Measuring the Integrity of OS with Hardware Support

ROSAEC workshop at HKUST

Hyungon Moon, Hojoon Lee, Jihoon Lee, Yunheung Paek, and Brent Hoon Kang

2012-01-25

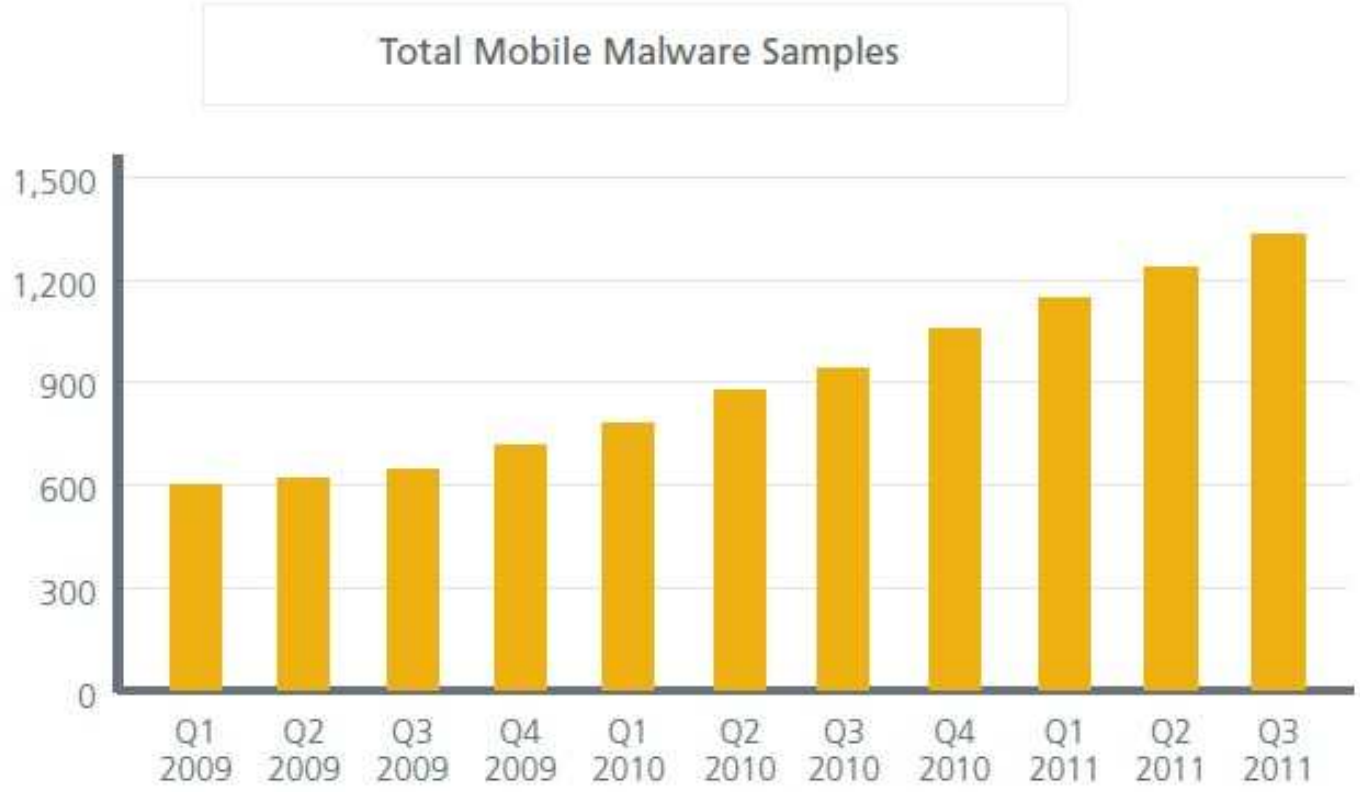
교육과학기술부
국가지정연구실
National Research Lab

연구 동기

- Computer system에서 발생할 수 있는 오류의 원인
  - ➔ Anything that causes abnormal behavior of SW or HW
  - Accidental Error
    - Software Bug
    - Hardware 오류
  - Intentional, Malicious Error
    - 악성코드의 공격
    - ➔ 어떻게 악성코드에의한 오류를 막는 지가 본 연구의 단기 목표

악성코드의 증가

- McAfee's Third Quarter 2011 Threats Report warns about rapid increase in mobile malware

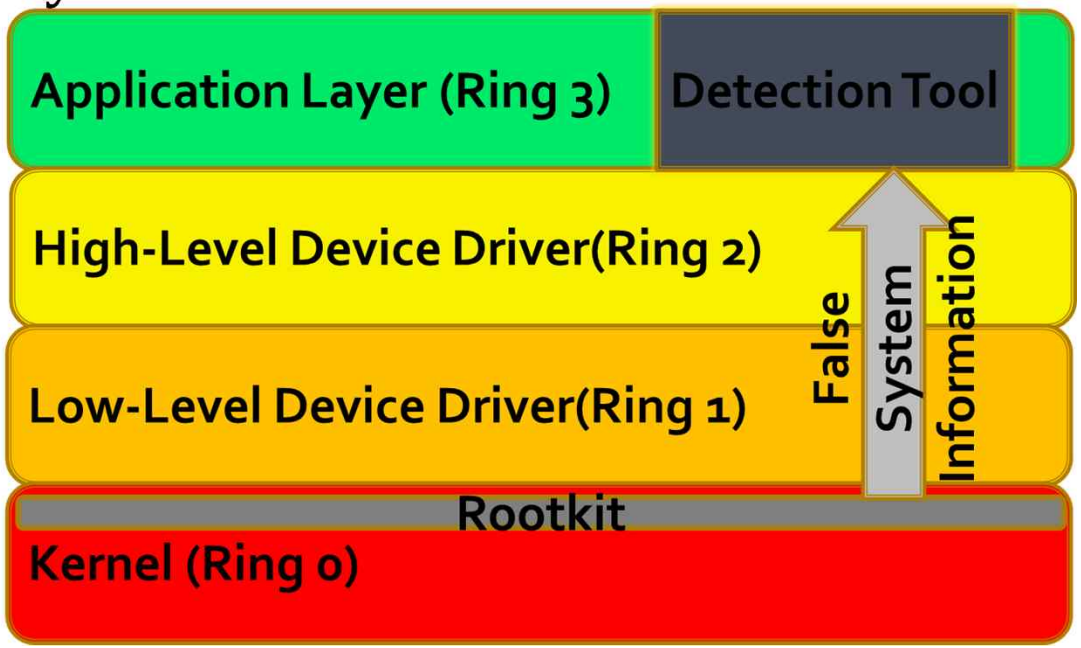


악성코드의 증가

- McAfee's 2012 Threats Predictions Report expects that
  - Malware-infected PCs have been used for cyber crimes, and this will happen the same in the mobile world
  - As mobile malware getting more and more elaborate, it will penetrate deep into the mobile operating systems
  - The attackers will start targeting mobile banking and more advanced techniques will appear

Challenges in Detection

- Rootkits manipulate victimized OS to report false information
- Detection/Recovery attempts within the system are not trustworthy



단기 목표를 해결 위한 방법론

컴퓨터 시스템의 무결성 (integrity) 을 실행 중 (runtime) 에 Hardware기반 으로 구성된 TCB (Trusted Computing Base) 를 이용하여 점검 (Measure)

기존의 연구

- VMM (Hypervisor) -based approaches
  - Various approaches for A1 (Secure Transactions) and A2 (Attack Detection)
    - *S. M. Lee et al.* 2008 [9]
      - Secure Transactions
    - *Secvisor* [7], *Petroni* and *Hicks* 2007 [8] -> A2
- Hardware-assisted approaches
  - *Copilot* [6] : Tried to achieve for A2
    - Use of PCI card for *x86* system
    - *Snapshot-based monitoring*
  - *ARM TrustZone* [10]: Provides A1
    - Use of modified processor, bus and additional IPs
    - Provides *NormalWorld* and *SecureWorld*

연구의 목적+

- 기존 연구의 한계
  - 순간 공격 (Transient attack)에 대한 탐지 불가능
  - 메인 시스템의 성능에 영향
    - Copilot: Memory bandwidth를 점유
    - HyperSentry: Processor의 time slot을 점유
- 이 한계를 극복하기 위한 방법
  - Processor와 Memory를 연결하는 hardware link의 traffic을 감시
    - 순간 공격 탐지 가능
    - 메인 시스템의 자원을 사용하지 않음

아이디어 구현

- Vigilare(2012?)
  - TCB: 우리가 추가한 Vigilare Processor와 그외 Hardware component들
  - Target: ARM processor와 Linux를 기반으로 하는 system
  - Processor와 Memory간의 traffic에 대해 다음을 점검
    - 쓰기 금지 구역에 쓰기 실행 (E.g. Kernel code, static jump tables)

앞으로

- 아이디어 구현 및 검증
- 정적 분석을 통한 감시 정책 생성
  - False Alarm 점검 코드 수행 보조
- 동적 분석의 도구로 활용