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

D1.2: The platform's system requirements analysis report †

Abstract: This deliverable focuses on the analysis of the system requirements of the envisioned THREAT-ARREST integrated platform. Moreover, we update the review of the state-of-the-art and practice on security, simulation, emulation and visualisation and training tools and services, to ensure that the consortium is aware of any new developments of technology in these areas and, therefore, the identified platform requirements are up-to-date and relevant.

Contractual Date of Delivery	31/12/2018
Actual Date of Delivery	31/12/2018
Deliverable Security Class	Public
Editor	*Marinos Tsantekidis (TUBS)*
Contributors	FORTH, STS, UMIL, DANAOS, TUV
Quality Assurance	*Dirk Wortmann (SIMPLAN), Konstantinos Fysarakis (STS)*

† *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.*

The *THREAT-ARREST* Consortium

Foundation for Research and Technology – Hellas (FORTH)	Greece
SIMPLAN AG (SIMPLAN)	Germany
Sphynx Technology Solutions (STS)	Switzerland
Università 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 Universität 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 Sanitaria Pugliese (ARES)	Italy

Document Revisions & Quality Assurance

Internal Reviewers

1. *Dirk Wortmann (SIMPLAN),*
2. *Konstantinos Fysarakis (STS)*

Revisions

Version	Date	By	Overview
1.3	21/12/2018	Editor	Addressed second reviewer's comments
1.2	20/12/2018	Editor	Addressed first reviewer's comments
1.1	19/12/2018	Editor	Added input to the requirements table
1.0	14/12/2018	Editor	Ported requirements to table format
0.9	03/12/2018	Editor	Added TUBS input
0.8	30/11/2018	Stelvio Cimato	Added UMIL input
0.7	14/11/2018	Panagiotis Varelas	Added DANAOS input
0.6	12/11/2018	George Hatzivasilis	Added FORTH input
0.5	07/11/2018	George Lefheriotis	Added TUV input
0.4	04/11/2018	Torsten Hildebrandt	Added SIMPLAN input
0.3	30/10/2018	Editor	Ported to correct template and revised TOC
0.2	29/10/2018	Konstantinos Fysarakis	Added STS input
0.1	27/09/2018	Editor	First Draft

Executive Summary

To ensure the technical soundness and industrial applicability of the THREAT-ARREST approach and training platform, in this deliverable we define and analyse a concrete list of requirements of cyber-systems and attack scenarios, for each of the key components and tools of the project architecture. Furthermore, we research the state-of-the-art and practice on all aspects of the project and present here any new developments of technology to ensure that the consortium is up-to-date. This work is developed under task “T1.2: Platform system requirements and technology updates”.

Table of Contents

1 INTRODUCTION 9

2 OVERALL THREAT-ARREST PLATFORM 10

3 SYSTEM REQUIREMENTS 11

3.1 ASSURANCE TOOL 11

3.1.1 Capabilities 11

3.1.2 Pertinent platform requirements 11

3.1.3 Advancements by THREAT-ARREST 13

3.2 SIMULATION TOOL 14

3.2.1 Capabilities 14

3.2.2 Pertinent platform requirements 15

3.2.3 Advancements by THREAT-ARREST 16

3.3 EMULATION TOOL 17

3.3.1 Capabilities 17

3.3.2 Pertinent platform requirements 18

3.3.3 Advancements by THREAT-ARREST 18

3.4 GAMIFICATION TOOL 19

3.4.1 Capabilities 19

3.4.2 Pertinent platform requirements 20

3.4.3 Advancements by THREAT-ARREST 21

3.5 TRAINING TOOL 21

3.5.1 Capabilities 22

3.5.2 Pertinent platform requirements 22

3.5.3 Advancements by THREAT-ARREST 23

3.6 VISUALIZATION TOOL 23

3.6.1 Capabilities 24

3.6.2 Pertinent platform requirements 24

3.6.3 Advancements by THREAT-ARREST 27

4 NEW DEVELOPMENTS AND PROGRESS BEYOND THE STATE-OF-THE-ART 28

4.1 SECURITY ASSURANCE AND MONITORING 28

4.2 SECURITY TESTING 28

4.3 SIMULATION 29

4.4 EMULATION 29

4.5 E-TRAINING ENVIRONMENTS 30

4.6 VISUALIZATION 31

4.7 SERIOUS GAMES 31

5 CONCLUSION 33

6 REFERENCES 34

List of Abbreviations

CTTP Cyber Threat and Training Preparation

Dx.x Deliverable x.x

Tx.x Task x.x

WPx Work Package x

TRLx Technology Readiness Level x

IO Input Output

IT Information Technology

VM Virtual Machine

SQL Structured Query Language

IoT Internet of Things

CPS Cyber-Physical System

OT Operational Technology

OER Open Educational Resources

InCTF Indian Capture the Flag

DFP Data Fabrication Platform

List of Tables

Table 1 – Assurance tool requirements 12
Table 2 – Simulation tool requirements 15
Table 3 – Emulation tool requirements 18
Table 4 – Gamification tool requirements 20
Table 5 – Training tool requirements 22
Table 6 – Visualization tool requirements 25

List of Figures

Figure 1. Overall THREAT-ARREST platform 10
Figure 2. The Assurance Tool within the THREAT-ARREST platform architecture 11
Figure 3. The Simulation Tool within the THREAT-ARREST platform architecture 14
Figure 4. The Emulation Tool within the THREAT-ARREST platform architecture 17
Figure 5. The Gamification Tool within the THREAT-ARREST platform architecture 19
Figure 6. The Training Tool within the THREAT-ARREST platform architecture 22
Figure 7. The Visualization Tool within the THREAT-ARREST platform architecture 24

1 Introduction

This deliverable is part of WP1 which tackles the issues of the project platform's requirements and design. The main contribution is the analysis of the pertinent platform requirements of the tools used, as well as the identification of any critical updates in terms of available technologies, taken place in the period between proposal submission and project initiation. The output will be fed to T6.1 and will also be used in WPs 2-5.

The goal of any virtual training platform is to familiarize the trainee with the actual system that is virtualized. In order for a training scenario to be effective, it has to be configured in such a way that it includes all the essential views of that system (e.g., actors, procedures, hardware/software, vulnerabilities, threats, possible attacks, etc.), thus making requirements analysis and configuration of the training environment of paramount importance. Significant effort when defining requirements and configuring a training scenario should be spent on the aspect of repeatability through which learning is strengthened. A basic configuration (e.g. in file format) must be used as a starting point for every iteration of the training process. This is important since sessions must be repeatable so that specific outcomes (derived from the identified requirements and basic configuration) can be reached in every execution, while retaining some randomness (derived from fine-tuning the configuration) so that they are not identical. From this stems, of course, the need for configuration management which will be editable in order to provide maintenance capabilities. Different training scenarios will use different configurations which, in turn, will drive different virtual system versions respectively, providing the necessary adaptability in the training platform.

The THREAT-ARREST consortium realizes that in order to ensure the technical soundness and industrial applicability of the THREAT-ARREST approach and training platform, the innovation development program of the project must be driven by requirements of cyber-systems and attack scenarios in domains making use of complex cyber-systems and related implementation technologies. To this end, the system requirements of the envisioned THREAT-ARREST integrated platform need to be defined and analysed.

Thus, this deliverable includes a mapping of the project's key components to specific platform functional requirements, which – in addition to the pilots' requirements analysis report (D1.1) – will provide input to the definition of the THREAT-ARREST architecture. This work will also assist in the initial identification of the exact form of training and simulation models for the pilot scenarios and cyber-systems, which constitutes one of the project's main objectives. Moreover, the document includes an update to the review of the state-of-the-art and practice on security assurance, simulation, emulation, gamification, training and visualisation tools and services, to ensure that the consortium is aware of any new developments of technology in these areas and, therefore, the identified platform requirements are up-to-date and relevant.

The rest of the document is structured as follows: **Chapter 2** presents the overall project platform, giving a short description of the tools that comprise it. **Chapter 3** lists the pertinent system requirements of these tools, providing a description of their capabilities followed by the advancements made by THREAT-ARREST in each one. **Chapter 4** provides an accounting of improvements and progress of the state-of-the-art in key relevant areas for THREAT-ARREST. Finally, **Chapter 5** concludes and links the deliverable content referring to other related tasks/deliverables.

2 Overall THREAT-ARREST platform

The THREAT-ARREST platform will offer training on: **(i)** known and new advanced cyber-attack scenarios, **(ii)** how to make effective and systematic use of different security tools developed to detect and/or respond to cyber-attacks in all the different layers of the implementation stack of a cyber-system, **(iii)** taking different types of actions against cyber-attack. The platform will comprise six key components, as can be seen in Figure 1.

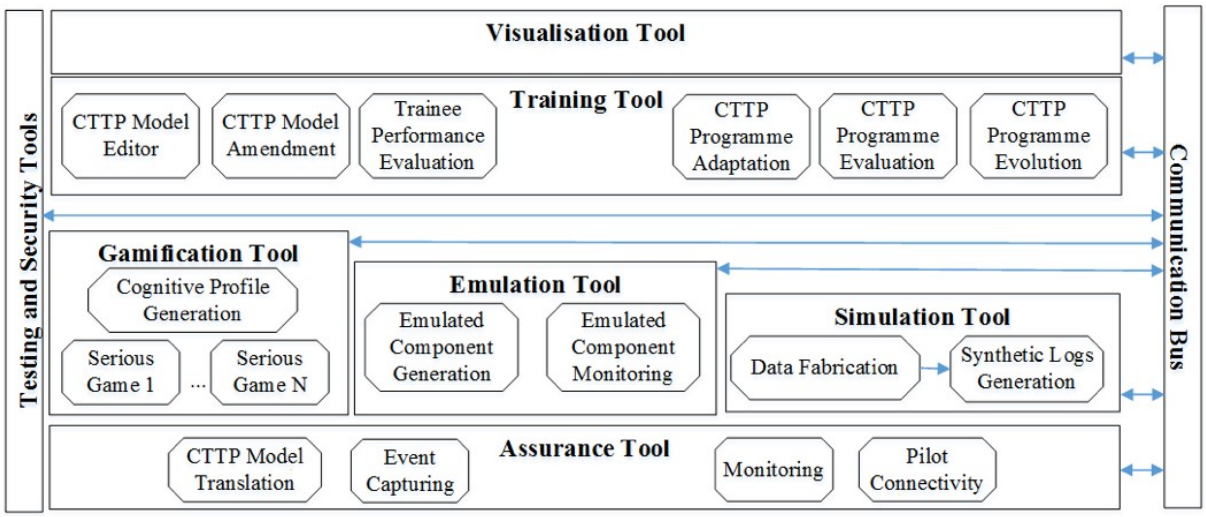

Figure 1. Overall THREAT-ARREST platform

The assurance tool supports the continuous assessment of the security of the cyber-system through the combination of runtime monitoring and dynamic testing, in order to provide information about the status of the actual cyber-system. It also collects runtime system events and generates alerts that provide the basis for setting up realistic simulations.

The simulation tool allows simulating individual cyber-system components and networks of such components to enable the simulation of entire training scenarios defined in CTTP programs.

The emulation platform supports the generation of emulated cyber-system components, in the form of interconnected VMs, equipped with the appropriate software stack. The platform relies on well-established generic machine emulators, possibly open-source [such as QEMU (Bellard 2005), VirtualBox (Oracle 2005), VMWare (VMWare 2018)] to achieve the generation of the emulated components at different levels. Using these frameworks, it is possible to select and emulate different attack scenarios.

The gamification tool hosts various serious games, scenarios and training evaluation mechanisms, which enable trainees to develop skills in being resilient to and preventing social engineering attacks (e.g., phishing, impersonation attacks etc.). The provided games are driven by the threats and assumptions specified in CTTP models (security assurance).

The training tool supports the definition of CTTP models and programs, the presentation of learning material/exercises of CTTP programs, enables trainee actions in response to cyber-threats, interactions with simulated and/or emulated cyber-system components, trainee performance evaluation, CTTP program evaluation and adaptation.

The visualization tool enables the graphical representation of simulations and emulations, the effect of training actions on simulated and emulated systems, as well as the status of the underlying components.

3 System Requirements

3.1 Assurance tool

The tool will be implemented under WP3 (Assurance driven CTTP models and CTTP programs creation) and the related tasks. The content will be documented in two phases with D3.2 (CTTP Models and Programs Specification Tool) and D3.6 (CTTP Models and Programs Adaptation Tool), respectively.

The assurance tool in the context of the THREAT-ARREST platform is highlighted in Figure 2 below.

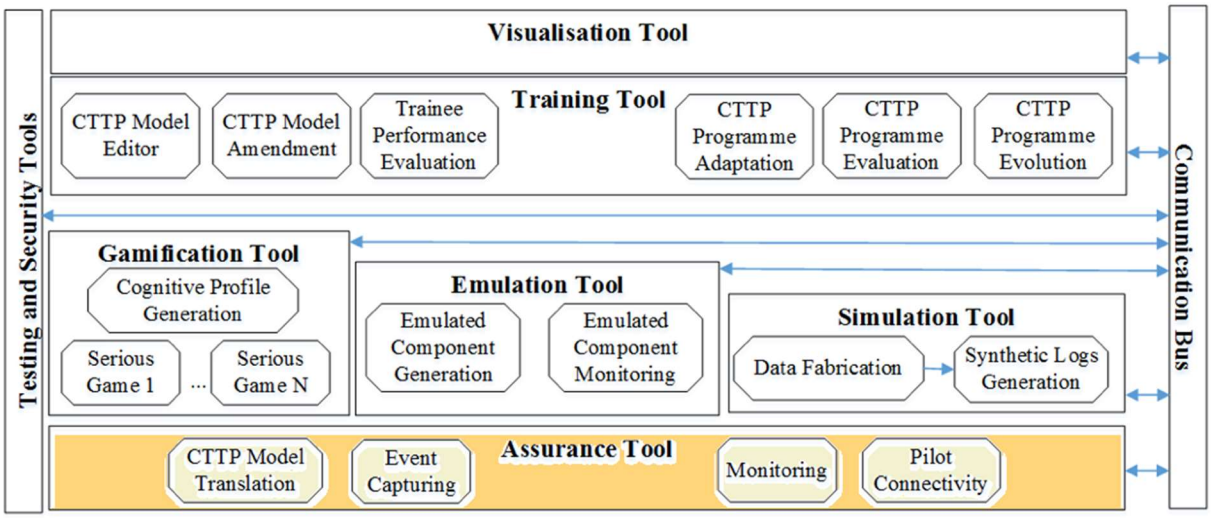

Figure 2. The Assurance Tool within the THREAT-ARREST platform architecture

3.1.1 Capabilities

The assurance tool carries out a continuous runtime assessment of the aspects of the target cyber-system that are important for a CTTP training program. These aspects are defined by the CTTP model (security assurance sub-model) and extracted via the appropriate translation mechanisms. For example, the CTTP model defines the components of the cyber-system that should be monitored, the events of these components that are of importance (e.g., operating system calls, external service calls, user actions) and the conditions that should be satisfied by them. It also defines dynamic system tests that should be executed at runtime and should be combined with monitoring to form hybrid assessments of security. The collected monitoring events and testing outcomes form the operational system evidence that is passed over to the simulation component to enable statistical profiling and thereby the generation of realistic simulations.

3.1.2 Pertinent platform requirements

Some key functional requirements related to the assurance tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below.

Table 1 – Assurance tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
AT_R_01	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	MUST	Simulation Tool (ST_R_06), Emulation Tool (ET_R_06), Training Tool (TT_R_06), AT_R_02
AT_R_02	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)	MUST	Piloting (actual cyber system) environment connectivity
AT_R_03	Provide support for monitoring security-related actions of trainees against the target cyber-system before and after the training to enable an evaluation of the effectiveness of the training	MUST	AT_R_02
AT_R_04	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	MUST	Simulation Tool (ST_R_01), Emulation Tool (ET_R_01), Training Tool (TT_R_02), Gamification Tool (GT_R_03)
AT_R_05	Provide support for security properties assessment from both the actual targeted cyber system and the simulated/emulated versions of it used in training	MUST	Simulation Tool (ST_R_02), Emulation Tool (ET_R_02), Gamification Tool (GT_R_04), Training Tool (TT_R_02), AT_R_06
AT_R_06	Support the collection of assurance assessment evidence and make it available to other layers of the THREAT-ARREST platform	MUST	Simulation Tool (ST_R_06), Emulation Tool (ET_R_06), Gamification Tool (GT_R_04), Training Tool (TT_R_06), Visualisation Tool (VT_R_02), AT_R_02

AT_R_07	Support the monitoring of conditions involving events collected from different layers of the THREAT-ARREST platform	SHOULD	AT_R_06
AT_R_08	MUST be configurable and support user authentication and authorization	MUST	-
AT_R_09	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	MUST	-
AT_R_10	Create and store a trace for each administration access to the tool and the associated actions (e.g. changes in settings, access of logs)	MUST	AT_R_08, AT_R_09
AT_R_11	Provide the following assurance assessment functions: specification of the 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	MUST	Simulation Tool (ST_R_01), Emulation Tool (ET_R_01), Training Tool (TT_R_02), AT_R_06
AT_R_12	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)	MUST	AT_R_06, AT_R_07, AT_R_08, AT_R_11
AT_R_13	Produce auditable assurance assessment results, including digital certificates (where appropriate), based on the evidence collected	MUST	AT_R_12
AT_R_14	Provide audit functions to allow for the review of the assurance tool functions and configuration integrity checks	SHOULD	AT_R_10, AT_R_13

3.1.3 Advancements by THREAT-ARREST

The security assurance platform that will form the core of the assurance tool enables clients to realise security assessments, based on industrial and international standards (e.g., cloud, network), through the use of continuous monitoring and testing. The platform makes use of industrial strength tools including vulnerability and penetration testing tools, and open-source

solutions such as the CUMULUS (CUMULUS, 2012) certification framework. Furthermore, it enables the configuration of security assessment, reporting and certification to the needs of different stakeholders ranging from senior management to external auditors and regulators.

The tool will advance from TRL6 to TRL7 in the course of the project, with a number of advancements that will also enable its effective integration into the THREAT-ARREST solution and the associated requirements (e.g. receiving monitoring input from the training and the various components of the SMART BEAR platform, feeding the realistic simulations). Moreover, the tool will offer customizable security data analytics applied to pertinent data, providing reasoning about the assurance state of the monitored infrastructure. The analytics and intelligence capability will utilize off-the-shelf hardware components coupled with custom software engines to provide a clear upgrade path, without vendor-specific lock-in. This will also require the development of mechanisms to support the connectivity and use of the platform as part of a cyber-threat training framework. These will require mechanisms supporting the implementation of continuous assurance by executing the assurance sub-model of CTTP models, APIs for monitoring/testing evidence and checks, reporting etc.

3.2 Simulation tool

The tool will be implemented under WP5 (Simulation Environment) and related T5.1 (Simulated components' generator), T5.2 (Statistical profiling of real event logs and generation of synthetic events logs), T5.3 (Simulated components network execution), T5.4 (Interconnection with Assurance, Training and Emulation modules). The content will be documented in two phases with D5.1 (Real event logs statistical profiling module and synthetic event log generator v1), D5.2 (Simulated components and network generator v1), D5.3 (The Simulation component IO module v1), D5.4 (Simulated components network execution module v1) and D5.5 (Real event logs statistical profiling module and synthetic event log generator v2), D5.6 (Simulated components and network generator v2), D5.7 (Simulated components network execution module v2), D5.8 (The Simulation component IO module v2), respectively. In the context of the THREAT-ARREST platform, the tool is highlighted in Figure 3 below.

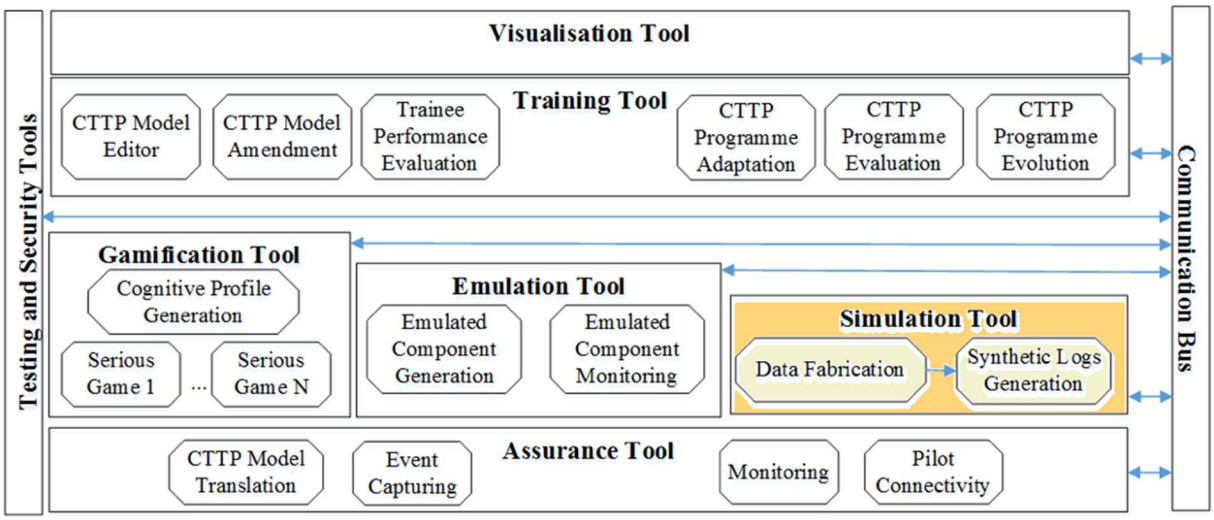

Figure 3. The Simulation Tool within the THREAT-ARREST platform architecture

3.2.1 Capabilities

The simulation tool enables the simulation of individual cyber-system components and actors as well as the networks of such components. The latter is based on propagating synthetic events through the structure of the cyber-system components networks that are to be simulated in different CTTP program scenarios. The simulation scenarios defined will consist of relevant

network components by parameterizing scenario templates predefined for training (e.g. (Hatzivasilis et al., 2017; Hatzivasilis et al., 2014)). These components will include actors in a training scenario (attacker, defender, user), as well as relevant communication network/IT components; their behaviour will be specified primarily by rules describing their reactions to relevant input events. The scenario templates will be defined using, connecting and parameterizing components from a library of components. When a simulation scenario is defined, the tool will create a simulation run. Moreover, the simulation tool exchanges information and commands with the emulation tool in scenarios where parts of the cyber-system are simulated, while others are emulated at the same time. It is also able to receive user input (via the training tool) and alter the behaviour of simulated components and networks based on it.

The tool supports static and dynamic statistical profiling of real cyber-system event logs, captured by the assurance tool, for various components of this system. Based on the detected statistical profiles and the customisation of the Data Fabrication Platform, it creates synthetic event logs, forming realistic simulations of the cyber-system. The generation of synthetic event logs may also be driven by directives defined in the CTTP model and/or community based threat models [e.g., models created based on new threats codified by ENISA (ENISA 2018), NIST (NIST 2018), OWASP (OWASP 2018), SANS (SANS 2018)]. This can create simulations of alternative cyber-system operation scenarios (e.g., scenarios in which attacks with different characteristics occur) as required by CTTP programs.

3.2.2 Pertinent platform requirements

Some key functional requirements related to the simulation tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below:

Table 2 – Simulation tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
ST_R_01	Allow the definition of simulation scenarios consisting of relevant network components by parameterizing scenario templates predefined for training	MUST	ST_R_04, Emulation tool (ET_R_03)
ST_R_02	Offer a library of simulated network components (modelling their structure and required behaviour)	MUST	-
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 specified primarily by rules describing their reactions to relevant input events	SHOULD	Assurance tool (AT_R_01, AT_R_02), Gamification tool (GT_R_01), Emulation tool (ET_R_03, ET_R_04)
ST_R_04	Allow scenario templates to be defined using, connecting and parameterizing components from the simulation library	MUST	ST_R_02

ST_R_05	Allow the creation of a simulation run given a simulation scenario definition	MUST	ST_R_01
ST_R_06	Allow triggering actions/events in the emulation component	SHOULD	Emulation tool (ET_R_04, ET_R_07)
ST_R_07	Receive and act upon events received from emulation	SHOULD	Emulation tool (ET_R_04, ET_R_07)
ST_R_08	Import and use synthetic and real event logs	MUST	Data Fabrication Platform
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)	MUST	Visualization tool (VT_R_01, VT_R_07)
ST_R_10	Receive and process user input (interactive simulation)	MUST	Training tool (TT_R_05, TT_R_06)
ST_R_11	Alter the behaviour of simulated components/networks based on user input	MUST	Assurance tool (AT_R_02), ST_R_10
ST_R_12	Synchronize simulation time with emulated components and training session progress	SHOULD	Emulation tool (ET_R_07), Training tool (TT_R_02, TT_R_05)
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	MUST	ST_R_05

3.2.3 Advancements by THREAT-ARREST

The simulation tool (jasima® - Java Simulator for Manufacturing and Logistics) is a software library for high-speed discrete event simulation. It has been in active development since 2008; it is independent, modifiable and can be used as a basis for advanced simulation-based optimization, e.g., automatic generation of dispatching rules. It is usable for simulation studies and analysis and can be easily integrated in decision support systems. (Current TRL: TRL6)

The simulator will be configured and adopted in order to meet the needs of simulating cyber-threats in general and of the THREAT-ARREST training platform (i.e., simulation of different layers in the cyber-systems implementation stack; interfacing with emulation and training and assurance tools. (Final TRL: TRL7)

Furthermore, the tool will leverage the Data Fabrication Platform (DFP) to create synthetic event logs which are essential in forming realistic simulations of the cyber-system. The DFP is a web-based platform for generating high-quality data for organization-wide testing, development and training. Data are generated from scratch, inflating existing databases or files,

moving existing data and transforming data from previously existing resources, such as old test databases, old test files or even production data. (Current TRL: TRL6)

The DFP will be configured and extended to translate specifications in CTTP models and statistical profiles into DFP rules, in order to enable synthetic event generation for the purposes of THREAT-ARREST. (Final TRL: TRL7)

3.3 Emulation tool

The tool will be implemented under WP2 (Emulation Tool) and related T2.1 (Emulated components' generator), T2.2 (Emulated components' monitor), T2.3 (Interlinking of emulated components), T2.4 (Interconnection with Assurance, Training and Simulation modules). The content will be documented in two phases with D2.1 (Emulated components generator module v1), D2.2 (Emulated components monitoring module), D2.3 (Interlinking of emulated components module v1), D2.4 (Emulation tool interoperability module v1) and D2.5 (Emulated components generator module v2), D2.6 (Interlinking of emulated components module v2), D2.7 (Emulation tool interoperability module v2), respectively.

Figure 4 depicts the emulation tool in the context of the THREAT-ARREST platform.

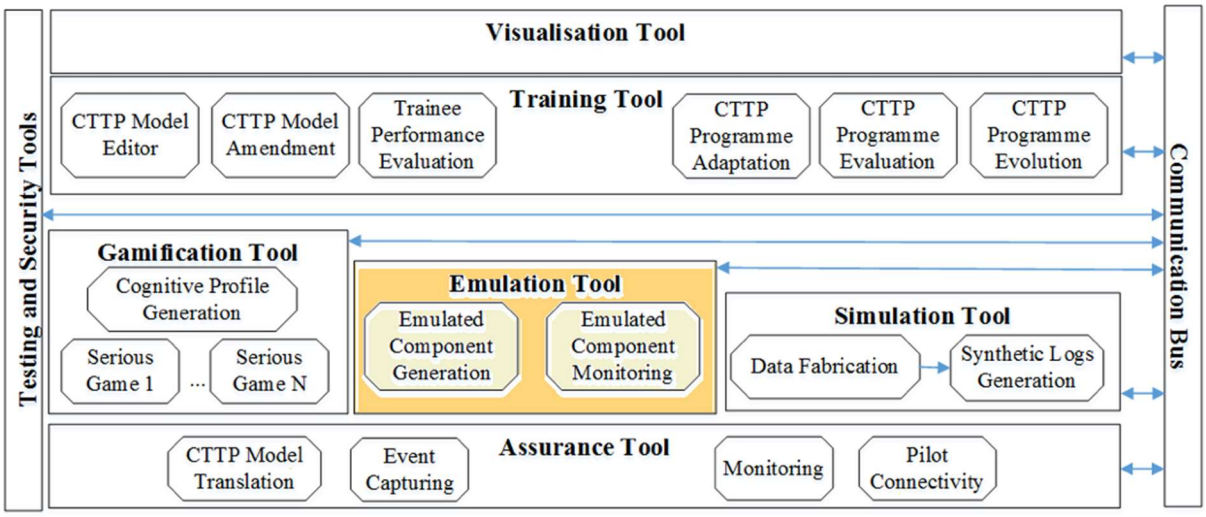

Figure 4. The Emulation Tool within the THREAT-ARREST platform architecture

3.3.1 Capabilities

This tool will enable the generation of emulated components, in the form of VMs. To do it, it will equip the various emulated components with the appropriate software applications, as described by the CTTP model and establish the physical and software level connections between them. These connections will enable the monitoring of the components, registering actions and events and the propagation of data among the other components of the cyber-system. The integration of the interconnected virtual machines will be supported by hypervisors such as OpenNebula project (OpenNebula 2018) and Openstack (Openstack Project 2018).

The emulated components generated by this tool will support interaction with the users, through the training tool and the visualization tool. Users will access information reported by the monitoring tool and will be able to continuously give commands and receive feedback by the emulated components.

3.3.2 Pertinent platform requirements

Some key functional requirements related to the emulation tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below.

Table 3 – Emulation tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
ET_R_01	Emulation sub-model of CTTP model will drive the definition of the emulated network and components	MUST	Simulation tool (ST_R_01)
ET_R_02	Align the training process with operational cyber-system security assurance mechanisms	SHOULD	Assurance tool (AT_R_01)
ET_R_03	The emulation tool will be enable to install software and communicate with external physical components as defined in the Emulation sub-model	MUST	Assurance tool (AT_R_02), Simulation tool (ST_R_03)
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.	MUST	Training tool (TT_R_02)
ET_R_05	Support the interaction with trainees of the CTTP program	SHOULD	Gamification tool (GT_R_13), Training tool (TT_R_05)
ET_R_06	Supply data on components status	MUST	Training tool (TT_R_02), Visualization tool (VT_R_01, VT_R_02, VT_R_08)
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	MUST	Simulation tool (ST_R_06)
ET_R_08	Ensure reproducibility. The same configuration with the same input and emulated components should have the same behaviour.	MUST	Simulation tool (ST_R_13)

3.3.3 Advancements by THREAT-ARREST

The capabilities of the emulation and penetration testing software/frameworks will be combined and expanded to achieve the automated generation and interconnection of emulated cyber-system components. These components will be equipped with the appropriate software stack, enabling the trainees to perform security mitigation tasks. The emulated tool of THREAT-

ARREST will also select cyber-system components and attacks based on CTTP models. (Final TRL: TRL7, the TRL of other capabilities of used tools will remain as is).

Domain specific languages are required for specifying CTTP models and programs and tool support must be given to adapt CTTP models and programs to new cyber-threats and/or changes in the cyber-systems.

3.4 Gamification tool

Even though computer cyber-defence improves, the user is still the weakest link in the security chain (Mitnick and Simon 2007). In this context, serious games constitute an emerging trend in security training that can improve the awareness of the organization's stakeholders regarding operations which are critically dependent on human factors, like cyber-attacks, threat elicitation and organizational defences (Beckers and Pape 2016; Shostack 2014).

Thus, the THREAT-ARREST platform will enhance the cyber-security training by developing the gamification component. The tool will support serious games and enable trainees to engage actively in cyber-defence, learn about attacks and elicit threats. The training process will be driven in CTTP programs and will include the creation of new scenarios, with support for different types of trainees (technical, administrators, security or non-security experts, etc.) and various difficulty levels. Moreover, pilot-specific games will be implemented, improving the cyber-protection of the beneficial stakeholders. These games will be mostly focused on the social engineering aspects of security and will allow the trainees to understand how the examined attacks may be performed and how to tackle them. The goal is to improve the average trainee skills in preventing the overall social engineering attacks by at least 80%.

Figure 5 depicts the gamification tool in the context of the THREAT-ARREST platform.

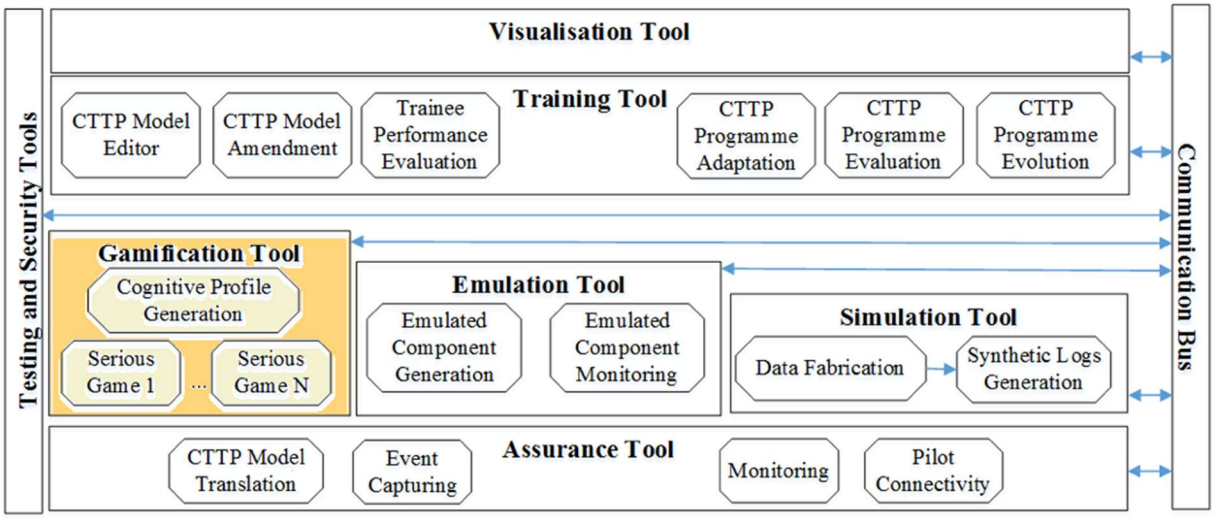

Figure 5. The Gamification Tool within the THREAT-ARREST platform architecture

The tool will be implemented under WP4 (Training and Visualization tools) and related T4.2 (Serious gaming tools). The content will be documented in two phases with D4.2 (THREAT-ARREST serious games v1) and D4.9 (THREAT-ARREST serious games v2), respectively.

3.4.1 Capabilities

Except from providing serious games, gamification will also support an initial cognitive profiling of trainees and measure their familiarity with various security concepts. The profile will be utilized for adjusting the difficulty level and the type of the training process. Furthermore, it will prompt the trainees to participate in serious games which evaluate whether they behave according to the examined security assumptions/policies, based on the security

assurance features in the CTTP models. The performance in these games will determine the evolution of the training course and will influence the defined emulation/simulation processes. Also, the tool will permit the post-training assessments of trainee awareness (in terms of knowledge, attitudes and behaviour) of these types of attacks that will be useful in tailoring other forms of CTTP training.

3.4.2 Pertinent platform requirements

Some key functional requirements related to the gamification tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below.

Table 4 – Gamification tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
GT_R_01	Authenticate each user before any action takes place	MUST	-
GT_R_02	Enforce proof of origin	MUST	-
GT_R_03	Provide games that are driven by the threats/assumptions which are specified in the CTTP models	MUST	Assurance tool (AT_R_01, AT_R_04)
GT_R_04	Evaluate the trainee's performance and provide related input to the emulation/simulation components in order to adjust the training process	MUST	Assurance tool (AT_R_03, AT_R_04, AT_R_06) Simulation tool (ST_R_01, ST_R_03) Emulation tool (ET_R_05)
GT_R_05	Deploy visualization techniques and cooperate with the visualization tool	MUST	Visualization tool (VT_R_01, VT_R_05, VT_R_11)
GT_R_06	Support a cognitive profiling of trainees and measure their familiarity with different security concepts	MUST	Assurance tool (AT_R_11)
GT_R_07	Adjust the type and the level of difficulty of the training process based on the user's profile	MUST	Assurance tool (AT_R_11)
GT_R_08	Support post-training assessments of trainee awareness which are useful in tailoring other forms of CTTP training	MUST	Assurance tool (AT_R_02, AT_R_03, AT_R_11) Training tool (TT_R_01)
GT_R_09	Host several serious games, scenarios and training evaluation mechanisms	MUST	Training tool (TT_R_01, TT_R_05, TT_R_06)

GT_R_10	Develop specific games that are focused on social engineering aspects	MUST	-
GT_R_11	Offer games and training suitable for non-security experts	SHOULD	Training tool (TT_R_01)
GT_R_12	Implement web/mobile application interfaces	SHOULD	Virtualization tool (VT_R_03, VT_R_04)
GT_R_13	Service many users in parallel	SHOULD	-

3.4.3 Advancements by THREAT-ARREST

THREAT-ARREST's ambition is to utilize current solutions in serious gaming that have been applied in the cyber-security domain and extend them by incorporating advanced visualisation tools as well as sophisticated training modules offering automated scenario/level configuration based on real-time assessment techniques and the CTTP models-based approach. The key advancement in THREAT-ARREST will be a model-driven gaming methodology that will be based on assumptions which are defined by security assurance models. The overall process will be further combined with simulation and emulation features in hybrid CTTP programs.

The developed serious games will mainly be applied for training against social engineering security threats and attacks and will be focused at enhancing trainees' ability to resist and mitigate social engineering attacks in realistic cyber-system environments. The development of serious games will adapt and extend the serious gaming tools of SEA (HATCH, AWARENESS QUEST, and PROTECT). These tools will be enhanced with (i) advanced scenarios of cyber threats' mitigation and (ii) new visualisation components. The resulting gamification tool will be at TRL7.

3.5 Training tool

The tool will be implemented under WP4 (Training and Visualization tools) and related T4.3 (Real time trainee performance assessment), T4.4 (CTTP program evaluation), T4.5 (Dynamic adaptation of CTTP programs), T4.6 (Interconnection with Assurance, Simulation and Emulation modules). The content will be documented in two phases with D4.3 (Training and Visualisation tools IO mechanisms v1), D4.4 (Real time trainee performance assessment v1), D4.5 (CTTP Program Adaptor v1) and D4.6 (Real time trainee performance assessment v2), D4.7 (CTTP Program Evaluator), D4.10 (CTTP Program Adaptor v2), D4.11 (Training and Visualisation tools IO mechanisms v2), respectively.

The training tool in the context of the THREAT-ARREST platform is highlighted in Figure 6 below.

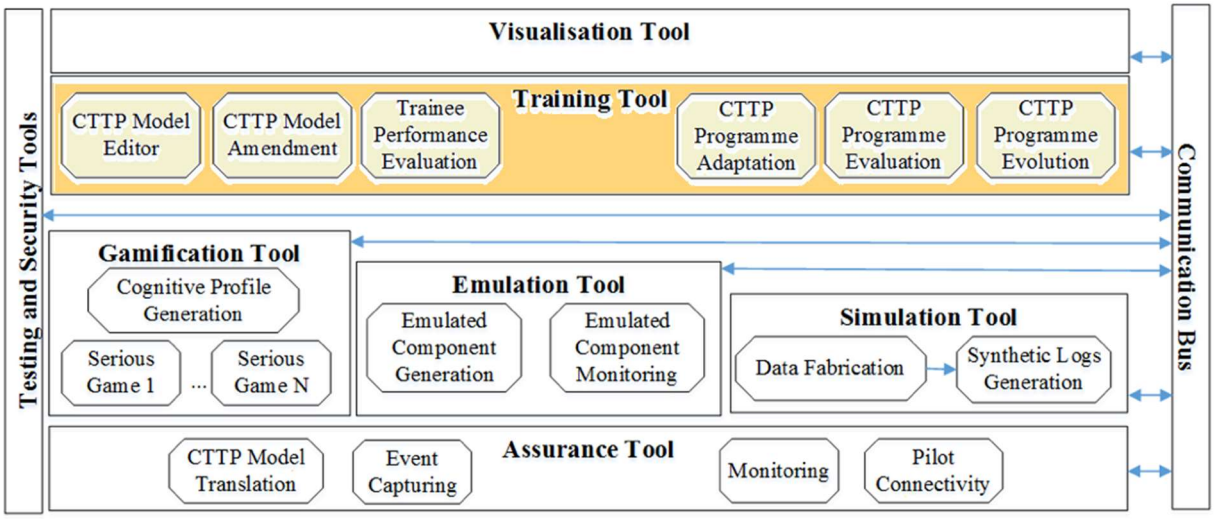

Figure 6. The Training Tool within the THREAT-ARREST platform architecture

3.5.1 Capabilities

Beyond supporting the definition of CTTP models and programs, the training tool will also ensure a high level of interactivity with the trainees and deliver the training scenarios, enabling them to respond, sending the appropriate commands to the emulated and simulated components. Also, it will continuously receive information about the status of the emulation and simulation, evaluating in real-time the state of progress based on the user's responses and their effects on the components and will determine the overall performance of the trainees. The tool will also be responsible for validating the assumptions of the assurance model based on the trainees' responses to the training scenarios and generate warnings in case these assumptions are violated. It will also be able to assess the performance of trainees and evaluate and adapt CTTP programs. Finally, the tool will collaborate with the visualization tool for the effective delivery of training.

3.5.2 Pertinent platform requirements

Some key functional requirements related to the training tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below.

Table 5 – Training tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
TT_R_01	Provide means to allow continuous collaboration with the serious gaming tool	MUST	Gamification tool (GT_R_04, GT_R_08)
TT_R_02	Offer a mechanism for real-time performance assessment of the trainees, whilst they undertake CTTP programs	MUST	Assurance tool (AT_R_01)
TT_R_03	Provide CTTP program evaluation functionalities, through mechanisms enabling the evaluation of the effectiveness of CTTP programs to	MUST	Assurance tool (AT_R_01, AT_R_03, AT_R_04, AT_R_05)

	inform and enable the continuous improvement of such programs		
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	MUST	TT_R_03, Assurance tool (AT_R_04)
TT_R_05	Support synchronous and asynchronous communication between the other THREAT–ARREST components	SHOULD	Assurance tool (AT_R_06), Simulation tool (SR_R_11), Gamification tool (GT_R_04), Emulation tool (ET_R_05)
TT_R_06	Provide means for efficient interconnection with the Assurance, Simulation and Emulation modules	SHOULD	

3.5.3 Advancements by THREAT-ARREST

THREAT-ARREST's ambition is to advance the existing training solutions, by providing additional capabilities in terms of parsing CTTP models and driving the operation of the system's emulated and simulated components. The envisioned e-training environment will also support high level of interactivity with the trainees in terms of (i) real-time assessment and (ii) automated scenarios' adjustment. It will also provide advanced trainee performance evaluation capabilities, including comparisons between actions on simulated/emulated and the real system components.

3.6 Visualization tool

One of the key objectives of the THREAT-ARREST project is to develop key capabilities for the effective delivery of CTTP programs, including the visualization of the operation and state of cyber-systems and the emergence and effects of attacks against them.

Visualization is a key prerequisite for the effective delivery of security training programs, as it enables trainees to better understand the operations and status of the underlying cyber-system, the attacks which have already been launched or are being launched against it and their effects and the impact of any defence actions taken to counter them.

Figure 7 below depicts the visualization tool in the context of the THREAT-ARREST platform.

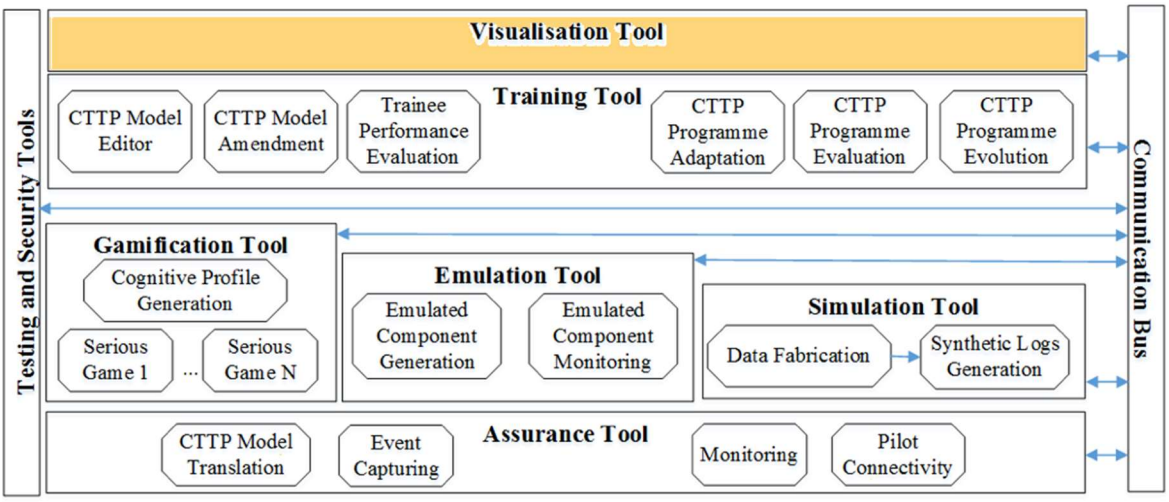

Figure 7. The Visualization Tool within the THREAT-ARREST platform architecture

The visualization tool will be implemented under WP4 (Training and Visualization tools) and related T4.1 (Visualization tools). The content will be documented in two phases with D4.1 (THREAT-ARREST visualisation tools v1) and D4.8 (THREAT-ARREST visualisation tools v2), respectively.

3.6.1 Capabilities

The visualization tool offers a user-friendly visualization environment for the THREAT-ARREST training platform and it will include Serious Gaming elements, in order to increase learning motivation for small and medium groups.

Using the THREAT-ARREST platform's overall capabilities, including the visualization tool, the framework's operators will be able to select the desired training scenarios and tune their parameters, also through the tool's ability to facilitate the creation, parameterization and interaction with the simulation and training platforms.

The visualization tool will be responsible for the representation of the status of the simulated and emulated components and the effects of the training actions on them. It will enable the trainees to have a clear view of the cyber-system status and attacks mounted against it, which will be updated in real-time. It will also provide information about their assessment status and enable real-time interaction. The visualization mechanisms will cover all the layers in the implementation stack of a cyber-system, use appropriate visualization metaphors for different types of attacks and system components, enable zoom-in and zoom-out views over the system, and be interactively controlled by the user.

Concluding, the visualization tool will be able to parse and make use of visualization scenarios contained in or referenced by the CTTP models.

3.6.2 Pertinent platform requirements

Some key functional requirements related to the visualization tool, in the context of its integration into the THREAT-ARREST platform and its corresponding functionality, are listed in the table below.

Table 6 – Visualization tool requirements

Req-ID	Description	Req Level (MUST/ SHOULD)	Dependencies
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	MUST	Emulation tool (ET_R_07), Simulation tool (ST_R_09), Gamification tool (GT_R_05), Training tool (TT_R_06), Assurance Tool (AT_R_06)
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	MUST	Emulation tool (ET_R_05 & 06), Simulation tool (ST_R_09), Gamification tool (GT_R_05), Training tool (TT_R_06), Assurance Tool (AT_R_06)
VT_R_03	Support a web-browser as the primary user interface, while being compatible with many platforms (Windows Client, Web, Mobile Device)	MUST	User Interface / Platform OS
VT_R_04	Be “integratable” with web-based user-interfaces of other platform components	MUST	Gamification tool (GT_R_12), User Interface / Platform OS
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	MUST	Emulation tool (ET_R_07), Simulation tool (ST_R_09), Gamification tool (GT_R_05), Training tool (TT_R_06), Platform OS
VT_R_06	Offer elements to navigate to GUI components of the other platform components	MUST	Emulation tool (ET_R_04), Simulation tool (ST_R_09), Gamification tool (GT_R_05), Training tool (TT_R_06)

VT_R_07	Offer real-time updating of visualization elements in response to changes in the connected data sources	MUST	Simulation tool (ST_R_09)
VT_R_08	Support synchronous and asynchronous communication between components	SHOULD	Emulation tool (ET_R_07), Simulation tool (ST_R_09), Gamification tool (GT_R_05), Training tool (TT_R_05)
VT_R_09	Be compatible with SIMPLAN's "Jasima" simulation tool	MUST	Simulation tool (Jasima) (ST_R_01)
VT_R_10	Offer a scenario definition language to describe visualization scenarios usable by simulation and other components	MUST	Emulation tool (ET_R_07), Simulation tool (ST_R_01), Gamification tool (GT_R_05), Training tool
VT_R_11	Include Serious Gaming elements in order to increase learning motivation for small and medium groups	MUST	Gamification tool (GT_R_09), Training tool (TT_R_01), CTTP program adaptor (TT_R_04)
VT_R_12	Implement basic visualization principles (expressiveness/ effectiveness/ congruence/ apprehension) and optimize a balance between adequate context and complexity	SHOULD	-
VT_R_13	Use appropriate visualization metaphors for different types of attacks and platform/simulated components	SHOULD	Emulation tool (ET_R_02), Simulation tool (ST_R_02)
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)	SHOULD	-
VT_R_15	Handle big and dynamic datasets and effectively support data abstraction over large numbers of data objects	SHOULD	-
VT_R_16	Offer elements to allow user interaction and provide means to define scenarios and training sessions	MUST	Training tool (TT_R_04), CTTP program adaptor (TT_R_04)

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)	MUST	User Interface
VT_R_18	Utilize real-time comparative performance measures, scenarios' reconfiguration and parameters' adjustment	SHOULD	CTTP modelling & CTTP program adaptor (TT_R_04)
VT_R_19	Be capable of post-process animation of simulation events	SHOULD	Simulation tool (ST_R_13), CTTP program adaptor (TT_R_04)

3.6.3 Advancements by THREAT-ARREST

The visualization platform will enable the visualization of simulations and the effect of training actions on simulated systems. The platform will facilitate the creation, parameterization and interaction with the simulation and training platforms. The tool will enable users to parameterize scenarios, trigger simulations, view their outcomes. (Current TRL: TRL5)

The "Jasima" simulation tool will be extended by visualization layers (Web, Mobile Device, Windows Client) for THREAT-ARREST based on existing technology, but as required for presenting the outcomes of simulation/emulation of cyber-system components in the project. (Final TRL: TRL7).

Moreover, the visualization platform will include Serious Gaming elements in order to increase learning motivation for small and medium groups. (Final TRL: TRL7)

4 New developments and progress beyond the state-of-the-art

4.1 Security assurance and monitoring

THREAT-ARREST is not focusing on advancing the state-of-the-art in security assurance and certification. It does aim, however, to connect continuous security assurance with cyber-security training and develop a platform based on synergies between the two. In particular, its vision is to articulate training (CTTP programs) based on security assurance schemes and to use evidence collected from continuous assurance assessment in order to create realistic simulations for CTTP programs while using the continuous monitoring of said assurance schemes to measure the performance of trainees following their training.

In terms of the monitoring elements that will be used to realize the continuous assessment, THREAT-ARREST will deploy the monitoring capabilities of the assurance tool that it will incorporate in its platform and extend them with event capturing and analysis capabilities at the emulation and simulation levels (e.g., user actions, emulated component responses etc.). It will also carry out statistical profiling of monitoring events. These capabilities are important for realising the overall innovative vision of THREAT-ARREST but do not constitute a significant advancement in the state-of-the-art in monitoring as such.

4.2 Security testing

Security vulnerabilities, like buffer overflows, cross-site scripting, SQL injection, and ransomware, are known problems with many related attacks being reported every year. Such vulnerabilities enable the attackers to disclose data and gain unauthorized access to devices. Thus, IT applications vulnerability remediation is now part of compliance checking of major governmental/commercial standards (Bird and Kim 2014).

Security testing (Felderer et al. 2015; Al-Ghamdi 2013) is the process of testing a system's components [e.g. protocols (Morais et al. 2013; Morais et al. 2009), applications (Bessayah et al. 2010; Bird and Kim 2014), networking infrastructure (Phong 2014), databases or other repositories (Raul 2009), etc.] for vulnerabilities that could enable malicious entities to penetrate them and steal confidential information, disrupt the intended functionality, encrypt data, or otherwise cause harm. Security testing is imperative for modern organizations and the main testing methods include:

- Proof verification (Morais et al. 2013; Morais et al. 2009)
- Automated code analysis tools (SmartBear Software 2005-2017)
- Fault injection (Bessayah et al. 2010)
- Fuzzy testing (Raul 2009)
- Penetration testing (Phong 2014)

Currently, most of the individual solutions for security testing target specific system components, such as web services (Salas and Martins 2014; Bessayah et al. 2010) and networks (Phong 2014) and concentrate in specific vulnerabilities for each case. Thus, an organization has to apply several tools and methodologies in order to accomplish a holistic security testing approach (Osterman Research 2016; Bird and Kim 2014). This fact requires higher budget, more educated personnel and increases the dependency for external auditors and evaluators. Factors that make the key decision makers to choose simpler/cheaper, but less effective solutions (Osterman Research 2016).

Moreover, the emergence of the IoT and CPSs has changed the cyber-security landscape. THREAT-ARREST starts from the idea that standard testing distribution is no longer consistent with the vulnerability population in the wild. Indeed, this distribution does not always coincide

with actual incidence data, e.g. as recorded by ZerODium Security (ZerODium Security 2015) and the NIST's National Vulnerability Database (NVD) (Black et al. 2008). Also, while automated security testing has evolved during the last years, further work is needed in order to enlarge the set of vulnerabilities tested by such scanners and include the OT level and the IT/OT interface.

Thus, THREAT-ARREST will progress beyond the state-of-the-art and will come up with new threat/vulnerability models to represent the IT/OT interface in CPSs. The goal is to generate test plans specific for each industrial domain. At the OT interface, the relative importance of detecting specific vulnerabilities is related to the target devices; THREAT-ARREST envisions a scanner with a lower detection rate that will be more effective in cases where the vulnerabilities it detects are individually more severe in the damage their exploits may do.

Therefore, THREAT-ARREST will provide a consolidated solution. Both automated and in-depth analysis will be supported for various system components and operations. Specifically, the platform will support the use of security testing, monitoring and assessment tools at different layers in the implementation stack of a cyber-system including:

- Network layer tools (e.g., intrusion detection systems, firewalls, honeypots/honeynets).
- Infrastructure layer tools [e.g., security monitors, passive and active penetration testing tools (e.g., configuration testing, SSL/TLS testing)].
- Application layer tools (e.g. security monitors, code analysis, as well as passive and active penetration testing tools such as authentication testing, DB testing, session management testing, data validation & injection testing).

Moreover, the proposed platform could be easily extended in order to model upcoming vulnerabilities and threats, enhancing the overall applicability and detection effectiveness.

4.3 Simulation

THREAT-ARREST's ambition is to provide a cyber-system simulation tool, driven by security assurance models. The tool will be able to simulate not only the network (as most of the current simulation tools do) but components of all layers in the implementation stack of a cyber-system, their processing behaviour and security properties.

After consulting SIMPLAN who are the expert partners of the consortium on simulation, we came to the conclusion that the initial literature/tool review on simulation was adequately comprehensive. We did not identify any recent developments in this area that are relevant to the project.

4.4 Emulation

The state-of-the-art in the context of emulation tools includes three main fields: (i) emulation at the network level, (ii) emulation at the host level and (iii) emulation at the platform level.

Network-level emulation tools provide techniques and frameworks for creating virtual nodes and establishing among them virtual networks, where Virtual Ethernet devices are created and installed into nodes and connected to the networks. Since several years, advancements in CPU architectures and networking infrastructures have allowed the rapid development of network emulators, for instance Dummynet (Rizzo 1997), NIST Net (Carson et al. 2003), ModelNet (Vahdat et al. 2002), IMUNES (Puljiz et al. 2006) and W-NINE (Conchon et al. 2009). Network emulation is a combination of real technology and emulation, allowing to run experiments using both real protocol implementations and network models, like for instance in Open vSwitch (Linux Foundation, 2018) and its fork Open Virtual Network.

Host-level emulation tools provide a protected environment to enable the host system to run software and use peripheral devices designed and implemented for the guest system. QEMU (Bellard 2005) is the most common and accepted solution for host emulators, supporting full system emulation in which a complete and unmodified operating system can run in a virtual machine and Linux user-mode emulation.

Platform-level emulation provides integration frameworks for host-level emulation tools, in particular based on QEMU. Many of them are available on the market, each one providing different features and license types: VMware vSphere (VMware 2018), Oracle VirtualBox (Oracle 2018), Xen (Linux Foundation 2018), OpenStack (OpenStack Project 2018) OpenNebula (OpenNebula 2018). Most of them provide users with a model-driven configuration approach. In fact, complex environments composed by virtual machines, network connections, installed software and specific configurations, are all modelled in XML files, characterized by vendor-specific syntax.

At platform level, cyber-range environments have been recently delivered for both educational and commercial purposes (OCCP 2018; Virginia Cyber Range 2018; Cyberbit 2018). Cyber-ranges are virtual infrastructures for managing the entire incident response process and training security professionals in realistic settings.

A further improvement with respect to the state-of-the-art regards the definition of suitable languages for the specification of CTTP models and programs. Such domain-specific languages should allow the description of the cyber-system architecture, the specification of sequences of events that result in cyber-attacks and the specification of the actions that stakeholders are expected to take against cyber-attacks. Moreover, such languages should be able to specify training scenarios (CTTP programs), focusing on threats over specific cyber-system components and should support a mechanism to adapt a given model to new cyber-threats and/or changes in the cyber-systems. This language will be based on schemes developed to specify security assurance models, which define physical and software architecture of the target system (i.e. target of evaluation) and security threats and controls. The language will also need to support the specifications for certain cyber-system components, as well as the CTTP training programs, including targeted components, attacks, security controls, stakeholders and the possible ways of simulating and emulating cyber-system components in order to realize a CTTP program. The definition of the language will be accompanied by the development of a language editor.

THREAT-ARREST will not focus on providing a significant advancement in the state-of-the-art in emulation, however its ambition is to provide a solution enabling the application of CTTP models to real-world emulation tools. In particular, THREAT-ARREST will provide a platform where emulation tools will be able to automate the process of a full-scale physical cyber-system emulation, based on well-defined architecture and security assurance models.

4.5 E-training environments

THREAT-ARREST's vision is to build on existing e-training environments and to advance them by enhancing them with additional capabilities. For instance, THREAT-ARREST will provide existing e-training environments with the ability to parse CTTP models, generated by the assurance tool, in order to drive the operation of the system's emulated and simulated components.

Moreover, THREAT-ARREST aims at enhancing existing e-training environments by providing a high level of interactivity with the trainees; the project's vision is to deploy functionalities that will ensure both real-time assessment of the trainees and automated scenarios' adjustment based on that assessment.

Finally, THREAT-ARREST is expected to advance current e-training solutions by incorporating in the envisioned platform advanced trainee performance evaluation capabilities, including comparisons between actions on simulated/emulated and the real system components.

Overall and in relation to existing e-training environments, THREAT-ARREST will bring to the market a complete solution, comprising advances in numerous technologies and fields (security assurance testing and monitoring, visualisation, serious games, simulation, emulation and training).

4.6 Visualization

THREAT-ARREST's ambition, in relation to the visualization part, is to build on existing techniques, follow an approach using a 2D/3D symbolic visualization and provide post-process animation of simulation events. The overall project's objective is to offer a visualization tool that is well-suited for the application domain of cyber-security training. Thus, the project's goal is not to necessarily advance the state-of-the-art in the field of visualization in general, but to have a useful and effective visualization component for cyber-security training and advance the state-of-the-art specifically for this.

In doing so, THREAT-ARREST's efforts on visualization will enhance existing tools by incorporating advanced interactive capabilities and real-time analytics in terms of performance assessment, scenarios' reconfiguration and parameters' adjustment.

State-of-the-art JavaScript libraries, that offer "low-level" charting capabilities (Hackernoon 2017; 1stwebdesigner 2018) will be used.

4.7 Serious games

Serious gaming focuses on improving the user's engagement/motivation/performance when executing a certain task. Game mechanics and elements are incorporated in the training process, thus making that task more attractive (Pedreira et al. 2015). Over the last years, serious gaming has attracted the interest of industry and academia for cyber-security training.

Schreuders and Butterfield (Schreuders and Butterfield 2016) developed serious games for 'Incident Response and Investigation' – a module covering incident response topics like information security management, network monitoring, log management, intrusion detection, and live/dead disk analysis. They also produce My XP – a novel and open-source virtual gamification learning environment with OER. Although these were developed in tandem and to complement each other, they can also be used independently: for example, computer security topics can be taught in labs without the gamification assessment aspect. Regarding cyber-security and training, Amorim et al. (Amorim et al. 2013) leverage gamification targeted in altering cases of unsafe behaviour. Another approach by K. Boopathi et al. (Boopathi et al. 2015) introduces gaming in the jeopardy round of InCTF. The present Jeopardy round of InCTF can be seen as take-away assignments, where participants are given a set of questions to solve in order to evaluate their knowledge in various security-related issues. Game of Threats (PwC 2018) is a digital game developed by PwC that simulates the speed and complexity of an actual cyber-breach. The solution integrates elements of gamification and game theory to produce an interactive learning experience, where the clients' team tries to defend itself from a threat actors' team (also played by company personnel). The game environment creates a realistic experience where both sides need to take quick and high-impact decisions with minimal information. At its core, Game of Threats is a critical decision-making game that is designed to reward the players which make good decisions while penalizing the teams for taking poor ones. Players end up with a better understanding of the steps they need to perform in order to better secure their organizations.

THREAT-ARREST enhances the current approaches of serious gaming in the cyber-security domain, by incorporating advanced visualisation tools as well as sophisticated training modules which provide automated configuration for difficulty levels and scenarios based on real-time assessment techniques. One key advancement will be the delivery of the model-driven gaming method that is focused on assumptions set by security assurance models, combined with simulation and emulation in hybrid CTTP programs. The developed games will mainly tackle social engineering vulnerabilities. The goal is to enhance the trainee's ability to defend against related attacks under a realistic cyber-system setting. SEA's serious games (i.e. HATCH, AWARENESS QUEST and PROTECT) will be extended for these purposes, adapting the inheriting requirements of the demonstrated THREAT-ARREST's pilots and application sectors.

5 Conclusion

In this early stage of the project, the system requirements of the tools, in addition to the pilots' security requirements, play an important role. The requirements identified and analysed here are essential to the quality and overall applicability of the training platform that will be developed and delivered at the end of the project. Which parts of the target cyber-system are important for a training program? Which are the training scenarios against which the trainees will be evaluated and how will this be carried out? Which of the components of the cyber-system will be emulated and how? Which attacks will be simulated and how? In what way the trainee will have a better understanding of these attacks and their effects on the underlying cyber-system? How will the training program be evaluated? The requirements laid out in this deliverable set the necessary background for providing comprehensive answers to these questions and also form a complete and detailed guide that will direct the future efforts in developing a concrete and comprehensive platform.

To that end, THREAT-ARREST relies on and goes beyond the state-of-the-art and practice on security assurance, simulation, emulation, gamification, training and visualisation. Although the project will feature targeted advancements in these fields, it will also enhance and connect advanced and sophisticated tools and services with cyber-security training and develop synergies among them. Furthermore, it will provide a solution enabling the application of CTTP models to real-world tools that will be able to automate the process of cyber-security training in a dynamic, full-scale CPS.

This deliverable – along with D1.1 – forms the basis on top of which the initial version of the reference architecture for the THREAT-ARREST platform will be developed. It will also be fed to T6.1 as well as WPs 2-5.

6 References

[1] CUMULUS 2012. "Certification infrastructure for multi-layer cloud services project. D2.2 Certification models". Available from: http://cordis.europa.eu/docs/projects/cnect/0/318580/080/deliverables/001-D22Certificationmodelsv1.pdf [24 December 2018]

[2] Al-Ghamdi, A. S. A.-M., 2013. A survey on software security testing techniques, International Journal of Computer Science and Telecommunications, vol. 4, issue 4, pp. 14-18.

[3] Amorim, J. A., Hendrix, M., Andler, S. F. and Gustavsson, P. M., 2013. Gamified training for cyber defence: methods and automated tools for situation and threat assessment, NATO Modelling & Simulation Group (NMSG) Multi-Workshop, MSG-111m Sydney, Australia, pp. 1-12.

[4] Beckers, K. and Pape, S., 2016. A serious game for eliciting social engineering security requirements, 24th IEEE International Conference on Requirements Engineering (RE). IEEE, Beijin, China, pp. 16-25.

[5] Bird, J. and Kim, F., 2014. A survey on application security programs and practices, SANS Institute, pp. 1-24.

[6] Black, P. E., Fong, E. N., Okun, V. and Gaucher, R., 2008. Software assurance tools: web application security scanner functional specification, National Institute of Standards and Technology Std., NIST-SP 500-269, Rev. 1.0, pp. 1-14.

[7] Boopathi, K., Sreejith and Bithin, A., 2015. Learning cyber security through gamification, Indian Journal of Science & Technology, vol. 8, issue 7, pp. 642-649.

[8] Felderer, M., Büchlein, M., Johns, M. et al., 2015. Security testing: a survey, Advances in Computers, Elsevier, vol. 101. pp. 1-51.

[9] Hatzivasilis, G., et al., 2014. ModConTR: a Modular and Configurable Trust and Reputation-based system for secure routing. 11th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'2014), IEEE, Doha, Qatar, 10-13 November 2014, pp. 56-63.

[10] Hatzivasilis, G., et al., 2017. SecRoute: End-to-End Secure Communications for Wireless Ad-hoc Networks. 22nd IEEE Symposium on Computers and Communications (ISCC 2017), IEEE, Heraklion, Crete, Greece, 03-06 July 2017, pp. 558-563.

[11] Mitnick, K. D. and Simon, W. L., 2007. The art of deception: controlling the human element of security, Wiley, pp. 1-372.

[12] Morais, A., Hwang, I., Cavalli A. and Martins E., 2013. Generating attack scenarios for the system security validation, Networking Science, Springer, vol. 2, issue 3-4, pp. 69-80.

[13] Morais A., Martins, E., Cavalli, A. and Jimenez, W. 2009. Security protocol testing using attack trees, International Conference on Computational Science and Engineering (CSE). IEEE Vancouver, Canada, pp. 690-697.

[14] Osterman Research, 2016. Security testing practices and priorities, Osterman Research Inc., pp. 1-15.

[15] Pedreira, O., Garcia, F., Brisaboa, N. and Piattini M., 2015. Gamification in software engineering – A systematic mapping, Information and Software Technology, Elsevier, vol. 57, pp. 157-168.

[16] Phong, C. T., 2014. A study of penetration testing tools and approaches, Auckland University of Technology, Master Thesis, New Zealand, pp. 1-115.

[17] Prensky, M., 2003. Digital game-based learning, Computers in Entertainment (CIE) – Theoretical and Practical Computer Applications in Entertainment, ACM, vol. 1, no. 1, pp. 21–21.

[18] PwC, 2018. Game of Threats, Available from: https://www.pwc.com/us/en/financial-services/cybersecurity-privacy/game-of-threats.html [24 December 2018]

[19] Raul, G., 2009. Case study: experiences on SQL language fuzz testing, 2nd International Workshop on Testing Database Systems (DBTest), ACM, Rhode Island, USA, article no. 3.

[20] Salas, M.I.P. and Martins, E., 2014. Security testing methodology for vulnerabilities detection of XSS in web services and WS-Security, Electronic Notes in Theoretical Computer Science, Elsevier, vol. 302, pp. 133-154.

[21] Schreuders, Z. C. and Butterfield, E., 2016. Gamification for teaching and learning computer security in higher education, USENIX Workshop on Advances in Security Education, USENIX, Austin, USA, pp. 1-8.

[22] Shostack, A., 2014. Threat modeling: designing for security, Wiley, Edition 1st, pp. 1-624.

[23] SmartBear Software, 2005-2017. SoapUI: The Web Services Security Testing Tool, Version 5.4. Available from: http://www.soapui.org [24 December 2018]

[24] Bessayah, F., Cavalli, A., Maja, W., Martins, E. and Valenti, A. W., 2010. A fault injection tool for web services, International Academic and Industrial Conference on Practice and Research Techniques, Springer, LNCS, vol. 6303, pp. 137-146.

[25] ZerODium Security, 2015. Available from: https://www.zerodium.com/ [24 December 2018]

[26] Rizzo L., 1997. Dummynet: a simple approach to the evaluation of network protocols. ACM Comput Commun Rev 27(1):31–41

[27] Carson M, and Santay D, 2003. NIST Net: a linux-based network emulation tool. ACM Comput Commun Rev 33(3):111–126

[28] Vahdat A, Yocum K, Walsh K et al., 2002. Scalability and Accuracy in a Large-Scale Network Emulator. Proc. of ACM/USENIX OSDI 36 (SI), 271-284

[29] Puljiz Z, Mikuc M, 2006. IMUNES Based Distributed Network Emulator. Proc. of 2006 International Conference on Software in Telecommunications and Computer Networks, pp. 198-203.

[30] Conchon E, Pérennou T, Garcia J et al., 2009. W-NINE: A Two-Stage Emulation Platform for Mobile and Wireless Systems. EURASIP Journal on Wireless Communications and Networking 2010: 149075.

[31] Bellard F, 2005. QEMU, a Fast and Portable Dynamic Translator, USENIX Annual Technical Conference, FREENIX Track. Vol. 41. Available from: https://www.qemu.org [24 December 2018]

[32] VMware, 2018. Server Virtualization Software – vSphere. Available from: https://www.vmware.com/products/vsphere.html [24 December 2018]

[33] Oracle, 2018. VM VirtualBox. Available from: https://www.virtualbox.org/ [24 December 2018]

[34] Linux Foundation, 2018. The Xen Project, the powerful open-source industry standard for virtualization. Available from: https://www.xenproject.org/ [24 December 2018]

[35] OpenStack Project, 2018. "Build the future of Open Infrastructure". Available from: https://www.openstack.org/ [24 December 2018]

[36] OpenNebula, 2018. "Flexible Enterprise Cloud Made Simple". Available from: https://opennebula.org/ [24 December 2018]

[37] Open-source Cyber Challenge Platform (OCCP), 2018. Available from: https://opencyberchallenge.net/ [24 December 2018]

[38] Virginia Cyber Range, 2018. Available from: http://virginiacyberrange.org [24 December 2018]

[39] Cyberbit, 2018. Hyper-Realistic Simulated Training. Available from: https://www.cyberbit.com/ [24 December 2018]

[40] Hackernoon, 2017. 9 Best JavaScript Charting Libraries. Available from: https://hackernoon.com/9-best-javascript-charting-libraries-46e7f4dc34e6 [24 December 2018]

[41] 1stwebdesigner, 2018. The Top 3D JavaScript Libraries for Web Designers. Available from: https://1stwebdesigner.com/3d-javascript-libraries/ [24 December 2018]

[42] ENISA, 2018. European Union Agency for Network and Information Security. Available from: https://www.enisa.europa.eu/ [24 December 2018]

[43] NIST, 2018. National Institute of Standards and Technology. Available from: https://www.nist.gov/ [24 December 2018]

[44] OWASP, 2018. Open Web Application Security Project. Available from: https://www.owasp.org/index.php/Main_Page [24 December 2018]

[45] SANS, 2018. Information Security Training. Available from: https://www.sans.org/ [24 December 2018]

[46] Linux Foundation, 2018. Open vSwitch. Available from: https://www.openvswitch.org/ [18 December 2018]