LCS와 n-gram 유사도를 이용한 패커 식별 방법

김영철

2017. 5. 17.

수행 목적

- 많은 프로그램들이 다양한 packer에 의해 packing 된 바이너리로 만들어지고 있다.
  ➔ 프로그램 분석이 어려워짐

- 프로그램에 사용된 packer를 식별하고 수동으로 unpacking 하는 과정이 필요하다.
  ➔ 특정 packer에 대해 unpacking 해주는 도구가 존재
  ➔ signature를 이용한 packer 식별 도구와 ML을 이용한 패커 식별 도구가 다수 존재
  ➔ IEICE 2014에서 발표된 ISAWA*의 방법을 응용하여 패커 식별 수행

*데이터 추출을 위해 참고한 논문

Packer

- Compressor
  - 프로그램의 크기를 줄이는 목적으로 사용된다.
  - Unpacking이 쉽다.

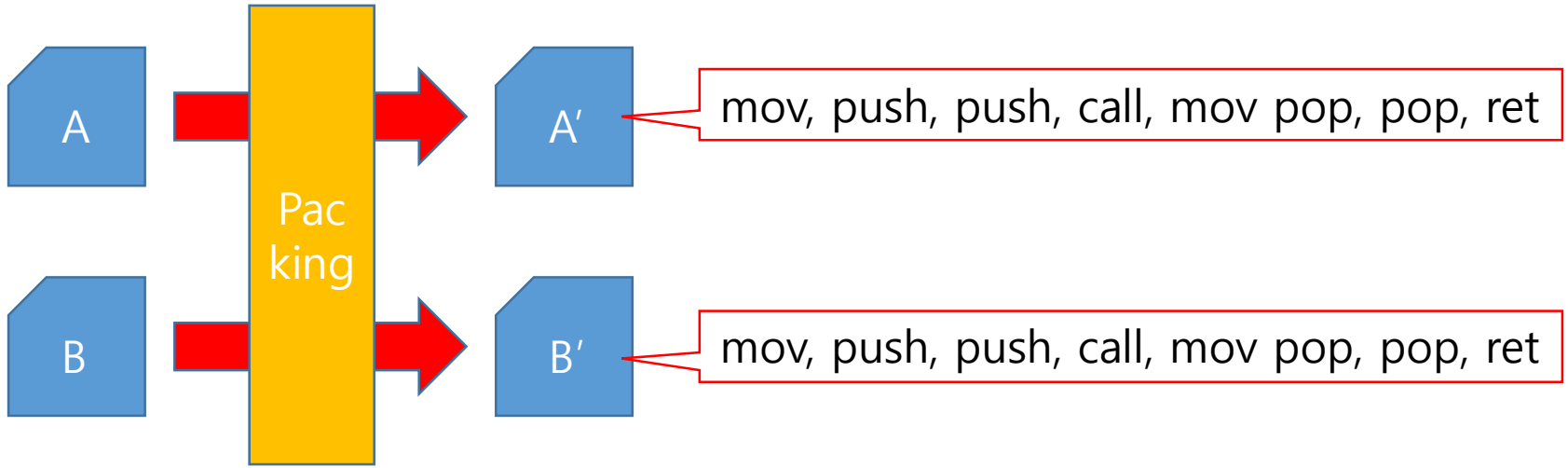
  ex) UPX, MEW 등

- Protector
  - 프로그램을 보호하는 목적으로 사용된다.
  - Anti-Reversing 기법이 적용되어 분석이 어렵다.

  ex) Yoda’s protect, ASProtect, Themida, Upack 등

ISAWA의 packer 식별 방법

- 같은 packer에 의해서 packing 된 두 바이너리가 있을 때, Entry Point 부터 N-byte 코드 간의 edit distance는 작다
- SVM을 이용하여



LCS와 n-gram 유사도를 이용한 packer 식별 방법

- slide 4에서 데이터를 수집하는 방법과 같이 바이너리의 Entry Point부터 N개의 mnemonic을 추출
- 두 N-mnemonics 간의 LCS 값과 n-gram 유사도 값을 계산
- 같은 packer로 생성된 바이너리 간의 LCS 값과 n-gram 유사도는 높을 것

실험

- packing을 위해 사용된 packer : UPX, PESpin, ~~MEW~~, ~~Petite~~, ~~Yoda's Protect~~
- packing 된 바이너리 : calc, cmd, control, msprint, notepad
- IDA Pro를 통해 EP를 찾고 mnemonic 추출
- mnemonic 수는 10~50까지 5단위로 설정
- 추출한 mnemonic의 수가 같은 데이터 간의 LCS와 n-gram 유사도 값을 계산

실험 결과

같은 packer로 packing 되었어도 LCS와 n-gram 유사도 값이 낮은 경우가 발생

여러가지 실험을 더 시도해야 할 것으로 보임