Hack The Box Netherlands

2nd Wednesday Hack-day meetup

Welcome!

300

Members in our group

Before we start

- When presentations are going, please **mute your microphone** (we all love the sound of mechanical keyboards, but it makes it hard to hear people)
- If you have a **question**, type it in the **chat**
- Portions of this meetup will be **recorded**

Discord Community

https://discord.gg/pWh7DSY

Slides and video

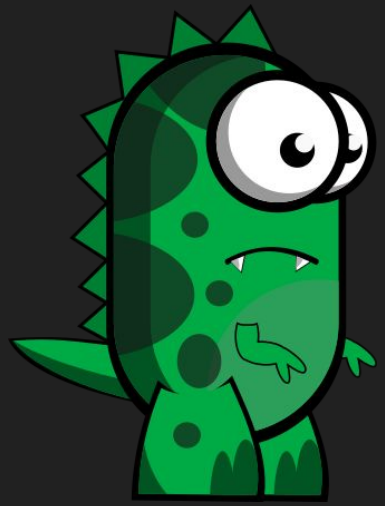
You can download these slides from our companion website

https://hackdewereld.nl

What to expect

- Today we have a presentation: **Just Enough Reverse Engineering (1 hour)**

- After this presentation you will have a choice:
  - Participate in a Reverse Engineering **workshop**
  - Hack on some machines in our **dedicated lab**



Just Enough Reverse Engineering

to score those Challenge points!

Arjen / credmp

Cyber Security Faculty of NOVI University of Applied Science

KASPERSKY lab

Certified Malware Reverse Engineering Instructor

Goal for today

- Introduce you to the exciting field of Reverse Engineering
- Show you several tools that form the basis of your RE toolkit
- Teach you a trick or 2 that are used by CTF challenge makers

What is reverse engineering?

the processes of extracting knowledge or design information from anything man-made

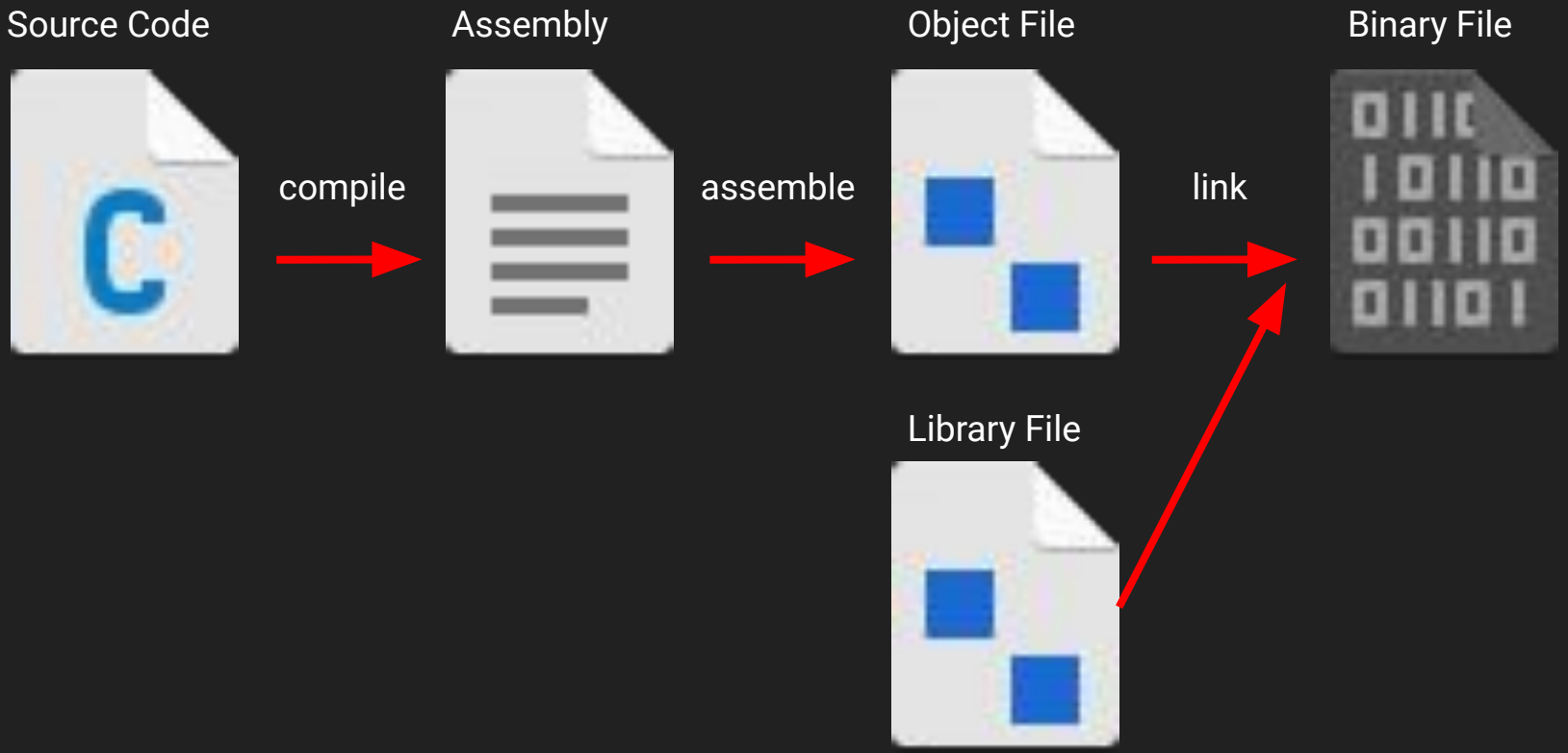
- Gives you a great understanding of how computer systems work
- Great fun if you like puzzles
- Allows you to peek “under the hood” of any product

Hack The Box Challenges

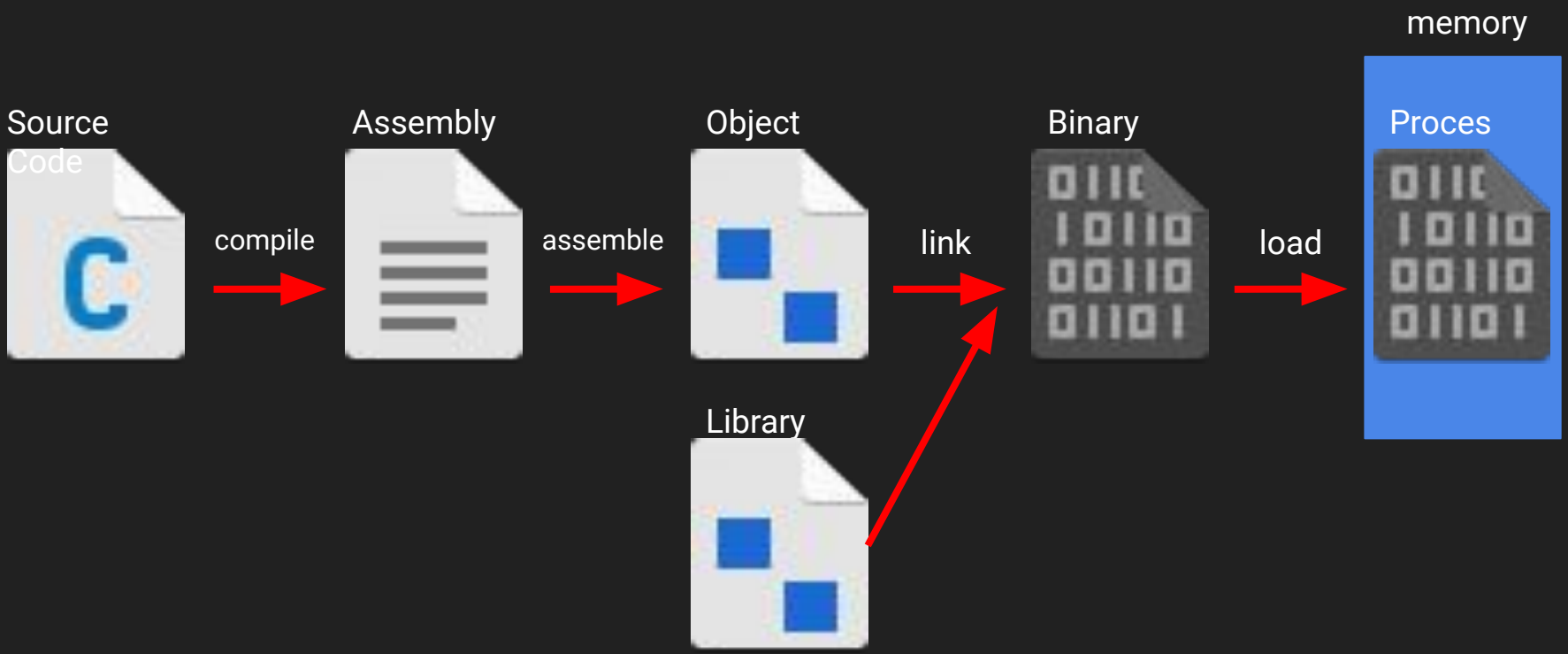
- Reversing is one type of challenge
- Today you will learn enough to:
  - Complete the easy challenges
  - Get started on the harder ones

What are binaries?

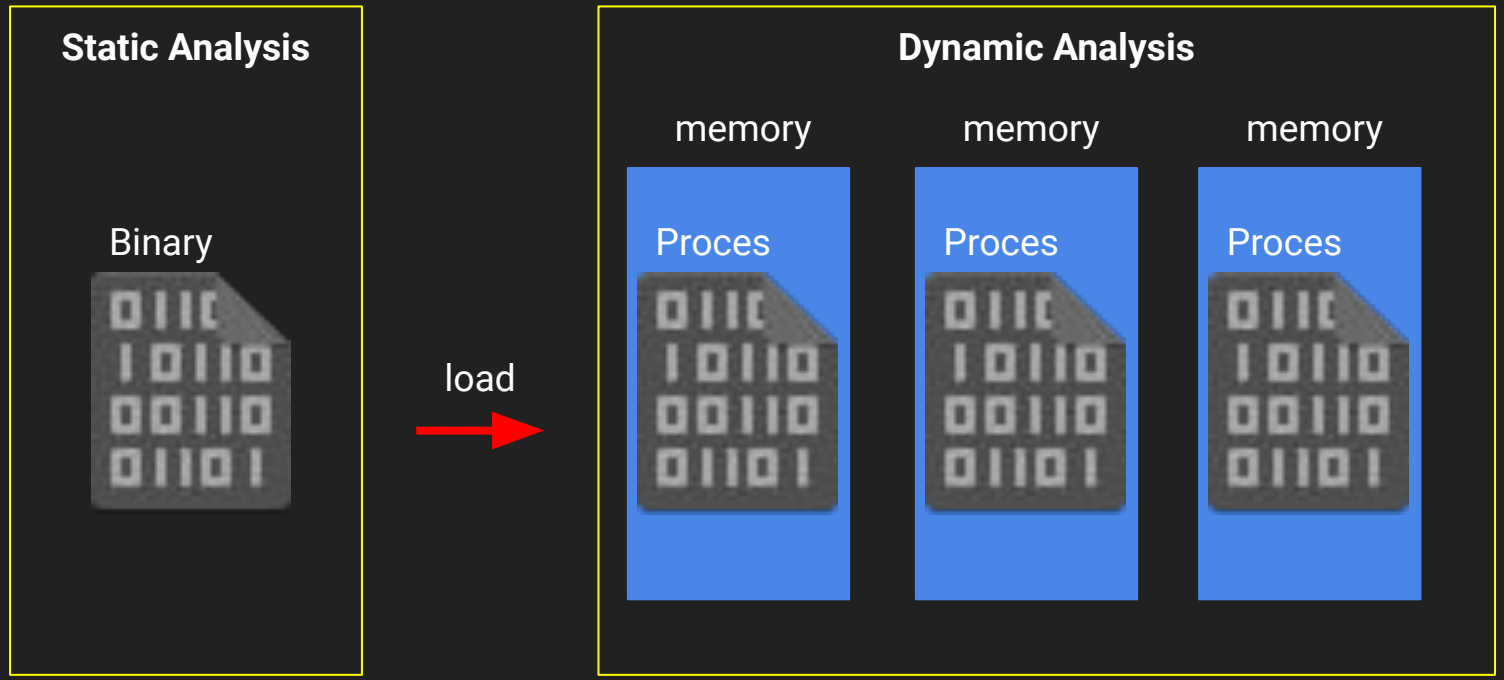
How binaries are created



How binaries are created



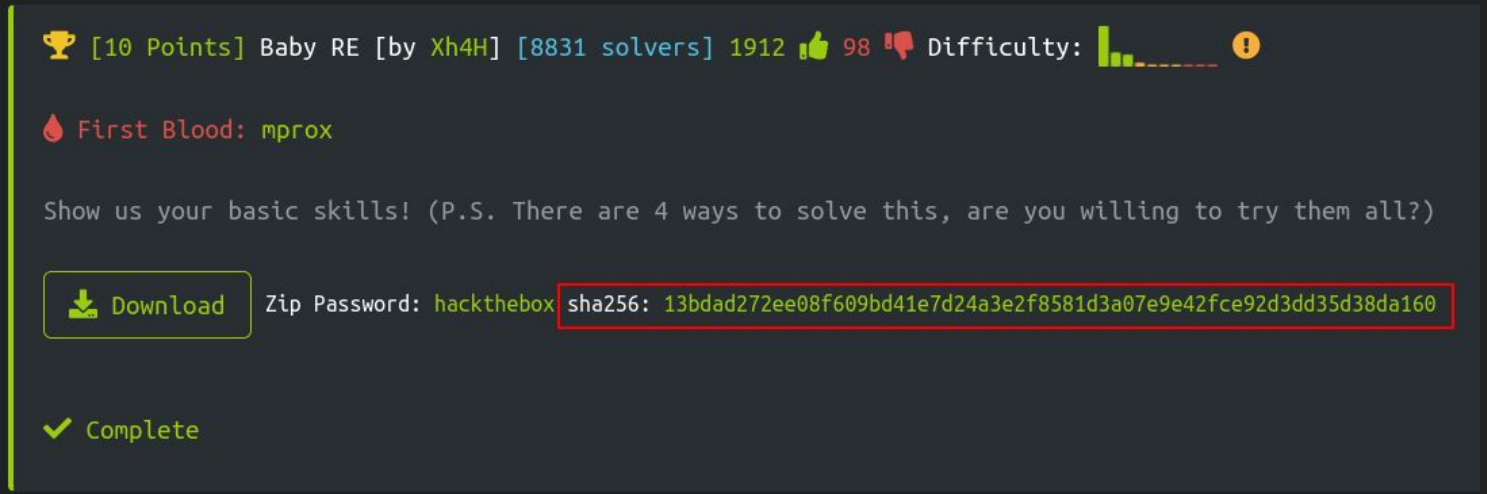
Reverse Engineering Domain



Static Analysis

Do we have the right binary?

- Ensure the zip file is correct by comparing hashes
- Tool: **md5sum** or **sha256sum**



Unpacking on Kali

- Use 7z to extract zip archives

```
❯ 7z x Baby_RE.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.utf8,Utf16=on,HugeFiles=on,64 bits,1

Scanning the drive for archives:
1 file, 2885 bytes (3 KiB)

Extracting archive: Baby_RE.zip
--
Path = Baby_RE.zip
Type = zip
Physical Size = 2885


Enter password (will not be echoed):
Everything is Ok
```

Tool: file

- used to determine filetype
- with binaries it shows so much more!
  - Executable type (ELF, PE, etc)
  - Architecture
  - Endianess
  - Statically or Dynamically linked

```
❯ file baby
baby: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=25adc5
3b89f781335a27bf1b81f5c4cb74581022, for GNU/Linux 3.2.0, not stripped
```

Tool: strings

- list printable strings in a binary
- allows you to determine program functions
- will show a lot of small strings as well, so need to filter

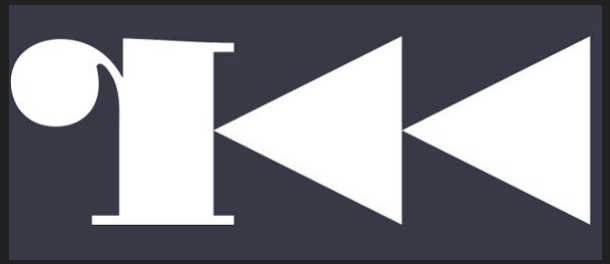
```
❯ strings -n 10 headache
/lib64/ld-linux-x86-64.so.2
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
[]A\A]A^A_
a15abe90c112d09369d9f9da9a8c046e
Initialising
Enter the key:
Login Failed!
Login success!
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0
HTB{not_so_easy_lol}
GCC: (Debian 8.3.0-19) 8.3.0
```

Is it packed? (Tool: upx)

- Packing “hides” the actual binary inside the binary
- It is only loaded at runtime and you can not examine it without unpacking
- Most popular packer is **upx**
- **upx -l <filename>** will show if the binary is packed

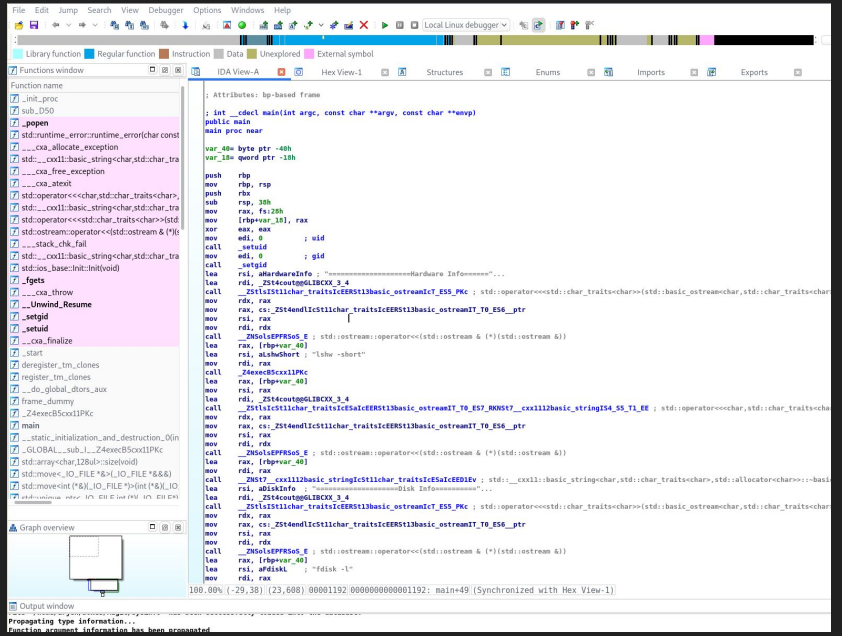
```
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
   2202568 ->    709524   32.21%   linux/amd64
```

Disassembler



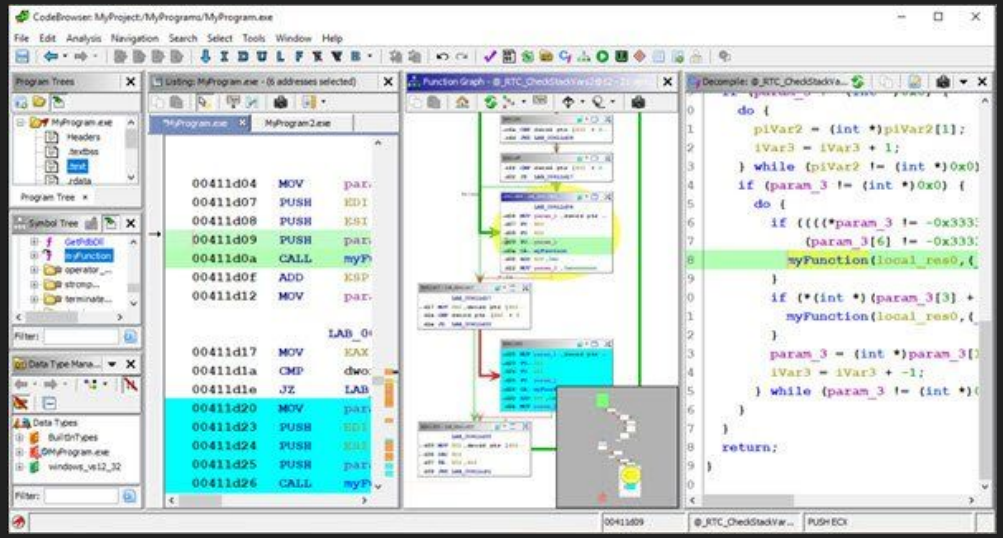
IDA Pro / Freeware

- Commercial tool from Hex-Rays
- Loved in the industry, but very expensive!
- Comes with a debugger



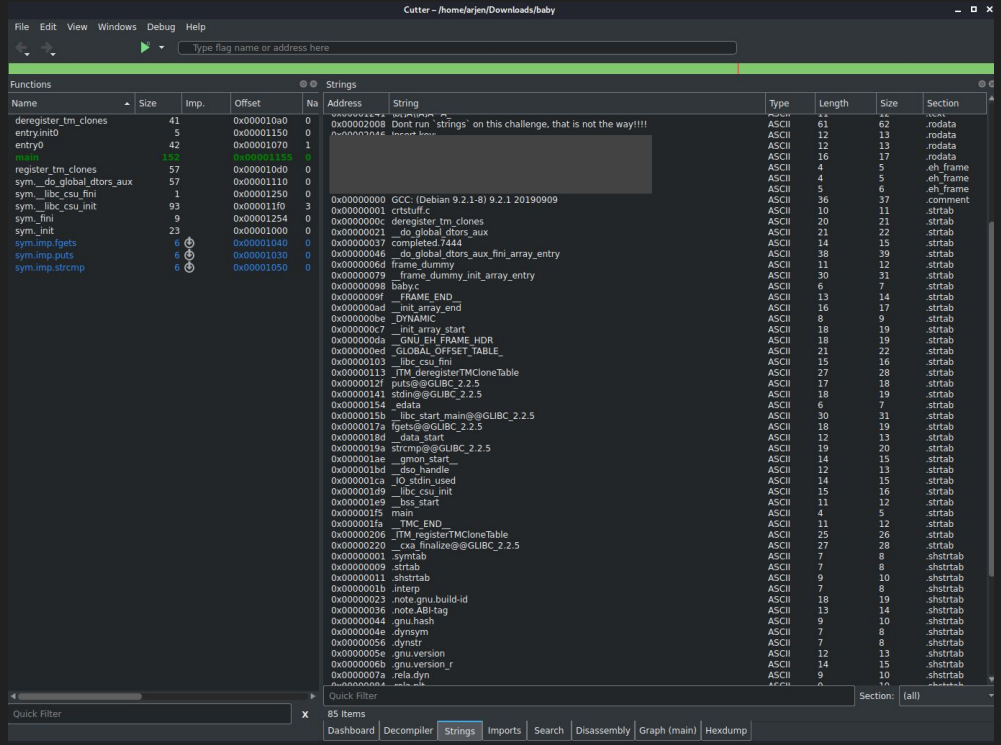
Ghidra

- Developed by the NSA
- Has been open sourced
- Not the most intuitive, but very powerful
- Comes with a **decompiler**!!!!
- But sadly no debugger yet....



Radare2 / Cutter

- Libre and portable reverse engineering framework
- More than just a disassembler
- Cutter (the GUI) now includes the **Ghidra decompiler**
- Features a **debugger** (Beta)



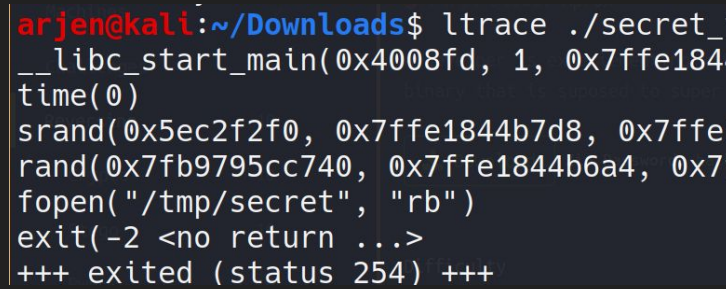
My advice

- When you get started, use **Ghidra**
- Examine the other tools once you finish the challenges (IDA / Cutter)
- Invest some time in learning 1 of these tools very well

Dynamic Analysis

Tracing

- Tool: ltrace
  - Traces library calls performed by the binary
  - C functions such as **puts, fgets, strcmp**



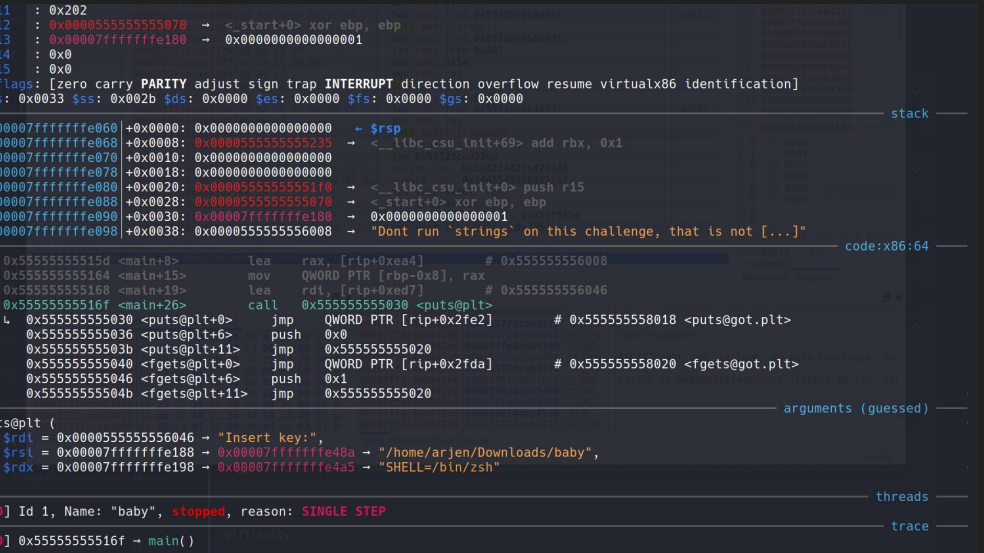
- Tool: strace
  - Traces system calls and signals performed by the binary
  - functions related to **memory management, file operations**

Debugger

- Use a debugger to **step through** the program
- You can also manipulate the program by **changing registers**
- Examine / dump memory of the proces
- Use IDA or Cutter or a dedicated debugger

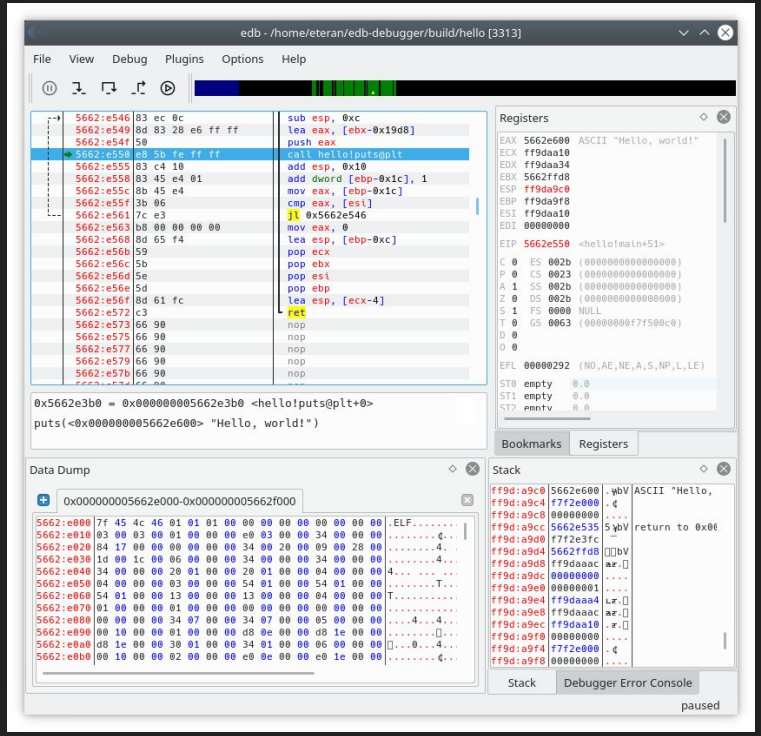
gdb

- The OG debugger under linux, console based
- Extendable, example of this is **GEF** (https://gef.readthedocs.io/en/master/)
- Extremely powerful
- Hard to learn



edb - Evan’s debugger

- GUI based debugger
- Great alternative to GDB for most cases



CTF Thing to know

XOR

- CTF developers love to XOR flags
- Operation on a binary level
- Output true (1) only when the values differ
- Easily implemented in any programming language (^)

0 0 1 1
0 1 0 1
0 1 1 0
0 1 0 1
0 0 1 1

Wrapping up

Where to go from here?

- Reversing challenges on Hack The Box
- https://beginners.re/ a guide to reverse engineering examining code and assembly
- https://malwareunicorn.org/#/workshops great workshops you can follow at home

Additional challenges

- Advanced: http://www.flare-on.com/ a yearly competition for RE
- http://crackmes.cf/ archive of 1000s of crackme challenges

You now know...

- how to perform basic static analysis
  - **sha256sum**, **file**, **strings**, **upx**, **Ghidra**

- how to get started with dynamic analysis
  - **ltrace**, **strace**,, **edb**

- how a **XOR** operation works
- **you can grab some challenge points!**

Thanks! Questions?

Stick around

- There will be 2 breakout rooms
  - 1 to get started reversing
  - 2 to do some hacking on HTB
- **credmp** will host the reversing breakout
- **DutchPyro** will host the HTB hacking breakout

- Let us know **in the chat** where you want to go and what your **HTB Username** is for the dedicated server!

Next Meetup

June 17th (6pm to 9pm)

Workshop

Challenges to try

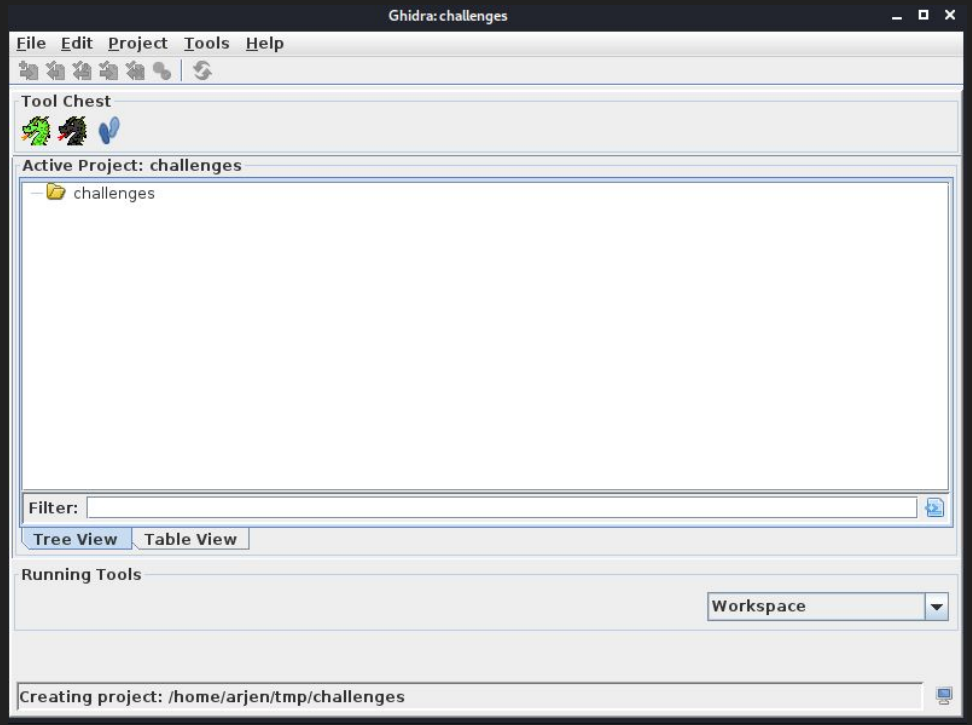
- BabyRE (easy)
- Exatlon (easy)
- Impossible Password (easy)
- Find the Secret Flag (medium)

Ghidra

- Download from https://ghidra-sre.org/
- Extract in your Downloads directory
- Run ghidraRun to start it

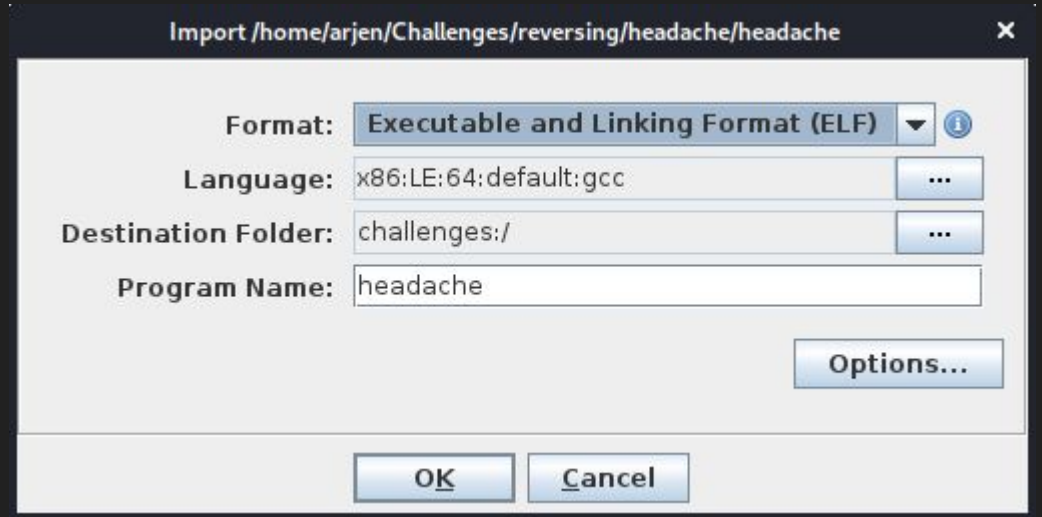
Ghidra first step

- File > **New Project**
- Give it a name (**challenges**)
- Click on **Finish**



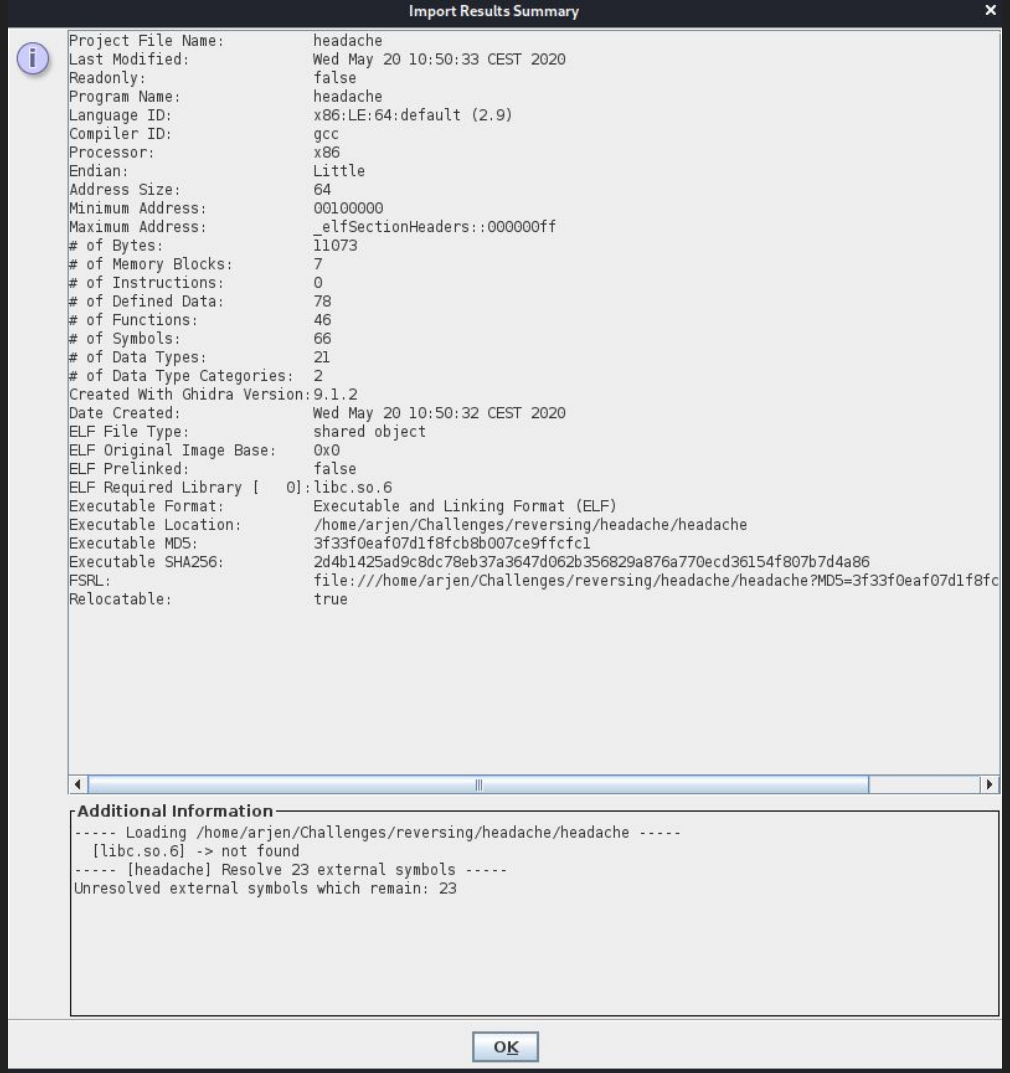
Add a binary

- File > Import File
- Select the unpacked challenge file
- Hit Ok in the popup dialog



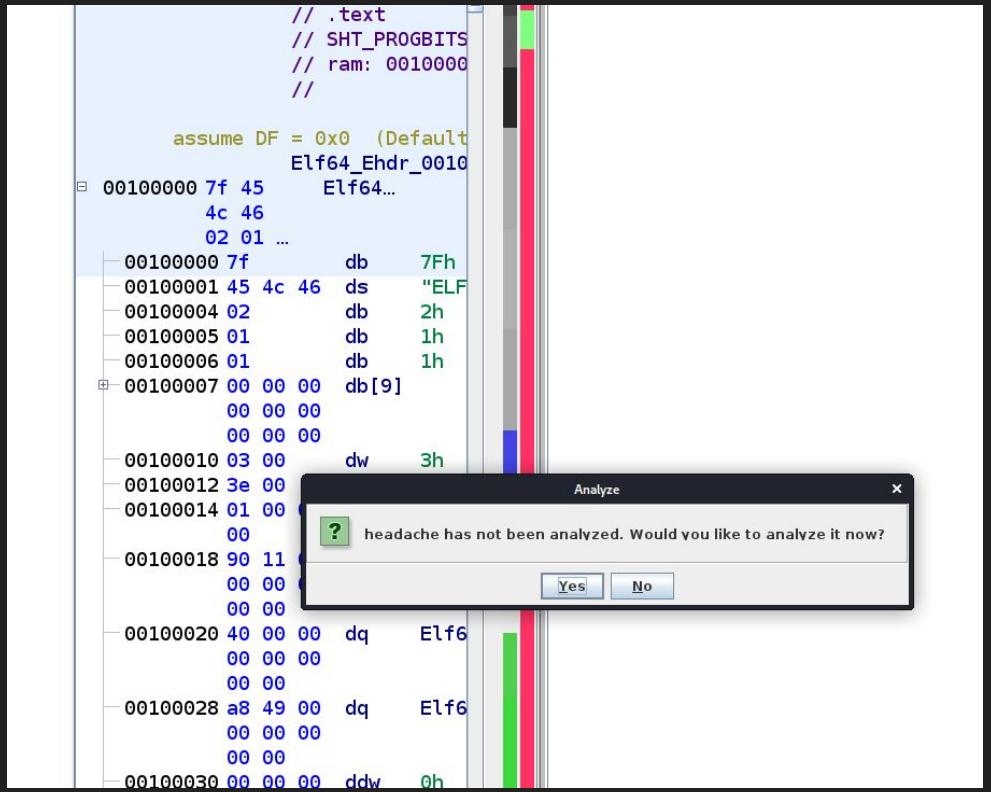
Details

- Scan it, hit **OK**



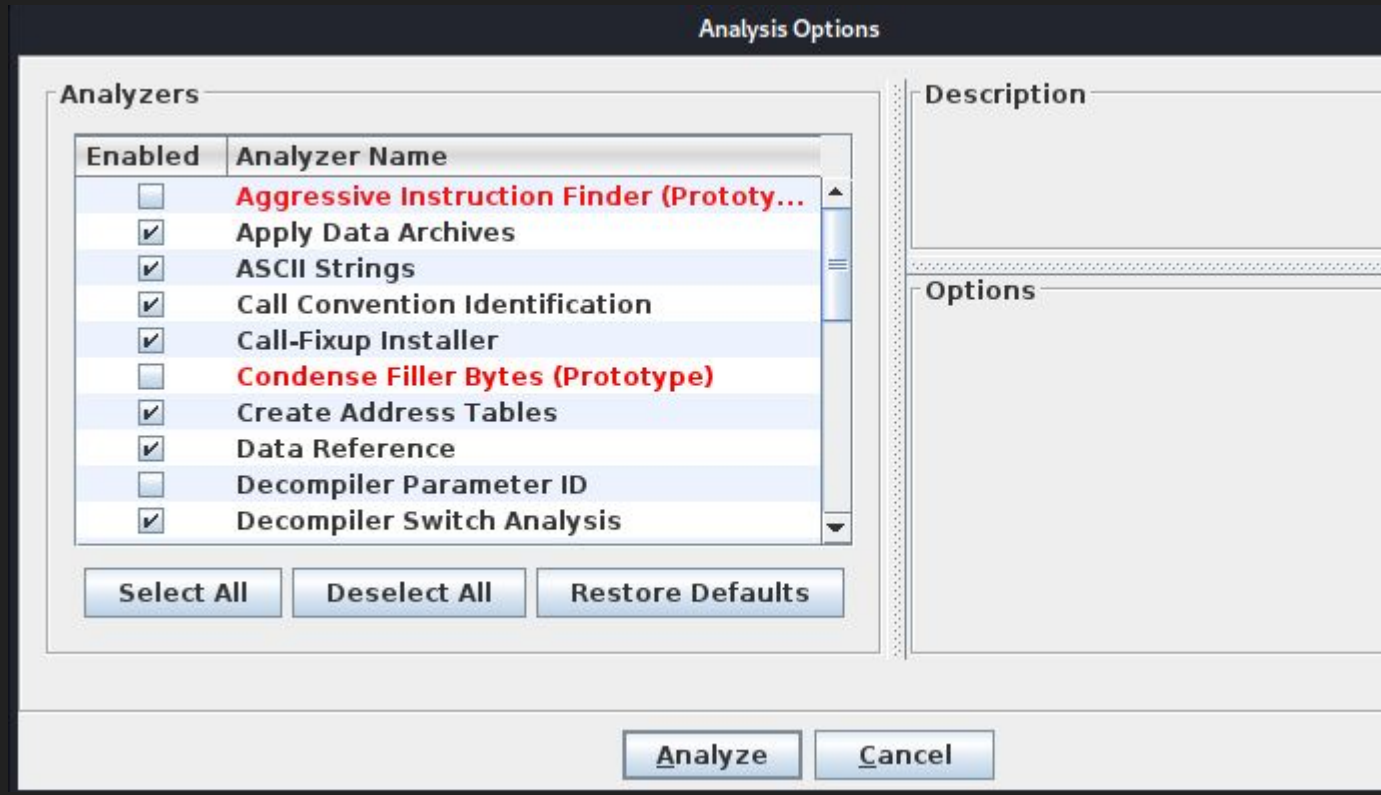
Start analyzing

- Double click on the binary
- Select Yes to analyse your binary



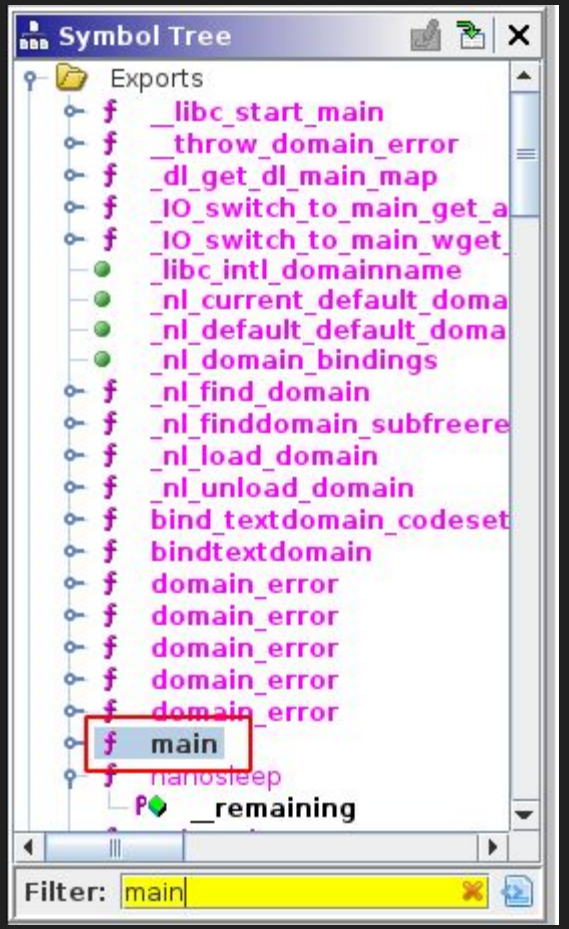
Analyze!

- Hit Analyze



Find the entrypoint

- In the **Symbol Tree** expand the **Exports**
- Search for something like **entry** or **main**
- Start your quest here.



Start your journey...