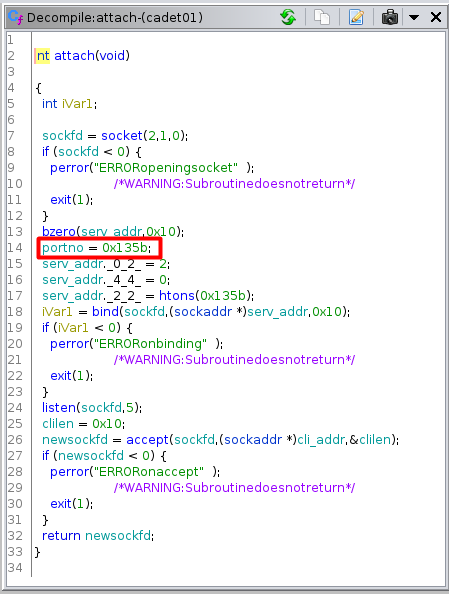
## Nhiệm vụ 3: ****Tìm cổng mạng của dịch vụ.****



Ta mở đến hàm attach



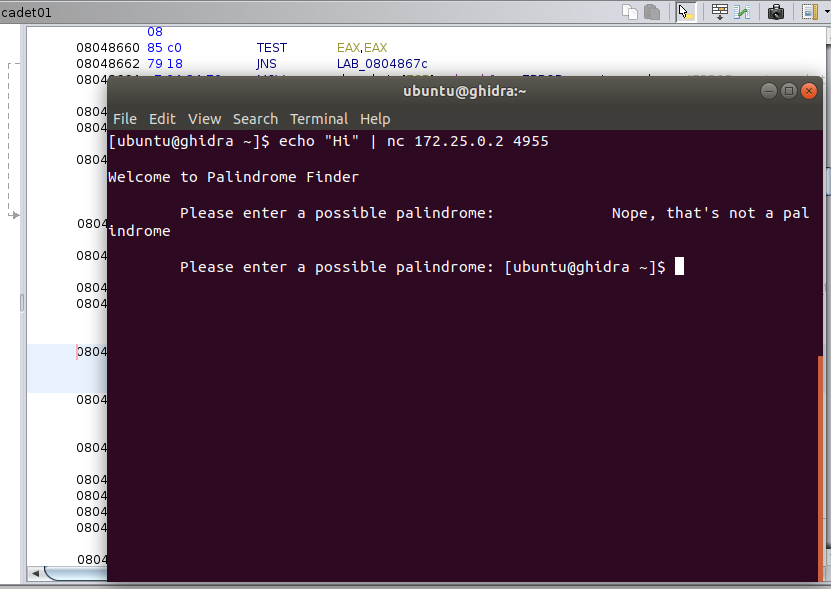
Ở bên phải ta thấy đoạn chương trình của hàm attach và thấy biến “portno=0x135b”, ta đổi sang mã nhị phân là 4955 => cổng là 4955



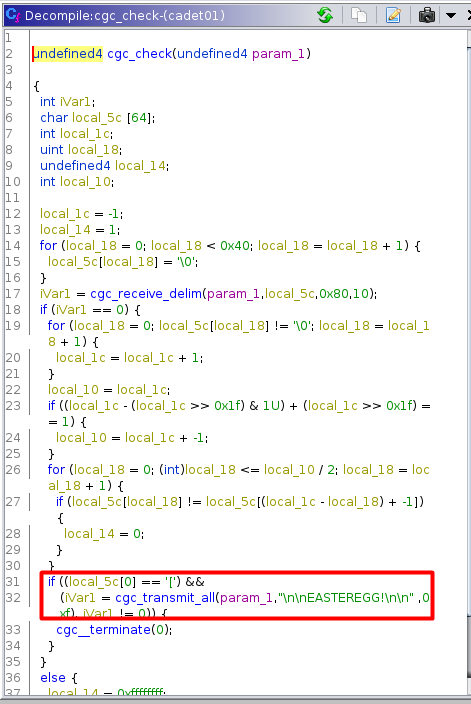
Xem qua các hàm của chương trình cadet01 để tìm số cổng mạng được sử dụng khi gắn kết với socket mạng. Sau khi đã tìm được số cổng, hãy sử dụng nó để kết nối đến server, sử dụng câu lệnh:

echo "Hi" | nc 172.25.0.2 <port number>

<port number>: số cổng tìm được là 4955. Sau khi kết nối thành công sinh viên sẽ nhận được phản hồi từ server.

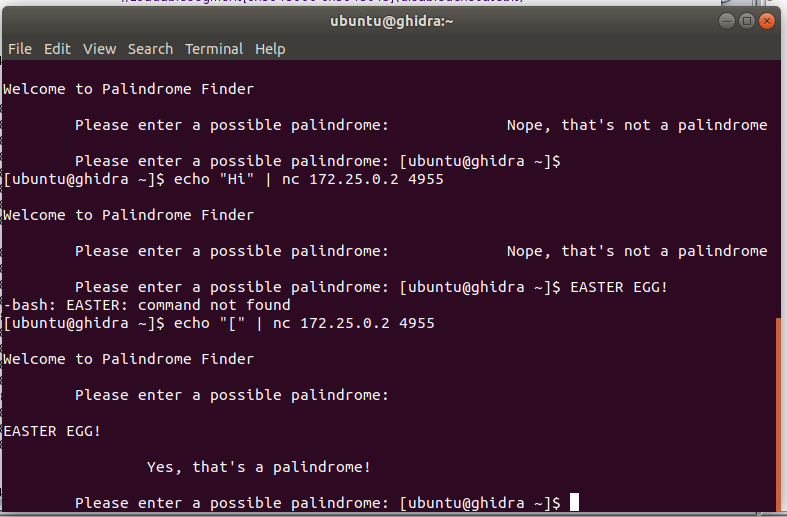
Kết nối thành công

## Nhiệm vụ 4: ****Tìm “Easter egg”.****



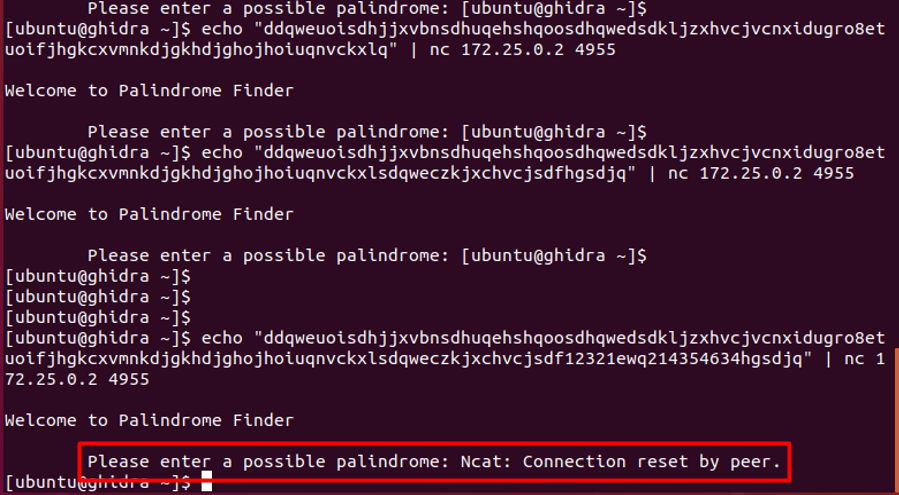
Một đầu vào cụ thể gửi đến dịch vụ sẽ khiến nó hiện “Easter egg”. Sử dụng Ghidra để xác định yêu cầu của đầu vào trong code. Sau đó dùng một chuỗi kí tự gửi đến server để hiện ra “Easter egg” này.

echo "[" | nc 172.25.0.2 4955



## Nhiệm vụ 5:  Làm gián đoạn dịch vụ.

nhập đầu vào để khiến dịch vụ bị lỗi. Hoàn thành nhiệm vụ nếu netcat hiển thị thông báo “Connection reset by peer”.



# **3. Checkwork.**

