ASSURED VISION TOWARDS TRUSTWORTHY “SYSTEMS-OF-SYSTEMS”

ASSURED Webinar:
Non-Intrusive Code Coverage: How to Use ASSURED for Trace-based Debugging and Runtime Analysis

Dimitris Karras

UBITECH

20/06/2023

www.project-assured.eu

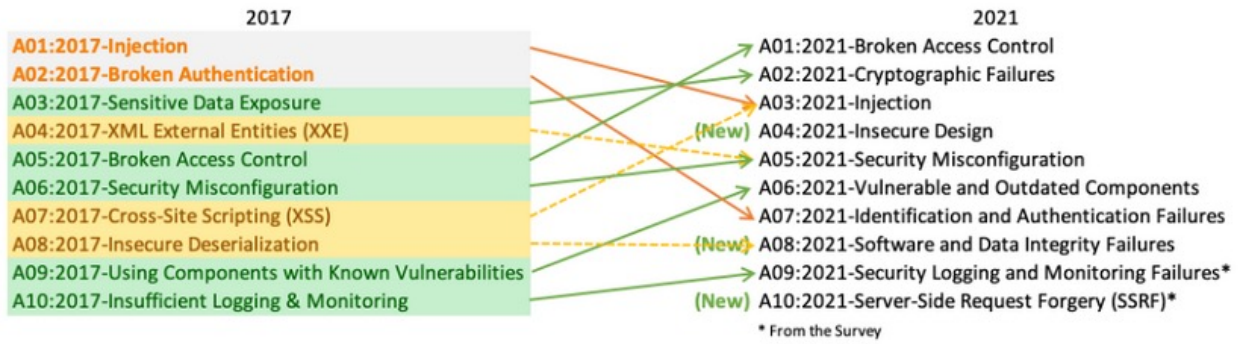
BACKGROUND AND MOTIVATION

As the demand for increasingly autonomous Cyber Physical Systems (CPSs) grows, so does the need for **certification mechanisms during runtime**.

- × Current methods towards validation require exhausting offline testing of every state scenario.
- ✓ Therefore, we aim to provide ***security and privacy guarantees for devices, as well as the system as a whole, during runtime!***

Novel assurance services are needed to ensure that operation does not put the systems or the people operating them in danger:

- Ensure **trusted execution** of (insecure) components
- Safeguard **code updates** against tampering
- Firmware and software **compliance** to execution policies



ENISA threat landscape report – top 10 threats

THE VISION OF ASSURED

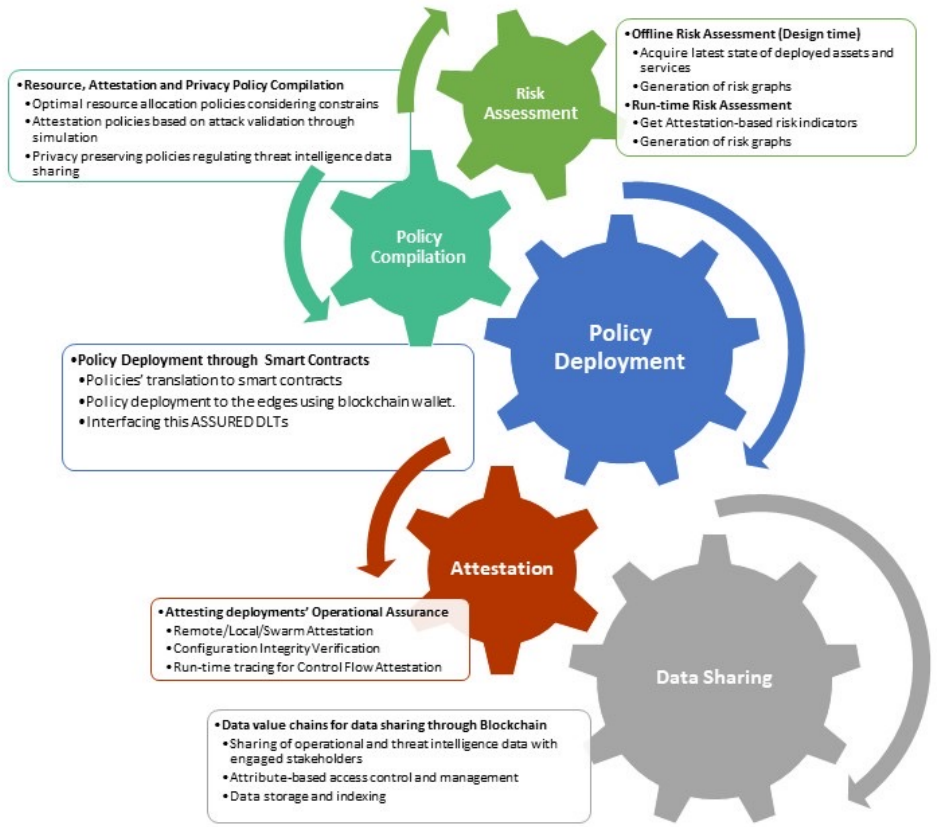
The core vision of ASSURED is the development of a complete framework that can provide **operational assurance** to large-scale Systems-of-Systems (SoS) comprising various **heterogeneous devices**, characterized by different **security and privacy requirements**.

Establishment of Trusted Service Graph Chains in next-generation "Systems-of-Systems" addressing **Security**, **Safety** and various levels of **Trustworthiness** for mixed-criticality services.

Adoption and implementation of the **Zero Trust** concept with the principle "*Never Trust, Always Verify*" for assuring vertical trust for all devices comprising the supply chain.

ASSURED RESEARCH IN TRUSTED COMPUTING, BLOCKCHAIN & LIGHTWEIGHT CRYPTO

Enhanced Operational Assurance	Increase trust to a device output by assessing its configuration & execution state – **Real-time tracing capabilities without impeding performance**
Risk Assessment	Identity risk interdependencies in a service graph chain that can affect the safety of the system
Threat Intelligence Information Sharing	Secure and auditable sharing of operational and attestation data only to authorized & authenticated devices & users – Useful for **Certification**
Decentralized Identity Management	How to ensure that each entity is the one that it claims to be – TC-based Wallet running at each user/device
Safety Zones Detection	Optimal Deployment of security (attestation) policies
Vulnerability Analysis	Protection of cyber-physical systems through run-time attack path analysis



ASSURED COMPONENTS

ATTESTATION ENABLERS

- Attest both the correct configuration & execution of a device
- Different types of attestation tasks depending on the requirements
- Control-flow Attestation, Configuration Integrity Verification, Direct Anonymous Attestation, Swarm Attestation
- Jury-based Attestation for resolving inconsistencies in the provided claims

Innovation:

- Lightweight attestation capabilities
- ML-based CFA
- Orchestration of the different attestation tasks depending on the policies (protection profiles)

RUNTIME TRACER

- Real-time tracing capabilities of the configuration state & control-flow graphs
- SW-based, HW-based, and hybrid
- Lightweight enough to operate in resource-constrained devices

Innovation:

- Does not affect software performance
- Non-intrusive

SSI WALLET

- Bridge for the secure management of cryptographic material & continuous authorization and authentication
- Following the SSI concept
- Capable of producing Verifiable Proofs for device attributes

Innovation:

- Protected under HW-based key (DAA)
- Merging of the SSI and trusted computing benefits

BLOCKCHAIN-BASED CONTROL

- Secure information exchange & data sharing
- Attribute-based Access Control
- Attribute-based Encryption
- Searchable Encryption

Innovation:

- Decentralized ABE
- **Certification capabilities** due to the auditable recording of all data transactions

ASSURED COMPONENTS

RISK ASSESSMENT

- Identification & Calculation of risk interdependency graph
- Based on definition of hw- & sw-assets from the system administrator
- Prerequisite for the calculation of optimized set of security policies

Innovation:

- Consider both **security & privacy** related vulnerabilities
- Attack Path Calculation

POLICY RECOMMENDATION

- Calculation of the optimized set of security policies
- Set of attestation policies
- Scheduling of attestation & computational tasks

Innovation:

- Multifactor constraint problem solving
- Different dimensions – convergence of security, safety, and resource

SECURITY CONTEXT BROKER

- Bridge for interacting with the Blockchain
- Deployment of attestation policies through smart contracts
- Attribute-based Access control capabilities

Innovation:

- Adoption of the gRPC concept for automatic dissemination of events
- Access control based on the use of Verifiable Credentials

ATTACK VALIDATION

- Virtual representation of the physical devices
- Processing of real-time system raw traces for attack path identification
- Simulation & Emulation of various attack vectors

Innovation:

- Device Behavioral Analysis
- Mutation Fuzzing & Concolic Testing

ENVISIONED USE CASES

SMART MANUFACTURING

- Accident prevention for humans working in tandem with machinery (robotic arms)
- Validation against a malicious user attempting modification
- Data integrity and trustworthiness for position of worker (equipped with RFID)

SMART AEROSPACE

- Need to increase the trustworthiness of all internal components of an aircraft
- Need for fast and secure SW updates
- Need for verification in the integration of new products and system level solutions
- Protection against security misconfiguration, vulnerable and outdated components

SMART CITIES

- Use of smoke and gas detection sensors, CCTV cameras
- Strong requirements for anonymity and privacy of users
- Strong authentication and authorization of various stakeholders when accessing sensitive information

SMART SATELLITES

- Collaborative execution of safety-critical processes by multiple satellites
- Lightweight authentication and secure communications
- Integrity of mission critical payloads and secure software updates

USE OF TRACING IN ASSURED

Remote Attestation: The method by which a Prover device authenticates the correctness of its configuration state and/or the execution of software processes, through the issuance of an attestation challenge by a Verifier and the provision of an appropriate response by the Prover.

- **Control Flow Attestation:** Verifies the correctness of the execution of a software process and aims to identify whether the execution flow is benign or malicious.
- **Configuration integrity Verification:** Verifies the correctness of the configuration state of a device, compared to a state that is known to be correct and trustworthy

- The **ASSURED Runtime Tracer** performs near real-time low-level code inspection and tracing to collect raw trace data as **attestation evidence** in both CFA and CIV, thus capturing the requirements of ASSURED remote attestation, while striking a balance between **precision**, **efficiency**, and **transparency!**

TYPES OF TRACING

The following tracing methodologies are available in the literature:

- **Software-based Tracing:** Employs a purely SW-based approach and does not require any additional hardware. However, it is the slowest approach, as it uses the computational capabilities of resource-constrained devices.
- **Hardware-based Tracing:** Requires hardware extensions to common processors to enable control-flow event tracing in parallel with the execution of the program. High hardware requirements, but also best performance.
- **Pseudo-HW-based or HW-assisted Tracing:** Does not require dedicated hardware but assumes the existence of appropriate hardware tracing features. For example, **Intel PT** and **ARM Coresight** architectures support required SW and HW features.

Each tracing methodology has its benefits and drawbacks! SW-based Tracing does not have access to additional HW, but HW-based Tracing is much more efficient.

THE ASSURED APPROACH TO TRACING

The priority in ASSURED is, given the SOTA, to *create a SW-based Tracer with optimized design and implementation*, with features that *aim to provide somewhat equivalent performance with the HW-based Tracer* (as close as possible).

As we will see in the following, the basic parameters considered in the design of the Tracer are:

- Do we need to attest the entire codebase of program in embedded system? No, because we perform **property-based attestation** – we are interested in specific properties/functions, where SW-based tracing is appropriate.
- Internal mode of operation of tracing: While the tracing per se is efficient, the most significant overhead is the **decoding phase**, to be able to extract semantic representation of graph in a way that makes sense and can be translated to execution. *What level of granularity in terms of attestation information do we need? We can do accurate attestation without debug symbols!*
- Evaluation and comparison against HW-based tracer, specifically **ARM Coresight** (which we implemented in ASSURED).

THE ASSURED APPROACH TO TRACING

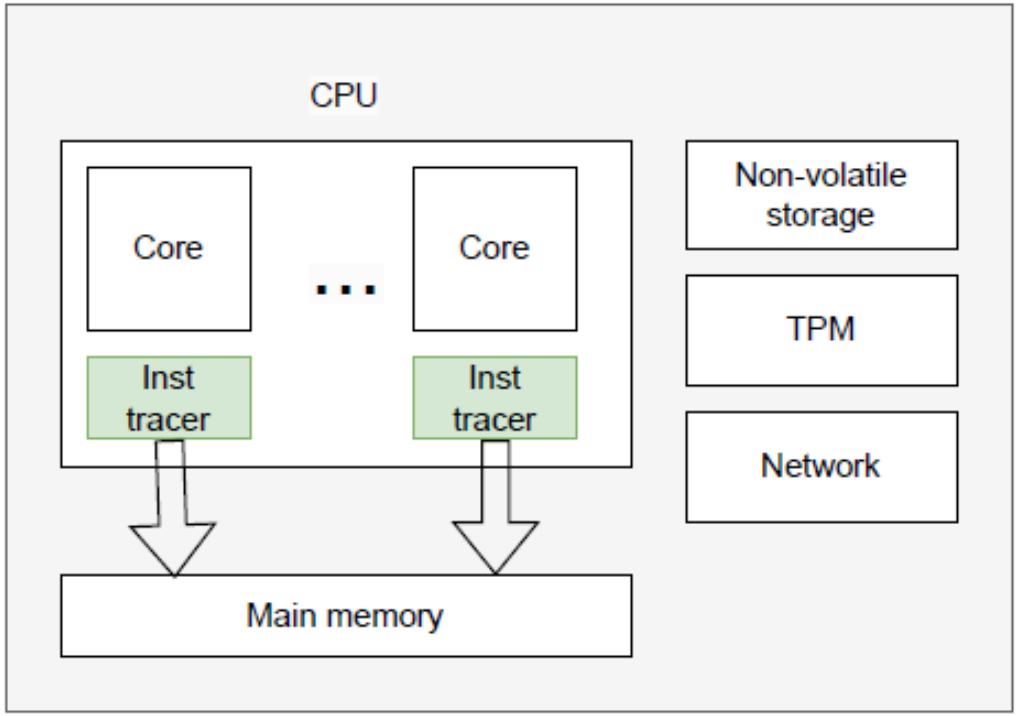
In ASSURED we aim to **reduce tracing performance impact** and **provide seamless integration with the executed programs** without requiring developers to change them in advance.

- *ASSURED is the first initiative towards providing purely SW-based Tracer capable of providing high level of detail and accuracy in tracing system functional properties!*

We use **hardware-assisted tracing capabilities** whenever they are available in commodity processors and **software-based Tracer** to efficiently recover information made available by the HW.

- *Since HW is highly optimized and has negligible impact on performance, and Tracer can run in parallel with other programs without interfering, we both achieve non-intrusiveness and reduce need for trap-based tracing using HW!*

EDGE DEVICE ARCHITECTURE



Edge device architecture to support Tracer: Tracer should be part of TCB.

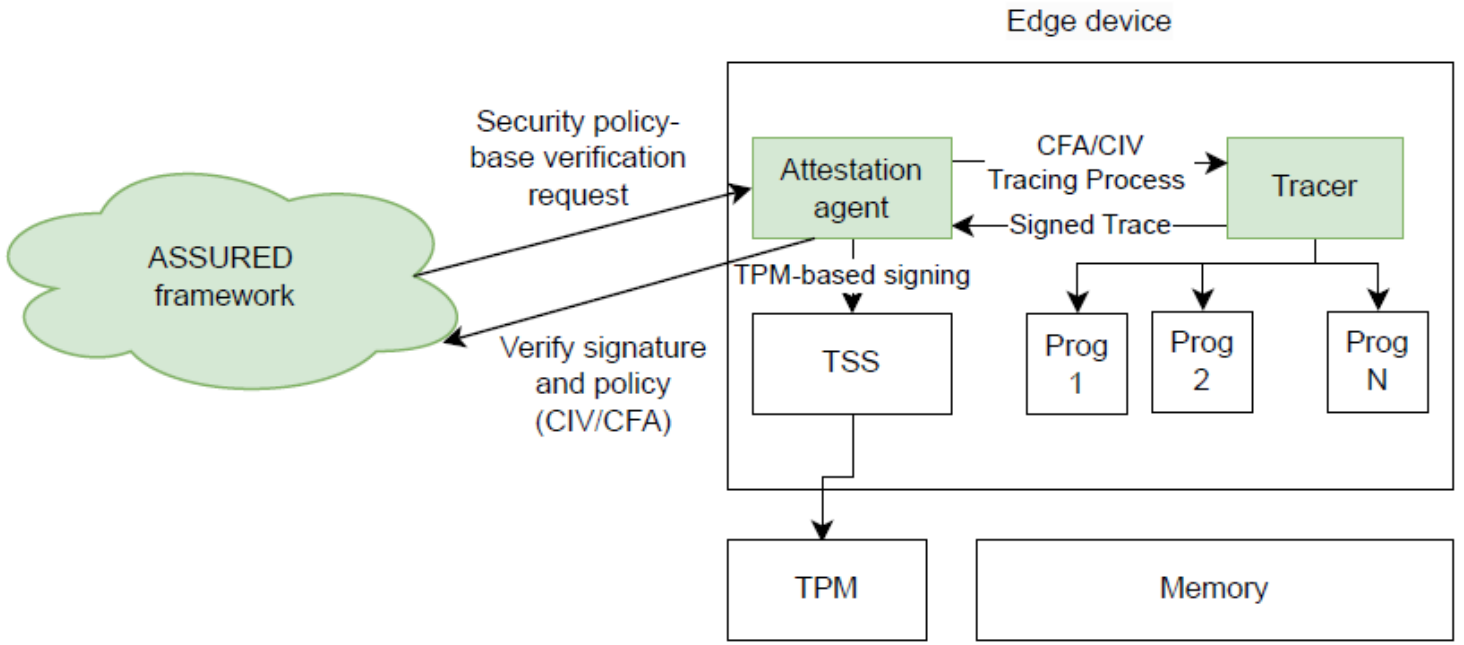
- **Trusted Platform Module (TPM)**: A physically isolated hardware module that acts as a Root-of-Trust (RoT) and is used to assert the authenticity of the traces.
- **Trusted Execution Environment (TEE):** Isolated execution environment that supports execution of security-sensitive code. Tracer is executed in TEE
- **TEE capable processor:** CPU that is able to support the execution of the TEE
- **HW tracing capable processor:** To support HW-based tracing, CPU that stores control flow event data in a dedicated memory location is needed.

ASSURED TRACER WORKFLOW

Existence of Trusted Component (TPM) prerequisite! High-level overview on tracing in the ASSURED workflow:

1. The **Attestation Agent (AA)** Acts as a gateway to the TPM-based Wallet, receives attestation policies from ledger
2. AA extracts properties to be traced and triggers **Tracer** to initiate operation (CFA, CIV)
3. AA verifies correct signing of traces with device TPM and signs them with **Attestation Key (AK)**
4. Verifier receives traces, validates signature, and performs attestation logic

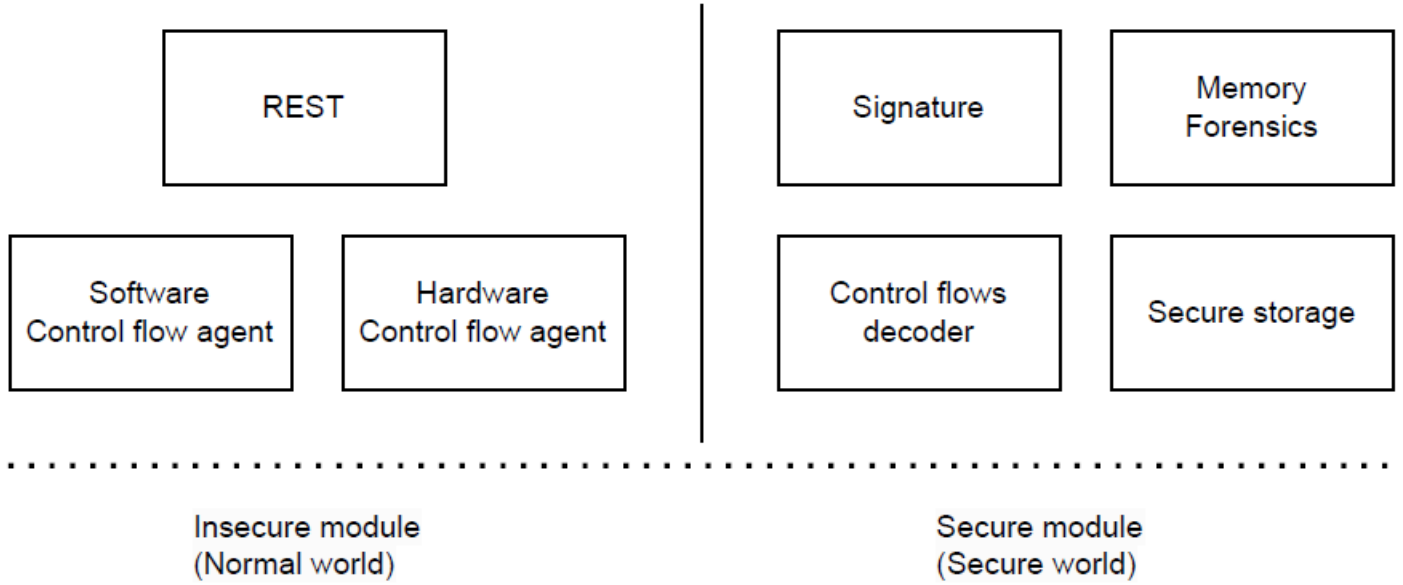
More details for establishment of secure-authenticated channel between tracer-TPM in following presentations!



TRACER SOFTWARE ARCHITECTURE

Tracer SW architecture (Secure World):

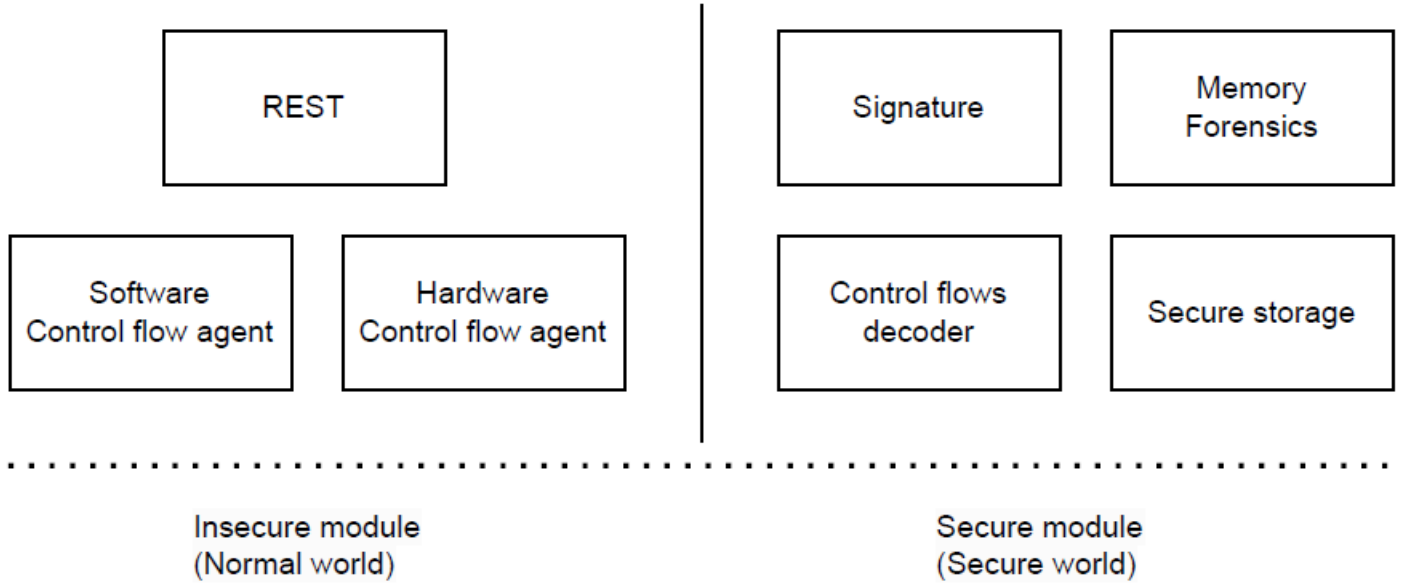
- **Secure Storage:** Designated configurable memory region to securely store control flows and runtime configurations.
- **Secure Signature:** Signing operation on the traces by the TEE using private key
- **Decode control flows:** For HW-assisted control-flow tracing, module that devices HW-generated packets
- **Memory forensics:** Used to recover runtime integrity measurements of executing processes in edge platforms



TRACER SOFTWARE ARCHITECTURE

Tracer SW architecture (Insecure World)

- **Communication Agent:** Implements REST service performing communication with the TPM, runtime CFA and CIV requests
- **SW Control Flow Agent:** Uses dynamic binary rewriting to extract visited basic blocks and pass them to safe storage.
- **HW Control Flow Agent:** Uses ARM Coresight tracing to extract visited basic blocks of trace application and pass them to safe storage.



OPEN TOPICS AND NEXT STEPS

- **Latency of HW-assisted tracing**: Tracing with ARM Coresight or IntelPT has negligible performance impact, but decoding, disassembly, and analysis are SW-based and have latency that may impact performance.
- **Kernel-level CFA tracing:** Verifying kernel execution integrity is an open problem. Kernel execution consists of non-deterministic sections, such as scheduling of different processes.
- **Concurrency issues:** In ASSURED, we focused on tracing single-threaded processes. Tracing multi-threaded programs may induce larger performance impact due to heavier usage of device resources and creation of non-deterministic CFGs.
- **CIV Tracing with Invalid Sections:** Integrity measurements in static files poses challenges, due to changes in some values in the memory due to dynamic loading of position-independent code sections, and pages that are fetched on-demand.

THANKS

PROJECT-ASSURED.EU

@Project_Assured

ASSURED project is funded by the EU's Horizon2020 programme under Grant Agreement number 952697