# EXERCISE #37

*SUPPLY CHAIN SECURITY REVIEW*

## Write your name and answer the following on a piece of paper

*Describe what typosquatting is in language-based package ecosystems and why it is a threat vector.*

# ADMINISTRIVIA AND ANNOUNCEMENTS

## Coding Project Clarifications

- Makefile:
- execution: analysis

python3 mycoolcode.py $1

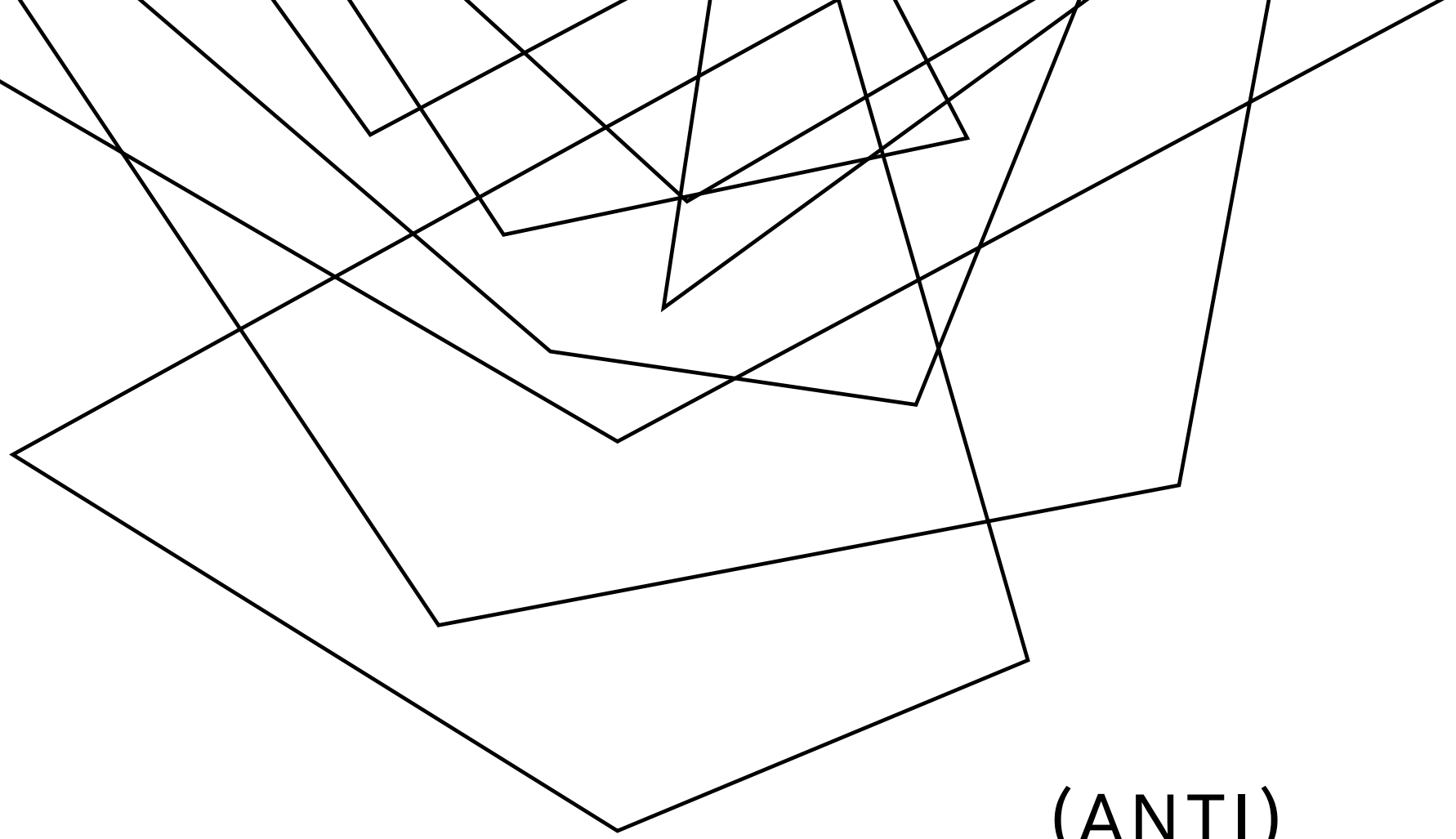
# ADMINISTRIVIA AND ANNOUNCEMENTS

This is the last lecture on new material

Monday - Review Static SSE

Wednesday - Review Dynamic SSE

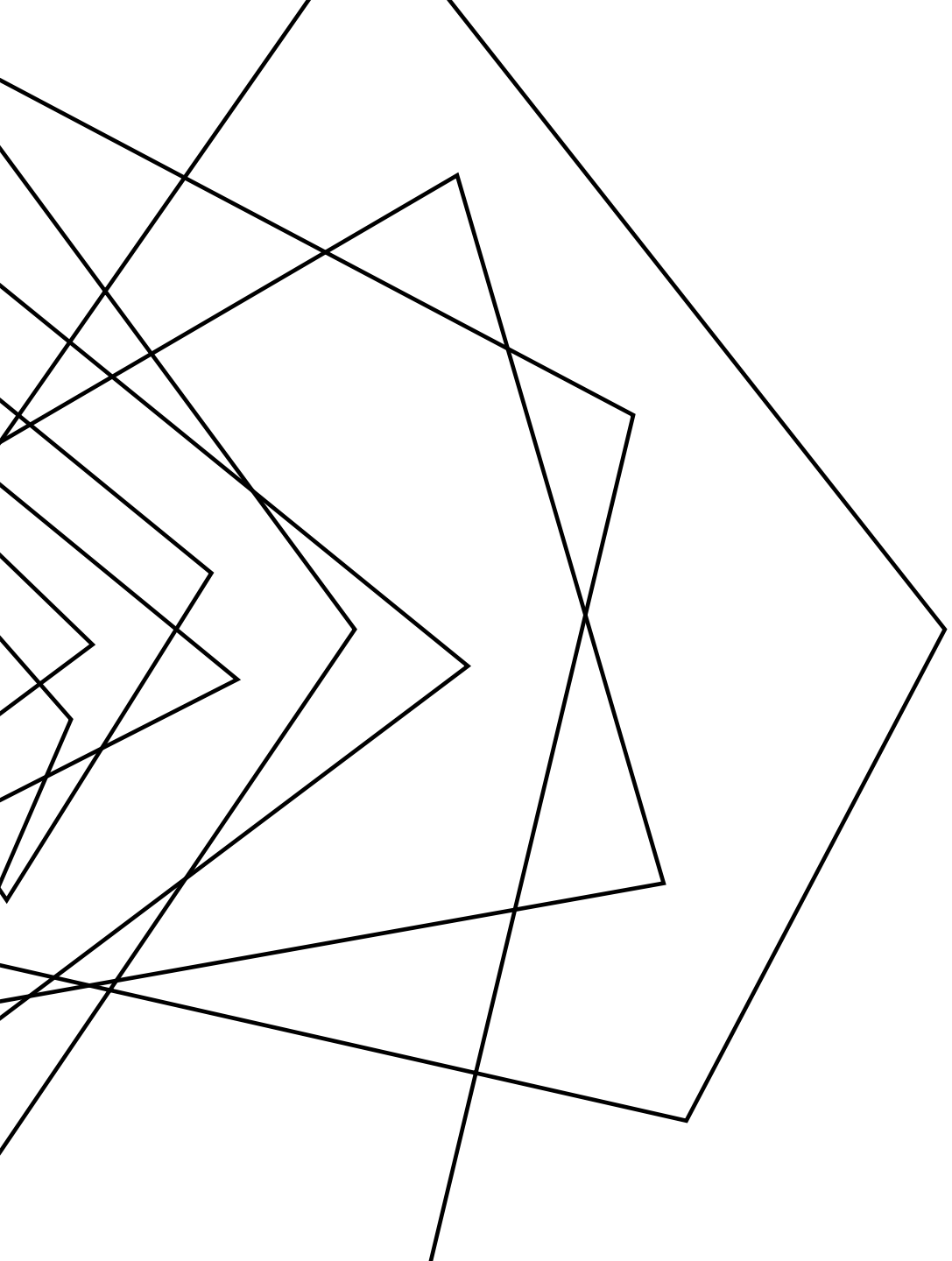
No exercises



# (ANTI)
# REVERSE ENGINEERING

EECS 677: Software Security Evaluation

Drew Davidson



# WHERE WE'RE AT

GRAB-BAG TOPICS!

# PREVIOUSLY: SUPPLY CHAIN SECURITY

## LECTURE REVIEW

### SOFTWARE SUPPLY CHAIN SECURITY

- Supply chain overview
- Threats
- Defenses

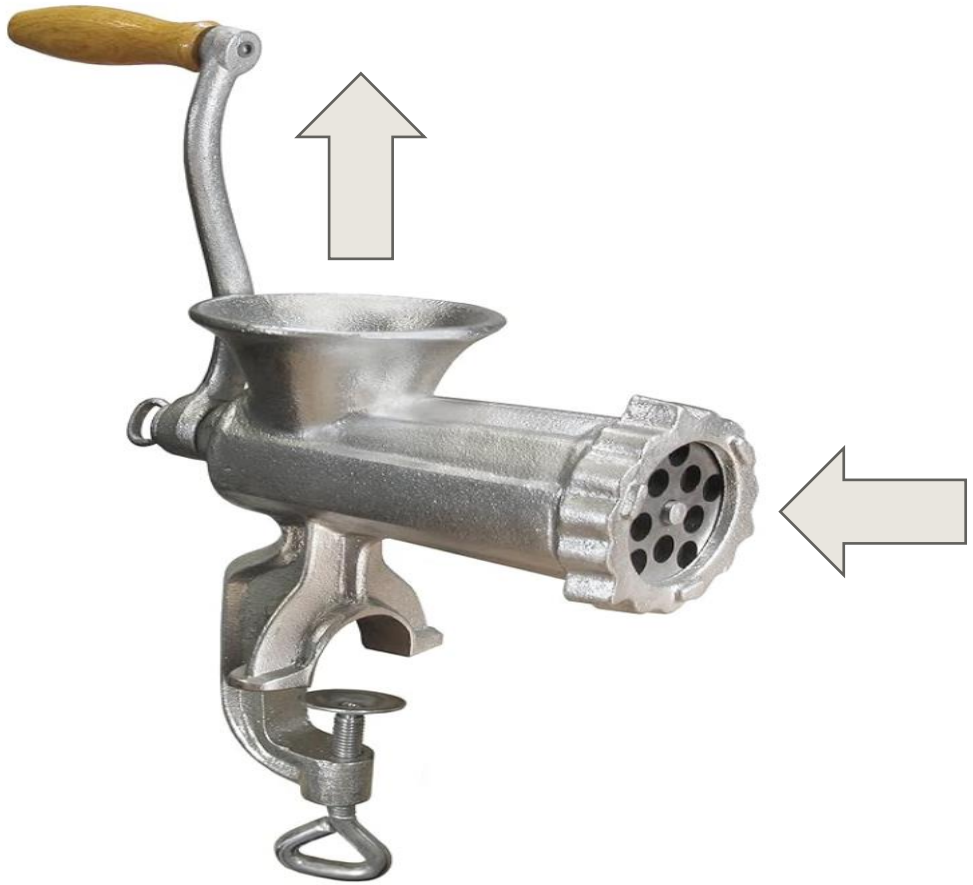
# THIS LECTURE

## REVERSE ENGINEERING

### REVERSE ENGINEERING

- Goals
- Challenges
- Tools
- ~~Evasio~~n

This wrap-up



# WHY DO WE NEED REVERSE ENGINEERING?

## OVERVIEW

### Simple Answer:

IP theft!

### Possibly-legitimate Answer

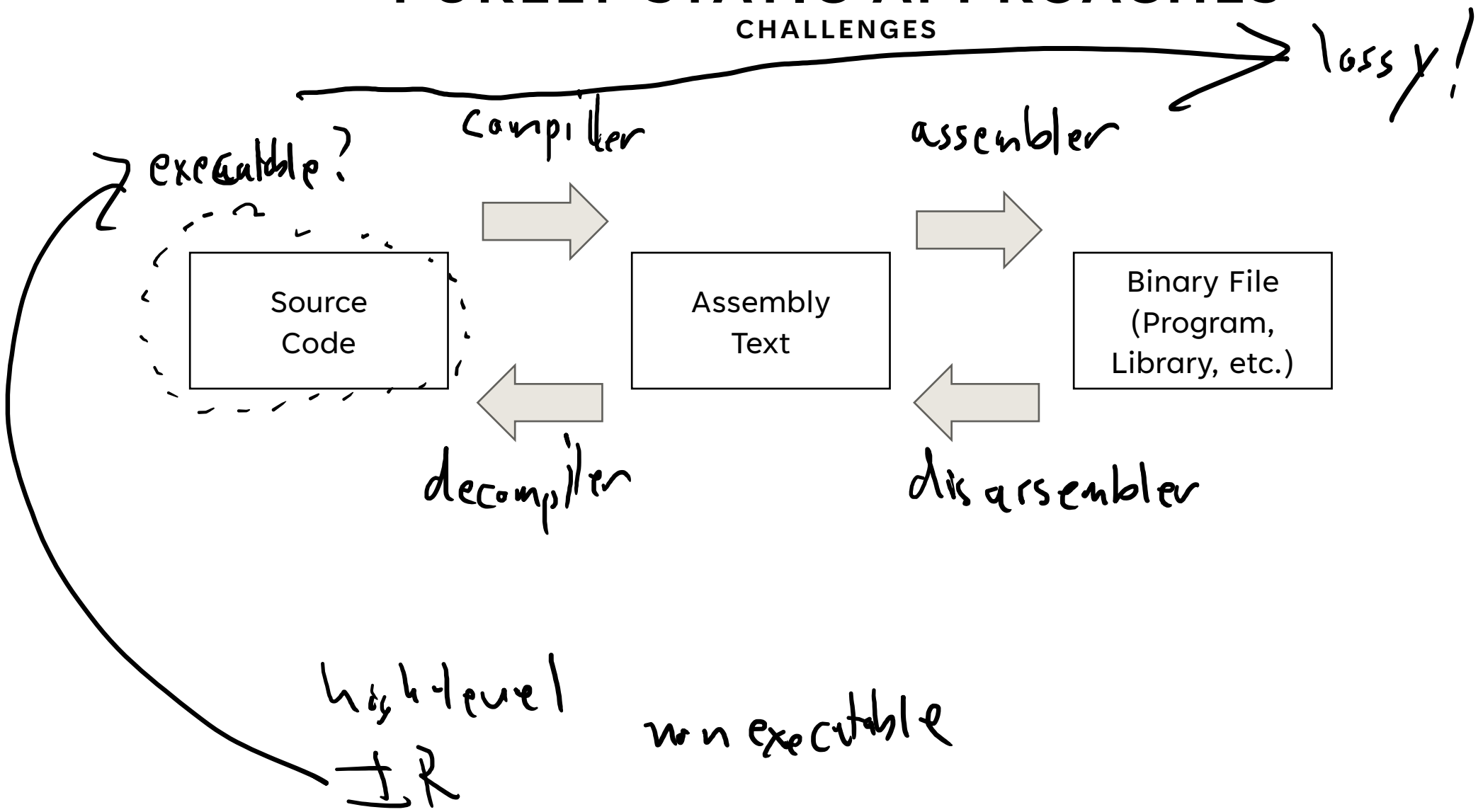
IP theft... of malware

### Answer that some people buy

Analysis of possibly-legitimate binary-only software

# PURELY STATIC APPROACHES

## CHALLENGES



# WHAT ABOUT DYNAMIC APPROACHES?

## ISSUES

Run the analysis target

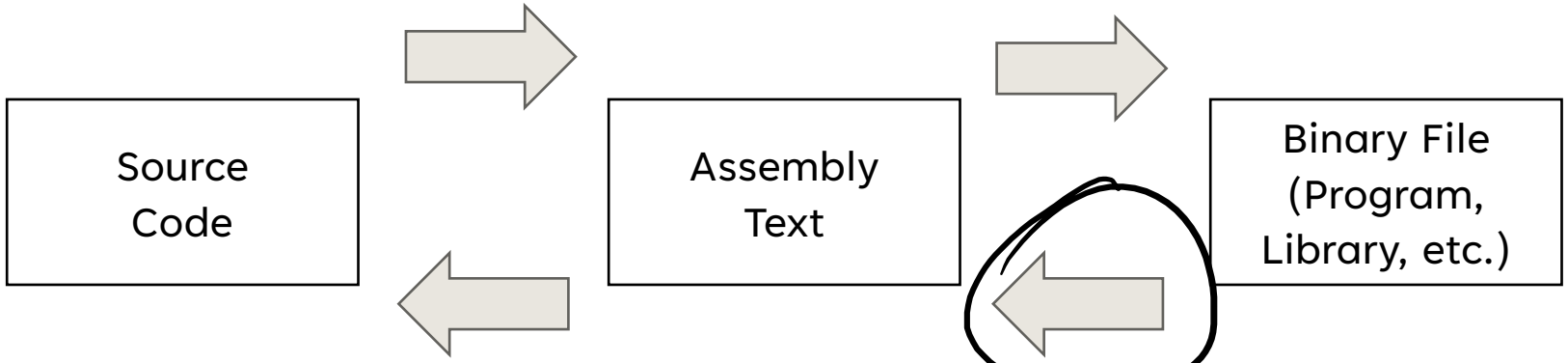
- Collect a bunch of traces
- ? Unify the traces

# CHALLENGES

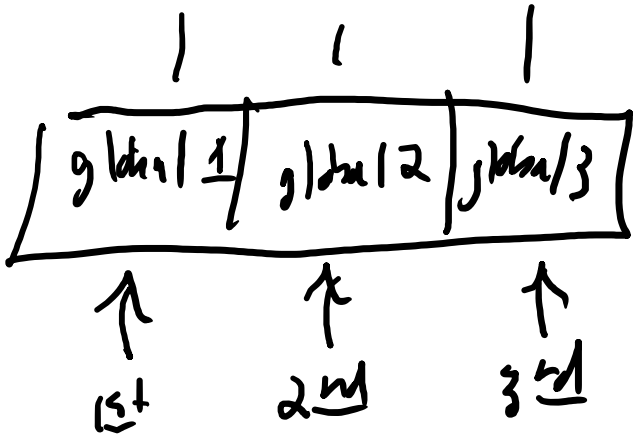
## OVERVIEW

# FOCUS ON DISASSEMBLY

## ISSUES




Why is this hard? Obfuscation!

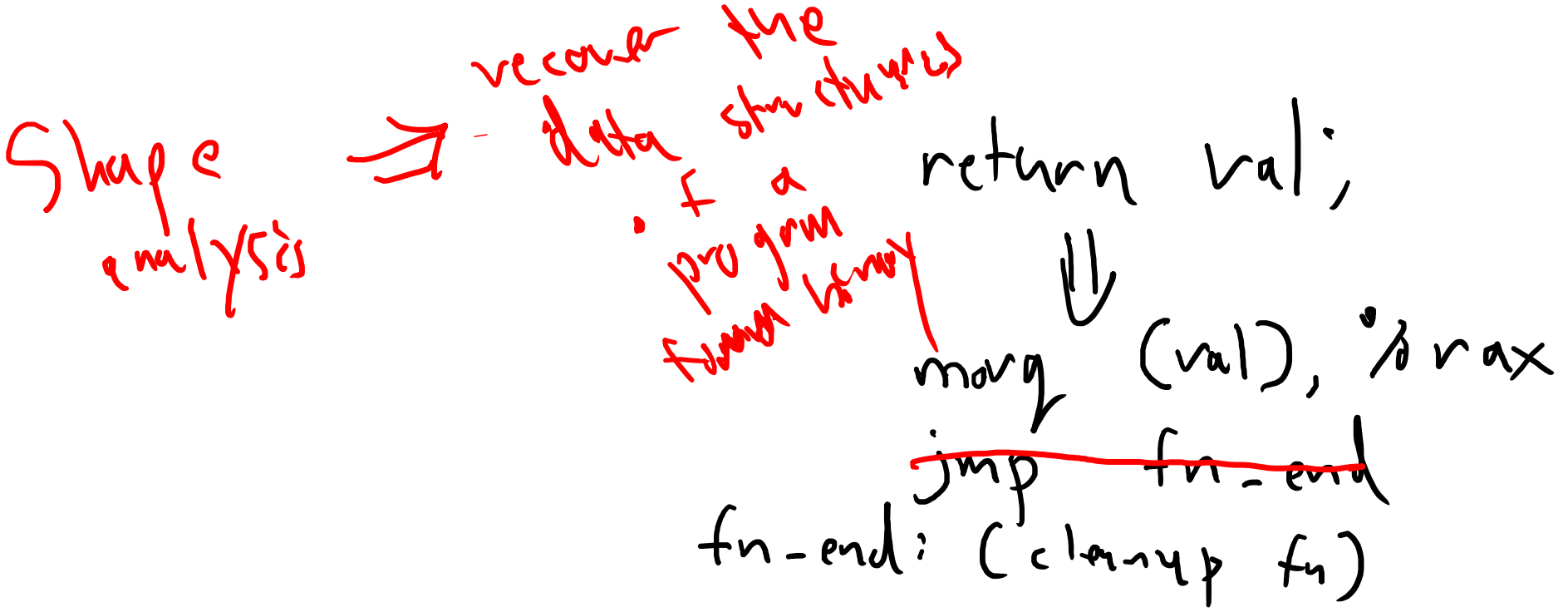


# FUNDAMENTALLY A LOSING GAME

## ISSUES

Execution needs less information than compilation, exacerbated by optimization

Implicit protocols are fine for execution, not for understanding



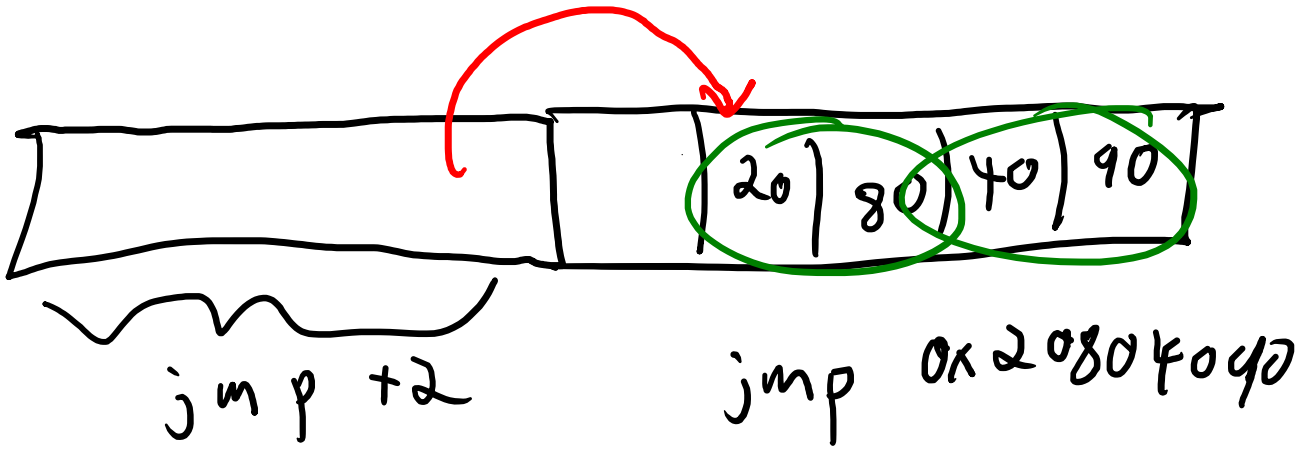
# INSTRUCTION RE-INTERPRETATION

## CHALLENGES

jmp +8

jmp +2

jmp 0x20804090



# TIME BOMBS

## CHALLENGES

```
X = getnanoseconds() & 0xFF

if (x == 12) {
    do bad
} else {
    play nice
}
```

# PACKING
## CHALLENGES

get key
decrypt all bytes starting at
0x4080

0x4080: 0x70 0x20

# TOOLS

TOOLS

## Old Answer

Ida Pro + Hex Rays

## New Answer

Ghidra

# GHIDRA

## REVERSE ENGINEERING: TOOLS

# GHIDRA: HISTORY

## REVERSE ENGINEERING: TOOLS

Internal project by the NSA since at least 2017, likely used for much longer

# GHIDRA: DEVELOPMENT

## REVERSE ENGINEERING: TOOLS

### AVAILABLE FROM THE NSA GITHUB PAGE

https://github.com/NationalSecurityAgency/ghidra/releases

C++ decompiler, frontend interface in Java+Swing

Facilities for both static reverse engineering and program exploration (i.e. debugging)

# WRAP-UP
## SOFTWARE SUPPLY CHAINS

### Reverse Engineering is Hard!

Some heuristic techniques might be ok

# THAT'S ALL FOLKS!

## SOFTWARE SUPPLY CHAINS

THIS MARKS THE END OF NEW MATERIAL IN THE CLASS

# THANKS FOR YOUR QUESTIONS!

## SOFTWARE SUPPLY CHAINS

SPECIAL THANKS TO EVERYONE THAT POSTED ON PIAZZA