Secure Software Design

Andey Robins

Spring 23 - Supplemental 3

Code Analysis

Outline

- Static Analysis
- Dyamic Analysis
- Reverse Engineering

Static Analysis

Compilers can only analyze the syntactic information of code.

Static Analyzers attempt to analyze the semantic information of code.

Static Analysis

```
[->> my-test-proj $ eslint .

/Users/prasenjitpaul/Documents/rough/my-test-proj/index.js
  1:16  error    'y' is defined but never used                no-unused-vars
  2:5   error    Expected indentation of 2 spaces but found 4 indent
  2:16  error    'z' is not defined                           no-undef
  5:11  warning  Missing semicolon                            semi

✖ 4 problems (3 errors, 1 warning)

->> my-test-proj $
```

Figure 1: Example linting with eslint

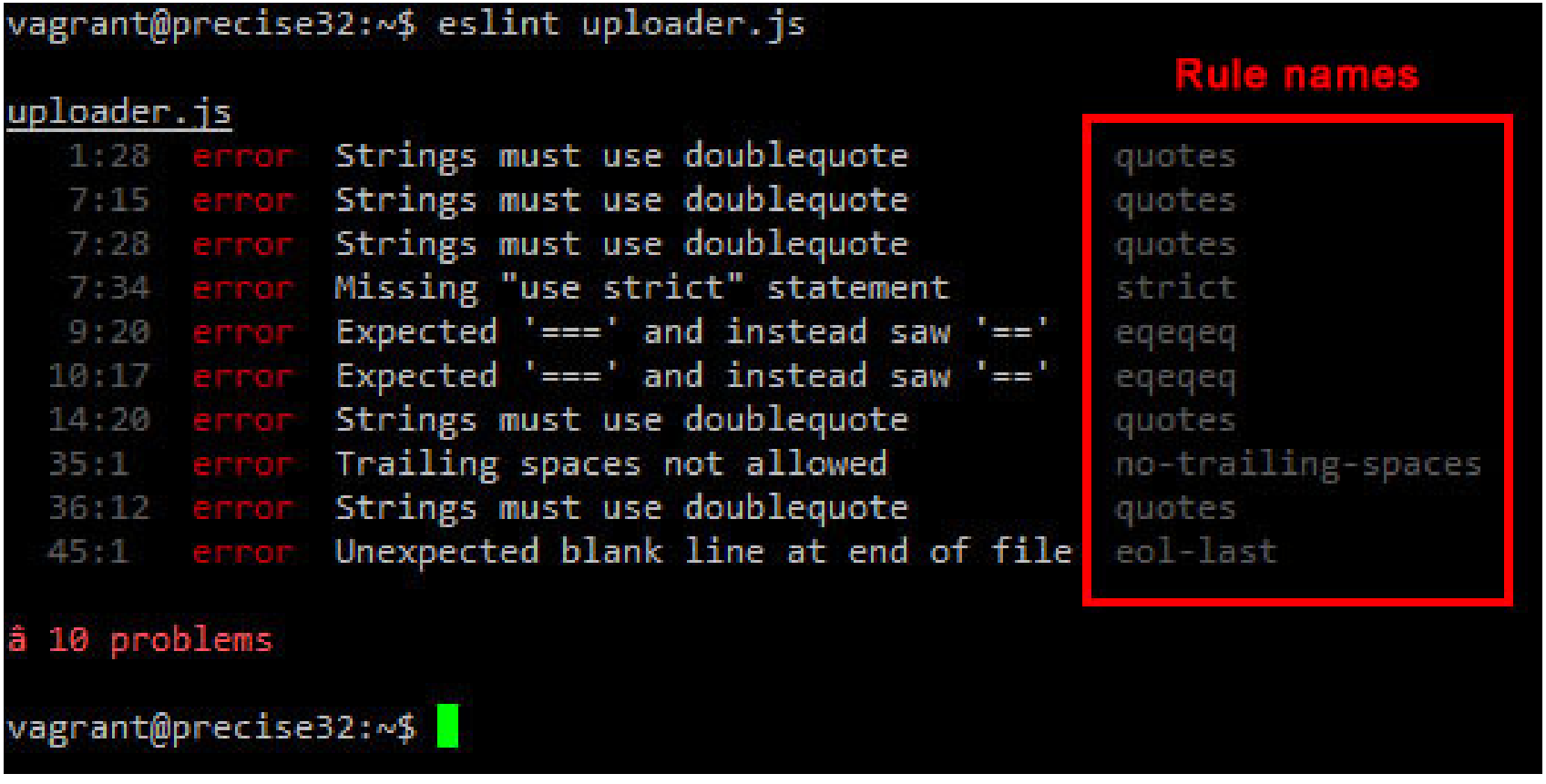

Figure 2: Further examples of linting

/usr/local/lib/python3.8/dist-packages/tensorflow
/python/keras/utils/data_utils.py,
blacklist,B310,MEDIUM,HIGH,
Audit url open for permitted schemes. Allowing
use of file:/ or custom schemes is often unexpected.,
259,8,[259],
https://bandit.readthedocs.io/en/latest/blacklists/
blacklist_calls.html#b310-urllib-urlopen

/usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/benchmarks_test_base.py,hardcoded_tmp_directory,B108,MEDIUM,MEDIUM,Probable insecure usage of temp file/directory.,29,30,[29],https://bandit.readthedocs.io/en/latest/plugins/b108_hardcoded_tmp_directory.html

Dynamic Analysis

Dynamic Analysis is all about analyzing code during execution. What systems does it interact with? What data has implications on other aspects of the system?

Taint Analysis

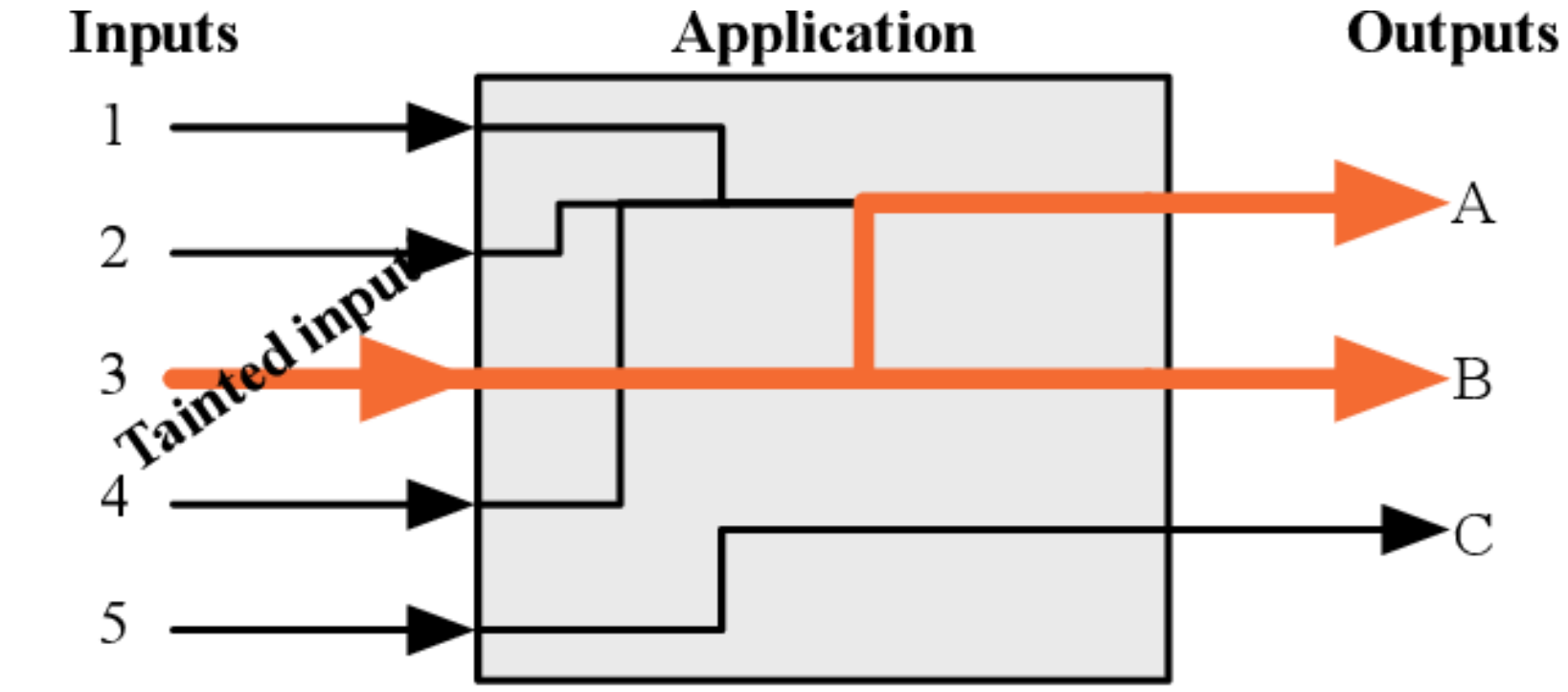

Figure 1: Example of taint tracking. Inputs enter

Figure 3: Visualization of Taint Analysis

Problems with Taint Analyis

It's soooo slow

Having to keep track of every operation and if it involves tainted data and then updating that data can effectively transform every operation into a dozen.

Taint Analysis and Vulnerability Chains

If we identify a single bug, one major potential application of taint analysis is to

Reverse Engineering

Taking binaries and converting them back to source code or some other semantic representation.

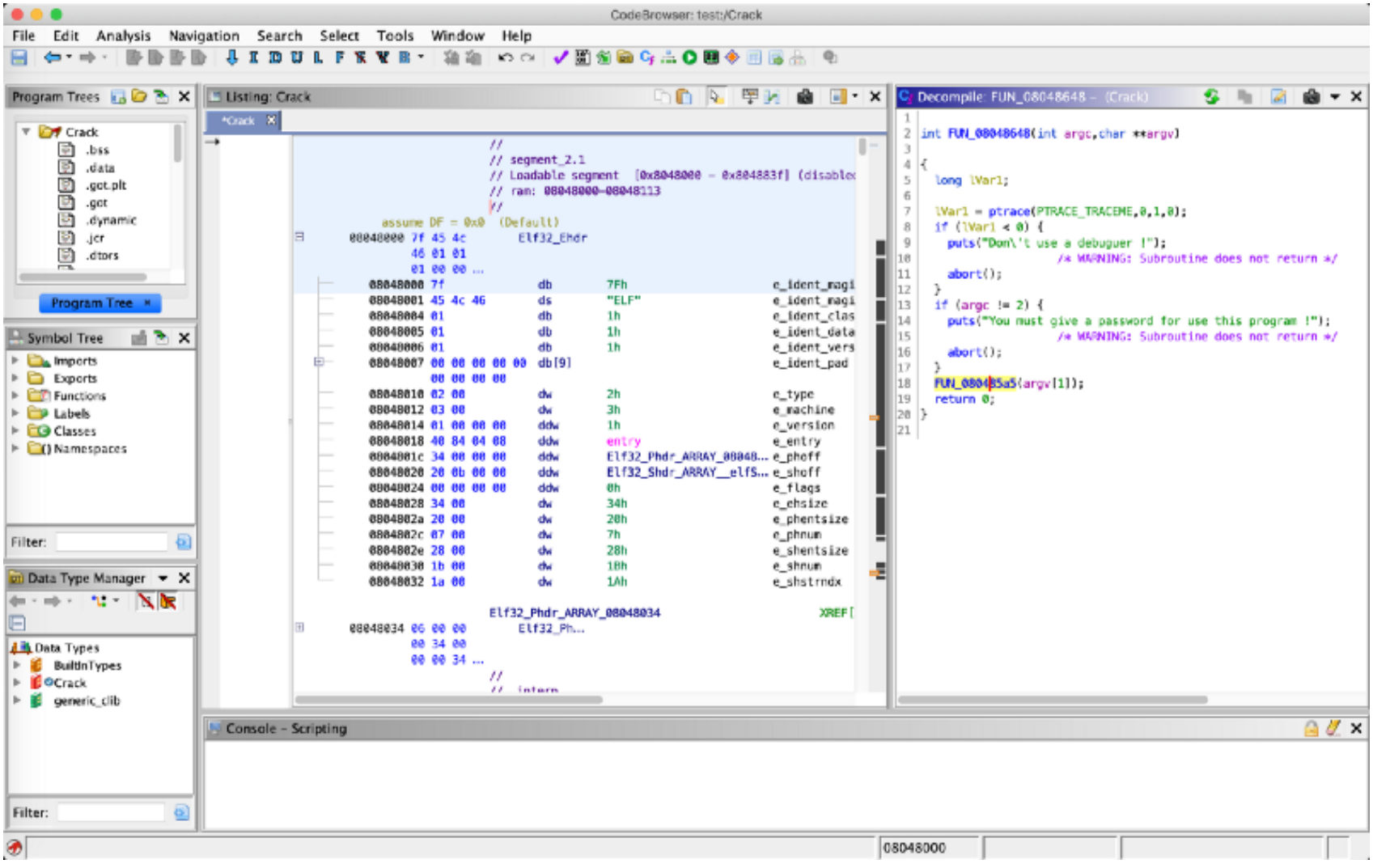

Figure 4: Ghidra, a standard reverse engineering tool

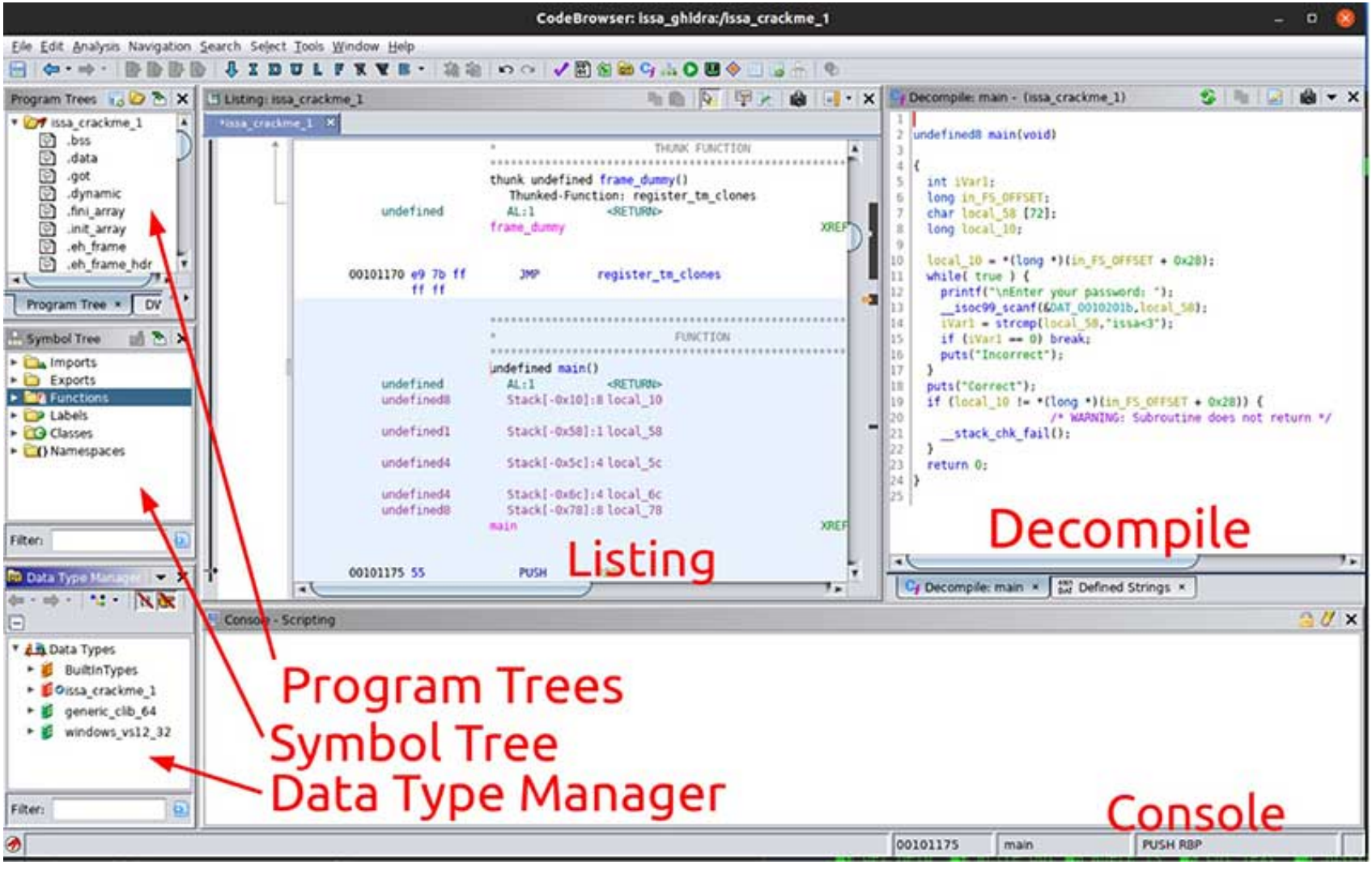

Figure 5: Parts of Ghidra

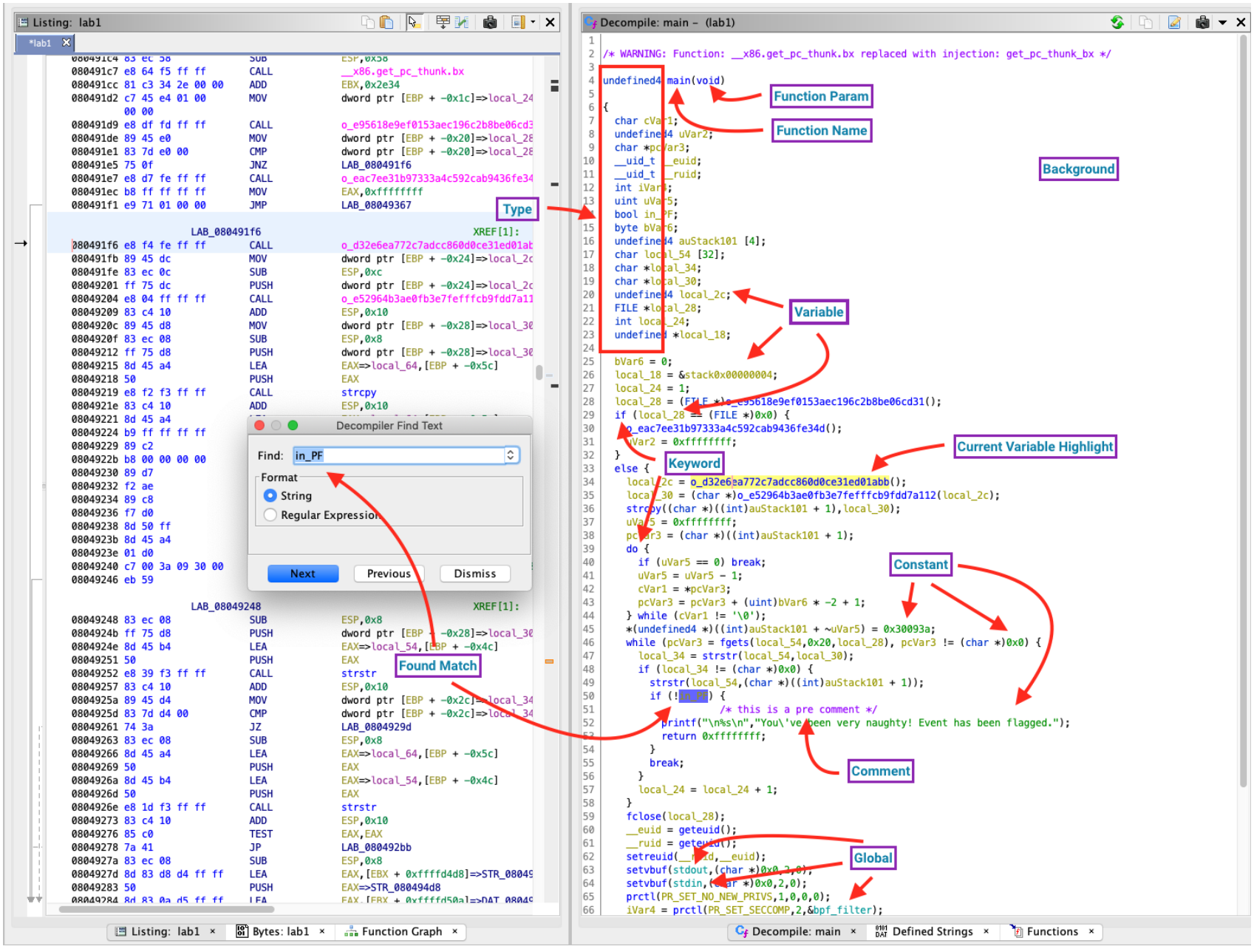

Figure 6: All the difficulties of reverse engineering

What Makes it Difficult?

1. What is data and what is code?
2. How do we find all reachable code without running it?
3. How can we run the code and know it would finish?

Halting Problem

When Do I Use Code Analysis?

1. Static Analysis -> you should start yesterday
2. Dynamic Analysis -> when trying to track down difficult bugs or when you want to assess the potential vulnerability chains
3. Reverse Engineering -> when you confirm there are compiler level problems or you don't have access to source