==== 김치콘 2017 BEST CFP #01 ====

submitted at 2017. 6. 3 오후 3:19:39

제목	**Hacking in Darkness: Return-oriented Programming against Secure Enclaves**	소요 시간	20~30분
키워드	Trusted Execution Environment, Return Oriented Programming, Intel SGX		

[초록]

Intel SGX는 하드웨어 기반의 Trusted Execution Environment (TEE)를 제공하여 기존의 보안위협들에 대한 효과적인 방어책을 제공하고 있다. Intel SGX는 소프트웨어 취약점이 존재하지 않는한, 프로그램 (Enclave)에 대한 강력한 무결성과 기밀성을 제공한다.

하지만 이전의 경험들을 돌이켜 보았을때, 소프트웨어에 취약점이 존재하지 않는 상황을 가정하기란 매우 어렵다고 할 수 있다. 지금까지 Intel SGX에 대한 공격이 side-channel에 집중되어 왔지만, 우리는 Intel SGX를 사용하는 프로그램에 소프트웨어 취약점이 존재하는 상황하에 이를 어떻게 공격을 할 수 있으며 어떤 효과를 얻을 수 있는지에 대한 물음에 대답하고자 한다.

우리는 Enclave에 대한 실용적인 공격 방식인 Dark-ROP를 제안하며, 이를 통하여서 Intel SGX가 Enclave의 코드와 데이터에 제공하는 기밀성과 무결성을 완벽하게 우회할 수 있는 방식을 제안한다. Dark-ROP는 Enclave 코드와 데이터에 대한 정보 없이도, 공격에 필요한 ROP 가젯들을 획득할 수 있는 방식을 제안하며, 획득한 가젯들을 통하여서 Intel SGX가 제공하는 데이터 실링과 리모트 어테스테이션을 우회하는 방법을 제안한다.

[개요]

먼저 Intel SGX에 대한 background를 제공함으로서, Intel SGX가 보장하는 Security guarantee에 대해 설명한다 (Integrity and confidentiality guarantees on Enclave). 또한 Intel SGX가 제공하는 기능인 Sealing과 Remote attestation에 대한 이해를 제공하며, 이러한 기능들이 Intel SGX가 제공하는 특별한 instruction인 ENCLU에 의해 제공됨을 설명한다.

다음으로 Intel SGX가 가정하고 있는 threat model에 대해 설명하고, Dark-ROP가 취하는 공격자의 capabilities에 대해 설명한다. Dark-ROP 에서 가정하고 있는 공격자 모델은 Intel SGX에서 정의하고 있는 공격자 모델과 전혀 다르지 않다.

우리는 Enclave의 코드와 데이터에 대한 정보가 전혀 없으며, 단지 취약점을 Fuzzing등을 통해 찾아내는 상황을 가정하고 있다. 이 상황에서 단순한 stack buffer overflow취약점을 통해, 어떻게 필요한 가젯들을 찾아낼 수 있는지에 대한 technique을 3단계에 걸쳐 설명한다.

첫번째 단계는 pop x, ret / pop x, pop y, ret 등의 pop something ret 형식의 가젯을 찾는다. 이때 side-channel 공격을 활용하여서, 쉽게 pop, ret 가젯과, pop의 갯수를 알아낼 수 있다. 두번째 단계에서는 ENCLU gadget을 찾는다. ENCLU gadget은 3번째 단계에서 활용할 EEXIT 가젯으로

사용할 수 있으며, Remote attestation과 sealing을 bypass하는데 사용된다. ENCLU gadget을 찾아내는 방법은 앞서 찾은 모든 가젯들을 chaining하여서, EEXIT이 invoke 될 수 있도록 유도하여 찾아낸다 (RAX = 0x4). ENCLU를 찾아낸 후, EEXIT을 활용하여 pop x, ret 가젯의 register들을 구분해 낸다. 즉 pop x, pop y, ret에서 x 와 y의 register가 무엇인지 찾아낸다. 세번째로는 memcpy 가젯을 찾는다. memcpy를 찾기 위해서 rsi, rdi, rcx register를 찾아낸 가젯으로 잘 셋팅해 주고, enclave program이 특정 데이터를 untrusted application 쪽으로 copy하도록 하여 값이 쓰여지는지 확인한다.
 이렇게 찾아낸 가젯들을 사용해서 Sealed data를 unseal할 수 있으며 (EGETKEY 로 sealing key를 찾아냄), Remote attestation을 bypass할 수 있다 (EREPORT instruction 사용). 이를 통해 Intel SGX가 제공하는 무결성과 기밀성을 모두 우회할 수 있다.

[발표 목적]

Intel SGX에 대한 background, threat model.
Intel SGX가 제공하는 무결성과 기밀성에 관해.
Intel SGX가 제공하는 기능인 sealing과 attestation에 대한 이해.
Intel SGX에 취약점이 있다면, 이를 어떻게 공격에 사용할 것인가?

[고려사항]

2017년 Usenix Security에 발표될 논문 입니다.

[투고 이유]

Intel SGX에 대해 알리고, 이를 사용할때 고려해야할 점을 알리고자 함.

이전에 다른 컨퍼런스에서 발표된 내용입니까? 예
새로운 도구에 대한 내용입니까? 아니오
새로운 취약점에 대한 내용입니까? 예
시연을 포함하고 있습니까? 예
기타 사항: