H2020-SU-DS-2019

Digital Security and privacy for citizens and Small and Medium Enterprises and Micro Enterprises

CyberKit4SME

Democratizing a Cyber Security Toolkit for SMEs and MEs

Project Nº 883188

D2.1

Requirements analysis

Responsible: aurelien.lecamus@gfi.world (Gfi Informatique)

Document Reference: D2.1

Dissemination Level: Public

Version: 1.0

Date: 27/11/2020

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883188.

Executive Summary

The main goal of this deliverable is to present the toolkit and therequirements that have been identified so they can be tracked during the course of the project.
In addition to providing core guidelines for the technical developments in WP4-WP5, the requirements will aid the integration of diverse toolset in the CyberKit4SME platform. A top-down approach will be followed with respect togathering the requirements concerning the user or business perspective.
This will be complemented by a bottomup approach that will aim to identify, collect, and analyse technical requirements emerging from the technical WPs of CyberKit4SME. Requirements will be collected from the SME validation partners, and analysed to determine how they can be addressed by CyberKit4SME tools, and what this implies for the technical development activities within WP4-WP5.
The analysis will produce a measurable and unambiguous requirement set, which will be tracked against the technical developments and system integration activities during the project lifecycle in order to ensure that the CyberKit4SME tools together address all mandatory functional and non-functional requirements.

Contributors Table

DOCUMENT SECTION	AUTHOR(S)	REVIEWER(S)
ALL	Clément LASCOLS (Gfi Informatique)	Aurélien LECAMUS (Gfi Informatique) Tiberiu COCIAS (ElektroBit)

Table of Contents

Table of Figures

Table of Acronyms and Definitions

Acronym	Definition
API	Application Programming Interface
CERT	Computer Emergency Response Team
CJML	Customer Journey Modelling Language

CoI	Community of Interest
CSIRT	Computer Security Incident Response Team
CTI	Cyber Threat Intelligence
GDPR	General Data Protection Regulation
IoC	Indicator of Compromise
ISO	International Organization for Standardization
ME	Micro Entreprise
NIS	Network and Information System
SaaS	Software as a Service
SDS	Secure Data Services
SL	Service Ledger
SME	Small and Medium sized Enterprise
SSM	System Security Modeller
STIX	Structured Threat Information Expression
TAXII	Trusted Automated Exchange of Intelligence Information
TTP	Tactics, Techniques and Procedures
WP	Work Package

I. INTRODUCTION

I.1. Purpose and organization of the document

This document aims to present the solutions used in the toolkit. It includes a global description of each solution, their system requirements (both hardware and software), an explanation of how those solutions must be installed and operated and a first approach of the toolkit operability.

The document also include the requirements identified in the initial discussion with the CyberKIT4SME use case partners in order to provide core guidelines for the technical developments in the next work packages. Those requirements are to be tracked and will be modified if necessary.

I.2. Scope and audience

This document is applicable to the CyberKit4SME project until the end of the project.

This is a public document.

II. CONTEXT AND OBJECTIVES

II.1. Overview of the project

CyberKit4SME aims to democratize a kit of cyber security tools and methods enabling SMEs/MEs to: Increase awareness of cybersecurity risks, vulnerabilities and attacks; Monitor and forecast risks; Manage risks using organisational, human and technical security measures with greater confidence; and Collaborate and share information in a collective security and data protection effort. Tools developed in the project are: Semi-automated ISO 27005 threat identification and risk mitigation analysis, using a knowledge base of technical and human/organisational risk factors; Encryption and isolation tools to protect data being stored, processed or exchanged; Security information and event management, using multiple data sources for threat detection and diagnosis, Blockchain tools for SMEs/MEs to share intelligence and incident reports with supply chain partners and with CERTs.

II.2. Objectives of the project

CyberKit4SME will make its tools cheaper and more usable by SME/ME, by

1) Exploiting synergies between tools in the kit to simplify the use of each;
2) Sharing information to increase the data available for threat detection and diagnosis at each SME/ME;
3) Embedding intelligence (e.g. machine reasoning and data analytics), to fill gaps in inputs and automate tasks such as risk analysis and security configuration.

The project will also use its tools and cyber range demos to train SMEs/MEs to identify their top threats and recognise and address them with greater confidence. Results will be validated by SME/ME in four critical sectors: Finance, Health Care, Energy and Transport. Outcomes include reducing the time/cost of cyber security awareness and protection, simplifying meeting and demonstrating compliance with NIS Directive and GDPR, protecting distributed assets from cloud services to edge devices, and engaging in secure supply chains with larger organisations. The project will also collaborate with related research projects and disseminate widely in the scientific community and in SME networks.

II.3. Regulatory constraints

CyberKit4SME aims to help SMEs and MEs demonstrate compliance with any regulation that imposes requirements for cyber security or data protection. In each sector there are regulations at European level (e.g. eIDAS in Health Care) or national level (e.g. BAIT requirements in the German Finance sector). Here we focus on EU regulations that apply to all the validation sectors: the NIS Directive and the GDPR.

III. PRESENTATION OF THE TOOLKIT SOLUTIONS

This chapter presents the tools that were selected to be part of the CyberKit4SME toolkit. In each case, the tools are based on previous and ongoing research by the relevant partner. The main focus in CyberKit4SME will be to adapt the tools and models to make them accessible by SMEs & MEs without the need for scarce and expensive specialist cyber security expertise.

Five tools are presented :

- The Secure Data Services (SDS), developed by IBM
- Keenaï, developed by Gfi
- The System Security Modeller (SSM), developed by the University of Southampton
- The Customer Journey Modelling Language (CJML), developed by the SINTEF
- The Service Ledger (SL), developed by the University of Southampton

III.1. Secure Data Services

The Secure Data Services (SDS), are services that protect data across its lifecycle as it is stored, accessed and used. This includes facilities to govern data access and validation across the whole data lifecycle. The usage of the SDS in the context of a use-case is presented in Figure 1.Usage examples :

  - Database engine: data insert / delete / get / query
  - Getting data from standard sources (streams and file formats)
  - Encrypted export/import of bulk data
  - Anonymized export of bulk data
  - Deleting individual records (e.g., for GDPR)
  - Cloud backup and ransomware protection
  - Use for data and/or metadata
- Applicability in wide SME ecosystem
  - demo inside existing CyberKit4SME usecases
    - e.g. database, data import/export
  - demo as additions to existing CyberKit4SME usecases
    - e.g. cloud backup
  - demo in new usecases (modelled after existing, or other typical SME apps)
    - all features
- Properties
  - free open source technology
    - no download or maintenance charges
  - easy to use
  - (almost) transparent data security support
  - standard interfaces and formats
  - helps to leverage hybrid cloud
    - offload SME IT tasks to public cloud: less expensive, less headache

- no cloud lock-in! use / switch to any public cloud
- keep extra-sensitive data on-premises

○ cutting-edge data security mechanisms

- contribute to Cyberkit4SME Beyond-SoTA and Standards goals
- contribute to leading open source repositories
- integrate in public clouds (leveraged by various customers, including SMEs)

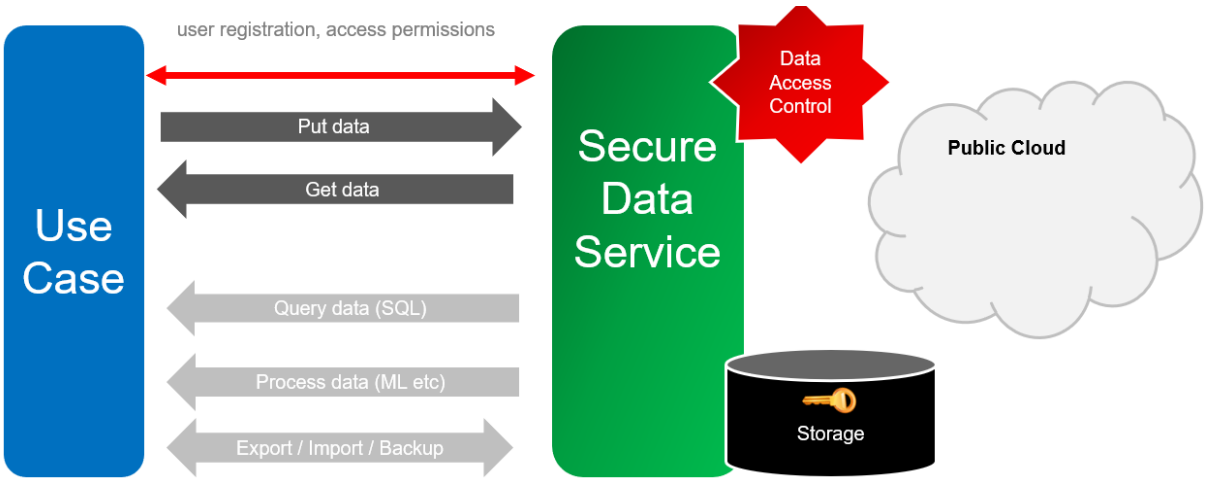

Figure 1. Presentation of the SDS

III.2. Keenaï

Keenaï is a tool that allows monitoring, through a single entry point, the whole security of the Information System. The solution is part of the SIEM's category (Security Information and Event Management).

The development of Keenaï started in 2009. It is under the responsibility of the Gfi Cybersecurity Business Unit (R&D located in Rennes, France).

Keenaï has received the French state support and security certification is in progress.

Keenaï main objectives:

- To record continuously the Information System activities,
- To detect internal and external attacks in real-time,
- To identify and reduce security threats.

The figure below (Figure 2) presents possible sources and the global functionalities of Keenaï :

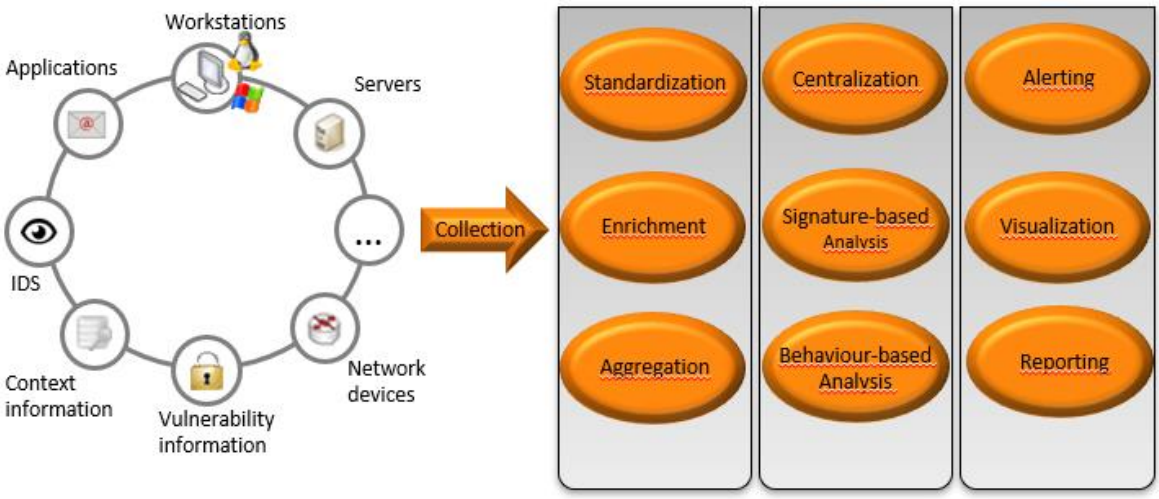

Figure 2. Sources and global functionalities

The main technologies used are :

- Based on a Big Data stack (Hadoop / Yarn)
- Apache Kafka: distributed data bus (logs, events, alerts, models, information)
- Elasticsearch: indexing and visualization of data in real time
- Apache Flink: distributed processing and analysis
- Hadoop Distributed File System (HDFS) : log storage, metric result storage for Machine Learning, …
- Apache Spark: batch processing
- Tomcat: web administration console
- MySQL: database for console data (users, profiles, rules, filters, …)
- Logstash: log standardization

III.3. System Security Modeller

The System Security Modeller (SSM) is a risk assessment tool for socio-economic systems. It combines a drag-and-drop graphical interface for drawing system models with an innovative machine-reasoning engine and detailed domain knowledgebase of threats and countermeasures to create a comprehensive view of the risks to a system and how to mitigate them.

The SSM automates much of the risk assessment procedure described in ISO 27005 and thereby supports ISO 27001 compliance. Through automation, a risk assessment is made methodical and reproducible and a security analyst may do a better job in less time.

To use the SSM the procedure is:

- Draw a model of the system, including relevant assets (networks, hosts, processes, data, people, places) and their relationships (such as which process uses what data).

- This requires an understanding of the physical/virtual infrastructure (network, hosts), the software and data used by a company and the environment (people, places).

- Identify the primary assets for the business (generally data and processes) and indicate the impact on the business that failures in those assets (such as loss of confidentiality) would cause.
  - This requires an understanding of the business.
- *The SSM then finds the threats to the system automatically using the built-in knowledgebase and through its understanding of attack-paths and threat cascades.*
  - This would normally be done (imperfectly) by a trained security analyst.
- Specify what control measures are already in place (such as passwords, firewalls, etc).
  - This requires an understanding of the information-security measures already in place.
- *The SSM then computes the risk of every threat to the system automatically.*
  - This is very hard to do by hand and would be done by a security analyst. It involves the use of the specified impacts, the inter-connectedness of the assets (to see how failures in the secondary assets affect the primary assets) and an understanding of how the controls that are in place effect the likelihood of each threat.
- Add additional controls suggested by the SSM and recompute the risk until the residual risk is acceptable for the business.
  - This requires an understanding of what information-security measures are possible for implementation.

All together the ISO 27005 risk assessment procedure is a complex process, requiring knowledge about many aspects of a business. Such a procedure has at times been considered too difficult for SMEs to manage, hence simpler descriptions of necessary security measures such as the UK's Cyber-Essentials have been introduced by regulators.

The System Security Modeller goes a long way to automating and simplifying the procedure and in Cyberkit4SME the additional needs of SMEs will be understood and addressed so that SMEs can also be supported in performing risk assessments of their systems and implementing the controls appropriate to their risks.

III.4. Customer Journey Modelling Language

Customer Journey Modelling Language (CJML) is a formal language for specification and visualisation of customer journeys and service processes.

CJML is centred around humans and human activities, regardless of their role being a customer, employee, user, or patient (as depicted in Figure 3 below).

Figure 3. Examples of customers

CJML differs from other diagrammatic languages in two principal ways:

- It models the service process from the user's point of view

- It aims at being intuitive for all users, and does not require a technical background

CJML consists of terminology, diagrams, methods and tools. The basic concepts of the language are customer journeys and touchpoints. (see Figure 4)

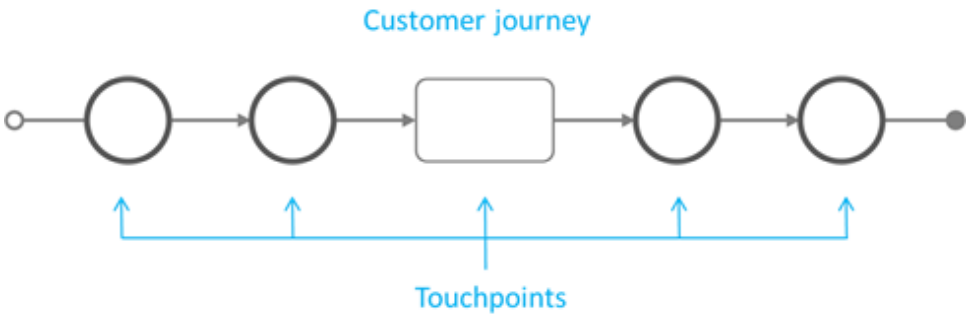

Figure 4. Customer journey and touchpoints

CJML is well suited for detailed and unambiguous modelling of user journeys and service processes that extends over time, being mediated by different communication channels.

CJML addresses the detailed interactions between

- a user and one or more service providers
- or a network of users and service providers (C2C, B2C, B2B2C etc)

CJML describes service processes in two states :

- The hypothetical state (planned journey)
- In a real context in a real context (actual journey).

Figure 5 below shows an example of those two states.

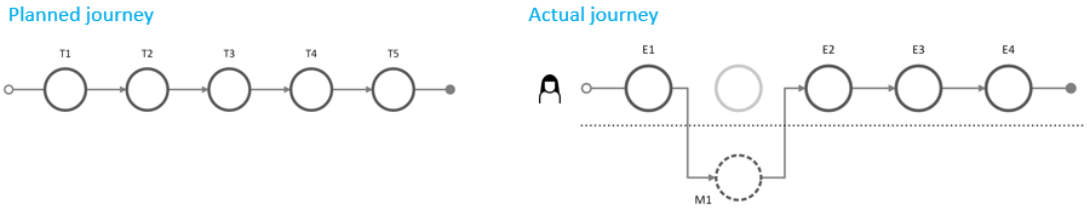

Figure 5. Example of a journey

CJML consists of terminology, diagrams, methods and tools. The basic concepts of the language are customer journeys and touchpoints.

At present, CJML is available as

- MS office templates
- stencils (Visio and OmniGraffle)
- graphical elements in bitmap format (png) or vector format (svg)

The aim is to develop a software tool for visualization and validation of CJML models

III.5. Service Ledger

The Service Ledger (SL) is a blockchain-as-a-service platform that offers programmable blockchain-enabled services that apply in several application scenarios.

Use case example : enabling the sharing of security information between different entities/organisations

- The organisations involved can share CTI information to improve the awareness and the defence against threats and malicious activities

SL platform is used for sharing CTI coming from

- the security and monitoring features of the SIEM tools
- the local authorities CERT/CSIRTs that aim to share CTI (like IoC and TTPs)
- Advantages of a solution based on blockchain
- Blockchain technology distributes *trust* and preserves *strong integrity* on shared CTI
- SL permits *privacy preserving* security information sharing allowing the deployment of *private*, *permissioned* blockchains – specific Communities of Interest (CoIs)

SL is build on top of the Algorand blockchain platform

Algorand[1] is one of the most valuable blockchain platforms. It's a *secure*, *decentralised*, and *scalable* solution

Service Ledger Main components (see Figure 6) :

- *Configuration manager*: Algorand deployer and configurer
- *Privacy manager*: policy-based privacy rules definition (to be finalised in WP5)
- *Data access control manager*: Programmable access control rules for CTI read/write operations
- *Orchestrator*: Connect the blockchain node with either a permissioned or permissionless Algorand network
- *Blockchain explorer*: user friendly dashboard for the blockchain inspection
- *Smart contract application manager*: deployer of Algorand's client applications that interact with the blockchain

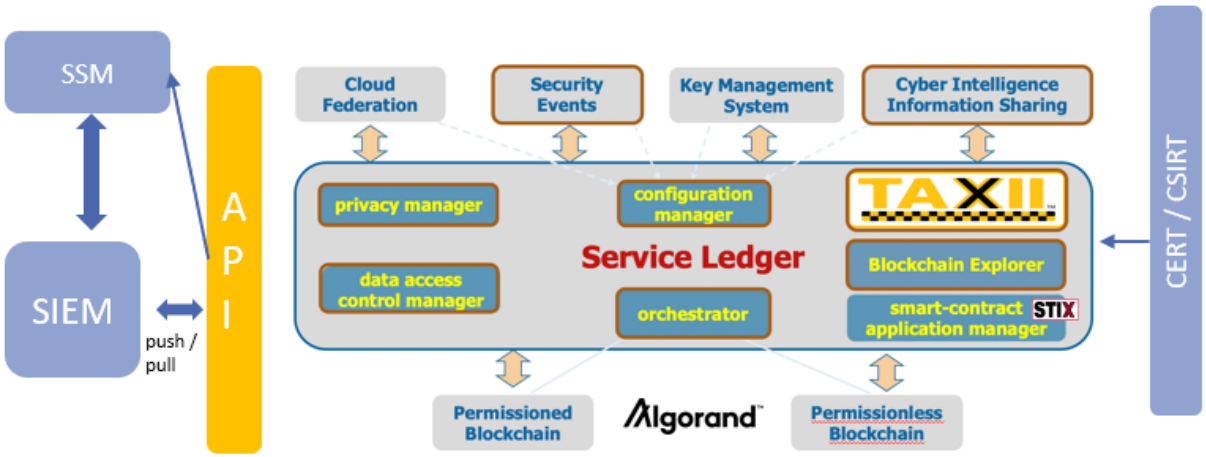

Figure 6. Presentation of the Service Ledger

- SL embeds a software module for the CTI sharing

[1] https://www.algorand.com

- Algorand's programmable smart contract able to persist and transact the blockchain CTI based on the STIX 2.0[2] standard
- TAXII 2.* protocol embedded in SL for the exchange of STIX CTI
  - CTI consumers can access CTI resources on SL over HTTPS
  - CTI producers can publish (and share) CTI resources on SL over HTTPS

- Possible usage of SL
  - An organisation take part in the sharing platform according to its privacy requirements
    - Configuration of Algorand blockchain instance and the privacy policies
  - Enable the sharing of CTI data either publicly or within a particular CoI
    - Configure the orchestrator to participate in an existing permissioned or permissionless blockchain
  - The organisation shares (or accesses) STIX data through the embedded client application by using the SL APIs – TAXII protocol

[2] https://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.html

IV. SYSTEM REQUIREMENTS

This chapter presents the system requirements for each tool presented in the previous section.

IV.1. Secure Data Services

IV.1.1. Hardware

- CPU : 4 cores
- RAM : 16 GB
- Disk : 12 GB

IV.1.2. Software

- OS : Any
- Lib : Java 8 (preferably 11)

IV.2. Keenaï

IV.2.1. Hardware

Depending on the target: number of Event Per Second, Information System topology, the time for log/event retention, high availability or not, …

A typical Keenaï instance :

- ~10 VMs/Docker Containers
- ~5 KEPS centralized
- High availability taken into consideration
- CPU : ~64 cores
- RAM : ~128 GB
- Disk : 4 TB depending on the data retention period

IV.2.2. Software

- OS : runs with Docker based on a Linux Debian 10 (Buster)
- Any host system running Docker but Linux OS are preferred

IV.3. System Security Modeller

IV.3.1. Hardware

For a small (e.g. 10) number of users and occassional use, the hardware requirements are quite light-weight and the System Security Modeller software can run on:

- CPU : 4 cores

- RAM : 8 GB
- Disk : 4 GB

IV.3.2. Software

The System Security Modeller (SSM) software is containerised using Docker. There are three containers:

- The main SSM container comprising Tomcat, the SSM software and a JenaTDB database.
- A MongoDB container for some authorisation data.
- A Keycloak container for authentication.

All three containers are Linux-based and the system is orchestrated through docker-compose.

IV.4. Customer Journey Modelling Language

IV.4.1. Hardware

Supporting operation of HTML5-enabled web browser.

The hardware needed to run CJML is not yet defined as of the writing of the present document.

IV.4.2. Software

Web browser (HTML5 support)

IV.5. Service Ledger

IV.5.1. Hardware

Minimum requirements for running a SL instance and the deployment of an Algorand participation node connected on the public main network and/or on a private network

- CPU : 6 cores
- RAM : 8 GB
- Disk : 1 TB

IV.5.2. Software

- OS : Ubuntu 18.04 LTS
- Algorand core (source code / docker image)
- Docker and docker compose

V. INSTALLATION & OPERATION

V.1. Secure Data Services

V.1.1. Installation mode

The installation procedure of the SDS is not totally defined yet, it will be presented in D5.2.

V.1.2. Operation mode

A user will interact with the SDS for :

- Database queries : data insert / delete / get
- Getting data from standard sources (streams and file formats)
- Encrypted export/import of bulk data
- Anonymized export of bulk data
- Deleting individual records (e.g., for GDPR)
- Cloud backup and ransomware protection

V.1.3. Administration mode

The installation procedure of the SDS is not totally defined yet, it will be presented in D5.2.

V.2. Keenaï

V.2.1. Installation mode

Keenaï uses Docker to facilitate the management, the orchestration and the deployment of its components.

Images are pulled from a registry or provided as archives.

The deployment and management of the Keenaï infrastructure is based on Docker Swarm and Docker compose.

A docker-compose file describes how the Keenaï multi-container docker applications should run.

Ansible is used to configure the environment through SSH before installation.

The installation of Keenaï is globally depicted in the Figure 7 below :

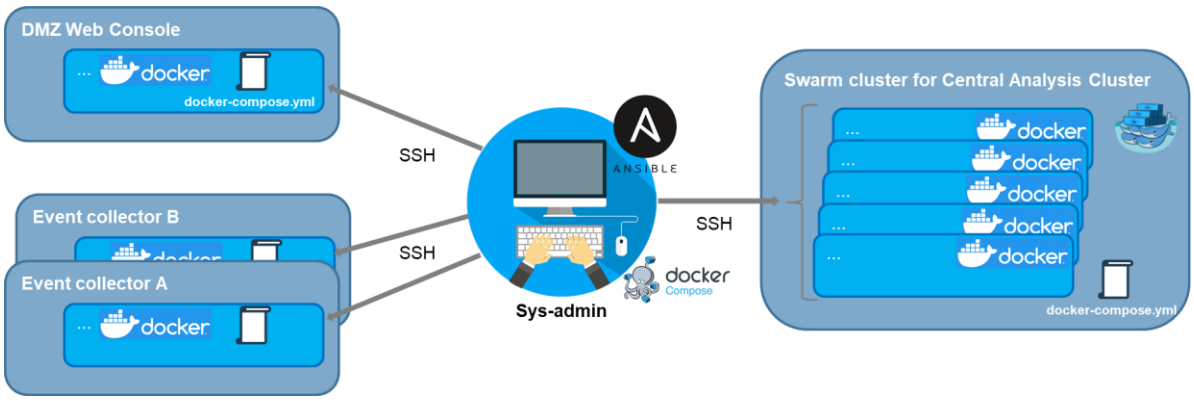

Figure 7. Keenaï installation mode

V.2.2. Operation mode

The Keenaï solution is multi-tenant and can be deployed both on SaaS or on-premise

The data collected is processed and analyzed in real-time thanks to a scalable and fully distributable architecture

The analysis is based on:

- Context addition
- Signature/Correlation engines
- Threat Intel integration
- Machine Learning unsupervised and supervised algorithms combination

The solution can be configured by the user:

- Administration and views
- Log parsers
- Correlation rule set
- …

The operation of Keenaï is depicted in the Figure 8 below :

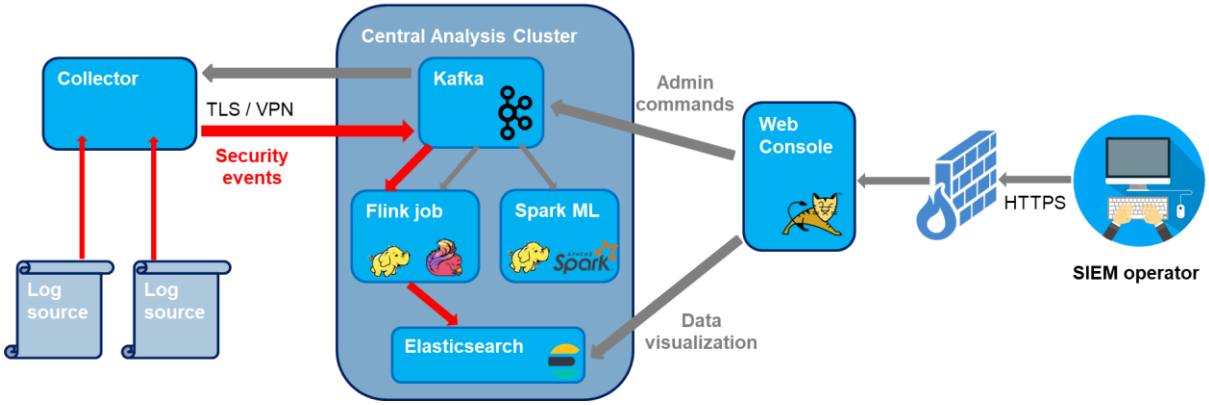

Figure 8. Keenaï operation mode

V.2.3. Administration mode

The state of the Keenaï installation can be monitored through

- Web consoles (Flink console, Hadoop Distributed File System console, …) and API
- System and application metrics (like influxdb or Grafana)

Configuration, update and maintenance operations are done, just like the installation, using Docker and Ansible.

V.3. System Security Modeller

V.3.1. Installation mode

The System Security Modeller requires a dock-compatible host system, such as one supporting docker-compose (currently supported) or Kubernetes (support to be added).

Installation via docker-compose is a simple matter of:

1. Obtaining access to the docker images either via a file transfer or authentication with a private Docker registry.
2. Executing ”docker-compose up”.

V.3.2. Operation mode

In operation, the System Security Modeller is a multi-user web application with authentication via Keycloak. A user will primarily interact with the SSM web interface but can also manage their account in the Keyclock web interface.

V.3.3. Administration mode

There are two administrative interfaces for the System Security Modeller:

1. The SSM administrative interface where updated knowledgebses can be uploaded.
2. The (standard) Keycloak administrative interface for managing user accounts. If desired, this may be configured to delegate authentication to a different identity provider such as a central SSO solution.

V.4. Customer Journey Modelling Language

V.4.1. Installation mode

CJML is a web-based application, thus it does not need to be installed.

V.4.2. Operation mode

CJML can be operated using HTML5 and JavaScript canvas.

V.4.3. Administration mode

Administration mode does not apply to CJML.

V.5. Service Ledger

V.5.1. Installation mode

Download and run SL with the embedded REST server (currently Node.js but could change)

V.5.2. Operation mode

Interaction with SL through RESTfull APIs (or web browser)

- SL functionalities
- Installation and configuration of an Algorand node / network
- Web based client application for the exchange of CTI
- Interaction with TAXII services

V.5.3. Administration mode

Access to the SL admin interface on browser, or use the SL admin console and APIs.

VI. INTEROPERABILITY

VI.1. Within the toolkit

VI.1.1. Secure Data Services

VI.1.1.a. Expected inputs

- Format : Any standard supported by Apache Spark (CSV, JSON, Parquet; Kafka)
  Or : direct API calls (application libraries) – put data
- Protocol : REST, Kafka, HL7 FHIR
- Port : configurable

VI.1.1.b. Possible outputs

- Format : Parquet, JSON
  Or: direct API calls (application libraries) – get / query data
- Protocol : REST
- Port : configurable

VI.1.2. Keenaï

VI.1.2.a. Expected inputs

Logs, events and alerts :

- Principles: the transmission mechanism in push mode from log sources is agentless and mainly based on Syslog (but not only) with the ability to secure communications
- Format: any format (raw logs, JSON, CSV, XML, multiline, …)
- Protocol:
  - Syslog (TLS, TCP or UDP)
  - on the Keenaï collector side, a Logstash agent is provided authorizing the use of tens additional protocols (Apache Kafka, file, TCP, HTTP, SNMPTRAP, S3, WMI, and many others…)
- Port: Syslog (common ports used: TLS/6514, TCP/601, UDP/514) or any other ports based on protocols accepted by Logstash

Knowledge (context, IoC, vulnerability information, …) :

- The ability to collect knowledge using flat CSV files: Indicators Of Compromises, enrichment, … where the retrieval and the deployment can be automated

- Possible extensions for communication in Cyberkit4SME:
  1. Development of a REST API in order to receive messages from other solutions (context, requests for mitigation, …) and to provide a secured access to the Keenaï management chain. Keenaï uses internaly Apache Kafka to manage the solution: a command topic (for administration, models/rules deployment, …) and an information/status topic.
  2. Development of interfaces based on HTTP(S)/REST to connect third-party tools in order to pull information

- In addition, existing integration with external solutions:
- LDAP to retrieve information about users, assets, …
- Threat Intel feeds from MISP or Palo Alto MineMeld for the managment of IoC: real-time analysis and sharing
- Vulnerability scanners (OpenVAS, Tenable) to retreive information about the Information System vulnerabilities
- Knowledge sharing on vulnerabilities (CVE referential, …) using JSON CVE format

VI.1.2.b. Possible outputs

- Alerting/Notification, a real-time module that can produce notifications with existing plugins:
  - SMTP
  - SNMP traps
  - Command/Script execution
  - HTTP(S) REST/ JSON
- The ability to process mitigation based on the notification module (use of scripts, web services, …) or extending its connectivity to existing reaction solutions
- Keenaï uses Elasticsearch as the hot storage for events and alerts: to share information, a possibility could be to provide a secure access to other tools. The solution can also easily fetch data from another Elasticsearch cluster

VI.1.3. System Security Modeller

VI.1.3.a. Expected inputs

All aspects of the System Security Modeller, including the creation and update of system models may all be done via the SSM's API[3].

VI.1.3.b. Possible outputs

The System Security Modeller can provide a description of a system model, the controls, threats and risk levels in JSON format via the API.

VI.1.4. Customer Journey Modelling Language

VI.1.4.a. Expected inputs

Predefined SVG and PNG icons

VI.1.4.b. Possible outputs

CJML diagrams (SVG and PNG format)

[3] https://software.zdmp.eu/docs/openapi/developer-tier/security-designer/openapi-3-schema.yaml/

VI.1.5. Service Ledger

VI.1.5.a. Expected inputs

- Format : Standard supported will be STIX 2.* for CTI and JSON format for the SL APIs
- Protocol : RPC call and REST over HTTPS (API calls)
- Port : configurable

VI.1.5.b. Possible outputs

- Format : Standard supported will be STIX 2.* for CTI and JSON format for the SL APIs
- Protocol : RPC call and REST over HTTPS (API calls)
- Port : configurable

VII. USE CASE RELATED REQUIREMENTS

VII.1. Functional Requirements

VII.1.1. General

SME/ME shall :

- Be able to generate a report over the past N weeks to show their clients the types and severity of threats that were seen and which we might help them with
- Be able to query the system for threats over a given period
- Have a tool to be able to monitor for and identify potential threats and mitigation strategies in the existing system

VII.1.2. Access Control

SME/ME shall :

- Have a utility to be able to manage new users when they come to the system (passwords, role-based access, access to different types of data etc.)
- Have a utility to be able to remove users when they leave (revoke access / passwords, clear up user-generated reports, etc.)

VII.1.3. Auditing

SME/ME shall :

- Have a utility to be able query the entire system to identify potential threats and mitigation on demand
- Have a utility to be able query the entire system to identify potential threats and mitigation running permanently as a background process and raising alerts if issues have been identified
- Have a standards-compliant audit tool to query access requests and denial
- Have a penetration testing process to be run on demand

VII.1.4. Interoperability

SME/ME shall :

- Be able to integrate services and / or data from different systems in accordance with relevant standards

VII.1.5. Physical Security

SME/ME shall :

- Have a utility to be able to manage physical access to sensitive areas (such as server rooms)
- Have the means to secure data in transit between servers and between servers and 3rd party devices (i.e., during remote installation)

- Have a utility to identify and secure different virtual areas of a server (i.e., secure data, algorithms, etc. to different levels according to their sensitivity)

VII.1.6. Provenance

SME/ME shall :

- Have the means to track where data have been sourced and what actions have been performed on the data since (i.e., generate a complete provenance record)
- Have the means to guarantee delivery of data / code without interference to its intended destination
- Have the means to track a user's activities during access sessions

VII.1.7. Software Development

SME/ME shall :

- Have the means to protect the development environment (including code, training / test data, etc.)
- Have the means to protect the intellectual property of their developmental environment (e.g., architecture, technical documentation)
- Have the means to check software for internal security exposures and robustness before release

VII.1.8. Software Execution

SME/ME shall :

- Have the means to secure module-to-module transfers and communication in the field (e.g., where modules in a distributed environment exchange information / data and / or communicate back to our own development environment)
- Have the means to protect software (and data) during execution

VII.1.9. Software Checking

SME/ME shall :

- Have a utility to be able to demonstrate functional support in terms of the features provided (i.e., to demonstrate that software meets the requirements agreed)
- Have a utility to be able to demonstrate functional support in terms of performance (e.g., SLA terms, KPIs for accuracy etc.)
- Have a utility to be able to check the robustness of software during design, prior to release, during deployment and during execution

VII.2. Non-functional Requirements

SME/ME shall :

- Have the means to identify potential threats from human behaviours
- Be able to be assisted by a support that is able to identify best security utilities to be used / integrated with their software
- Have the means to be able to differentiate threats to their customers and those which affect them internally

- Receive training on cybersecurity for their own engineers / employees
- Receive training on cybersecurity for their customers to be offered as a service
- Provide customers with support for how to identify and protect against threats they may encounter (e.g., to recover when problems arise)
- Provide training and support for their own employees to encourage security-aware behaviours
- Have the means to validate compliance with appropriate standards and regulations (ISO 27001, GDPR etc)
- Have a checklist (both manual and automated) for a given process, e.g., onboarding a customer; removing a customer etc. (cf. Items VII.1.2)
- Have the means to check the integrity of a deliverable (e.g., that software components are the same as those sent from our system to the customer when they are deployed at the customer location)

VIII. SECURITY REQUIREMENTS

VIII.1. Within the toolkit

- Communication between modules shall be secured
- Communication between modules shall involve a mutual authentication
- Data confidentiality shall be ensured
- Data integrity shall be ensured
- Password management shall be ensured
- User activity shall be tracked and logged
- The communication between sources and collectors (into the SIEM) shall be encrypted
- High availability shall be ensured for the vital modules of the toolkit (like the SIEM)
- A multilevel security access shall be supported

IX. SYNTHESIS OF ALL THE REQUIREMENTS

Requirements are subject to change. Those changes are available for consultation on the SharePoint of the project, in the folder Documents/Work Packages/WP2 :

WP2_CyberKIT4SME_Requirements.xlsx