SGX 취약점 동향

한성대학교

김경호, 권혁동, 김현준, 서화정

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

1. What is Intel SGX | TEE

- TEE (Trusted Execution Environment)
  - 신뢰 할 수 있는 실행 환경
  - 메인 프로세서 내부의 보안 영역으로 격리된 환경에서 운영체제와 병렬 실행
  - 격리된 환경을 통한 응용프로그램의 무결성 및 기밀성 제공
  - H/W, S/W 측면에서 보안성 극대화
  - 신뢰 공간과 비신뢰 공간 (Secure , Normal)

  - Intel SGX(Software Guard eXtensions)
  - ARM TrustZone
  - AMD PSP(Platform Security Processor)

1. What is Intel SGX | Intel SGX

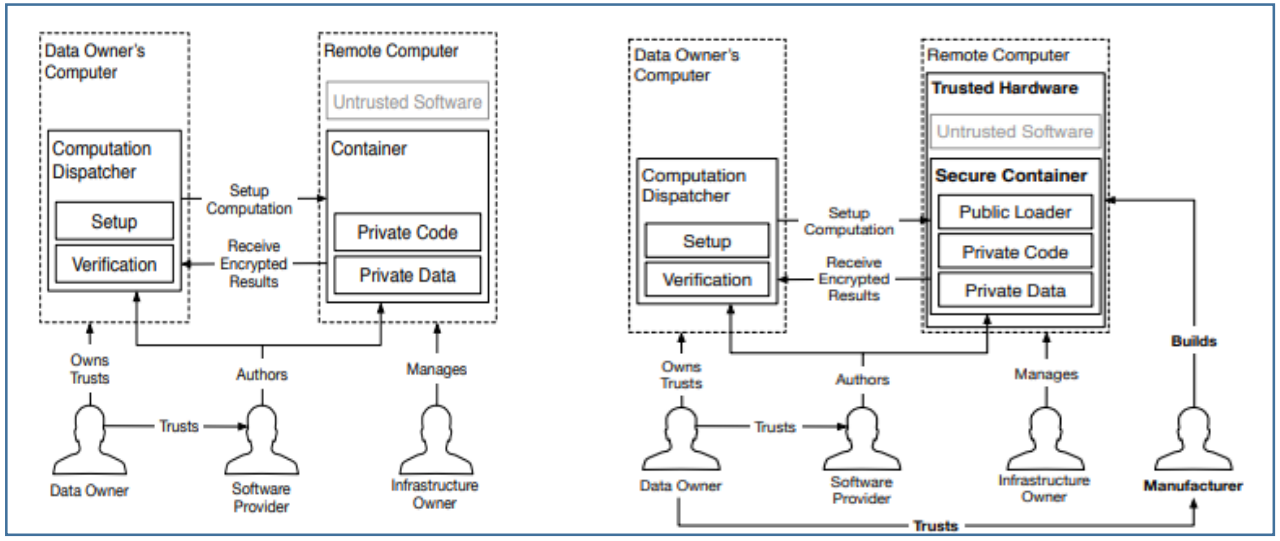
- Intel SGX

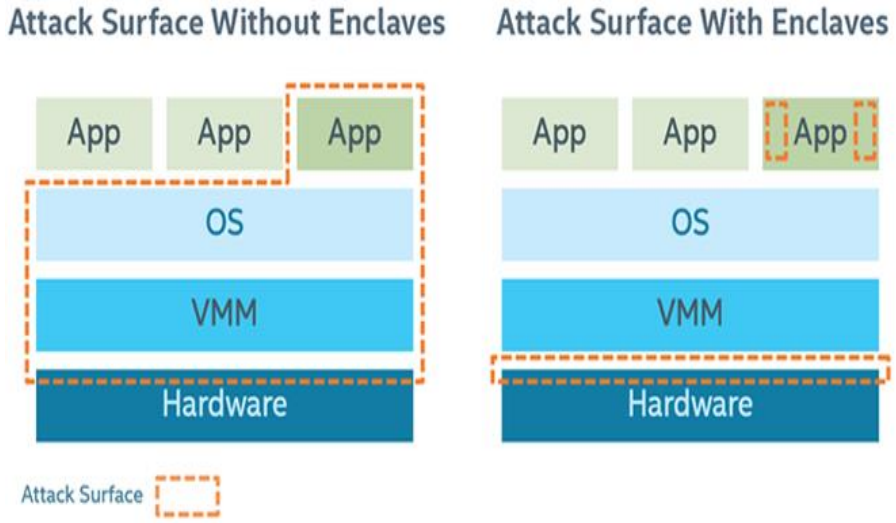
Intel에서 제공하는 CPU 명령어 코드

Enclave라는 Secure Container 제공

운영체제, 하이퍼바이저 포함 어떤 수준의 권한으로도 접근이 불가능

Enclave를 사용함으로써 공격 범위를 효과적으로 경감





1. What is Intel SGX | H/W Security

- MEE(Memory Encryption Engine) 을 이용하여 메모리 암호화

  (물리적 메모리 공격 보호)

- Enclave 실행 후 Disk 저장 파일 암호화

  (디스크에 저장된 중요 데이터 보호)

- 다양한 종류의 암호화 키 제공

- 부채널 공격을 방어하는 메커니즘은 존재하지 않음

  (개발자가 부채널 내성 프로그래밍을 해야함)

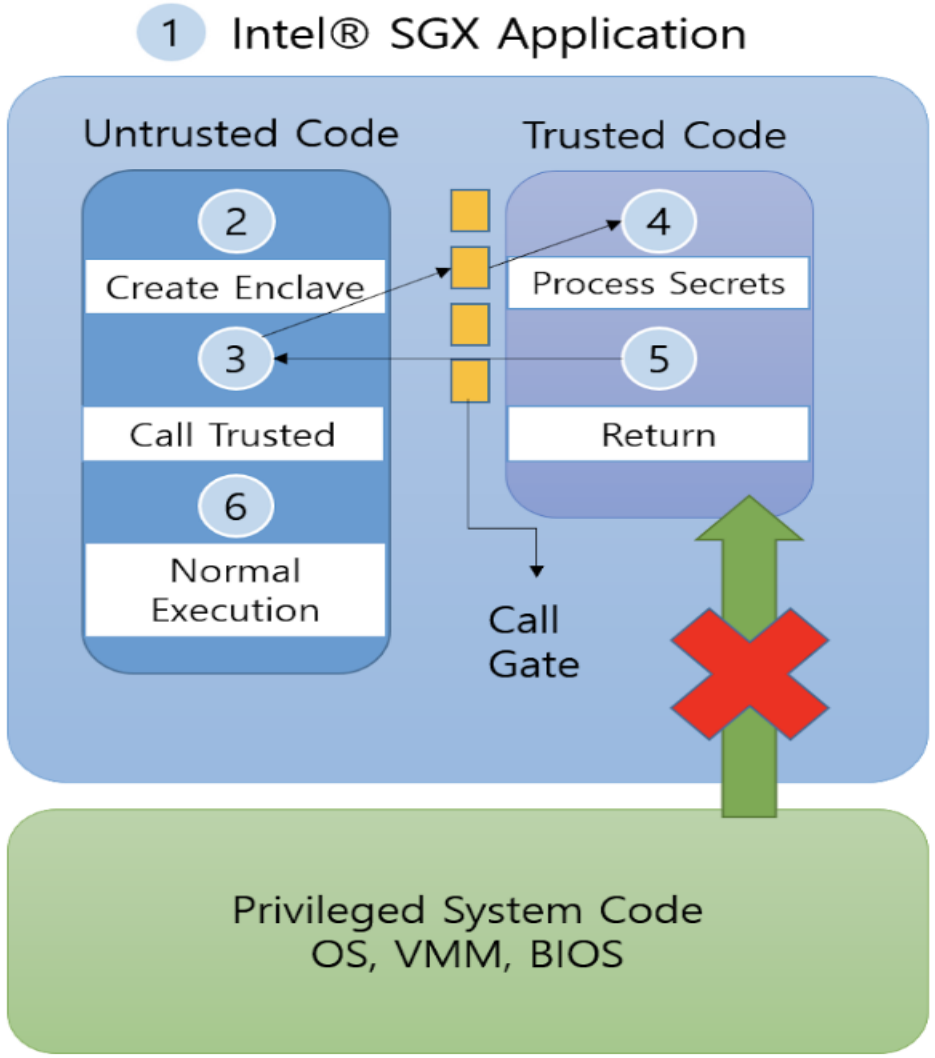
1. What is Intel SGX | S/W Security

- BIOS, VMM, OS 등의 높은 권한의 System S/W도 접근 불가능
  (System S/W 해킹시에도 저장된 데이터 무결성 및 기밀성 보장)

- jump, function call 등의 분기문으로 Enclave 메모리 접근 불가
  (StackOverflow 같은 비정상적 분기를 이용한 메모리 접근 방어)

- 리버싱을 방어하는 메커니즘은 존재하지 않음
  (Source Code 작성시 하드코딩 및 취약점 존재하는 코딩 자제)

1. What is Intel SGX | Mechanism

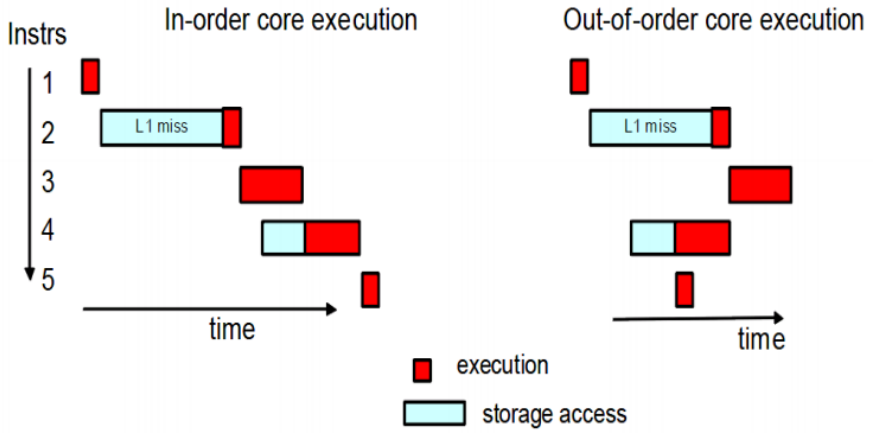


1. Trusted zone, Untrusted zone 설정
2. Enclave 생성 (Trusted Zone)
3. Enclave 실행 (ECALLS)
4. 프로세스 수행 (외부에서 접근 불가능)
5. Return으로 반환값 반환 (OCALLS)
   (Enclave 데이터는 여전히 Trusted memory에 저장)
6. 기존 연산 수행

2. Background

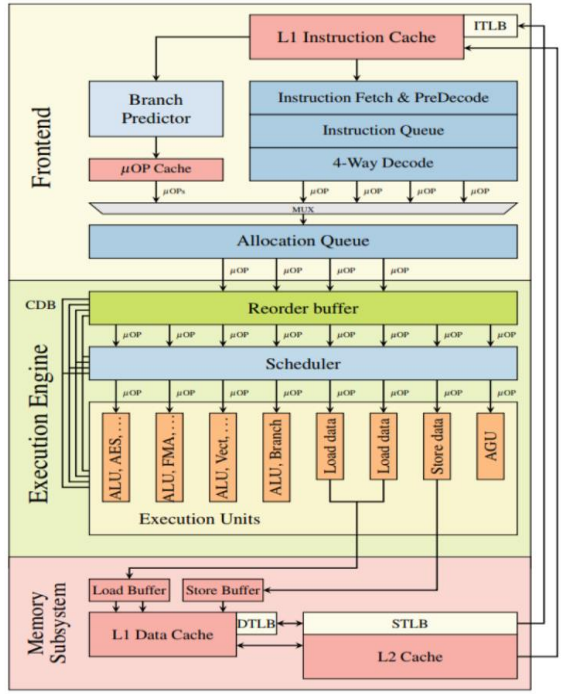
- 예측 실행 (Speculative Execution)
  - If 문이나 분기문을 사전에 예측하여 미리 연산하여 캐시에 저장
  - 분기 예측이 틀린 경우 폐기됨 (프로그램에서 확인 불가능)

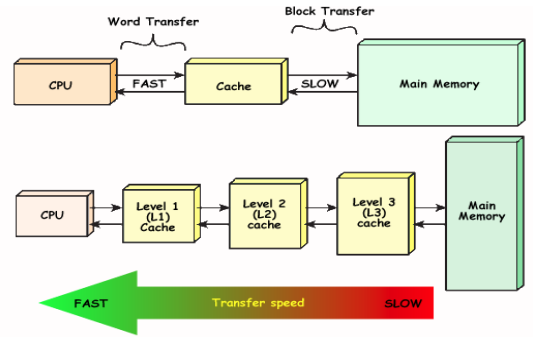
- 비순차 실행 (Out of Order Execution)
  - 명령어 특성상 연산 속도가 다르기 때문에 연산 최적화를 위해 순서를 바꿈
  - 따라서 뒤에 사용될 명령어가 미리 실행될 수 있음



2. Background | CPU Cache

- Memory Architecture
  - 메모리의 종류에 따라 액세스 속도차이가 존재
  - 액세스 속도차이로 인한 오버헤드를 막기 위해 계층이 존재
  - CPU 코어에 가까울수록 연산속도 빠르고 크기가 작음
  - Cache Hit와 Cache Miss를 이용하여 Cache Update
  - Cache Miss인 경우 메모리에서 캐시로 값을 Load
  - Cache Hit 확률을 높이기 위한 다양한 방법론이 존재

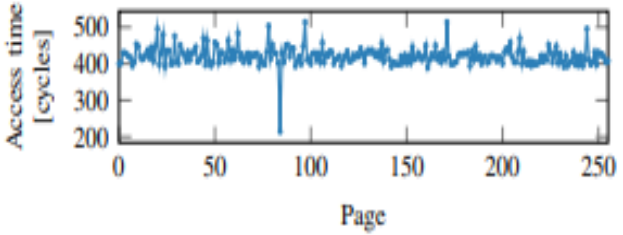




2. Background | Cache Timing Attack

- Cache Timing Attack

  - Cache Hit와 Miss의 액세스 속도 차이를 이용한 부채널 공격
  - SGX는 부채널 공격에 대한 내성이 없기 때문에 대부분의 취약점이 Cache Timing Attack 에서 나옴
  - 대표적인 공격 방법으로 Flush + Reload, Prime + Probe



!! Flush + Reload

공격 대상 메모리 준비 후 모든 캐시 삭제 (Flush) -> 무조건 Cache Miss 발생

비정상적인 방법으로 캐시에 비밀 데이터 적재

메모리를 읽어서 액세스 시간 측정 후 다른 메모리에 비해 액세스 속도가 빠른 곳을 찾아냄 (Reload)

!! Prime + Probe

공격 대상 메모리 준비 후 액세스 하여 캐시 적재 (Prime)

비정상적인 방법으로 캐시에 비밀 데이터 적재 -> 기존의 정보가 탈락됨

메모리를 읽어서 액세스 시간 측정 후 다른 메모리에 비해 액세스 속도가 느린 곳(탈락)을 찾아냄 (Probe)

2. Background | Spectre

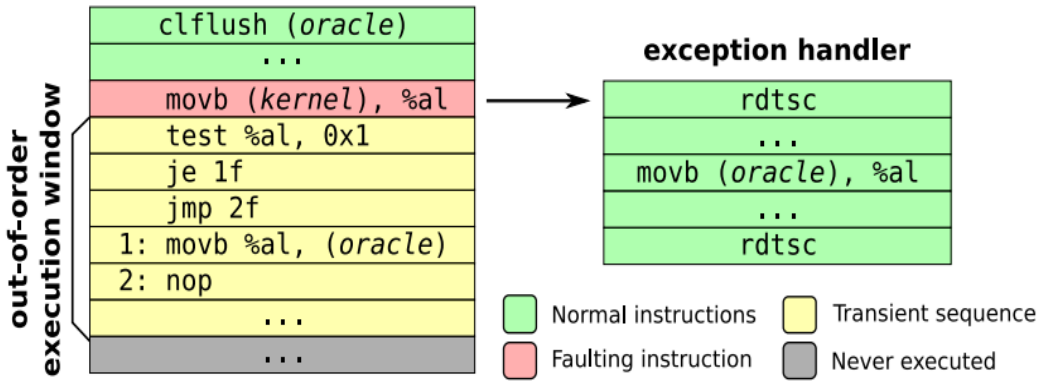
- Spectre

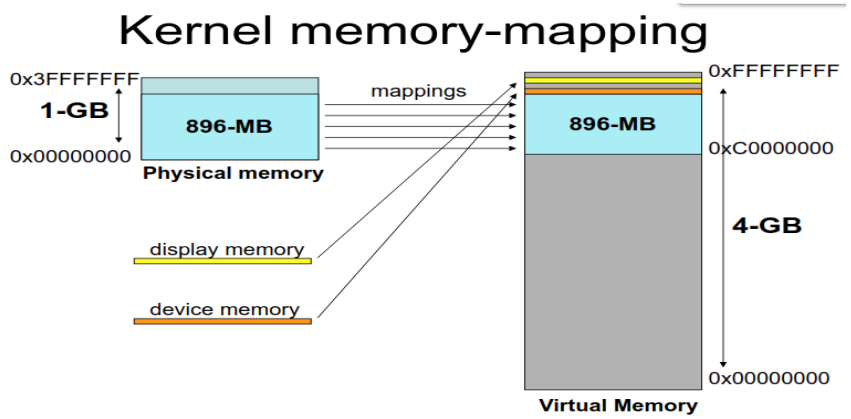
  - 2017년에 발표된 CPU 취약점
  - **예측 실행(Speculative Execution)**의 취약점을 이용함
  - if 문을 이용한 예측 실행과 분기 예측을 이용한 취약점이 존재
  - 예측 실행을 통하여 접근 불가능한 데이터를 캐시에 저장
  - 캐시에 저장된 데이터가 폐기되기 전에 Cache Timing Attack -> 데이터 유출
  - Intel, ARM, AMD 대부분 제조사의 CPU에 동일한 취약점 존재

```
if (x < array1_size)
    y = array2[array1[x] * 4096];
```

2. Background | Meltdown

- Meltdown
  - 2017년에 발표된 Intel CPU 취약점
  - **비순차 실행(Out of Order Execution)**의 취약점을 이용함
  - 비순차적인 명령어를 통하여 접근 불가능한 데이터를 캐시에 저장
  - 캐시에 저장된 데이터가 폐기되기 전 Cache Timing Attack으로 데이터 유출
  - SGX 또한 Intel CPU에서 동작함으로 동일한 취약점을 가짐
  - 커널 메모리 공간을 접근할 수 있어서 커널에서 돌아가는 모든 프로그램의 정보가 다 유출될 수 있음





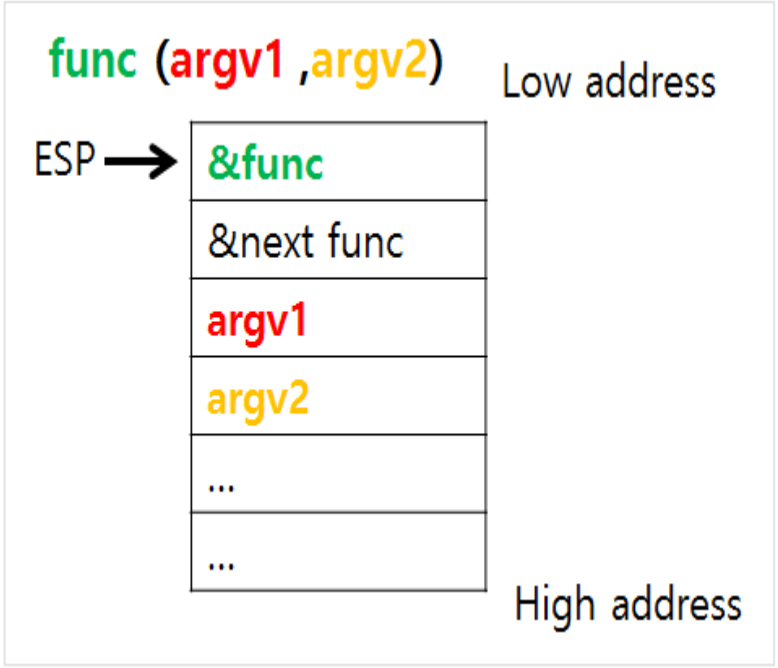
2. Background | ROP Attack

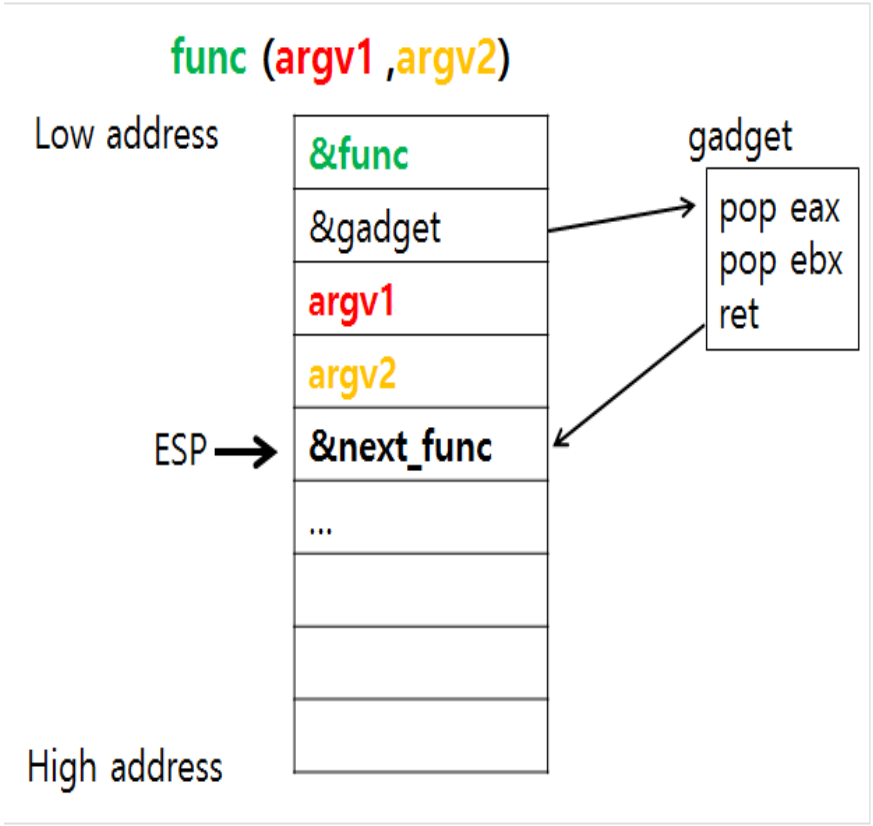
- ROP(Return Oriented Programming) 란?
  - strcpy, scanf와 같은 취약점 있는 함수를 이용한 Buffer Overflow 취약점이 있는 코드를 Gadget을 이용하여 함수 호출 및 조작하는 공격
  - 메모리 취약점을 막기 위한 방어 기법인 ASLR, DEP 등도 우회 가능

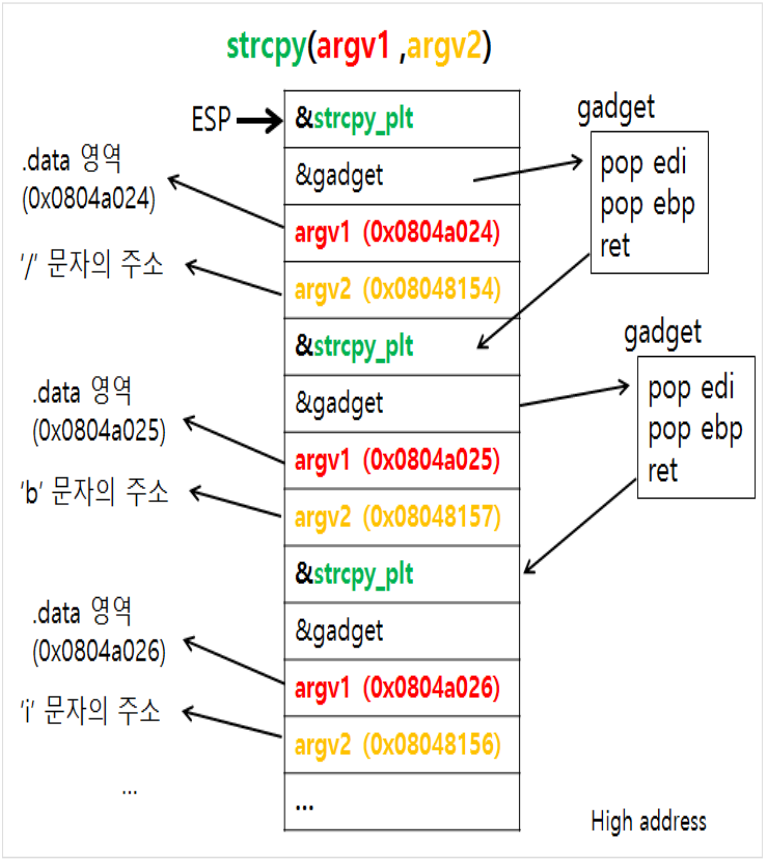
- Gadget
  - 공격을 하기 위한 코드 조각
- ASLR(Address Space Layout Randomization)
  - 실행 및 호출 할 때 마다 주소 배치를 무작위로 배정하는 기법
- DEP/NX(Data Execution Protection/Non Executable)
  - 코드 영역 제외한 다른 영역에서 실행 권한 x

2. Background | ROP Attack

- 함수 호출 과정에서 Stack Memory를 이용한 Parameter 전달
- 함수 끝나면 이전 실행했던 Code로 Jump







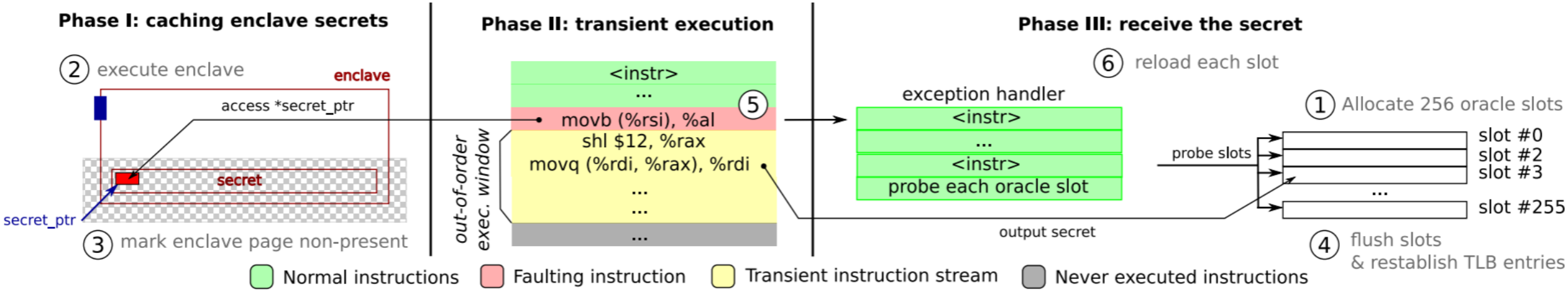
3. Intel SGX Vulnerability Research Trends

- Spectre, Meltdown
- Cache Attack
- ROP Attack
- SGX Security Feature

Foreshadow

- **CVE-2018-3615**
- **USENIX'18**
- **2018년도 1월 발견 → 2018년도 8월 공개**
- **Foreshadow (L1 Terminal Fault)**
  - Meltdown을 SGX에 적용한 결과

Foreshadow



- **Assumption**
  - Meltdown 공격을 바로 적용해서는 정보 유출 불가능
  - enclave의 비밀 정보가 L1 캐싱되어있어야 함
- **Translation Lookaside Buffer (TLB)**

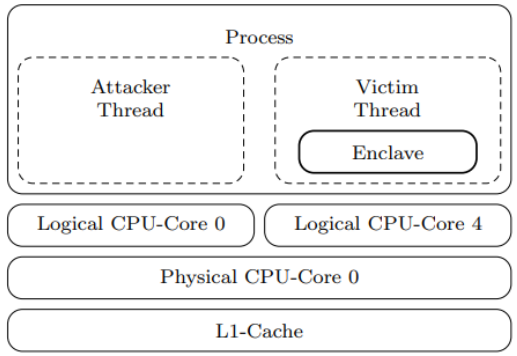
Transient execution

```
1 foreshadow:
2   # %rdi: oracle
3   # %rsi: secret_ptr
4
5   movb (%rsi), %al
6   shl $12, %rax
7   movq (%rdi, %rax), %rdi
8   retq
```

```
1 void foreshadow(
2     uint8_t *oracle,
3     uint8_t *secret_ptr)
4 {
5   uint8_t v = *secret_ptr;
6   v = v * 0x1000;
7   uint64_t o = oracle[v];
8 }
```

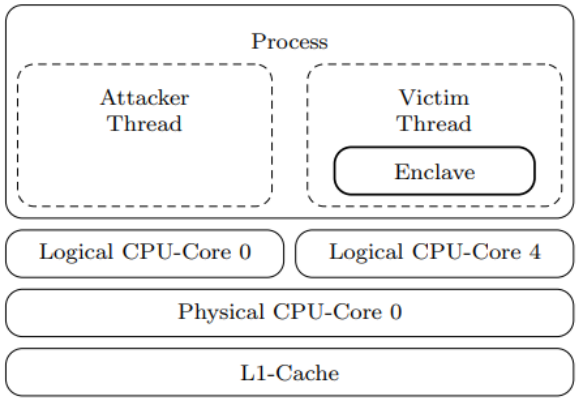
Cache Attack

- Enclave에서 작동하는 코드에 캐시 타이밍 공격을 제시
- Intel PMC(Performance Monitoring Counter)를 이용한 캐시 히트 검사
- PMC, CPU 고정, 쓰레드의 hyperthreading affinity 등 Enclave 접근을 제외한 모든 System 권한 필요
- Victim Thread, Attacker Thread가 동일 코어에 Hyperthreading을 통해 같은 L1-cache를 공유 해야함



Cache Attack (AES)

- EuroSec'17
- AES에 대한 타이밍 공격 (L1 Cache 대상)
- Priming & Probing을 이용한 Enclave 데이터 확인 및 Neve & Seifert's elimination method를 이용한 AES Key 추출
- Attacker Thread와 Victim Thread를 단일 코어에서 병렬로 실행되는 Hyperthreading을 이용
- AES 마지막 라운드에서 공격자 쓰레드에서 PMC를 사용한 Cache Probing
- Attacker Thread와 Victim Thread의 동기화 필요



Hacking in Darkness

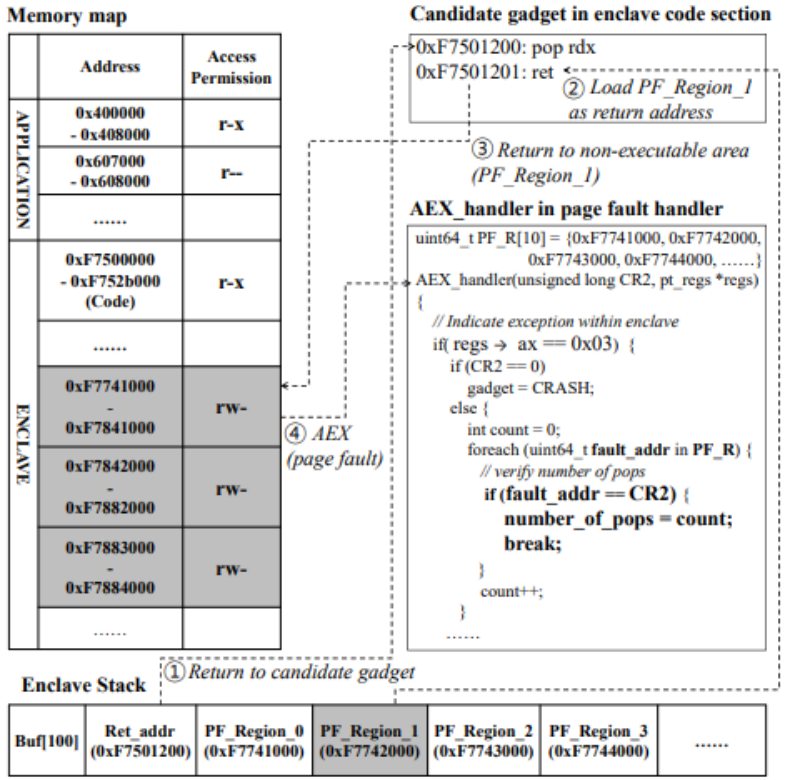
- USENIX Security’ 17
- ROP를 이용한 메모리 손상 취약점 이용
- Enclave의 상태를 3개의 Oracle을 이용해 전송
- ROP를 이용한 Malware 실행
- Malware를 활용한 MITM(Man in the Middle) Attack 으로 Remote Attestation 조작 가능

Hacking in Darkness (Assumption)

- 코드에 ENCLU 명령이 있어야함
  - Gadget을 찾을 때 사용
- 코드에 ROP gadget 이 있어야함 (pop register)
  - 레지스터를 이용해 Parameter 전달
- 코드에 취약점 함수가 있어야함 (memcpy, strcpy, etc..)
- Enclave 내부 접근을 제외한 System 전체 권한이 있어야함
- Intel의 표준 SDK로 컴파일 해야함

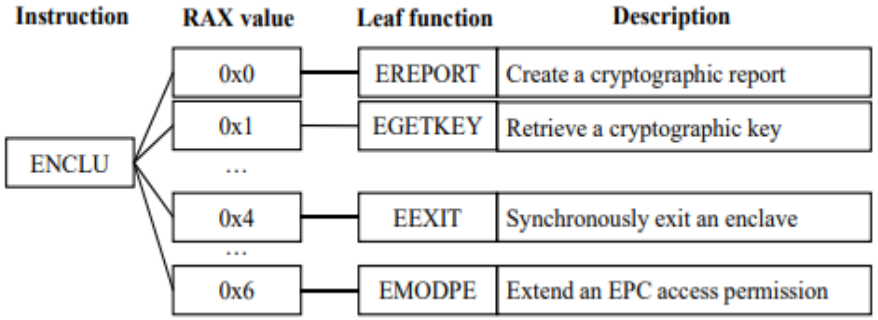
Hacking in Darkness (Oracle 1)

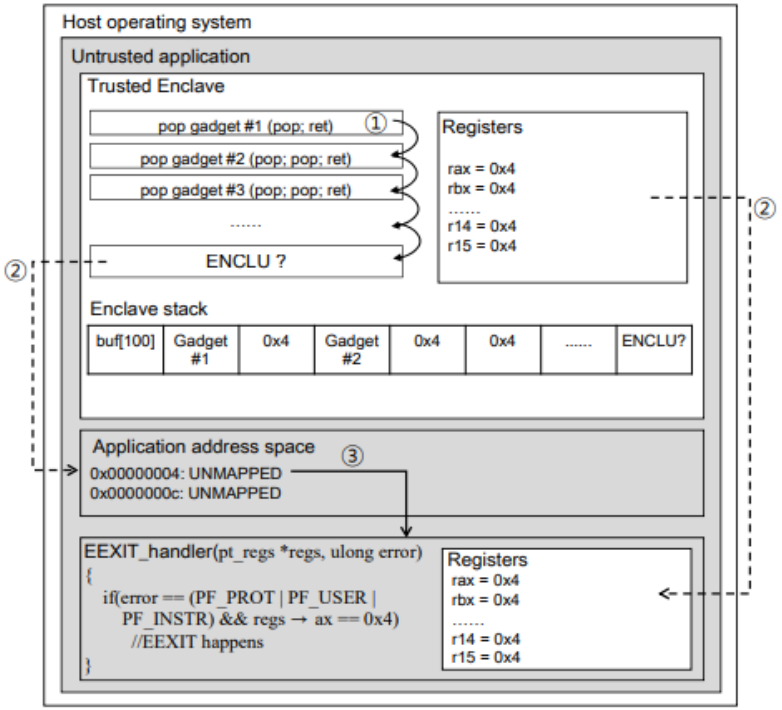
- 레지스터를 이용하는 Gadget 찾기
- AEX 및 Page fault를 이용
  - 예외 발생시 cr2레지스터를 이용하여 pop의 개수를 파악

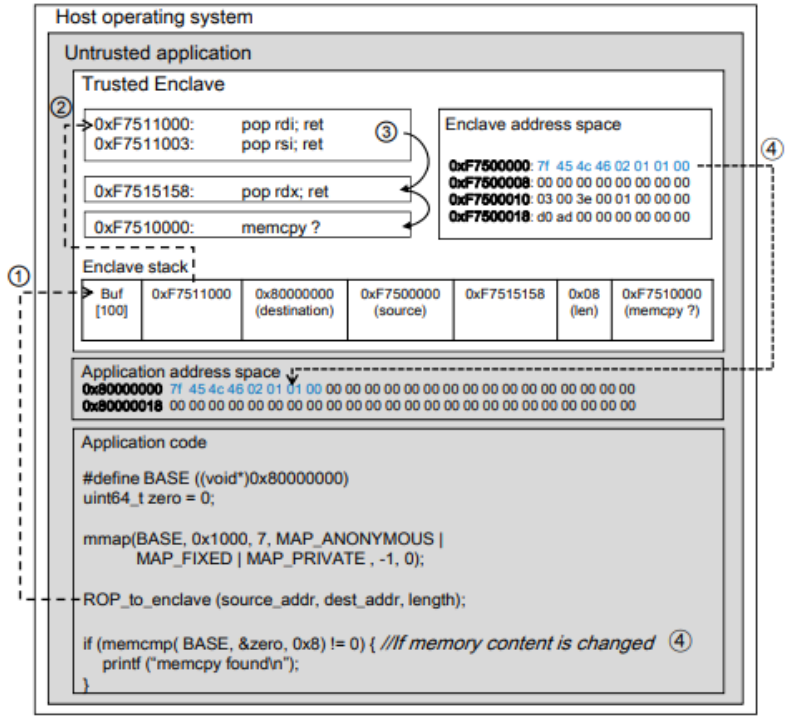


Hacking in Darkness (Oracle 2)

- 찾은 Gadget 중 필요한 레지스터를 사용하는 Gadget 구분
- EEXIT 함수를 이용하여 식별
  - ENCLU 명령을 찾은 뒤 Parameter에 찾은 Gadget을 사용
  - EAX의 값이 0x4인 경우 Enclave가 종료됨







Hacking in Darkness (Oracle 3)

- Memcpy() 함수를 이용하여 ROP Chain 제작
- 비 신뢰 메모리에 Enclave 메모리 Copy
- 비밀 데이터 해킹 뿐만 아니라 Malware 코드 실행 또한 가능

SGX-Bomb (Row Hammer Attack)

- DRAM 의 하드웨어적 결함
- Cell 밀집도가 높아져서 한 Row에 반복적 접근을 할 경우 bit flip 발생
- DDR4에서도 동일한 취약점 발견 사례 있음

SGX-Bomb

- Rowhammer 공격을 이용한 Bit flip 발생
- Bit flip을 이용하여 Enclave의 무결성 검사 실패 유도
- 무결성 실패 -> System 정지 -> Dos 발생
- User 권한으로 공격 가능
- 대상 머신의 DRAM 모듈은 Rowhammer 공격에 취약

Q & A