PLEDGER

The PLEDGER approach to Security on Cloud-Edge environments

Olga Segou, PhD
Senior RID Specialist
Research and Innovation Development Department
Intrasoft International S.A.

This project has received funding from the European Union's Horizon 2020 research and Innovation Programme under Grant Agreement No. 871536

PLEDGER Performance optimization and edge computing orchestration for enhanced experience and Quality of Service

- **Current approaches on edge computing are not sufficient to address this forthcoming massive usage of edge computing, especially in the frame of large IoT deployments in smart cities and industrial applications.**
- The main goal in such scenarios is to ensure that the overall offered **Quality of Service (QoS)** fits the application needs over the edge or edge/cloud deployment.
  - **Speed and latency** issues have been identified as the top barrier in this domain, while **cost and reliability** (meeting the provider Service Level Agreements - SLAs) are the top and second most important factors for evaluating edge and cloud services.
  - Furthermore, achieving **trust** in such large scale IoT deployments is another crucial area of interest. A distributed trust technology, ensuring scalability, privacy, and reliability, is a cornerstone for the growth of IoT and edge computing environments.

  - **Performance and server faults are not the only threats to QoS! Cyberattacks can be just as disruptive!**

Security Challenges in the Cloud-Edge Continuum

- **Dynamic and decentralized nature of Cloud-Edge Continuum deployments**
  - Services can instantly be instantiated and torn-down
  - Multitude of software platforms (Big Data, Blockchain, FaaS, ..) and hardware platforms (sensors, AR/VR, CAM On-Board Units etc.) to support modern use cases
- **Latency, availability and throughput constraints**
  - Modern use cases require guarantees for critical services
  - Cybersecurity measures can also affect performance
- **Establishing a HW and SW root-of-trust**
  - Trusted third party developers
  - Trusted infrastructures
  - Monitoring and remote attestation
- **Prevention, Detection and Threat Sharing**
  - How do you even start?
  - Lack of common understanding
    - Terminology
    - Threat analysis methodologies
  - Threat sharing
  - Etc.

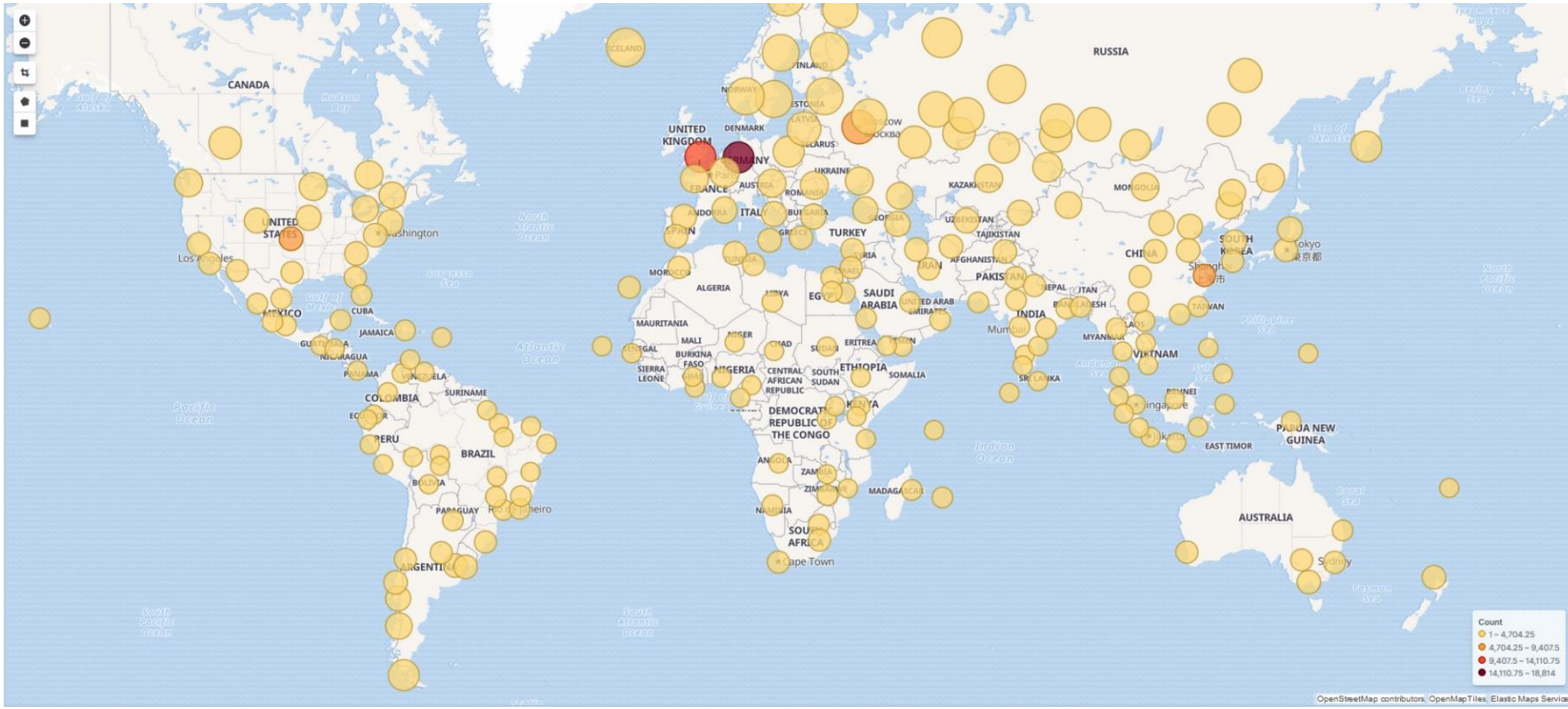
A week in the life..

Figure 1: IDS outputs.

- **During one week of running an Intrusion Detection system (single instance)**
- **Critical alerts**
  - (Network Time Protocol) NTP-based Distributed Denial of Service
  - RST Injection attack
- **Warnings for malformed packets and incomplete handshakes: more insight is needed!**

Where the traffic comes from

Figure 2: Traffic sources.



PLEDGER

Securing the Edge-Cloud Continuum

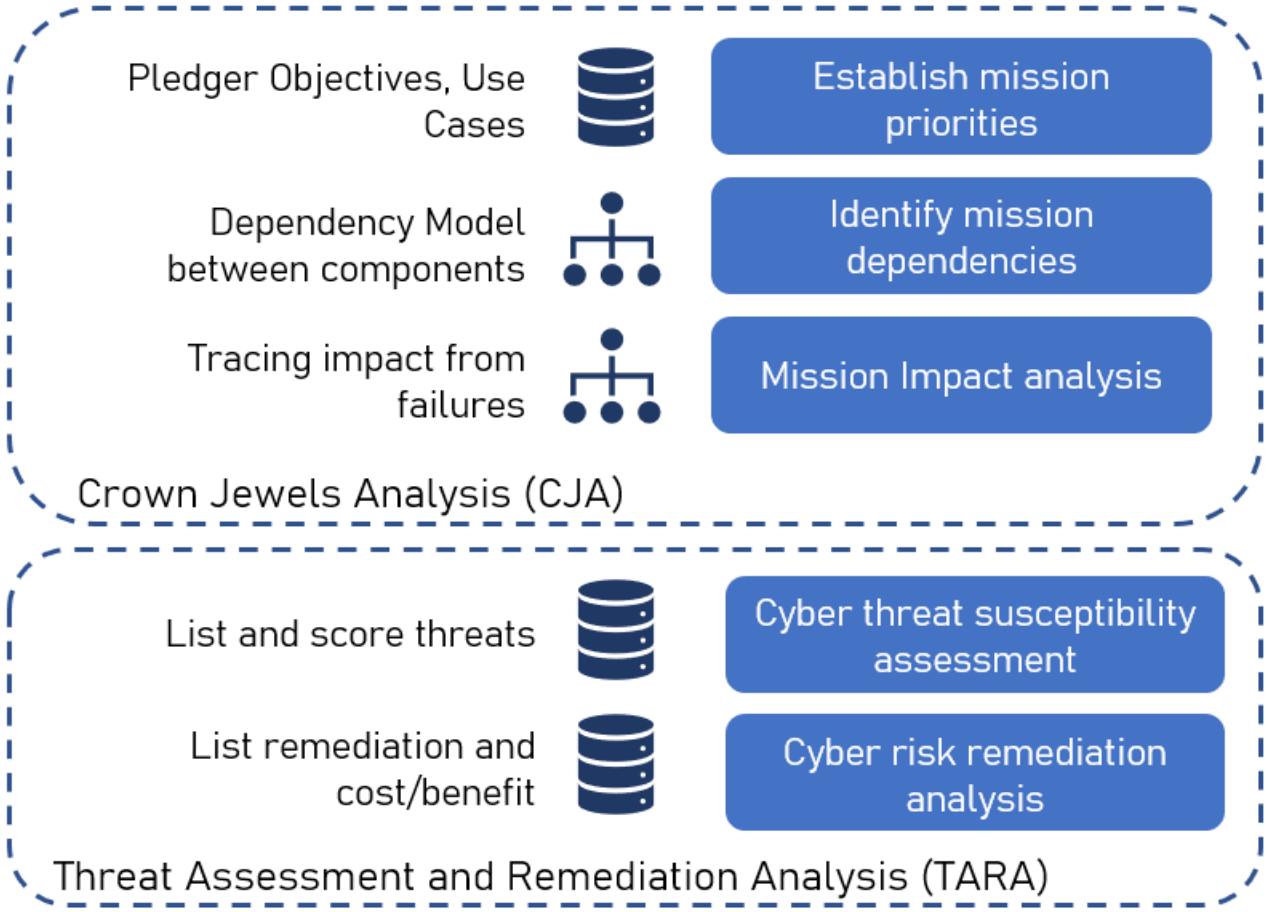
Overview of threat analysis methodology

- MITRE Mission Assurance Engineering Process
- Crown Jewel Analysis: Dependency models & Impact Tracing
- Threat assessment
- Risk Remediation

Preliminary Results

- Threats and Remediations
- Pledger Conceptual Architecture for Security

Pledger adopts and adapts MITRE's Mission Assurance Engineering Process



MAEP's aim is to ensure a system can fulfil its mission objectives even in an adversarial environment

CJA identifies dependencies and the components that should be prioritized (optional)

TARA analyses each threat and proposes mitigation measures

CJA & TARA can be used independently

Figure 3: MAEP methodology overview.

CJA: Dependency Models

Pledger Objectives, Pilot Objectives

Main Operational Tasks (Decision Support, SLA monitoring, Data Management etc)

Orchestration, Monitoring, Big Data Platform, etc. (integration matrix)

Kubernetes, Kafka, Hyperledger, ELK etc.

For each of these we can apply the Impact Tracing & Threat Analysis

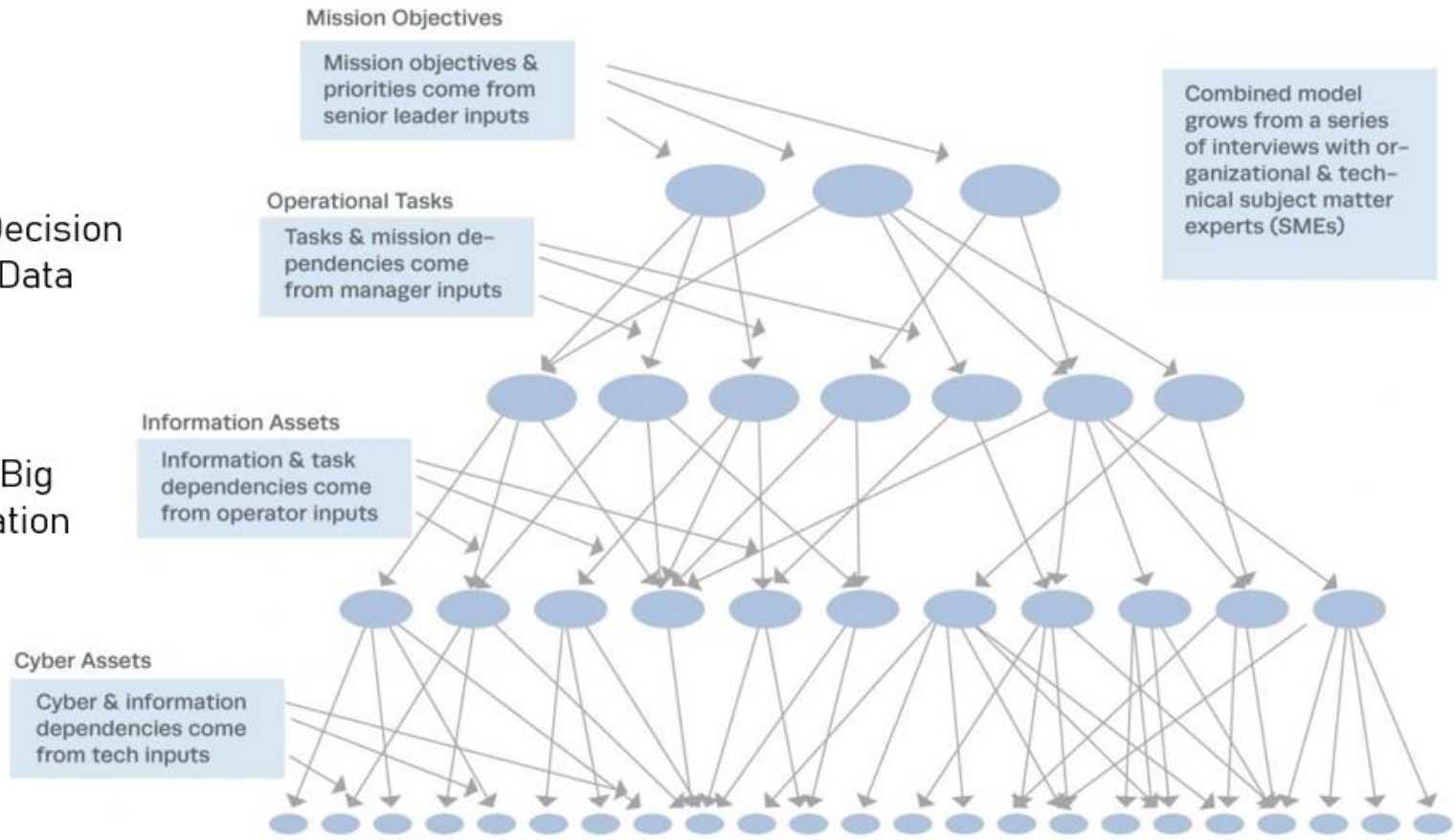


Figure 4: Dependency analysis.

CJA: Impact tracing

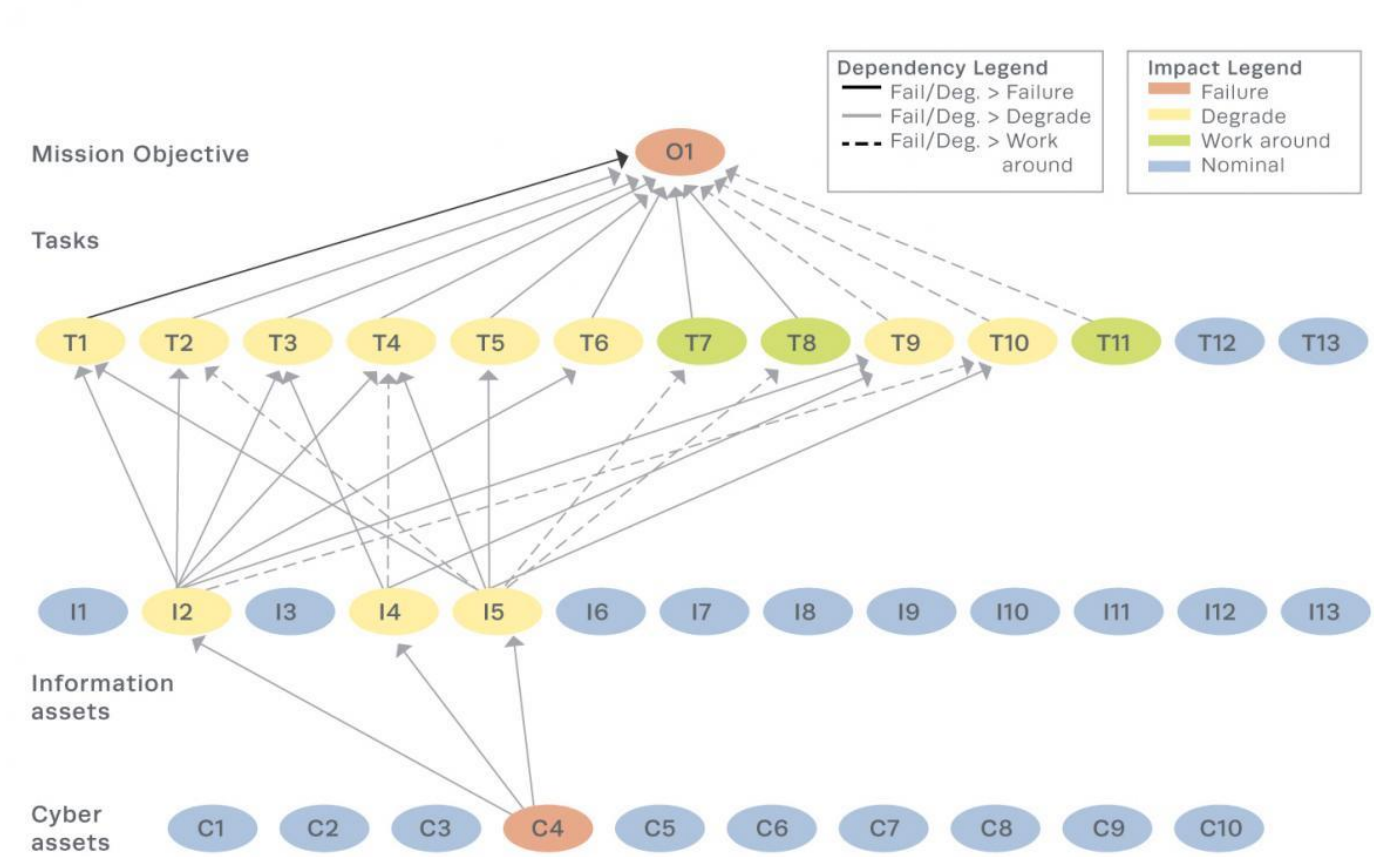


Figure 5: Impact of a failure of a component.

Threat analysis: start with a Security SotA

- Known attacks: what are the most common attacks already recorded for a specific component?
- Known/Common vulnerabilities
  - CVE vulnerabilities (Common Vulnerabilities and Exposures)
  - CVSS Common Vulnerability Scoring System
- MITRE ATT&CK threat knowledge base
- ENISA Publications , include many reports on threats and on good practices/mitigations e.g.:
  - 2020 Threat landscape report – top 15 threats
  - 2020 Threat landscape report – Sectoral/Thematic Threat analysis
  - Big Data Threat landscape
  - ENISA has many related reports and you can apply filter to the publications

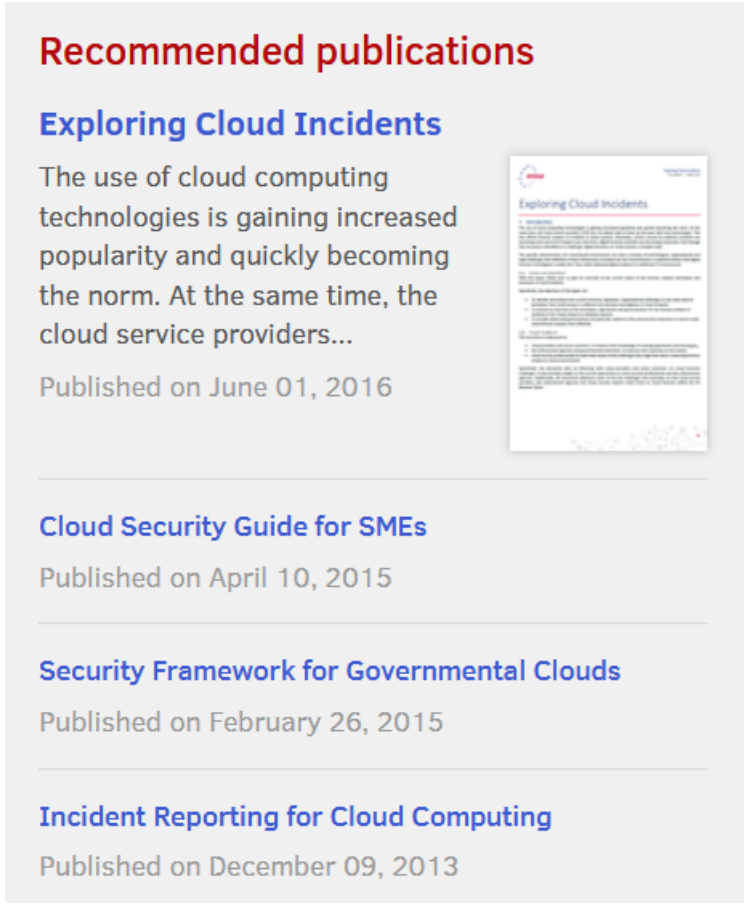


Figure 6: Sources from ENISA.

Examples of Threats and Remediation measures

- **You can end up with a very long list!**
  - **Denial of Service** against the Blockchain network and Blockchain applications
  - Using the server infrastructure as the “middleman” for a DDoS
  - **Code injection** to the secure Docker Registry (DNS Host Rebinding/Shadow Containers)
  - ..
  - Cloud Infrastructure/Service Discovery

- **Remediations ranging from simple configurations to deployments of security assets:**
- CVE tracking: for open-source tools that we use
- Traditional perimeter defenses: Firewall, IDPS
- Restrict Privileged Access, Access Control Lists to authorize applications and users
- Certificates and Secure Connections
- Disable any service that is not necessary
- …
- Machine Learning and Anomaly Detection

Prioritisation is required!

Assessing the threat criticality and the recommendations' utility versus cost

- **MITRE proposes a threat scoring algorithm**
  - What is the impact to data confidentiality, data integrity, availability etc? What is the estimated time to recover? What are the required skills of the attacker? How detectable is the attack?
  - **It is not a just question anymore of the availability of a service, but of its guaranteed QoS: Degradation of service can be as much of an issue as Denial of Service.**
  - **Our approach accepts that "In the Edge/Cloud continuum, impacts to latency, throughput etc. should be included in the scoring"**
- **Estimate the utility of the recommendations:**
  - Do they prevent, mitigate, limit or just detect the threat?
  - What is the estimated rate of success?
- **Estimate the lifecycle cost of the recommendation:**
  - Costs of acquisition: Cost to research, develop, deploy and integrate a recommendation/
  - Costs of operation: Cost to operate, train personnel, maintain and dispose

- **On-going work! To be concluded by July 2021!**

Data Gathering, Indexing, Visualisation or Decision Support, Threat Sharing	Kafka, Prometheus, ELK Stack (Elasticsearch/Logstash/Kibana), Grafana, Pledger's DSS

Security Defenses

Off-the-path defenses:
- Firewall
- Intrusion Detection (signature-based or statistical)
- Anomaly Detection

Other:
- Static or Dynamic Code Analysis
- CVE Tracking for open-source components

K8S cluster:
- security scan of the applications' base container images, lib dependencies, applications' K8S descriptors
- Cluster runtime monitoring

Infrastructure Hardening

Big Data:
- Authentication
- Authorisation
- Encryption
- Anonymisation on-the-fly

CI/CD:
- HTTPS/TLS Connections
- Certificates
- Trusted Private Registry

K8S Cluster:
- Application hardening
- Cluster hardening (registry image whitelist)
- Guidelines to build immutable app containers
- Limit node and app labels

Blockchain:
- Use **Whisper** to communicate information for the handshake between client and server nodes in the BC

PLEDGER

Thanks for your time!