A cybersecurity framework to guarantee reliability and trust for digital service chains – GUARD

Matteo Repetto[1], Armend Duzha[2] and Joanna Kołodziej[3]

[1] IMATI-CNR Genova, Italy, e–mail:matteo.repetto@ge.imati.cnr.it

[2] Maggioli, Sant'Arcangelo di Romagna, Italy, e–mail: armend.duzha@maggioli.it

[3] Research and Academic Computer Network (NASK), Warsaw, Poland, e–mail: joanna.kolodziej@nask.pl

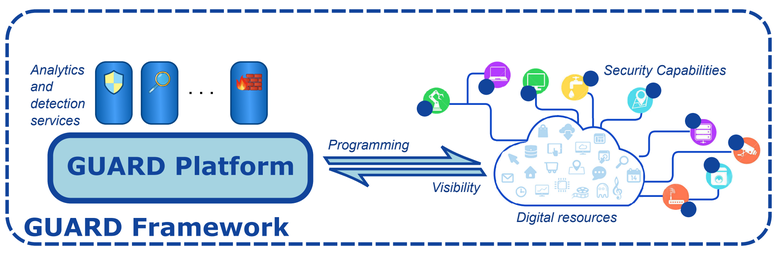
Abstract

The GUARD project developed an extensible platform for building detection and analytics services for advanced assurance and protection of trustworthy and reliable business chains which span multiple administrative domains and heterogeneous infrastructures. GUARD advocates the implementation of embedded security capabilities in digital services, that can be accessed and orchestrated through API similar to what already happens for management and operation purposes. GUARD features are demonstrated on two challenging use cases, in the Smart Mobility and eHealth domains.

The GUARD Framework

The GUARD framework is conceived as a new paradigm to implement detection and analytic processes for digital service chains. It includes:

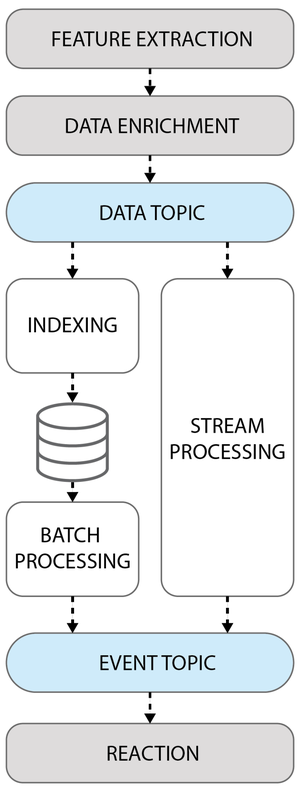
- security capabilities exposed by digital services, which could be implemented by common security agents;
- a platform that orchestrates security capabilities by discovering, configuring and chaining them in *security analytics pipelines* (SAPs);
- detection and analytics services, for discovering attacks and anomalies throughout the whole service chain.



Overview of the GUARD framework.

GUARD Security Analytics Pipeline – SAP

The processing workflow where monitoring and inspection data feeds a battery of detection algorithms and analytics engines, according to typical Security Information and Event Management (SIEM) architectures.
GUARD SAP model extends from local agents to the internal platform and includes stream and batch processing patterns.

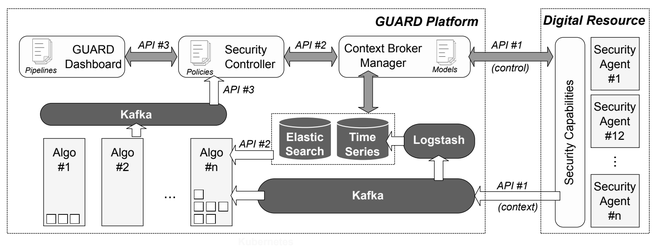


The logical stages of a GUARD SAP.

Acknowledgment

The work of all authors of this paper was supported in part by the European Commission, under Grant Agreement no. 833456 (GUARD).

The GUARD software architecture

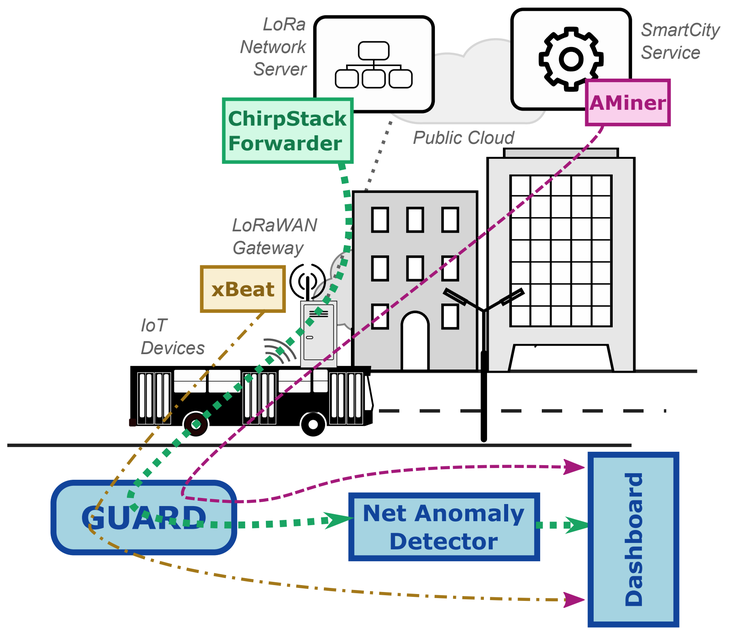


Software architecture of the GUARD Platform.

- **Context Broker Manager (CB-Manager)** – The interface and abstraction layer that discovers the service topology and manages remote security capabilities.
- **Security Controller** – The smart engine that takes decision based on control policies, triggers response and mitigation actions based on the evolving context (events, conditions) generated by the detection algorithms or directly by remote agents, is responsible for the setup of SAPs.
- **GUARD Dashboard** – The user interface to visualize the service topology, security features, and data generated by agents and detection algorithms, features an editor for creating analytics pipelines, starting from the list of available detection and analytics algorithms, as well as security capabilities of each digital resource.

Smart City Use Case

Protection of Smart City services.

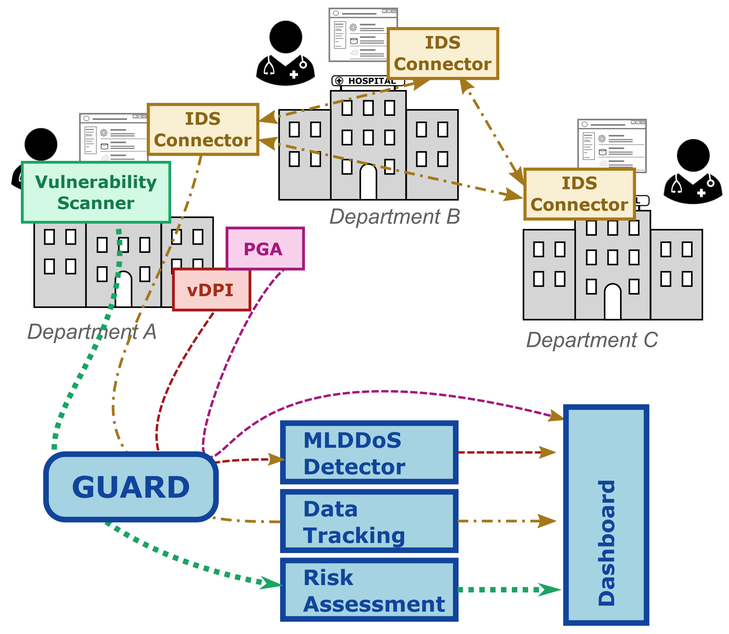


Three SAPs designed:

- Detection of attacks to Web Services and internal MQTT;
- Detection of anomalies in LoRa traffic;
- Monitoring remote gateways.

e–Health Use Case

Data sovereignty and service integrity.



A data sharing service and three SAPs designed:

- Risk assessment by vulnerability scanning;
- Flow classification to mitigate DoS;
- Signature detection and periodic update from the Network Telescope.

Conclusions

- GUARD framework is based on a robust abstraction paradigm that leverages detection and monitoring capabilities embedded in each digital component, hence overcoming the impossibility to deploy hardware or software agents in 3rd party's infrastructures.
- GUARD allows to discover at runtime both relevant security-related characteristics of each digital resource (e.g., hardware/software environment, software version, monitoring and enforcement agents) and their relationships, which are used to build and update the service topology. Such knowledge is the necessary building block to create security analytics pipelines that feed detection algorithms in a SIEM-like architecture.
- The project has also investigated code augmentation to provide programmatic visibility over digital resources. In this respect, the eBPF technology has been investigated to safely inject custom code into a running kernel.