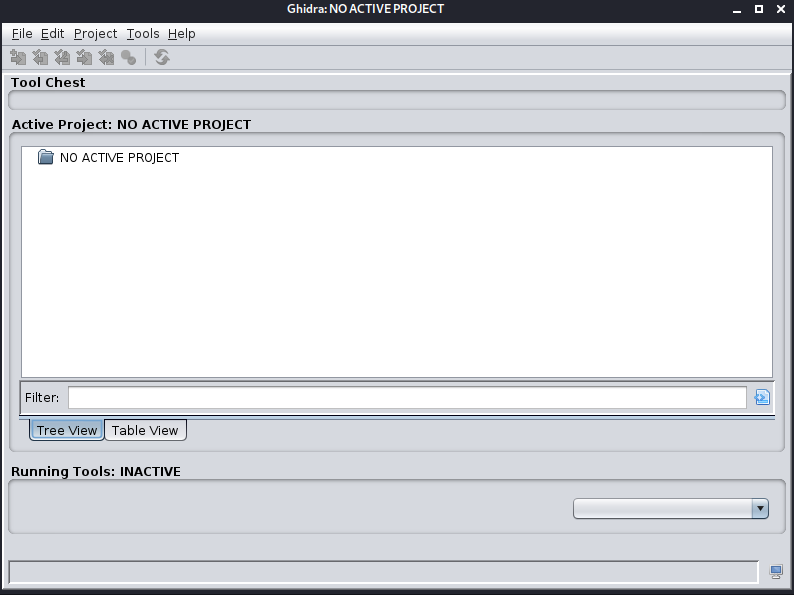
Intro

GHIDRA is a tool created by the NSA that allows the user to analyze binaries. Nó nổi tiếng vì nó là trình dịch ngược đáng kinh ngạc, cái để chuyển đổi assembly trong binary to C.  
Tải xuống GHIDRA có thể được tìm thấy ở đây (https://ghidra-sre.org/). Lưu ý: Kiến thức assembly và C được highly recommended   
Lưu ý: Ghidra requires một phiên bản bán mới của java(11). Nếu bạn không có jre hoặc jdk, bạn có thể cài đặt nó với  
sudo apt install openjdk-13-jre openjdk-13-jdk

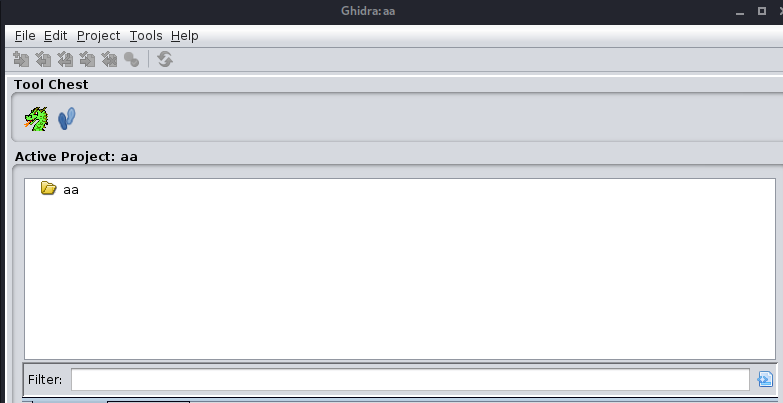
Creating A New Project

Khi bạn đã giải nén thư mục ghidra và tải xuống java, bạn sẽ thấy cấu trúc tệp tương tự như thế này.  
https://i.imgur.com/1TZ63yM.jpg

Execute the ghidraRun file by running and you will arrive on this screen, after ghidra finishes loading.  


On this screen điều hướng đề File -> New project -> Non-shared-Project.  
On that screen make a project directory, and a name, and click finish

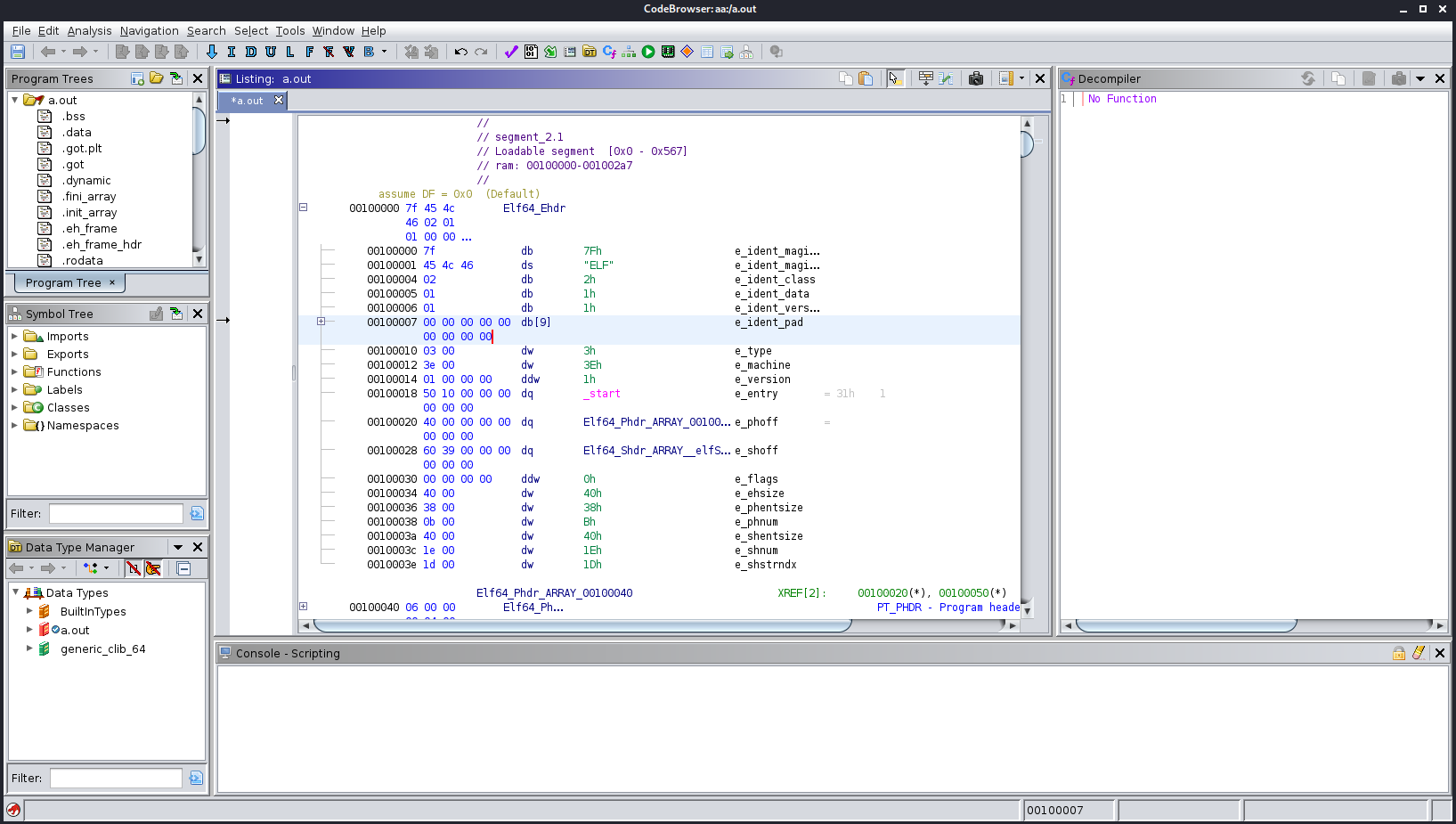
Analyzing a Binary

After creating a project, you will be chào đón with a screen similar to this  


Khi bạn ở màn hình này, go to File -> Import File. Form there, điều hướng nơi bạn đã tải xuống, bao gồm binary ‘example1’, and import it.  
(Note: Ghidra will automatically phát hiện what type on binary the file is (Example x86, x86\_64). )

From there, double click the binary, and you will be thúc đẩy to analyze the binary, click Analyze -> Yes, and you will see a screen similar như này xuất hiện

(Note: Ghidra has rất nhiều tùy chọn tùy chỉnh for analyzation, but nhưng để sử dụng chung, tùy chọn mặc định được chọn cung cấp tất cả các tính năng cần thiết.)



This is the main screen for analyzing binaries, and it allows you to see every part, of the binary.

Để sử dụng thực tế, phần thú vị nhất của màn hình này là 'Symbol Tree'. Nó cho phép bạn xem tất cả files đã import to creat the binary và view all user created function.

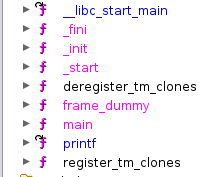
For example, given the C code:

#include <stdio.h>

int main(){

printf("hi!");

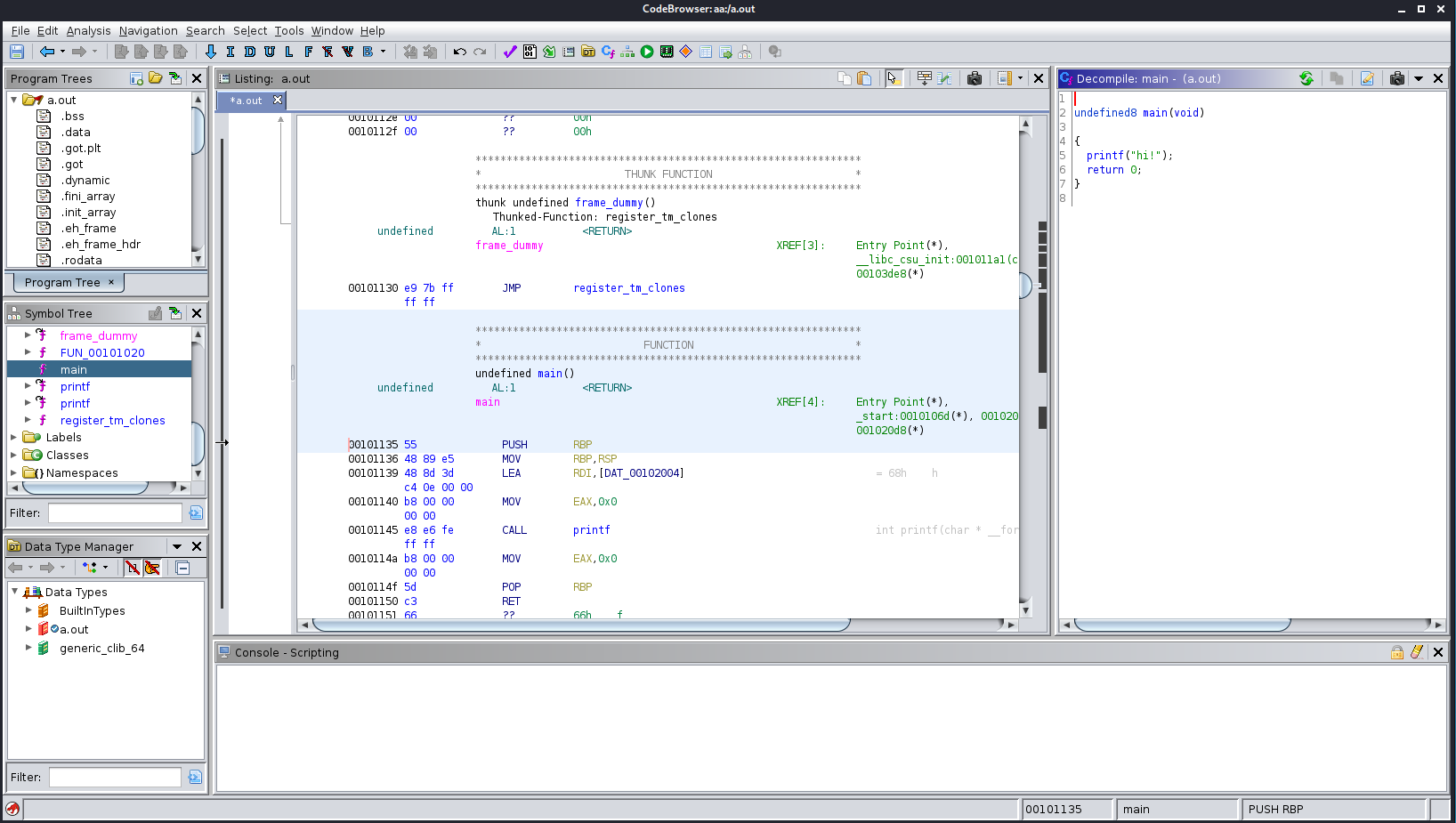
}

Bạn sẽ có thể nhìn thấy main và printf trong tab functions của Symbol Tree.  


(Note: for general binary analysis, bạn sẽ không quan tâm đến các function bắt đầu với '\_' vì đó là những hàm được tạo ra trong quá trình biên soạn(compilation))

Điều này cực kỳ hữu ích, vì nó cho phép bạn xem các functions mà the binary creator làm.

In the example1 binary, double click the “main” function and you should be prsented(trình bày) với màn hình này.



Bạn có thể nhìn thấy cả the disassembly, và the decompilation, trên màn hình giống nhau!   
From here, bạn có thể nhấp vào xung quanh tất cả các function khác nhau mà a binary có, và xem C code và assembly.

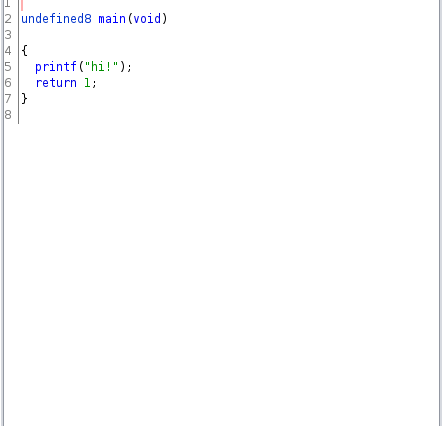
Các hoạt động khác

Trong khi các nhiệm vụ trước đó, đã cho thấy hầu hết những gì bạn cần để analyze binaries. Vẫn còn một số mẹo và thủ thuật hữu ích để biết.

**Section 1 – Patching Binaries**

Đôi khi sẽ có những lúc bạn muốn patch(vá) (The art of changing assembly instructions (nghệ thuật thay đổi…)) một binary. Ghidra cung cấp hỗ trợ cho điều này. In the example binary, điều hướng đến main function và double click 'return 0', trong phần decompilation. Bạn sẽ thấy màn hình này.



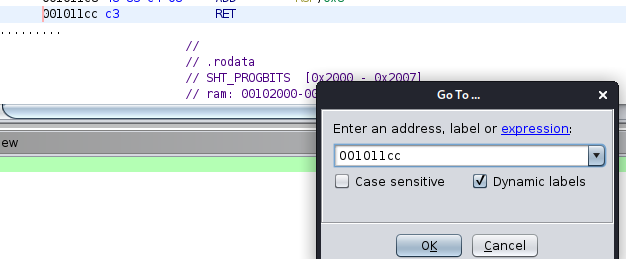
Chuột phải vào the asm instruction, MOV EAX,0x0 và click ‘Patch Instruction’. From there, you can change it thành bất cứ điều gì bạn muốn. Trong trường hợp thay đổi ‘MOV EAX,0x0’ thành ‘MOV EAX,0x1’. Việc decompilation sẽ cập nhật và bạn sẽ thấy màn hình này.  


Các instruction bạn vá có ảnh hưởng(effect) trong việc decompilation, bởi vậy It’s useful for checking your work.

**Section 2 – Searching**

Ghidra supports đi đến các phần khác nhau của memory, khi được cung cấp a memory address. Bấm Navigation(Ở bar the top)->Go to và input a memory address.

Example:



Lưu ý: Ghidra hỗ trợ misc operations; Tuy nhiên, đây là những thứ tôi đã sử dụng nhiều nhất.

For a full list see this: <https://ghidra-sre.org/CheatSheet.html>