# REIL: A platform-independent intermediate representation of disassembled code for static code analysis

CanSecWest 2009
Thomas Dullien, Sebastian Porst

김영철

2016. 2. 15.

# Review

- REIL 소개
  ➔ 다양한 플랫폼에 작동 가능한 중간언어
  ➔ 17개의 instruction

- 다섯 그룹으로 분류되는 instruction set
  ➔ the arithmetic instructions
  ➔ the bitwise instructions
  ➔ the conditional instructions
  ➔ the data transfer instructions
  ➔ other instructions

# The REIL architecture

- 스택이 없는 레지스터 기반의 구조
  ➔ $t_0$, $t_1$, ... $t_{<number>}$
  ➔ 사용 가능한 레지스터 수 무제한
  ➔ 레지스터의 크기 = 사용될 위치의 operand 크기

# The REIL architecture

- virtual REIL machine
  - ➔ flat memory model
  - ➔ memory segment, memory selector 없음
  - ➔ REIL memory는 무한함
  - ➔ memory segment 시뮬레이션 가능
  - ➔ 엔디언을 다루는 메커니즘 X
    - ➔ 메모리 접근 REIL instruction이 생성될 때 처리
      - ➔ 런타임에 엔디언이 바뀌는 경우 문제 발생
        - ➔ 보안 측면에 거의 관련이 없음

# Translating native code to REIL

- REIL translator
  - ➔ native assembly 코드 조각을 번역하는 것을 반복
  - ➔ 다음에 올 instruction을 예상하지 않음
  - ➔ 이전 instruction으로 부터 생성된 정보 요구 X
  - ➔ 하나의 native instruction을 여러 REIL instruction으로 맵핑
    - ➔ native assembly와 REIL instruction의 관련성을 파괴
      - ➔ REIL instruction들을 8bit left

# Limitations of REIL

- 모든 프로세서에 대해 translator가 있는 것은 아님
  ➔ 32-bit x64, 32-bit PowerPC, 32-bit ARM 존재
  ➔ 보안 문제와 관련이 많은 프로세서

- FPU instructions, privileged instruction 번역 불가
  ➔ 보안 문제와 밀접한 관련이 없음

# Limitations of REIL

- 플랫폼 독립적인 측면으로 보았을 때, exception을 다루지 않음.
  ➔ dividing by zero exceptions … 무시

- self-modifying code 처리 불가
  ➔ 초기 번역 후에 REIL code 고정

# The future of REIL

- 더 많은 REIL translator 구축

- REIL 코드의 질 향상
  ➔ zero size operand의 출현
  ➔ bit 단위의 operand를 사용하여 더 좋은 성능을 낼 수 있음

# The future of REIL

- REIL translation의 정확성 향상
  ➔ extend, reduce instruction
  extend – 값을 유지하면서 size를 늘림
  reduce – 값을 유지하면서 size를 줄임
  ➔ reduce의 경우 값의 불변을 보장 못함

- operand type 종류 늘림
  ➔ register index 등
  ➔ register bank에 index로 addressing되는 경우 번역 불가

# Related work

- 보안적 측면을 위한 중간언어
  ➔ Mihai Chiriac의 중간언어
    ➔ instruction은 한번에 하나의 효과만 있음
    ➔ 무한한 virtual register를 가질 수 있음

  ➔ ERESI project의 ELIR
    ➔ platform-independent

# Related work

- 보안적 측면을 위한 중간언어
  ➔ IDA Pro & Hex-Rays의 IR
    ➔ single-responsibility rule 파괴
    ➔ 정수 문자와 포인터 구별

  ➔ CodeSurfer/X86
    ➔ REIL과 얼마나 유사한지 알 수 없음

  ➔ AbsInt의 CRL2
    ➔ 제어 흐름 분석에 초점을 맞춤
    ➔ annotation이 많아 복잡함

# Conclusions

- 정적 분석에 사용 가능한 REIL 완성
- BinNavi에 탑재
- x86, PowerPC, ARM에 대한 플랫폼 제약 없이 작동