

Static Analysis, Dynamic Analysis and Symbolic Execution Techniques Applied to Security

Introduction & Case Study

Xinyi Li

May 5, 2020

Overview

Background

Static Analysis

Case 1: Detect PDF Malware

Program Analyzer

A program that takes other programs as input and decides whether or not they have a certain **property**¹.

Static Analysis

- ▶ Analysis of programs without executing
- ▶ Reason for non-trivial properties

Dynamic Analysis

- ▶ Analysis of programs by actual executing
- ▶ Common testing methods for a desire property

Symbolic Execution

- ▶ Analysis of programs by executing with **symbolic** inputs
- ▶ Determine what inputs cause each part of a program to execute

¹Anders Møller and Michael I. Schwartzbach. *Static Program Analysis*. Department of Computer Science, Aarhus University. 2018.

Program Analyzer

A program that takes other programs as input and decides whether or not they have a certain **property**¹.

Static Analysis

- ▶ Analysis of programs **without** executing
- ▶ Reason for **non-trivial** properties

Dynamic Analysis

- ▶ Analysis of programs **by actual** executing
- ▶ Common **testing** methods for a desired property

Symbolic Execution

- ▶ Analysis of programs by **executing** with **symbolic** inputs
- ▶ Determine what inputs cause each part of a program to execute

¹Anders Møller and Michael I. Schwartzbach. *Static Program Analysis*. Department of Computer Science, Aarhus University. 2018.

No Free Lunch

Limitations of Program Analysis

Program testing can be used to show the presence of bugs, but never to show their absence.²

²Edsger W. Dijkstra. “Notes on Structured Programming”. circulated privately. Apr. 1970.

³Henry Gordon Rice. “Classes of recursively enumerable sets and their decision problems”. In: *Transactions of the American Mathematical Society* 74.2 (1953), pp. 358–366.

No Free Lunch

Limitations of Program Analysis

*Program testing can be used to show the presence of bugs, but never to show their absence.*²

Rice's theorem³

All interesting questions about the behavior (*i.e. non-trivial properties*) of programs (written in Turing-complete programming languages) are **undecidable**.

²Edsger W. Dijkstra. "Notes on Structured Programming". circulated privately. Apr. 1970.

³Henry Gordon Rice. "Classes of recursively enumerable sets and their decision problems". In: *Transactions of the American Mathematical Society* 74.2 (1953), pp. 358–366.

Case 1: Malicious JavaScript PDF Extension⁴

⁴Alexander Jordan, François Gauthier, Behnaz Hassanshahi, et al. “Unacceptable Behavior: Robust PDF Malware Detection Using Abstract Interpretation”. In: *Proceedings of the 14th ACM SIGSAC Workshop on Programming Languages and Analysis for Security*. 2019, pp. 19–30.

Code

```
1 while not q.empty():
2     p = q.get()
3     p_list = os.listdir(p)
4     for i in p_list:
5         temp_p = os.path.join(p, i)
6         if os.path.isdir(temp_p):
7             q.put(temp_p)
8             continue
9         # do something
10    print(temp_p)
```