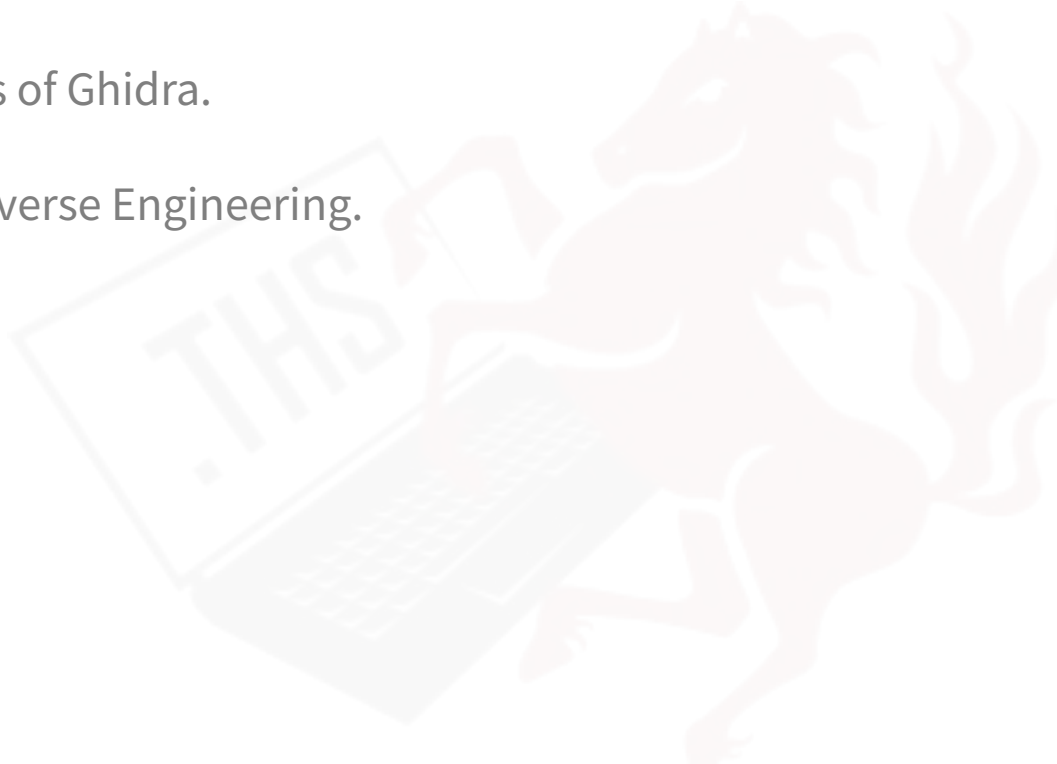
Binary Reversing with Ghidra

Jerre Starink
2023-05-22

Today

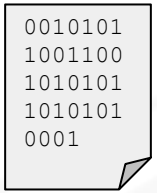
- Reversing / solving a challenge based on real Malware.
- Exploring commonly used features of Ghidra.
- Discussing various strategies in Reverse Engineering.



Tools we will be using

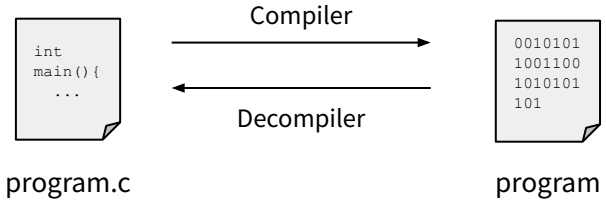
- Wireshark (https://www.wireshark.org/)
- Ghidra (https://ghidra-sre.org/)
- Python (https://www.python.org/)

What is this program doing?



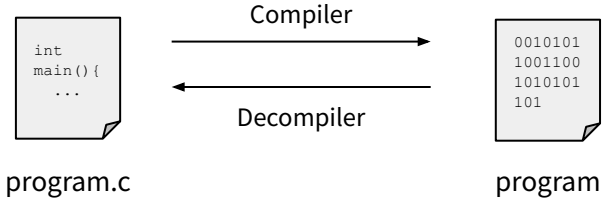
program

Two main strategies

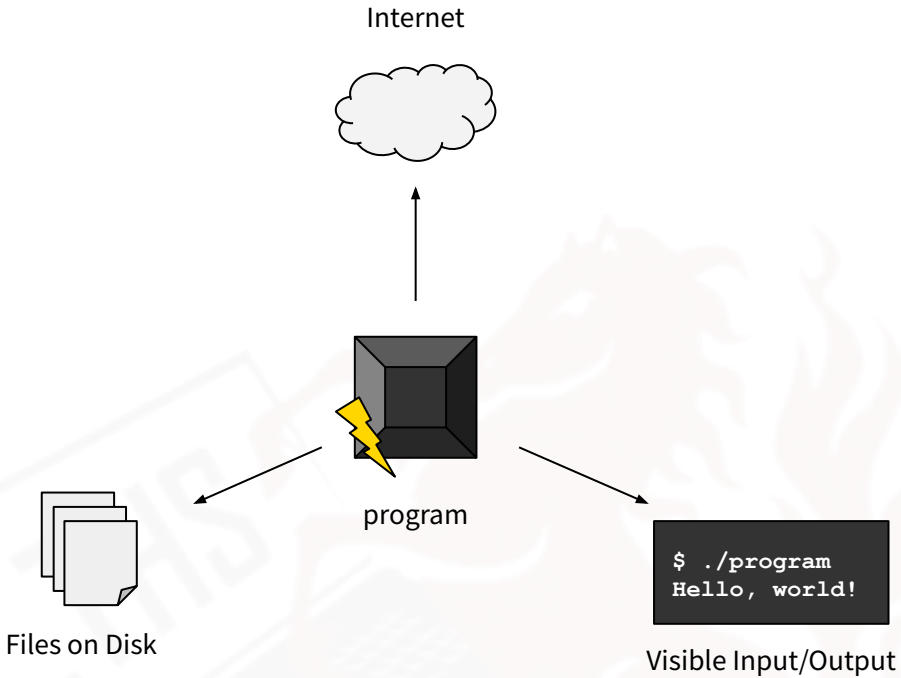


Static Analysis

Two main strategies

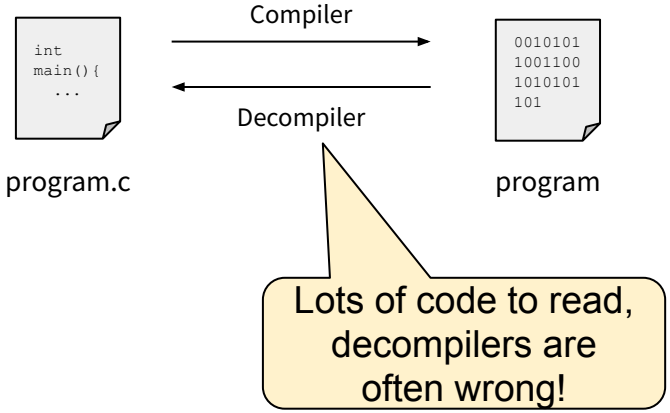


Static Analysis

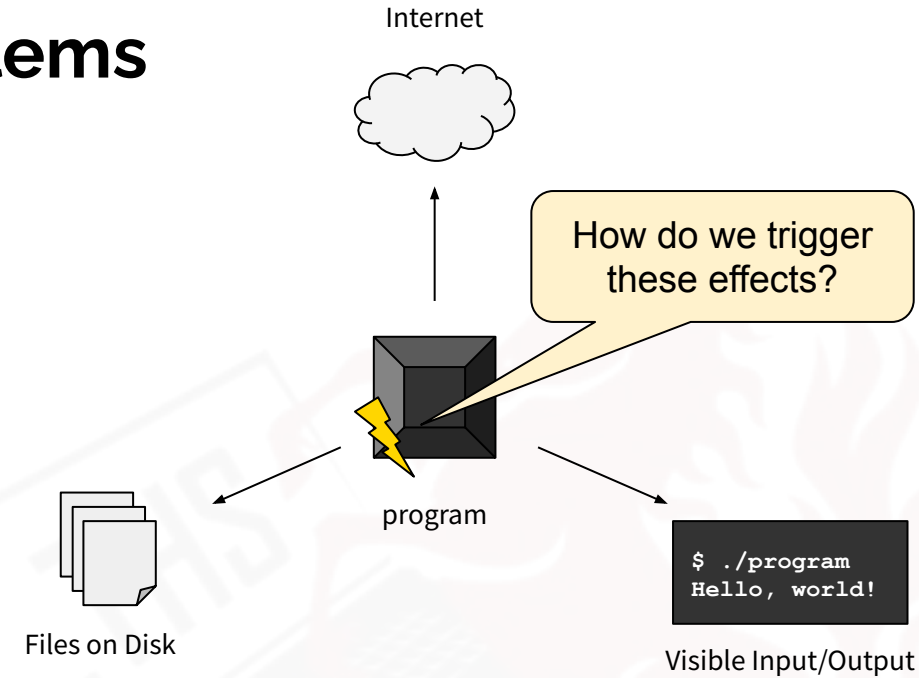


Dynamic Analysis

Two main ~~strategies~~ problems



Static Analysis



Dynamic Analysis

What is Reverse Engineering Really?

- **Pattern Matching** and **Educated Guesses.**
  - **What** do I expect to be in this program?
  - **How** could this be implemented (roughly)?
  - What kind of **evidence** would I see?
    - Used Strings
    - Required Function Calls
    - Expected Input / Output

Analyzing Malware

Challenge: Command and Control

https://ths.eemcs.utwente.nl

Key Takeaways

- It requires a lot of patience.
  - It is normal to spend a lot of time on a single program (or even function).
- It requires good time management.
  - It is usually a waste of time to try and understand everything in a program.
  - High-level constructions are more important than implementation details.
- Focus on the end-goal (This can be difficult!)

Questions?

Jerre Starink - j.a.l.starink@utwente.nl
Twente Hacking Squad: https://ths.eemcs.utwente.nl