Model-Driven Cyber Range Training: A Cyber Security Assurance Perspective

Iason Somarakis(✉), Michail Smyrlis, Konstantinos Fysarakis, and George Spanoudakis

Sphynx Technology Solutions AG, Zug, Switzerland
{somarakis,smyrlis,fysarakis,spanoudakis}@sphynx.ch
http://www.sphynx.ch

Abstract. Security demands are increasing for all types of organisations, due to the ever-closer integration of computing infrastructures and smart devices into all aspects of the organisational operations. Consequently, the need for security-aware employees in every role of an organisation increases in accordance. Cyber Range training emerges as a promising solution, allowing employees to train in both realistic environments and scenarios and gaining hands-on experience in security aspects of varied complexity, depending on their role and level of expertise. To that end, this work introduces a model-driven approach for Cyber Range training that facilitates the generation of tailor-made training scenarios based on a comprehensive model-based description of the organisation and its security posture. Additionally, our approach facilitates the automated deployment of such training environments, tailored to each defined scenario, through simulation and emulation means. To further highlight the usability of the proposed approach, this work also presents scenarios focusing on phishing threats, with increasing level of complexity and difficulty.

Keywords: Cyber Range training · Model driven engineering · Security assurance

1 Introduction

The insufficient knowledge of security procedures and the lack of security awareness across different types of employees within an organisation, combined with the rapid technological advancements (e.g., 5G, the Internet of Things - IoT) [1] that transform various domains (e.g., energy, health-care), provide fertile ground for various threat actors (sophisticated or otherwise) to carry out successful attacks that may significantly damage tangible and intangible assets [2]. Organisations own or access a vast number of cyber systems that can be exposed through numerous known and unknown attack vectors; as organisations advance technologically, the complexity of their systems and their security

A. P. Fournaris et al. (Eds.): ESORICS 2019 Workshops, LNCS 11981, pp. 172–184, 2020.
https://doi.org/10.1007/978-3-030-42051-2_12

further increase. Nevertheless, the security awareness and security expertise of employees typically, does not increase at the same pace. This is especially critical for organisations that handle sensitive data (e.g., hospitals [3]) or are part of critical infrastructures (e.g., smart energy grids [4]). Therefore, to protect their assets and mitigate potential attacks, such organisations need to train their employees to appropriately respond to the security challenges of this era. This includes, educating them with the latest learning resources that will allow them to comprehend the security related changes introduced by the new technologies and giving them access to training scenarios that realistically represent situations that may occur in their organisation. In this manner, Cyber Security training that is not explicitly designed to fit the requirements of an organisation and does not have the ability to easily adapt to the rapidly changing landscape, is insufficient and quickly becomes obsolete. Thus, the importance for a dynamic and continuously up-to-date cyber security training environment emerges.

Motivated by the above, this work aims to highlight the potential of model-driven Cyber-Range training that: (i) is applicable to any type of a system; (ii) is able to represent the actual assets of an organisation and generate training scenarios based on them; (iii) offers scalability and adaptability, by enabling adjustments to the model as the organisation evolves; (iv) is up-to-date regarding threat intelligence, considering new vulnerabilities discovered [5]) or changes to the organisation's setup (e.g., adding new systems that may introduce new vulnerabilities).

The remainder of this paper is organised as follows: Sect. 2 presents an overview of the background, related works and how the proposed approach overcomes the limitations of current commercial solutions; Sect. 3 describes the adopted security assurance methodology; Sect. 4 provides a detailed model of an example scenario and two variations; and finally, Sect. 5 summarises the paper and sets future goals.

2 Background and Related Work

This work is based on the definition of a security assurance model, adopting and extending state of the art approaches in model-driven cyber assurance and certification, simulation, emulation, and e-training cyber range tools and platforms.

The security assurance's focus is to evaluate ICT systems, products and services with regard to security standards and security properties [6]. To achieve this, the proposed approach follows certification schemes such as CUMULUS [7], an open source model driven framework, capable of executing automatically different types of certification schemes for cloud services. It was introduced to close the gap of automation that other certification frameworks lacked (STAR [8], ECSTA [9]). In this work, cyber training leverages the continuous security assurance enabled by the assurance model to use its elements (e.g., Threats, Security Controls, monitoring evidence) and create realistic simulations for CyberRange training programmes, while monitoring the assurance schemes to measure the performance of the trainees following training.

To cover the needs for the implementation of the training environment several tools for simulation and emulation can be considered to support automatic deployment of the emulated components and facilitate the communication across simulated and real assets (see Sect. 3.2). Regarding the simulation requirements, the Cyber-Range sub-model needs to be able to accommodate a detailed representation of simulation environments and its components in order to support automatic generation of the simulation demands of the training programme; thus, several open source discrete event-driven simulators were examined. The NS-3 [10] provides support of TCP, routing and multicast protocols over wired and wireless networks and has the ability to run software on simulated models. GNS3 [11] is an open source network simulator mainly focuses on Cisco and Jupiter software. Netkit [12] is a command-line based simulator tool that uses user-mode linux to create network nodes. Finally, OMNet++ [13] is another open source discrete event simulator that offers a highly scalable and modular framework primarily for building any-kind of network (e.g., wired, wireless, on-chip) simulators. OMNet's community is vast providing domain-specific support for sensor networks, wireless ad-hoc networks, internet protocols, performance modelling, photonic networks etc. Considering the emulation requirements, two major virtualisation tools are OpenStack [14] that features deployment and management of virtual machines and Docker [15] that uses an engine to host containers of virtualised software.

Considering external sources for keeping the security assurance model up to date with changes in the threat landscape, various established cyber security threat and vulnerability lists can be considered; e.g., OWASP [16], ENISA [17], NIST [5]. Additionally, state of the art research efforts such as project CIPSEC [18] can provide valuable insights on personnel training courses, know-how on forensics analysis tools and education for protection against cascading effects.

Furthermore, various products established in the market of Cyber Training must be considered in order to identify gaps and needs in the domain. Kaspersky Interactive Protection Simulation (KIPS) [19] targets senior managers and decision makers to increase their security awareness by offering 6 scenarios (i.e., Corporation, Bank, eGovernment, Transport, Power Station, Water Plant) with related types of attacks. The Adaptive Awareness Portal [20] offers modular means for building your own training programmes but it emphasises on security awareness training, social engineering scenarios and e-learning management. Sophos Phish Threat [21] is another phishing training solution that utilises phishing simulations to educate and tests its end users. Inspired e-learning's Security Awareness Training [22] is a role-based solution educating against phishing scenario via a combination of videos and immersive situation-based role-playing scenarios. Finally, literature [23,24] indicates that the gamification of cyber range training offers promising results. This approach is followed by PwC's Game of Threats [25]; a solution that simulates cyber security breaches and uses gamification and game theory to provide a realistic game environment for an interactive blue team/red team experience. While there are various solutions in the market, most offer a fixed number of scenarios, role/domain specific limitations, minimal automation, and often lack the interaction with actual emulated cyber environments, thus lacking in realism.

3 Security Assurance Modelling

The proposed model-driven approach to Cyber-Range training is based on the definition of a Security Assurance Model that enables the systematic representation of the target organisation, its assets and their relations, and, ultimately, its security posture. This comprehensive approach allows us to identify and describe the assets of the system, their relations and their corresponding threats; the sequence of events that leads to the manifestation of these threats, alongside the responsible threat actor/s; the actions that trainees are expected to take against these attacks and the tools that may be used for this purpose; targets regarding the preparedness and effectiveness level that the trainees targeted by a Cyber-Range training programme are expected to achieve and how these levels may be measured in different stages of the delivery of the programme; and, finally, information on how the system can simulate and emulate the components necessary for its implementation. Additionally, it supports the effortless integration of potential changes in the composition of the organisation to the model (e.g., hiring a person, introducing a different job role, acquiring a software or hardware, removing old hardware, the disclosure of new vulnerabilities etc.); and enabling the generation of updated or brand-new Cyber Range training scenarios driven by these changes. To support the training, the core model is extended with the Cyber-Range sub-model that provides training relevant information; this allows it to build custom training scenarios for known cyber-attacks; new cyber-attacks; learning how to effectively and systematically utilise different security tools; learning the procedure for various types of actions (e.g., preparedness, detection and analysis, incident response, post incident response) and security processes in the target organisations and availability for different types of users of the system (e.g., end-user, administrator, technician, security engineer, blue/red team). This approach allows us to provide automated means for generation of tailored Cyber-Range training programmes that align with the organisation's composition and security requirements. The core assurance model and the Cyber-Range sub-model will be described in the following sections.

3.1 The Security Assurance Model

The core of the defined security assurance model are the organisations' assets (i.e., anything of value to the organisation), as well as the interplay between threats, vulnerabilities, security properties and security controls. For the sake of brevity, a view of the assurance model depicting the above is shown in Fig. 1. In the above, an asset can be a software asset (Software Architecture Layer (SAL) or Physical Architecture Layer (PAL)), a hardware asset, a physical infrastructure asset, data, person or a process. An asset inherits a number of attributes that are grouped into a single element namely the SecurityAssuranceModelElement.

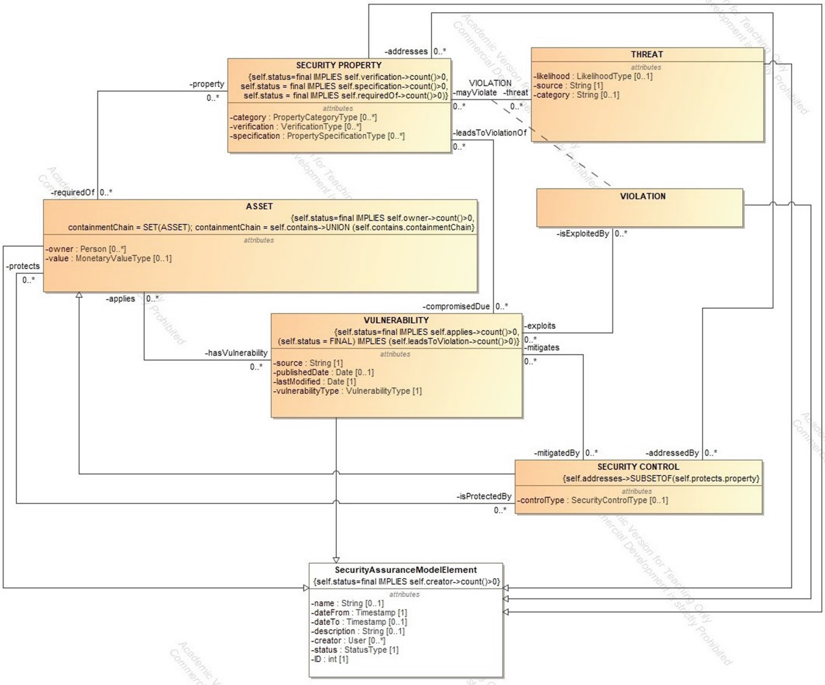
-addresses 0..*
SECURITY PROPERTY
{self.status=final IMPLIES self.verification->count()>0,
self.status = final IMPLIES self.specification->count()>0,
self.status = final IMPLIES self.requiredOf->count()>0)}
attributes
-category : PropertyCategoryType [0..*]
-verification : VerificationType [0..*]
-specification : PropertySpecificationType [0..*]
-property
0..*
VIOLATION
-mayViolate
-threat
0..*
0..*
-leadsToViolationOf
0..*
THREAT
attributes
-likelihood : LikelihoodType [0..1]
-source : String [1]
-category : String [0..1]
VIOLATION
-isExploitedBy 0..*
-requiredOf 0..*
ASSET
{self.status=final IMPLIES self.owner->count()>0,
containmentChain = SET(ASSET); containmentChain = self.contains->UNION (self.contains.containmentChain}
attributes
-owner : Person [0..*]
-value : MonetaryValueType [0..1]
-protects
0..*
-applies 0..*
-compromisedDue 0..*
VULNERABILITY
{self.status=final IMPLIES self.applies->count()>0,
(self.status = FINAL) IMPLIES (self.leadsToViolation->count()>0)}
attributes
-source : String [1]
-publishedDate : Date [0..1]
-lastModified : Date [1]
-vulnerabilityType : VulnerabilityType [1]
-hasVulnerability
0..*
-exploits
0..*
-mitigates
0..*
-mitigatedBy 0..*
-addressedBy 0..*
SECURITY CONTROL
{self.addresses->SUBSETOF(self.protects.property}
attributes
-controlType : SecurityControlType [0..1]
-isProtectedBy
0..*
SecurityAssuranceModelElement
{self.status=final IMPLIES self.creator->count()>0}
attributes
-name : String [0..1]
-dateFrom : Timestamp [1]
-dateTo : Timestamp [0..1]
-description : String [0..1]
-creator : User [0..*]
-status : StatusType [1]
-ID : int [1]

Fig. 1. Cyber Range sub-model UML diagram

The status of this element must be equal to final in order for an asset to have all its attributes/interconnections set. An asset may have security properties and be subject to vulnerabilities. A security property can be of type (a) integrity, (b) confidentiality, (c) availability; it also contains a verification attribute which describes the way a security property is verified and a specification of it; a security property is addressed by security control and may be required by assets. Threat is any circumstance or event with the potential to adversary impact an asset through unauthorised access, destruction, disclosure, modification of data, and/or denial of services; a threat exploits a number of vulnerabilities and violates a number of security properties. Vulnerability is a weakness an adversary could take advantage of to comprise the security properties of a resources; a vulnerability applies to a number of assets and is exploited by a threat which may lead to the violation of a security property if a security control is not in place or properly set up; a vulnerability can either be of physical or computational type. The latter applies to a computational asset (i.e., a Software or Hardware asset) and (mostly) follows the structure provided by National Vulnerability Database (NVD) while the former applies to a physical asset. Finally, a security control protects assets, address security properties and mitigates vulnerabilities. For the sake of brevity, a view of the assurance model depicting the above is shown in Fig. 1.

3.2 Cyber-Range Sub-model

The Cyber-Range sub-model is developed to provide essential information for the specification and implementation of the Cyber-Range training programmes. To accomplish this, the Cyber-Range sub-model extends the security assurance core model in the sense of utilising certain elements from it. Specifically, to generate training scenarios tailored to a target cyber system the cyber range sub-model considers the system's assets, threats, security properties and security controls; information provided by the core security assurance model. For example, if a system isn't prone to phishing attacks, then the cyber range sub-model will not generate a phishing training scenario. Furthermore, to generate different types of difficulty and execution steps of a training scenario, the cyber-range sub model considers information about a person asset (i.e., its role within the organisation). For instance, if an organisation doesn't have any security experts then scenarios targeting this role will not be generated. Additionally, the core security assurance model defines the sequence of events that lead to the manifestation of the threats which is utilised by the Cyber-Range sub- model to drive the different phases of the training scenario. Subsequently the Cyber-Range sub-model defines the threat actors (e.g., external attacker, insider) that cause the aforementioned sequence of events, as it is important for the purposes of the training. The Cyber-Range sub-model extends the network module of the core security assurance model to support virtual networking required for the communication between the emulation's Virtual Machines' (VM); this information (i.e., as a class) can also be of use to the security assurance model to support interactions between virtual systems within the actual cyber-system. Finally, Cyber-Range sub-model uses the organisations software assets to describe the components of the emulation and makes use of the inheritance and other associations with the asset. For example, the containment relationship between assets is utilised by the emulation to describe the link between a VM and its operating system. The diagram of the model is provided in the form of a Unified Modelling Language (UML) class diagram (see Fig. 2) together with a detailed description below.

The Training Programme is specified by a brief **description** of the programme; a measurable **goal** for the training (e.g., in a phishing scenario, the trainee has to identify at least 50% of the phishing emails); **roles** that the programme aims to train (e.g., end-user, administrator, technician, security engineer); **types** of training (e.g., analysis, detection, preparedness, security awareness); **legal frameworks** that align with the programme (e.g., GDPR compliance scenario); a **difficulty** value that indicates the difficulty rating (e.g., a phishing training scenario could be represented very simply to considerably complex).

The training programme covers one or more **Assets**, **Threats** and **Security Properties**; zero or more **Security Controls**. For example, a phishing scenario concerns an end-user asset, covers a phishing specific threat, that involves the confidentiality security property and involve a spam filter as a security control. In this example, if the target organization doesn't employ a spam filter, then it won't be included in the training.

The training programme records zero or more **actual trace** evidence for debugging or training reasons, including system and traffic logs; The training programme sets one or more expected trace to track the progress of a user during its training. For example, an end-user examines a malicious email, the email contains a link, the expected trace is to monitor if the user presses the link or not. In this example, the training programme isn't concerned with any actual trace; however, if this scenario involved a security expert (instead of an end-user) that needs to investigate the origins of the aforementioned link, then the training programme will need to monitor actual trace to follow the user's investigation path (e.g., packets send).

The training programme supports one more **training programme executions**. This class defines the actions enabled for the training programme considering the role of the account undergoing the training (e.g., in a phishing scenario the end user wouldn't have the same actions enabled versus a security engineer or administrator). The **account** class is utilised by one or more **person**, for example, a red team/blue team might have a single account for training their team members. Another example, an account can be used by a security engineer that wants to train on different positions, like system admin, forensics, blue/red team etc. A person can belong to zero or more groups.

The training programme consists of **phases**, that are the stages the programme is deployed; in some cases this class is driven by a **sequence of events** that lead to the specific manifestation of a threat generated by zero or more **threat actors** or in other cases this represent stages of the training scenario (e.g., for example on a blue/red team scenario, on phase 1 the blue team secures the system and on phases 2 the red team tries to exploit it and blue team defends it).

The implementation of the Cyber Range training will be accomplished through emulation and simulation of the components and the interactions among them. In some cases where the training requires an additional level of realism interactions with real assets (e.g., specialised devices like Global Position System (GPS), or actual devices like an email server), will be accommodated as well. The Cyber range model will provide the necessary information and links to resources to support the automated deployment of the playable training programmes via various simulation (e.g., OMNet++) and emulation tools (e.g., OpenStack). This information is described in the simulation and emulation submodels. A training programme may involve more than one simulation and emulation sub-models, considering its deployment phases. For example, a training programme may deploy one virtual machine with a set of configurations for the implementation of its first phase and an additional virtual machine with different configurations for its second phase, or more than one virtual machines for its third phase. Similarly, the simulation model of training programme may deploy a simulation environment on one phase and a different simulation environment on another phase.

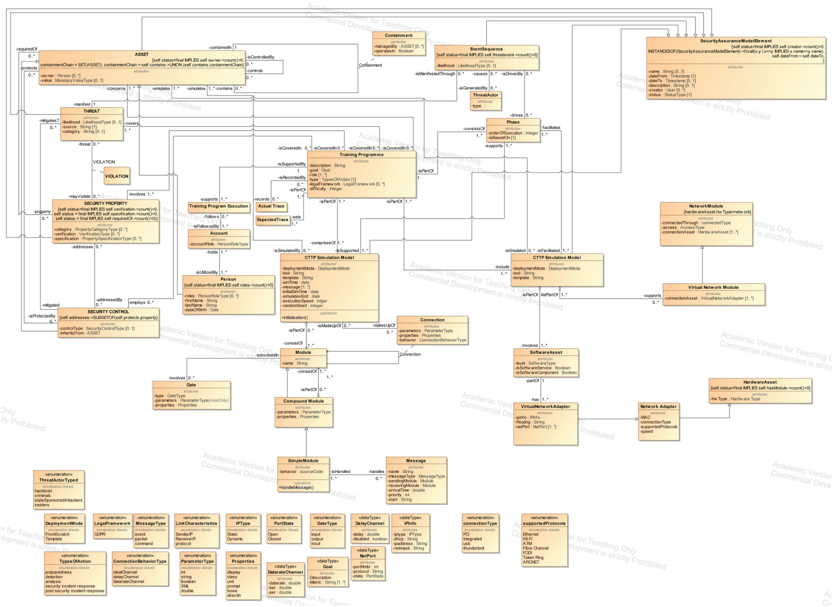

Fig. 2. Phishing scenario emulated and simulated components

Simulation Sub-model. The Simulation Sub-Model is responsible for indicating if a component can be simulated and is describing the simulation environment and its individual components. The simulation model specifies: if an asset can be simulated; the deployment mode for the simulated component (e.g., build from scratch or by using a template); which tool is used to realise the simulation (e.g., OMNet++, NS3); the required, initialisation operations (e.g., instantiation scripts); the time and date that the simulation started and ended; the current simulation and date time; the execution speed that describes how fast the simulation is passing; the random seed value that influences the random operations of the simulation in order to have reproducible randomness; the list of messages that are exchanged among the simulation's modules (e.g., events, commands, packets) during the simulation; Additionally, the simulation model facilitates different simulation environments if required by different phases of the training programme. To describe the composition, the model follows the paradigm of the Network Description (NED) language that is used by the discrete simulator OMNet++ which describes the simulation environment as topology of modules and connections between them. A module is a node in the composition of the simulation; it has a distinctive name, it can be either a simple or compound module and supports communication with other modules via Gates. A **module** is a node in the composition of the simulation; it has a distinctive name, it can be either a simple or compound module and supports communication with other modules via **Gates**.

A **compound module** consists of modules (i.e., simple or compound); it has one or more gates that enable the communication across modules in the compound system and the modules outside of it.

A **simple module** contains source code, specifying the behaviour of a simulated component (e.g., hardware, software), it involves one or more gates to enable the communication with other modules and, it uses operations to handle messages.

Gates can be either input, output and in-out and serve as a connection points between two modules.

A **connection** links two gates and has a specific behaviour, which is defined by characteristics; a variety of characteristics are supported but two main are the data-rate and the delay. All objects of the simulation model can be further specified by a set of parameters and properties.

Parameters are variables that further define an object; variables can hold different simple values (e.g., int, double, boolean, string) or even complex ones (e.g., XML). For example, for a simple module simulating an Application, a set of parameters may define the communication protocol (e.g., UDP, TCP, ICMP), destination address, packet length of message etc.

Properties are various meta-data that can be attached to objects of the simulation model. For example, properties can be statistics that are needed to track the progress of a trainee, such as end-to-end delay, jitter etc. Properties can also be rendering information of the specific object for the GUI.

Emulation Sub-model. The emulation sub-model indicates if a component can be emulated and is responsible for defining the information required by the training programme to emulate components and facilitate connections between them, with simulated components and possibly external real assets (e.g., external email server via the internet). Thus, the emulation sub-model includes information about the resources for instantiating and configuring the various emulated components, the deployment mode for them (e.g., from scratch, template) and the tool that will carry out the emulation (e.g., OpenStack). The resources involve images (i.e., software and OS settings), hardware characteristics (e.g., memory, Central Processing Unit (CPU) cores, storage) and connections details (e.g., internal connection of OpenStack resources or external communication).

The emulation sub-model describes the emulation environment as a structure of one or more software assets and zero or more virtual network modules.

Software assets are a set of programs used to operate computers and execute specific tasks; software can either be SAL or a PAL. SAL is an application later software module (e.g., the sources code of a software implemented within the organisation). PAL is platform level software, which describes an abstract software platform (e.g., a virtual machine, web server, OpenStack). Software asset inherits from asset the containment association; this indicates that an asset can contain or be managed by another asset. A containment can illustrate a deployment relation if an asset is contained in another asset and it operates within the containment. For instance, a software asset can be contained in a hardware asset, a software (SAL) operates in a virtual machine (PAL) and an Operating System (OS) controls the virtual machine.

The **virtual network module**, specifies the network configuration information necessary to support the communication between the emulated nodes, such as IP address, netmask, protocol and routing.

Finally, the emulation sub-model supports one or more phases, supporting the capability to modify the emulated components throughout the training scenario, according to the different phases that the training programme consists of.

4 Sample Scenario

In this section, three variants of training focused around phishing attacks are used to demonstrate the use of the proposed approach. The first is a simple phishing scenario where the trainee analyses a sequence of emails, to identify their legitimacy or malicious intent. The second is more complex, deployed on a realistic environment where the trainee uses their email client to identify and quarantine (i.e., isolate in the spam folder) phishing emails. The third is a capture the flag scenario. This involves the red and blue team and adds another level of complexity. More specifically, the trainer places vulnerabilities on a emulated smart home system, where the blue team tries to secure it and the red team tries to infiltrate it, scoring flags for each vulnerability exploited.

4.1 Simple Phishing Scenario

This is a social engineering scenario that targets trainees with low security expertise. In this simple scenario the user is trained on identifying phishing email attempts. The user logs into the Cyber Range application and is presented with a sequence of emails; their target is to select which of them are legitimate or malicious.

Scenario Modelling. By using the Cyber-Range sub-model the above scenario is specified below. The **Training Programme** class includes a description (i.e., Simple Phishing Scenario), a goal (i.e., identify 50% of the malicious emails), a type (i.e., preparedness), role (i.e., low privilege end-user), difficulty (i.e., 1). The **actual trace** that the training programme records include, system logs, traffic logs and the fake emails that were generated during this scenario. The **expected trace** that the training programme sets include the correct and wrong answers of the user. The **Training Programme Execution** class defines that the only actions allowed to the user is to indicate, via the Cyber Range software, if an email is legitimate or malicious. This scenario has only one **phase**, that is the presentation of the email sequence; when that phase concludes the scenario completed. The **Simulation Model** defines that the deployment mode for this scenario is “from scratch”; the tool that is used to implement the simulation is “Omnet”; the simulation timer starts with 0; messages that contain events; specifically the event that starts the simulation, 10 emails with varying states (i.e., legitimate or malicious), and the event that concludes the simulation; the randomness seed is set to 1; execution speed is set to 1; the starting time and date of the simulation is 20:00 25/8/2019; the end time and date of the simulation is 21:00 25/8/2019. The simulation topology consists of a simple module, an email generator application. The **Emulation Model** defines that the deployment mode for this scenario is "preset"; the tool that is used to implement the emulation is "Open-Stack"; and also provides a path to the preset template "/path/simplePhishingVM". The emulation sub-model consists of one Software-PAL asset (virtual machine) that will contain two Software-SAL assets (1) the operating system "Linux" (2) the simulation software. The scenario's emulated and simulated components are displayed in Fig. 3, while the populated model for this scenario is displayed in Fig. 4.

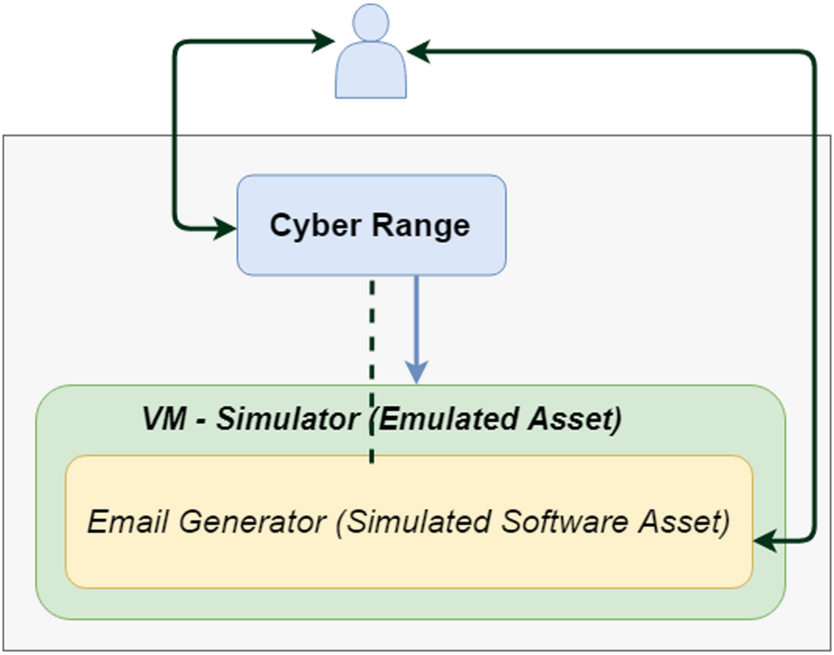
Cyber Range
VM - Simulator (Emulated Asset)
Email Generator (Simulated Software Asset)

Fig. 3. Cyber Range sub-model Object diagram

Besides the aforementioned simplistic scenario, the Cyber-Range sub-model can facilitate more complex and advanced scenarios that will be explored as part of our future work.

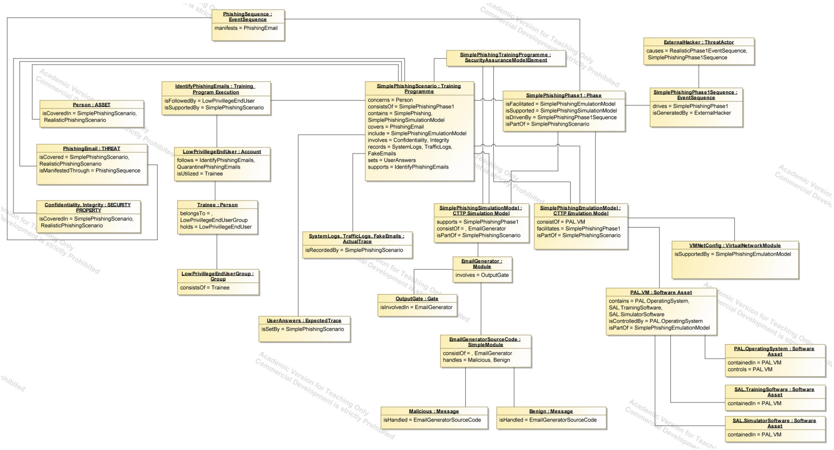
PhishingSequence : EventSequence
manifests = PhishingEmail
Person : ASSET
isCoveredIn = SimplePhishingScenario, RealisticPhishingScenario
PhishingEmail : THREAT
isCovered = SimplePhishingScenario, RealisticPhishingScenario
isManifestedThrough = PhishingSequence
Confidentiality, Integrity : SECURITY PROPERTY
isCoveredIn = SimplePhishingScenario, RealisticPhishingScenario
IdentifyPhishingEmails : Training Program Execution
isFollowedBy = LowPrivillegeEndUser
isSupportedBy = SimplePhishingScenario
LowPrivillegeEndUser : Account
follows = IdentifyPhishingEmails, QuarantinePhishingEmails
isUtilized = Trainee
Trainee : Person
belongsTo = , LowPrivillegeEndUserGroup
holds = LowPrivillegeEndUser
LowPrivillegeEndUserGroup : Group
consistsOf = Trainee
SimplePhishingTrainingProgramme : SecurityAssuranceModelElement
SimplePhishingScenario : Training Programme
concerns = Person
consistsOf = SimplePhishingPhase1
contains = SimplePhishing, SimplePhishingSimulationModel
covers = PhishingEmail
include = SimplePhishingEmulationModel
involves = Confidentiality, Integrity
records = SystemLogs, TrafficLogs, FakeEmails
sets = UserAnswers
supports = IdentifyPhishingEmails
SimplePhishingPhase1 : Phase
isFacilitated = SimplePhishingEmulationModel
isSupported = SimplePhishingSimulationModel
isDrivenBy = SimplePhishingPhase1Sequence
isPartOf = SimplePhishingScenario
ExternalHacker : ThreatActor
causes = RealisticPhase1EventSequence, SimplePhishingPhase1Sequence
SimplePhishingPhase1Sequence : EventSequence
drives = SimplePhishingPhase1
isGeneratedBy = ExternalHacker
SystemLogs, TrafficLogs, FakeEmails : ActualTrace
isRecordedBy = SimplePhishingScenario
UserAnswers : ExpectedTrace
isSetBy = SimplePhishingScenario
SimplePhishingSimulationModel : CTTP Simulation Model
supports = SimplePhishingPhase1
consistOf = , EmailGenerator
isPartOf = SimplePhishingScenario
SimplePhishingEmulationModel : CTTP Emulation Model
consistOf = PAL.VM
facilitates = SimplePhishingPhase1
isPartOf = SimplePhishingScenario
EmailGenerator : Module
involves = OutputGate
OutputGate : Gate
isInvolvedIn = EmailGenerator
EmailGeneratorSourceCode : SimpleModule
consistOf = , EmailGenerator
handles = Malicious, Benign
Malicious : Message
isHandled = EmailGeneratorSourceCode
Benign : Message
isHandled = EmailGeneratorSourceCode
VMNetConfig : VirtualNetworkModule
isSupportedBy = SimplePhishingEmulationModel
PAL.VM : Software Asset
contains = PAL.OperatingSystem, SAL.TrainingSoftware, SAL.SimulatorSoftware
isControlledBy = PAL.OperatingSystem
isPartOf = SimplePhishingEmulationModel
PAL.OperatingSystem : Software Asset
containedIn = PAL.VM
controls = PAL.VM
SAL.TrainingSoftware : Software Asset
containedIn = PAL.VM
SAL.SimulatorSoftware : Software Asset
containedIn = PAL.VM

Fig. 4. Cyber-Range sub-model

5 Conclusions and Future Work

This paper presented a model-driven approach to Cyber-Range training based on the definition of a security assurance model, extending it to facilitate the definition of Cyber-Range training programmes, via the Cyber-Range sub-model. The Cyber-Range sub-model is developed with applicability and scalability in mind, so to offer realistic training scenarios via a hybrid approach of simulation and emulation, which satisfy the security training demands tailored to any specific organisation. While the security assurance model is used to model the organisation as a whole, the Cyber-Range sub-model facilitates the specification, implementation and automatic generation of Cyber-Range training programmes. To this end, the Cyber-Range sub-model links the Assets, Security Properties, Security Controls, and Threats covered in the scenario and defines it, in terms of training information (e.g., description of the scenario, goal), simulation (e.g., simulation tool, components) and emulation (e.g., emulation tool, components). This approach, along with the integration of state of the art simulation and emulation solutions, enables the automated deployment of cyber range training programmes tailored the specific organisation, in realistic environments, while also considering changes in the threat landscape (as encompassed in the assurance model).

Next steps will focus on further refining the Cyber-Range sub-model, testing its applicability in more complex and advanced scenarios, clearly defining the goal and scoring functions supported by the model for trainee performance evaluation. Special focus will be given on improving the simulation and emulation sub-models and integrating additional simulation and emulation tools [26] design. Moreover, the aim is to develop and demonstrate a proof of concept converting a Cyber-Range sub-model to a playable training scenario. Finally, the applicability of the proposed approach will be investigated in different domains, covering smart home, health-care, smart shipping environments.

References

1. A guide to the Internet of Things (2015). https://www-ssl.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html
2. Rantos, K., Fysarakis, K., Manifavas, C.: How effective is your security awareness program? An evaluation methodology. Inf. Secur. J.: Glob. Perspect. **21**(6), 328–345 (2012)
3. Lack of Security Awareness Training Leaves Healthcare Organizations Exposed to Cyberattacks. https://www.hipaajournal.com/lack-of-security-awareness-training-healthcare-cyberattacks/
4. ENISA Smart Grid Security. https://www.enisa.europa.eu/topics/critical-information-infrastructures-and-services/smart-grids/smart-grids-and-smart-metering/ENISA_Annex%20II%20-%20Security%20Aspects%20of%20Smart%20Grid.pdf
5. National Vulnerability Database (NVD). NIST. https://www.nist.gov/programs-projects/national-vulnerability-database-nvd
6. Lagazio, M., Barnard-Wills, D., Rodrigues, R., Wright, D.: Certification Schemes for Cloud Computing. EU Commission Report, Digital Agenta for Europe (2014)
7. CUMULUS Project. Certification infrastructure for multi-layer cloud services project. D2.2 Certification models (2012). http://cordis.europa.eu/docs/projects/cnect/0/318580/080/deliverables/001-D22Certificationmodelsv1.pdf
8. Cloud Security Alliance, CSA Security, Trust and Assurance Registry (STAR). https://cloudsecurityalliance.org/star/
9. EuroCloud Start Audit. https://resilience.enisa.europa.eu/cloud-computing-certification/list-of-cloud-certification-schemes/eurocloud-star-audit
10. NS-3. https://www.nsnam.org/overview/what-is-ns-3/
11. GNS3. https://www.gns3.com/
12. Netkit. http://wiki.netkit.org/
13. OMNet++ Discrete Event Simulator. http://www.omnetpp.org
14. OpenStack. https://www.openstack.org/
15. Docker. https://www.docker.com/
16. OWASP Attack Categories. OWASP. https://www.owasp.org/index.php/Category:Attack
17. ENISA. https://www.enisa.europa.eu/
18. CIPSEC-EU Project. http://www.cipsec.eu/
19. Kaspersky Interactive Protection Simulation (KIPS). https://www.kaspersky.com/enterprise-security/security-awareness
20. MediaPro's Adaptive Awareness Portal. http://www.mediapro.com/adaptive-awareness-framework/adaptive-awareness-portal
21. Sophos Phish Threat. https://www.sophos.com/en-us/products/phish-threat.aspx
22. Inspired eLearning's Security Awareness Training. https://inspiredelearning.com/security-awareness/
23. Amorim, J.A., et al.: Gamified Training for Cyber Defence: Methods and Automated Tools for Situation and Threat Assessment (2013)
24. Boopathi, K., et al.: Learning Cyber Security Through Gamification (2015)
25. PwC's Game of Threats. https://www.pwc.co.uk/issues/cyber-security-data-privacy/services/game-of-threats.html
26. Jasima Discrete Event Simulator. https://www.simplan.de/en/software-2/jasima/