ADS presentation

Neline van Ginkel

June 24, 2018

Distrinet-iMinds, KU Leuven

Table of Contents

Introduction

Introduction to Protected Module Architectures

- Current Trusted Computing Base is huge (software & libraries, OS, hardware)
- Goal: Reduce Trusted Computing Base
- Security Architecture without need to trust OS
- Intel SGX: implementation of PMA on hardware level
  Supports hardware-based attestation & sealing

Introduction to Web Application Security

- Two parts: Server and Client
- Security on server: traditional methods for application security
- Security on client (browser): Same Origin Policy
- Origin: combination of protocol, host and port
- Security in transit: SSL/TLS (HTTPS)
- Client can not be trusted for (correct) execution of code (NoScript, adblocker, XSS)

Research so far

Literature (selection)

- Protected Module Architectures
  - Protected Software Module Architectures (Strackx et al.)
  - Innovative Instructions and Software Model for Isolated Execution (McKeen et al.)
  - Shielding applications from an untrusted cloud with Haven (Baumann et al.)
  - ...
- Formal verification
  - Secure Compilation to Modern Processors (Agten et al.)
  - Sound Modular Verification of C Code Executing in an Unverified Context (Agten et al.)

Literature (selection)

- Web security
  - Towards Tierless Web Development without Tearless Languages (Philips et al.)
  - NodeSentry: Least-privilege Library Integration for Server-Side JavaScript (De Groef et al.)
  - Protecting Users by Confining JavaScript with COWL (Stefan et al.)
  - ...
- Low-level side-channel attacks
  - Controlled-Channel Attacks: Deterministic Side Channels for Untrusted Operating Systems (Xu et al.)
  - Predicting Secret Keys Via Branch Prediction (Acıiçmez et al.)
  - ...

Publications, Experiments & Projects

- Paper: Towards Safe Enclaves (published in HotSpot 2016)
- Journal paper + experiments: Sound Modular Verification of C Code Executing in an Unverified Context (in preparation)
- Project: Tearless (in collaboration with VUB, in progress)
- Experiment: Sidechannel on SGX enclave with power consumption (not accurate enough)
- Experiment: Sidechannel on SGX enclave with branch prediction:
  - Only timing (collaboration with Raoul Strackx)
  - Branch Trace Store with "predicted" bit
  - Last Branch Record with "predicted" bit
- Experiment: Several small experiments with SGX enclaves:
  - Rust code in SGX enclave
  - Testing SGX instructions

Publications, Experiments & Projects

- Experiment: Several smaller experiments with web security:
  - Working with code from Tearless project
  - Experimenting with CSP
  - Working with nodejs & meteor
  - Diving into v8 source code

Research Plan

Research Plan

- Long-term Goal: Increasing security on the web with Protected Module Architectures
- Current situation: Code executed at client-side can not be trusted for sensitive operations (server-side verification needed)
- Ideally: run calculations client-side without need for exposing data and/or code with strong guarantees about correct execution
- Examples:
  - Calculations with sensitive information
  - Improved offline functionality
  - Input validation

Research Plan: Practical

- Create a browser extension communicating with an enclave.

  Google Chrome extension with Native Messaging

- Load an enclave at the client and let server verify it is correctly loaded.

  CPU based attestation with Intel SGX

- Provide an interface to load custom (public) code into the enclave

  Load arbitrary code / interpret/JIT JavaScript / ...

Research Plan: Practical

- Provide an interface to transfer data into the enclave with end-to-end encryption

  CPU based sealing with Intel SGX / key exchange protocol such as Diffie-Hellman

- Provide an interface for enclaves to communicate with untrusted JavaScript in browser (ocalls)

Research Plan: Practical

- Ensure current web policies are still enforced within browser extension

  f.e. Same Origin Policy / Content Security Policy / ...

- Define guidelines to write 'safe' API's

  Prevent sidechannels / privacy leaks / ... to untrusted context

- Formulate security guarantees provided by PMA & infrastructure

Contribution to Bachelor and Master education

Contribution to Bachelor and Master education

- Development of Secure Software: Project (2015-2016, 2016-2017, . . .)
- Informatica Werktuigen: Exercise Sessions (2015-2016, . . .)
- Design of Software Systems: Project (2015-2016)
- Guiding thesis: Building Memory-Safe SGX Enclaves (2015-2016)

- Summer school: IPICS 2016
- Distrinet CTF team Hacknam Style: internal workshops

Planned course units

Planned course units

- Formal Systems and their Applications
- Academic English: writing skills