Reverse Engineering of Malware

Matej Kašťák

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 612 66 Brno - Královo Pole
login@fit.vutbr.cz

April 10, 2021

Agenda & Goals

- Introduce reverse engineering and basic concepts
- Showcase tools used for reverse engineering
- Use the tools on a real malware sample (Demo)

What is reverse engineering?

- Analyze foreign unknown files
- Extract signatures that can be used for blocking malware
- Gain knowledge how to better protect ourselves

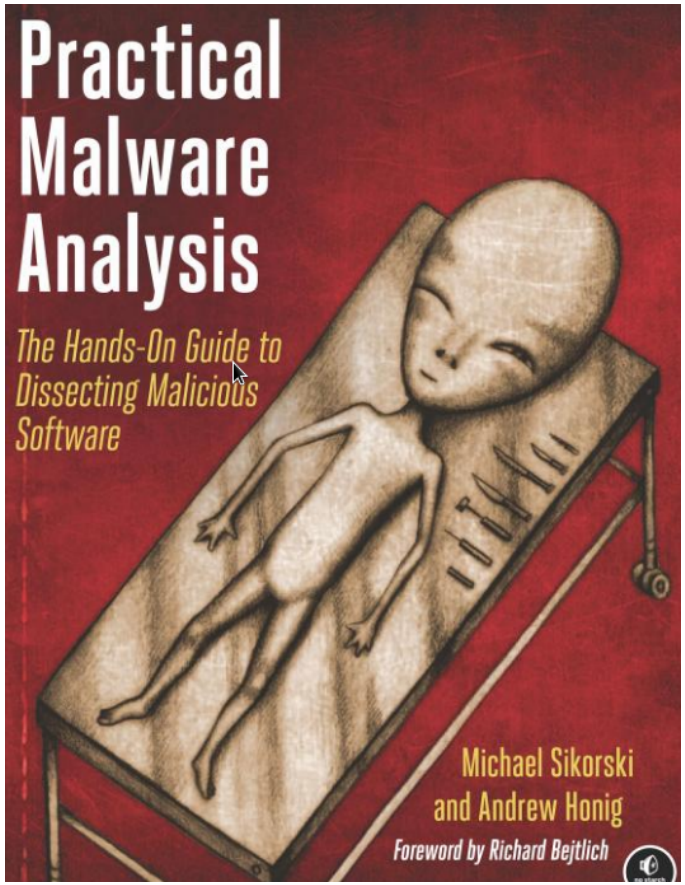

Figure: Practical Malware Analysis

Start with the right tools

Figure: Common depiction of a hacker :)

- Specialized and complex tools
- Require good low level understanding of systems
- Many of the tools are really really expensive

Static analysis

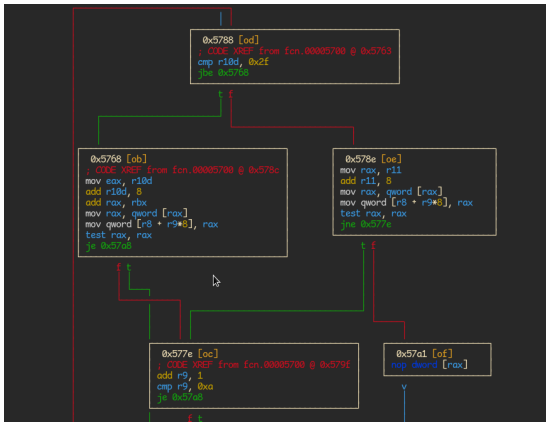

Figure: Radare2 example

- Disassemblers
  - r2/rizin cutter
  - embedded in other tools
- Decompilers
  - IDA
  - Ghidra
  - RetDec
- Misc
  - strings
  - entropy
  - file analyzers

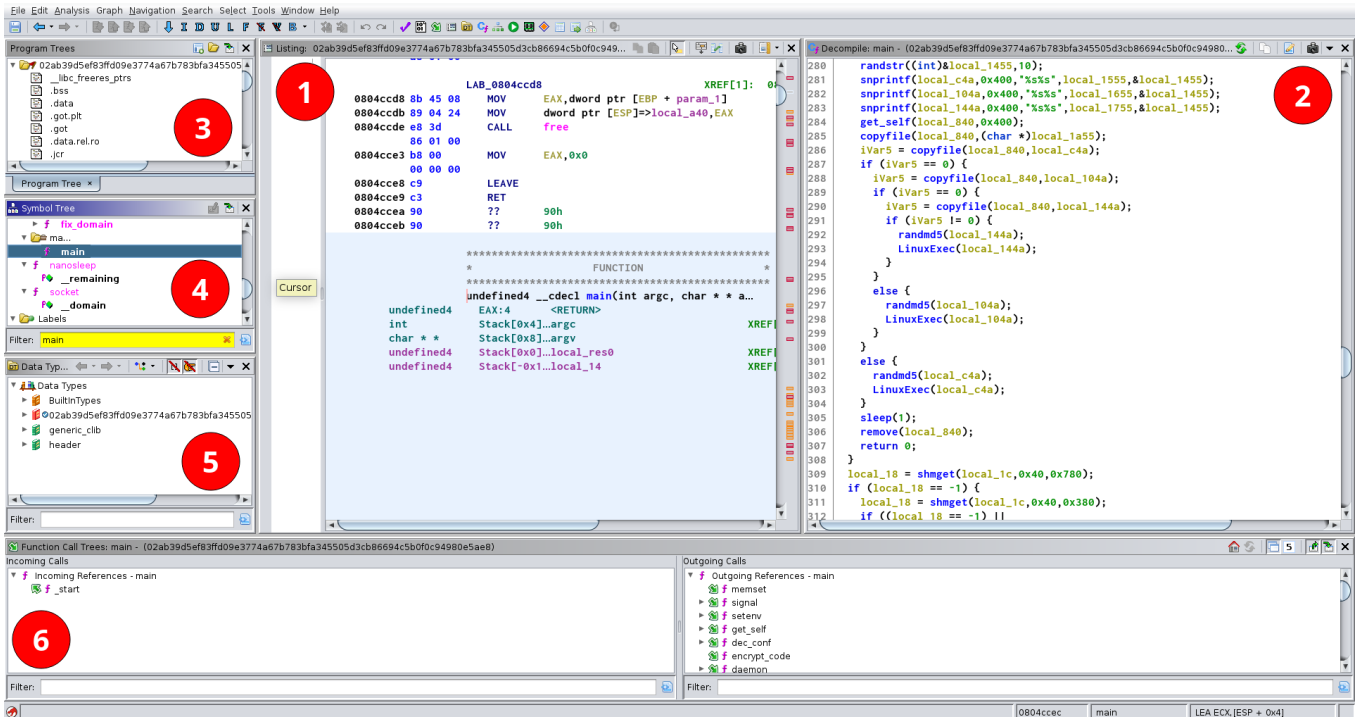

Figure: Ghidra user's interface.

- Open source competitor to IDA pro
- Sometimes lacking functionality

Dynamic analysis

- Debuggers
  - WinDbg / OlyDbg
  - GDB (stace, ltrace)
- Sandboxes
  - Cuckoo
- Symbolic execution
  - Angr

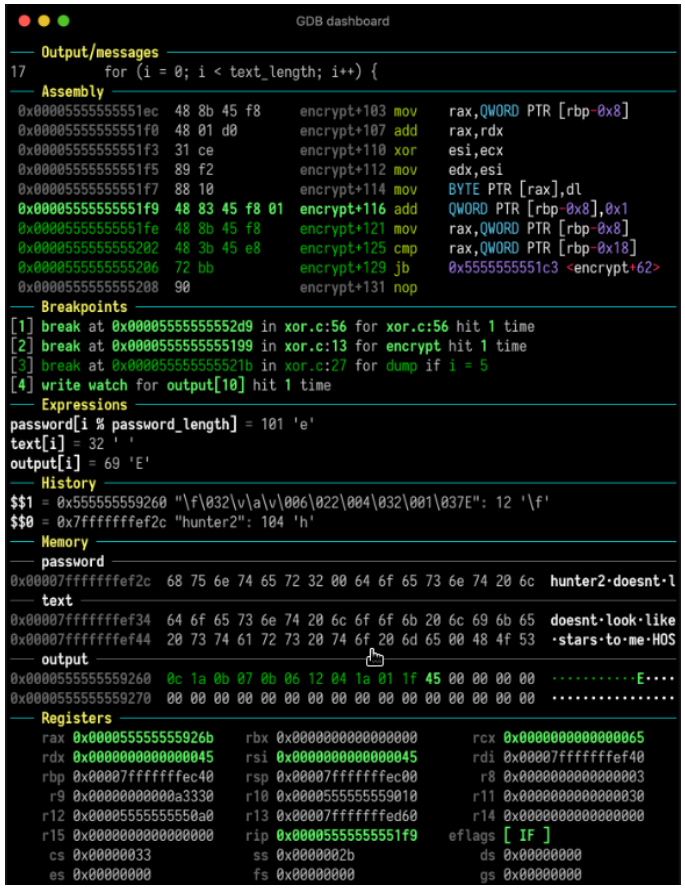

Figure: Example GDB output. Taken from `https://github.com/cyrus-and/gdb-dashboard`

Cuckoo reports

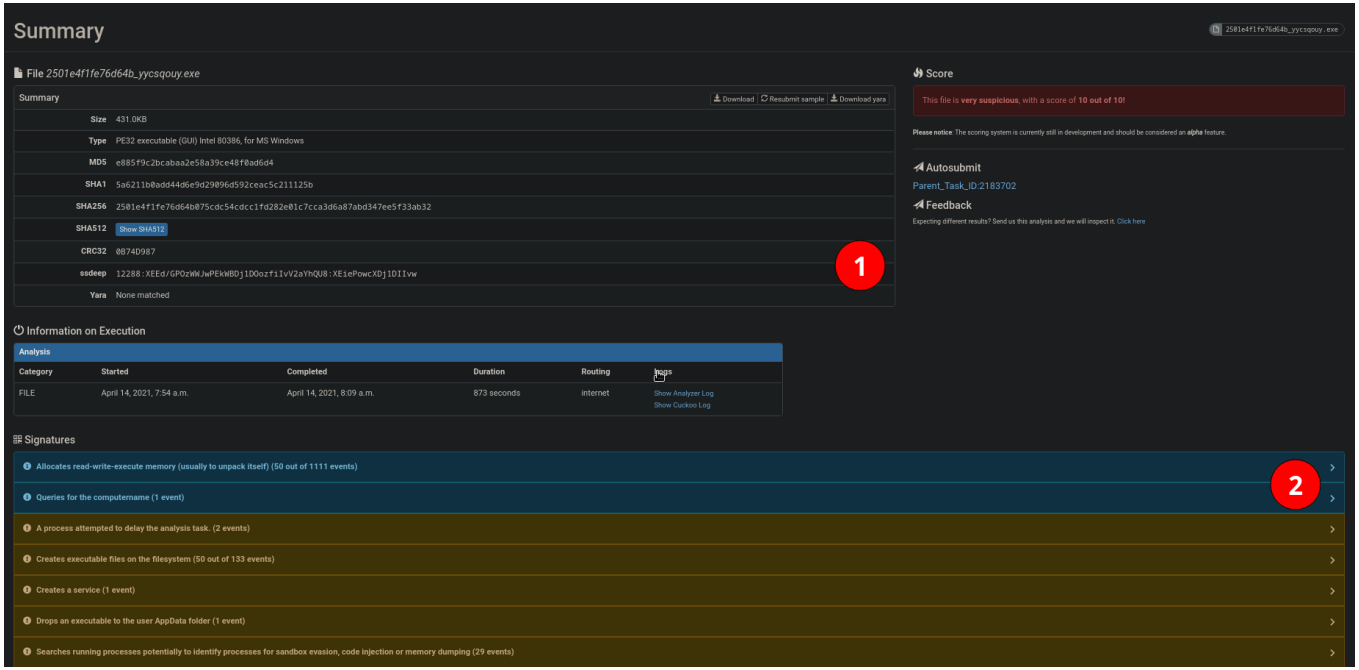

Figure: Example cuckoo analysis. Taken from `https://cuckoo.cert.ee/`

Packers

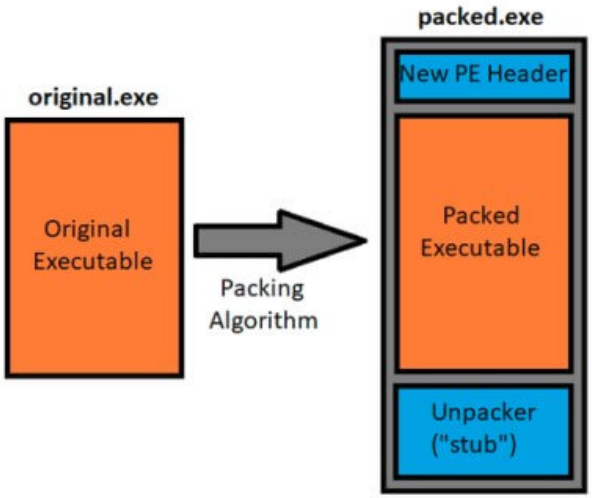

Figure: Packing diagram. Taken from https://www.arridae.com/blogs/Packed-Malware.php

- Goal is to avoid detection
- Original binary is embedded into a new one
- New binary decodes and executes the original code
- Examples
  - UPX
  - Armadillo

Demo

Summary

- Introduced basic concepts and tools used for reverse engineering
- Analyzed a malware sample using Ghidra
- Created program that is able to reverse the encryption schema