Binary Similarity Detection Using Machine Learning

ACM CCS 2018 PLAS Workshop, 2018. 10
Noam Shalev, Nimrod Partush

김영철

2018. 10. 25.

Introduction

- 오픈 소스 코드의 취약점 발견 확률이 증가.
- 새로운 취약점이 발견될 때마다 관련된 여러 가지 버전의 바이너리에도 영향을 끼침.
- compiler + optimization level + architecture에 바이너리 유사도 캐치
  - SMT solver는 정확하지만 느림.
  - 빠른 방법은 낮은 정확도를 보임.

- proc2vec method를 이용하여 빠르고 정확한 방법 제시

Related Work

- syntactic techniques : cross-compiler에 관한 탐지 불가.
- dynamic analysis : computationally heavy techniques.

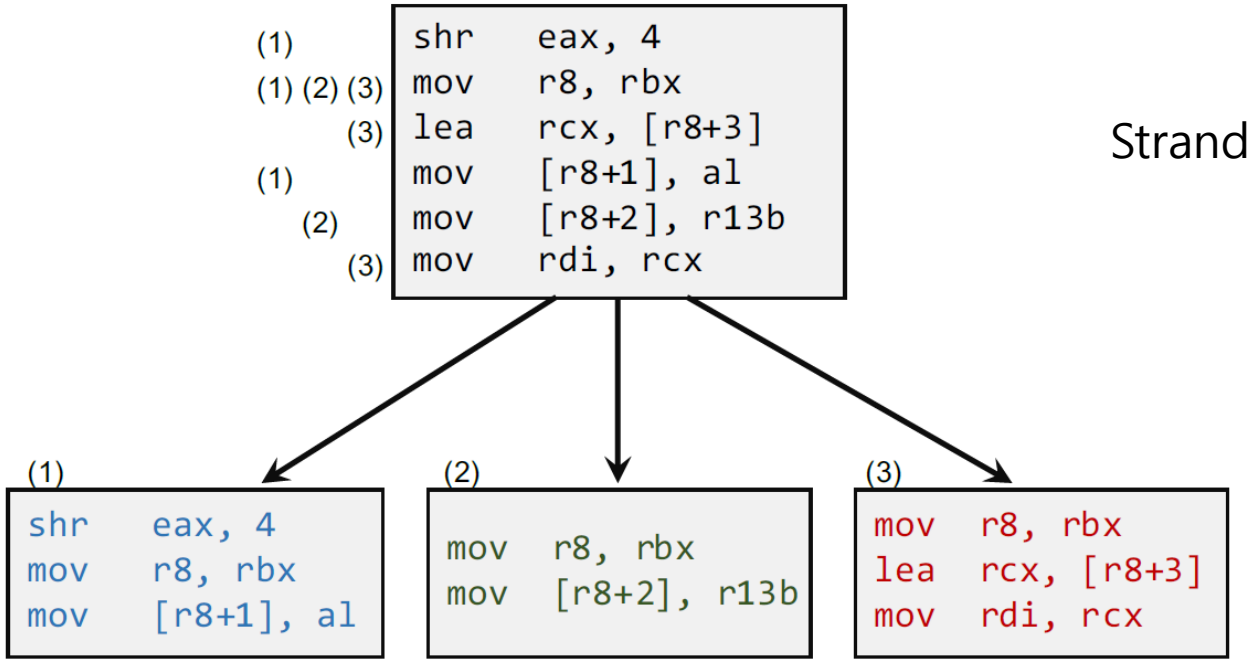
- David et al. [4] : 코드 조각을 strand로 쪼개어 유사도 측정, SMT solver를 이용. 느림.

- 여기서는 strand에 기반한 접근법을 Neural Network에 적용하여 빠르고 정확도 높은 방법을 연구. (Zeek tool)

[4] Yaniv David et al. "Statistical Similarity of Binaries.", PLDI 2016

Zeek

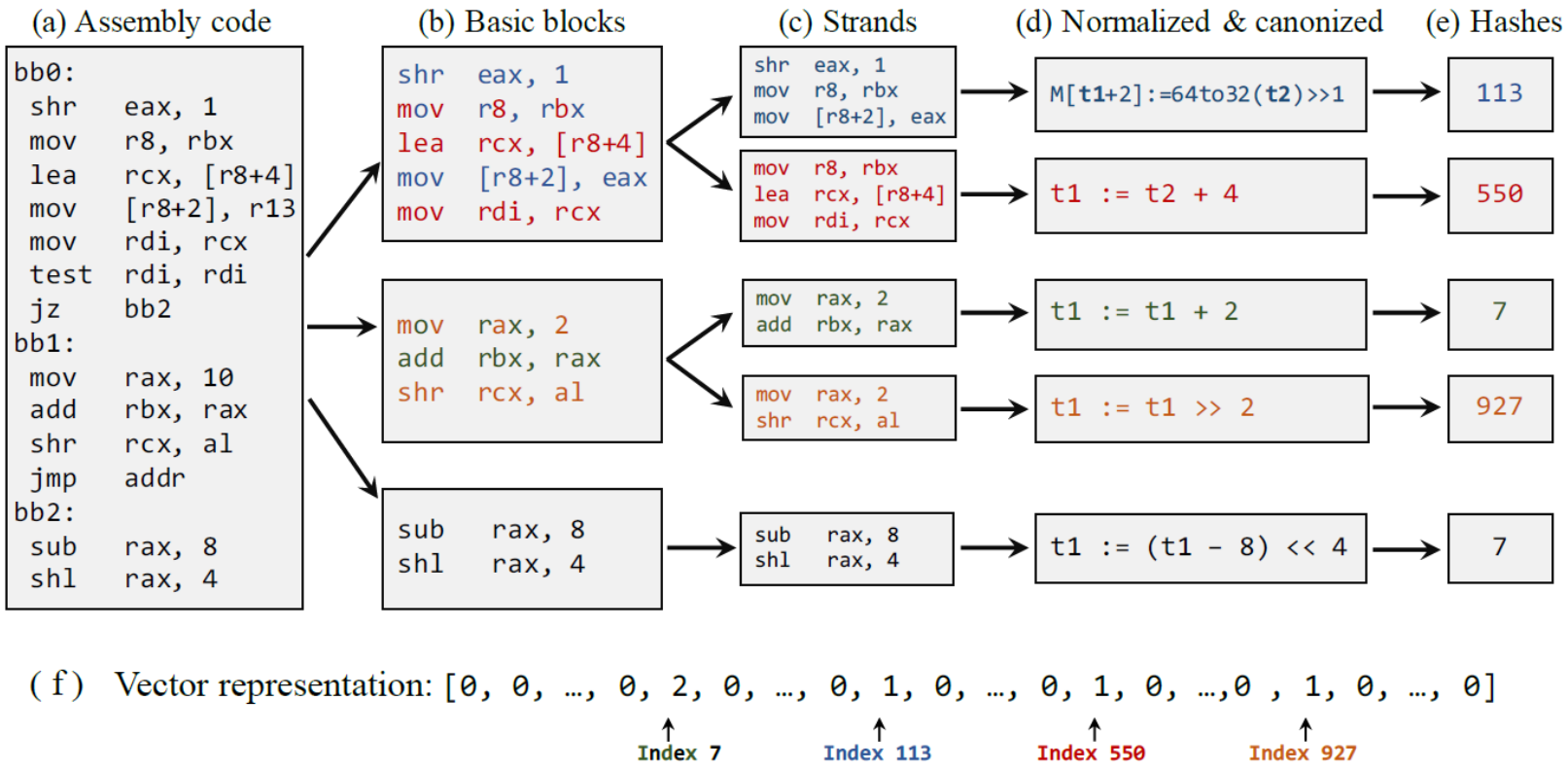
- Strands as Features



Strand : 특정 value를 컴퓨팅하는 instruction의 집합.
연속적으로 나타낼 필요 없음.

Zeek

- proc2vec (5steps)



Zeek

- proc2vec

  step1. procedure를 basic block으로 쪼갬.

  step2. basic block을 strand로 쪼갬. (VEX IR 이용)

  step3. 각 strand와 의미적으로 동일한 textual representation을 생성.

  step4. textual representation에 n-bit MD5 hash를 적용. 범위(0 ~ 2^n)

  step5. hash 값을 index로 이용하여 vector 생성. (각 요소의 합은 해당 프로시저 내의 strand의 수)

Zeek

- Data Generation