PALANTIR

Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises

[H2020 – Grant Agreement No. 883335]

PALANTIR: Zero-trust architecture for Managed Security Service Provider

Trust in Execution Platforms Track

Dr. Maxime COMPASTIÉ
i2CAT Foundation,
maxime.compastie@i2cat.net

Prof. Antonio LIOY
Politecnico di Torino,
antonio.lioy@polito.it

C&ESAR 2022
Rennes (FR)
November 16th 2022

Agenda

- PALANTIR Project Introduction
  - Trust Problem Statement
  - Objectives
- Adopted Approach
  - Threat Modelling
  - Zero-trust Architecture
  - Remote Attestation
  - Security Orchestration
- Qualitative and Quantitative Evaluation
- Conclusion & Future Work

Introduction

- Motivation: Limited investment capacity from European SME/ME in Cybersecurity.
  - Externalise Cybersecurity (e.g. to Managed Security Service Provider, MSSP).
- Objective: Conceive & deliver a cybersecurity platform to MSSP and organisation internal usage.
  - Security capabilities as extended VNFs,
  - Deployed close to resources needing protecti
  - Available from as-a-service marketplace.
- H2020 PALANTIR project indicators:
  - EC-funded Innovation Action (IA),
  - 17 partners,
  - 5,3 M€ total budget,
  - 36 months duration (ends in 2023-08).

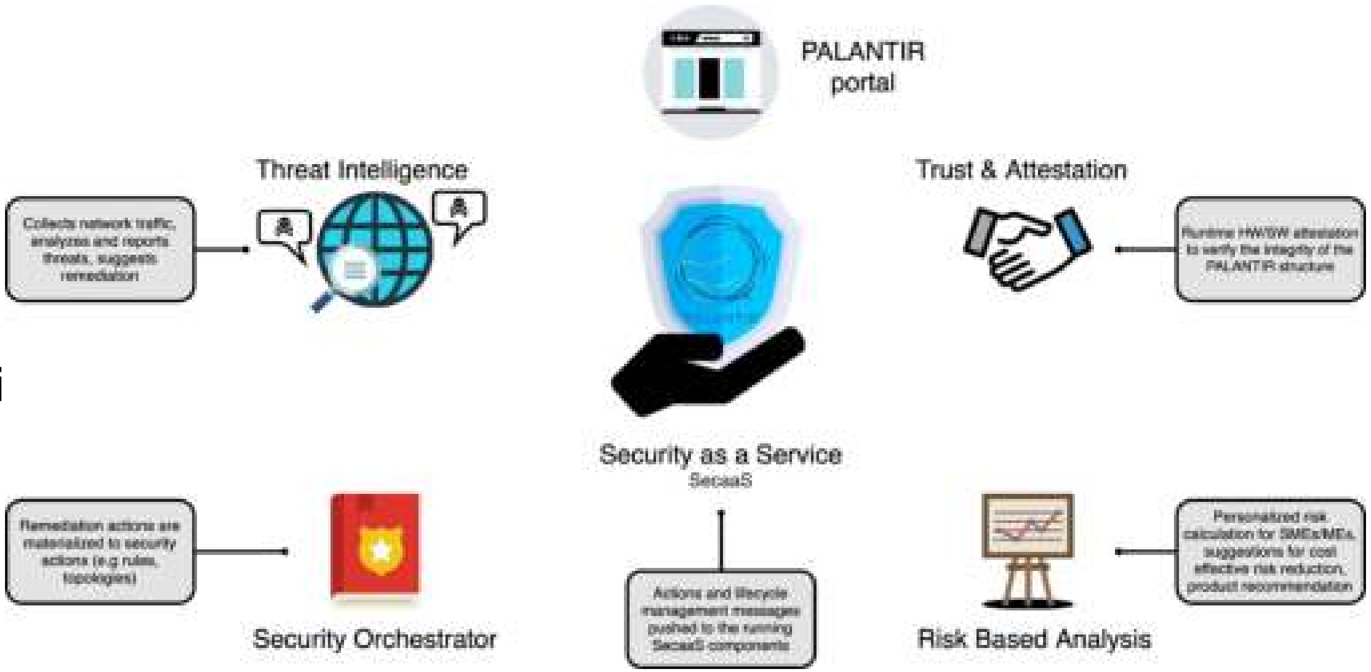


Figure 1: PALANTIR's SecaaS concept

Problem Identification

- "*Available from as-a-Service Marketplace*": The marketplace is open to contribution from third party developers … **But can we trust published SCs?**
  - Intentional malevolent behaviour: e.g. the malware case
    - Opportunity from a malicious developer to target vulnerable subscribers resources using a powerful distribution vector (PALANTIR Service Provider and its infrastructure).
      - Aggravated by the pervasiveness of some deployment models.
  - Unintentional malevolent behaviour: e.g. deficient secure programming practices, software supply-chain issue.
    - Creation by a developer of points of vulnerability in subscriber's infrastructure … and to the service and infrastructure providers as well.
      - Different levels of vulnerability: application, runtime, OS kernel and hardware,
      - Vulnerability surface exploitable by potential intruders in MSSP infrastructure.

=> Security validation process for SC published in the marketplace is necessary but insufficient.

Zero-Trust Approach

- Question: How to elaborate a trust model for a distributed MSSP?
- Approach: Do not trust any security capabilities instances, constantly monitor their integrity.
  - Zero-Trust: No participant in a network should be trusted.
  - Application of this principle to SC instances.
- Contributions:
  1. Trust model for MSSP deployment,
  2. Assessment strategy to continued integrity of asset,
  3. Orchestration techniques and interactions to enforce these strategies,
  4. Implementation & evaluation of the technical stack of the architecture

Threat Model

	PALANTIR Provider	**Infrastructure Provider**	**SC Developer**	**Subscriber**
Supervised components	PALANTIR Platform	Security Capabilities Hosting Infrastructure	Security Capabilities	Resources Needing Protection
Capacity to respond alone to threats	✓	✓	Limited (require new development)	✗
Proactive respon se to threats	✓	If contractually obliged	Limited to their knowledge of secure coding practice and sw. supply-chain	✗
Trust level	Trusted	Semi-honest	Semi-honest	Least trust

PALANTIR Zero-Trust Architecture

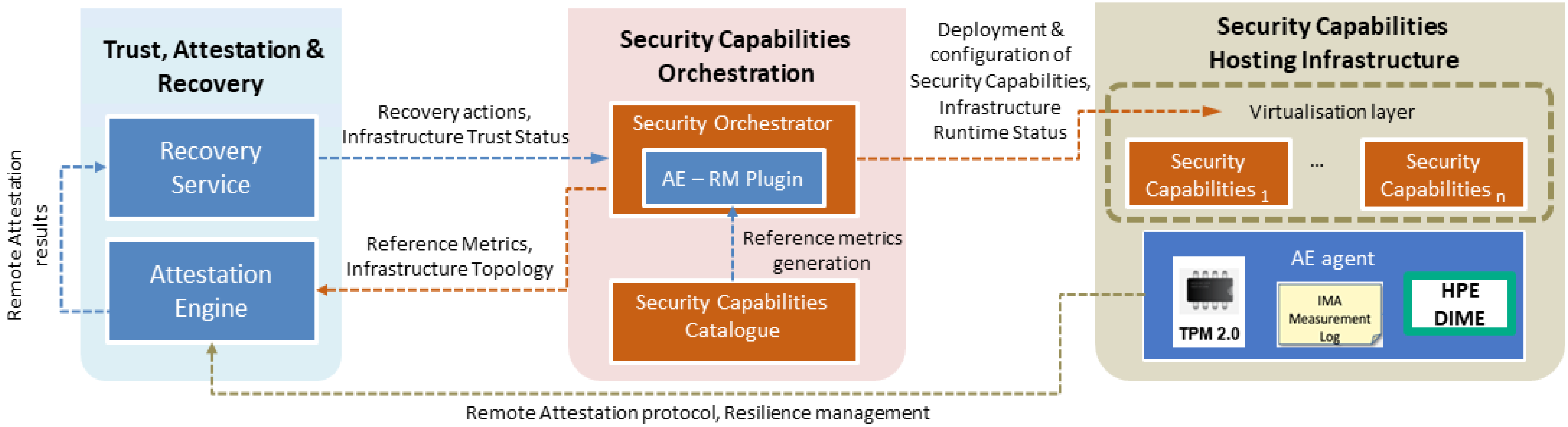
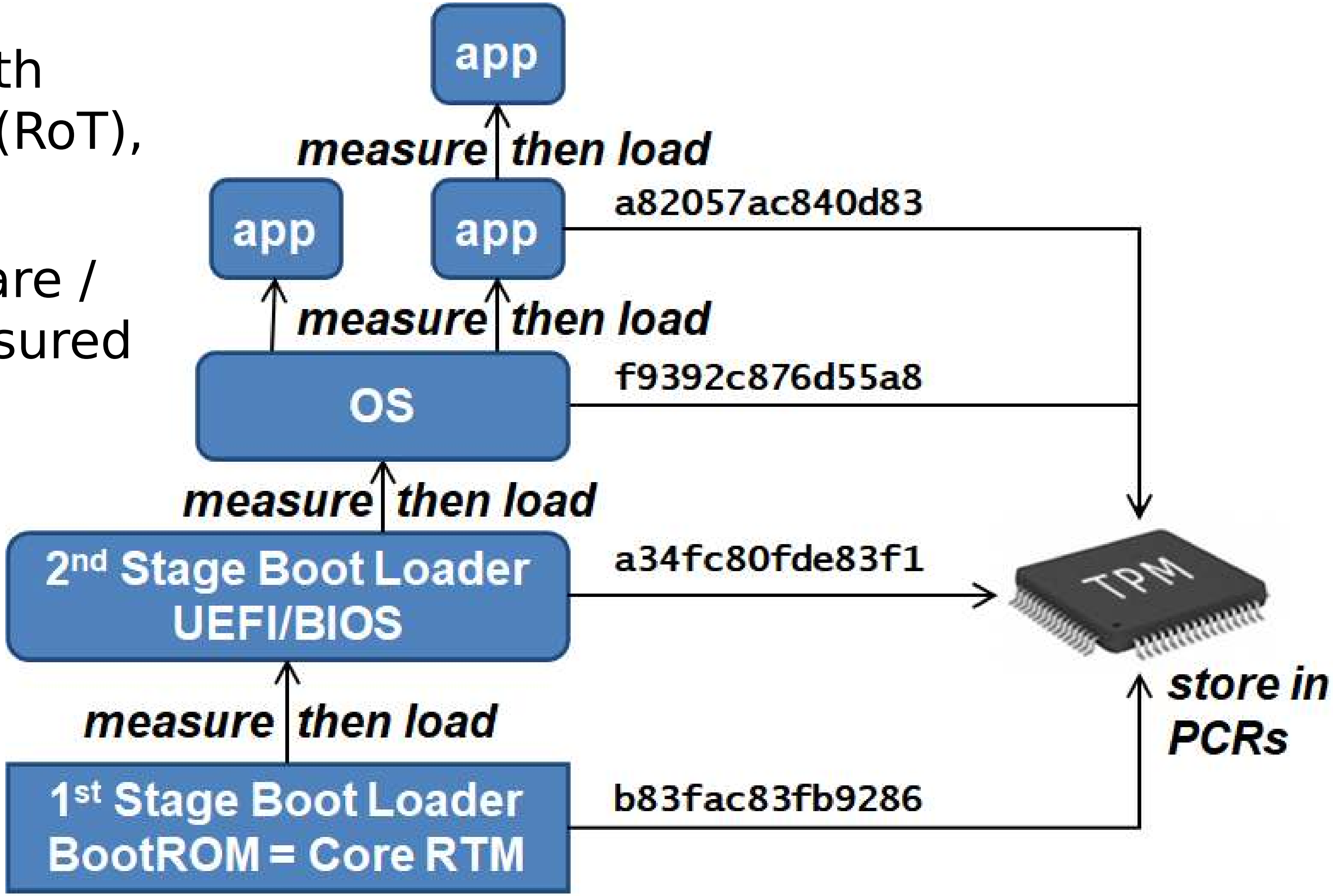


Figure 2: PALANTIR's Zero-Trust Architecture

- 3-tiers architecture
  - Trust, Attestation & Recovery: Detect integrity compromise (AE) and supervise countermeasures (RS),
  - Security Capability Orchestration: Store knowledge on available SCs (SCC), oversee the lifecycle of their instances (SC),
  - Security Capabilities Hosting Infrastructure: Provide facilities to operate SC instances (virtualization layer) and retrieve metrics from Integrity Measurement Architecture (AE-agent).

Measured boot (and operations)

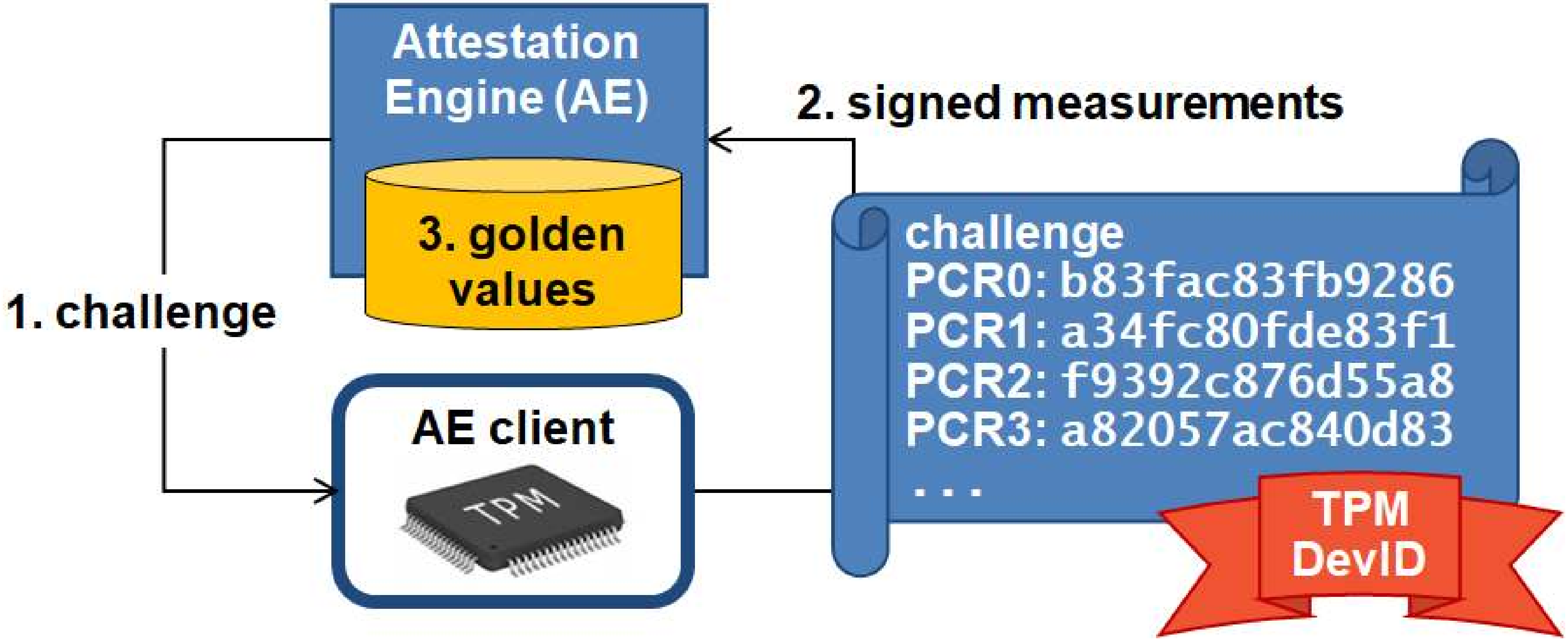
- Each node equipped with a physical root-of-trust (RoT), the TPM
- TPM (and proper firmware / software) used for measured boot



Remote Attestation I

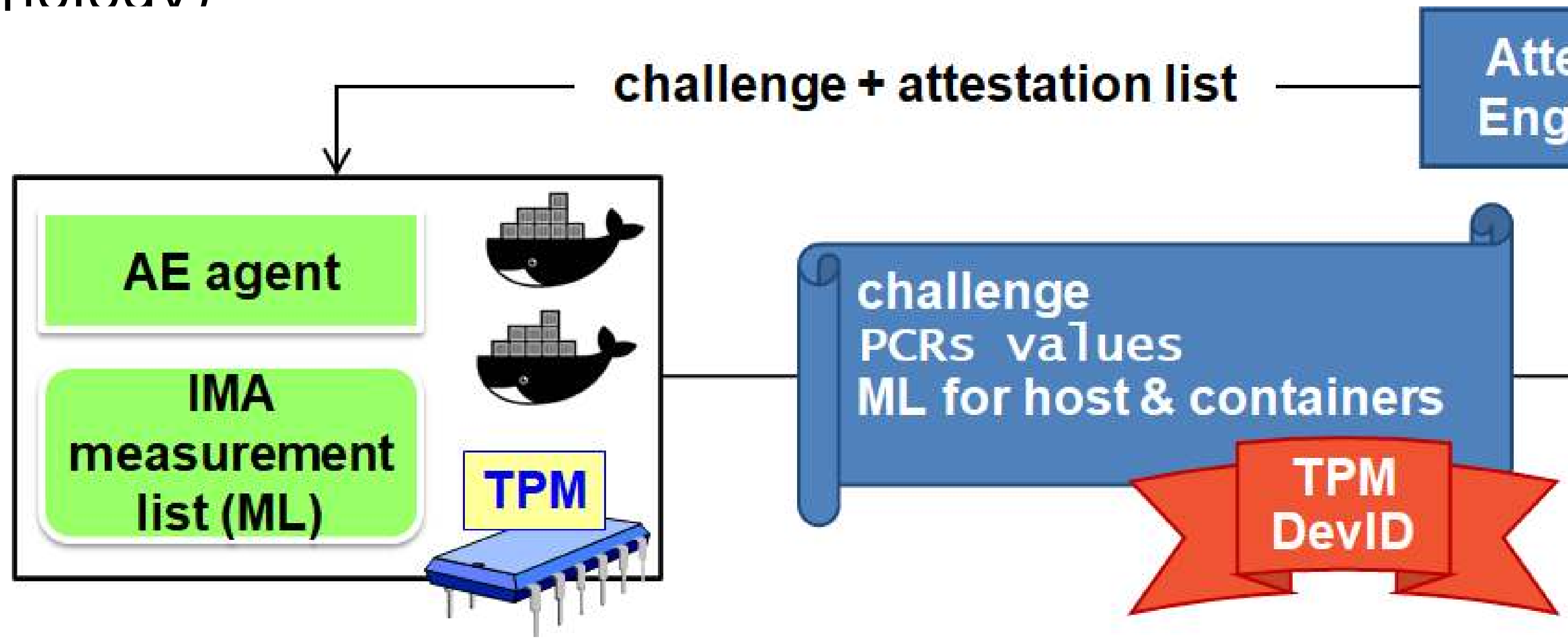
Basic remote attestation procedure:

1. challenge (=nonce)
2. measurements (and nonce) returned signed with the device's key
3. validate signature (crypto + ID) and check measurements against Golden Values



Remote Attestation II

- application-level operations (exec, read, …) are measured by Linux IMA (Integrity Measurement Architecture)
- IMA extended to measure also operations inside containers
- detection of compromised host (stop host with all its containers) or compromised container (restart only that container, may be with a different technology)



Security Orchestration

- **Two layers for coordination:**
  - Upper-level decision logic (recovery service):
    - Elicit the remediation procedures based on AE results,
    - Coordinate the conduction of the remediation procedures on SCs.
  - Lower-level enforcement level (Security Orchestratior)
    - Expose the interfaces to the upper layer to act on SCs.
    - Interface with a 3rd party orchestrator (Management and orchestration software), to conduct lifecycle operation on regular VNFs.
      - (e.g. reinstanciation, redeployment of an equivalent SC)
    - Enforce mutual authentication, autorisation and encryption between SO and scrutinised SCs.
      - applied to SO-SC interfaces and SO-SCHI (VNFM-NF and Orchestrator-VIM in ETSI terminology)

Evaluation: SC Integrity Measurement

- Orchestrator provides to AE the list of nodes and deployed SCs, along with their "golden values"
- Attestation Engine will periodically provide integrity status for:
  - Hardware (tested by changing the reference measure)
  - Firmware (tested by disabling secure boot and rebooting the platform)
  - Operating System (tested by adding and executing a new malicious binary)
  - Runtime - with DIME (tested by injecting a new kernel module)
  - SC (tested by modification of a legitimate binary, change of a configuration file, addition and execution of a new malicious binary)
- Based on the attestation result, the Orchestrator decides an appropriate remediation action (restart node, select a different node, restart container, select different technology for the same SC)

Evaluation: Remediation Decision-Making Process

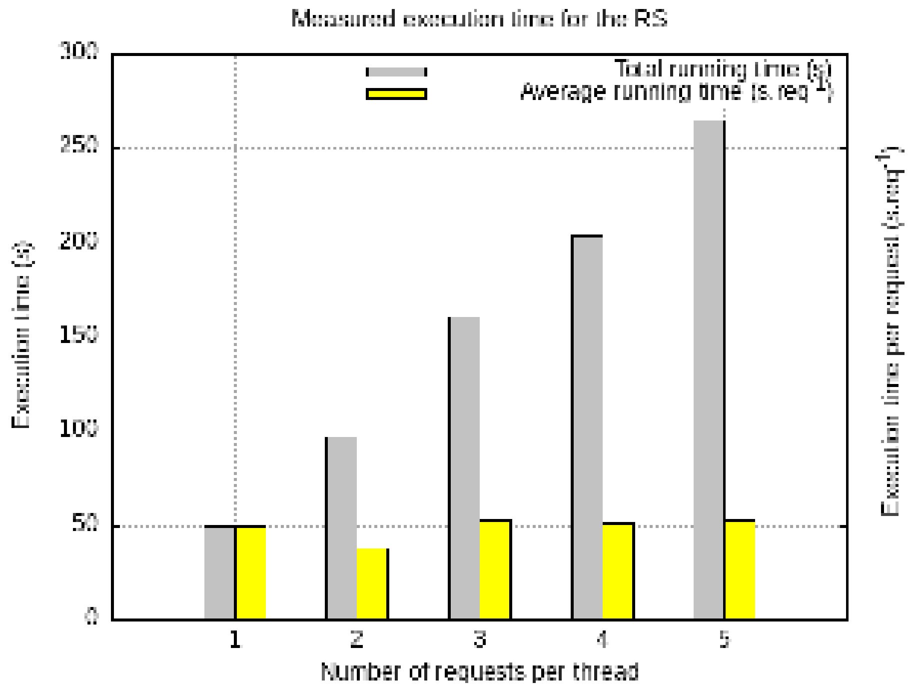
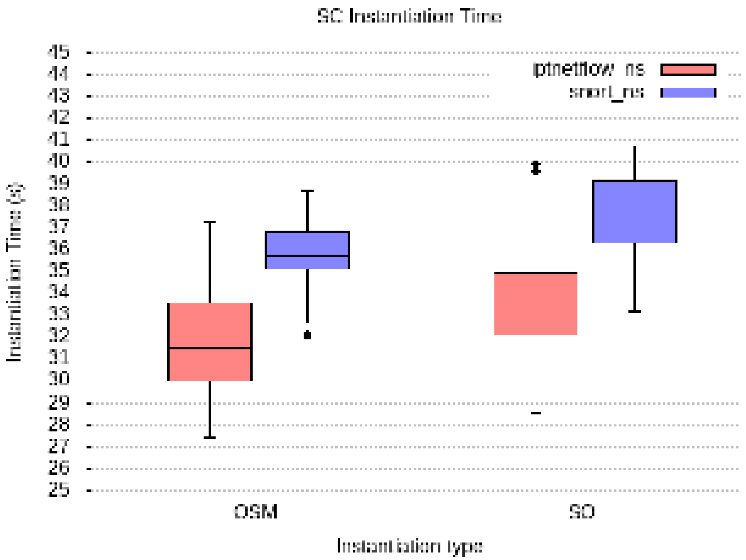
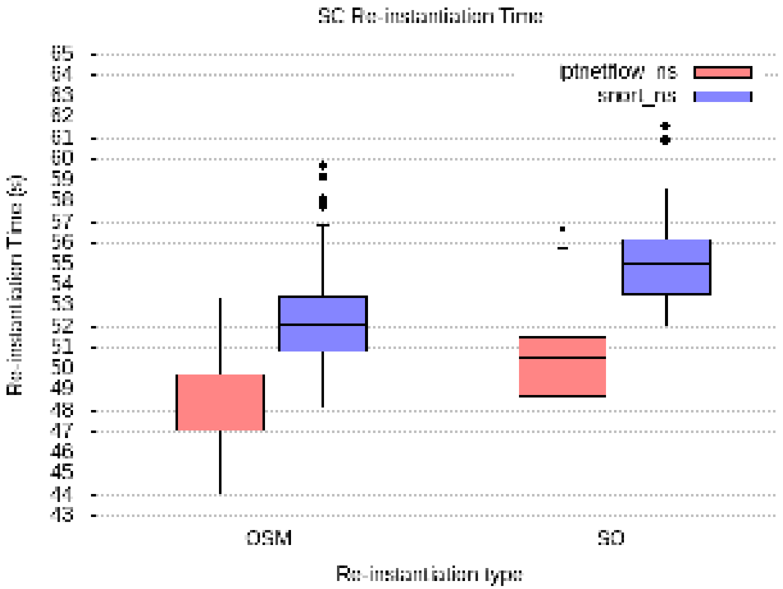
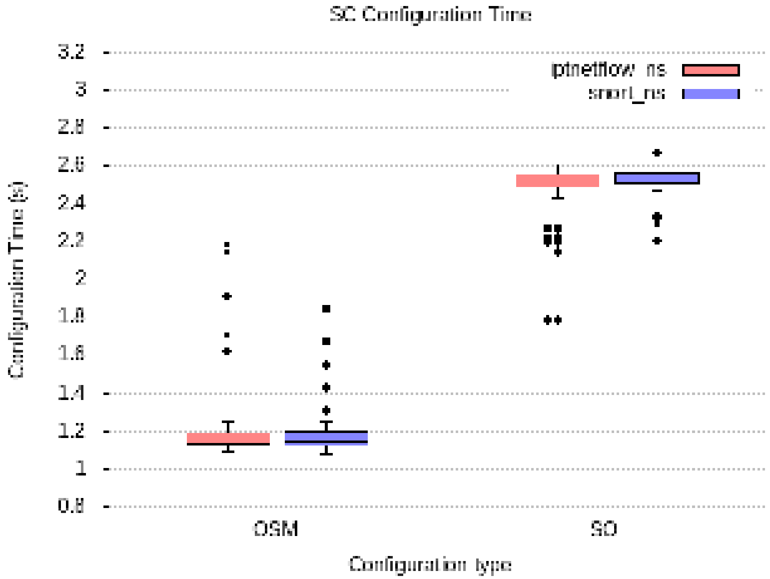


Figure 3: Evaluation times of the RS component

Evaluation: Security Orchestration for the Decision Enforcement







Figures 4 (left) and 5 (middle): Distribution of the instantiation (left) and re-instantiation (right) times across SCs between SO and OSM

Figure 6: Distribution of the configuration times across SCs between SO and OSM

Evaluation: PALANTIR Zero-Trust Attestation

- Attestation of the SCHI + SC's integrity performed with a polling approach
  - Avoids DOS attacks
  - Push from SC unreliable (could be stopped by attackers)
- Basic performance:
  - attestation cycle for one SC 1.2-1.6s (16-32 SCs)
  - 0.7s for "quote" creation (constant, mostly depends upon TPM) + network & verification times
- Experiment:
  - Attestation every 2s, notifications to RS every 10s (one remediation at a time)
  - Less than 120s to stop the attack (avg 72s) = detected by AE, remediation suggested by RS (SC removal), and implemented by SO
- Performance improvements for attestation
  - Parallelization of attestation cycles
  - Bottleneck is TPM (can we improve it?) not network or verifier

Conclusion & Future Work

- Achieved modern ZTA for MSSP
  - Application to SecaaS principle,
  - Proposed architecture with prototype.
- Based on standard hardware and (mostly) open-source software
- Good performance
  - Quantitative evaluation provided.
- Possible improvements:
  - Detection of in-memory file-less attacks,
  - Support of attestation for hardware components,
  - Generation of golden values,
  - Use of attestation logs for forensics analysis
  - Extending ZTA security model to customer infrastructure as well.

Follow us

https://www.palantir-project.eu/

@ProjectPalantir

PALANTIR Project

info@palantir-project.eu

https://github.com/palantir-h2020/

PALANTIR has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 883335

Back-up Slide

Discussion

ZT tenet (NIST SP800-207)	PALANTIR?	Explanation
All data sources and computing services are resources.	**YES**	All SCs in the SCHI are resources of the PALANTIR ZTA.
All communication is secured regardless of network location.	**YES**	All communication between the PALANTIR components in the control plane and the SCs and SCHI, is secured.
Access to individual enterprise resources is granted on a per-session basis.	almost	Access request is granted on a per-session basis for most of individual PALANTIR resources.
Access to resources is determined by dynamic policy—including the observable state of client identity, application/service, and the requesting asset—and may include other behavioural and environmental attributes.	**YES**	Access to PALANTIR resources depends on dynamic policies since, when the security posture of a resource get compromised, it is immediately isolated and remediated by the actions enforced by the RS and the SO.
The enterprise monitors and measures the integrity and security posture of all owned and associated assets.	**YES**	The integrity and security posture of all PALANTIR resources is continuously monitored through the AE.
All resource authentication and authorization are dynamic and strictly enforced before access is allowed.	almost	Access to most of PALANTIR resources is granted with dynamic policies for authentication and authorisation.
The enterprise collects as much information		Monitoring data from the AE are used to