TEE 기반의 Intel SGX 취약점 연구 동향

김경호* 권용빈* 김현지* 김현준* 서화정*

*한성대학교 IT융합공학과

TEE-based Intel SGX Vulnerability Research Trend

Kyung-Ho Kim* Yong-Bin Kwon* Hyun-Ji Kim*

Hyun-Jun Kim* Hwa-Jeong Seo*

*Division of IT Convergence Engineering, Hansung University.

요 약

Intel SGX(Software Guard Extension)는 하드웨어 기반의 독립된 실행환경을 제공하여 데이터의 무결성 및 기밀성을 유지하는 대표적인 TEE(Trusted Execution Environment) 보안 플랫폼이다. 하지만 SGX의 독립된 실행환경은 CPU 코어, Cache, 메모리 등 프로세서의 실행 환경을 그대로 사용한다. 따라서 기존 프로세서의 취약점을 이용해 Intel SGX에서 동작하는 데이터의 무결성 및 기밀성을 훼손하는 다양한 취약점 연구가 진행 중이다. 본 논문에서는 Intel SGX에 대한 취약점 연구 동향에 대해서 살펴보고, 이에 따른 향후 연구 전망을 제시하고자 한다.

I. 서론

현대의 컴퓨터 시스템은 운영체제와 같은 시스템 소프트웨어를 기반으로 사용자의 편의성을 극대화한다. 이러한 시스템 소프트웨어는 시스템의 전반적인 정보를 관리하기 위해 관리자 권한을 설정하여 시스템을 관리한다. 따라서 공격자가 비정상적인 방법으로 관리자 권한을 빼앗게 된다면 해당 시스템에서 동작하는 프로세스들의 데이터 무결성 및 기밀성이 훼손될 수 있다.

Intel SGX는 기존 시스템의 단점을 극복하기 위해 하드웨어 기반의 독립된 실행환경을 제공한다. 독립된 실행환경에서 동작하는 응용 프로그램의 데이터는 시스템 소프트웨어나 다른 응용 프로그램이 비정상적으로는 절대 접근할 수 없다. 따라서 공격자가 시스템의 관리자 권한을 탈취하더라도 SGX 내부에서 동작하는 데이터는 무결성과 기밀성을 유지할 수 있다.[1,2]

하지만 Intel SGX는 하드웨어적으로 완벽히 분리된 것이 아니라 CPU 코어, Cache, 메모리 등 기존 프로세스의 실행환경을 공유한다. 따라서 Meltdown 및 Spectre, Cache 접근 시간을 이용한 부채널 공격, SGX 내부에서 동작하는 코드에 사용된 취약점 함수를 이용한 ROP (Return-Oriented Programming) 공격 등 SGX의 독립된 실행환경에 기존 프로세서의 취약점을 적용시킨 다양한 연구 사례가 지속적으로 발표되고 있다.

본 논문에서는 Intel SGX에 대한 대표적인 취약점 연구 동향에 대해서 살펴보고자 한다. 본 논문의 구성은 다음과 같다. 2장에서 Intel SGX에 대한 설명과 기존 프로세서 및 프로그램의 취약점에 대해 기술한다. 3장에서는 Intel SGX의 최신 취약점 연구 동향을 알아보고 4장에서는 결론 및 향후 연구 방향에 대해 기술한다.

II. 관련 연구

2.1 Intel SGX

Intel SGX는 대표적인 TEE 기반의 보안 플랫폼으로 Enclave라는 독립된 실행환경을 제공한다. Enclave는 PRM(Processor Reserved Memory)이라는 Enclave 전용 메모리를 따로 분리하여 다른 소프트웨어의 접근을 차단한다. 또한 Enclave가 사용하는 분리된 메모리는 MEE (Memory Encryption Engine) 기술을 이용하여 데이터가 암호화된 뒤 저장되기 때문에 공격자의 직접 물리 메모리 공격에 대한 대응성을 가진다.

또한 Enclave는 확장성을 위해 기존 프로세스의 주소변환 체계를 그대로 사용한다. 따라서 비정상적인 분기문을 이용하여 Enclave에 대한 접근을 할 수 있는데 이런 경우 비정상적인 방법으로 Enclave 메모리에 접근한 경우 Abort page semantic을 발생시켜 모든 값을 -1로 리턴하고 모든 연산을 무시하여 비정상적인 접근을 차단한다.

Intel SGX는 ECALL과 OCALL을 이용하여 Enclave를 동작한다. ECALL 함수는 Enclave 내부에서 동작하는 함수를 호출하는 것으로 ECALL 함수 호출된 이후로는 Enclave에서 동작하기 때문에 안전한 실행환경을 보장한다. 그리고 OCALL 함수는 Enclave 내부에서 동작하는 함수에서 신뢰할 수 없는 영역의 함수를 호출하는 것으로 신뢰할 수 있는 영역과 신뢰할 수 없는 영역 사이에 안전한 방식으로 동작할 수 있도록 지원한다.

또한 Intel SGX는 Key Derivation Material을 이용해 다양한 종류의 암호화 키를 제공한다.[1] Intel SGX에서 사용되는 다양한 종류의 키 값은 Enclave 생성 중에 생성되는 매개 변수의 값에 따라 결정되기 때문에 공격자가 키 값을 알아내는 것이 불가능하다.

2.2 Meltdown & Spectre

2018년에 새롭게 발표된 취약점인 Meltdown과 Spectre는 프로세서의 연산시간 단축을 위한 최적화 기법의 취약점을 이용한 공격이다.[3,4]

Meltdown은 Intel CPU에서 발생된 취약점으로 최적화 기술 중 하나인 비순차 실행으로 인해 발생하는 취약점을 이용한 공격이다. 최신 프로세서는 다수의 연산을 한 번에 처리할 수 있기 때문에 서로 의존성이 없는 연산은 순서에 상관없이 실행한다. 이 과정에서 사전에 연산된 결과를 Cache에 저장하는데 이러한 특징을 이용하여 순차적인 실행에서는 실행되지 않을 비밀 데이터 접근 명령이 비순차 실행으로 인하여 Cache에 저장된다. 그리고 Cache Timing Attack을 통해 Cache에 저장된 비밀 정보를 탈취한다.

Spectre는 대부분의 프로세서가 사용하는 예측 실행으로 인해 발생하는 취약점을 이용한 공격이다. 대표적인 예측 실행으로 분기 예측이 있는데 특정 조건문의 결과가 지속적으로 동일하면 해당 명령을 사전에 연산한 뒤 Cache에 저장한다. 이 때 조건문 결과가 다르면 Cache에 저장된 데이터는 Flush하지만 Cache에 로드되고 Flush되는 사이에 Cache Timing Attack을 통해 비밀 정보를 탈취한다.[5]

2.3 ROP Attack

프로세스가 실행되면 운영체제는 가상 메모리를 할당하여 명령어 및 라이브러리 함수를 할당한다. 호출된 함수는 실행된 이후에 이전에 호출한 함수로 돌아가야 하기 때문에 스택 메모리에 이전 함수의 주소를 저장한다. 따라서 모든 함수에는 실행이 끝난 후 스택 메모리에 저장된 주소로 분기하는 명령어가 포함된다.

ROP Attack은 이러한 특징을 이용하여 취약점이 존재하는 라이브러리 함수를 사용한 경우 Return 주소를 공격자가 임의로 변경하여 원하는 연산을 수행할 수 있다.

ROP Attack을 막기 위해 ASLR(Address Space Layout Randomization), DEP(Data Execution Prevention) 등 대응 기법이 존재하지만 PLT(Procedure Linkage Table) 및 GOT(Global Offset Table)를 이용한 우회 기법 또한 존재한다. 이러한 공격을 막기 위한 방법은 취약점이 존재하는 함수를 사용을 자제하는 것이다.[6]

III. 취약점 동향

3.1 Foreshadow

2018년에 발표된 Foreshadow Attack은 Intel CPU의 취약점인 Meltdown을 Intel SGX에 적용한 공격이다. Enclave는 어떠한 권한의 소프트웨어도 접근 할 수 없어야 하지만 비순차 실행을 이용하여 비정상적인 접근을 통해 Cache에 비밀 데이터 저장이 가능하다.[7]

Foreshadow의 공격 과정은 Meltdown과 동일하다. 우선 Enclave의 비밀 데이터에 접근하는 명령어를 이용한다. 정상적인 실행 흐름이라면 Abort page semantic이 발생하여 접근을 차단하지만 비순차 실행에 의하여 모두 비워진 L1 Cache에 비밀 데이터를 저장한 뒤 데이터 접근 속도 차이를 이용한 Cache Timing Attack인 Flush + Reload 방식을 이용하여 비밀 데이터를 탈취한다.

기존의 Meltdown과 같이 관리자 권한이 없어도 공격이 가능하지만 관리자 권한을 이용하면 데이터를 L1 Cache에 정확하게 저장할 수 있고 공격의 성공률을 높일 수 있다.

Intel은 Foreshadow 공격에 대응하기 위해 소프트웨어 보안 패치 및 하드웨어 마이크로코드 수정을 진행하였고 Intel TSX(Transactio-nal Synchronization Extensions)을 이용한 예외 탐지 연구 또한 활발히 진행되고 있다.

3.2 Cache Attack

Intel SGX는 부채널 공격에 내성을 가지지 않는다. 또한 일반적인 프로그램과 똑같이 Cache 메모리를 사용하기 때문에 Enclave 내부에서 동작하는 응용 프로그램이 Cache를 사용하는 경우 접근 속도 차이를 이용한 부채널 공격 연구 사례가 발표되었다.[8,9]

대부분의 Cache Attack은 동일한 코어에서 하이퍼쓰레딩을 이용하여 공격자 쓰레드와 희생자 쓰레드를 동시에 실행시킨다. 하이퍼쓰레딩의 경우 각 쓰레드가 L1 Cache를 공유하게 되는데 이 때 Intel PMC(Performance Monitori-ng Counter)를 이용하여 Cache Hit, Miss를 판단하여 Cache에 저장된 데이터를 탈취 할 수 있다.

대표적인 Cache Attack으로 희생자 쓰레드에서 동작하는 T-Table을 이용한 AES 암호화 알고리즘을 마지막 라운드에서 PMC를 이용하여 Prime + Probe 방식의 Cache Timing Attack을 이용하여 T-Table을 알아내고 Neve&Seifert의 제거(Elimination) 방법을 구현하여 AES의 마지막 라운드의 비밀 키를 추출할 수 있다.

3.2 ROP Attack

보편적인 해킹 기법인 ROP 공격은 동일한 방식의 취약점으로 Enclave 내부에서도 실행될 수 있다. 우선 공격자가 Enclave 내부의 코드에 대해서 역공학(Reversing)을 통해 취약점이 존재하는 코드를 알고 있다는 전제 하에 공격을 진행한다.[10]

Enclave에서 동작하는 응용프로그램의 경우 공격자가 정적 분석을 통해서 Gadget을 얻을 수 없기 때문에 Enclave의 AEX(Asynchronou-s Enclave Exit)의 특징을 이용하여 찾아낸 Gadget의 POP 명령어 수를 알아낸다. 그 이후 찾아낸 Gadget들 중 공격에 필요한 Gadget를 찾기 위해 Enclave에서 사용하는 명령어인 ENCLU 명령어를 사용하여 Leaf Function으로 지정된 함수들 중 EEXIT 함수 실행을 확인하여 공격에 필요한 Gadget을 찾아낸 후 Gadget Chaining을 이용하여 공격을 수행한다.

이러한 ROP Attack을 막기 위해 실행 시 Enclave 메모리를 무작위화하는 대응방안인 SGX-shield와 같은 다양한 연구가 진행 중이다.

3.3 SGX-Bomb

대부분의 취약점은 기존의 운영체제 및 시스템에 존재하는 취약점을 Intel SGX에 적용시켜서 공격한다. 하지만 SGX-Bomb의 경우 하드웨어적 결함인 Rowhammer Attack을 이용하여 Enclave의 무결성 검사를 실패하게 만들어 서비스 거부 공격(Denial-of-Service At-tack)을 수행한다.[11]

SGX-Bomb은 DRAM의 Cell 밀집도가 높아짐에 따라 동일한 메모리 뱅크에 반복적으로 접근하여 하드웨어적 결함으로 인한 Bit flip을

발생시키는 Rowhammer Attack을 수행하여 Enclave 내부의 메모리의 값의 변동을 일으킨다. Enclave는 비정상적인 데이터의 변화를 막기 위해 주기적으로 무결성 검사를 수행하는데 무결성 검사에 실패하게 되면 소프트웨어 방어를 목적으로 프로세스를 중지한다. 그 결과 Intel SGX상에서 원격 서비스를 제공하는 서버의 경우 프로세스가 중지되는 서비스 거부 공격을 당하게 된다.

IV. 결론

본 논문에서는 TEE 기반의 Intel SGX에 대한 최신 취약점 연구 동향에 대해서 알아보았다. Intel SGX에서 발생하는 대부분의 취약점은 기존 시스템 취약점을 적용시킨 공격이 대부분이다. Intel은 이러한 취약점을 막기 위해 소프트웨어 보안 패치 및 마이크로코드 업데이트를 진행하고 있다. 하지만 여전히 Cache를 이용한 부채널 공격에 대한 대응 방안을 제공하지 않기 때문에 사용자가 부채널 공격에 강인한 응용 프로그램을 제작해야 한다.

본 논문에서 알아본 취약점 연구 동향을 토대로 앞으로의 연구 방향은 Intel SGX 상에서 부채널 공격에 안전한 암호 구현 및 새로운 취약점 발견과 그에 맞는 대응 방안을 제시하는데 있다.

[참고문헌]

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

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

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

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

[5] Gruss, Daniel, et al. "Flush+ Flush: a fast and stealthy cache attack." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Cham, 2016.

[6] Hund, Ralf, Carsten Willems, and Thorsten Holz. "Practical timing side channel attacks against kernel space ASLR." 2013 IEEE Symposium on Security and Privacy. IEEE, 2013.

[7] 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.

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

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

[10] Lee, Jaehyuk, et al. "Hacking in darkness: Return-oriented programming against secure enclaves." 26th {USENIX} Security Symposium ({USENIX} Security 17). 2017.

[11] 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.