Capability Assessment Methodology and Comparative Analysis of Cybersecurity Training Platforms

Razvan Beuran[a], Jan Vykopal[b], Daniela Belajová[b], Pavel Čeleda[b], Yasuo Tan[a], Yoichi Shinoda[a]

[a] *Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan*
[b] *Masaryk University, Brno, Czech Republic*

Abstract

Cybersecurity training is a key endeavour for ensuring that the IT workforce possess the knowledge and practical skills required to counter the ever-increasing cybersecurity threats that our society is faced with. While some related systems, such as Capture The Flag platforms, have been available for almost one decade, platforms that support full-fledged cybersecurity training exercises have only been released as open source in recent years. Given the complexity of such cybersecurity training platforms, the question that arises is how to meaningfully evaluate and compare their capabilities in order to identify the most suitable solution for a given type of organization and/or training activity.

In this paper, we introduce a capability assessment methodology for cybersecurity training platforms that focuses on the three key aspects of training: content representation, environment management, and training facilitation. The assessment tool that we developed is used to evaluate two open-source cybersecurity training platforms, CyTrONE and KYPO. We then conduct a comparative analysis of these two platforms based on our first-hand developer experience with them, and discuss the lessons learned from implementing, deploying and using these platforms. The assessment tool and the detailed technical comparative analysis that we conducted are intended as instruments and references for anyone who plans to deploy or develop cybersecurity training platforms.

Keywords: capability assessment, comparative analysis, cybersecurity training platforms, cyber range, cybersecurity training exercises

1. Introduction

Cybersecurity training activities are essential for providing the IT workforce with the required knowledge and practical skills for handling the cybersecurity threats that increasingly threaten all organizations [1]. Various types of systems related to cybersecurity training, such as Capture The Flag (CTF) platforms, have been available for almost one decade, but they only include simple tasks and do not provide virtual network environments for trainees [2]. Neverthe-

less, several platforms that support realistic full-fledged cybersecurity training exercises have been developed and released as open source in recent years [3, 4].

We consider that any cybersecurity training platform must cover the three key aspects of training: (i) content representation, (ii) environment management, and (iii) training facilitation. The training content is the set of explicit or implicit tasks that the participants must solve, together with the description of the network environment the participants must interact with in order to solve those tasks. For the training to take place, that network environment—which is composed of virtual or physical hosts—also needs to be created and managed. The term *cyber range* is often used for this environment, but in this paper we alternatively use the term *sandbox* for clarity purposes. In addition, various training facilitation features can be used during the training, such as assigning tasks to participants via the training platform, or following their progress.

Research Goal. Given their heterogeneous functionality, cybersecurity training platforms are inherently highly-complex systems, therefore assessing and comparing them is difficult. The question that this paper focuses on as means to address this issue is: *How to evaluate and compare cybersecurity training platforms capabilities meaningfully and effectively in order to determine the most suitable platform for a given type of organization and/or training activity?*

Being able to answer this question is first of all of interest for any organization that is planning training platform deployments, as it would make it possible to evaluate objectively existing alternatives in order to find the solution that best matches their needs. Secondly, such an assessment methodology would enable academic institutions and commercial companies that are envisaging training platform development to determine the key features they should implement in order to meet various levels of functionality or usability requirements. Moreover, for ongoing development it makes it possible to determine the weak aspects of a platform, so that future implementation efforts can be channeled toward the most cost-effective or necessary improvements. In conclusion, the target audience for our paper are all the cybersecurity training platform stakeholders in charge of deploying, administering and operating the platforms, as well as those involved in their design and development.

Paper Contributions. Using our first-hand experience with training platform development, in particular CyTrONE [3, 5] and KYPO [4, 6], we have created a capability assessment tool—a set of assessment criteria that make it possible to comprehensively evaluate a given training platform development along the three mentioned dimensions: *content representation*, *environment management*, and *training facilitation*. To illustrate the use of this assessment tool, we have conducted an evaluation of CyTrONE and KYPO, emphasizing the advantages presented by this assessment methodology when comparing training platforms. This assessment was made possible by the open-source nature of the two platforms, meaning that their implementation details are publicly available, which is not the case for proprietary systems.

In addition, we have conducted a comparative analysis of CyTrONE and KYPO, focusing on specific elements of interest regarding the three aspects of training, comparing in detail the approaches employed by the analyzed platforms. Furthermore, we discuss the lessons learned from implementing, deploying and using these platforms for actual cybersecurity training activities. Such a detailed technical comparative analysis was made possible by our deep involvement in all the design and implementation decisions concerning CyTrONE and KYPO. The comparative analysis, along with the discussion of the lessons learned by developing the platforms, is intended as a reference for any organization that will attempt to deploy or develop cybersecurity training platforms.

In summary, the main contributions of this paper are as follows:

- Propose a general methodology for cybersecurity training platform evaluation, including a comprehensive capability assessment tool that is provided as the supplementary material of this paper.
- Use the assessment methodology to evaluate CyTrONE and KYPO, the two open-source training platforms that we have independently designed and implemented, and also conducted their comparative analysis.
- Collect and systematize a set of lessons learned from our experience with the training platforms to further support any organization that plans to develop or deploy cybersecurity training platforms.

Paper Structure. The remainder of the paper is structured as follows. Section 2 summarizes related work on cyber range assessment, as well as alternative cybersecurity training tools. Section 3 introduces the cybersecurity training platform assessment methodology that we have developed. Then, in Section 4, we give an overview of the cybersecurity training platforms that are the focus of our analysis, CyTrONE and KYPO. This is followed by Section 5, in which we evaluate the two platforms using the assessment tool we developed, present the results of the comparative analysis we conducted, then summarize the lessons learned from implementing and using the assessed training platforms. The paper ends with conclusions, acknowledgments and references. A supplementary material, which contains the full version of the capability assessment tool, is also provided.

2. Related work

While to the best of our knowledge no analysis on capability assessment has been conducted specifically for cybersecurity training platforms, several comprehensive reviews that focus mainly on the training environment aspect, the cyber range, have been published, as discussed below. We also introduce several alternative training tools that are, however, not within the scope of our paper. Finally, we examine two tools intended for the comparison of numerous capabilities or features that are similar in function to the capability assessment tool presented in this paper.

2.1. Cyber Range Analysis

Yamin et al. [7] reviewed one hundred academic papers on cyber ranges and security testbeds that were published between 2002 and 2018. Their survey studied features and use cases for such systems, and synthesized a unified functional architecture, mainly in regard with training environment-related aspects. Although elements specific to training platforms, such as training content, are not considered explicitly, their proposed architecture is definitely valuable, and we have extended and used it as a basis for describing cybersecurity training platforms, as it will be discussed in Section 3.1.

The development and advancements of cyber ranges have motivated other recently published reviews on this topic. Chouliaras et al. [8] conducted a systematic survey of ten cyber ranges developed by educational institutions in Europe and the USA in the last decade. The desk research was followed by structured interviews with technical directors or managers of these cyber ranges. The paper presents their answers to 12 questions related to system components used to design, create, implement, and operate a cyber range platform. Next, Ukwandu et al. [9] surveyed 44 cyber ranges and testbeds described in academic papers published in 2015–2020. This paper also discusses technologies, scenarios, and applications of the cyber ranges. In particular, it distinguishes four training methods: *gamification*, *mock attack training*, *role-based training*, and *exercise*.

Aschmann [10] proposed a capability maturity model for cyber ranges as means to assess their characteristics. He identified 19 core capability elements for a cyber range, including range and user management, support of various run-time environments, generation of scenario, environment and traffic in the environment, monitoring, security of the range itself, and facility. Each element is assessed according to five capability levels (from *initial* to *ultimate*), as well as five maturity levels (from *initial* to *highly advanced*). The proposed model is complex, and covers both main cyber range use cases, which are testing and education, but lacks sufficient detail for the assessment of platforms developed mainly for education and training purposes. For instance, a learning management system is considered as a core cyber range element, but the model does not define specific features or criteria that can be used to evaluate it from the point of view of delivering training.

2.2. Alternative Training Tools

Our paper focuses on open-source cybersecurity training platforms, because all the implementation details can be determined for them, making a comprehensive assessment possible. However, various commercial cybersecurity training solutions/products also exist, such as the SANS Cyber Ranges (e.g., SANS Netwars Continuous) [11]. Even though such solutions can be in general assessed using the methodology we propose, the lack of publicly available details may prevent a thorough assessment of some of the features.

Moreover, while the target of our analysis are cybersecurity training platforms that make use of cyber ranges, other tools exist for improving fundamental cybersecurity skills. Many of these tools are available to end users via the

Software as a Service (SaaS) model, and examples include Hack The Box [12], TryHackMe [13], and Project Ares [14]. The first two mentioned platforms are used by hundreds of thousands users worldwide, as they provide some basic training content and features for free, although other content and features are offered as a paid service. On the other hand, Project Ares is only available to paying customers. Since no detailed information about the internal architecture and features of these platforms is publicly available, we consider them to be out of scope for this paper.

There are also open-source platforms for facilitating CTF type of games and competitions, such as CTFd, Mellivora, PicoCTF, or FacebookCTF [2]. They make possible assigning tasks via attached files, assessing participant answers, and displaying scoreboards. However, the majority of these platforms do not provide any features for creating the network environment required for solving the presented tasks, i.e., they cannot host full-fledged training. For this reason, even if their open-source nature gives access to implementation details, they are also considered outside the scope of this paper.

2.3. Tools for Capability Assessment

A comparison of multiple features or capabilities is generally delivered to the reader using visual tools. Here we mention two specialized tools for such comparisons, each from a distinct field.

The SIM3 Online Tool [15] is a self-assessment tool for SIM3, the Security Incident Management Maturity Model developed by the Open CSIRT Foundation. The model defines over 40 maturity parameters of computer security incident response teams (CSIRTs) that are divided into four categories. The tool allows to select the maturity level of each parameter, and visualizes the maturity level of a CSIRT as a spider chart.

The CC2020 Visualization Tool [16, 17] compares degree programs according to 34 topics areas defined in the ACM/IEEE-CS Computing Curricula 2020. Users can assign a minimum and maximum value to each topic area required in their degree program, and then display line charts with values of the currently-approved curricula and other degree programs.

3. Capability Assessment Methodology

In this section, we present the capability assessment tool that we have developed for the evaluation of cybersecurity training platforms. This assessment tool is the core of our capability assessment methodology, and it consists in a series of criteria that are used to determine what are the capabilities of a given cybersecurity training platform, and how these capabilities align with those of an "ideal" training platform. The assessment is conducted from the following three perspectives: (i) *training content representation*; (ii) *network environment management*; (iii) *training activity facilitation*.

Below we discuss first a general architecture for cybersecurity training platforms, and how it maps onto the three aforementioned perspectives. Then we

present each of the assessment criteria that we identified (for a total of 58 criteria) grouped according to those three perspectives. For each criterion we provide several choices ordered by platform capability level, from the least to the most capable. The capability level is indicated for each possible choice, although in some cases the choices can be at the same level of capability. Level 1 represents the minimal requirement for a certain type of functionality; however, Level 0 is to be assigned if that particular functionality is missing completely. Level 3 is the maximum capability level in our assessment, although some criteria provide less than three choices.

3.1. Generic Architecture of Cybersecurity Training Platforms

In order to make training possible at a satisfactory level, we consider that any cybersecurity training platform must cover the following three key aspects:

- *Training Content Representation*: Provides a way to represent the explicit or implicit tasks that the participants must solve, together with the description of the network environment the participants must interact with in order to solve those tasks.
- *Network Environment Management*: Provides a mechanism to create the network environment used in the training, orchestrate actions for modifying it during the training (if needed), and destroy the environment once the training ends to free resources.
- *Training Activity Facilitation*: Provides features that improve the usability of the training platform, such as assigning tasks to participants via the training platform, making it possible for instructors to follow participant progress during the training, and recording participant results.

The three aspects above provide a high-level capability perspective on the platforms, and for their implementation the coordination of several system components is necessary. In Figure 1, we show a generic architecture for cybersecurity training platforms in which we emphasize those components that are key for making possible full-fledged training activities.

For the names of the components, we rely on the conventions introduced in the cyber range and security testbed functional architecture presented in [7], while adapting their definitions to match the training platform scope of our analysis. Note that we have excluded the *Testing Module*, whose functionality is related to system security assessment via a testbed, not training. As a new component we have added *Training Content*, which is not a functional but a logical component that plays a critical role when evaluating a platform. The components that we retained and their updated definitions are as follows:

- *Portal*: An interface between the training instructors and participants, and the training platform
- *Management*: A module that manages the roles of the participants and assigns training environment resources to them

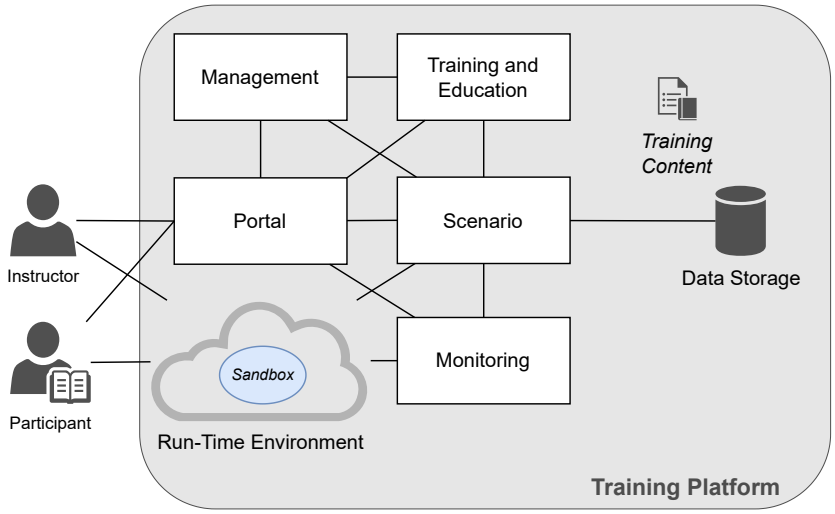


Figure 1: Generic architecture for a full-fledged cybersecurity training platform.

- *Training and Education*: A module that provides a tutoring system for the training, including assessment and feedback functions
- *Scenario*: A module that provides mechanisms for creating, editing, deploying, executing, controlling and destroying training scenarios/activities
- *Monitoring*: A module that provides the capability of monitoring training activities
- *Training Content*: A representation of the training scenarios that are to be used for training activities
- *Data Storage*: A module that stores all the artifacts needed to execute the training scenarios
- *Run-Time Environment*: An infrastructure layer (physical, virtual, hybrid or cloud platforms) on which the training content is deployed via sandboxes

To map the above functional components into the three assessment aspects that we defined, the following workflows should be considered (cf. Figure 1):

1. Instructors use the *Portal* and the *Management* module to access the training content and initiate the training activity via the *Scenario* module.

2. The *Scenario* module retrieves the training content representation from *Data Storage*, then registers the included tasks into the *Training and Education* module, and creates the associated training environment on the *Run-Time Environment*.
3. Participants use the *Portal* and the *Training and Education* module to access the training tasks and the training environment.
4. Instructors use the *Monitoring* module to monitor the progress of the training and the *Training and Education* module for assessment and feedback, then use the *Scenario* module to terminate the training.

Based on these workflows, several dependencies between capability assessment perspectives and the functional components of a cybersecurity training platform can be ascertained, as summarized in Table 1.

Table 1: Dependencies between capability assessment perspectives and the functional components of a cybersecurity training platform

Capability Perspectives	**Dependent Platform Components**
Content Representation	*Training Content*, *Scenario*, *Data Storage*, *Training and Education*
Environment Management	*Management*, *Scenario*, *Data Storage*, *Runtime Environment*
Training Facilitation	*Portal*, *Management*, *Training and Education*, *Monitoring*

3.2. Training Content Representation

Training content representation is a key aspect of a cybersecurity training platform, since it determines how an instructor can create, edit, and maintain existing content, and also how a training platform instantiates the exercise for participants and monitors its progress. Therefore this is the first perspective of our capability assessment methodology.

Note that in the following we will use the term *host* to refer generically to the nodes that are part of the network environment. We also use the term *hypervisor* for the server on which hosts are typically run as virtual machines. Since most platforms use virtualization technologies to improve performance and scalability, we focus mainly on such solutions, but physical platforms can be considered similarly without any loss in generality. Moreover, from here on we will mainly use the term *sandbox* as a shorter way of denoting the network environment used in training.

Training content representation refers to two interdependent aspects: (i) the *sandbox definition*, which specifies the composition of the sandbox in terms of hosts and network topology; (ii) the *training definition*, which describes the tasks assigned to participants, expected correct answers or milestones, services and data to be attacked, defended, or analyzed, and so on.

3.2.1. Sandbox Definition

The assessment criteria related to the sandbox definition (SD) capabilities are summarized in Table 2. Note that the criteria are grouped logically into:

- *Functional Criteria* (SD-1 through SD-7): Criteria related to the functionality provided by the sandbox definitions, such as sandbox customization, supported types of hosts and operating systems, security features, etc.
- *Usability Criteria* (SD-8 through SD-13): Criteria related to the usability of the sandbox definitions, such as how to create the definitions, how they are represented and validated, availability of documentation, etc.

Table 2: Sandbox definition capability assessment criteria

No.	**Assessment Criteria: Functional**
SD-1	Overall customization capabilities for sandboxes
SD-2	Supported categories of hosts in sandboxes
SD-3	Constraints on the CPU architectures and OSs of hosts in sandboxes
SD-4	Mechanisms for customizing hosts in sandboxes
SD-5	Support for security-related sandbox configuration features
SD-6	Support for physical devices (IoT, SCADA/ICS) in sandboxes
SD-7	Constraints on the network topologies in sandboxes
No.	**Assessment Criteria: Usability**
SD-8	Mechanisms for creating/editing sandbox definitions
SD-9	Security measures for isolating sandboxes
SD-10	Representation format for the sandbox definition
SD-11	Validation mechanism for the sandbox definition
SD-12	Availability of documentation for the sandbox definition format
SD-13	Availability of sample sandbox definitions and building blocks

The first page of the capability assessment tool printout, including the description of criteria SD-1 through SD-5, is shown in Figure 2; see the supplementary material for the full version of the tool.

3.2.2. Training Definition

The assessment criteria related to the training definition (TD) capabilities are summarized in Table 3. Note that the criteria are grouped logically into:

- *Functional Criteria* (TD-1 through TD-6): Criteria related to the functionality provided by the training definitions, such as supported training types, ways of structuring the training, supported types of questions and answers, etc.
- *Usability Criteria* (TD-7 through TD-12): Criteria related to the usability of the training definitions, such as how to create the definitions, how they are represented and validated, availability of documentation, etc.

Capability Assessment Tool v1.0

1. Sandbox Definition Capability Assessment Criteria

1.1. Functional Criteria

SD-1: Overall customization capabilities for sandboxes

- ☐ Level 1 – Instructors can only select from predefined sandbox definitions made available by platform developers.
- ☐ Level 2 – Instructors can define custom sandboxes, but only from building blocks (such as hosts, types of networks) provided by platform developers.
- ☐ Level 3 – Instructors can define custom sandboxes from building blocks provided either by platform developers or by themselves, although constraints on complexity and size may apply.

SD-2: Supported categories of hosts in sandboxes

- ☐ Level 1 – Full-fledged virtual or bare-metal hosts only.
- ☐ Level 1 – Container-based hosts only.
- ☐ Level 2 – Both full-fledged virtual or bare-metal hosts, and container-based hosts.

SD-3: Constraints on the CPU architectures and OSs of hosts in sandboxes

- ☐ Level 1 – Only hosts of the same CPU architecture and same family of OSs with the hypervisor, such as only Linux x86 hosts on a Linux x86 hypervisor, or only Windows x86 hosts on a Windows x86 hypervisor.
- ☐ Level 2 – Only hosts of the same CPU architecture, but more than one type of OS, such as Linux x86 and Windows x86 hosts on an x86 hypervisor.
- ☐ Level 3 – Hosts of more than one type of CPU architecture and more than one type of OS, such as Linux hosts with x86 CPU and Android hosts with ARM CPU.

SD-4: Mechanisms for customizing hosts in sandboxes[1]

- ☐ Level 1 – Instructors must provide final binary images of custom hosts in a specific format used by the underlying infrastructure (such as VMware, VirtualBox, KVM).
- ☐ Level 2 – Instructors do not need to build final binary images of custom hosts in a specific format, but must learn the *proprietary system* used by the platform for host provisioning.
- ☐ Level 2 – Instructors do not need to build final binary images of custom hosts in a specific format, but must be proficient in a particular *third-party configuration management system* (such as Ansible, Puppet, Chef) used by the platform for host provisioning.
- ☐ Level 3 – Instructors do not need to build final binary images of custom hosts or be proficient in a platform-specific configuration management system; the platform provides a library of fine-grained building blocks (such as vulnerable services, application stacks) that can be combined together as needed.

SD-5: Support for security-related sandbox configuration features

- ☐ Level 1 – Specific security-related *configurations*, such as for firewalls, can be performed via the sandbox definition.
- ☐ Level 1 – Security-related training content can be *created* via sandbox definition features, such as network attack emulation, traffic capture, malware emulation.
- ☐ Level 2 – Both security-related configurations and training content can be created via the sandbox definition.

Figure 2: Sample of sandbox definition capability assessment criteria.

Table 3: Training definition capability assessment criteria

No.	Assessment Criteria: Functional
TD-1	Types of training supported in training definitions
TD-2	Structuring of individual tasks/milestones
TD-3	Types of questions supported in training definitions
TD-4	Types of answers supported in training definitions
TD-5	Support for specifying a trainee's role in a training
TD-6	Support for rich formatting in training definitions
No.	**Assessment Criteria: Usability**
TD-7	Mechanisms for creating/editing training definitions
TD-8	Support for sharing/hiding the training content between instructors within the platform
TD-9	Representation format for the training definition
TD-10	Validation mechanism for the training definition
TD-11	Availability of documentation for the training definition format
TD-12	Availability of sample training definitions

The description of criterion TD-1 is shown in Figure 3 as an example; see the supplementary material for the full version of the capability assessment tool.

TD-1: Types of training supported in training definitions

- ☐ Level 1 – Task-based training: questions are presented ordered or unordered via a GUI, and trainees' answers are validated, but the GUI does not interact with the sandbox; similar to a jeopardy CTF game.
- ☐ Level 1 – Milestone-based training: the platform checks the status of the sandbox (network services or data stored on hosts) to automatically determine whether milestones defined by instructors are met or not; similar to attack-defense or King of the Hill CTFs, or cyber defense/offense exercises.
- ☐ Level 2 – Both task-based and milestone-based modes are supported.

Figure 3: Sample of a training definition capability assessment criterion.

3.3. Network Environment Management

Network environment management is the core feature of a cybersecurity training platform. Management starts with environment creation, by which the sandbox definition is "transformed" into instances of an actual network environment for a particular training session intended for a certain number of trainees. Other aspects related to the network environment management include deleting some or all instances of the environment, as well as any other form of training orchestration, such as performing actions related to an ongoing training (e.g., controlled attacks).

The assessment criteria related to network environment management (EM) capabilities are summarized in Table 4. Note that the criteria are grouped logically into:

- *Functional Criteria* (EM-1 through EM-5): Criteria related to network environment management functionality, such as environment creation and control, support for executing action and network attacks, etc.
- *Performance Criteria* (EM-6 through EM-8): Criteria related to network environment management performance, such as reliability and efficiency of environment creation, and resource consumption.
- *Usability Criteria* (EM-9 through EM-13): Criteria related to network environment management usability, such as the type of user interface, degree of automation, and monitoring features.

Table 4: Network environment management capability assessment criteria

No.	**Assessment Criteria: Functional**
EM-1	Flexibility of network environment creation for parallel training activities
EM-2	Access control features for network environment management
EM-3	Support for automated action execution in a network environment *after* its creation
EM-4	Support for network attack execution functionality
EM-5	Support for background traffic generation functionality
No.	**Assessment Criteria: Performance**
EM-6	Reliability of network environment creation
EM-7	Time efficiency of network environment creation
EM-8	Resource consumption for a given network environment
No.	**Assessment Criteria: Usability**
EM-9	Type of user interface for network environment creation/deletion
EM-10	Degree of automation for network environment creation
EM-11	Ability to monitor the network environment creation process
EM-12	Degree of automation for network environment deletion
EM-13	Ability to monitor the network environment deletion process

The description of criterion EM-1 is shown in Figure 4 as an example; see the supplementary material for the full version of the capability assessment tool.

EM-1: Flexibility of network environment creation for parallel training activities

☐ Level 1 – Only multiple instances of the same network environment can be created for training activities conducted in parallel.

☐ Level 2 – Different network environments can be created even for training activities conducted in parallel.

Figure 4: Sample of an environment management capability assessment criterion.

3.4. Training Activity Facilitation

Facilitating training activities makes it possible to deliver quality training. This is true both from the perspective of participants and instructors, and can include features such as the use of a learning management system during the training and scaffolding mechanisms. This is because the quality of interface between the training participants and the training platform can lead to an efficient and engaging training, but can also make it worse.

Next, we introduce the assessment criteria related to training activity facilitation (AF) capabilities, as summarized in Table 5. The criteria are grouped into several logical categories:

- *Pre-training Setup Features* (AF-1 through AF-6): Criteria related to the pre-training setup of the platform, such as deployment automation, training content import, training session visibility and access time control, etc.
- *Training Execution Features* (AF-7 through AF-18): Criteria related to the actual training execution, such as use of a learning management system (LMS) or an intelligent tutoring system (ITS), various educational features (e.g., scaffolding, cheating prevention and detection), facilitation for instructors to access sandboxes and gather learning analytics, etc.
- *Post-training Assessment Features* (AF-19 and AF-20): Criteria related to post-training assessment, such as export of training data, and result analysis across different training sessions.

The description of criterion AF-1 is shown in Figure 5 as an example; see the supplementary material for the full version of the capability assessment tool.

AF-1: Degree of automation for the deployment of the platform itself

☐ Level 1 – There are ad-hoc scripts automating the deployment on a host with a specific operating system/hardware.

☐ Level 2 – There are configuration files following the declarative Infrastructure as Code (IaC) approach[4] for platform deployment.

Figure 5: Sample of an activity facilitation capability assessment criterion.

3.5. Practical Considerations

The thoroughness of the capability assessment methodology that we developed has lead to a certain intrinsic complexity of the assessment tool. In this section, we provide some practical considerations regarding its potential users, and propose some assessment profiles that simplify its application in practice.

3.5.1. Potential Users

The capability assessment tool that we developed can be employed by different kinds of cybersecurity training platform stakeholders, both those that are in charge of deploying, administering and operating the platforms, as well as those involved in their design and development.

Table 5: Training activity facilitation capability assessment criteria

No.	Assessment Criteria: Pre-training Setup
AF-1	Degree of automation for the deployment of the platform itself
AF-2	Support for importing training content into the platform
AF-3	Trainee access management features of the training platform
AF-4	Visibility control features for training sessions
AF-5	Access time control features for training sessions
AF-6	Degree of instructor assistance needed by trainees for participating in training sessions
No.	**Assessment Criteria: Training Execution**
AF-7	Use of an LMS or ITS during training
AF-8	Features of the LMS or ITS used during training
AF-9	Types of scaffolding mechanisms available during training
AF-10	Cheating prevention or detection features
AF-11	Manner of accessing hosts in the created network environment
AF-12	Type of user experience when accessing hosts in the created network environment
AF-13	Support for instructors accessing in-use hosts in sandboxes
AF-14	Support for assigning one sandbox to multiple participants in the same team
AF-15	Visualization of the network environment topology
AF-16	Support for trainees resuming a training session after exiting
AF-17	Situational awareness and learning analytics features for instructors
AF-18	Learning analytics features for trainees
No.	**Assessment Criteria: Post-training Assessment**
AF-19	Support for exporting data collected during the training session or participants' score for further processing outside the platform
AF-20	Support for analyzing the progress or results of participants across different training sessions

Let's consider first the perspective of cybersecurity training platform users, and how they can benefit from using the capability assessment tool. Thus, an organization that wants to deploy a cybersecurity training platform can use it to evaluate several existing solutions. Based on the detailed results provided, operators are able to determine which of the assessed platforms is most suitable, depending on their target use and requirements regarding the training activities, such as teaching students, training professionals, etc.

As for cybersecurity training platform developers, let's imagine that a team is making plans to develop a new training platform. The developers can then use the capability assessment tool to determine the set of features that the new training platform should target, thus guiding the processes of defining requirements and implementing the platform. On the other hand, for an existing

platform its developers can use the capability assessment tool to identify the areas in which the platform functionality is lacking, hence it can guide the process of improving such platforms.

While these are only some examples of potential ways of using the capability assessment tool, Section 5 discusses how the results of the assessment can be of practical use.

3.5.2. Assessment Profiles

Not all users may require all the features of the capability assessment tool that we developed. In order to simplify the assessment process, we propose the use of *assessment profiles* as a way to restrict the assessment to only those aspects that are of interest for a particular scenario.

Simple Training Activity. The first profile we discuss is that of a simple training activity, in which the focus is on training related to basic security skills, and education aspects are of little concern. For such a case, the assessment can be done only from the perspective of the criteria mentioned in Table 6, thus reducing the total number of criteria to be assessed from 58 to 20.

Table 6: Assessment criteria for the "Simple Training Activity" profile

Assessment Perspective	**Assessment Criteria**
Content Definition	SD-1, SD-8 through SD-10, SD-12, SD-13, TD-1, TD-2, TD-6, TD-7, TD-9, TD-11, TD-12
Environment Management	EM-1, EM-6, EM-10, EM-12
Activity Facilitation	AF-1, AF-3, AF-12

Unsupervised Training Activity. The second profile focuses on training sessions that can be accessed by students without assistance from instructors at anytime during a specified time period. For such a case, the assessment can be done from the perspective of the criteria mentioned in Table 7, thus reducing the total number of criteria to be assessed from 58 to 37.

Table 7: Assessment criteria for the "Unsupervised Training Activity" profile

Assessment Perspective	**Assessment Criteria**
Content Definition	SD-1, SD-2, SD-7 through SD-13, TD-1, TD-2, TD-5, TD-7 through TD-10
Environment Management	EM-1, EM-6, EM-9 through EM-13
Activity Facilitation	AF-1, AF-3, AF-5 through AF-9, AF-11, AF-13, AF-15, AF-16, AF-18, AF-19, AF-21

Platform as a Service. The third exemplary profile is intended for assessing the capabilities of a platform that provides cybersecurity training as a service to third parties who run their training sessions remotely. Table 8 lists the individual criteria for this profile, which are reduced from 58 to 37.

Table 8: Assessment criteria for the "Platform as a Service" profile

Assessment Perspective	**Assessment Criteria**
Content Definition	SD-1, SD-2, SD-4, SD-7 through SD-13, TD-1, TD-2, TD-5, TD-7 through TD-12
Environment Management	EM-1, EM-2, EM-6, EM-8 through EM-13
Activity Facilitation	AF-1 through AF-8, AF-18

4. Training Platform Overview

In this section, we introduce the cybersecurity training platforms that we will later assess and compare, CyTrONE and KYPO:

- *CyTrONE*: Integrated cybersecurity training framework developed since 2015 by the Japan Advanced Institute of Science and Technology (JAIST), in Ishikawa, Japan [5]. CyTrONE has been used for various education and training activities, both in Japan and in other countries, as well as for training content development, including in collaboration with Japanese commercial companies.

- *KYPO*: Cyber range platform that has been iteratively improved and used in practice since 2013, with its current third generation being developed by Masaryk University, Czech Republic since 2018 [6]. KYPO has been used for teaching cybersecurity in several organizations in Czech Republic and Europe for various target groups of learners (high school students, undergraduates, graduates, and professional learners).

In what follows, we describe the main characteristics of each platform by following the component structure discussed in Section 3.1.

4.1. CyTrONE

The CyTrONE framework [5] has been developed by the Cyber Range Organization and Design (CROND) NEC-endowed chair at JAIST as an integrated cybersecurity training platform, and various CyTrONE modules have been released as open source on the CROND GitHub page [3]. During the years, CyTrONE has been used for several types of training activities, both in Japan (e.g., for courses taught at JAIST and Keio University) and in several other universities in various countries around the world (France, Greece, UK, Uruguay). In addition to the training content developed by CROND, external parties, such as the Tokyo Metropolitan College in Japan, have also contributed CyTrONE

training content; we also developed content in collaboration with commercial companies in Japan (e.g., Allied Telesis Academy). Details regarding CyTrONE are presented below:

- *Portal*: CyTrONE utilizes a web interface, named "CyTrONE Door," that is used by instructors to create and destroy training sessions based on content registered in the CyTrONE database, and to check the active sessions. The other interactions of the users, both instructors and participants, with the training are conducted via the Moodle Learning Management System (LMS) that is integrated with the framework, whose functionality is described in *Training and Education*.

- *Management*: The module named CyTrONE includes the top-level management functionality of the framework, and drives the execution of the other modules, such as CyLMS and CyRIS, for lower-level management tasks. These modules are described in detail in *Scenario*. Note that instructors typically do not interact with CyLMS or CyRIS directly, but via the simplified interface provided by CyTrONE, either using a CLI or the Door web interface mentioned above. Participants do not have any management permissions for CyTrONE training activities.

- *Training and Education*: CyTrONE users mainly interact with the training via the web interface provided by Moodle LMS integrated with the framework. Participants can check via Moodle the training activity description and questions (tasks) assigned to them, as well as submit their answers. Instructors use Moodle to confirm the progress of the trainees. CyTrONE-based training uses scenarios composed of a set of tasks that can be either independent from each other, or for which the order is critical for completing the training. Trainees are provided with an account name and password they use to access the Moodle LMS to discover the training content, and also the network environment on which they need to carry out various actions in order to solve the assigned tasks. By default, the correct answers for each task are the same for all participants, but extension features are also available to create environments with custom solutions for each participant, thus preventing cheating.

- *Scenario*: The functionality related to enacting training scenarios is split in CyTrONE amongst several modules. Thus, CyLMS is used to interact with the Moodle LMS, and has functions for registering the *training definition* into Moodle, as well as deleting that content when the activity ends. Another module, named CyRIS, is used to prepare the network environment used during a certain training activity based on the corresponding *sandbox definition*. CyRIS is also used to destroy the training environment once the activity ends. While the network environment created by CyRIS is static, an additional module, named CyPROM, can be used to make dynamic changes to the network environment based on trainee progress, e.g., by conducting cyber attacks.

- *Monitoring*: CyTrONE does not provide any specialized monitoring functionality, relying instead on the capabilities of the Moodle LMS for this purpose. Thus, instructors can use the Moodle GUI to retrieve detailed information about the answers submitted by each trainee. However, if a detailed analysis of the network environment is needed for a particular trainee, then the environment used by that trainee should be accessed, for example to check the command execution history.

- *Training Content*: A training activity is defined by a set of two files. The *training definition* contains all the information that is to be registered via CyLMS in Moodle, such as activity overview, questions and their correct answers, hints, etc. The *sandbox definition* contains all the information that is to be used by CyRIS to create the network environment for that particular training, such as the details about each guest Virtual Machine (VM) and how it is to be configured, and the network topology. These two files are written in the YAML format, so they are easy to modify for updating the content, to compare for determine any differences, and so on. The two definition files are stored together with the set of resources (binary files, flags, etc.) required for that particular training activity. Several sets of CyTrONE training content are available with the source code and also at [18].

- *Data Storage*: All data required by CyTrONE is stored on the main CyTrONE host, and is used as needed to register training content in Moodle and create the training environment. To manage the data and the users specific database files are employed, again using YAML format for easy editing. In addition to the YAML files described in *Training Content*, base VMs are stored as "building blocks" for network environment creation. As for LMS-specific education analytics, they are stored directly by Moodle, and can be consulted as needed.

- *Run-Time Environment*: CyTrONE and its modules run on Ubuntu LTS physical servers, and the training network environment created by CyRIS based on the *sandbox definition* is deployed using Kernel-based Virtual Machine (KVM) virtualization technology. Multiple physical servers can be used simultaneously if large-scale training activities are to be conducted. Alternatively, the Amazon Web Services (AWS) Elastic Computer Cloud (EC2) platform can be used to deploy the network environment. For the Moodle LMS, a KVM virtual machine is typically used as well on the main CyTrONE server, but employing a dedicated Moodle physical server is also possible.

4.2. KYPO Cyber Range Platform

KYPO Cyber Range Platform (KYPO CRP) [6] is a cloud-based platform providing an interactive learning environment to deliver hands-on cybersecurity classes, on-site or remotely. The environment is designed for multiple simultaneous training sessions with cybersecurity assignments and learner assessments.

The platform covers various training use cases, from individual assignments with step-by-step instructions or solved autonomously to long-term team projects or serious games, e.g., capture the flag or defense exercises. Regardless of the training format, trainees interact with a learning environment. Details regarding KYPO are as follows:

- *Portal*: KYPO CRP enables users to interact with the platform through a web interface (KYPO portal). KYPO recognizes four essential roles: *trainee*, *instructor*, *designer*, and *administrator*. Trainees only have limited access to run new training, continue unfinished ones, or access the results of those already finished. Instructors can organize training sessions with defined parameters and manage cloud resources. Designers prepare training scenarios. Finally, administrators include all other roles and set roles of other users.

- *Management*: The KYPO Portal provides tools to manage access rights for groups or individual users and to manage scenario definitions. Management functions are available only to users with higher privileges than trainees (instructors etc.). Before any session, the instructor checks that estimated resources for the training are available in the cloud. The estimation is based on the number of trainees and the sandbox size. Afterward, the instructor allocates a pool of sandboxes (instances of network environments), which are then assigned to trainees.

- *Training and Education*: KYPO CRP directly supports linear and adaptive training sessions [19] for individual participants, and allows using the created network environment for other types of exercises. Trainees enter a training session by entering an access code provided by the instructor. After that, they solve predefined tasks one by one. Linear training contains tasks, which are presented to a trainee in a fixed order, regardless their performance in the previous tasks whereas the adaptive training mode selects the most suitable task. This feature is enabled by a monitoring component. KYPO also provides features for prevention and detection of cheating [20]. Each trainee can be provided with a personalized sandbox with different answers to be found so that the system can detect, for example, whether a trainee submitted some else's flag.

- *Scenario*: Management of training sessions is provided by two key KYPO components. While *Training service* takes care of the training session life cycle, *Sandbox service* is responsible for creating and managing the runtime virtual environment where the training session takes place. Both services rely on human-readable definition files, see *Training Content* further in this section.

- *Monitoring*: To provide monitoring of students' progress, the platform processes events both from network environment and a LMS/ITS component. The events are predefined machine-readable logs in a Syslog protocol

format [21]. Examples of such events include displaying hints or the solution, typing a command at the command line of a host, and starting or finishing the training and its phases. Based on these data, statistics and dashboards are displayed to instructors within the KYPO portal. Such data can be also used for formative assessments of students with individual feedback or summative assessment, i.e., testing and grading based on a particular scoring system.

- *Training Content*: Three different definition files define a particular training. *Training definition* specifies consecutive tasks (or a set of tasks) that trainees have to solve. It is a machine-readable description of the training in a JSON format. The generic structure consists of introductory information and several training phases where students must prove the correct solution by submitting a text answer. The optional hints or worked-out solutions can be provided as well as questionnaires representing pre- or post-tests. Further, network environment is defined by *sandbox definition*, which consists of *provisioning definition* and *topology definition* files. The topology definition specifies the network topology with routers, hosts (base boxes), and network or router mappings identifiers. The provisioning definition specifies the configuration of individual hosts (VMs). The file provides input for the software configuration management system responsible for installing and configuring hosts deployed as base boxes (Ansible). Both files have YAML format following the standard conventions.

- *Data Storage*: For training and sandbox definitions (topology and provisioning), a public or private Gitlab repository is used as primary data storage. As central storage of learning analytics events, the ElasticStack software stack (Elasticsearch, Logstash, and Kibana) [22] is used.

- *Run-Time Environment*: KYPO CRP transforms the topology definition into a Terraform configuration, which is applied to the OpenStack cloud computing platform. Once the instantiation of the training network and its hosts is done, KYPO provisions the base boxes using Ansible, which executes the provisioning definition. KYPO CRP also supports using the local computers of trainees to deploy the network environment defined by the sandbox definition by using Vagrant and VirtualBox installed on trainees' hosts.

5. Capability Assessment and Comparative Analysis

In this section, we start by using the capability assessment methodology introduced in Section 3 to evaluate CyTrONE and KYPO. We then proceed to a more detailed comparative analysis based on our first-hand perspective and developers of these platforms. Finally, we provide an overview of the lessons learned through the development and deployment of the mentioned platforms.

5.1. Assessment Results

Using the capability assessment tool we have evaluated CyTrONE and KYPO based on their publicly available features at the moment of writing (December 2022). Below we present the detailed results, as well as an overall assessment.

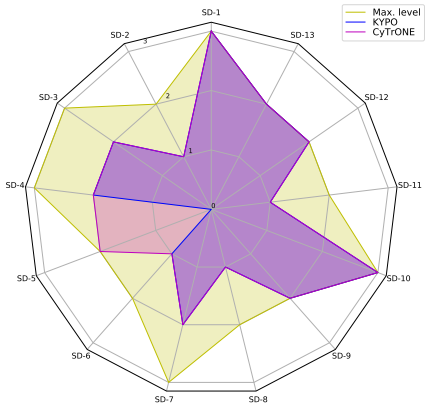
Detailed Results. In Figure 6, we use radar chart representations for each class of assessment criteria—sandbox definition, training definition, environment management and activity facilitation—plotting the results of both CyTrONE and KYPO on the same graph. We also plot the maximum level for each criterion to illustrate how close a given platform is from providing maximum functionality for that criterion. This radar chart representation makes possible a detailed feature-by-feature comparison, as we will detail next.

From Figure 6a, we observe that CyTrONE and KYPO are very close to each other in terms of functionality. The only notable difference is that KYPO has no security-related sandbox configuration features (criterion SD-5), whereas CyTrONE included such features by design. Another way to use the radar chart is to look at areas that correspond to groups of assessment criteria. For example, by looking at the area for criteria SD-1 through SD-7 (functional criteria) in Figure 6a, we can conclude that there are several criteria for which the maximum values are not attained, emphasizing a certain lack of the corresponding functionality and a potential for improving these features in the future. On the other hand, the area for criteria SD-8 through SD-13 (usability criteria) shows that the two platforms have identical profiles and the maximum values are attained in most cases, demonstrating the maturity of the platforms from this perspective, as well as the low potential for future improvement. A similar analysis can be conducted for the other radar chart representations to identify capability areas that are lacking and/or can be improved.

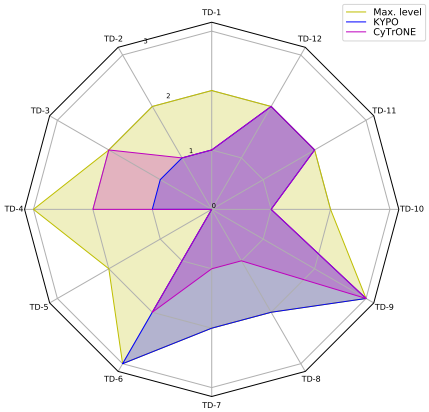
For Figure 6b, we note that while CyTrONE and KYPO are again similar in terms of functionality, CyTrONE has some advantages in terms of the types of questions and answers supported in training definitions (criteria TD-3 and TD-4). On the other hand, KYPO provides better support for the rich formatting of training definitions (criterion TD-6), and includes usability features regarding the editing of training definitions and the sharing/hiding of training content (criteria TD-7 and TD-8).

Figure 6c emphasizes other differences between CyTrONE and KYPO. In particular, the superior support for automated action and attack execution which CyTrONE provides via its module named CyPROM (criteria EM-3 and EM-4). We can also observe that neither CyTrONE nor KYPO provide any built-in support for background traffic generation (criterion EM-5), although this functionality could be achieved via the use of external tools.

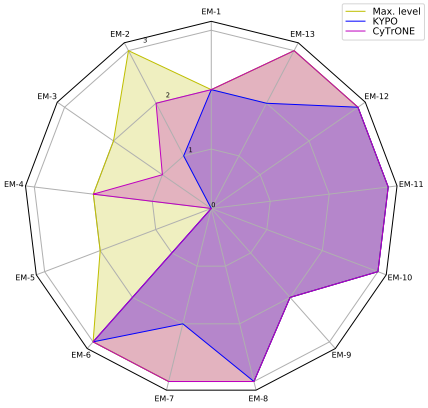
In Figure 6d, the superior functionality of KYPO in terms of training activity facilitation is clear, as it scores higher than CyTrONE in most of the assessed criteria. We also notice that the assessment for KYPO is the maximum possible level for most criteria related to activity facilitation, signifying that KYPO is very close to being an "ideal" platform from this perspective.



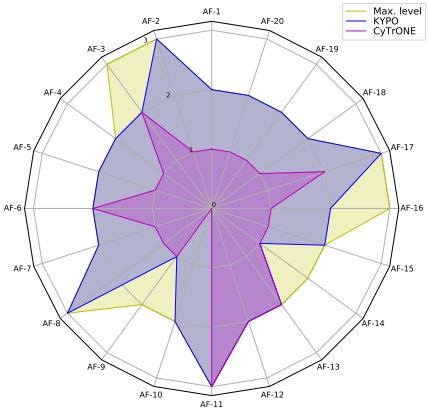
(a) Sandbox definition (SD) capability assessment.



(b) Training definition (TD) capability assessment.



(c) Training environment management (EM) capability assessment.



(d) Training activity facilitation (AF) capability assessment.

Figure 6: Detailed capability assessment results for CyTrONE and KYPO.

Overall Assessment. In order to assess the total capabilities per class of assessment criteria, we have also created the bar plot shown in Figure 7, in which for each class the total capability for that class is displayed as a percentage of the maximum possible score for that particular class. The figure also shows the overall capability for all the four classes combined.

From Figure 7, we observe that CyTrONE has somewhat higher capabilities related to sandbox definition (SD) by about 6%, whereas KYPO has higher capabilities in regard with training definition (TD) by about 4%. As for the network environment management (EM), the functionality of CyTrONE is again

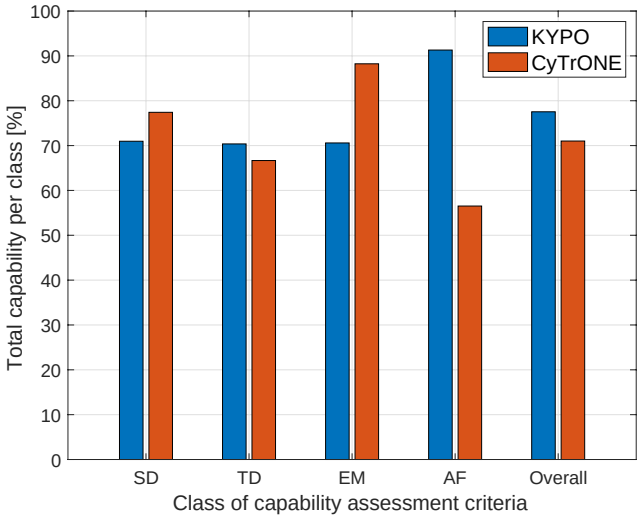


Figure 7: Capability assessment results for CyTrONE and KYPO displayed per class of assessment criteria (Sandbox Definition, SD; Training Definition, TD; Environment Management, EM; Activity Facilitation, AF), and as overall results.

higher, this time by about 18%. A very significant difference is nevertheless observed for training activity facilitation, where the capabilities of KYPO are 35% higher when compared to those of CyTrONE, emphasizing that KYPO is a more mature platform from this point of view.

Figure 7 also shows that both CyTrONE and KYPO have overall good sets of features, with capability approximately larger than 70% for each class (the only exception being the about 57% capability of CyTrONE for activity facilitation). The right-hand side bars also emphasize the overall good capability levels of the two platforms, both of them having an average capability of more than 70%, with KYPO having a higher capability compared to CyTrONE by about 7%.

The quantitative analysis that we have presented in this section demonstrates the manner in which our capability assessment tool can be used to make a detailed analysis of several platforms, enabling stakeholders to make informed and objective decisions about topics such as: which features of a platform should be extended, which platform is more feature rich for a certain class of criteria, and in the end which platform is overall superior.

5.2. Comparative Analysis

Our position as main developers of the assessed cybersecurity training platforms makes it possible to conduct an even deeper comparative analysis. The analysis will be performed along the three perspectives defined in Section 3.1, and we will focus mainly on the differences in how CyTrONE and KYPO were designed and implemented, and the reasons behind those differences.

5.2.1. Training Content Representation

The main differences between CyTrONE and KYPO from a training content representation point of view are as follows:

- *Representation format*: While both CyTrONE and KYPO use YAML format for the *sandbox definition*, the *training definition* is represented using YAML in CyTrONE and JSON in KYPO. The YAML format makes it easy to edit content by hand, whereas the JSON format, which is more verbose and difficult to organize, makes this more difficult, and the file must be generated via an external tool. While the format itself does not affect the functionality of the platforms, the advantages and disadvantages of each representation format must be considered before choosing an approach/platform.

- *Security-specific sandbox configuration*: CyTrONE has several security-specific features that facilitate training content creation without the use of external tools, thus simplifying the task of content creation. These features include built-in configuration capabilities for firewalls, as well as generation of security-related training content, such as network attack emulation, traffic capture, malware emulation.

- *Trainee role assignment*: CyTrONE is built with individual trainees in mind, such as individual students; however, KYPO includes several features that make it possible to organize participants in teams, as well as assigning their roles in a training, such as attacker or defender. This makes it possible to conduct more realistic training activities that mimic real-life situations, such as handling security incidents as a team in a company.

5.2.2. Network Environment Management

Next, we summarize the main differences between CyTrONE and KYPO from a network environment management point of view:

- *Virtualization technique*: CyTrONE relies on KVM for VM deployment, whereas KYPO uses OpenStack technology. The KVM approach has minimum installation overhead, as only the appropriate packages need to be installed on the physical servers used for deployment. On the other hand, OpenStack requires setting up the servers to use this technology, which requires more advanced skills for the administrators, and dedicating those servers to training. From a developer's perspective however, the virtualization capabilities of KVM are very basic, and wrappers need to be built around them when implementing components such as the *Management* or *Scenario* modules. OpenStack provides built-in higher level functionality, hence makes the training platform implementation easier and potentially more powerful. Consequently, the virtualization techniques used by a platform should be considered independently from deployment and implementation perspectives, as a trade-off may be necessary to find the most appropriate solution.

- *Dynamic training environments*: CyTrONE includes several features that make it possible to create dynamic training environments in which the environment is changed on purpose depending on time triggers or the progress of the trainees. These features include support for automated action execution after the environment is created, as well as support for attack execution during the training. Such features were implemented in CyTrONE to improve the training realism by making it possible to use training environments that mimic real-life ones more closely.

5.2.3. Training Activity Facilitation

While CyTrONE includes several basic training activity facilitation features, our assessment demonstrated that KYPO is superior from this point of view in almost all respects, and below we emphasize its main advantages:

- *Pre-training setup*: KYPO has better support for training content and training session management.
- *Training execution*: KYPO includes several advanced features related to scaffolding for learning, cheating prevention/detection, situational awareness and learning analytics.
- *Post-training assessment*: KYPO has better support for exporting training data and scores, as well as analyzing participant progress and results.

5.3. Lessons Learned

In this section, we discuss the lessons learned from the implementation of the cybersecurity training platforms over a period of more than seven years, as well as their deployment and actual cybersecurity training activities we conducted.

5.3.1. Deployment Aspects for Administrators

The first perspective we present is that of system administrators that need to deploy and maintain cybersecurity training platforms. The lessons we learned from this perspective are as follows:

- The operating system for the hosts on which the platform is deployed should be stable, and not have frequent feature updates that may break functionality, thus making platform source code maintenance easier (for example, the once every two years Long-Term-Support (LTS) updates of Ubuntu often required rewriting part of the CyTrONE source code to restore functionality). Stability is also relevant for containers, which may need to be reconfigured if the operating system changes significantly.
- When sandboxes are deployed via an on-site cloud computing infrastructure, such as OpenStack, installing and maintaining that cloud platform introduces an overhead that may not be negligible. Such operational costs can be reduced by using pubic clouds (e.g., AWS EC2), but in this case access delays may occur, and a trade-off may need to be made in terms

of the types of training activities that can be deployed from a platform security perspective.

- Good documentation regarding the installation procedure, or even better, support for easy deployment (e.g., one-click) are essential in lowering the barrier for trialing and eventually adopting a given training platform.

5.3.2. Implementation Aspects for Developers

Next we will take the perspective of developers. Thus, the main lessons learned from our experience with implementing cybersecurity training platforms are as follows:

- Developers should choose programming languages that are easy to master (e.g., Python) and well-supported DevOps tools and software development methods (such as Continuous Integration and Deployment/Delivery, CI/CD) in order to facilitate both the actual implementation and the quality control process.

- Development should follow the Open-Source Software (OSS) paradigm with public releases that encourage the use of the software as well as contributions by third parties.

- Agile development methodology should be used, especially in the early stages of platform implementation or for new functionality, so that modules can be quickly prototyped, then tested in realistic conditions, with the received feedback being used to fine tune the implementation and drive further development.

- Given the sensitive nature of cybersecurity training, the platform itself should be subjected to security, vulnerability and penetration testing to ensure that it is safe to use.

- Cybersecurity training platforms are complex systems, and their implementation requires developers and testers with skills related to many areas, such as cybersecurity, operating systems, education and learning, etc. Therefore, a multi-disciplinary team should be assigned for their design and development.

5.3.3. Training Aspects for Instructors and Participants

Finally, we shall take the perspective of the users of cybersecurity training platforms, both instructors and participants, who do not necessarily have technical knowledge about the platforms and their underlying technologies. The lessons we learned from this perspective are as follows:

- Sample training content and base sandbox components are important for making it possible for instructors to understand the features of a platform, and to create original training content by modifying/updating them. Moreover, training content creation should be facilitated via the use of

GUIs, standard environment configuration tools, but also employ human-readable representation formats, so that both ordinary and power users can be accommodated.

- Training content representation should be flexible enough to enable sharing the content between instructors, both within the same and different institutions, so that new training content can be created by extending/modifying existing content. In this context, the platform support for content import/export becomes critical.

- Both instructors and participants should have access to situational awareness and learning analytics features that make it possible to follow aspects such as the use of training environment resources, training progress and training results, so that the training activity is as effective as possible.

- The platform should provide features allowing its seamless integration into existing workflows of the training process and of the organization providing the training. For instance, the platform should be ready to authenticate its users using existing identity providers or allow importing user accounts from external systems. Also, the platform should provide features for exporting scores or training progress to external learning management systems. Such interoperability saves instructors time, particularly for large classes of participants.

6. Conclusion

In this paper, we proposed a capability assessment methodology for cybersecurity training platforms that focuses on the three key aspects of training: content representation, environment management, and training facilitation. As the core of this methodology we have developed a detailed assessment tool that can be used to quantify the capabilities of cybersecurity training platforms. The assessment tool has a total of 58 criteria pertaining to the three aforementioned perspectives. For each perspective we grouped the criteria according to their scope, such as platform functionality, performance, or usability, thus facilitating their interpretation. In addition, we have defined several assessment profiles that make it possible to focus the assessment on those criteria that matter for a certain deployment target, such as unsupervised training activity.

The capability assessment tool that we developed essentially establishes a cybersecurity training platform benchmark that can be employed in several manners, as follows:

- Platform administrators/operators can use it to evaluate several existing solutions to determine which platform is most suitable to deploy, depending on the specific intended use and requirements they may have.

- Platform designers/developers can use it to determine the target set of features for a new training platform, or identify the areas in which the functionality of an existing platform needs to be improved.

To demonstrate its applicability, we have employed the capability assessment methodology to evaluate two open-source platforms, CyTrONE and KYPO. The detailed capability assessment results made it possible to determine which specific features a platform misses, such as security-related configuration features missing in KYPO (criterion SD-5), or certain usability features regarding creating/editing and sharing/hiding of training content (criteria TD-7 and TD-8). Overall, we were able to establish that KYPO has superior capabilities in our benchmark, with the highest difference being observed for training activity facilitation features.

We have also conducted a comparative analysis of CyTrONE and KYPO based on the results of the capability assessment, as well as our first-hand experience as their developers, comparing the approaches employed for each of them. The main conclusions of our analysis can be summarized as follows:

- Training content representation included technical differences in terms of representation format, but the functionality was overall similar; specific features also exist, such as the security-related configurations of CyTrONE, and the team-related features of KYPO.

- Network environment creation used different virtualization technologies, a fundamental choice that lead to various differences is terms of the sandbox creation mechanisms; in addition, CyTrONE has a mechanism for applying changes to the training environment during the training.

- Training activity facilitation is the area in which most differences were observed, with KYPO providing a richer set of features than CyTrONE that ensures a more user-friendly training experience.

In addition, we have discussed the lessons learned from implementing, deploying and using cybersecurity training platforms over a period of more than seven years. The main takeaways are as follows:

- System (and training) administrators should ensure that the platform is supported on stable operating systems, thus reducing the maintenance effort and costs, and the overhead of maintaining any necessary on-site cloud computing infrastructure should be accounted for; good documentation and easy deployment features are highly desirable.

- Developers should follow sound programming methodologies, using well-supported programming languages and DevOps tools; agile development is recommended as development strategy; moreover, the platform itself should be subjected to security testing.

- Instructors and participants should leverage the situational awareness and learning analytics functionality of the platform (if available) in order to maximize the effectiveness of the training; in addition, instructors should actively extend and modify existing training content as a way to speed up content development.

The current version of the capability assessment tool that we developed is provided to the public as a supplementary material to this paper. We plan to collaborate with other training platform developers to evaluate their platforms, as well as use their feedback to further enhance the proposed capability assessment methodology. Moreover, on the development side, we plan to focus on some of the key points that we have identified as lacking in the current cybersecurity training platform implementations in order to improve them. In particular, for CyTrONE training activity facilitation features are an important needed addition, and for KYPO security-related configuration features would improve the overall capabilities of the platform.

Acknowledgment

The authors would like to thank all the developers of CyTrONE and KYPO for their invaluable contributions. This research was partially supported by the European Regional Development Fund (ERDF) project *CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence* (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] G. Subasu, L. Rosu, I. Badoi, Modeling and simulation architecture for training in cyber defence education, in: 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2017, pp. 1–4. doi:10.1109/ECAI.2017.8166396.

[2] S. Kucek, M. Leitner, An Empirical Survey of Functions and Configurations of Open-Source Capture the Flag (CTF) Environments, Journal of Network and Computer Applications 151 (2020). doi:10.1016/j.jnca.2019.102470.

[3] Cyber Range Organisation and Design (CROND), CROND GitHub page, https://github.com/crond-jaist, [Last accessed on November 24, 2022] (2022).

[4] Masaryk University, KYPO Cyber Range Platform, https://crp.kypo.muni.cz, [Last accessed on November 24, 2022] (2022).

[5] R. Beuran, D. Tang, C. Pham, K. Chinen, Y. Tan, Y. Shinoda, Integrated framework for hands-on cybersecurity training: CyTrONE, Computers & Security 78C (2018) 43–59. doi:10.1016/j.cose.2018.06.001.

[6] J. Vykopal, P. Čeleda, P. Seda, V. Švábenský, D. Tovarňák, Scalable Learning Environments for Teaching Cybersecurity Hands-on, in: 2021 IEEE Frontiers in Education Conference (FIE), IEEE, New York, NY, USA, 2021, pp. 1–9. doi:10.1109/FIE49875.2021.9637180.

[7] M. M. Yamin, B. Katt, V. Gkioulos, Cyber ranges and security testbeds: Scenarios, functions, tools and architecture, Computers & Security 88 (2020) 101636. doi:10.1016/j.cose.2019.101636.

[8] N. Chouliaras, G. Kittes, I. Kantzavelou, L. Maglaras, G. Pantziou, M. A. Ferrag, Cyber Ranges and TestBeds for Education, Training, and Research, Applied Sciences 11 (4) (2021). doi:10.3390/app11041809.

[9] E. Ukwandu, M. A. B. Farah, H. Hindy, D. Brosset, D. Kavallieros, R. Atkinson, C. Tachtatzis, M. Bures, I. Andonovic, X. Bellekens, A Review of Cyber-Ranges and Test-Beds: Current and Future Trends, Sensors 20 (24) (2020). doi:10.3390/s20247148.

[10] M. J. Aschmann, Towards a capability maturity model for a cyber range, Master's thesis, Rhodes University; Faculty of Science, Computer Science (2020).

[11] SANS Institute, SANS Cyber Ranges, https://www.sans.org/cyber-ranges/, [Last accessed on November 24, 2022] (2022).

[12] Hack The Box, Hack The Box, https://www.hackthebox.com, [Last accessed on November 24, 2022] (2022).

[13] TryHackMe, TryHackMe, https://www.tryhackme.com, [Last accessed on November 24, 2022] (2022).

[14] Circadence, Project Ares, https://projectares.academy, [Last accessed on November 24, 2022] (2022).

[15] Open CSIRT Foundation, SIM3 Online Tool, https://sim3-check.opencsirt.org, [Last accessed on November 24, 2022] (2022).

[16] A. Clear, E. Cuadros-Vargas, S. Takada, CC2020 Visualization Tool, https://cc.spc.org.pe, [Last accessed on November 24, 2022] (2022).

[17] A. Clear, E. Cuadros-Vargas, S. Takada, CC2020 Visualization Tool, in: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2, SIGCSE 2022, ACM, New York, NY, USA, 2022, p. 1063–1064. doi:10.1145/3478432.3499029.

[18] Cyber Range Organization and Design (CROND), CROND achievements: Cybersecurity training content, https://www.jaist.ac.jp/misc/crond/achievements-en.html, [Last accessed on November 24, 2022] (2022).

[19] J. Vykopal, P. Seda, V. Švábenský, P. Čeleda, Smart Environment for Adaptive Learning of Cybersecurity Skills, IEEE Transactions on Learning Technologies (2022). doi:10.1109/TLT.2022.3216345.

[20] J. Vykopal, V. Švábenský, P. Seda, P. Čeleda, Preventing Cheating in Hands-on Lab Assignments, in: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1, SIGCSE 2022, ACM, New York, NY, USA, 2022, p. 78–84. doi:10.1145/3478431.3499420.

[21] R. Gerhards, The Syslog Protocol, RFC 5424, https://rfc-editor.org/rfc/rfc5424.txt, [Last accessed on November 24, 2022] (March 2009).

[22] Elasticsearch, What is the ELK Stack? Elasticsearch B.V., https://www.elastic.co/what-is/elk-stack, [Last accessed on November 24, 2022] (2022).