D5.2

HEIR Minimum Viable Product

Project number	883275
Project acronym	HEIR
Project title	A secure Healthcare Environment for Informatics Resilience
Start date of the project	September 1st, 2020
Duration	36 months
Programme	H2020-SU-DS-2019

Deliverable type	Demonstrator
Deliverable reference no.	D5.2
Workpackage	WP5
Due date	08-2021-M12
Actual submission date	08/09/2021

Deliverable lead	ITML
Editors	George Tsakirakis (ITML), Panagiotis Rodosthenous (ITML)
Contributors	Gavrilut Dragos, Prelipcean Bogdan (BD), Iulia Ilie(SIE), Andreas Alexopoulos, Leonidas Kalipollitis (AEGIS)
Reviewers	Eftychia Lakka (FORTH), Michalis Vakalellis (AEGIS)
Dissemination level	PU
Revision	FINAL / 1.0
Keywords	MVP Demonstrator Integration

Abstract

This deliverable focuses on the continuous integration towards the realisation of the HEIR framework. The overall architecture of the MVP is described accompanied by a short description of the individual components that were developed and deployed in the MVP environment.

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author`s view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883275

Executive Summary

The deliverable D5.2 presents the work done towards the integration of the various components that form the Minimum Viable Product - MVP and the realisation of the HEIR framework. This initial prototype incorporates a limited number of components and therefore delivers a first partial functionality which will be used as a rule of thumb towards the delivery of the fully integrated platform and the release of the first and final versions of the HEIR integrated prototypes. Furthermore, this initial implementation and subsequent available testbed will greatly facilitate the development, crystallization and verification of the complete foreseen architecture and designed functionalities.

The selection of components that were incorporated for the MVP was based both on the nature and available resources of the University General Hospital of Heraklion (PAGNI) environment as well as on the identification and prioritization of the minimum mandatory number of components that were required in order to deliver the first set of functionalities and the showcasing of a concise basis to server as a preliminary proof of concept. Based on this, the monitored environment of PAGNI use case at present consists solely of computer endpoints, as a first step towards the future releases which will among others include the security monitoring of medical devices as well as the mechanism that will ensure the security of all web transactions related to patients' medical data.

The integrated components of the MVP include i) the Novel Heir Client ii) the Threat Detection Module iii) the Local Rama Score Calculator iv) the Heir Client GUI v) the Interactive Forensics module vi) the HEIR Aggregator and vii) the Observatory. All these components are briefly presented in this document and are further described in additional detail in their respective deliverables.

Table of Contents

List of Abbreviations

ACL	Access Control List
FVT	Forensics Visualisation Toolkit
GUI	Graphical User Interface
HCC	HEIR Cryptographic Checker
HCG	HEIR Client GUI
HET	HEIR Exploit Tester
HNM	HEIR Network Module
MVP	Minimum Viable Product
RAMA	Risk Assessment for Medical Applications
VPN	Virtual Private Network

List of Figures

1. Introduction

1.1 Scope and Objectives

This deliverable is linked with WP5 and Task 5.2. In general, the outcome of this WP will be an end-to-end integrated cybersecurity framework for healthcare systems with the objective to (i) design and develop the HEIR secure data fusion and management infrastructure, (ii) implement the integrated HEIR framework that realises the envisioned HEIR technology convergence and (iii) support the commercialisation activities of HEIR by releasing a stable and reliable solution for any end-to-end industrial healthcare environment.

Specifically, based on the conceptual architecture definition of Task 1.3, the technology convergence will be realised through the implementation and deployment of the HEIR framework, offering security and privacy in an end-to-end healthcare environment. Interoperability, scalability, and performance aspects are taken into consideration. The continuous software evolution guarantees a reliable technology development with the required level of integrity. In addition, part of this task is the preparation of an integration plan on how the various elements of the HEIR solution will be adapted and integrated in a common framework. This includes the MVP (M12), the 1st complete prototype (M18) and the 2nd prototype (M30). Appropriate pilots will be deployed and tested before the official release of the solution. Quality Assurance and Control procedures will be also carried out as defined in T1.3 and T7.4. Since this deliverable runs until the end of the project (M36), at present it only includes the work done so far on the development of the MVP solution.

1.2 Document Structure

This deliverable is composed out of four sections which purpose is described next:

- Section 1: Introduction of the deliverable
- Section 2: indicates the MVP use case scenarios
- Section 3: Describes the MVP architecture, deployment, integration and components
- Section 4: summarizes the progress so far and the plans for the upcoming phases of the project

1.3 Relation to other Tasks and Work Packages

Overall due to its integration role, the relation of this deliverable to the rest of the work and project tasks extends beyond the scope and tasks of WP5 and therefore is closely associated to the overall tasks and deliverables of the rest of the work packages.

More specifically this deliverable is strongly connected to i) WP2 & D2.1 “HEIR facilitators”, ii) WP3 & D3.1 “The HEIR 1st layer of services package for the MVP” iii) WP4 & D4.1 “The HEIR 2nd layer of services package for the MVP” and iv) WP6 Task 6.2. “Framework deployment and execution of real-life demonstrations”

2. MVP Use Case Scenarios

Based on the greatest feasibility of achieving the goals of the MVP as well as the available resources, PAGNIwas deemed as the best candidate to host the MVP environment.

As described further in D2.1 “HEIR facilitators”, D3.1 “The HEIR 1st layer of services package for the MVP”, D4.1 “The HEIR 2nd layer of services package for the MVP” the defined use case is as follows

- The HEIR Client in PAGNI generates events and RAMA Score.
- Events are anonymized and sent to the HEIR Database.
- RAMA Score is sent to the HEIR Database.
- the 1st layer of visualizations fetch data and present them in the GUI including the Local RAMA score.
- Global benchmark is calculated.
- 2nd layer visualisations fetch data from the HEIR Database and present them in the UI.
- Anonymous Security Stakeholders can access statistical data and view aggregated, global information about RAMA Scores and cybersecurity events.

In addition to the above, the HEIR Interactive Forensics module was also deployed on the PAGNI Virtual Machine (VM) client, based on which analytical detailed information about the captured security events and relevant system information is displayed via the HEIR Interactive Forensics GUI. At present this source of security events and information is not yet included in the RAMA score calculation and will be incorporated as a module of the HEIR Client for the upcoming release (M18).

Pending appropriate clearance from the IT department of PAGNI to install the HEIR client in a number of production workstations belonging to various departments, the collected data for the MVP use case derive from a single client workstation (VM).

3. MVP Architecture

3.1 Architecture Overview

The architecture overview and module connections are depicted in Figure 1 HEIR Architecture overviewFigure 1

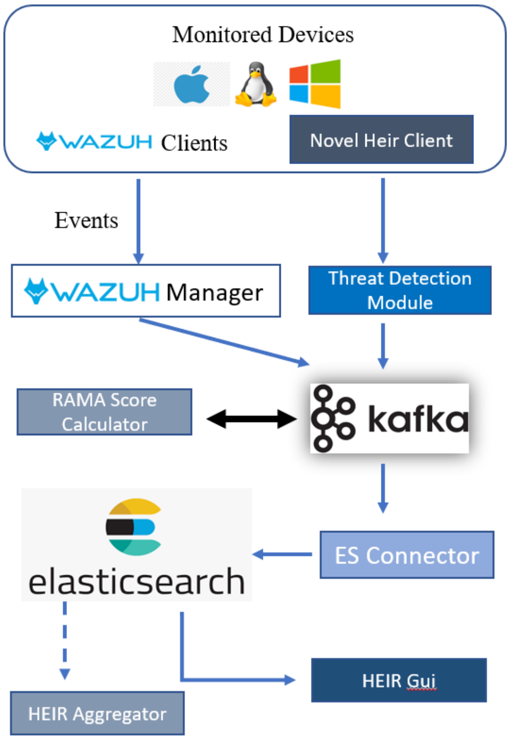


Figure 1 HEIR Architecture overview

Apart from the individual HEIR components that will be further described in their respective section, the core components of the architecture are:

- The lightweight Wazuh[1] agents monitor the devices and report to the Wazuh manager.
- The Wazuh manager analyses the reported information and produces categorized events which are passed to the Kafka[2] broker.
- The Kafka broker offers a highly available, horizontally scalable messaging broker for the whole platform. The broker is used both for the Wazuh data pipeline and for the other components.
- The Kafka to Elasticsearch[3] connector (ES connector), which consumes data from the Kafka cluster, edits and aggregates appropriately and saves the output to Elasticsearch. As far as the Wazuh data flow is concerned, the ES connector parses the events produced by the Wazuh manager and groups the logs so that multitenancy on the data can be achieved.
- A highly available Elasticsearch cluster provided for storage.

[1] https://wazuh.com/
[2] https://kafka.apache.org/
[3] https://www.elastic.co/

The architecture is designed to be robust, highly available and achieve high throughput. With these goals in mind all components apart from the agents are deployed in Kubernetes[4]. The Wazuh manager is deployed as a cluster, so that traffic spikes can be met by scaling of the managers. The Kafka cluster follows the same principles and any need for further bandwidth augmentation is met by increasing the number of operating brokers in the cluster. The deployment includes SASL_SSL authentication and authorization. Each entity interacting with the Kafka cluster is identified by individual certificates and authorized per action, per topic and per group. The authorization is managed by Access Control List (ACL).

The ES connector is designed to be stateless and efficient. If traffic should be consumed faster, the Kafka topics partitions can be increased, and the number of connectors can scale to match them and consume concurrently the events.

The Elasticsearch is deployed in Kubernetes using the best practices and offering the best of both technologies.

3.2 Deployment

For the deployment of the MVP environment PAGNI provided the following resources.

- A single VM (32 Vcores, 32GB RAM, 100GB NVMe storage.)
- A WIN10 VM (Single Core,4GB RAM, 40 GB Storage)

All connections to the environment are done via Virtual Private Network (VPN), for which all participating partners have been issued individual VPN profiles. In addition, all partners are given their own accounts in the main VM and are able to setup their components as dockerized images that are managed via Kubernetes. All deployed components run as containerised services with the exception of the HEIR Client binary which runs exclusively on the client workstation and communicates its results to the Kafka message broker.

The current Kubernetes core Deployments, are:

1. Es Connector: The es-connector is a Spring Boot[5] application consuming logs from the Kafka cluster and saves them to Elasticsearch offering multitenancy on the logs on the index level (e.g., the logs of group 1 are saved in index agents_1-YYYY.MM.dd). It is deployed by plain Kubernetes artifacts in the single VM but can be scaled if more VMs are added.
2. Elasticsearch is deployed and operates using the Elastic's team instructions.
3. Kibana[6] is deployed to provide visualizations on the data reaching Elasticsearch.
4. A Kafka cluster is deployed using custom Kubernetes artifacts and a private Certificate Authority to issue certificates. The Kafka operations are managed using a custom Spring Boot application which wraps often used actions in a convenient REST API.
5. Wazuh is deployed as a cluster using Kubernetes artifacts provided by the Wazuh team and customized to fit the purposes of the current deployment.

[4] https://kubernetes.io/
[5] https://spring.io/projects/spring-boot
[6] https://www.elastic.co/what-is/kibana

The generated ACL entries up until now are depicted in the table below:

Topic	Principal	Access
RamaToHeirGUI	SPHYNX	READ/WRITE
RamaToHeirGUI	AEGIS	READ
HeirClientToRama	SPHYNX	READ/WRITE
HeirClientToRama	BDEFENDER	READ/WRITE
HeirClientToRama	MVP	READ/WRITE
HeirClientToRama	SIEMENS	READ/WRITE

3.3 MVP Components

The MVP components are hereby described in their respective sections.

3.3.1 Novel HEIR Client

The HEIR Client collects and processes information, either on the endpoint level or centralized level. The architecture of the HEIR client is modular and can plug in several analysis components (HEIR Network Module, HEIR Cryptographic Checker, HEIR Exploit Tester, Risk Analytics, Vulnerability Assessment). For the MVP context, the Vulnerability Assessment and HEIR Exploit Tester module are used. The HEIR Client architecture is depicted in Figure 2.

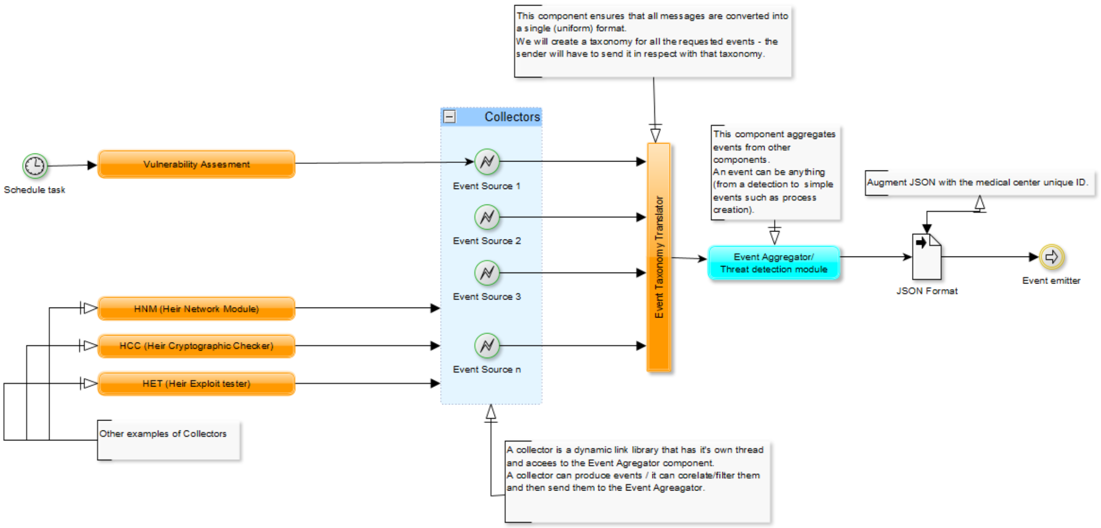

This component ensures that all messages are converted into a single (uniform) format.
We will create a taxonomy for all the reques ted events - the sender will have to send it in respect with that taxonomy.
Collectors
Schedule task
Vulnerability Assesment
Event Source 1
Event Source 2
Event Source 3
Event Source n
HNM (Heir Network Module)
HCC (Heir Cryptographic Checker)
HET (Heir Exploit tester)
Other examples of Collectors
Event Taxonomy Translator
This component aggregates events from other components.
An event can be anything (from a detection to simple events such as process creation).
Event Aggregator/ Threat detection module
Augment JSON with the medical center unique ID.
JSON Format
Event emitter
A collector is a dynamic link library that has it's own thread and accees to the Event Agregator component.
A collector can produce events / it can corelate/filter them and then send them to the Event Agreagator.

Figure 2 HEIR Client Architecture

The modules also named Collectors will send events with information about threats, security metrics, risks. This information will also be received from the HEIR Agent of the facilitators package. The events are normalized and converted to a single and uniform format by the Event Taxonomy Translator component. The normalized events from different modules are aggregated by the Event Aggregator component which also correlates and augments the events and then the events are further emitted to the other components. The external components are the SIEM module and RAMA score calculator. The events are submitted to the other modules (ex: RAMA Score Calculator) by event publishing to Kafka Message Broker. The other modules will subscribe to the topics of interest.

For the MVP the provided modules are the HEIR Exploit Tester (HET) and the Vulnerability Assessment module which provides information about:

- system misconfiguration on the endpoint system
- application vulnerability assessment

3.3.2 Threat Detection Module

The threat detection module is a central module that can centralize and correlate the information regarding threats from the other components (the threat detection components from the facilitators or the HEIR Exploit Tester module. The architecture of the module is mainly similar with the internal architecture of the HEIR Client with the focus on the Event Aggregator and Augmentation component as depicted in Figure 3.

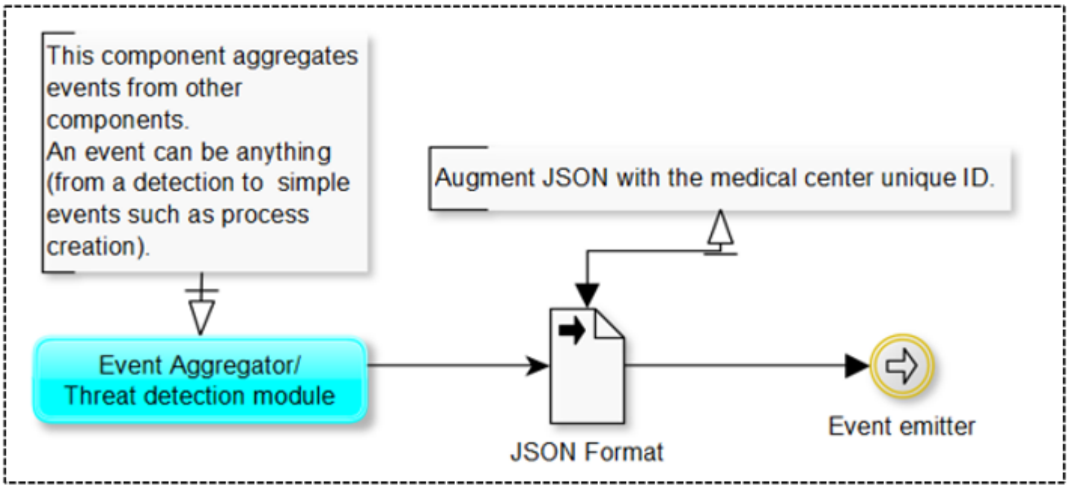

This component aggregates events from other components.
An event can be anything (from a detection to simple events such as process creation).
Augment JSON with the medical center unique ID.
Event Aggregator/ Threat detection module
JSON Format
Event emitter

Figure 3 Threat Detection Module – High-level architecture

The events are submitted to the other modules (e.g., the RAMA Score Calculator) by event publishing to Kafka Message Broker. The other modules will subscribe to the topics of interest

3.3.3 Local Rama Score Calculator

The Local RAMA Score Calculator (part of the 1st layer of services package) score acts as a benchmark for the IT security of a hospital or healthcare facility. It is responsible for estimating the attack surface and resilience of the medical devices by incorporating several critical issues in a live manner. To calculate the score, the RAMA Score Calculator receives aggregated input from several HEIR components, through the HEIR Client. For the MVP, the Local RAMA Score calculator will receive input from the Vulnerability Assessment Module and HET. To do so, the local RAMA Score Calculator subscribes to the "HeirClientToRama" Kafka topic, receives the aggregated events of the above-mentioned components, calculates the Local RAMA Score, and metadata and continuously provides the output to the 'RamaToHeirGUI'. Figure 4 presents the component's high-level architecture. More information on this component and the communication/integration plan of the 1st layer of services packages' components is available in "D3.1- The HEIR 1st layer of services package for the MVP".

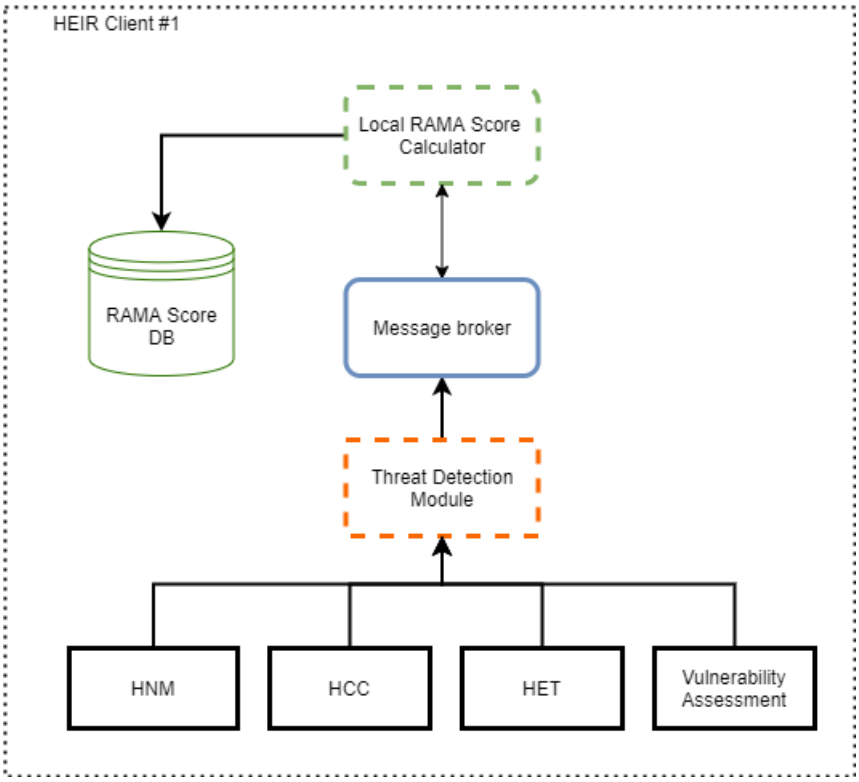


Figure 4 Local RAMA Score Calculator high-level architecture

3.3.4 HEIR Interactive Forensics

The HEIR SIEM is based on the Wazuh open-source solution which provides a multitude of security related services that continuously monitor an IT infrastructure. All data is collected by lightweight agents which run on the monitored systems, collecting events, and forwarding them to the Wazuh Manager, where data is aggregated, analysed, indexed, and stored. This ensures that the resources needed at the client level is kept to a minimum since the security intelligence and data analysis is solely performed at the server level. Wazuh clients run on many different platforms, including Windows, Linux, Mac OS X, AIX, Solaris and HP-UX.

The events reported by the Wazuh agents are the outcome of a wide range of tasks such as

- Inventory of running processes and installed applications
- Log and events data collection
- File and registry keys integrity monitoring
- Monitoring of open ports and network configuration
- Configuration assessment and policy monitoring

These events are received by the Wazuh server and processed through a toolset of decoders and rules, using threat intelligence to look for well-known Indicators Of Compromises (IOCs). As a result of this analysis, all events are appointed a severity level enabling the administrators to focus on the crucial issues that need to be addressed. This is further delivered via customized alerts that are sent to an Elastic Stack[7] which also provides a powerful interface for data visualization and analysis via its integration with Kibana.

The HEIR SIEM is solely responsible for collecting all the information portrayed in the Forensics Visualisation Toolkit (FVT) GUI.

The Interactive Forensics module is based on AEGIS' FVT to display analytical information about the captured security events and relevant system information. The module allows users to drill down to individual assets monitored by HEIR and see monitoring metrics as well as events captured by the SIEM in order to gain situational awareness at a short time. The tool's internal communication flow follows the same pattern as in the HEIR Client GUI (HCG) (Figure 5). However different data sources are interrogated, and the visualization elements serve a different purpose as analysed in "D2.1 The HEIR facilitators package: MVP".

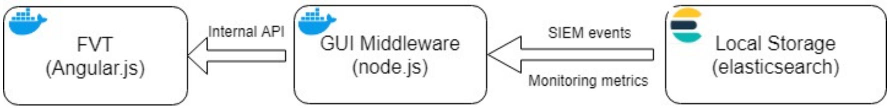
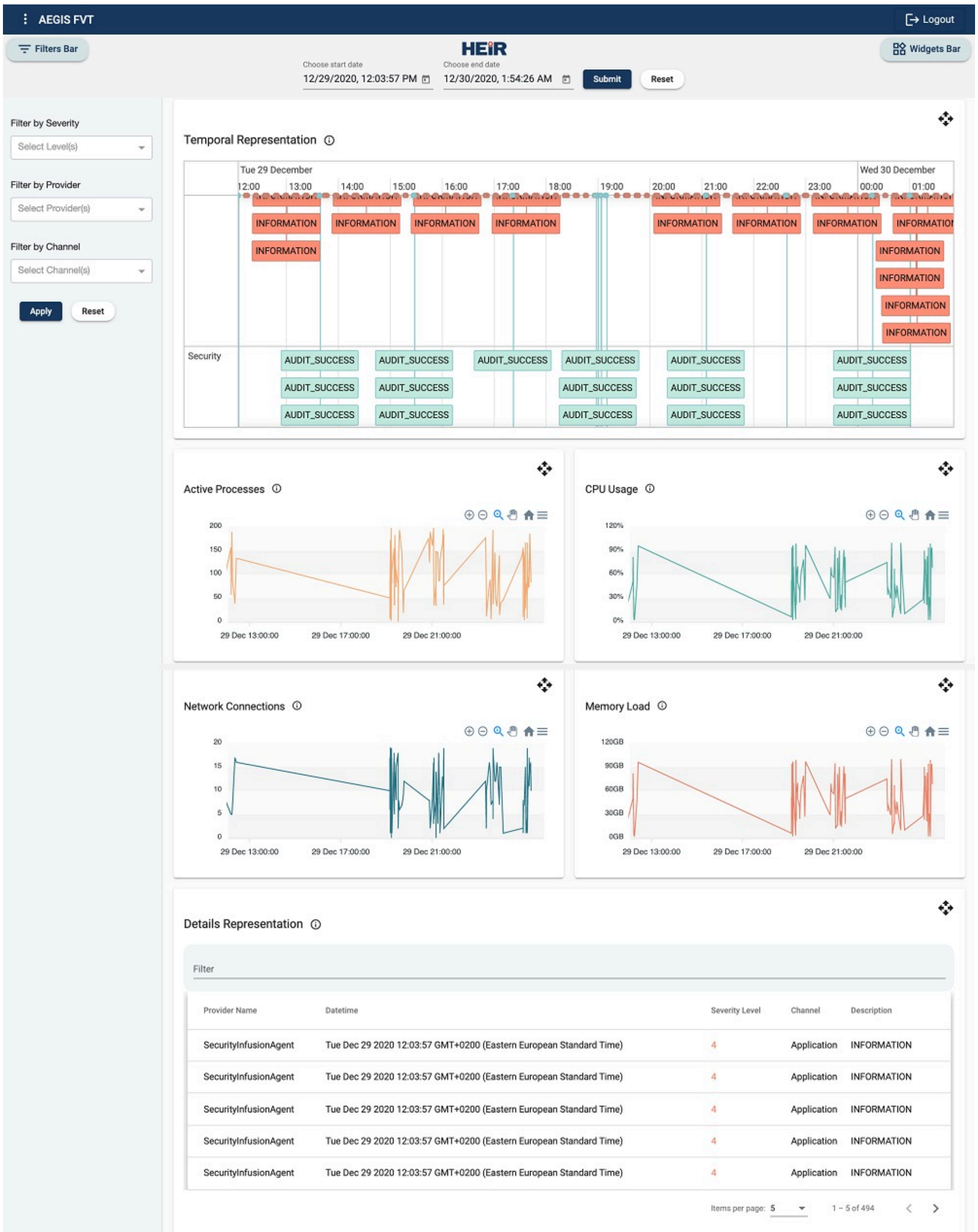


Figure 5 FVT internal communication flow

The main screen of FVT is presented in Figure 6.

[7] https://www.elastic.co/elastic-stack/



Choose start date: 12/29/2020, 12:03:57 PM

Choose end date: 12/30/2020, 1:54:26 AM

Provider Name	Datetime	Severity Level	Channel	Description
SecurityInfusionAgent	Tue Dec 29 2020 12:03:57 GMT+0200 (Eastern European Standard Time)	4	Application	INFORMATION
SecurityInfusionAgent	Tue Dec 29 2020 12:03:57 GMT+0200 (Eastern European Standard Time)	4	Application	INFORMATION
SecurityInfusionAgent	Tue Dec 29 2020 12:03:57 GMT+0200 (Eastern European Standard Time)	4	Application	INFORMATION
SecurityInfusionAgent	Tue Dec 29 2020 12:03:57 GMT+0200 (Eastern European Standard Time)	4	Application	INFORMATION
SecurityInfusionAgent	Tue Dec 29 2020 12:03:57 GMT+0200 (Eastern European Standard Time)	4	Application	INFORMATION

Items per page: 5   1 – 5 of 494

Figure 6 FVT Main page

3.3.5 HEIR Client GUI

The HCG includes visualizations of information generated by the 1st level services running inside the hospital environment, i.e., PAGNI's infrastructure in the case of MVP. HCG runs as a containerised service which fetches data found in the local storage facility of the MVP deployment via the accompanying GUI Middleware which serves the internal API required by HCG to communicate with other components as seen in the following Figure 7.

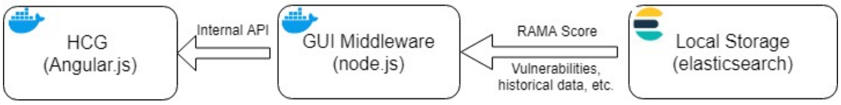
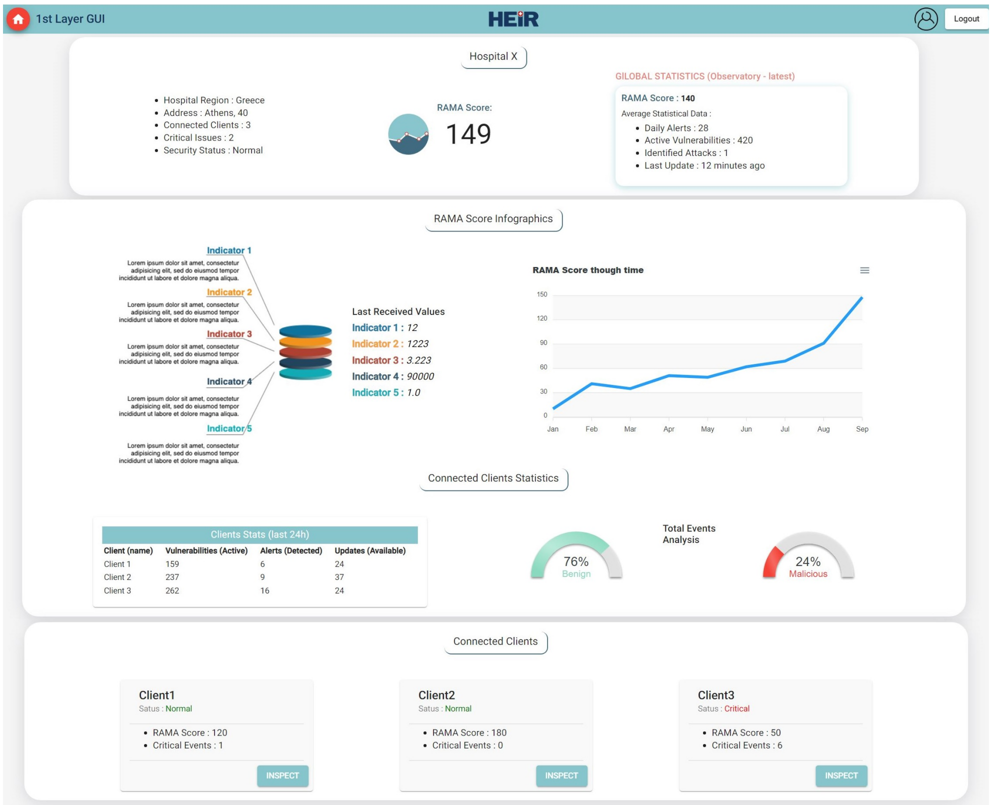


Figure 7 HCG Internal Communication Flow

The displayed data includes RAMA Score and associated metadata generated by the HEIR Client. HCG also links to the interactive Forensics module as presented in the next paragraph. The MVP version of HCG is analytically described in “D3.1 The HEIR 1st layer of services package for the MVP”. Figure 8 depicts the main screen of the GUI. Users can drill down to further security information by clicking Inspect Client, which navigated them to the HEIR Interactive Forensics GUI.



- Hospital Region : Greece
- Address : Athens, 40
- Connected Clients : 3
- Critical Issues : 2
- Security Status : Normal

RAMA Score:
149

GILOBAL STATISTICS (Observatory - latest)

RAMA Score : 140

Average Statistical Data :

- Daily Alerts : 28
- Active Vulnerabilities : 420
- Identified Attacks : 1
- Last Update : 12 minutes ago

Indicator 1
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Indicator 2
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Indicator 3
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Indicator 4
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Indicator 5
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Last Received Values
Indicator 1 : *12*
Indicator 2 : *1223*
Indicator 3 : *3.223*
Indicator 4 : *90000*
Indicator 5 : *1.0*

Clients Stats (last 24h)

Client (name)	Vulnerabilities (Active)	Alerts (Detected)	Updates (Available)
Client 1	159	6	24
Client 2	237	9	37
Client 3	262	16	24

Client1
Satus : Normal

- RAMA Score : 120
- Critical Events : 1

Client2
Satus : Normal

- RAMA Score : 180
- Critical Events : 0

Client3
Satus : Critical

- RAMA Score : 50
- Critical Events : 6

Figure 8 HCG Main page

3.3.6 HEIR Aggregator

The HEIR Aggregator is a component designed for health institution running multiple independent departments for which a global cybersecurity status is required.

The Aggregator collects and compiles statistical information on possible events or vulnerabilities discovered by the HEIR clients for the independent departments. An aggregate local RAMA score is also computed after having been provided with multiple local RAMA scores by the HEIR clients deployed on the individual departments.

In the current version of the MVP deployed in the PAGNI environment, the Aggregator receives input from a single client and runs on the same machine as the HEIR client in a containerized environment.

However, for the following iterations, the Aggregator will be deployed on an independent machine from those where the HEIR clients run and receive input from the multiple HEIR clients.

The HEIR Aggregator input is given by JSON files written by the Kafka connector to an Elasticsearch storage in the "rama-heir-gui" index.

The HEIR Aggregator is triggered based on a user-defined schedule (e.g., hourly), read the most recent outputs from the HEIR clients in the Elasticsearch storage, compute the aggregates, and write the aggregated values for RAMA and the event statistics to the Elasticsearch storage, where they can be accessed by the HEIR GUI. In the following iterations of the MVP, the Aggregator will also send its output to the HEIR Observatory database.

The process of compiling the cybersecurity status from the HEIR clients deployed in one health institution is shown below in Figure 9.

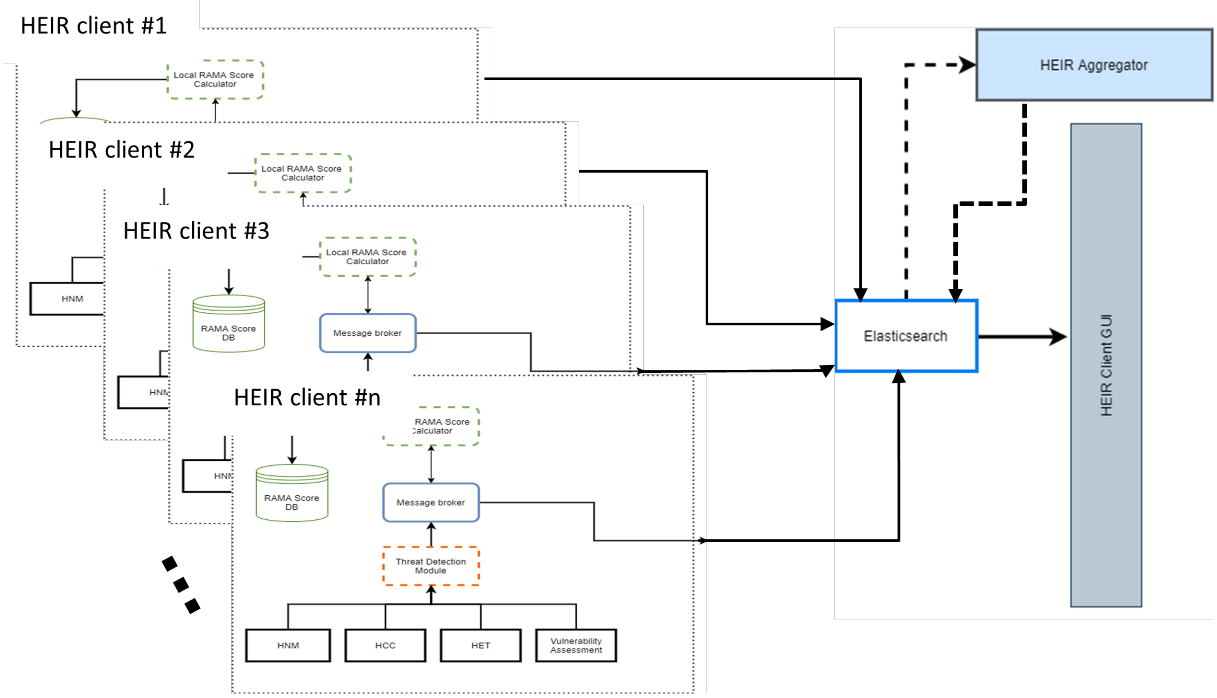


Figure 9 HEIR Aggregator

3.3.7 HEIR Observatory

The HEIR Observatory is responsible to collect, analyse and present the results of all the deployed HEIR Clients in order to provide global insights on the level of security in healthcare environments. For the MVP, the Observatory pulls data from the single HEIR Client of PAGNI using the deployment and communication mechanism followed by the other visualisation components (HCG and FVT) as seen in Figure 10.

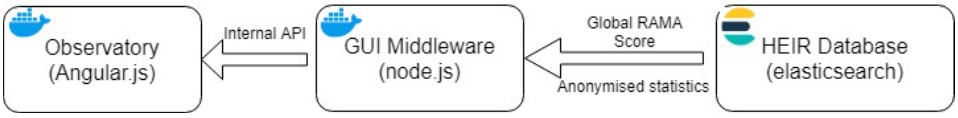
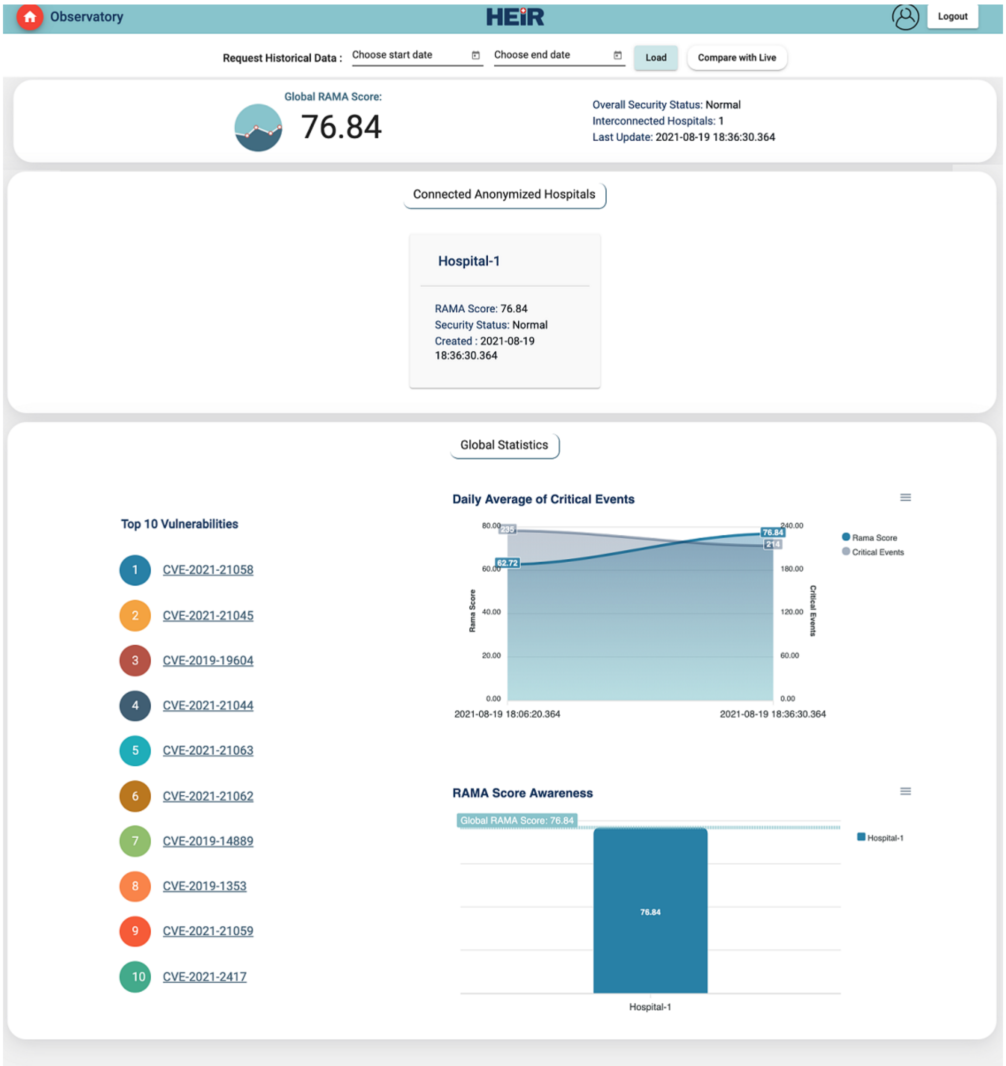


Figure 10 Observatory internal communication flow

The main page of the Observatory displays statistics of the Global RAMA Score and the historical evolution of data like the captured events or detected vulnerabilities. “D4.1 The HEIR 2nd layer of services package for the MVP” describes the complete functionality of the Observatory in details. Figure 11 below illustrates the main page of the Observatory.



Global RAMA Score:

76.84

Overall Security Status: Normal
Interconnected Hospitals: 1
Last Update: 2021-08-19 18:36:30.364

Connected Anonymized Hospitals

Hospital-1

RAMA Score: 76.84
Security Status: Normal
Created : 2021-08-19 18:36:30.364

Global Statistics

Top 10 Vulnerabilities

1. CVE-2021-21058
2. CVE-2021-21045
3. CVE-2019-19604
4. CVE-2021-21044
5. CVE-2021-21063
6. CVE-2021-21062
7. CVE-2019-14889
8. CVE-2019-1353
9. CVE-2021-21059
10. CVE-2021-2417

Figure 11 Observatory Main page

4. Conclusion

This deliverable presents the work done towards the integration of the various components that form the MVP and the realisation of the HEIR framework. This initial prototype incorporates a limited number of components and therefore delivers a first partial functionality which will be used as a rule of thumb towards the delivery of the fully integrated platform and the release of the first and final versions of the HEIR integrated prototypes. Furthermore, this initial implementation and subsequent available testbed will greatly facilitate the development, crystallization and verification of the complete foreseen architecture and designed functionalities.

Following this MVP release, two further releases of the platform will be delivered, the first complete version in M18 and the final version of 2nd layer of services package in M26 of the project.