Horizon 2020
European Union funding
for Research & Innovation

Cyber Security PPP: Addressing Advanced Cyber Security Threats and Threat Actors

Cyber Security Threats and Threat Actors Training - Assurance Driven Multi- Layer, end-to-end Simulation and Training

D6.1: Initial Prototype of Integrated THREAT-ARREST Platform†

Abstract: This document presents the first version of the integrated THREAT-ARREST platform. The THREAT-ARREST training is offered as a service to organizations through a Web-based GUI. The first version of the platform is released along three full-fledged training scenarios for Smart Energy, Smart Transportation, and Healthcare, each addressing trainees of different knowledge and skills. Credentials to access the platform are provided in the document to facilitate demonstration and validation in pilot activities. The first version successfully integrates and orchestrates the various training capabilities such as emulation, simulation, gamification, visualization, user scoring, and CTTP model creation.

† *The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 786890.*

Contractual Date of Delivery	30/04/2020
Actual Date of Delivery	30/04/2020
Deliverable Security Class	Public
Editor	Hristo Koshutanski (ATOS)
Contributors	Fulvio Frati (UMIL), Torsten Hildebrandt (SIMPLAN), George Hatzivasilis (FORTH), K. Fysarakis, M. Smyrlis, G. Spanoudakis (STS), Oleg Blinder (IBM), Ludger Goeke, Sebastian Pape (SEA), George Leftheriotis (TUV), George Tsakirakis, George Bravos (ITML), Martin Kunc (CZNIC).
Quality Assurance	George Hatzivasilis (FORTH), Dirk Wortmann (SIMPLAN).

The *THREAT-ARREST* Consortium

Foundation for Research and Technology – Hellas (FORTH)	Greece
SIMPLAN AG (SIMPLAN)	Germany
Sphynx Technology Solutions (STS)	Switzerland
Universita Degli Studi di Milano (UMIL)	Italy
ATOS Spain S.A. (ATOS)	Spain
IBM Israel – Science and Technology LTD (IBM)	Israel
Social Engineering Academy GMBH (SEA)	Germany
Information Technology for Market Leadership (ITML)	Greece
Bird & Bird LLP (B&B)	United Kingdom
Technische Universitaet Braunschweig (TUBS)	Germany
CZ.NIC, ZSPO (CZNIC)	Czech Republic
DANAOS Shipping Company LTD (DANAOS)	Cyprus
TUV HELLAS TUV NORD (TUV)	Greece
LIGHTSOURCE LAB LTD (LSE)	Ireland
Agenzia Regionale Strategica per la Salute ed il Sociale (ARESS)	Italy

Document Revisions & Quality Assurance

Internal Reviewers

1. *George Hatzivasilis (FORTH),*
2. *Dirk Wortmann (SIMPLAN).*

Revisions

Version	Date	By	Overview
1.0	30/04/2020	Editor	Revision based on PCC & PTC Review comments
0.9	29/04/2020	Editor	Revision based on quality review comments.
0.8	22/04/2020	CZNIC	CZNIC contributed to Section 3.1
0.7	17/04/2020	CZNIC	CZNIC contributed to Sections 3.2, 3.5
0.6	16/04/2020	Editor	ATOS contributed to Sections 1, 2.1, 2.2, 3.3, 4, and 5. It also includes revision of partners' contributions.
0.5	14/04/2020	SEA	SEA contributed to Sections 2.3.2 and 4.
0.4	06/04/2020	TUV, SIMPLAN	TUV contributed to Section 3.3, SIMPLAN contributed to Sections 2.3.4 and 4.
0.3	02/04/2020	ITML, IBM	ITML contributed to Section 4, IBM contributed to Sections 2.3.5 and 2.3.6.
0.2	31/03/2020	UMIL, STS, FORTH	UMIL contributed to Sections 2.3.3 and 4, STS contributed to Section 2.3.1 and 4, FORTH revised the structure of Section 3 and contributed to Sections 3.4 and 4.
0.1	25/03/2020	Editor	First Draft

Executive Summary

This deliverable presents the first version of the integrated THREAT-ARREST platform. The platform is integrated and deployed on a bare metal server at LeaseWeb.com. The THREAT-ARREST training is offered as a service to organizations through a Web-based GUI. The first version successfully integrates and orchestrates the various training capabilities such as cyber system emulation, simulation, gamification, user assessment and scoring, and CTTP model editor.

Importantly, the first version of the platform is released with three full-fledged training scenarios for the different project use cases – Smart Energy (smart home & IoT), Smart Transportation (shipping), and Healthcare, each addressing trainees of different knowledge and skills. Credentials to access the platform are provided in the document to facilitate demonstration and validation of training in pilot activities.

This document extends the initial platform architecture in "D1.3 – THREAT-ARREST platform's initial reference architecture" and provides further details on components' communications including message broker communications and APIs, along with the network view of the different components' deployment. It overviews the platform's requirements defined in "D1.2 The platform's system requirements analysis report" and how these have been addressed in the first version of the platform. Out of 76 requirements, 46 requirements have been (fully or partially) addressed in the first version, while 13 requirements in progress and 17 not started. We refer to deliverable "D6.2 – Initial Installation and usage guidelines for the THREAT–ARREST platform" for the installation and usage guidelines of the THREAT-ARREST platform first version.

This document reports activities and results of tasks "T6.1 – Integration of tools and components into the THREAT-ARREST platform" and "T6.2 – Security of the THREAT-ARREST platform". The final version of the platform is due M32 and will be reported in "D6.4 – Final Prototype of Integrated THREAT-ARREST platform".

Table of Contents

List of Abbreviations

CTTP Cyber Threat and Training Preparation

DFP Data Fabrication Platform

DoA Description of Action

EMon Emulated Components Monitor

ET Emulation Tool

GT Gamification Tool

I/O Input/Output

IaaS Infrastructure as a Service

IAM Identity & Access Management

IDS Intrusion Detection System

IT Information Technology

JSON JavaScript Object Notation (data-interchange format)

JWT JSON Web Token

MB Message Broker

REST Representational State Transfer (cf. RESTful Web services)

SQL Structured Query Language

SSH Secure Shell

ST Simulation Tool

TT Training Tool

VM Virtual Machine

VT Visualisation Tool

WAF Web Application Firewall

List of Figures

List of Tables

1 Introduction

The first version of the THREAT-ARREST integrated platform has been achieved and released in M20 according to the DoA. The first version is the result of the integration of the latest version of the various tools developed in the technical work packages WP2, WP3, WP4 and WP5.

Early in the second year of the project, it was identified the need of a dedicated high-end server hosting the training services of THREAT-ARREST. It was agreed with the relevant technical partners the hardware and software requirements of their components and, consequently, it was derived a common specification of the minimum hardware requirements in terms of CPU, RAM, SSD, and bandwidth to host the THREAT-ARREST platform. Several Infrastructure-as-a-Service (IaaS) providers were examined their offers and selected the one offering the most suitable hardware configuration on an acceptable cost.

After a successful order of a dedicated bare metal server (by mid-January 2020), the integration process started with a technical workshop held in Milan, Italy on 20-21 January 2020. Several important decisions were taken on organisation and integration of the various tools into one functional platform. Since February 2020, a weekly consortium-wide technical teleconference (of two hours duration) is taking place to systematically address and synchronise technical activities on platform integration and operation. Additionally, 5 interim technical teleconferences and a *two-day* virtual workshop (consortium wide) on 16-17 March 2020 took place to better address the technical issues encountered in the platform integration process and boost activities and final decisions towards the first platform version.

The platform integration process had to accommodate, through small iterative cycles[1] (development-integration-testing) the evolution and enhancement of individual tools development within WP2–WP5 to ensure the first version of the platform successfully integrates latest version of the tools. The main difficulty in the integration process was the high dependability and interaction between the different tools which introduced the need of more frequent and agile discussions and iterations with technical partners.

The first version of the platform integrates:

- Emulation, Simulation, Gamification, Visualization, and Training Tools;
- CTTP modelling components such as the CTTP model editor, language, etc. as part of the Assurance Tool integration;
- Dashboard of the platform (as part of the Training Tool) offering management, training and administrative capabilities for trainers, trainees and admins;
- Model driven instantiation of training scenarios;
- Real time assessment/scoring of trainees; and
- Basic security mechanisms and configurations.

We note that the Assurance Tool, by the DoA, is to be developed in a later stage of the project and its integration is limited to the CTTP modelling capabilities in this first version. We refer to deliverable "D6.2 – Initial Installation and usage guidelines for the THREAT–ARREST platform" on the installation procedures and usage guidelines of the platform.

1.1 THREAT-ARREST Training as a Service

Following the latest platform domain purchase, and TLS certificate acquisition through letsencrypt.com certificate authority service, the platform's Dashboard (front-end) is accessible at https://www.threat-arrest.org.

[1] Similar to the Scrum methodology of project management and software development.

Table 1: Credentials for THREAT-ARREST Demonstration

	Trainee Username	**Trainee** Password	**Trainer** Username	**Trainer** Password
Smart Home & IoT	uc1trainee	uc1trainee!	uc1trainer	uc1trainer!
Smart Shipping	uc2trainee	uc2trainee!	uc2trainer	uc2trainer!
Healthcare	uc3trainee	uc3trainee!	uc3trainer	uc3trainer!

Table 1 shows the usernames and passwords for the entities created for demonstration of platform capabilities. They regard both trainees and trainers, and their access to training scenarios. Trainees are pre-assigned different training scenarios. Based on the first version, three training scenarios have been created for the different project use cases – Smart Home & IoT, Smart Shipping, and Healthcare, targeting trainees of different categories and skills. In the following, we list the videos of the different platform demonstrations uploaded on YouTube that can server as guidelines on how to use the platform functionality.

Table 2: THREAT-ARREST Demonstration Videos

THREAT-ARREST Demo Description	Link to Video
THREAT-ARREST Smart Energy Scenario Demo	https://youtu.be/0vGNXkne_wM
THREAT-ARREST Shipping Scenario Demo	https://youtu.be/vs8T1oZoha0
THREAT-ARREST Healthcare Scenario Demo	https://youtu.be/iFmFTBVWeio
THREAT-ARREST Training Tool Demo	https://youtu.be/DGOg1sEENCY
THREAT-ARREST CTTP Model Editor Demo	https://youtu.be/TR2jeRVLSlY
THREAT-ARREST Data Fabrication Platform (IBM) Demo	https://youtu.be/K0UiFgfWoHk

1.2 Document Structure

The rest of the document is structured as follows. Section 2 presents the integrated platform, particularly, the bare metal server specification, how to access the platform's Dashboard, the platform architecture and communications. Section 3 presents details of the platform security considered in the first version from a higher-level overview of confidentiality, integrity and availability to security mechanisms, security configuration and hardening of the server, and first vulnerability analysis. Section 4 presents the status of tools' requirements and how they are addressed in this first version. Section 5 concludes the document and outlines next steps of platform integration for the second and final version.

2 Integrated Platform First Version

The first version of the platform has been integrated and deployed on a dedicated bare metal server[2] at LeaseWeb.com located in the Netherlands. This decision allows the project consortium outreach and offer concept demonstration as a service to other organisations.

In the following we detail the server information underpinning the THREAT-ARREST platform:

- Server: Dell R630
- CPU: 2x Intel Deca-Core Xeon E5-2630v4 (20 physical/40 logical CPU cores)
- RAM: 192 GB DDR4
- Hard Disc: 2x960 GB SSD
- Uplink port speed: 1 x 1000 Mbps Full-duplex
- Bandwidth: 250 TB (Volume) monthly renewal
- IPv4: 3
- Software: Ubuntu 16.04; OpenStack Queens

The first version of the THREAT-ARREST platform consists of the integration of the following main components:

- Training Tool with Dashboard – The main front-end to the platform functionality and interfaces of all other tools. Responsible for user management, training initialisation and orchestration with respect to other tools, user authentication and security session management, and trainee assessment.
- Gamification Tool – The delivery of game-based training. The game called PROTECT is the game available in the first version of the platform.
- Visualisation Tool – Progressive in-browser visualisation (real-time) of state of cyber system simulation and emulation, and interactive front end to the simulation environment for user-selected actions.
- Emulation Tool – Responsible for the generation of an emulated cyber-system environment and infrastructure, and access management to the environment (through the Remote Access component below).
- Emulation Environment Monitoring – Responsible for monitoring resource utilisation in an emulated cyber system environment (created by the Emulation Tool). Resources utilisation vary from CPU usage, RAM usage, disc usage, network usage, etc.
- Simulation Tool – Responsible for the simulation of a cyber-system environment or components, the provisioning of an interactive simulation process (through the Visualisation Tool), and simulation of other processes, such as attacks or probes for the verification (observation) of user actions performed in the emulation environment. The last aspect entails a direct integration and connection between the simulation processes and the emulated cyber system environment.
- Assurance Tool – Responsible for the generation and management of CTTP models, in this first version of the platform. It offers means of retrieval of CTTP models to other tools of the platform, especially to the Training Tool.

[2] https://www.rackspace.com/library/what-is-a-bare-metal-server

- Message Broker – Responsible for efficient and reliable delivery of messages between the platform components, especially addressing the needs of asynchronous communications and messaging.
- Remote Access – Responsible for providing access to the platform services and components. An Apache Guacamole server is part of the Remote Access component for remote access and management to the emulated cyber system environment, and port forwarding setup to offering remote access to the individual platform tools each running in a dedicated environment (virtual machine).
- Data Fabrication Platform – Responsible for synthetic data and security logs fabrication needed for the training models and programs.

2.1 Architecture

The initial architecture was presented in "D1.3 – THREAT-ARREST platform's initial reference architecture". Few changes were faced in the period from month 6 to month 18. Mainly, a new component was introduced – a Dashboard, and a message broker was explicitly defined – RabbitMQ. The Dashboard component was defined as part of the Training Tool and will integrate platform functionality and services under a unified GUI. The RabbitMQ message broker component was selected to address reliable message delivery to all components with such necessities. For instance, emulation environment monitoring or system simulation processes require real time asynchronous exchange and handling of messages on the side of Training Tool's Dashboard. The notion of "real time asynchronous" is defined with respect to a cyber system emulation environment actions and events, and with respect to a cyber system simulation process execution.

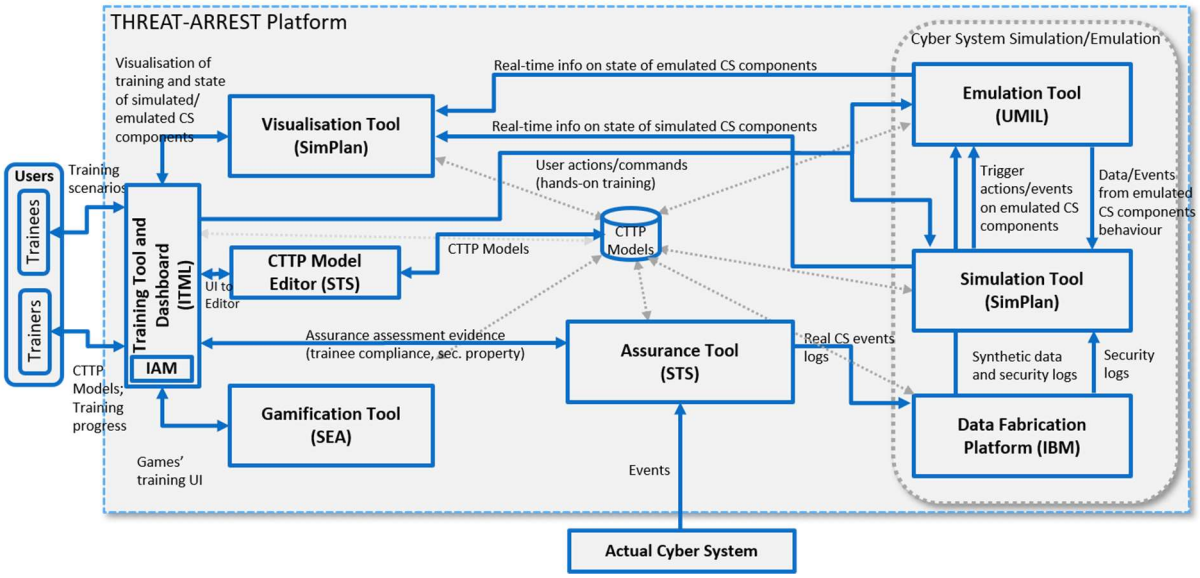

Figure 1: THREAT-ARREST Components Dataflow (high-level view)

Figure 1 presents a high-level interactions and dataflow of the platform components. It has very similar dataflow with respect to the initial architecture. We will briefly recall the main dataflow aspects from D1.3 important to consider for the platform integration activities.

- The Assurance Tool collects and monitors events coming from the real cyber system subject of simulation and emulation. There is a dedicated component, Event Captor, which will be present in the real cyber system to ensure all relevant events are properly captured and communicated to the Assurance Tool. A message broker is required to handle and deliver all events from a cyber system.

- The Simulation Tool sends real time asynchronous messages on state of simulated cyber system components or processes (e.g., GPS receiver, periodic attack probes, smart plug or smart device power states/consumption, etc.) to the Visualisation Tool, which is part of the Dashboard integration. A message broker is required to handle and deliver such a number of messages.
- The Emulation Tool monitors the state of the emulated cyber system environment, particularly the underlying hardware resource usage such as CPU usage, RAM usage, hard disk usage, network usage, etc. Such monitoring statistics are sent with high frequency to the Visualisation Tool for graphics visualisation to trainees. A message broker is required to handle and deliver such a number of messages.
- The Visualisation Tool provides an interactive front-end to the simulation environment where trainees can select different options or actions during cyber system simulation and continue training accordingly. Such user actions are delivered to the Simulation Tool through a message a message broker service.
- The Training Tool is the main orchestrator of all platform tools' functionalities. As such, it requires initialisation of a cyber system emulation/simulation which in turn requires asynchronous confirmation when the emulated/simulated environment is ready for training purses. A message broker is used to address such results of initialisation. The same also applies on notifications by the Data Fabrication Platform when ready with the fabricated synthetic security logs.

Although the need of a message broker is well identified, all other components' communications are defined to follow the RESTful style API, such as access to CTTP models at the Assurance Tool, or initialisation and access to serious games at the Gamification Tool, etc.

In the first version of the integrated platform, two means of integration of platform components are defined – REST API and Message Broker enabled communications. These will be described in next sections of the document.

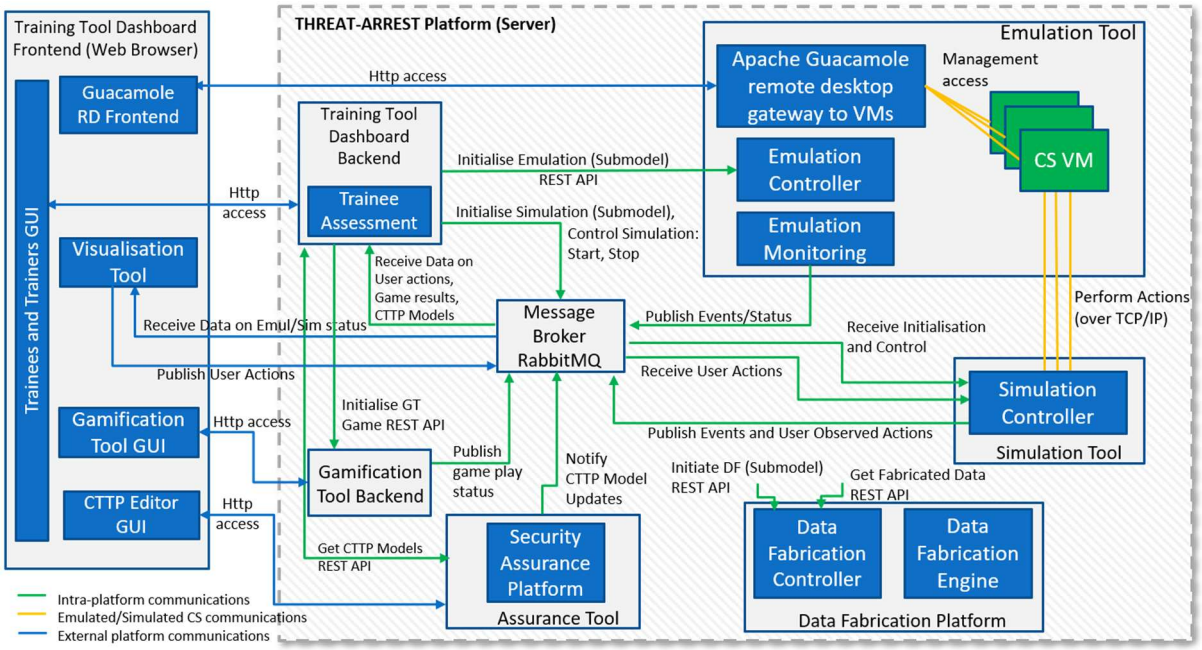

Figure 2: THREAT-ARREST Components Communications

Figure 2 shows a view of the components' communications in the first version of the integrated platform. The main scope of the view is to illustrate the established means of integration and communications between the components.

As shown in the figure, communications are mostly centred and mediated by the message broker component while some communications go through REST API. It is also shown a view of the Dashboard front-end (a trainee's web browser) communications with the different platform components illustrated as external platform communications.

The Dashboard, provided by the Training Tool, serves as the central starting point for users to access the various platform components. It integrates the graphical interfaces of:

- Guacamole software for login to the VMs of the emulation environment,
- Visualization of the cyber system simulation/emulation environment, and
- Gamification tool, currently the PROTECT game GUI.

The Training Tool acts as the orchestrator of the training process and implements real time trainee performance assessment by comparing trainee performance with expected traces defined in the CTTP model.

Looking from a training perspective, there are three types of communications defined in the platform:

- Intra-platform communications between the tools shown in a green colour underpinning internal platform functionality/operation;
- External platform communications between trainee's web browser and the platform server (backend), shown in a blue colour; and
- Emulated/simulated cyber system communications shown in orange colour in Figure 2. These communications are defined between components of the emulated cyber system environment, and between cyber system simulation or simulation processes (e.g., attack probes) and the emulated cyber system environment. Although these communications are internal to the platform, they represent a different category of communications defined by CTTP models and according to an existing cyber system infrastructure scope of training.

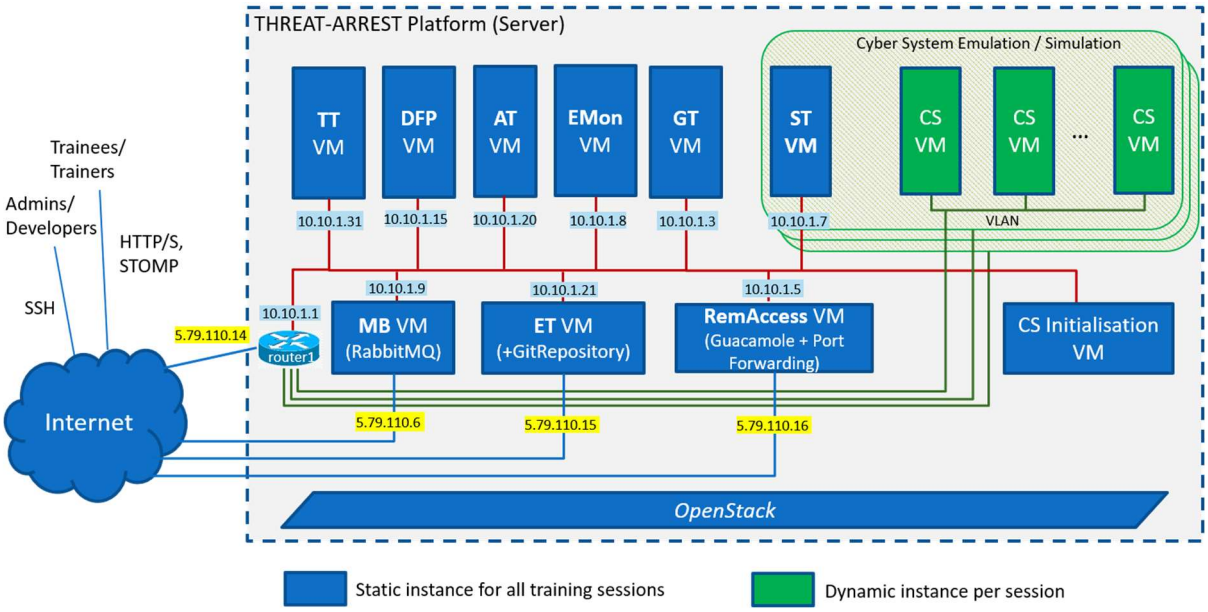

Figure 3: THREAT-ARREST Components Network View

Figure 3 shows the network view of the integrated platform and how the different components are networked together to enable the communications discussed before (Figure 2). OpenStack version Queens has been installed on the bare metal server for the management of hardware resources and operational environments for each platform component.

The following Virtual Machines (VMs) have been created for the first version of the integrated platform:

- **TT**: Training Tool and Dashboard
- **DFP**: Data Fabrication Platform
- **AT**: Assurance Tool
- **EMon**: Emulated Components Monitor (monitoring resource usage of green VMs)
- **GT**: Gamification Tool (currently hosting PROTECT game)
- **ST**: Simulation Tool
- **MB**: Message Broker (hosting a RabbitMQ instance)
- **ET**: Emulation Tool (Hosting Emulation Controller). The Git repository is still an ongoing activity and will be fully operational shortly after M20.
- **RemAccess**: Remote Access to all platform tools/services except the Message Broker service which is run in a different VM. The RemAccess VM hosts an instance of the Apache Guacamole Server offering remote (desktop) access to the VMs of the emulation environment (green VMs in the figure), and implements port forwarding to internal to the platform tools' VMs.

The VMs described above are created and deployed at a platform deployment stage and are static instances across all training sessions. There are identified by a blue colour in the figure. There are a set of VMs dynamically created at a training session initialisation by the Emulation Tool, and are identified as green VMs in Figure 3. The green VMs represent the cyber system emulation environment for a given training session.

There is one virtual subnet created to connect the VMs above (the blue VMs), and one router created to connect the blue VMs with the Internet. Importantly, the green VMs (when dynamically created) are in a different subnet, according to a CTTP model, but connected to the router to allow blue VMs access the green VMs, especially the Simulation Tool (ST) VM needs access to the green VMs for the purposes of training (e.g., simulating an attack or probe to the green VMs). This network configuration also allows the green VMs connect to the Internet when needed for the purposes of training.

The yellow highlighted IP addresses are public IP addresses provided by LeaseWeb provider. The THREAT-ARREST platform is accessible through the RemAccess machine's IP address and port forwarded to the respective local VMs of the platform components. Table 3 shows the Tool VMs and the assigned local and public IP addresses.

Table 3: Tool VMs and IPv4 Addresses

Tool VM	Local IP Address	Public IP Address
TT VM	10.10.1.31	-
DFP VM	10.10.1.15	-
AT VM	10.10.1.20	-
EMon VM	10.10.1.8	-
GT VM	10.10.1.3	-
ST VM	10.10.1.7	-
MB VM	10.10.1.9	5.79.110.6

ET VM	10.10.1.21	5.79.110.15
RemAccess VM	10.10.1.5	5.79.110.16

Table 4 shows the port forwarding setup for the first version of the platform. We refer to deliverable D6.2 on details of network settings and services running on each tool's VM along with the corresponding hardware and software requirements.

Table 4: Port Forwarding to Platform's Components

RemAccess TCP Port	Local Tool VM and TCP Port
2022	GT port 22
2080	GT port 80
2443	GT port 443
3022	TT port 22
80	TT port 80
443	TT port 443
38080	TT port 8080
4022	DFP port 22
4080	DFP port 80
5022	AT port 22
5080	AT port 80
58080	AT port 8080
5443	AT port 443
53306	AT port 3306
5672	AT port 5672
6022	EMon port 22
6080	EMon port 80
6808	EMon port 8080
1022	ST port 22
18080	ST port 8080

Given above settings, access to the THREAT-ARREST Training Tool Dashboard is at http://5.79.110.16 and https://5.79.110.16 which port forwards to the Training Tool's VM at 10.10.1.31:80 and 10.10.1.31:443, respectively.
Following the latest platform domain purchase and TLS certificate acquisition through letsencrypt.com authority, the platform's Dashboard is exclusively accessible at https://www.threat-arrest.org.

Training session initialization
Several tools of the platform are initialised upon a trainee selects a specific training scenario. In the first version of the platform the following tools are initialised for a training session: ET, ST, GT. The initialisation is performed after a successful trainee login session and through a JSON Web Token[3] (JWT) token with an agreed structure to carry all necessary details for initialisation.

Figure 4 shows the JWT structure consisting of:

i) userID of trainee authentication session;

[3] https://jwt.io/

ii) roleID of the trainee in a given training scenario (such as a "trainee" generic role or more specific one such as "captain", "smart home user", "security officer", etc);

iii) SessionID of the current training session ID;

iv) ScenarioID of the training scenario selected; and

v) CTTPsubmodel corresponding to the training submodel of a given tool of the given training scenario, such as Emulation, Simulation or Gamification submodel.

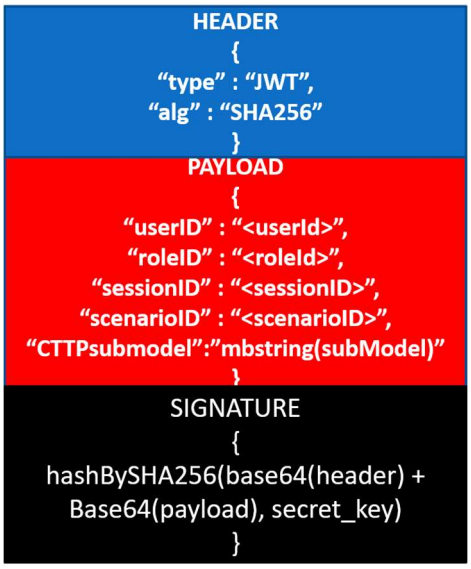

Figure 4: JWT used for Tools Initialisation

2.2 Message Broker Communications

It has been deployed RabbitMQ[4] software to address the message broker needs of the platform. To this end, several exchanges and queues were created to address each component's communication needs. We refer to Section 2 of "D2.4 – Emulation tool interoperability module v1", "D4.3 – Training and Visualisation tools IO mechanisms v1", or "D5.3 – The Simulation component IO module v1" for an introduction and concept of Exchanges and Queues.

Message Broker Exchanges

By design, each tool of the THREAT-ARREST platform is defined one or more static exchanges at the message broker to publish messages across all training sessions. To distinguish messages from one training session to another, it was decided to use Exchanges of type 'topic' and routing key of the format <ScenarioID>.<SessionID> to each message being published. We note that for tool initialisation purposes, an Exchange of type 'fanout' is defined per relevant tool that allows messages posted to such Exchange always reach the tools listening for initialisation. We also note that different design decisions can be achieved of types Exchanges (such as through default broker exchanges "amq.topic" or "amq.fanout" or "amq.direct" etc.) but with similar performance or efficiency. A revision of the message broker communications shortly after M20 will be performed with the aim to re-evaluate tools' necessities and improve communications.

Table 5 lists the Exchanges created at the message broker to facilitate tools' communications the first version of the integrated platform. The "Publisher" column identifies the tool using the Exchange to publish messages while the "Consumer" column the tool consuming the messages.

[4] https://www.rabbitmq.com/

We note a consumer tool needs to create or subscribe to a Queue in order to consume messages from Exchanges.

Table 5: Message Broker Exchanges

Exchange Name	Type	Publisher	Consumer	Routing Key Structure	Description
ta.csemulation.init	fanout	TT	ET		Cyber system emulation initialisation for a training scenario and session. Training scenario and session IDs in payload of the initialisation message.
ta.csemulation.initresult	topic	ET	TT	ScenarioID.SessionID	Result of emulation initialisation for a given training scenario and session.
ta.csemulation.moninit	fanout	TT	EMon		Resource monitoring initialisation for an emulated cyber system environment. Training scenario and session IDs in payload of the initialisation message.
ta.csemulation.moninitresult	topic	EMon	TT	ScenarioID.SessionID	Results of monitoring initialisation for a given training scenario and session.
ta.csemulation.monstats	topic	EMon	VT	ScenarioID.SessionID	Monitoring statistics of an emulated cyber system of a given training scenario and session.
ta.cssimulation.init	fanout	TT	ST		Cyber system simulation initialisation for a training scenario and session. Training scenario and session IDs in payload of the initialisation message.
ta.cssimulation.initresult	topic	ST	TT	ScenarioID.SessionID	Result of a simulation initialisation for a given training scenario and session.
ta.cssimulation.control	topic	TT	ST	ScenarioID.SessionID	Simulation control commands (start, stop) for a given training scenario and session
ta.cssimulation.controlresult	topic	ST	TT	ScenarioID.SessionID	Result of simulation control commands (start,

Exchange Name	Type	Publisher	Consumer	Routing Key Structure	Description
					stop) for a given training scenario and session.
ta.cssimulation.events	topic	ST	VT	ScenarioID.SessionID	Events and status of cyber system simulation.
ta.cssimulation.useractions	topic	ST	TT	ScenarioID.SessionID	User (trainee) actions observed from a simulation environment/activity for a given training scenario and session.
ta.visualisation.useractions	topic	VT	ST	ScenarioID.SessionID	User actions performed on the Dashboard/ Visualisation Tool, e.g. any user choice (selection) during a hands-on training session, for a given training scenario and session.
ta.datafabrication.status	topic	DFP	ET, ST	ScenarioID.SessionID	Status of data fabrication process when security logs are ready and how to access those in a given training scenario and session.
ta.gamification.statusresults	topic	GT	TT	ScenarioID.SessionID	Results of a game played by a trainee in a given training scenario and session.
ta.assurance.notifytt	Fanout	AT	TT		Notify the TT upon creation or update on a Training Programme

Message Broker Queues

It has been defined a dynamic notion of queues for a majority of message exchanges in the first version of the platform. Queues are created dynamically by each tool upon training scenario initialization and are valid only until a training session is over (when a training scenario is initialized a training session ID is created). They are auto-deleted upon training session termination. Some queues are created statically for tools initialization and last across all training sessions. They are set up upon initial platform deployment. Table 6 presents queues created to facilitate communications between the Visualisation Tool and the Simulation Tool. The table illustrates the way MB communications are addressed in this first version of the platform. We refer to D6.2 for details of message broker installation and setup.

Table 6: Message Broker Queues for Visualisation and Simulation Tools Communications; queue properties are always "Durable; Auto-delete"

Tool Sending	Tool Receiving	Queue Name	Binding Key	Exchange Name
EMon	VT	<dynamically generated>	ScenarioID.SessionID	ta.csemulation.monstats
TT, VT	ST	<dynamically generated>		ta.cssimulation.init
ST	TT, VT	<dynamically generated>		ta.cssimulation.initresult
TT, VT	ST	<dynamically generated>	ScenarioID.SessionID	ta.cssimulation.control
ST	TT, VT	<dynamically generated>	ScenarioID.SessionID	ta.cssimulation.controlresult
ST	VT	<dynamically generated>	ScenarioID.SessionID	ta.cssimulation.events
VT	ST	<dynamically generated>	ScenarioID.SessionID	ta.visualisation.useractions
ST	TT, VT	<dynamically generated>	ScenarioID.SessionID	ta.cssimulation.useractions

2.3 REST API

We list the APIs provided by each platform component to offer its functionality to other platform components in this first version of the integrated platform. We note that some of the APIs will be subject of further revision in the next version of the platform.

2.3.1 Assurance Tool API

2.3.1.1 Overview

hostname = 5.79.110.16:58080

Projects (projectID):

1. Response & Mitigation
2. Navigation combo attack (phishing email and GPS spoofing)
3. Incident Response

Organisation (organisationID):

1. Lightsource LAB LTD
2. DANAOS Shipping Company LTD
3. Agenzia Regionale Strategica per la Salute ed il Sociale

2.3.1.2 Emulation Model

Insert Emulation Model
Insert a new emulation model. Each emulation model is attached to a specific project and scenario. `On success (201 Created)` Emulation Model inserted for project: {projectID} and organisation: {organisationID}
POST $hostname. **/assurancetool/rest/api/emulation/insert/**
Content-Type: application/json

Update Emulation Model
Update an existing emulation model. Each emulation model is attached to a specific project and scenario. `On success (200 Ok)` Emulation Model update for project: {projectID} and organisation: {organisationID}
PUT $hostname. **/assurancetool/rest/api/emulation/update/**
Content-Type: application/json

Get Emulation Model (XML format)
Get a unique emulation model (in XML format) `On success (200 Ok)` Returns the existing Emulation Models (XML).
GET $hostname. **/assurancetool/rest/api/emulation/getXML/{organizationID}/{projectID}/**

Get Unique Emulation Model (JSON format)
Get a unique emulation model. `On success (200 Ok)` Returns an Emulation Model (JSON Object)
GET $hostname. **/assurancetool/rest/api/emulation/get/single/{organizationID}/{projectID}/**

Get Emulation Models
Get all the emulation models for a specific project and organisation. `On success (200 Ok)` Returns the existing Emulation Models (JSON Array)
GET $hostname. **/assurancetool/rest/api/emulation/get/{organizationID}/{projectID}/**

2.3.1.3 Simulation Model

Insert Simulation Model

Insert a new simulation model. Each simulation model is attached to a specific project and scenario. `On success (201 Created)` Simulation Model inserted for project: {projectID} and organisation: {organisationID}
POST $hostname.**/assurancetool/rest/api/simulation/insert/**
Content-Type: application/json

Update Simulation Model
Update an existing simulation model. Each simulation model is attached to a specific project and scenario. `On success (200 Ok)` Simulation Model update for project: {projectID} and organisation: {organisationID}
PUT $hostname. **/assurancetool/rest/api/ simulation /update/**
Content-Type: application/json

Get Unique Simulation Model (JSON format)
Get a unique simulation model. `On success (200 Ok)` Returns a Simulation Model (JSON Object)
GET $hostname. **/assurancetool/rest/api/simulation/get/single/{organizationID}/{projectID}/**

Get Simulation Models
Get all the simulation models for a specific project and organisation. `On success (200 Ok)` Returns the existing Simulation Models (JSON Array)
GET $hostname./**assurancetool/rest/api/simulation/get/{organizationID}/{projectID}/**

2.3.1.4 Training Programme

Insert Training Programme
Insert a new simulation model. Each simulation model is attached to a specific project and scenario.

`On success (201 Created)` {"organisationID":1,"trainingID":4,"projectID":1,"status":"create","timestamp":"…"} This function also notifies – through RabbitMQ- the Training Tool that a new Training Programme was created
POST $hostname.**/assurancetool/rest/api/training/insert/**
Content-Type: application/json

Update Training Programme
Update an existing training programme. Each training programme is attached to a specific project and scenario. `On success (200 Ok)` {"organisationID":1,"trainingID":4,"projectID":1,"status":"update","timestamp":"…"} This function also notifies – through RabbitMQ- the Training Tool that a new Training Programme was created
PUT $hostname. **/assurancetool/rest/api/training/update/**
Content-Type: application/json

Get Unique Training Programme (JSON format)
Get a unique training programme. `On success (200 Ok)` Returns a Training Programme (JSON Object)
GET $hostname.**/assurancetool/rest/api/training/get/single/{organizationID}/{projectID}/**

Get Training Programmes
Get all the training programmes for a specific project and organisation. `On success (200 Ok)` Returns the existing Training Programmes (JSON Array)
GET $hostname.**/assurancetool/rest/api/training/get/{organizationID}/{projectID}/**

2.3.1.5 Gamification Model

Insert Gamification Model

Insert a new gamification model. Each gamification model is attached to a specific project and scenario. `On success (201 Created)` Gamification Model inserted for project: {projectID} and organisation: {organisationID}
POST $hostname. **/assurancetool/rest/api/gamification/insert/**
Content-Type: application/json

Update Gamification Model
Update a gamification emulation model. Each gamification model is attached to a specific project and scenario. `On success (200 Ok)` Gamification Model update for project: {projectID} and organisation: {organisationID}
PUT $hostname. **/assurancetool/rest/api/gamification/update/**
Content-Type: application/json

Get Unique Gamification Model (JSON format)
Get a unique gamification model. `On success (200 Ok)` Returns a Gamification Model (JSON Object)
GET $hostname. **/assurancetool/rest/api/gamification/get/single/{organizationID}/{projectID}/**

Get Gamification Models
Get all the gamification models for a specific project and organisation. `On success (200 Ok)` Returns a Gamification Model (JSON Array)
GET $hostname./**assurancetool/rest/api/gamification/get/{organizationID}/{projectID}/**

2.3.1.6 Data Fabrication Model

Insert Data Fabrication Model

Insert a new data fabrication model. Each data fabrication model is attached to a specific project and scenario. `On success (201 Created)` Data Fabrication Model inserted for project: {projectID} and organisation: {organisationID}
POST $hostname. **/assurancetool/rest/api/datafabrication/insert/**
Content-Type: application/json

Update Data Fabrication Model
Update a data fabrication emulation model. Each data fabrication model is attached to a specific project and scenario. `On success (200 Ok)` Data Fabrication Model update for project: {projectID} and organisation: {organisationID}
PUT $hostname. **/assurancetool/rest/api/datafabrication/update/**
Content-Type: application/json

Get Unique Data Fabrication Model (JSON format)
Get a unique data fabrication model. `On success (200 Ok)` Returns a Data Fabrication Model (JSON Object)
GET $hostname. **/assurancetool/rest/api/datafabrication/get/single/{organizationID}/{projectID}/**

Get Data Fabrication Models
Get all the data fabrication models for a specific project and organisation. `On success (200 Ok)` Returns the existing Data Fabrication Models (JSON Array)
GET $hostname./**assurancetool/rest/api/datafabrication/get/{organizationID}/{projectID}/**

2.3.2 Gamification Tool API

A game is initiated by the trainee through the Dashboard of the Training Tool. When the trainee presses the button to start a game, the Training Tool invokes the game in the Gamification Tool by calling the corresponding URL. This URL contains all necessary parameter values for the instantiation of a game in a certain training session. After a game has been finished, the Gaming Tool returns the result and other necessary information to the Training Tool.

The following API is used for instantiating a game by the Training Tool. The called URL contains a parameter value in the form of a JSON Web Token (JWT) which represents all necessary information according to the instantiation of a game and the session the game is associated with, see Section 2.1. Figure 5 shows such an URL by the example of starting the game PROTECT. Regarding the instantiation information the JWT contains an instance of the gamification CTTP submodel (see deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1"). This submodel contains information like the used learning content, the difficulty level and the game time. The values of the parameters *roleID*, *sessionID*, *userID* and *scenarioID* in the JWT allow the Training Tool to associate a played game to a certain trainee and the corresponding training session. The remaining parameters of the JWT provide information regarding its time of issue and expiration time

Start a game from the Training Tool
The Training Tool invokes a game of the Gamification Tool for a specific gaming scenario. On success (200 Ok) Input: gamificationCttpSubmodel, roleID, sessionID, exp *(expiration time of token),* userID, scenarioID, iat *(time of token issue)*
GET $hostname**/?val <token>**

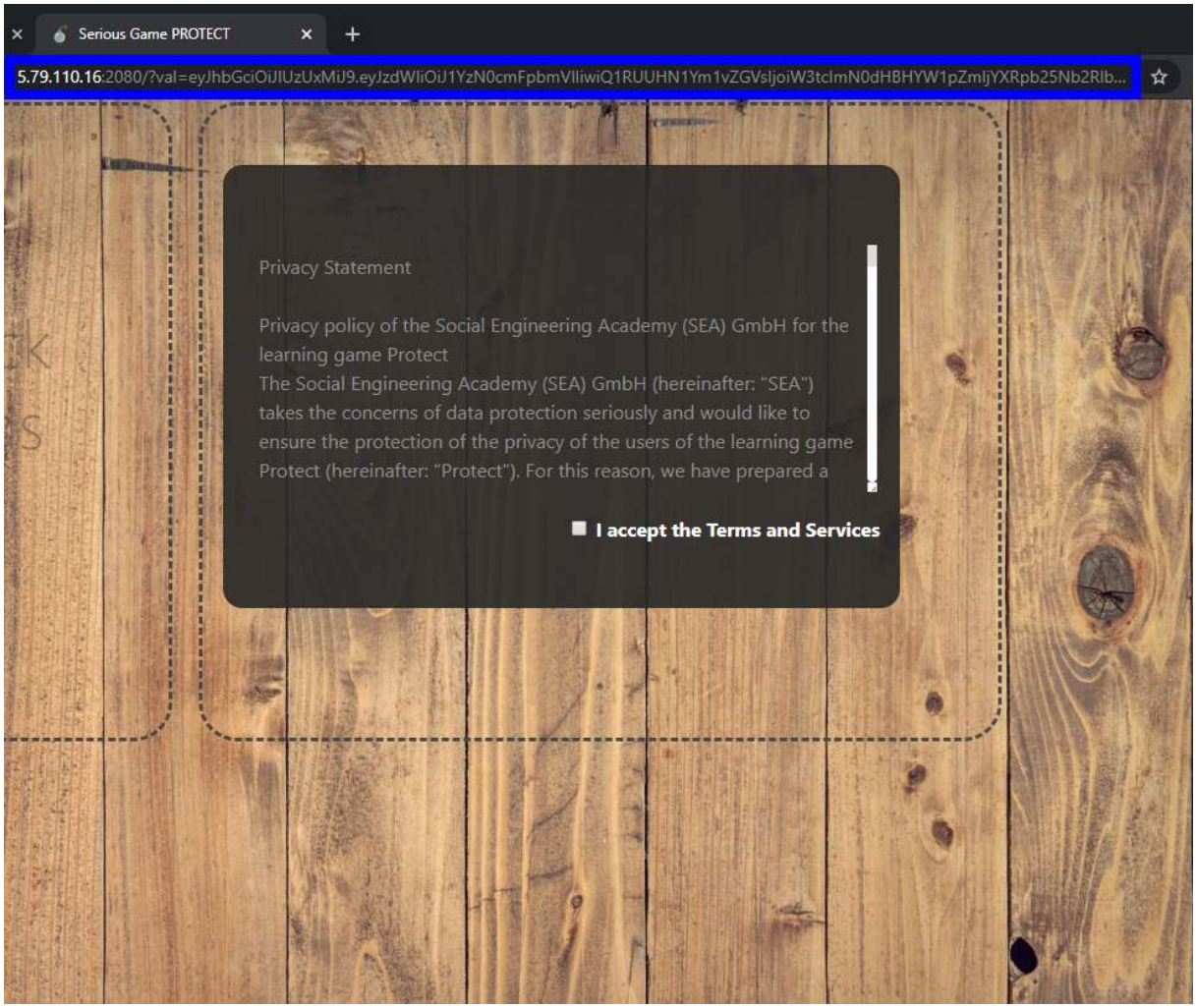

Figure 5: URL for instantiating a game by the example of PROTECT

The subsequent API is used for returning results of finished games (see Figure 6) from the Gamification Tool to the Training Tool. Therefore, the Gamification Tool informs the Training Tool via the message broker about the result of a finished game. A published result contains game specific information like the score, number of lost lives and the remaining game time. Additionally, it includes information for the association of the gaming results to a trainee and the relevant training session. The value of the parameter *status_code* indicates if a game has been played without any technical issues.

Return the game's results back to the Training Tool
The Gamification Tool evaluates the game of a trainee and reports the result back to the Training Tool. The Training Tool parses this information and assesses the trainee's performance based on the related CTTP model. Broker message via the exchange *ta.gamification.statusresults* (in a JSON format)
Message format: {

```
    "status_code": [Indicates if the playing of the game was error-free]
    "scenarioId": [ID of training scenario],
    "sessionId": [ID of training session],
    "userId":[ID of user],
    "roleId" [ID regarding the role of user],
    "score": [Final score]
    "remainingTime": [Remaining game time at the end of the game]
    "numberLostLives": [Numbers of lives lost during a game]
    "gameStatues": [Indicates if a game has been won or lost]
    "maximumScore": [The highest score that can be achieved in a particular gaming
        session]
}
```

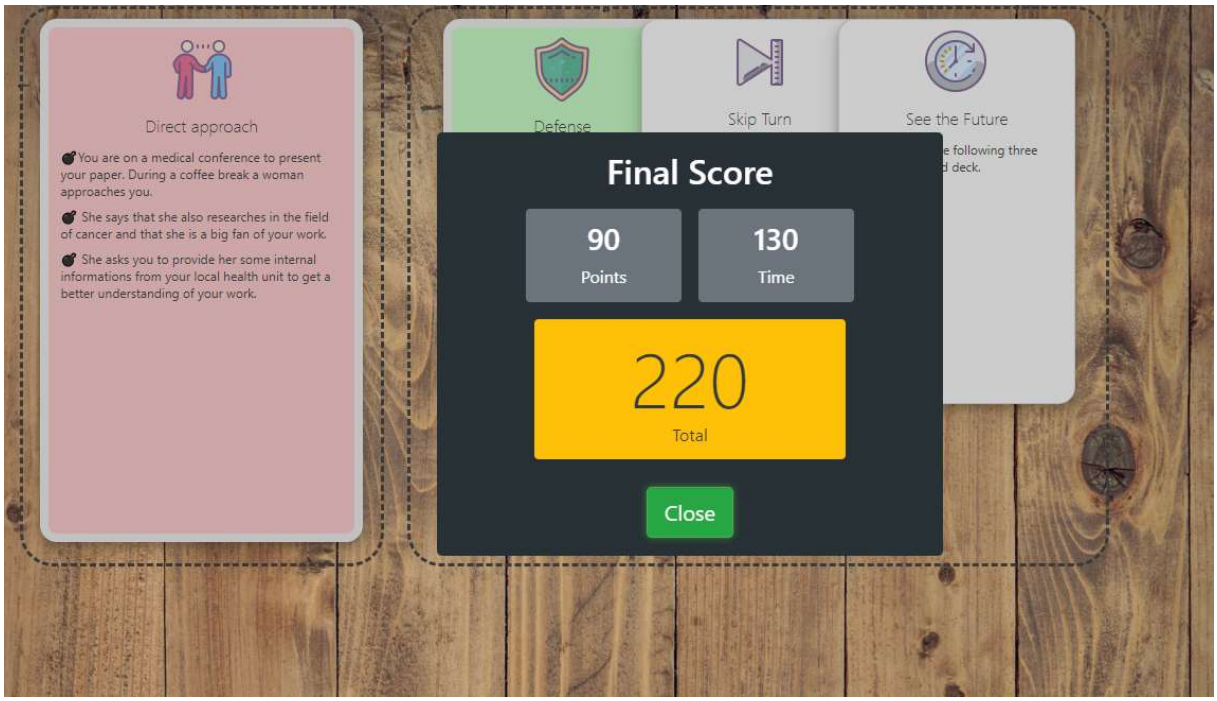

Figure 6: End of a game by the example of PROTECT

2.3.3 Emulation Tool API

The emulated environment is created by the ET upon request of the TT. The ET, in its first version, provides the TT with the following two methods, that allows the tool to create the training environment, taking in input the XML of the Emulation Sub-model, and releasing the environment at the end of the training scenario to save server resources.

Get Deployed Stack Data by XML
Require the instantiation of a new stack and get the connection data for all the VMs composing the stack. It accepts in input the XML versions of the specific instance of the Emulation CTTP submodel. `On success (200 Ok)` Returns the connection data (JSON Array) relative to the deployed stack.

Example of returning data:

```
[
  {
    "vmname": "WindowsMachine",
    "vmuser": "WindowsMachine_Admin",
    "vmpwd": "f46b2abda4",
    "guacamolelink":
"http://$hostname:8080/guacamole/#/client/NDkzAGMAcG9zdGdyZXNxbA==/
        ?username=WindowsMachine_Admin&password=f46b2abda4"
  }
]
```

POST http://$ET:8080/**emulation/getVMfromXML -H "accept: application/json" -H "Content-Type: application/json" -d "XML Scenario"**

Delete Deployed Stack
Delete an instantiated stack. `On success (200 Ok)` Return `ok`.
DELETE http://$ET:8080/emulation/deletestack?stackname=<scenario name>

2.3.4 Visualisation Tool API

The Visualisation Tool (VT) shows the status of an active training session to a trainee. To do so, the Training Tool opens a new browser tab in the trainee's web browser. The URL to be opened points to the Visualization Tool backend and contains all parameters required to initialize it for a particular training session. These parameters are contained in the JSON web token (see "Figure 4: JWT used for Tools Initialisation" for its structure). In particular, the scenarioID, sessionID and roleID are used at the moment.

Therefore, the VT frontend is loaded by a GET request to the following URL:

Load Visualization Tool
GET http://<VT_URL>/<ScenarioId>/index.html?jwt=<token>

2.3.5 Emulated Components Monitor API

Emulated Components Monitor (a.k.a. Resource Monitor or just Monitor), as it is described in D2.2: "Emulated components monitoring module", is a collection of collaborative services aimed to provide accurate readings of the Platform's hardware and virtual resources and make them available for both automatic resource management tools and resource visualisation tools.

As it is described in the D6.2 "Initial installation and usage guidelines for the THREAT-ARREST platform", the Monitor is implemented as a Web Service, hosted by an Apache Tomcat Web Server, permanently running on the EMon VM (shown on Figure 3).

The Resource Monitor supports two modes of operation; the appropriate REST APIs are described below.

2.3.5.1 Resource Monitor API

The Resource Monitor is capable of providing the resource readings through a series of its REST API GET calls to the Monitor http endpoint:

`http://<emon-host-IP>:8080/ResourceMonitor/monitor`

2.3.5.1.1 Get Servers (VMs)

This GET function call provides a JSON-formatted list of Servers (VMs) currently available at the Threat-Arrest Emulation Environment.

```
curl -X GET http://<emon-host-IP>:8080/ResourceMonitor/monitor/servers
```

2.3.5.1.2 Get Resources

This GET function call expects a JSON-formatted list of the VM names as a query param `vms`, and provides JSON-formatted resource readings for the specified VMs:

```
curl -X GET http://<emon-host-IP>:8080/ResourceMonitor/monitor/resources?vms=<VM name list>
```

In case the query param `vms` is omitted, this function call returns resource readings for all the available VMs:

```
curl -X GET http://<emon-host-IP>:8080/ResourceMonitor/monitor/resources
```

2.3.5.2 Resource Monitor Controller API

The Resource Monitor is capable of periodically posting the resource readings to a permanent message exchange `ta.csemulation.monstats` at the RabbitMQ Message Broker, described in chapter 2.2 of this document. The periodic postings are controlled through the REST API POST commands to the Monitor Actions http endpoint:

`http://<emon-host-IP>:8080/ResourceMonitor/monitor/actions`

2.3.5.2.1 Start

This POST method starts a resource monitoring session and periodic posting to the RabbitMQ Message Broker's message exchange mentioned above. This method expects a tuple of a Scenario ID and a Session ID as URL path params, which is used as a binding key for the Message Broker's exchange: `<Scenario ID>.<Session ID>`. In addition, this function expects an optional JSON-formatted list of the VM names as a query param `vms`.

In case the `vms` list is provided, the Monitor posts resource readings for the specified VMs:

```
curl -X POST http://<emon-host-IP>:8080/ResourceMonitor/monitor/actions/start/{scenario-ID}/{session-ID}?vms=<VM name list>
```

Otherwise readings for all the available VMs are posted:

```
curl -X POST http://<emon-host-IP>:8080/ResourceMonitor/monitor/actions/start/{scenario-ID}/{session-ID}
```

2.3.5.2.2 Stop

This POST method stops previously started resource monitoring and posting session. This method expects a tuple of a Scenario ID and a Session ID as URL path params to stop the appropriate session:

```
curl -X POST http://<emon-host-IP>:8080/ResourceMonitor/monitor/actions/stop/{scenario-
ID}/{session-ID}
```

2.3.6 Data Fabrication Platform API

As it's described in "D5.1 – Real event logs statistical profiling module and synthetic event log generator v1" document, IBM's Data Fabrication Platform (DFP) (IBM, 2017) is a web-based central platform for generating high-quality data for testing, development, and training. As it's also described in D5.1, the IBM Data Fabrication Platform has been enhanced to support the THREAT-ARREST project requirements. The DFP is enriched with an ability to generate sequences of simulated cyber-events in general, and synthetic security events log files in particular.

By the time being, when this document is written and released, the enhanced Data Fabrication Platform is not integrated yet within the THREAT-ARREST Platform. It is rather deployed externally and being used off-line as a stand-alone application for fabrication of both, the static DB records as well as dynamic scenario log files. The DFP is being modified to be deployed as a Web Service, hosted under an Apache Tomcat Server. This enhancement along with other important features will be documented in the future "D5.5 – Real event logs statistical profiling module and synthetic event log generator v2" deliverable. The DFP's functionality will be available through its REST API, which we plan to describe along with its integration and interaction with other THREAT-ARREST Platform components in the future "D6.4 – Final Prototype of Integrated THREAT–ARREST platform" deliverable.

3 Platform Security

In this section, we present the security mechanisms and provisions of the THREAT-ARREST platform itself. This includes:

i) the analysis of the *current threat landscape* for e-learning platforms,

ii) the identification of the *potential vulnerable elements in the THREAT-ARREST* architecture,

iii) the *main defence mechanisms* that have been deployed so far as well as the provisions for the next finally integrated version of the platform,

iv) the *user security and privacy preservation*, and

v) a *penetration testing* analysis that will verify the actual security level of the documented mechanisms and policies.

3.1 Security Guidelines and E-learning Platform Security

General security guidelines regarding e-learning platforms focus on specific e-learning platforms like Moodle and alike. Since THREAT-ARREST e-learning platform mainly consists of in-house developed software general security guidelines for e-learning platforms are not applicable.

Security of THREAT-ARREST e-learning platform should focus mainly in these topics.

a) Privacy of user data

b) Platform resiliency against malicious user misuse

c) Platform integrity

Privacy of user data is very important for the platform since it may result into unforeseeable outcomes (like employee arguing about salary differences based on colleague score). Therefore, it is very important to assure that data are secured against unauthorized access. That is achieved by using SQL statements and strict access separation.

User – student or an agent utilizing such users access might have malicious intentions, not only on the platform itself but itself, but also the rest of the Internet. Possible outgoing attacks must be blocked in order to prevent criminal liability.

Finally given the nature of THREAT-ARREST e-learning platform depending on type of scenario user may be given rights to execute code on the platform (i.e. virtual machine). All changes done by user on such virtual machine has to be reverted. That is easily achieved by reverting accessed virtual machine. Possible further access from such virtual machine towards the rest of the platform must be prevented on network level (i.e. firewall), closest to the user.

3.2 Common Threats to the THREAT-ARREST Platform

Common security threats to the THREAT-ARREST Platform can be easily described in form of threat actors.

Automated scanners – automated scanners search for a known vulnerability. When found it is exploited and most often the victim computer is added to the botnet[5]. This is usually lowest risk, because no harm is done to the system itself. The system is then misused either for other attacks or for mining cryptocurrencies. This is also easiest to prevent just by keeping the software up to date.

[5] https://en.wikipedia.org/wiki/Botnet

Malicious external attacker – this threat actor can be divided into several subcategories mainly by their objectives. First would be script kiddie wanting to prove himself/herself. Script kiddie is pretty much comparable to a good automated scanner. On the other end of the chain is skilled hacker or APT with aim to either steal personal data or to steal ideas and or to ask for ransom – to either regain control or not share personal data. This actor is probably the most dangerous, but the probability of targeting THREAT-ARREST platform is low.

Malicious user – This threat actor would probably seek to improve its own score or to spy on others. Even trying to obtain personal information on its colleagues. The improved risk here relies on ability to access parts of platform meant for authenticated user which results in greater attack surface. On the other hand, generally this kind of threat actor possess skills of a script kiddie. However, since we are securing platform for educating users in a field of cyber security, chances to come across an enthusiast in this field are higher. For the "defenders" this present form of incidents that are easier explained and tracked since the user can be eventually located and identified.

Malicious insider – Generally most dangerous threat actor. Given the knowledge about internal network and elevated privileges.

While the ordering of threat actors mention is from least dangerous to most dangerous. The probability of such encounter is exactly opposite - from most probable to least probable. Based on current state of things, the biggest threat is posed from Malicious external attacker. That is because it's more likely then Malicious user and Malicious attacker.

3.3 Overview of Components/Functions vs Security Mechanisms vs C/I/A/Auth Properties

We will overview the security aspects addressed in the first version of the integrated platform. These aspects are considered important for understanding and evaluating the security posture of THREAT-ARREST and seeking technical compliance with the General Data Protection Regulation (GDPR) (GDPR, 2016).

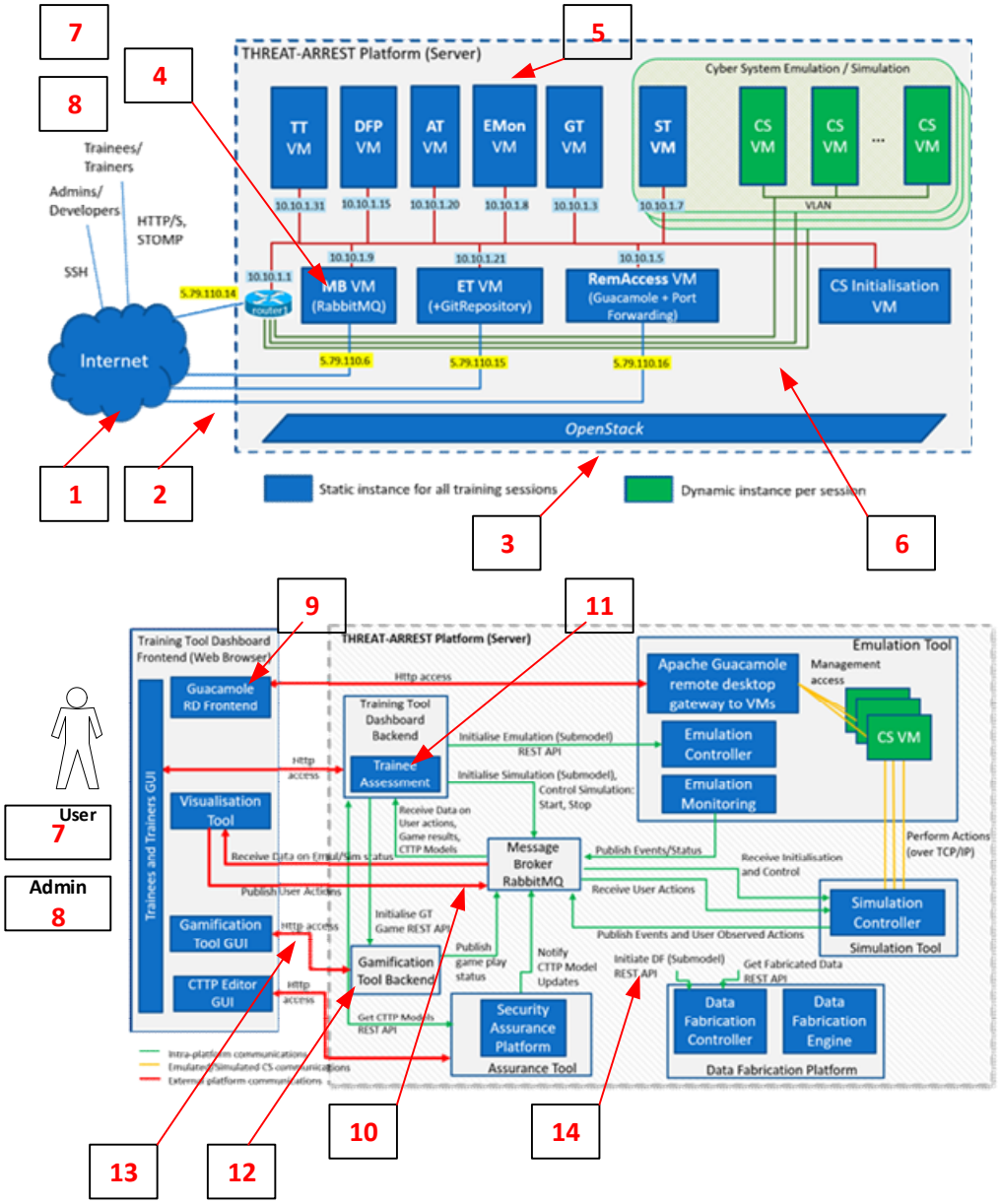

Figure 7: THREAT-ARREST Platform Security Schematics

Figure 7 shows the main aspects subject of security protection. Fourteen schematics are identified each indicating specific security means (mechanisms) considered to address a given platform component, function or user of the platform. Table 7 details the security schematics, the components or functions they apply, the security mechanisms/actions taken, and the security properties they imply on the components or functions. They complement each other and altogether define the security posture of the platform.

Table 7: THREAT-ARREST Platform Security Overview – Components/Functions vs Security Mechanisms vs C/I/A/Auth Properties

#	THREAT-ARREST Platform Component / Function	Security Mechanisms	C	I	Av	Auth
1	THREAT-ARREST Domain	TLS Certificate server side (all users' connections via HTTPS).	☑	☑		☑
2	(Public) Cloud Provider / Perimeter	Perimeter FWs / IDS / WAF DDoS protection	☑	☑	☑	
3	OpenStack Installation	Secure Configuration / Latest version / patches	☑	☑		☑

#	THREAT-ARREST Platform Component / Function	Security Mechanisms	C	I	Av	Auth
		Whitelisting / ACLs Disabling IP Forwarding Change default ports for known services (avoid DoS/brute force attacks on known SSH ports)				
4	Core VM (Ubuntu)	Secure Configuration / Latest OS version / patches Server Hardening (*see section 3.4.2)*	☑	☑		
5	Platform Tools VMs deployment	Secure VM configuration / OS version / patches Server Hardening (as above) Other SW installed / secure configuration (e.g., change default credentials for RabbitMQ Broker)	☑	☑		
6	THREAT-ARREST Platform Network deployment	Tools Monitoring (EMU tool)	☑	☑		
7	Users web access	HTTPS (TLS) User Authentication for all platform external communications Role-based user access rights, Strong Password Policy	☑			☑
8	Admins access	SSH, SFTP	☑			☑
9	Guacamole RD Frontend (Remote Access VM)	HTTPS connection between frontend and Gateway	☑			☑
10	RabbitMQ communication / message exchange	RabbitMQ TLS certificate installation for VT – RabbitMQ communications	☑	☑		☑
11	Training Tool Database	Secure Configuration (version / patching, users / Database ports)	☑	☑		☑
12	Gamification Tool Database	Secure Configuration (version / patching, users / Database ports)	☑	☑		☑
13	External Platform Communications (Frontend – Backend)	HTTPS, STOMP over TLS	☑	☑		☑
14	THREAT-ARREST Tools REST APIs	All externally accessible REST APIs are offered over TLS (HTTPS) with server-side certificate authentication. User authentication and JWT token access to all externally accessible APIs	☑	☑		
15	THREAT-ARREST Tools initialization (form Training Tool)	JWT		☑		☑

C = Confidentiality, I = Integrity Av = Availability Auth = Authentication

3.4 Protection Mechanisms

Following the above-mentioned analyses, we deploy the related defenses and tackle the common threats that were applicable to our platform.

For the current version of the platform, we hire a bare mental server from the cloud provider LeaseWeb. The provider applies the *cloud-related defenses*, including an intrusion detection mechanism that continuously monitor our server as well as a web-application firewall where we can set some main filtering and access control policies.

Then, we gain access to the server and install the core Operating System (OS), an Ubuntu 16.04 LTS. We apply a list of *server hardening policies* (e.g. software update/upgrade, Password rules, Firewall, Anti-virus, Rootkit scan, disable Telnet, etc.), setting the main security configurations. Afterwards, we install OpenStack in this OS and securely configure the hyper-visioning environment (e.g. whitelisting, IP forwarding policies, change the default protocol ports to avoid automated attacks, etc.).

The process is repeated for every deployed Virtual Machine (VM). For each tool, we acquire a VM through OpenStack. The tool owner performed the related secure configuration of the VM (the aforementioned server hardening strategies) and secured the application software that was installed afterwards. For example, for the installation of the RabbitMQ broker, we first set the main protection mechanisms and policies for the related VM and then configured the broker itself as well (i.e. change the default credentials for the pre-install user accounts 'admin' and 'guest').

After establishing the core security mechanisms, we safeguard the user-related access and information. These mainly involve:

i) a ***role-based access control*** policy, where each user type has specific access rights and privileges for data-in-process,
ii) a ***secure external communication channel with HTTPS***, which protects the data-in-transit, as well as,
iii) the ***encryption of the user's private data*** within the platform (i.e. in the Training Tool database), which shields the data-in-rest and fulfils one of the technical aspects for GDPR compliance.

Finally, in order to verify that all the security mechanisms are in place and work properly, we plan to perform a rigorous penetration testing analysis (pentest). The pentest will also verify that the abovementioned identified threats are successfully mitigated. In the current phase, we have set the pentest environment and performed an initial examination on a low-scale analysis. The full testing will be completed along with the final distribution of the platform and the full integration of the underlying components and mechanisms.

The next subsections detail all these concepts.

3.4.1 Security by Infrastructure Provider

LeaseWeb is an Infrastructure-as-a-Service (IaaS) provider offering dedicated servers and cloud services. It deploys the core and state-of-the-art *cloud-security* solutions and provides *secure housing*, which are top-tier ISO certified. Also, it operates an *intrusion detection system (IDS)* that continuously monitors all servers, clouds, and attacks on their public IPs. Moreover, it offers us a *Web Application Firewall (WAF)*[6] with which we can securely create web

[6] LeaseWeb Web Application Firewall: https://kb.leaseweb.com/products/cyber-security/web-application-firewall/getting-started-with-web-application-firewall

applications and services, and administrate the server's users (invite users, set access policies, etc.). The next figure depicts the management dashboard.

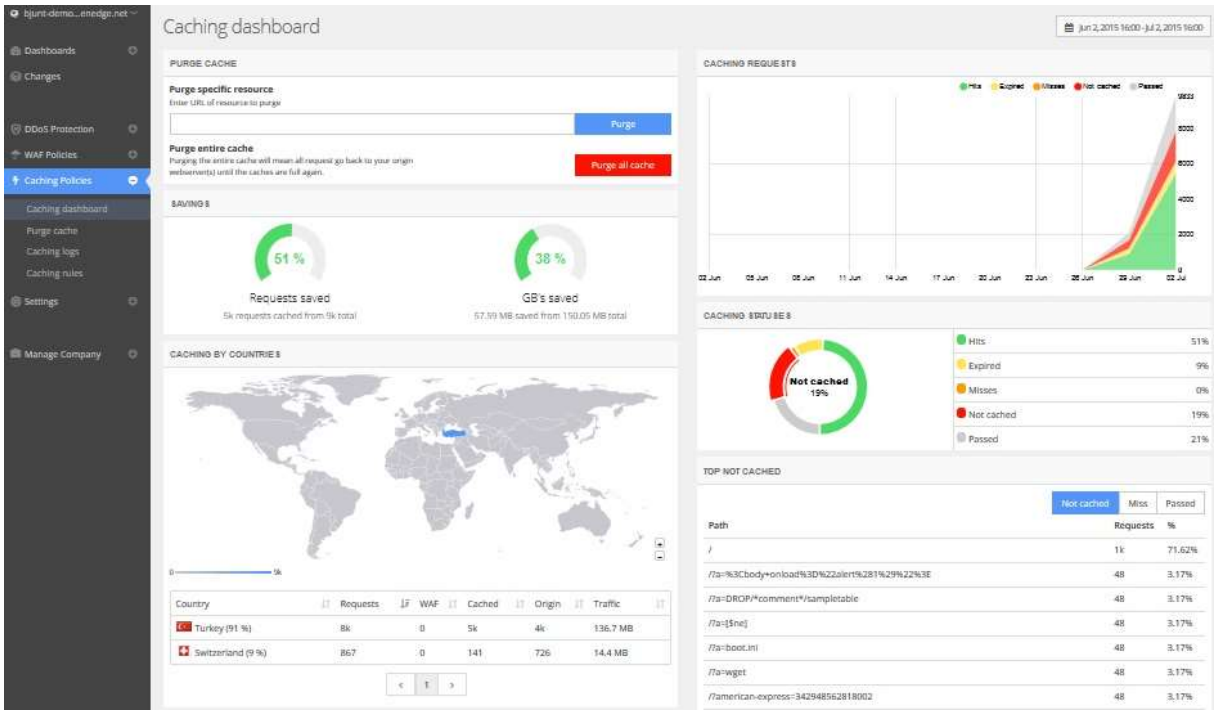

Figure 8: The WAF Dashboard

Backup services are enabled, enhancing the recovery features of the THREAT-ARREST platform. The functionality is supported by the Acronis Backup[7] software, while the backup server is located in a different country than the working server. Specifically, the bare metal server is installed in Netherlands and the backup server in Germany.

3.4.2 Core VM Setting – Ubuntu

As a central computational resource, we have to take into consideration the security of the server. First of all, we start by performing a series of *server hardening* policies which mainly include the following *20 interventions*:

- update and upgrade the utilized packages and the operating system
- remove unnecessary packages
- detect weak passwords and update
- verify that no accounts have empty passwords
- set password rules
- disable USB devices
- secure any Apache server running in this machine
- examine which services start at boot time in order to verify that there are no malicious services starting with booting and running in the background
- delete all world-writable files
- configure *iptables* to block common attacks, like SYN flooding and spoofing
- install *Logwatch* to monitor suspicious log messages
- install and configure the Uncomplicated Firewall (UFW), which is the main solution for Ubuntu
- secure configuration of SSH

[7] Acronis Backup: https://www.acronis.com/en-us/tutorials/AcronisBackup/

- disable telnet
- secure configuration of *sysctl* to prevent the main flooding attacks and IP spoofing
- lock user accounts after some failed login attempts
- use *netstat* and check for hidden open ports
- set root permissions for the core system files
- install *chkrootkit* and scan for rootkits
- install the open source antivirus ClamAV and scan for viruses

The specific commands are documented in the installation guidelines under the related deliverable "D6.2 – Initial installation and usage guidelines for the THREAT-ARREST platform".

3.4.3 Deployment of THREAT-ARREST Tools

After secure configuration of the server, we proceed with the installation and secure setting of OpenStack[8] (refer also to Table 7 schematics #3). Then, we deploy through OpenStack a VM for each of the main Tools and platform components. The following VMs are acquired:

- Tools
  - Training Tool (TT VM)
  - Assurance and CTTP modelling Tool (AT VM)
  - Emulation Tool (ET VM)
  - Simulation Tool (ST VM)
  - Gamification Tool (GT VM)
  - Data Fabrication Tool (DFP VM)
- Components
  - Remote Access (RemAccess VM) for internal IP forwarding
  - Message Broker (MB VM)
  - Emulation Monitoring (EMon VM)

For each one of them, we perform the server hardening policies, as with the case of the core VM. Henceforth, we start installing application software and setting its security mechanisms. This mainly includes:

- The secure configuration of the RabbitMQ in the MB VM
- The secure configuration of the databases in the Training and Assurance Tools (TT VM and AT VM)

More details can be found in the installation guidelines under the deliverable "D6.2 – Initial installation and usage guidelines for the THREAT-ARREST platform".

Given the server hardening procedures and OpenStack configuration settings[9], security for the internal network of the Tools has been considered without additional communication security means such as encryption and integrity for instance by means of TLS.

3.4.4 User Security & Privacy

Except from the main security configuration of the main platform components, we also need to implement protection mechanisms to preserve the user's overall security. First of all, the communications between the user and the platform is protected by HTTPS, offering confidentiality and integrity. The authentication of the user is materialized with the use of passwords. Therefore, a strong password policy is enforced for the creation of this secret

[8] OpenStack security guide: https://docs.openstack.org/security-guide/
[9] https://docs.openstack.org/security-guide/networking/services-security-best-practices.html

information. Once authenticated, the user's access rights and privileges are determined by his/her role.

Moreover, the THREAT-ARREST platform must preserve the user's privacy and be compliant with GDPR. In the deliverable "D8.10 – Interim Legal framework monitoring & THREAT-ARREST alignment report", we defined a legal monitoring framework that is supported by the platform. There, we also document the protection mechanisms that are mandatory in order to be compliant with the legislation. From a technical aspect, this mainly includes the maintenance of the personal data in an encrypted form by the Training Tool.

Other privacy policies, which will be fully supported in the final version of the platform, will ensure the principles of:

(i) lawfulness, fairness and transparency,
(ii) purpose limitation,
(iii) data minimization,
(iv) accuracy,
(v) storage limitation, and
(vi) accountability.

Also, the overall defense mechanisms must facilitate the management of a potential data breach.

In the current phase, we are delivering the first integrated version of the platform. Therefore, at the moment we have not process any personal data. Afterwards, we will proceed with the pilots' set up and the evaluation under the piloting environment. Prior the use of the platform by real trainees and trainers, we will have also to follow the Ethics Plan that is documented under the "WP10 – Ethics requirements" deliverables. Thereupon, we inform the user about the processing of personal data and grant his/her consent.

3.5 Pen Test and Vulnerability Analysis Methodology

It is very important not only to set the security parameters but also to verify the results. In order to do that we will conduct penetration test and vulnerability analysis of the Threat Arrest platform. We will start with visibility of public ports/services, that part will be done without direct access to THREAT ARREST internal network. For testing and exploitation itself we will utilize a temporary virtual machine in the internal network in order to avoid any detection mechanism that may reside on path from the pentester towards the tested servers i.e. IDS system of the cloud provider. This is a general best practice as it avoids unnecessary alerts and possible needless investigation.

Though main focus of the testing should still be those services, that are publicly accessible, testing of local services should not be neglected. That is because once an attacker obtains a foothold inside a local network, local network services become available for lateral movement.

Proper pentest should also consider act of a malicious user. That means that not only the login fields should be tested, but also any input available to authenticated user. Regularly there is not enough time to properly test every input field (any data that are sent from the user that the user can change) so it is up to the time possibilities and pentester to choose which fields should be tested more thoroughly.

Our plan is to scan the internal network with nmap[10] to find live hosts and open ports. After we utilise its version scan which we will compare with the actual versions and check for known vulnerabilities within CVE database (i.e. cvedetails.com). Known vulnerabilities will then be

[10] https://nmap.org/

included in the final report and the most promising ones will be tested (exploited if successful). For that we will use not only metasploit[11] but also other available sources.

For web application penetration testing, we will use Burp[12] and sqlmap[13]. We will try to overcome login screens without using valid credentials and furthermore we will test other inputs that can be altered by the attacker.

Other tools and methods might be used based on future needs.

[11] https://www.metasploit.com/
[12] https://portswigger.net/burp
[13] http://sqlmap.org/

4 Requirements Addressed

We will present how the tool's requirements specified in D1.2 (THREAT-ARREST D1.2, 2018) have been addressed in the first version of the integrated platform. A short description of the means by which a requirement is addressed is provided supporting the status of the given requirement. We define the following five indicative categories of a requirement's status: i) *Fulfilled = 100%,* ii) *Partially Fulfilled ≥ 50%*, iii) *In Progress < 50%*, and iv) *Not Started* until M20=0%.

Out of 76 tools' requirements, 46 requirements have been addressed in the first platform version (with 18 fulfilled and 28 partially fulfilled), while 13 requirements are defined in progress, and 17 requirements not started. Table 8 lists all tools' requirements along with their status and a short description of the means they are addressed.

Table 8: THREAT-ARREST Platform Requirements Status

Req-ID	Description	Status	Means (how addressed)
Assurance Tool			
AT_R_01	Provide the CTTP Models and Programmes Specification Language	Fulfilled	The CTTP Models and Programmes Specification Language is responsible for constructing the CTTP Models. The language was described in D3.1 and finalised in D3.2.
AT_R_02	Provide the CTTP Models and Programmes Specification Model Tool	Fulfilled	The CTTP Models and Programmes Specification Tool is responsible for creating and parsing the CTTP Models to the Training Tool. The tool was described in D3.2. A web editor was also provided.
AT_R_03	Provide the CTTP Models and Programmes Specifications	Partially Fulfilled	The initial CTTP Models and Programmes Specifications were described in D3.3. The final version of the deliverables will be included in D3.5.
AT_R_04	Provide support for the monitoring of all security properties of the target cyber-system and the emulated/simulated versions of it used in CTTP training programs, as long as the latter can be monitored	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_05	Provide support for the monitoring of actions of trainees, who are also users of the target cyber-system, that are related to security properties of the target actual cyber system (e.g. compliance to security restrictions)	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_06	Provide support for monitoring security-related actions of trainees against the target cyber-system before and after the	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the

Req-ID	Description	Status	Means (how addressed)
	training to enable an evaluation of the effectiveness of the training		CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_07	Provide support for monitoring conditions related to assessing the level of compliance of the trainee actions to expectations set by the security assurance sub-model of the CTTP model, as extracted by the CTTP model translation	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_08	Provide support for security properties assessment from both the actual targeted cyber system and the simulated/emulated versions of it used in training	In progress	The Core CTTP model identified the actual targeted cyber system assets. Each asset will be assured for one or more -identified- security properties.
AT_R_09	Support the collection of assurance assessment evidence and make it available to other layers of the THREAT-ARREST platform	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_10	Support the monitoring of conditions involving events collected from different layers of the THREAT-ARREST platform	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_11	MUST be configurable and support user authentication and authorization	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_12	Provide a set of assurance assessment support administration functions, including the retrieval of the collected assurance assessment evidence (i.e., events) and the specification of rules to be used for security assurance assessment	In Progress	The Core CTTP model identified the actual targeted cyber system assets. The rules that will be used for the assessment will be based on the identified assets.
AT_R_13	Create and store a trace for each administration access to the tool and the associated actions (e.g. changes in settings, access of logs)	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_14	Provide the following assurance assessment functions: specification of the	In progress	The Core CTTP model identified the actual targeted cyber system

Req-ID	Description	Status	Means (how addressed)
	target cyber system to be assessed, specification of the monitoring and testing interfaces that may be used for assurance assessment, specification of conditions regarding trainee actions to that need to be monitored, specification of restrictions regarding the accessing of evidence collected through the assessment process		assets. These assets will act as a starting point for creating the functions described in AT_R_11
AT_R_15	For each monitoring session, store the primitive monitoring events used for assurance with a clear record of their producers, contents and their time of occurrence; and the results of the checking of monitoring conditions of different types against these events (e.g. cyber system security monitoring rules, trainee actions monitoring rules)	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_16	Produce auditable assurance assessment results, including digital certificates (where appropriate), based on the evidence collected	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
AT_R_17	Provide audit functions to allow for the review of the assurance tool functions and configuration integrity checks	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The monitoring component will be part of the final integration.
Simulation Tool			
ST_R_01	Allow the definition of simulation scenarios consisting of relevant network components by parameterizing scenario templates predefined for training	Fulfilled	Simulation sub-model from the assurance tool is used for defining and initializing the simulation scenario. See deliverable D3.3 for v1 of the simulation submodels and D3.5 for v2 of those. See D5.2 for v1 for simulated components definition and D5.6 for v2.
ST_R_02	Offer a library of simulated network components (modelling their structure and required behaviour)	In Progress	A set of components as required by the pilots were defined any implemented. This library will be extended considerably in the second project phase. Refer to deliverable D5.2 for v1 of simulated network components and D5.6 for v2.
ST_R_03	Components in the component library should include actors in a training scenario (attacker, defender, user) as well as relevant communication network/IT components; their behaviour will be	Fulfilled	Components of all described classes were defined; behaviour can be modelled in an event-based or process-based fashion (see discrete event simulation "world views"). Refer to

Req-ID	Description	Status	Means (how addressed)
	specified primarily by rules describing their reactions to relevant input events		deliverable D5.2 for v1 of simulated components and D5.6 for v2.
ST_R_04	Allow scenario templates to be defined using, connecting and parameterizing components from the simulation library	Partially Fulfilled	A scenario template is maintained in a git repository. Furthermore, the definitions of the current pilot scenarios can be used to create new simulation scenario. Refer to deliverable D5.2 for v1 of simulated components and D5.6 for v2.
ST_R_05	Allow the creation of a simulation run given a simulation scenario definition	Partially Fulfilled	Simulation runs are controlled by the software component "Simulation Controller". It reads requests via the message broker and is currently used by the training tool to configure/initialize a simulation run. Currently the template-based method as defined for the CTTP Simulation Sub-model is implemented. This will be extended in the next project phase to also implement the "from scratch" method. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2.
ST_R_06	Allow triggering actions/events in the emulation component	Partially Fulfilled	In the smart home and healthcare scenarios the simulation is used to interact with LSE's gateway component as well as a PostreSQL database (both provided as emulated components). Refer to deliverable D5.2 for v1 of simulated components and D5.6 for v2.
ST_R_07	Receive and act upon events received from emulation	Partially Fulfilled	see ST_R_06
ST_R_08	Import and use synthetic and real event logs	In Progress	A first version of an event log reader was implemented. It will be extended in the second project phase as the data fabrication platform is integrated more advanced training scenarios requested by the pilots. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2.
ST_R_09	Provide real-time information to users of the system about the current state of the simulation (usually displayed via the visualization component)	Partially Fulfilled	The simulation provides a mechanism to subscribe to state changes in the simulation. Using this mechanism any interested component (in particular the

Req-ID	Description	Status	Means (how addressed)
			visualisation tool) can receive update messages in an asynchronous way via the message broker. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2. Refer to D4.1 for v1 of the Visualisation Tool and to D4.8 for v2.
ST_R_10	Receive and process user input (interactive simulation)	Fulfilled	User actions can be triggered by e.g. the visualisation tool. They are sent to the simulation tool via the message broker. The simulation components get notified about such events and react appropriately. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2. Refer to D4.1 for v1 of the Visualisation Tool and to D4.8 for v2.
ST_R_11	Alter the behaviour of simulated components/networks based on user input	Fulfilled	User input is received by the simulation tool using via a dedicated broker exchange. Messages send there are read by the simulation tool and forwarded to the respective simulated component so it can react to it appropriately. Refer to D6.1 for v1 of platform architecture and message broker communications. Refer to D5.4 for v1 of simulated components network execution module and D5.7 for v2. Refer to D4.1 for v1 of the Visualisation Tool and to D4.8 for v2.
ST_R_12	Synchronize simulation time with emulated components and training session progress	In Progress	Simulation time currently can be defined in the CTTP simulation sub-model to start at arbitrary predefined data/time values. It can progress synchronized to real time (“wall time”) using a certain real-time factor or as quickly as possible. In the second project phase also more advanced schemes with simulation times “jumping” for both simulated and emulated components will be implemented as required by the pilots. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2.

Req-ID	Description	Status	Means (how addressed)
ST_R_13	Ensure repeatability and randomness. Every execution of a scenario, using basic configuration with the same input, should produce the same results. At the same time, some randomness should be ensured by modifying the initial configuration/input, in order the results not be identical	Fulfilled	A random seed can be specified in the CTTP Simulation sub-model. When this seed is set to a particular value, then simulation runs are exactly repeatable. Using "-1" as a special value will always use a different random seed for each training execution. Refer to deliverable D5.4 for v1 of simulated components network execution module and D5.7 for v2.
Emulation Tool			
ET_R_01	Emulation sub-model of CTTP model will drive the definition of the emulated network and components	Fulfilled	The Emulation Tool deploys the requested training scenario starting from the XML version of the Emulation CTTP sub-model. The scenario is composed of VMs and network connections. Refer to D2.1 for v1 of the emulated components generator module and to D2.5 for v2. Refer to D3.3 for v1 of the emulation submodels and to D3.5 for v2 of those.
ET_R_02	Align the training process with operational cyber-system security assurance mechanisms	Not Started	During the first integration of the THREAT-ARREST platform, the Assurance tool was responsible for creating the CTTP models and parse them to the tools. The alignment between the training process and the assurance results will be part of the future integration.
ET_R_03	The emulation tool will be enabled to install software and communicate with external physical components as defined in the Emulation sub-model	Fulfilled	The Emulation Tool can install additional software and configure already installed package through the Scripts section of the Emulation CTTP sub-model. Refer to D2.1 for v1 of the emulated components generator module and to D2.5 for v2. Refer to D2.3 for v1 of emulated components interconnection and to D6.1 for v1 of the platform architecture and its network view allowing emulated components connections with external entities.
ET_R_04	Users can interact with the emulated components and their actions are saved in accessible logs. Enable defend and attack actions by individual users and user groups and the logging of these actions.	Partially Fulfilled	The user can access the VMs of the training scenario via Remote Desktop or SSH through the Guacamole gateway. The log of individual user activities is expected in the final release of the ET. Refer to D2.1

Req-ID	Description	Status	Means (how addressed)
			for v1 of the emulated components generator module and to D2.5 for v2.
ET_R_05	Support the interaction with trainees of the CTTP program	Fulfilled	The user can access the VMs of the training scenario via Remote Desktop or SSH through the Guacamole gateway. Refer to D2.3 for v1 of emulated components interconnection and to D6.1 for v1 of the platform architecture and its components' communication view.
ET_R_06	Supply data on components status	Fulfilled	Emulated Components Monitor has been implemented and deployed on the EMon VM. The Monitor provides VM resource readings through a REST API as well as periodically posts the readings to a predefined Message Broker exchange. Refer to D2.2 on emulated components monitoring module.
ET_R_07	Support the propagation of data and other stimuli generated by emulated components to other (simulated or emulated) parts of a cyber-system	Partially Fulfilled	The emulated components interact with the Simulation Tool, where requested by the scenario, and the Training Tool that trigger the deployment of the scenario. A complete integration with the system is expected in the final version of the tool, which will include the support and use of the Message Broker. Refer to D2.3 for v1 of emulated components interconnection and to D6.1 for v1 of the platform architecture and its components' communication view. Refer to D2.4 for v1 of emulated tool interconnection with the other components including the Simulation Tool, and to D2.7 v2 of those.
ET_R_08	Ensure reproducibility. The same configuration with the same input and emulated components should have the same behaviour.	Fulfilled	The same training scenario XML description always deploy the same VMs stack. The deployed VMs are always at their initial status. Refer to D2.1 for v1 of the emulated components generator module and to D2.5 for v2. Refer to D3.3 for v1 of the emulation submodels and to D3.5 for v2 of those.
Gamification Tool			
GT_R_01	Authenticate each user before any action takes place	Partially Fulfilled	The users are authenticated by the Training Tool. The Gamification Tool is accessible

Req-ID	Description	Status	Means (how addressed)
			through the Dashboard. The Gamification Tool creates a session for the individual communication with each user based on a JWT token created by the Training Tool. The first version of the Training Tool is detailed in "D4.4 – Real time trainee performance assessment v1".
GT_R_02	Enforce proof of origin	Partially Fulfilled	In the current version, the Gamification Tool is accessible through the Training Tool. The Training Tool orchestrates the training process and management of users. The Gamification Tool is accessible through a valid JWT token created by the Training Tool. A message broker mediates results of the Gamification Tool with the Training Tool. The verification means are the same as above.
GT_R_03	Provide games that are driven by the threats/assumptions which are specified in the CTTP models	Partially Fulfilled	The instantiation of the games is driven by the CTTP models. For the PROTECT game that is fully integrated in the first version of the THREAT-ARREST platform. In this context, the identifier of the card deck for the scenario is given as input from the gamification CTTP submodel. The games are presented in the deliverable "D4.2 – THREAT-ARREST serious games v1", while the CTTP models are described in the deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1". The next version of games will be detailed in the deliverable "D4.9 – THREAT-ARREST serious games v2" due at M30 and the final CTTP models will be presented in the deliverable "D3.5 – Reference CTTP Models and Programmes Specifications v2" due at M30.
GT_R_04	Evaluate the trainee's performance and provide related input to the	Partially Fulfilled	The Gamification Tool "evaluates" the trainee's

Req-ID	Description	Status	Means (how addressed)
	emulation/simulation components in order to adjust the training process		performance and provides this information directly to the Training Tool. This process is documented in the deliverables “D4.2 – THREAT-ARREST serious games v1” and “D4.4 – Real time trainee performance assessment v1”. The Training Tool can provide input to the other platform components. These interactions are detailed in the deliverable “D4.3 – Training and Visualisation tools IO mechanisms v1”. The underlying components and processes will be finalized in the deliverables “D4.9 – THREAT-ARREST serious games v2” due at M30, “D4.6 – Real time trainee performance assessment v2” due at M28, and “D4.11 – Training and Visualisation tools IO mechanisms v2” due at M30.
GT_R_05	Deploy visualization techniques and cooperate with the visualization tool	Partially Fulfilled	Each game has its own graphical web interface. The interface is integrated in the Dashboard that provides a unified view to the user. The first version of the Visualization Tools is detailed in the deliverable “D4.1 – THREAT-ARREST visualisation tools v1”. The final version of the Visualization Tools will be presented in the deliverable “D4.8 – THREAT-ARREST visualisation tools v2” due at M30.
GT_R_06	Support a cognitive profiling of trainees and measure their familiarity with different security concepts	Partially Fulfilled	Each game has its own trainee evaluation method. The final result for a game reflects the trainee’s familiarity with the specific security concepts of the playing scenario, which is instantiated by the CTTP model. The gamification CTTP submodel includes the scenario identification, while the related security concepts are detailed in the core CTTP model.

Req-ID	Description	Status	Means (how addressed)
			The initial CTTP models are described in the deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1". The final CTTP models along with the cognitive profiling specifications will be presented in the deliverable "D3.5 – Reference CTTP Models and Programmes Specifications v2" due at M30.
GT_R_07	Adjust the type and the level of difficulty of the training process based on the user's profile	Partially Fulfilled	The difficulty level can be adjusted by the activated CTTP model. The related CTTP gamification sub-model is detailed in the deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1". Therefore, the main functionality has been already implemented. The dynamic adaptation of the difficulty level will be documented in the upcoming deliverables "D4.5 – CTTP Programme Adaptor v1" due at M24. The final versions of the related tools and processes will be presented in the deliverables "D3.5 – Reference CTTP Models and Programmes Specifications v2" due at M30 and "D4.10 – CTTP Programme Adaptor v2" due at M30, respectively.
GT_R_08	Support post-training assessments of trainee awareness which are useful in tailoring other forms of CTTP training	Not Started	The metrics for post-training assessment will be defined in the second version of the THREAT-ARREST platform and the full integration of the Assurance Tool in the piloting systems. Then, the trainee's performance in the serious games would be taken into consideration for the adaptation of the related CTTP programmes. These features will be presented in the deliverables "D3.4 – CTTP Models and Programmes Adaptation Procedures" due at

Req-ID	Description	Status	Means (how addressed)
			M24, "D4.5 – CTTP Programme Adaptor v1" due at M24, "D4.7 – CTTP Programme Evaluator" due at M28, "D3.6 – CTTP Models and Programmes Adaptation Tool" due at M30, "D4.10 – CTTP Programme Adaptor v2" due at M30.
GT_R_09	Host several serious games, scenarios and training evaluation mechanisms	In Progress	At the current version, the game PROTECT is fully integrated in the platform. Based on this, we have also implemented 4 main scenarios (1 general for social engineering and 1 per pilot). The game has its own evaluation method (scoring) and part of it can be driven and adjusted by the model. The games are presented in the deliverable "D4.2 – THREAT-ARREST serious games v1", while the initial CTTP models and the scenarios are described in the deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1", and the trainee evaluation mechanisms are detailed in the deliverable "D4.4 – Real time trainee performance assessment v1". Their final versions will be documented in the deliverables "D4.9 – THREAT-ARREST serious games v2" due at M30, "D3.5 – Reference CTTP Models and Programmes Specifications v2" due at M30, and "D4.6 – Real time trainee performance assessment v2" due at M28.
GT_R_10	Develop specific games that are focused on social engineering aspects	Partially Fulfilled	Currently, all the developed games target social engineering aspects. The games are presented in the deliverable "D4.2 – THREAT-ARREST serious games v1", while the initial training scenarios are described in the deliverable "D3.3 – Reference CTTP Models and Programmes Specifications v1". Their final versions will be documented in the deliverables "D4.9 – THREAT-ARREST serious games v2" due at M30

Req-ID	Description	Status	Means (how addressed)
			and "D3.5 – Reference CTTP Models and Programmes Specifications v2" due at M30, respectively.
GT_R_11	Offer games and training suitable for non-security experts	Partially Fulfilled	All three games will support scenarios that will be suitable for non-security experts. At the moment, we have created such scenarios for the games PROTECT. The verification means are the same as above.
GT_R_12	Implement web/mobile application interfaces	Partially Fulfilled	The games PROTECT support web interfaces. A specific user interface is currently under development which is especially designed to be displayed on mobile devices. The initial versions of the games are presented in the deliverable "D4.2 – THREAT-ARREST serious games v1", while the final ones will be detailed in the deliverable "D4.9 – THREAT-ARREST serious games v2" due at M30
GT_R_13	Service many users in parallel	Partially Fulfilled	The games PROTECT can be played by several distinct players. Each player has his/her own session and they do not interact with each other. The verification means are the same as above.
Training Tool			
TT_R_01	Provide means to allow continuous collaboration with the serious gaming tool	Partially Fulfilled	Collaboration with the gaming tool has been established in terms of initiating a game and interpreting the results after the game is finished. Real-time collaboration management between the two components is in progress in collaboration with SEA. Refer to D6.1 on v1 of platform integration and its components' communication view, and to D6.4 for final version of platform integration. Refer to D4.3 for v1 of Training Tool interconnection with the other components including the Gamification Tool, and to D4.11 for v2 of these interconnections.

Req-ID	Description	Status	Means (how addressed)
TT_R_02	Offer a mechanism for real-time performance assessment of the trainees, whilst they undertake CTTP programs	Partially Fulfilled	Real time performance assessment mechanisms have been described and designed; performance assessment is integrated for most modules; real-time updates of the performance assessment is in progress. Refer to D4.4 for v1 of trainee performance assessment and to D4.6 for v2 of the assessment component.
TT_R_03	Provide CTTP program evaluation functionalities, through mechanisms enabling the evaluation of the effectiveness of CTTP programs to inform and enable the continuous improvement of such programs	In Progress	The 1st step has been completed, including (i) the definition of the metrics for CTTP models evaluation; (ii) Quantification of acquired security skills obtained from trainees and their supervisors; (iii) definition of performance measures regarding the undertaking of the CTTP programme and (iv) identification of metrics for the level of compliance of these actions to expectations set by the security assurance sub-model of the CTTP model. Refer to D4.7 for details on the CTTP Programme Evaluator.
TT_R_04	Support and facilitate the dynamic adaptation of CTTP programs, through systematic procedures enabling: (a) dynamic tailoring of CTTP programs to the needs of individual trainees, and (b) continuous improvement of CTTP programs	Not Started	Activity not started yet. To be reported in the next platform release.
TT_R_05	Support synchronous and asynchronous communication between the other THREAT–ARREST components	Partially Fulfilled	Asynchronous communication is achieved with all THREAT-AREST components; synchronous communication is partially achieved through REST API for synchronous communication and through message broker for asynchronous communications. Refer to D6.1 on v1 of the integrated platform and its message broker components' communication, and to D6.4 for v2 of the integrated platform and its communications. Refer to D4.3 for v1 of Training Tool interconnection with the other components, and to D4.11 for v2 of these interconnections.
TT_R_06	Provide means for efficient interconnection with the Assurance, Simulation and Emulation modules	Partially Fulfilled	Achieved through REST APIs and message broker except reception of gameplay progress

Req-ID	Description	Status	Means (how addressed)
			and results from the Emulation Tool. Refer to D6.1 on v1 of the integrated platform and its components' communication view, and to D6.4 for v2 of the integrated platform and its communications. Refer to D4.3 for v1 of Training Tool interconnection with the other components, and to D4.11 for v2 of these interconnections.
Visualization Tool			
VT_R_01	Offer means to connect data sources (simulation, emulation, etc.) to the visual elements and cover all the layers in the implementation stack of the overall THREAT-ARREST platform	Fulfilled	User visualisation tool uses the event notification service offered by the Simulation Tool to subscribe to any values updates only for the elements that are really shown to the user. Additionally, any event messages sent by the emulation tool can be easily subscribed to when defining the visualisation scenario. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool. Refer to deliverable D5.2 for v1 of simulated network components and D5.6 for v2.
VT_R_02	Cover the state of the real and the simulated/emulated cyber system components, the attacks upon them and the effects of user actions	Fulfilled	Using the asynchronous communication mechanism offered by the THREAT-ARREST platform, this requirement is achieved and demonstrated by the demo scenarios. Refer to D6.1 for v1 of the integrated platform and its components' communication. Refer to D4.3 on v1 of interconnection of Visualisation Tool with the Emulation and Simulation tools, and to D4.11 for v2 of these communications.
VT_R_03	Support a web-browser as the primary user interface, while being compatible with many platforms (Windows Client, Web, Mobile Device)	Partially Fulfilled	Currently the Visualisation is accessed via a web browser and is optimized of bigger displays such as those offered by laptops or tablets. Providing native clients to installed on, e.g., Windows machines will be addressed in the next project phase. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_04	Be "integratable" with web-based user-interfaces of other platform components	Partially Fulfilled	Currently the frontend of the visualisation tool is shown in a separate browser tab. In future version of the platform it will

Req-ID	Description	Status	Means (how addressed)
			also be possible to show it directly as part of the Dashboard/Training Tool as an HTML IFRAME element. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_05	Provide means to allow real-time bi-directional communication between platform components (both front-end and back-end) in a cloud/web-based environment	Fulfilled	Implemented using the message broker and STOMP messages. Both the front-end (via a Javascript implementation) and the back-end (using a Java reference implementation) can be used to implement communication. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool. Refer to D6.1 on v1 of the integrated platform and its components' communication view. Refer to D4.3 on v1 of interconnection of Visualisation Tool with the Emulation and Simulation tools, and to D4.11 for v2 of these communications.
VT_R_06	Offer elements to navigate to GUI components of the other platform components	In Progress	Currently implemented by offering links back to Training Tool and to Guacamole to access VMs of the Emulation Tool. Refer to D2.3 on v1 of interlinking emulated components, and to D2.6 for v2 of those.
VT_R_07	Offer real-time updating of visualization elements in response to changes in the connected data sources	Fulfilled	Implemented to show real-time data and monitoring information from the Emulation and Simulation Tools. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool. Refer to D6.1 on v1 of the integrated platform and its components' communication view. Refer to D4.3 on v1 of interconnection of Visualisation Tool with the Emulation and Simulation tools, and to D4.11 for v2 of these communications.
VT_R_08	Support synchronous and asynchronous communication between components	Fulfilled	Asynchronous communication is achieved using the message broker. This means of communication is used for example to update certain visualization elements whenever certain state updates take place in the simulation. Synchronous communication can be implemented on top of asynchronous communication by

Req-ID	Description	Status	Means (how addressed)
			sending action requests on one queue and then waiting on a matching response message on another. This is used for instance when the visualisation tool subscribes to value updates from the simulation. Refer to D6.1 on v1 of the integrated platform and its components' communication view. Refer to D4.3 on v1 of interconnection of Visualisation Tool with the Emulation and Simulation tools, and to D4.11 for v2 of these communications
VT_R_09	Be compatible with SIMPLAN's "Jasima" simulation tool[14]	Fulfilled	Working integration of the Visualisation Tool and the Simulation Tool are demonstrated by the pilot scenarios. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool. Refer to D5.4 on details of v1 of integration of Visualisation Tool with the Simulation Tool, and to D5.7 for v2 of this integration.
VT_R_10	Offer a scenario definition language to describe visualization scenarios usable by simulation and other components	Partially Fulfilled	Visualisation scenarios are defined in textual form primarily using HTML and Svelte component. This allows training developers to use the full power of the Web platform to define visualisations. This approach will be made more user friendly in the next project phase by, e.g., providing a tutorial and more examples of visualisations. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_11	Include Serious Gaming elements in order to increase learning motivation for small and medium groups	In Progress	This requirement was partially addressed in the currently implemented scenarios, but will be more in the focus of the work to be done in the next project phase. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_12	Implement basic visualization principles (expressiveness/ effectiveness/ congruence/ apprehension) and optimize a balance between adequate context and complexity	Not Started	This requirement will be addressed in the next project phase to fine-tune the visualisation views. Refer to D4.8 on v2 of the Visualisation Tool.

[14] Jasima Simulator: https://www.simplan.de/en/software/jasima/

Req-ID	Description	Status	Means (how addressed)
VT_R_13	Use appropriate visualization metaphors for different types of attacks and platform/simulated components	In Progress	While this was already done for the current demo scenarios, more work on this will be performed in the next project phase. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_14	Offer visualizations that can consist of various textual (tables, labels) and graphical elements (various 2D charts; 3D layout views – symbolic visualization of simulation events)	Partially Fulfilled	The current library of visualization components already contains elements for labels, tables, 2D charts and a map view. It will be extended in the next project phase with additional elements including basic 3D visualisations. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_15	Handle big and dynamic datasets and effectively support data abstraction over large numbers of data objects	Not Started	To be implemented in the next project phase as part of more advanced training scenarios. Refer to D4.8 on v2 of the Visualisation Tool.
VT_R_16	Offer elements to allow user interaction and provide means to define scenarios and training sessions	Partially Fulfilled	There are pre-defined visualisation elements that can be used to define user actions. They will trigger actions to be performed dynamically during a simulation run. Visualization scenarios are defined using the Visual Studio Code IDE and converted in a build step to the format usable as the Visualisation Tool frontend. Refer to D4.1 on v1 of the Visualisation Tool and to D4.8 on v2 of the tool.
VT_R_17	Offer hierarchical modelling of visualization views (each containing various visualization elements); a user should be able to navigate in this hierarchy (drill down/zoom up)	In Progress	Visualization scenarios can be defined by main visualization views that are shown as tabs in the visualization. Each tab can consist of an arbitrary number of visualization elements, structured in a hierarchical way. Extensions to this mechanism will be implemented in the next project phase to allow further means to navigate between component, such as drill down/zoom up or a “breadcrumbs” navigation bar. Refer to D4.8 on v2 of the Visualisation Tool. Refer to D3.3 for v1 of the CTTP models and D3.5 for v2 of those.
VT_R_18	Utilize real-time comparative performance measures, scenarios’ reconfiguration and parameters’ adjustment	Not Started	This is an advanced feature that will be implemented in the following months as mandated

Req-ID	Description	Status	Means (how addressed)
			by the pilot scenarios. Refer to D4.8 on v2 of the Visualisation Tool, and to D3.5 for v2 of the CTTP models.
VT_R_19	Be capable of post-process animation of simulation events	In Progress	It is currently already possible to use the value update mechanism and user actions send by the Simulation Tool to trigger animation actions in the visualization tool. This mechanism will be extended to allow more advanced visualisations and animations as required by the pilots in the second phase of the project. Refer to D4.8 on v2 of the Visualisation Tool.

5 Conclusions and Next Steps

We have presented the first version of the integrated THREAT-ARREST platform. The platform has been successfully integrated and deployed on a bare metal server at LeaseWeb.com provider. The main difficulty in the integration and deployment process was the high dependability and interactions among the platform components which introduced the need of more frequent and agile iterations on tools integration and testing.

The platform is already available for demonstration on cyber security training with several training scenarios already available for use. Access to the platform is enabled through the URL https://www.threat-arrest.org.

Based on the first version, three training scenarios have been implemented for the different use cases targeting trainees of different categories and skills. The three training scenarios involve the various platform capabilities such as emulation, simulation, gamification, and demonstrate the potential of the platform to create virtual labs (cyber system training environment) of different complexity in terms of assets emulation/simulation with associated networks and attacks.

There are several objectives of integration remaining for the next stage of the project for the second and final version of the platform. These include:

- *Refinement of components integration and their communication channels*. This includes integration of new version of components and their functionalities, improved message broker integration and communications;
- *Addressing platform scalability for multi-user sessions and multi-scenario training.* This requires efficient and scalable access to and resource management of multiple *simultaneous* virtual lab environments for the different training sessions;
- *Full deployment and integration of the Assurance Tool's capability* in the platform and the required Event Captors deployment in the piloting systems leading to the implementation of the continuous security assurance operation of the platform;
- *Deployment and integration of the Data Fabrication Platform* for dynamic and on-demand fabrication of synthetic security/cyber-attack logs for the needs of training;
- *Deployment of CTTP programme certification and evaluation methods*. This particularly regards the integration of the new and advanced capabilities of the Training Tool planned for the second version of the platform; and
- *Re-evaluation and enhancement of platform' security and usage guidelines*. This regards analysis of security by design mechanisms and performing penetration tests and vulnerability analysis on the platform's services.

The final version of the platform is due M32 reported in "D6.4 – Final Prototype of Integrated THREAT-ARREST platform" with the final installation and evaluation due M36 and reported in "D6.6 – Final Installation and usage guidelines for the THREAT-ARREST platform."

References

GDPR, 2016. European Parliament, Council of The European Union: Regulation (EU), 2016/679 General Data Protection Regulation (GDPR). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679 [26 February 2019]

IBM, 2017 "Create high-quality test data while minimizing the risks of using sensitive production data." *IBM InfoSphere Optim Test Data Fabrication*, https://www.ibm.com/ilen/marketplace/infosphere-optim-test-data-fabrication

Jasima, 2019. Jasima: Java Simulator for Manufacturing and Logistics. SimPlan AG. https://www.simplan.de/en/software/jasima/ [26 February 2019]

OpenStack, 2019. https://www.openstack.org/ [26 February 2019]

RabbitMQ, 2019. RabbitMQ: An open source messaging broker. https://www.rabbitmq.com/ [26 February 2019]

THREAT-ARREST DoA, 2018. THREAT-ARREST Grant Agreement Annex I – "Description of Action" (DoA).

THREAT-ARREST D1.1, 2018. The pilots' requirements analysis report. THREAT-ARREST Project deliverable D1.1, https://www.threat-arrest.eu

THREAT-ARREST D1.2, 2018. The platform's system requirements analysis report. THREAT-ARREST Project deliverable D1.2, https://www.threat-arrest.eu

THREAT-ARREST D3.3, 2020. Reference CTTP Models and Programmes Specifications v1. THREAT-ARREST Project deliverable D3.3, https://www.threat-arrest.eu