

# Software Forensics Overview

**Yanet Manzano**  
**Florida State University**  
**manzano@cs.fsu.edu**

manzano@cs.fsu.edu

1

## Outline

- Introduction
- Computer Programming Language
- Source Code
- Flexibility
- Software Forensics
- Software Forensics Applications
- Additional Sources of Information
- Motivation for Software Forensic
- Practice of Software Forensics
- Analyzing Executable Code
- Analyzing Source Code
- Forensic Analysis Final Step
- Review
- Conclusion
- Bibliography

manzano@cs.fsu.edu

2

## Introduction

- computer programs are written in **source code**
- **Source code**
  - produced by programmers or generated by programs
  - written in a **computer programming language**

manzano@cs.fsu.edu

3

## **Computer Programming Language**

- can be treated as a form of language from a linguistic perspective, and differ in
- **Generation:**
  - time that they were devised and reflecting their level of abstraction
- **Type:**
  - procedural, declarative, object\_oriented, and functional
- Can be examined from a forensic viewpoint

manzano@cs.fsu.edu

4

## **Source Code**

- Source code is more formal and restrictive than spoken or written languages
- However, computer programmers still have a large degree of **flexibility** when writing a program to achieve a particular function

manzano@cs.fsu.edu

5

## **Flexibility**

- Includes
  - algorithm used to solve the problem
  - code layout
    - spacing, indentation, bordering characters used to set off sections of code, etc.
  - stylistic manner in which the algorithm is implemented
    - particular choice of program statements used, variable names, etc.

manzano@cs.fsu.edu

6

## ***Flexibility***

- Other flexibilities include selecting:
  - the computer platform
  - programming language
  - compiler
  - text editor to be used
  - etc.
  
- These additional decisions may allow the programmer some further degrees of freedom, and expressiveness

manzano@cs.fsu.edu

7

## **Software Forensics**

- Features of a computer program (algorithm, layout, style, and environment) can be specific to certain programmers or types of programmer
  
- Particular combinations of features and programming idioms can make up a programmer's problem\_solving vocabulary
  
- **Therefore**, computer programs contain some degree of information that provides evidence of the author's identity and characteristics

manzano@cs.fsu.edu

8

## Software Forensic Applications

### Authorship Analysis

- **Author discrimination:**
  - task of deciding whether some pieces of code were written by a single author or by (some number of) different authors.
  - calculation of some similarity between the two or more pieces of code

manzano@cs.fsu.edu

9

## Software Forensic Applications

### Authorship Analysis

- **Author identification:**
  - determine the likelihood of a particular author having written some piece(s) of code
  - usually based on other code samples from that programmer. Example: a virus

manzano@cs.fsu.edu

10

## Software Forensic Applications

### Authorship Analysis

- **Author characterization:**
  - determining some characteristics of the programmer
  - Example: particular educational background due to the programming style and techniques used

## Software Forensic Applications

### Authorship Analysis

- **Author intent determination:**
  - determine whether code that has had an undesired effect was written with deliberate malice, or was the result of an accidental error
  - can be extended to check for negligence

## Additional Sources of Evidence

- In addition to analyzing a source code you can also analyze **object code/executable code**
- By decompiling it into source code with some information loss (optimization)
- Info obtain: compiler and/or platform used, from certain features contained in the executable that suggest the
- Info obtain varies, in general **source code** is the better source of evidence

manzano@cs.fsu.edu

13

## Motivation for Software Forensics

- **Threats:** virus, worms, Trojan horses, logic bomb, plagiarism (theft of code)

*Aggregate cost of reported computer crime and security breaches in 2002*

CRIME TYPE	LOSSES
Theft of proprietary info	over 170 million
Financial Fraud	over 115 million
Virus	over 49 million
Unauthorized insider access	over 4 million

manzano@cs.fsu.edu

14

## Practice of Software Forensics

- Psychological analysis of code can be performed
- A more scientific approach: quantitative and qualitative measurements made on computer program **source code** and **object code**
  - automatically extracted by analysis tools
  - calculated by an expert
  - using some combination of these two methods.

manzano@cs.fsu.edu

15

## Analyzing *Executable Code*

- **Useful Features**
  - Data structure and algorithm
  - Compiler and system information
  - Programming skills and system knowledge
  - Choice of system calls
  - Errors

manzano@cs.fsu.edu

16

## **Analyzing *Source Code***

### **Useful Features**

- Language
- Formatting
- Special features
  - like conditional compilation construct specially those involving initialization and declaration files
- Comment styles
- Variable names
- Spelling and grammar
- Use of language features

manzano@cs.fsu.edu

17

## **Analyzing *Source Code***

### **Useful Features**

- Scoping
  - ration of global to local identifiers)
- Execution path
  - Ex: code fully functional but never reference by any execution path)
- Bugs
- Metrics
  - software metrics: number of lines of code per function, number of blank lines

manzano@cs.fsu.edu

18

## Final Step of the Forensic Analysis

- Once these metrics have been extracted, a number of different modeling techniques, such as cluster analysis can be used to derive models
- The form of the model, the technique used, and the metrics of use all depend greatly on the purpose of the analysis and on the information available

## Review

- The **fundamental assumption** of software forensics is that programmers tend to have coding styles that are distinct, at least to some degree
- As such these styles and features are often recognizable in source code analysis
- Software Forensic Goal: analyzing computer programs authorship for legal reasons

## Conclusion

- Software Forensics can be, and has been used for a number of diverse tasks

### **More Common Applications**

- Areas of malicious code analysis
- Detection of plagiarism (code theft)

### **Less common areas**

- psychological studies of programming
- assessing source code for quality
- identifying authors of code for maintenance purposes

manzano@cs.fsu.edu

21

## Bibliography

- Andrew Gray, Philip Sallis, and Stephen MacDonell. "Software Forensics: Extending Authorship Analysis Techniques to Computer Programs." Proceedings of the 3rd Biannual Conference of the International Association of Forensic Linguists (IAFL). Durham NC, USA, 1997
- Ivan Krsul. "Authorship Analysis: Identifying The Author of a Program." CSD-TR-94-030, Department of Computer Science, Purdue University, 1994
- Eugene H. Spafford and Stephen A. Weeber. "Software Forensics: Can We Track Code to its Authors?." Purdue TR CSD-TR92010, SERC TR SERC-TR 110-P, Department of Computer Sciences, Purdue University, 1992
- Robert W. Sebesta. Programming Languages Fourth Edition. Addison-Wesley Longman Inc. 1999
- "2002 CSI/FBI Computer Crime and Security Survey." Computer Security Issues & Trends. CSI Institute

manzano@cs.fsu.edu

22