Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.6, Dec. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.6.1255

비 캐시 하드웨어 이벤트를 이용한 캐시 부채널 공격 실시간 탐지*

김 호 동,[1†] 허 준 범[2‡]
[1,2]고려대학교 (대학원생, 교수)

Real-Time Detection of Cache Side-Channel Attacks Using Non-Cache Hardware Events*

Hodong Kim,[1†] Junbeom Hur[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

캐시 부채널 공격은 CPU의 공유 캐시 자원을 이용하여 시스템에서 민감한 정보를 획득해내는 공격 유형이다. 최근 모바일 시스템에서 클라우드까지 다양한 환경에 공격이 가해짐에 따라 많은 탐지 전략이 제안되었다. 기존의 캐시 부채널 공격들은 많은 수의 캐시 이벤트를 발생시키는 특징을 가지고 있었기 때문에 기존의 탐지 기법은 대부분 캐시 이벤트를 주의 깊게 모니터링하는 것에 기반하여 설계되었다. 그러나 최근에 제안된 공격은 공격 중에 캐시 이벤트를 적게 유발하는 경향이 있다. 예를 들어 PRIME+ABORT 공격은 캐시에 접근하여 액세스 시간을 측정하는 대신 Intel TSX를 활용한다. 이러한 특징으로 인해 캐시 이벤트 기반 탐지 기법은 해당 공격을 탐지하기 어렵다. 본 논문에서는 PRIME+ABORT 공격에 대한 심층 분석을 수행하여 캐시 이벤트 이외에 공격 탐지에 활용 가능한 유용한 하드웨어 이벤트를 밝힌다. 이를 기반으로, PRIME+ABORT 공격 탐지 기법인 PRIME+ABORT Detector를 제시하고, 실험을 통해 제안한 탐지 기법이 0.3%의 성능 오버헤드로 99.5%의 탐지 성공률을 달성할 수 있음을 보인다.

ABSTRACT

Cache side-channel attack is a class of attacks to retrieve sensitive information from a system by exploiting shared cache resources in CPUs. As the attacks are delivered to wide range of environments from mobile systems to cloud systems recently, many detection strategies have been proposed. Since the conventional cache side-channel attacks are likely to incur tremendous number of cache events, most of the previous detection mechanisms were designed to carefully monitor mostly cache events. However, recently proposed attacks tend to incur less cache events during the attack. PRIME+ABORT attack, for example, leverages the Intel TSX instead of accessing cache to measure access time. Because of the characteristic, attack detection mechanisms based on cache events may hardly detect the attack. In this paper, we conduct an in-depth analysis of the PRIME+ABORT attack to identify the other useful hardware events for detection rather than cache events. Based on our finding, we present a novel mechanism called PRIME+ABORT Detector to detect the PRIME+ABORT attack and demonstrate that the detection mechanism can achieve 99.5% success rates with 0.3% performance overhead.

Keywords: Real-time attack detection, Cache side-channel attack, PRIME+ABORT

Received(10. 05. 2020), Modified(11. 18. 2020), Accepted(11. 18. 2020)

* 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-00533, 컴퓨터 프로세서의 구조적 보안 취약점 검증 및 공격 탐지대응), (ICT명품인재양성 사업 IITP-2020-0-01819)

† 주저자, hdkim@isslab.korea.ac.kr
‡ 교신저자, jbhur@korea.ac.kr(Corresponding author)

I. 서 론

지난 10년 간 CPU의 공유 캐시 메모리를 이용하여 암호화 라이브러리의 비밀 키와 같은 시스템에서 민감한 정보를 획득할 수 있는 여러 캐시 부채널 공격이 제안되었다[1]-[4]. 캐시 부채널 공격이 광범위한 최신 시스템에서 실질적인 위협을 일으키고 있기 때문에 실제로 캐시 부채널 공격으로부터 데이터를 보호하는 것은 복합적이고 어려운 분석을 요한다[5]-[7].

전통적인 캐시 부채널 공격들은 일반적으로 캐시 이벤트 시간차 부채널(timing side-channel)을 유도하도록 설계되었다. 따라서 기존에 제안된 캐시 부채널 공격 탐지 기법들은 캐시 부채널 공격과 관련된 하드웨어 이벤트를 모니터링함으로써 공격을 탐지하도록 설계되었는데, 특히 캐시 미스(cache miss) 또는 히트(cache hit)와 같은 캐시 액세스 이벤트를 모니터링함으로써 캐시 부채널 공격을 탐지하였다. 그러나 FLUSH+FLUSH[3]와 PRIME+ABORT[4] 같은 최근에 발표된 캐시 부채널 공격은 다른 방식으로 하드웨어 이벤트를 활용하여 공격 중에 캐시 이벤트를 덜 발생 시키도록 설계되다. 예를 들어, PRIME+ABORT 공격은 캐시 히트 또는 미스 이벤트로 인해 발생하는 시간차 부채널 대신 Intel TSX 하드웨어를 활용한다. 따라서 기존에 발표된 대부분의 캐시 이벤트 모니터링 기반 탐지 기법들은 적은 수의 캐시 이벤트를 유발하는 '은밀한' 공격을 실시간으로 탐지하는 것은 매우 어렵다. 따라서 공격과 밀접한 관련이 있어 공격 탐지 측면에서 효과적인 적절한 하드웨어 이벤트를 찾는 것은 다양한 캐시 부채널 공격에 대한 탐지 기법 개발에서 가장 중요한 목표 중 하나이다.

따라서 본 논문에서는 이러한 최근의 은밀한 캐시 부채널 공격을 보다 효율적으로 탐지할 수 있는 비 캐시 이벤트 기반의 실시간 공격 탐지 기법에 대해 제안하고, 성능 분석을 통해 제안한 기법의 효율성을 검증한다. 구체적으로 본 연구는 다음의 기여를 가지고 있다.

1. 본 논문에서는 PRIME+ABORT 캐시 부채널 공격에 관련된 하드웨어 이벤트에 대한 심층 분석을 수행한다.
2. 분석한 하드웨어 이벤트를 이용해 최초로 PRIME+ABORT 공격에 대한 실시간 탐지 기법을 제안한다.
3. 성능분석을 통해 제안하는 기법이 실용적인 환경에서 PRIME+ABORT 공격을 99.5%의 정확도로 탐지할 수 있으며, 0.3%의 성능 오버헤드를 가짐을 보인다.

II. 배 경

2.1 Hardware Performance Counter

Hardware Performance Counter (HPC)는 low-level 성능 분석을 위해 단위 시간 동안 발생한 캐시 미스, 캐시 히트, 잘못 예측된 분기 등과 같은 CPU 관련 하드웨어 이벤트의 횟수를 저장할 수 있도록 CPU에 내장된 특수 목적 레지스터이다.

Intel Process Counter Monitor (PCM)은 Intel 아키텍처의 다양한 하드웨어 이벤트를 모니터링하기 할 수 있도록 Intel 아키텍처의 HPC 인 Performance Monitoring Unit (PMU)에 액세스하기위한 유틸리티 샘플을 제공한다[8],[9]. HPC는 시스템에서 실행되고있는 이벤트를 이해할 수 있는 다양한 정보를 제공할 수 있게 설계되었으므로 이를 분석하여 시스템에서 공격을 감지하는 데 사용할 수 있다.

2.2 캐시 부채널 공격

최신 CPU는 성능 최적화를 위해 많은 기능을 지원한다. 이러한 최적화 기술 중 계층적 캐시, 메모리 중복 제거, Intel TSX는 공유 리소스를 더 잘 사용하기 위해 도입되었다.

캐시의 계층 구조는 크기와 액세스 속도가 다양한 여러 계층의 캐시 레이어로 구성된다. 예를 들어 Intel Skylake CPU 아키텍처에서는 용량이 작지만 빠른 L1 캐시, 중간 크기와 및 중간 속도의 L2 캐시, 비교적 크지만 느린 속도의 L3 last-level 캐시(LLC)로 구성된 3계층으로 이뤄져있다. 캐시 계층 구조의 여러 코어에서 자주 액세스하는 데이터의 경우, 각 코어에 대해 동일한 내용의 여러 복사본을 두는 대신 메모리 중복 제거 기능을 사용하여 LLC에 해당 데이터의 단일 적재본을 두고 여러 코어들이 공유하여 사용하는 방식으로 캐시의 효율을 높일 수 있다.

Intel TSX는 다중 스레드 프로그램 실행시 더 나은 동시성을 위해 하드웨어 트랜잭션 메모리를 지원하는 x86 instruction set architecture (ISA)의

확장이다. 이 기능이 지원되면 write set에 대한 write 또는 read set에 대한 eviction으로 인해 충돌이 발생하기 전까지는, 여러 스레드가 critical section을 병렬로 실행할 수 있다. Critical section에서 충돌이 감지되면 실행을 중단시키고 동적으로 삭제한다.

2.2.1 FLUSH+RELOAD 공격

여러 프로세서들은 메모리 중복 제거기능을 통해 동일한 데이터의 단일 적제본을 함께 공유하여 사용할 수 있다. FLUSH+RELOAD 공격은 clflush 명령[2]을 사용하여 이 기능을 이용한다. 공격을 시작하기 위해 공격자는 clflush 명령을 호출하여 희생자와 공유하고있는 공격대상 캐시 라인에서 데이터를 flush한다. 그 후 희생자가 대상 캐시 라인에 액세스할 수 있는 시간동안 기다린다. 마지막으로 공격자는 데이터를 공격대상 캐시 라인에 다시 로드(reload)하고 액세스 시간을 측정한다. 만약 희생자가 공격자가 실행 한 flush 및 reload 작업간에 해당 캐시라인에 액세스하면 캐시 히트가 발생하여 그렇지 않았을 경우에 비해 상대적으로 액세스 시간이 단축된다. 이러한 타이밍 차이를 수집하면 공격자가 캐시 라인을 통해 희생자의 실행 패턴을 획득할 수 있다.

2.2.2 PRIME+ABORT 공격

PRIME+ABORT의 가장 두드러진 특징은 캐시 히트와 미스 이벤트 사이의 시간차 대신 Intel TSX 하드웨어를 공격에 이용하는 것이다[4]. LLC에 대한 PRIME+ABORT 공격의 구체적인 과정은 prime 단계에서 캐시 세트를 공격자의 데이터로 완전히 채울 수 있도록 충분한 캐시 라인에 액세스하여 공격대상 캐시 세트에 데이터를 불러오는 것으로 시작된다. Abort 단계에서 공격자는 희생자가 자신의 데이터를 불러오기 위해 공격자의 데이터를 evict 하려는 시도를 통해 트랜잭션을 중단시킬 때까지 기다린다. 이 순간 Intel TSX 하드웨어는 즉시 하드웨어 콜백으로 트랜잭션의 중단을 공격자에게 알린다. 공격자는 prime 단계 이후 피해자가 액세스할 때까지 기다릴 수 있기 때문에 캐시 활동 유발을 줄일 수 있다.

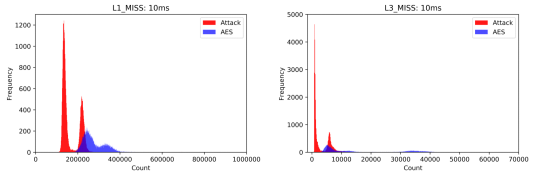


Fig. 1. Cache-related hardware events of PRIME+ABORT attack

III. PRIME+ABORT Detector

3.1 PRIME+ABORT 공격 분석

PRIME+ABORT 공격의 캐시 활동을 분석하기 위해 다양한 조건에서 L1 및 L3 캐시 미스 횟수를 측정하였다: (1) GnuPG 1.4.13의 RSA 복호화 실행; (2) RSA 복호화 중 FLUSH+RELOAD 공격 실행; (3) Openssl 1.0.2j의 AES 복호화 실행; (4) AES 복호화 중 PRIME+ABORT 실행.

RSA에 대한 FLUSH+RELOAD 공격은 많은 캐시 미스를 유발하는 것으로 널리 알려져 있으므로 캐시 미스가 적은 경우와 비교하기 위해 선택하였다. 첫 번째와 두 번째 조건에서의 측정결과를 살펴볼 때, FLUSH+RELOAD 공격이 있는 경우 RSA 복호화를 단독으로 수행하는 경우보다 25% 많은 캐시 미스가 발생하는 것으로 나타났다. 이러한 된 캐시 미스의 증가는 진행중인 공격의 주요한 지표가 될 수 있다. 그러나 세 번째 및 네 번째 조건에서 두 조건 모두 비슷한 캐시 미스를 발생시키는 것으로 관찰되었다. 이는 PRIME+ABORT 공격 탐지에 캐시 미스 수를 사용하기 어렵다는 것을 보여준다.

Fig. 1은 세 번째 및 네 번째 조건하에 10ms의 시간 간격으로 캡처한 350,000개의 캐시 활동 샘플을 나타낸다. 두 조건의 캐시 미스의 분포가 서로 겹치는 영역이 있어 횟수만을 가지고 어떤 조건에 해당하는 경우인지 했는지 구별하기 어려움을 알 수 있다. 이에 더하여, 실제 설정에서 많은 선량한 프로그램들이 캐시 이벤트를 발생시키기 때문에 겹치는 영역은 더 크게 나타날 수 있다. 따라서 캐시 이벤트를

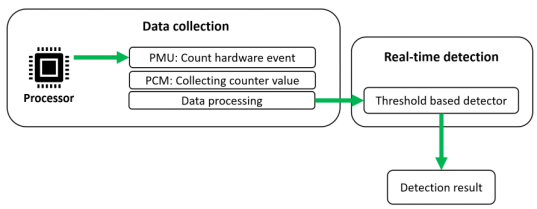


Fig. 2. Overview of PRIME+ABORT Detector

관찰하는 것만으로 PRIME+ABORT 공격을 구분하는 것은 매우 어렵다.

3.2 PRIME+ABORT Detector의 설계

Fig. 2는 하드웨어 이벤트 수를 수집하는 data collection 부분과 수집된 데이터를 처리하는 real-time detection의 두 부분으로 구성된 PRIME+ABORT Detector의 구조를 나타낸 것이다. 탐지과정은 모니터링할 하드웨어 이벤트를 의미하는 event number와 umask를 사용하여 PMU를 프로그래밍하는 것으로 시작된다. 다음으로 프로그래밍된 PMU는 각 코어에서 발생하는 지정된 하드웨어 이벤트의 수를 측정한다. 측정된 값은 Intel PCM의 샘플코드를 이용해 구현된 PCM 컴포넌트를 통해 수집되며, 이 컴포넌트는 이벤트 식별, PMU 프로그래밍, PMU 데이터 가져오기를 포함한다. Data processing 컴포넌트는 PMU에서 수집한 값을 일정한 포맷으로 가공하기 위한 역할을 한다. 이를 위해 csv 파일 형식으로 시간의 순서에 따라 이벤트의 횟수를 기록하도록 구성하였다. 마지막으로 real-time detection 단계에서 threshold 기반 기법을 사용하여 공격이 현재 진행 중인지 판별한다.

구현 환경으로 16GB RAM과 Intel i7-6700 Skylake 프로세서가 장착된 시스템을 사용하였다. Intel i7-6700 Skylake 프로세서는 하이퍼 스레딩을 지원하는 3.40GHz 쿼드코어 구성이고, 8MB의 LLC를 가지고 있다. 2019년 11월 12일 버전의 Intel PCM을 기반으로 PRIME+ABORT Detector를 구현하였고 Ubuntu 16.04 LTS 에서 실험을 진행하였다. Skylake 프로세서는 PMU를 이용해 최대 세 개의 이벤트를 모니터링 할 수 있도록 지원하므로 이를 최대한 활용하기 위해 PRIME+ABORT 공격을 감지하는데 효과적인 세 개의 이벤트를 선택하였다.

탐지에 유용한 하드웨어 이벤트를 선택하기 위해 AES 복호화를 실행하는 경우와 PRIME+ABORT 공격을 실행하는 동안 여러 하드웨어 이벤트 수를 수집하였다. Fig. 3은 10ms 간격으로 측정한 네 가지 이벤트의 각 35,000샘플의 분포를 나타낸다. 그림에서 x 축과 y 축은 각각 측정된 이벤트 수와 각 횟수의 빈도를 나타낸다. 파란색과 빨간색 영역은 각각 AES 복호화와 PRIME+ABORT 공격의 경우를 나타낸다. 다양한 하드웨어 이벤트 중 RTM_RETIRED.START, RTM_RETIRED.ABORTED, TX_MEM.ABORT_CAPACITY_WRITE의 세 가지 이벤트가 그림과 같이 가장 눈에 띄는 분포를 보여주었다. 따라서 이 이벤트들을 탐지 기법에 적용하기로 하였다.

3.3 PRIME+ABORT 공격 실시간 탐지

AES 복호화와 PRIME+ABORT 공격을 각각 약 11,200회, 8,500회 수행하는 동안 선택한 하드웨어 이벤트 수를 1초마다 측정하였다. AES 복호화의 경우, RTM_RETIRED.START는 평균 3,807회 발생했으며 최대값과 최소값은 99,825회와 1,323회였다. RTM_RETIRED.ABORTED는 평균 106회, 최대 917회, 최소 32회 발생했다. TX_MEM.ABORT_CAPACITY_WRITE는 평균 17회, 최대 916회, 최소 0회 발생했다. PRIME+ABORT 공격의 경우, RTM_RETIRED.START는 평균 151,790회 발생했으며 최대 785,806회, 최소 129,431회

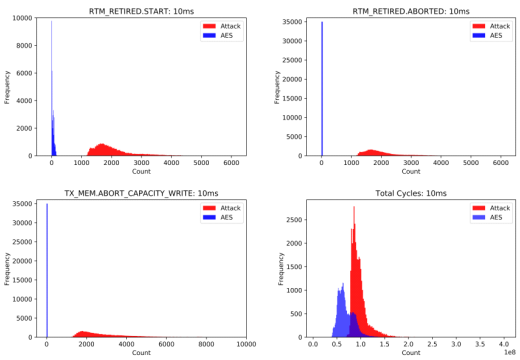


Fig. 3. Distribution of hardware event counts

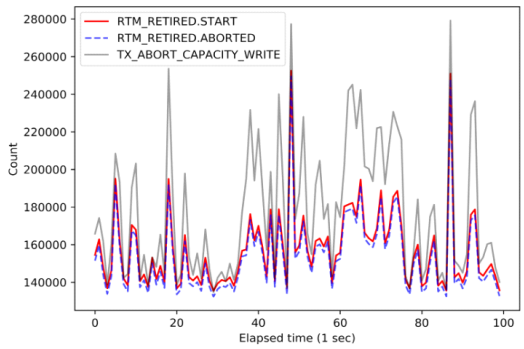


Fig. 4. Monitored hardware events of PRIME+ABORT attack

발생했다. RTM_RETIRED.ABORTED는 평균 148,503회, 최대 727,473회, 최소 128,208회 발생했다. TX_MEM.ABORT_CAPACITY_WRITE는 평균 169,891회, 최대 978,230 회, 최소 128,341회 발생했다.

Fig. 4는 PRIME+ABORT 공격시 수집한 측정값 중, 시간상 연속된 100개의 샘플을 나타낸 것이다. 세 이벤트는 비슷한 시점에 증감을 보이는 패턴이 있다. TX_MEM.ABORT_CAPACITY_WRITE가 가장 높은 최고점을 보이고, RTM_RETIRED.START와 RTM_RETIRED.ABORTED는 비교적 적은 차를 보인다. 우리는 이 패턴들을 이용하여 PRIME+ABORT 공격을 특정하는 threshold로 이용하고자 하였다.

하드웨어 이벤트 수 측정값은 AES 복호화와 PRIME+ABORT 공격의 두 경우에서 중첩되지 않았으며, 공격의 경우에서 모든 하드웨어 이벤트값이 매우 큰 값을 보였다. 이는 AES 복호화 과정이 PRIME+ABORT 공격만큼 Intel TSX를 활발히 활용하지 않기 때문이다. PRIME+ABORT 공격시 수집한 값에 공격대상인 AES 복호화 실행의 이벤트 값이 포함되어 있음에도, 하드웨어 이벤트값의 증감에 주는 영향은 미미하였다. 하지만, AES와 같은 선량한 프로그램이 Intel TSX를 더욱 적극적으로 활용하는 경우에는 공격시 수집한 값의 범위에 영향을 줄 수 있고, 결과적으로 threshold를 이용한 탐지결과에 영향을 줄 수 있다. 이 영향을 줄이기 위해 공격시 수집한 값과 선량한 프로그램인 AES 복호화 과정에서 수집한 값의 평균값의 차를 구해 값을 가공하였다.

가공한 값을 이용해 PRIME+ABORT공격을 탐지하기 위해 정한 threshold는 두 가지이다. 첫 번째로, RTM_RETIRED.START와 RTM_RETIRED.ABORTED 값의 차가 보이는 값의 패턴을 탐지하고자 500회인 경우부터 시작해 500회씩 증가시키며 가장 성공적인 결과를 보인 값을 threshold로 정하였다. 두 번째로, TX_MEM.ABORT_CAPACITY_WRITE가 가장 큰 최고점을 보일 때 RTM_RETIRED.ABORTED와 TX_MEM.ABORT_CAPACITY_WRITE 간의 차가 보이는 값의 패턴을 탐지하고자 TX_MEM.ABORT_CAPACITY_WRITE가 50,000회인 경우부터 5,000씩 증가시키고, RTM_RETIRED.ABORTED와 TX_MEM.ABORT_CAPACITY_WRITE의 차가 100,000회인 경우부터 5000회씩 증가시키며 가장 성공적인 값을 threshold로 정하였다. 추가적으로 PRIME+ABORT 공격시 수집한 값의 초반부에 있는 공격 준비단계에서 평균 5,000,000 회의, RTM_RETIRED.START 이벤트가 발생하는 특징이 있었다. 준비단계를 구분하기 위해 RTM_RETIRED.START 3,000,000회부터 시작하여 모든 준비단계를 포함할 때까지 100,000회씩 값을 증가시키며 가장 성공적인 결과를 보인 값을 threshold를 선정하였다.

PRIME+ABORT 공격을 탐지하기 위해 측정값을 분석하여 다음과 같이 threshold를 선택하였다.

- RTM_RETIRED.START의 값이 3,500,000회보다 큰 경우 공격 준비 단계이다.
- RTM_RETIRED.START 값이 100,000회보다 크고 RTM_RETIRED.START 와 RTM_RETIRED.ABORTED 값의 차가 4,000회 미만이면 실제 공격단계이다.
- TX_MEM.ABORT_CAPACITY_WRITE 값이 80,000회보다 크고 RTM_RETIRED.ABORTED와 TX_MEM.ABORT_CAPACITY_WRITE 값의 차이가 150,000회 미만인 경우 실제 공격단계이다.

이를 이용해 콘솔 모드에서 감지 결과를 1초마다 출력하도록 PRIME+ABORT Detector를 구현했다. 30분간 실행한 실험에서 PRIME+ABORT Detector는 99.5%의 정확도와 0.3%의 성능 오버헤드로 PRIME+ABORT 공격을 탐지하는 결과를 내었다.

IV. 결 론

이전에 발표된 캐시 부채널 공격 탐지 기법은 주로 캐시 이벤트 모니터링을 기반으로 한다. FLUSH+FLUSH 공격은 기존 탐지 기법으로는 탐지할 수 없는 은밀한 캐시 부채널 공격의 출현을 알렸다. 이 연구에서는 비 캐시 이벤트를 이용하면 이러한 은밀한 캐시 부채널 공격을 효과적으로 탐지할 수 있음을 실험을 통해 보였다. FLUSH+RELOAD와 같은 기존의 캐시 부채널 공격에 비해 캐시 활동이 적은 PRIME+ABORT 공격을 분석하였고, 이를 바탕으로 PRIME+ABORT 공격을 탐지하는데 유용한 비 캐시 하드웨어 이벤트를 찾아내었다. 이를 기반으로 실시간 탐지 기법을 설계했으며, 실험을 통해 99.5%의

탐지 정확의 성능을 보였다.

본 연구는 탐지 기법을 성공적으로 적용하기 위해 세부 설정이 필수적이라는 한계를 가지고 있다. 향후에 머신러닝 기법을 도입하여 세부 설정 과정을 효율적으로 자동화하는 방법으로 이 한계를 극복하기 위한 추가 연구를 수행할 것이다.

References

[1] F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee, "Last-level cache side-channel attacks are practical," Proceedings of the 2015 IEEE Symposium on Security and Privacy, pp. 605 - 622, May 2015

[2] Y. Yarom and K. Falkner, "Flush+ reload: a high resolution, low noise, l3 cache side-channel attack," Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), pp. 719 - 732, Aug. 2014.

[3] D. Gruss, C. Maurice, K. Wagner, and S. Mangard, "Flush+flush: a fast and stealthy cache attack," Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 279 - 299. Sep. 2016.

[4] C. Disselkoen, D. Kohlbrenner, L. Porter, and D. Tullsen, "Prime+abort: A timer-free high-precision l3 cache attack using intel tsx," Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), pp. 51 - 67, Aug. 2017.

[5] G. Irazoqui, M. S. Inci, T. Eisenbarth, and B. Sunar, "Wait a minute! a fast, cross-vm attack on aes." in International Workshop on Recent Advances in Intrusion Detection, pp. 299 - 319, Sep. 2014

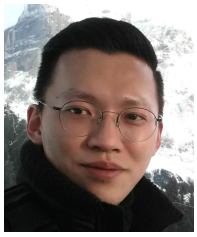
[6] D. Wang, A. Neupane, Z. Qian, N. B. Abu-Ghazaleh, S. V. Krishnamurthy, E. J. Colbert, and P. Yu, "Unveiling your keystrokes: A cachebased side-channel attack on graphics libraries." Proceedings of the The Network and Distributed System Security Symposium 2019 (NDSS, 2019), Feb. 2019.

[7] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-tenant sidechannel attacks in paas clouds," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 990 - 1003, Nov. 2014.

[8] Intel, Intel 64 and IA-32 Architectures Performance Monitoring Events, 2017

[9] Intel, Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2, 2016

〈저 자 소 개〉



김 호 동 (Hodong Kim) 학생회원
2017년 8월: 중앙대학교 컴퓨터공학과 졸업
2020년 2월: 고려대학교 컴퓨터학 석사
2020년 3월~현재: 고려대학교 컴퓨터학과 박사과정
〈관심분야〉 정보보호, 캐시 부채널, 시스템 취약점, 보안을 위한 기계학습

허 준 범 (Junbeom Hur) 종신회원
2001년 2월: 고려대학교 컴퓨터공학 졸업
2005년 8월: 한국과학기술원 전산학 석사
2009년 8월: 한국과학기술원 전산학 박사
2009년 9월~2011년 8월: University of Illinois at Urbana-Champaign 박사후 연구원
2011년 9월~2015년 2월: 중앙대학교 컴퓨터공학부 조교수
2015년 3월~2016년 8월: 고려대학교 컴퓨터학과 조교수
2016년 9월~현재: 고려대학교 컴퓨터학과 부교수
〈관심분야〉 응용 암호, 네트워크 보안, 클라우드 보안, 시스템 취약점