Intel SGX 취약점 연구 동향

김경호*, 권혁동*, 김현준*, 서화정*
*한성대학교 IT융합공학과
e-mail:pgm.kkh@gmail.com

Intel SGX Vulnerability Research Trends

Kyung-Ho Kim*, Hyeok-Dong Kwon*, Hyun-Jun Kim*, Hwa-Jeong Seo*
*Divison of IT convergence engineering, Hansung University

요 약

2017년에 대표적인 프로세서 취약점인 Meltdown과 Spectre가 발표됨에 따라 Intel의 CPU에서 다양한 취약점이 노출되었다. 이 취약점은 하드웨어 기반으로 신뢰 할 수 있는 환경을 보장해주는 TEE(Trusted Execution Environment) 기술을 사용하는 Intel의 SGX(Software Guard Extensions)에서도 유효하다. 따라서 이러한 취약점을 이용하여 신뢰 할 수 있는 환경에서 데이터의 무결성 및 기밀성을 훼손하는 다양한 공격 연구가 활발히 이루어지고 있다. 본 논문에서는 Intel SGX의 다양한 공격 연구에 대한 최신 동향을 살펴보며, 이에 따른 향후 연구 전망을 제시하고자 한다.

1. 서론

최근 운영체제와 하이퍼바이저 같은 시스템 소프트웨어에 대한 취약점 연구가 지속적으로 진행되고 있다. 이러한 취약점으로 인해 관리자 권한을 탈취당하거나 커널 메모리에 저장된 데이터를 탈취당하는 경우 해당 시스템에서 동작하는 모든 응용 프로그램 데이터의 무결성 및 기밀성이 훼손될 수 있다.

이러한 단점을 보완하기 위해 신뢰할 수 있는 실행 환경을 따로 분리하여 운영체제 같은 시스템 소프트웨어에서부터 다른 응용 프로그램까지 모든 소프트웨어의 접근을 제어하는 기술인 TEE(Trusted Execution Environment) 기술이 소개되었다.[1-5] TEE 기술의 경우 하드웨어 기반의 독립적인 실행 환경을 제공하여 특정한 함수를 이용한 접근을 제외하고는 어떠한 권한의 소프트웨어도 접근할 수 없기 때문에 응용 프로그램이 사용하는 데이터의 무결성과 기밀성을 지킬 수 있다. 이러한 TEE 기술을 기반으로 하는 대표적인 플랫폼으로는 Intel의 SGX(Software Guard Extensions), ARM의 TrustZone, AMD의 PSP(Platform Security Processor) 등이 있다.

하지만 2017년에 CPU 취약점인 Meltdown과 Spectre가 발표되면서 기존 CPU와 동일한 메커니즘을 사용하는 TEE 기반의 플랫폼들 또한 동일한 취약점을 가진다. 그 중 Intel SGX는 Meltdown과 Spectre 공격에 의해 분리된 실행 환경의 비밀 데이터가 공격자에게 노출되는 연구사례가 지속적으로 나오고 있다.[6,7]

본 논문에서는 Meltdown과 Spectre 및 Intel SGX에 대한 취약점 최신 연구 동향을 살펴보고자 한다. 본 논문의 구성은 다음과 같다. 2장에서 Intel SGX와 Meltdown과 Spectre에 대해 기술한다. 3장에서는 Intel SGX에 대한 취약점 연구 최신 동향을 알아보고 4장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 배경 지식

2.1 Intel SGX

Intel SGX는 그림 1과 같이 Enclave라는 격리된 실행 환경을 제공하여 보안성을 유지한다. Enclave를 제외하고 운영체제, VMM(Virtual Machine Monitor)을 포함한 모든 부분을 신뢰할 수 없는 영역으로 분리하기 때문에 운영체제 및 VMM의 관리자 권한을 탈취당하더라도 Enclave에서 동작하는 데이터를 안전하게 지킬 수 있다.

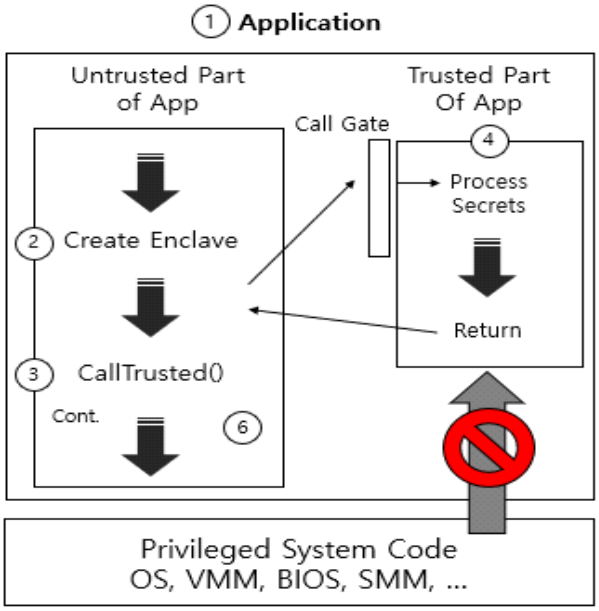
또한 Enclave는 확장성과 범용성을 가지기 위해 기존 시스템의 메모리와 메모리 주소 체계를 그대로 사용한다. 따라서 기존 시스템과 동일한 방식의 가상 주소를 사용하고 빠른 주소 변환을 위해 TLB(Translation Lookaside Buffer) 캐시를 사용한다. 기존 주소 체계의 취약점인 물리 메모리 주소를 이용한 직접 메모리 접근이나 비정상적인 분기문을 이용한 Enclave 메모리 접근을 막기 위해 정상적인 수행 흐름이 아닌 비정상적인 방법으로 Enclave 메모리로 접근하는 경우 시스템 에러를 발생시켜 접근을 차단한다. 그리고 MEE(Memory Encryption Engine) 기술을 이용하여 메모리에 저장된 데이터를 암호화 하여 메모리로 직접 접근하여 데이터를 확인하는 경우에도 데이터의 기밀성을 지킬 수 있다.

독립적인 실행 환경인 Enclave 또한 신뢰할 수 없는 영역에서 데이터를 받아오거나 다른 Enclave와 데이터를 교환해야하는 경우가 발생한다. 이런 경우 Intel SGX는

Enclave가 생성되는 과정에서 측정한 Measurement 값을 이용하여 자기 자신을 증명하는 Attestation 메커니즘을 이용하여 안전한 통신 채널을 구축한 뒤 데이터를 교환한다.

Enclave에서 실행되고 나온 결과물을 Disk에 저장하는 경우 Sealing 메커니즘을 이용하여 암호화를 진행한 뒤 Disk에 저장한다. 따라서 해당 데이터는 그 데이터를 생성한 Enclave에서만 복호화하여 다시 사용할 수 있고 설정에 따라 다른 Enclave에서도 확인할 수 있다.

Intel SGX는 정해진 상황에 따라 다양한 Key를 사용한다. 이러한 Key들은 키 분배 시스템에 의해서 다양한 매개변수의 조합으로 이루어진다. 이러한 매개변수 중 CPU칩이 생산될 때 CPU 내부에 저장되는 비밀 데이터도 사용되기 때문에 생성한 CPU를 제외하고는 아무도 알 수 없는 안전한 Key를 생성할 수 있다.



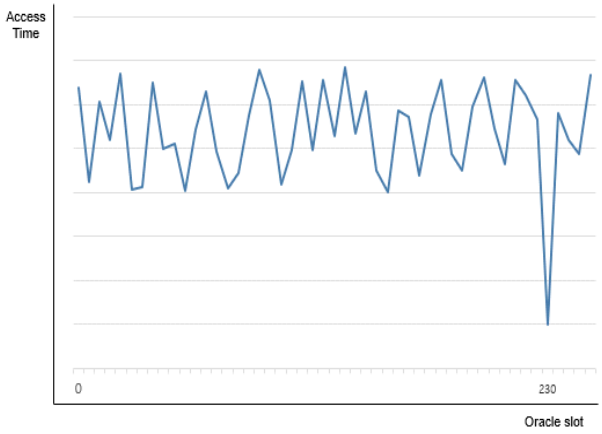
(그림 1) Intel SGX Process

2.2 Meltdown & Spectre

2017년 처음 등장한 Meltdown과 Spectre는 기존 프로세서의 치명적인 보안 결점을 이용하여 프로세서에서 실행 중인 프로그램 데이터를 비정상적으로 캐시 메모리에 저장하고 Cache Timing Attack을 이용해 실행 시간 차이를 이용하여 비밀 데이터가 저장된 위치를 파악할 수 있다. 그림 2 와 같이 캐시 메모리에 있는 모든 데이터를 지운 뒤 특정 데이터만 저장하고 실행 시간을 분석하면 캐시에 저장된 데이터에 접근할 때 실행 시간이 훨씬 빠르다는 점을 이용하여 데이터가 저장된 곳을 유추할 수 있다.

Meltdown은 Intel CPU 최적화 기술 중 하나인 비순차적 명령어 처리 기술의 보안 허점을 이용하여 응용 프로그램의 권한으로는 접근 할 수 없는 커널 메모리에 접근할 수 있다. 따라서 이러한 취약점을 발생시키는 코드가 들어있는 응용프로그램을 실행시켜 운영체제의 시스템 메모리에 접근할 수 있다. Meltdown 공격 과정은 우선 캐시 메모리에 저장된 데이터를 전부 지운 후 Intel CPU의 예측 실행의 특징을 이용하여 접근 할 수 없는 메모리의 데이터를 로드한다. 권한이 없는 메모리에 대한 로드이기 때문에 인터럽트를 발생시켜 실행을 멈추지만 Intel의 예측 실행 기술에 의하여 인터럽트가 발생하여 실행이 멈추기 전에 캐시에 해당 데이터가 저장된다. 따라서 모든 페이지에 접근하여 그 실행 속도를 비교해보면 예측 실행에 의해 캐시에 저장된 데이터에 접근하는 경우 실행 속도가 훨씬 빠르기 때문에 이를 비교하면 해당 값이 저장된 캐시 메모리를 알아낼 수 있고 이를 통해 비밀 데이터를 탈취할 수 있다.

Spectre는 대부분의 CPU 제조사에서 사용하는 최적화 기술인 분기 예측과 같은 추측 실행에서 나오는 취약점을 이용한다. 분기 예측은 if문 같은 조건문을 실행하는 경우 연산 시간을 최적화하기 위해서 조건문의 결과에 따라 달라지는 실행문을 사전에 연산하여 캐시에 저장한다. 따라서 조건문이 참인 경우와 거짓인 경우 수행될 실행문이 모두 캐시에 저장된다. 그리고 조건문이 참인 경우 거짓일 때 실행될 결과는 버려서 데이터의 기밀성을 유지한다. 하지만 Meltdown과 같이 Cache Timing Attack을 이용하여 저장된 값의 위치를 추측하고 해당 캐시에 접근하여 비밀 데이터를 탈취할 수 있다. Spectre의 경우 공격을 위해서 사전에 프로그램 코드에 대해 정확히 이해하고 공격자가 원하는 데이터를 사용하는 취약점 코드를 찾아야하기 때문에 공격이 상당히 까다롭지만 소프트웨어 패치로 100% 막기가 힘들고 대부분의 제조사에서 동일한 취약점을 가진다.

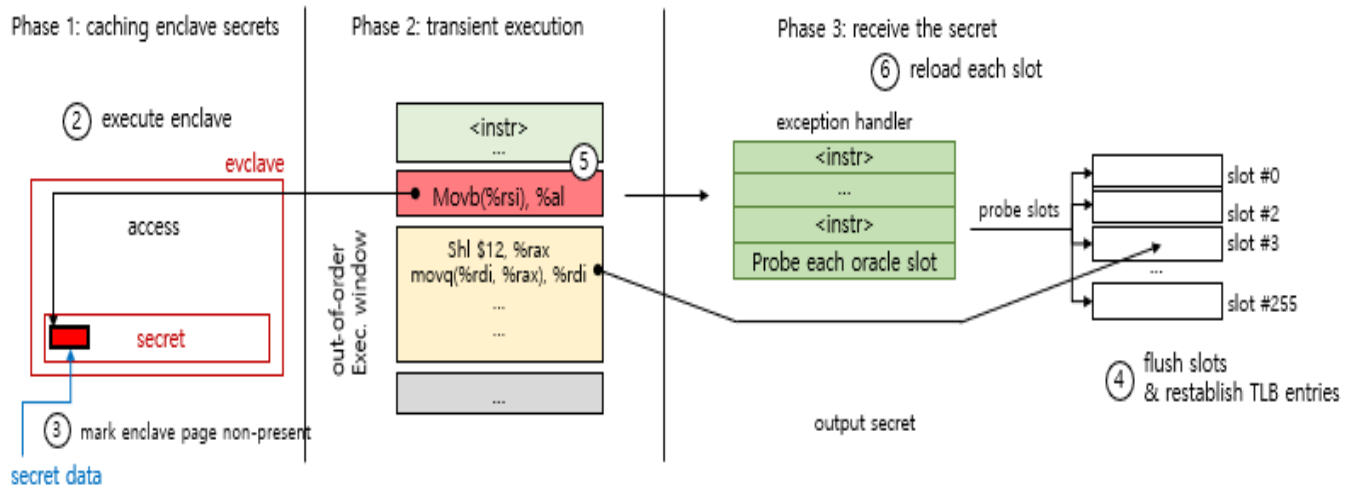


(그림 2) Cache Timing Attack

3. 취약점 연구 동향

3.1 Spectre & Cache Attack

2장 배경지식에서 설명한 Spectre는 기존 프로세서들이 실행 시간 최적화를 위해 분기 예측을 하는 과정에서



(그림 3) Meltdown Attack Process

발생하는 취약점을 이용한 공격이다. Intel SGX 또한 동일한 메커니즘을 사용하는 플랫폼으로서 분기 예측 취약점을 가지고 있다. 따라서 동일한 코드로 분리된 실행 환경인 Enclave가 사용하는 메모리에 저장된 데이터를 스니핑하여 비밀 데이터의 기밀성을 훼손시키는 연구가 진행되었다.

또한 일반적인 프로세서에서 Cache Timing Attack을 이용하여 AES 암호화 알고리즘의 비밀키를 알아내는 공격에 성공하였는데 이와 비슷한 방법을 이용하여 Intel SGX의 Enclave에서 동작하는 AES에 대하여 Prime&Probe 알고리즘을 구현하여 캐시에 저장된 데이터를 실행 속도 차이로 확인하고 Neve&Seifert의 제거(Elimination) 방법을 구현하여 AES의 마지막 라운드의 비밀키를 추출할 수 있다.[8, 9]

3.2 Foreshadow

2018년 발표된 Foreshadow는 기존의 Meltdown과 비슷한 비순차적 명령어 처리 기술의 허점을 이용하여 Intel SGX에서 보호된 메모리에 접근하고 Enclave에서 사용하는 전체적인 암호화 키 값을 추출할 수 있다. 따라서 로컬 및 원격 Attestation 과정을 위조하여 원격 계산에 대한 데이터의 무결성 보장을 완전히 훼손한다. [10]

Foreshadow의 경우 커널 메모리를 공격 대상으로 하는 Meltdown과 다르게 Enclave의 보호된 메모리 영역을 공격 대상으로 하며 시스템 권한이 없더라도 공격할 수 있다. Foreshadow의 전반적인 과정은 그림 3와 같고 L1 캐시에 비밀 데이터를 저장한 뒤 비순차적 명령어 처리로 인하여 Enclave 비밀 데이터를 역참조한다. 이전의 Meltdown과 같은 page fault를 이용한 일시적인 공격과는 다르게 SGX의 경우 승인되지 않은 Enclave 메모리를 역참조하는 경우 Abort Page Semantics이 발생하여 데이터가 모두 -1 값으로 대체 된 후 Enclave가 종료되기 때문에 기존의 공격 방법을 그대로 사용할 수 없다. 또한 Enclave가 시작되거나 종료되는 경우 보안성 유지를 위해 모든 TLB를 flush하기 때문에 실행 중에 Oracle 슬롯에 액세스하면 상당한 시간이 소요되어 비밀 데이터를 전달할 수 없다. 이러한 Abort Page Semantics의 실행을 막기 위해서 Foreshadow는 각 Oracle 슬롯에 대해 4개의 TLB 항목을 설정한 뒤 256개의 Oracle 슬롯을 clflush 명령을 실행하여 프로세서 슬롯에 Oracle 슬롯이 없도록 하여 이 제한을 극복할 수 있다.

따라서 Enclave 메모리에 비정상적으로 접근 했을 때 Abort Page Semantics의 실행이 적용되지 않고 Page fault가 발생하여 예외 처리기가 호출되고 Oracle 버퍼에 비밀 데이터가 저장된 상태로 종료되었기 때문에 Cache Timing Attack을 통하여 CPU 캐시에 저장된 비밀 데이터를 알아낼 수 있다.

3.3 Malware

SGX는 Enclave라는 완벽히 격리된 실행 환경을 제공한다. Enclave가 생성되고 실행되는 순간에는 다른 소프트웨어는 절대 접근할 수 없기 때문에 보안성을 유지할 수 있다. 하지만 이러한 격리된 공간에서 실행되는 프로그램에 Malware 소프트웨어를 숨긴다면 응용 프로그램에 의해 실수로 실행되는 순간 치명적인 보안 결함을 가질 수 있다.

SGX-ROP라고 불리는 이 공격은 Enclave 내에 코드 재사용 공격을 구성한 다음 응용프로그램에 의해 실수로 실행되게 하여 동작한다. 이 기술은 ASLR, Stack Canaries, Address Sanitizer 와 같은 메모리 보안 기술을 우회 할 수 있기 때문에 사용자의 부주의에 의해 실행되는 순간 Enclave 내부에서 Malware가 안전하게 보호됨으로 서비스 거부나 개인 정보가 담긴 파일 도용 등 데이터의 무결성과 기밀성을 훼손당할 수 있다. [11]

3.4 SGX-Bomb

Enclave는 하드웨어 레벨 분리로 다른 소프트웨어의 접근을 차단하고 메모리를 항상 암호화 하여 보안성을 유지한다. 또한 데이터의 무결성 트리를 사용하여 데이터 무결성을 검증하는데 이 과정에서 프로세서가 무결성의 위반을 감지하면 더 이상의 손상을 막기 위하여 자체적으로 시스템을 잠근다. 이런 경우 시스템을 재부팅하여 다시 실행하여 무결성을 유지한다. 이러한 방법은 개인용 컴퓨터에는 올바른 해결책일지 몰라도 서비스 제공자의 입장에서는 심각한 서비스 거부 공격이 발생할 수 있다.

SGX-Bomb 공격의 경우 공격자가 DRAM 뱅크에서 충돌하는 행을 찾은 다음 반복적으로 실행한다. 그리고 Rowhammer 공격을 수행하여 비트 플립을 발생시켜 Enclave 메모리 영역에 대해 읽기 시도를 하여 Enclave 데이터 무결성을 위반한다. 따라서 서비스를 제공하는 서버의 SGX 시스템은 더 이상의 무결성 훼손을 막기 위하여 자체적으로 시스템을 잠그게 되어 심각한 서비스 거부 공격이 발생한다.[12]

SGX-Bomb 공격의 경우 SGX의 보안성의 허점을 이용한 공격으로 283초만에 시스템을 정지시킬 수 있기 때문에 클라우드 시스템 공급자에게 심각한 위협이 될 수 있다.

4. 결론

본 논문에서는 최근 활발한 연구가 진행 중인 TEE기술을 기반으로 한 Intel SGX에 대한 최신 취약점 연구 동향에 대해서 알아보았다. 이러한 취약점들은 대부분 프로세서가 연산을 최적화하기 위하여 사용하는 예측 실행 및 추측 실행에서 나오는 보안 허점을 이용한다. 최근 Intel에서 주기적인 소프트웨어 보안 패치로 대부분의 공격에 대해 대응하고 있고, 소프트웨어 보안 패치로 100% 대응할 수 없기 때문에 취약점을 보완한 하드웨어 개발과 같은 지속적인 연구가 이루어지고 있다.

본 논문에서 알아본 취약점 연구 동향을 토대로 앞으로의 연구 방향은 Intel SGX 및 시스템 소프트웨어에 대한 지속적인 연구를 통해 기존 취약점의 메커니즘을 완벽하게 이해하고 취약점 코드 분석을 통해 공격 패턴을 파악하여 그에 따른 대응 방안을 제시하고 최종적으로 기존의 취약점을 응용하여 새로운 Intel SGX 취약점 발견 및 이에 맞는 대응 방안을 제시하는데 있다.

참고문헌

[1] Payne, Ray, et al. "Integrated security suite architecture and system software/hardware." U.S. Patent Application No. 10/843,180.

[2] Anati, Ittai, et al. "Innovative technology for CPU based attestation and sealing." Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy. Vol. 13. ACM New York, NY, USA, 2013.

[3] Brasser, Ferdinand, et al. "DR. SGX: hardening SGX enclaves against cache attacks with data location randomization." arXiv preprint arXiv:1709.09917 (2017).

[4] Brasser, Ferdinand, et al. "Software grand exposure:{SGX} cache attacks are practical." 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17). 2017.

[5] Costan, Victor, and Srinivas Devadas. "Intel SGX Explained." IACR Cryptology ePrint Archive 2016.086 (2016): 1-118.

[6] Lipp, Moritz, et al. "Meltdown." arXiv preprint arXiv:1801.01207 (2018).

[7] Kocher, Paul, et al. "Spectre attacks: Exploiting speculative execution." arXiv preprint arXiv:1801.01203 (2018).

[8] Neve, M., and Seifert, J.-P. Advances on access-driven cache attacks on aes. In International Workshop on Selected Areas in Cryptography (2006), Springer, pp. 147 - 162.

[9] Götzfried, Johannes, et al. "Cache attacks on Intel SGX." Proceedings of the 10th European Workshop on Systems Security. ACM, 2017.

[10] Van Bulck, Jo, et al. "Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient out-of-order execution." 27th {USENIX} Security Symposium ({USENIX} Security 18). 2018.

[11] Schwarz, Michael, et al. "Malware guard extension: Using SGX to conceal cache attacks." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Cham, 2017.

[12] Jang, Yeongjin, et al. "SGX-Bomb: Locking down the processor via Rowhammer attack." Proceedings of the 2nd Workshop on System Software for Trusted Execution. ACM, 2017.