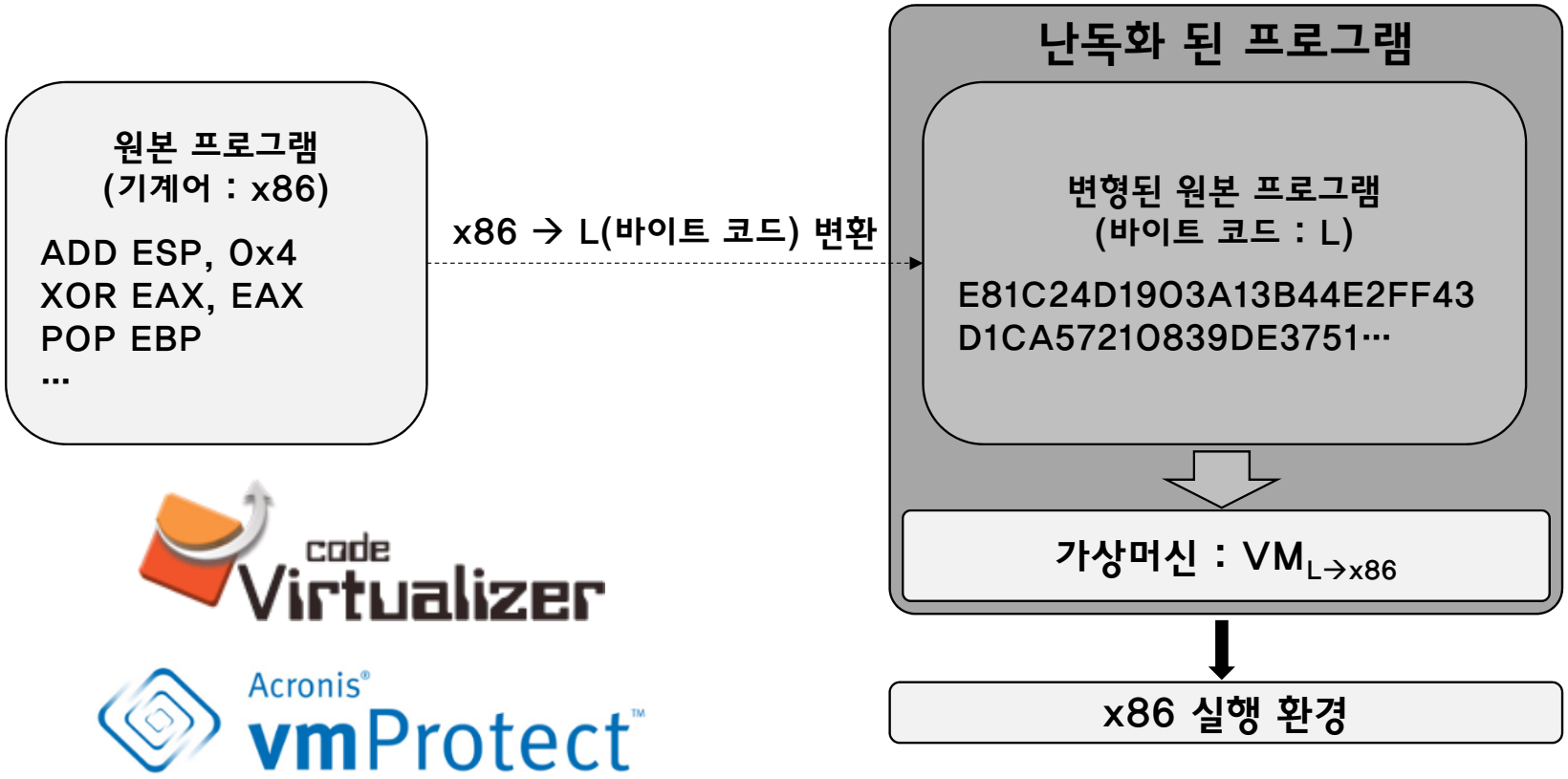
가상머신 기반 난독화 바이너리의 동적 분석

PLRG KAIST 이성호 류석영

가상머신 기반 난독화?


원본 프로그램
(기계어 : x86)
ADD ESP, 0x4
XOR EAX, EAX
POP EBP
…
x86 → L(바이트 코드) 변환
난독화 된 프로그램
변형된 원본 프로그램
(바이트 코드 : L)
E81C24D1903A13B44E2FF43
D1CA57210839DE3751…
가상머신 : VM L→x86
x86 실행 환경
code Virtualizer
Acronis
vmProtect

왜 하나요?

- 악성코드가 가상머신 기반 난독화로 보호되는 경우
  - 시그니쳐 기반 보안 프로그램으로 검출 X
  - 악성코드 검출을 위한 정적, 동적 분석이 가상머신에 의해 방해
  - ➔ 검출해 내기가 어려움

분석의 어려움?

- 바이트 코드의 의미는 난독화 시기 마다 다름
  - 가상 머신의 코드 또한 난독화 시기 마다 다름
  - 기 분석정보의 활용 어려움
- 보여지는 것은 오로지 가상머신
  - 바이트 코드: 데이터
  - 가상머신: 프로그램

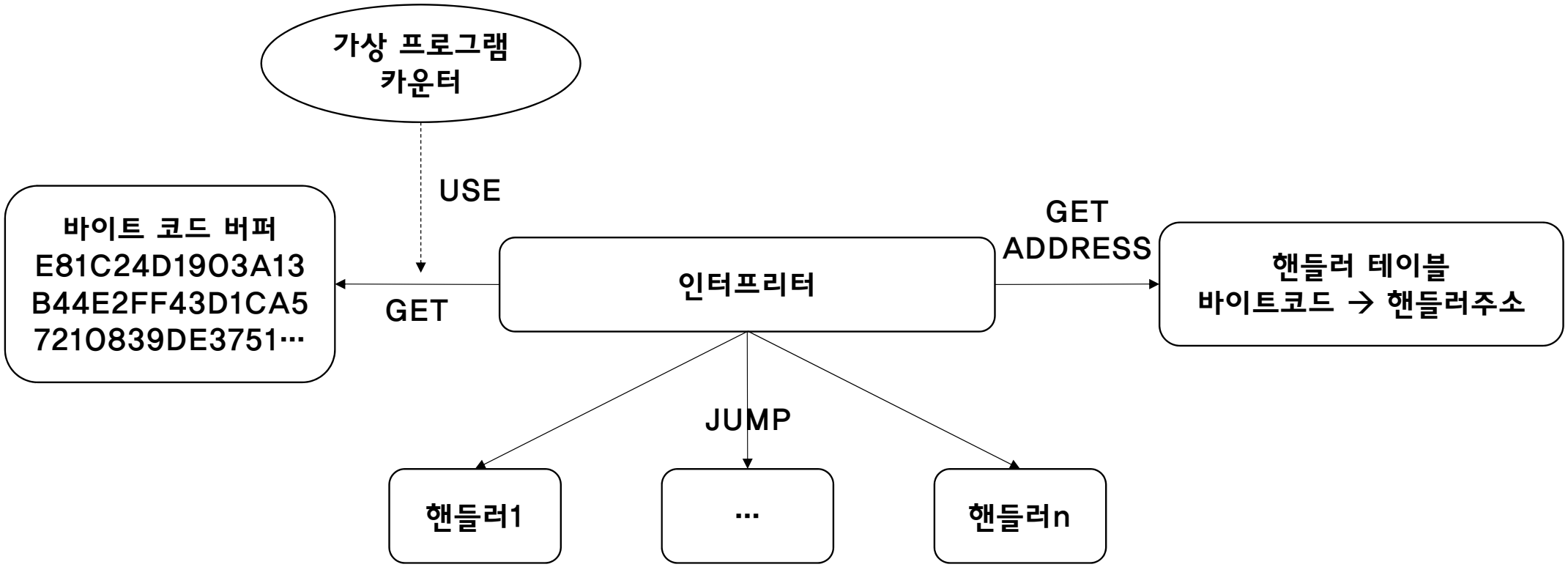
분석의 어려움?

- 정적 분석
  - 원본 프로그램은 임의의 바이트 코드로 변형되어 데이터 형태로 존재
  - 가상머신이 프로그램으로써 바이트 코드의 의미를 수행 ➔ 가상머신만 보임
- 동적 분석
  - 가상머신 자체와 바이트 코드 의미가 번갈아 수행
  - 바이트 코드 프로그램 실행만을 추출해 내기 어려움

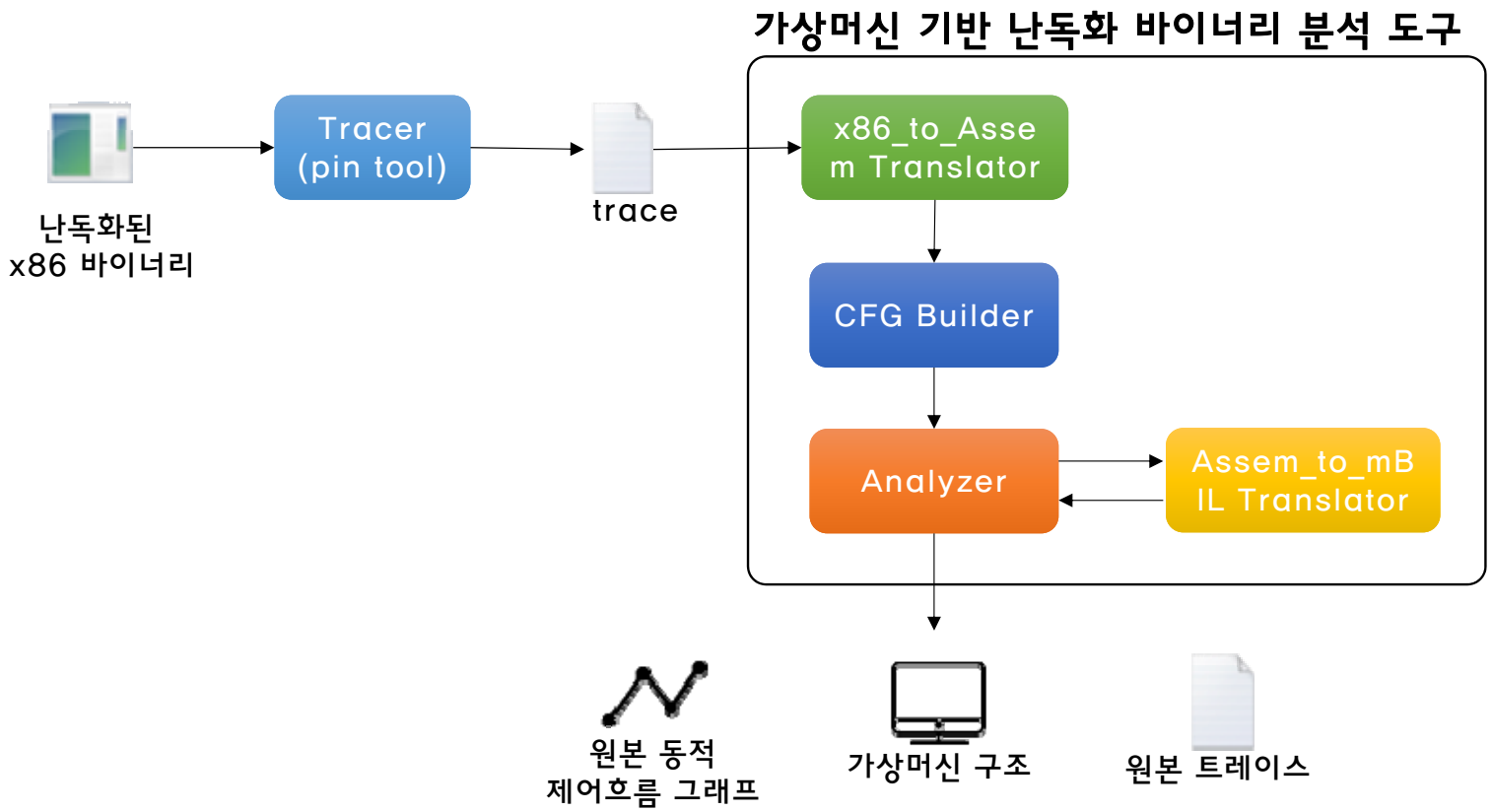
목표는 원본 프로그램 추출 및 구조 분석!

1. 상용 도구가 생성하는 가상머신을 구조화
2. 동적 분석을 통해 원본 프로그램에 대한 의미 있는 결과 도출
    I. 동적 제어흐름 그래프
    II. 실행 트레이스

상용도구의 가상머신 구조(VMP, CV)



도구 구성


가상머신 기반 난독화 바이너리 분석 도구
Tracer
(pin tool)
trace
x86_to_Asse
m Translator
CFG Builder
Analyzer
Assem_to_mB
IL Translator
난독화된
x86 바이너리
원본 동적
제어흐름 그래프
가상머신 구조
원본 트레이스

감사합니다

질문 있으신가요?