악성코드 검출을 위한 확장된 프로세서 트레이스 디코더 구조 연구★

강 승 애*, 김 영 수**, 김 종 현**, 김 현 철***

요 약

지금까지 오랜 시간 동안 범용 프로세서는 개발자에게 버그 수정을 할 수 있는 도구들을 제공하기 위해 전용 하드웨어/소프트웨어 트레이싱 모듈을 제공했다. 전용 하드웨어 트레이서는 성능 분석 및 디버깅에 모두 사용되는 막대한 양의 데이터를 로그로 실시간으로 생성한다. 프로세서 트레이스 (PT)는 CPU에서 실행되는 분기를 추적하는 Intel CPU를 위한 새로운 하드웨어 기반 추적 기능으로 최소한의 노력으로 모든 실행 코드의 제어 흐름을 재구성할 수 있다. 이러한 하드웨어 트레이스 기능들은 운영체제에 통합되어 프로파일 링 및 디버깅 메커니즘과의 긴밀한 통합이 가능하게 되었다. 본 논문에서는 윈도우 환경에서 PT가 제공하는 기능을 이용하여 실시간 트레이스 및 악성코드 검출을 위한 기본 데이터를 제공하는 확장된 PT 디코더 구조를 제안하였다.

A study of extended processor trace decoder structure for malicious code detection

Seungae Kang* , Youngsoo Kim**, Jonghyun Kim**, Hyuncheol Kim***

ABSTRACT

For a long time now, general-purpose processors have provided dedicated hardware / software tracing modules to provide developers with tools to fix bugs. A hardware tracer generates its enormous data into a log that is used for both performance analysis and debugging. Processor Trace (PT) is a new hardware-based tracing feature for Intel CPUs that traces branches executing on the CPU, which allows the reconstruction of the control flow of all executed code with minimal labor. Hardware tracer has been integrated into the operating system, which allows tight integration with its profiling and debugging mechanisms. However, in the Windows environment, existing studies related to PT focused on decoding only one flow in sequence. In this paper, we propose an extended PT decoder structure that provides basic data for real-time trace and malicious code detection using the functions provided by PT in Windows environment.

Key words : Tracing, Processor Trace, Flow Reconstruction, Malicious Code Detection, Flow Detection.

접수일(2018년 11월 30일), 게재확정일(2018년 12월 17일)

★본 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1D1A3B03035922)

★본 논문은 2018년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2016-0-00078, 맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발)

* 남서울대학교 스포츠건강관리학과 교수

** 한국전자통신연구원(ETRI)

*** 남서울대학교 컴퓨터소프트웨어학과 교수

1. 서 론

지금까지 오랜 시간 동안 범용 프로세서는 개발자에게 버그 수정을 할 수 있는 도구들을 제공하기 위해 전용 하드웨어/소프트웨어 트레이싱 모듈을 제공했다. 그러나 이러한 전용 하드웨어 트레이싱 모듈을 제공하는 추세는 데스크톱 또는 서버급 시스템보다는 임베디드 시스템의 경우에 더 두드러지는 점이었다. 이처럼 오랫동안 임베디드 컨트롤러는 디버깅 및 원격 트레이싱을 수행하는 데 사용할 수 있는 JTAG (Joint Test Action Group) 인터페이스를 지원했다. 이러한 특수 인터페이스는 일반적으로 프로세서 하드웨어에서 내장된 디버깅 지원 기능을 사용하는 인서킷 에뮬레이터와 직접 연결되어 원하는 기능을 수행하였다 [1].

한편 가상머신 (VM: Virtual Machine)을 기반으로 범용 하드웨어에서 다양한 프로그램을 수행하는 가상화 기술이 핵심 클라우드 인프라가 되었다. 따라서 클라우드 기반 가상화 인프라에서 다수의 VM이 협력하여 하나의 맞춤형 보안서비스를 제공하는 경우, 기존의 디버깅 툴이나 프로파일링 툴은 더는 성능측정이나 무결성 검증 도구로 사용할 수 없게 된다 [2][3].

이러한 단점을 해결하기 위해서 트레이싱 (Tracing) 방법을 사용하며, 트레이싱에서는 프로그램을 수행하는 것과 동시에 최소한의 오버헤드만으로 필요한 정보를 동시에 기록한다. 트레이싱에서는 필요한 정보를 수집하기 위해 브레이크 포인트와 비슷하게 트레이스 포인트를 정적/동적으로 설정할 수 있다. 주로 OS 커널이나 보안과 밀접하게 관련된 프로그램처럼 복잡하고 중요한 시스템 소프트웨어의 부분들이 트레이스 포인트의 주된 사용처이다. 솔라리스, 리눅스, 그리고 윈도우까지 대부분 tracing을 지원하며 tracing은 커널모드와 사용자 스페이스 모드에서 모두 동작할 수 있다.

현재 가장 많이 사용되는 하드웨어 기반 전용 트레이서로는 (그림 1), (그림 2)에서와 같이 ARM Coresight와 Intel PT (Processor Trace)가 대표적이며, 이와 같은 하드웨어 트레이서는 성능 분석 및 디버깅에 모두 사용되는 막대한 양의 데이터를 로그로 실시간으로 생성한다. 즉 프로세서 트레이스는 CPU에서 실행되는 분기를 트레이스 하는 Intel CPU를 위한 새로운 하드웨어 기반 트레이스 기능으로 최소한의 노력으로 모든 실행 코드의 제어 흐름을 재구성할 수 있다는 장점을 제공한다[4][5][6].

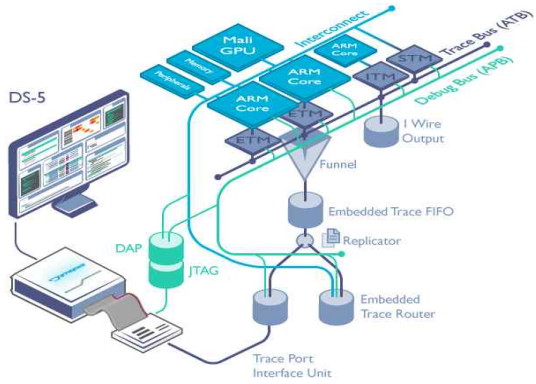
본 논문에서는 윈도우 환경에서 PT가 제공하는 기능을 이용하여 실시간 트레이스 및 악성코드 검출을 위한 기본 데이터를 제공하는 확장된 PT 디코더 구조를 제안하였다. 본 논문의 구성은 다음과 같다. 2장에서는 Intel PT 기반의 전용 하드웨어 트레이스와 관련된 선행 기술들의 조사 및 분석을 수행하였다. 3장에서는 실시간 트레이스 및 악성코드 검출을 위해 본 논문에서 제안하고 있는 확장된 프로세서 디코더의 구조를 기술하였다. 마지막으로 4장에서는 확장된 PT 디코더 구조와 관련된 결론과 향후 추가적인 연구 내용을 기술하였다.

2. 하드웨어 트레이스

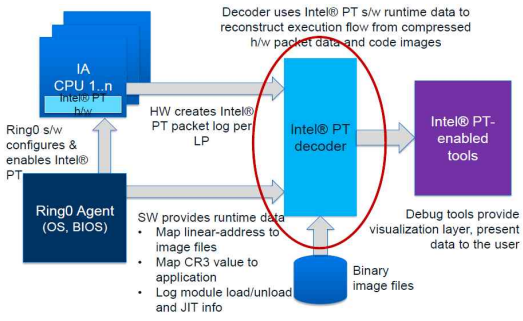
2.1 하드웨어 기반 트레이스와 PT

<표 1>에서와 같이 지속적인 하드웨어 기반 트레이서 방식이 개발되고 있는 동안 하드웨어 트레이스 프로그램은 운영체제에 통합 (e.g. Windows 10 Redstone 5에서의 WinIPT) 되어 프로파일 링 및 디버깅 메커니즘과의 긴밀한 통합이 가능해졌다. 이를 통해 서로 다른 프로세스에 대해 서로 다른 트레이스 버퍼를 사용하고 루트가 아닌 사용자가 하드웨어 기반 트레이스 기능을 사용할 수 있게 되었다.

PT는 시스템 동작에 최소한의 오버헤드만 주면서 소프트웨어 실행의 모든 정보를 저장하는 하드웨어 기능이다. 기본적으로 PT 디코더 소프트웨어는 5% 미만의 오버헤드만을 가지면서 트레이스 로그로부터 정확한 소프트웨어 실행 흐름 (flow)을 추출할 수 있다. 또한, 다른 트레이스와의 동기를 위해 사이클 카운트와 타임스탬프 정보 또한 저장할 수 있다 [7].



(그림 1) ARM Coresight 구조



(그림 2) Intel PT 구성

<표 2>에서 나타내고 있는 바와 같이 PT는 사용자 및 커널 레벨 코드의 명령어들을 볼 수 있으며, 명령 분기의 세부 단계까지 사이클 정보를 수집할 수 있다. 기존의 트레이스 정보와 비교하여 PT는 기록할 수 있는 트레이스 정보의 유형과 양에 대해 훨씬 빠르고 유연성이 있다. 이를 통해 매우 상세한 실행 흐름을 구성할 수 있으며 최대 정밀도 수준에서 성능 및 정확성 디버깅을 쉽게 할 수 있다 [5][6][7].

<표 1> 하드웨어 기반 트레이싱 방식

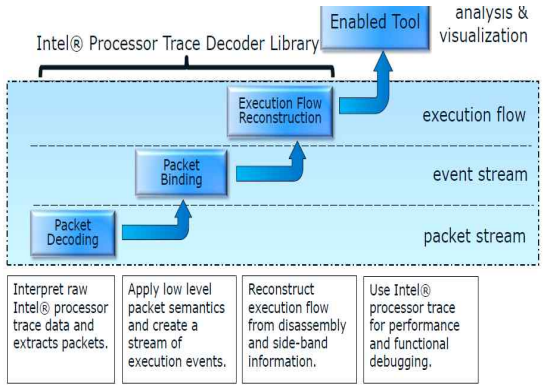
특징	내용
x86 (LBR/BTS)	- Intrusive - Limited timing information - Non-architectural
ARM	ETM/PTM
PowerPC	BHRB
x86	PT

<표 2> PT가 제공하는 특징과 능력

특징	내용
Control Flow	Exact Control Flow 정보
Mode 관련 정보	Timing Paging TSX state Execution 모드 Core to bus 클럭 ratio
Output	Highly Compressed Packet Output
Filtering	Privilege Level (CPL) Address Space (CR3)
Output Location	Memory

PT가 제공하는 장점 중의 하나로는 소스 코드의 변경이 필요하지 않다는 점이며 컨텍스트 스위칭 (Context switches), 주소 스페이스 변경 (address space modification) 등과 같은 운영체제의 사이드밴드 (sideband) 정보가 필요하지 않는다는 점이다. 단지 디코드 트레이스를 위해 오브젝트 코드만을 필요로 한다 [4].

PT의 제어 플로우 트레이싱 기능은 프로그램 플로우를 추론하기 위한 모든 브랜치 (branch)들을 기록하며, MSRs 들을 구성하고, 버퍼 셋업 및 트레이스 패킷을 생성한다. 아울러 (그림 3)에서 나타내고 있는 바와 같이 생성된 트레이스 패킷을 버퍼에 저장하거나 전달 계층으로 전달한다.

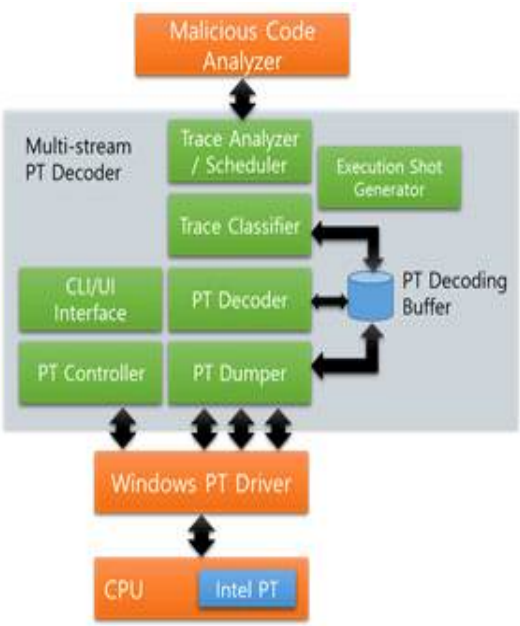


(그림 3) Intel PT 디코더 라이브러리와 디코딩

(그림 2)에서와 같이 NFV 아키텍처에서 RA를 수행하는 컴포넌트는 오케스트레이터 (Orchestrator)이며 오케스트레이터는 VNFM (VNF Manager)과 VIM (Virtualized Infrastructure Manager)을 통하여 VNFs 들을 관리한다. 오케스트레이터는 모든 조건을 고려하여 VNFs 할당하며 VNFM과 VIM을 통하여 이를 수행한다.

3. 확장된 프로세서 트레이스 디코더 구조

(그림 4)는 악성코드 검출을 위해 본 논문에서 제안하고 있는 확장된 프로세서 트레이스 디코더의 전체적인 구조를 보여준다. (그림 4)에서 볼 수 있듯이, 확장된 PT 디코더는 기존 윈도우 PT 커널 드라이버를 확장하여 다중 프로세서 트레이스 스트림까지 지원할 수 있는 구조로 되어 있다.



(그림 4) 확장된 프로세서 트레이스 디코더 구조

(그림 4)에서 볼 수 있듯이 제안된 확장 프로세서 트레이스 디코더는 트레이스 분류기, PT 디코더, PT 덤퍼, PT 컨트롤러, CLI 인터페이스, 트레이스 분석기 및 실행 샷 생성기와 같은 요소로 구성된다.

우선 PT 덤퍼는 윈도우 커널 드라이버로부터 추출된 PT 패킷을 지정된 방식에 따라 PT 디코딩 버퍼에 저장하는 기능을 수행한다. 즉, PT 덤퍼는 커널 드라이버로부터 수신된 PT 패킷을 PT 컨트롤러 구성에 따라 단일 스트림 또는 다중 스트림으로 PT 디코드 버퍼에 저장한다.

PT 제어기는 PT의 다양한 동작과 관련된 명령을 윈도우 커널 드라이버로 전송하는 기능을 수행한다. 즉, PT 컨트롤러는 덤프 시작 및 중지, 덤프 버퍼 생성 및 삭제, 멀티 스트림 처리를 위한 다양한 PT 구성 작업을 수행한다.

CLI 인터페이스는 PT 컨트롤러를 제어하고 다양한 입력값을 제공하는 명령행 인터페이스를 제공한다.

PT 디코더는 PT 디코딩 버퍼에 저장된 PT 패킷을 실시간으로 디코딩하여 이를 디스어셈블 된 형태로 저장한다.

PT 추적 분류기 및 추적 분석기는 악성코드 탐지기가 필요로 하는 정보를 실시간으로 추출한다. 예를 들어 함수 호출 수와 같은 다양한 정보가 추출된다.

마지막으로, PT 디코딩 버퍼는 멀티 스트림 디코더를 구성하는 다양한 요소 간에 데이터를 전송하는 데 사용된다. 파일 형식이 아닌 PT 디코더 요소 간에 메모리 주소 만 전송된다.

한편 경우에 따라 실행 추적만으로는 악성코드를 탐지할 수 없다. 따라서 본 논문에서는 특정 시간 실행 추적의 샷을 저장하고 비교하여 악성코드를 탐지하는 방법을 제안하였다. 실행 샷 생성기는 PT 컨트롤러가 지정한 시간 단위로 실행 샷을 저장한다.

4. 결 론

하드웨어 기반 트레이스 기능은 성능측정, 디버깅은 물론 시스템의 전체적인 실행을 파악하여 악성코드와 같은 비정상적인 실행 흐름을 추적은 중요한 도구이다. 하드웨어 트레이서 중 PT는 CPU에서 실행되는 분기를 추적하는 Intel CPU를 위한 새로운 하드웨어 기반 트레이서 기능으로 최소의 노력으로 모든 실행 코드의 제어 흐름을 재구성할

수 있다. 또한, 사용자 및 커널 수준 코드를 볼 수 있으며 명령 분기의 세부 단계까지 주기 정보를 수집 할 수 있다.

WindowsIntelPT 드라이버는 마이크로소프트 윈도우 환경에서 Intel Skylake 아키텍처를 지원하는 CPU의 PT 기능을 구현하고 있고, 본 논문은 이러한 WindowsIntelPT 프로그램을 수정하여 악성코드 검출을 위한 확장된 프로세서 트레이스 디코더 구조를 제안하였다.

참고문헌

[1] Napoleon C. Paxton, “Cloud Security: A Review of Current Issues and Proposed Solutions,” International Conference on Collaboration and Internet Computing (CIC), pp. 452-455, 2016.

[2] Tahira Mahboob; Maryam Zahid; Gulnoor Ahmad, “Adopting information security techniques for cloud computing—A survey,” International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), pp. 7-11, 2016.

[3] Jörg Thalheim; Pramod Bhatotia; Christof Fetzer, “INSPECTOR: Data Provenance Using Intel Processor Trace (PT),” International Conference on Distributed Computing Systems (ICDCS), pp. 25-34, 2016.

[4] Khalid El Makkaoui; Abdellah Ezzati; Abderrahim Beni-Hssane; Cina Motamed, “Cloud security and privacy model for providing secure cloud services,” 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), pp. 81-86, 2016.

[5] Bob Duncan; Alfred Bratterud; Andreas Happe, “Enhancing cloud security and privacy: Time for a new approach?,” International Conference on Innovative Computing Technology (INTECH), pp. 110-115, 2016.

[6] Sin-Fu Lai; Hui-Kai Su; Wen-Hsu Hsiao; Kim-Joan Chen, “Design and implementation of cloud security defense system with software defined networking technologies,” International Conference on Information and Communication Technology Convergence (ICTC), pp. 292-207, 2016.

[7] Andi Kleen, “Simple Intel CPU processor tracing on Linux,” https://github.com/andikleen/simple-pt

〔저자소개〕

강 승 애 (Seuungae Kang)
1995년 2월 이화여자대학교 학사
1997년 8월 이화여자대학교 석사
2006년 8월 이화여자대학교 박사
2006년 9월 ~ 현재 남서울대학교 스포츠건강관리학과 교수

email : sahome@nsu.ac.kr

김 영 수 (Youngsoo Kim)
1998년 2월 성균관대학교 학사
2000년 2월 성균관대학교 석사
2009년 8월 성균관대학교 박사
2000년 2월 ~ 현재 한국전자통신연구원 책임연구원
email : blitzkrieg@etri.re.kr

김 종 현 (Jonghyun Kim)
2000년 2월 오클라호마 주립대 석사
2005년 2월 오클라호마 주립대 박사
1995년 ~ 1998년 삼성전자 연구원
2005년 ~ 현재 한국전자통신연구원 책임연구원
email : jhk@etri.re.k

김 현 철 (Hyuncheol Kim)
1990년 2월 성균관대학교 학사
1992년 2월 성균관대학교 석사
2005년 8월 성균관대학교 박사
2006년 9월 ~ 현재 남서울대학교 컴퓨터소프트웨어학과 교수
email : hckim@nsu.ac.kr