



## REWIRE - Cybersecurity Skills Alliance A New Vision for Europe

---

# R4.3.1. REWIRE Cyber Range scenario and development framework



|                             |   |
|-----------------------------|---|
| <b>Title</b>                | REWIRE Cyber range scenario and development framework   |
| <b>Document description</b> | This document describes the REWIRE process to design and develop scenarios in the REWIRE Cyber Range. Also specifies the Scenario Sharing Platform. |
| <b>Nature</b>               | Public  |
| <b>Task</b>                 | T4.3  |
| <b>Status</b>               | Final version   |
| <b>WP</b>                   | WP4   |
| <b>Lead Partner</b>         | P10. MU   |
| <b>Partners Involved</b>    | P24. URL, P23. Caixa, P.21 Tecnico Lisboa, P25. KTH, P6. APIROPLUS, P20. BME, P18. Read-Lab, P22. Uni Telecom, P17. AMC, P1. MRU P2. EKT, P15 UL    |
| <b>Date</b>                 | 25/05/2023  |

### Disclaimer:

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

## CONTENTS

|   |           |
|---|-----------|
| <b>1. Executive Summary .....</b>                         | <b>4</b>  |
| <b>2. Introduction .....</b>                              | <b>5</b>  |
| <b>3. State of the Art .....</b>                          | <b>6</b>  |
| <b>4. Cyber Range and scenario-based learning .....</b>   | <b>14</b> |
| 4.1. Basic Theory .....                                   | 14        |
| 4.2. Open Format .....                                    | 18        |
| <b>5. Scenario DESIGN Process .....</b>                   | <b>19</b> |
| 5.1. Analytical Phase .....                               | 19        |
| 5.2. Definition Phase .....                               | 20        |
| 5.3. Development Phase .....                              | 21        |
| 5.4. Delivery Phase .....                                 | 21        |
| 5.5. Improvement Phase .....                              | 21        |
| <b>6. Scenario IMPLEMENTATION IN KYPO CRP .....</b>       | <b>22</b> |
| 6.1. Terminology .....                                    | 22        |
| 6.2. User roles .....                                     | 23        |
| 6.3. KYPO CRP Architecture.....                           | 25        |
| 6.4. Overview of the scenario implementation process..... | 26        |
| 6.5. Sandbox deployment detail .....                      | 28        |
| 6.5.1. Sandbox definition .....                           | 28        |
| 6.5.2. Pools.....   | 38        |
| 6.6. Training deployment detail.....                      | 41        |
| 6.6.1. Training definition.....                           | 41        |
| 6.6.2. Training instance.....                             | 58        |
| 6.6.3. Training run.....                                  | 61        |
| 6.7. Summary .....  | 67        |
| <b>7. Scenario Sharing platform.....</b>                  | <b>68</b> |

---

|   |           |
|---|-----------|
| 7.1. Open Content .....                             | 68        |
| 7.2. Marketplace .....                              | 68        |
| 7.3. Scenario Sharing Platform Requirements .....   | 69        |
| 7.3.1. Web-based access .....                       | 69        |
| 7.3.2. Role-based access control .....              | 69        |
| 7.3.3. Version control .....                        | 69        |
| 7.3.4. Code review .....                            | 69        |
| 7.3.5. Continuous integration .....                 | 70        |
| 7.3.6. Training scenario production levels .....    | 70        |
| 7.3.7. KYPO support .....                           | 70        |
| <b>8. Conclusions .....</b>                         | <b>71</b> |
| <b>9. References .....</b>                          | <b>72</b> |
| <b>10. List of Abbreviations and Acronyms .....</b> | <b>75</b> |
| <b>11. List of Figures .....</b>                    | <b>77</b> |
| <b>12. List of Tables .....</b>                     | <b>79</b> |

## 1. EXECUTIVE SUMMARY

With a growing need for more cybersecurity experts, cyber ranges started being utilized as state-of-the-art tools for hands-on training and education several years ago. Unfortunately, cyber ranges are not as widespread as they could be till today due to two main constraints – the cost of the cyber range itself and the lack of available quality content. The latter issue is addressed in this document. Scenario for a cyber range, instead called content, is a complicated multidisciplinary project if it is supposed to be built correctly and have high quality. It is necessary to use knowledge of IT and network engineering together with knowledge of virtualization, andragogy, and cyber security.

The document focuses on defining basic knowledge for scenario-based learning in cyber ranges and later defining a scenario design process and its implementation in the KYPO Cyber Range Platform (CRP). The design process is based on common continual improvement processes like PDCA (*Plan Do Check Act*) and should provide a generic checklist and guidance during content development for any cyber range. More technical details are made available for KYPO CRP and scenario implementation there. The last part of the document is focused on a scenario-sharing platform that is crucial for creating a common place for collaboration and exchange of scenarios between organizations actively using KYPO CRP to train professionals or educate students.

## **2. INTRODUCTION**

After deploying the REWIRE Cyber Range, powered by KYPO CRP, the next step in developing REWIRE courses is to provide a tool for partners and other stakeholders interested in defining, designing, and implementing scenarios in the Cyber Range. These scenarios simulate real-world challenges in a risk-free, controlled environment and can be customized to specific learning objectives and scenarios, from basic training to advanced persistent threat simulations. Participants use tools like virtual machines and attack scripts to develop practical cybersecurity skills, improving their ability to detect, respond to, and prevent cyberattacks.

This document proposes a state-of-the-art approach to demonstrate various perspectives on creating exercises and hands-on activities. It also presents a scenario-based learning methodology and recommended steps to follow. After outlining the overview and methodology, it presents the scenario design process and the different phases to follow. Regarding the REWIRE Cyber Range and how to deploy scenarios, the document provides a scenario implementation description, showing the platform's terminology, architecture, and the detailed process for developing scenarios. Finally, it describes the scenario sharing platform, including its approach and requirements.

### 3. STATE OF THE ART

Cybersecurity is an increasingly important topic in today's digital world, as businesses and individuals become more reliant on technology for daily operations and communication. However, the need for cybersecurity measures often outstrips the number of experts available to implement and maintain them. This can lead to vulnerabilities and breaches that can have serious consequences for individuals, businesses, and even nations.

Europe is particularly affected by the shortage of cybersecurity experts, as the demand for these skills has grown rapidly in recent years. This has led to a skills gap in the field, which is exacerbated by the fact that cybersecurity is a complex, multidisciplinary and constantly evolving area that requires ongoing training and education [1]. Discovering, addressing, and updating the necessary security skills is not an easy task. Given the need to find cybersecurity professionals, companies, governments, and academia have defined their needed profiles and related skills in an ad-hoc manner, resulting in a large set of diverse definitions for the same needs. This has led to the need for homogenization of these profiles and skills to align the European perspective and thus be able to create more efficient training programs that are capable of meeting the current and future training needs. The most remarkable outcome of these efforts is ENISA's ECSF [2], which has successfully redefined/reduced cybersecurity profiles obtaining 12 profiles covering all necessary current skills (it also has an updating process for the inclusion of future skills).

To address the skills gap, European governments and businesses are investing in education and training programs to help develop the next generation of cybersecurity experts. Some countries also offer incentives to attract talent from abroad, while others partner with academic institutions to create specialized cybersecurity programs [3]. To ensure that training is effective and meets the needs of Europe, cybersecurity education and training programs should focus on providing training for the 12 profiles identified in ENISA's ECSF.

Despite these efforts, the shortage of cybersecurity experts remains a challenge, and it is likely that demand will continue to outstrip supply for the foreseeable future. This highlights the need for continued investment in education and training programs and the importance of promoting cybersecurity as a career path to attract more talent to the field. There are multiple initiatives across Europe that encourage the importance of cybersecurity from an early age, which can lead to increased interest from young people in adopting a career in cybersecurity later on [4]. Providing a good guide to teachers of the basic levels of education is crucial. A guide with attractive and interesting content that arouses curiosity to specialize in cybersecurity field, at the same time that it promotes awareness of cybersecurity and highlights the need for experts in the cybersecurity sector. The CONCORDIA project proposes high-school teachers teaching methodology and materials for them to adopt with their pupils (covering cybersafety and cybersecurity topics) [4].

Creating training programs that adapt to current cybersecurity needs is not trivial. According to a study conducted by the CONCORDIA project [5], numerous cybersecurity courses are available in the market, with many free options on MOOC platforms. These courses are particularly attractive to employees, as they offer flexibility in terms of studying time and can be tailored to their professional commitments. Additionally, face-to-face courses for middle and senior managers, as well as technical experts training in cyber ranges, are popular choices. The study also highlighted several popular cybersecurity learning platforms, including those identified by CONCORDIA and ECSO surveys: Coursera, edX, LinkedIn Learning, Cybrary platform, ISACA online, offline and mixed courses at different levels, Udacity platform and Cyberwiser. There are also multiple offers of European Master's educational programs (university level) that cover the different skills and knowledge needs within cybersecurity. The CyberSec4Europe project collects a large number of them, classify the skills and knowledge they cover and the country members of the EU that offer these courses [6]. It should be noted that while the listed cybersecurity educational solutions in the EU are aimed at the same market, each platform has structured its course content based on the education provider model, without reference to any common competence or skill framework. As a result, it becomes challenging to compare the various offers and their appeal [7]. Then, the current training programs suffer from a lack of consistency in incorporating a competency framework and career path in their design. As a result, individuals face difficulty in selecting the appropriate course to meet their educational needs or professional requirements. Various challenges hinder cybersecurity education, such as a shortage of cybersecurity educators, inadequate industry engagement, limited knowledge of the job market, outdated or impractical educational platforms, and difficulties in keeping up with external developments. The CONCORDIA project has created a *“Methodology for the creation and deployment of new courses and/or teaching materials for cybersecurity professionals”* [7] that aims addressing this issue by considering the actual needs of both the industry impacted by cybersecurity and the industry professionals. On the other hand, CyberSec4Europe project dedicate part of its efforts to establish a framework for cybersecurity professional categories and a skill level scale to aid in developing educational resources and criteria for demonstrating qualifications. The framework proposed enables the visualization of essential cybersecurity skills and provides guidelines and tools for designing capability building instruments. This includes identifying knowledge units and curricula, specifying learning objectives and competencies, developing training and awareness, and conducting activities to apply and test such competencies [8][9]. The pilot projects CONCORDIA and CyberSec4Europe have dedicated resources to create a European education ecosystem that addresses the needs of cybersecurity skills and knowledge.

Regardless of the structure of the training course, the competencies it covers, and its objectives, there is a key issue to address: what is the most efficient way to teach cybersecurity. Several methodologies can be effective for teaching cybersecurity, depending on the audience, the level of technical expertise required, and the training goals. Some examples are presented below:



- **Hands-on training.** It involves providing practical exercises and simulations that enable students to apply their knowledge in real-world scenarios. It is especially effective for teaching technical skills such as network security, vulnerability assessments, and penetration testing.
- **Scenario-based learning.** It involves presenting students with realistic scenarios that require them to apply cybersecurity concepts and techniques to solve problems. Scenario-based learning can help students develop critical thinking skills and prepare them to handle real-world security incidents.
- **Gamification.** It involves turning cybersecurity training into a game, which can make the learning experience more engaging and interactive. Gamification can be especially effective for teaching cybersecurity to non-technical employees, who may find traditional training methods dull or intimidating [10][11][12].
- **Blended learning.** It combines different types of training, such as online courses, classroom lectures, and hands-on exercises, to provide a comprehensive learning experience. Blended learning can help accommodate different learning styles and provide flexibility for students with busy schedules.
- **Certification programs.** It involves providing training that prepares students for industry-standard certification exams, such as CompTIA Security+ or Certified Information Systems Security Professional (CISSP). Certification programs can provide a clear path for students to demonstrate their expertise and enhance their job prospects.

The most effective methodology for teaching cybersecurity will depend on the specific needs and goals of the students and the organization providing the training. REWIRE's approach to conducting these training courses resembles the Blended Learning method, leveraging the potential of VLEs (Virtual Learning Environments) to incorporate online courses that combine theory and practice. The practical aspect can be covered in various ways, through hands-on training or scenario-based learning, depending on the course and the training needs and objectives for each profile addressed. The crucial aspect here, particularly for the training of more technical skills, is how to support the practical learning.

In the last few years, Testbeds and Cyber Ranges focused on the cybersecurity sector have proliferated precisely to respond to the technical training needs of professionals and newcomers in the industry. The systematic literature review of unclassified cyber ranges and security testbeds done in [13], found that interest in Cyber Ranges and security Testbeds has increased in recent years and scenarios play a major role in their development for testing, experimentation, and education. These scenarios are executed on emulated, simulated, hybrid, and real equipment environments, and can be either static or dynamic. Most use cases focus on red and blue team training, but attention needs to be given to white and green teams for scenario development and management. There is a trend towards the use of autonomous teams to reduce the time required for cyber security exercises, tests, and experiments, but

methods to model their behavior are missing. Hybrid environments that combine emulation, simulation, and real equipment are used to create realistic cyber security environments. Despite some aspects that need more attention and improvement in their implementation, seems that both Cyber Ranges and Cybersecurity Testbeds are viable solutions to support education and training in the necessary skills at the European level.

Cyber Ranges provide trainees with closed and controlled environments that include all the necessary tools, networks, and user simulations for training and education purposes. By doing so, they enable trainees to practice realistic scenarios that would otherwise be impossible to execute, while minimizing the risk of a threat getting out of control. Once the great usefulness of cyber ranges for cybersecurity education has been determined, it is necessary to know how to operate them and how to create the most suitable training scenarios to address the necessary skills according to the training that is to be carried out. There exist several platforms like Deterlab, eLearning Security, The Hacker Accademy, Offensive Security, SmallWorld, KYPO CRP, among others [14]. While the documentation for each platform thoroughly explains its characteristics, functionalities, architecture, and modules, there is a lack of standardization in scenario building. Although each platform typically provides toolkits or a Scenario Definition Language to assist users in designing their exercises, it is unclear which important aspects need to be considered when creating a scenario for a specific purpose or for covering specific needs.

Researchers in [13] present a generic definition of how a cybersecurity scenario inside a cyber range should be deployed or which things should be considered to deploy it. A scenario defines the execution environment and the storyline with the execution steps of the training exercise to be conducted. A generic characterization of a scenario, as presented in this systematic review of several cyber ranges, addresses the following elements to be developed:

- **Purpose.** To define the objectives of the scenario. For Cyber Range environment this means, i.e. the execution of a cyber security training exercise (education) or the experimentation validation of new cyber security tools and techniques (research). We can extrapolate this concept as the purpose desired of the training under development.
- **Scenario Description.** It represents all the information and meta-information required to deploy and execute the scenario. The scenario description should include at least the scenario model definition which describe (depending on the overall capabilities of the CR): virtual machines/containers, networks and connections with the machines/containers, the storyline, monitoring hooks, tool configuration. Beside the required information to provision the environment, other meta information could be included: title, purpose, scoring, type, hints, writeup, different categorizations etc. The scenarios can be statically defined based on specific scripts or use some form of Infrastructure as Code (IaC).

- **Storyline.** It describes how the exercise will be executed. The actions that can be done and events that build the scenario, and also triggered events that are different than those expected, depending on students' actions. The storyline constitutes the overall understating and controlling of the scenario (how the exercise will be executed).
- **Type.** It can be static or dynamic. In a static scenario no changes are applied during the execution of the exercise. In a dynamic scenario there is a dynamic component that will make changes during the execution of the scenario. For example, a simulator, or a traffic generator that can be injected, or executed, during the exercise.
- **Domain.** The domain indicates the application domain of the scenario. Examples: hybrid network applications, Networking, SCADA systems, social engineering, IoT systems, critical infrastructure, Cloud based systems, and autonomous systems.
- **Tools.** The tools needed for the creation of the environment of the scenario, and/or the tools which are used in the development of a storyline.
- **Lifecycle/Management.** It involves creating, generating, editing, deploying and executing a cyber security scenario (components, dashboards, automation techniques). (1) Creation/editing: A designer dashboard or components to generate cyber security scenarios using different automation techniques. Mostly Cyber Range scenarios are created in human readable languages like XML and JSON. (2) Deployment: components responsible for deploying network resources, applications, vulnerable software or systems, etc. (3) Execution: module that can control the scenario flow, like start, stop and pause scenario execution (orchestration). (4) Generation: components that are used to generate different events within the scenario execution (i.e. automatic attacks, traffic generation).

As described in [15], a scenario should address an existing potential cyber incident and a clear characterization is needed to (1) provide a common terminology in scenario description facilitating communication among the community and (2) classify the literature in identifying emphasized and overlooked study areas. Researchers conclude that finding elements that can be included in a typical cybersecurity scenario is important. They define the following broad scenario elements: (1) An attacker, (2) Users that the attacker targets initially, (3) A cyber system and data that the attacker targets, (4) System security personnel that detects the incident, (5) Interactions between the attacker, users, the system and the security personnel and, (6) A wide network infrastructure that facilitates connection between cyber systems and people.

In [16] researchers of Cyberwiser project describe a scenario development method composed of two major components. The first, the **Scenario Design Workflow (SDW)** to describe the steps of the development method, the participating roles and their interaction. It follows an iterative approach between its steps to allow the creation of complex scenarios. The different steps are: (1) **Designing** - Description of network design, application configuration and timeline of events which generates the Scenario Design Request (SDR), (2) **Creation** - Creates

the scenario and assess whether all the necessary information regarding the configuration of the advanced platform components has been provided, (3) **Validation** - The scenario is checked for completeness, correctness and any asset requests, (4) **Instantiation** - The scenario is instantiated and the next step/component is notified, (5) **Testing and configuration** - The scenario is tested to verify that it meets the needs of the training and, (6) **Finalization** – Scenario is ready to its execution. The second component, the **SDR** which contains all the necessary information required to create a scenario. It is an Step-by-step approach, composed by the following steps:

- **General Scenario information.** Overview (Name, description, motivation, Duration, Category, Type).
- **Network topology.** infrastructural layout of the scenario. Network topology diagram and the required information for each component. Name, description, operating system, virtual machine details, Policies (when applies), and network interface of gateways, workstations, servers. Network (virtual switch), external network (switch to connect to real HW/world), Network appliance (representing physical device), Server appliance (physical server).
- **Application configuration.** Name, description, version, Configuration details of each application.
- **Timeline.** Definition of events and triggered events (start, stop, actor, action, event).
- **Access control and visibility.** Role Access Based Control mechanism.
- **Performance Evaluation.** via the Performance evaluator (scenario monitoring, impact assessment, and the provision of a captured flag or test in a questionnaire). Name, description, evaluation method.
- **Scenario files.** Files necessary to the execution of the scenario. Possible files to be included: Exercise description, Step-by-step instructions, Packet captures, Malwares, Configuration files, Disk images files, Memory image files.
- **Questionnaire.** Multiple questionnaires can be added. Name of questionnaire, description, passing score, user, question 1, answer mode, answer, answer score, question hints, penalty points.

Moreover, 11 different examples and scenario designs are presented in [16], each one of them explained with Description, Network topology, Training flow and Scenario files.

When designing and implementing a scenario, it is important to consider incorporating a built-in Q&A system or similar activities. This feature allows for the collection of valuable information on the trainees' learning progress. If the Cyber Range includes tools and mechanisms to perform and visualize the results, trainees can receive instant feedback on their performance and progress during the exercise. Additionally, trainers and instructors can track the level of knowledge acquired by the trainees effectively.

Considering the reviewed literature, creating an effective cybersecurity exercise that will help participants develop the skills and knowledge they need requires careful planning and execution. The following are fundamental steps to consider:

- **Define the objectives.** The first step is to clearly define the learning objectives of the exercise. Determine the specific skills and knowledge you want participants to learn through the exercise (what are the specific cybersecurity threats the exercise will address?). Defining clear learning objectives will help to design an exercise that is effective in meeting the skills addressed.
- **Choose the scenario.** Choose a realistic scenario replicating real-world cybersecurity threats and allowing participants to practice the skills and knowledge they should learn. The scenario should be challenging but not overwhelming. Scenario should consider participants' roles and skill levels.
  - **Select the appropriate tools and software.** Choose the tools and software that will be used in the exercise. This may include virtual machines, networking equipment, operating systems, and cybersecurity tools. Make sure that the tools and software are appropriate for the skill level of the participants.
  - **Design the environment.** Create a simulated environment to replicate the real-world scenario where participants can practice their skills. This may include setting up virtual machines, configuring networks, and creating security vulnerabilities that participants will need to identify and fix.
- **Develop the exercise materials.** Create the materials that participants will need to complete the exercise, such as lab manuals, instructions, case studies, mock systems, and data sets. Make sure that the materials are realistic and relevant to the scenario and also clear, concise, and easy to follow.
- **Define the rules of the exercise.** Define the rules of the exercise, such as the time limit, the objectives, and the evaluation criteria. Make sure that the rules are clear and easy to understand and clearly show the exercise's expectations and limitations.
- **Conduct a pilot test.** Before conducting the exercise with a large group, conduct a pilot test with a small group to identify any issues or challenges that need to be addressed.
- **Conduct the exercise.** Conduct the exercise with the participants, providing guidance and support as needed. Encourage participants to work together and share their knowledge and expertise.
  - **Monitor progress and provide feedback:** Monitor participants' progress throughout the exercise and provide feedback to help them improve their skills. This can include identifying areas for improvement, providing suggestions for alternative approaches, and recognizing areas of success.

- **Evaluate:** After the exercise is complete, debrief the participants to discuss what worked well and what could be improved. Evaluate the exercise based on the defined objectives and make any necessary adjustments for future exercises.

Designing and implementing a cybersecurity exercise tailored to the learning of specific skills takes a lot of time. In fact, one of the limitations encountered regarding the creation of exercises and scenarios for Cyber Ranges is that sharing this work is difficult because each platform uses its solution for scenario design and implementation. One of the goals in developing Cyber Ranges should be to standardize these developments to make sharing easier [17][18].



## 4. CYBER RANGE AND SCENARIO-BASED LEARNING

### 4.1. Basic Theory

Cybersecurity education is full of challenges. The current knowledge and trends are transferred to the educational process with a delay, due to the constant development of this field. The practicality and usefulness of this education is also an issue – the central theme of this concern is training versus education [19]. While education tends to focus on the reasons, the theory and the mechanisms behind the material, training supports both the creation of a sustained workforce pipeline and the professional development. That is why we need a mechanism to ensure that students gain required knowledge, skills, and competencies.

Such a mechanism can take the form of a specific methodology, with the help of which we define important parameters for effective cybersecurity education. For example, under these steps:

1. Choose a Framework,
2. Define the Learning Goals,
3. Scenario Design and Deployment,
4. Evaluation [20].

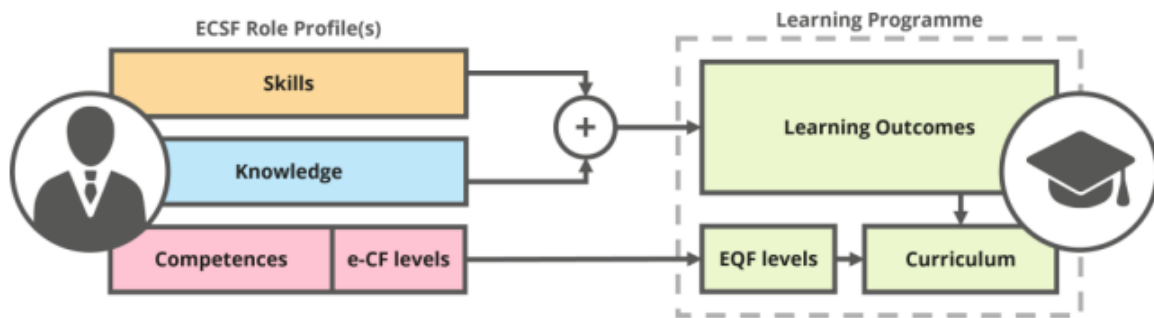
### Choose a Framework

The choice of a cybersecurity educational framework is mainly defined by the learning needs related to knowledge, skills and abilities that are needed to develop. In this case, most relevant cybersecurity frameworks are European Cybersecurity Skills Framework (ECSF) by ENISA and MITRE ATT&CK™ framework.

European agency ENISA focused on cybersecurity, developed a European Cybersecurity Skills Framework. The framework is used as a tool to help identify and articulate tasks, competencies, skills, and knowledge associated with the roles of European cybersecurity professionals. It helps to create terminology for better understanding between individuals, employers, and providers of learning activities. Simultaneously the framework is used to reduce knowledge gaps and brings many other benefits to different target groups, such as policymakers and government stakeholders, organizations, professional associations, etc. It also helps to specify offers/demands in the labor market [21].

There is name and define 12 cybersecurity roles in total with their competencies, knowledge, and responsibilities – Chief Information Security Officer (CISO); Cyber Incident Responder; Cyber Legal, Policy & Compliance Officer; Cyber Threat Intelligence Specialist; Cybersecurity Architect, Cybersecurity Auditor; Cybersecurity Educator; Cybersecurity Implementer; Cybersecurity Researcher; Cybersecurity Risk Manager; Digital Forensics Investigator and Penetration Tester [2]. Each defined role describes in detail information such as their alternative titles, deliverables, main tasks, essential skills and knowledge, e-Competences, etc.

In addition, the framework holds up the design of cybersecurity-related training programs, redesigning curriculums or collaboration across institutions in learning programs. It can be used to define learning outcomes, difficulty, evaluations, assessments, or needed skills and knowledge. Another advantage is an easy expansion to Europe thanks to the unification of terminology.



**Figure 1: ECSF profiles guiding cybersecurity professional learning [2]**

It is essential to always consider technological developments, rapidly changing requirements, and the emergence of new job positions.

The MITRE framework is known as a knowledge base following up real-world cyber adversary tactics and techniques. The framework reflects the various phases of an adversary's attack lifecycle and the platforms they are known to target. It can be used to identify security gaps and prioritize mitigations based on risk. MITRE ATT&CK Framework covers these fourteen categorized areas – Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Command and Control, Exfiltration, Impact. In 2020, there are about 245 techniques in the Enterprise model, with MITRE regularly updating the discovered techniques by both cybersecurity researchers and hackers alike [22]. Each technique is described in more detail, connected to the related list of sub-techniques, a list of known mitigation methods and detection methods, written down related references, and additional resources related to the technique.

MITRE ATT&CK framework can be used as an inspiration while creating learning scenarios based on real situations. It can help participations get the required knowledge to detect threats to their organizations. Emphasis on practicing red and blue team skills may be applied.



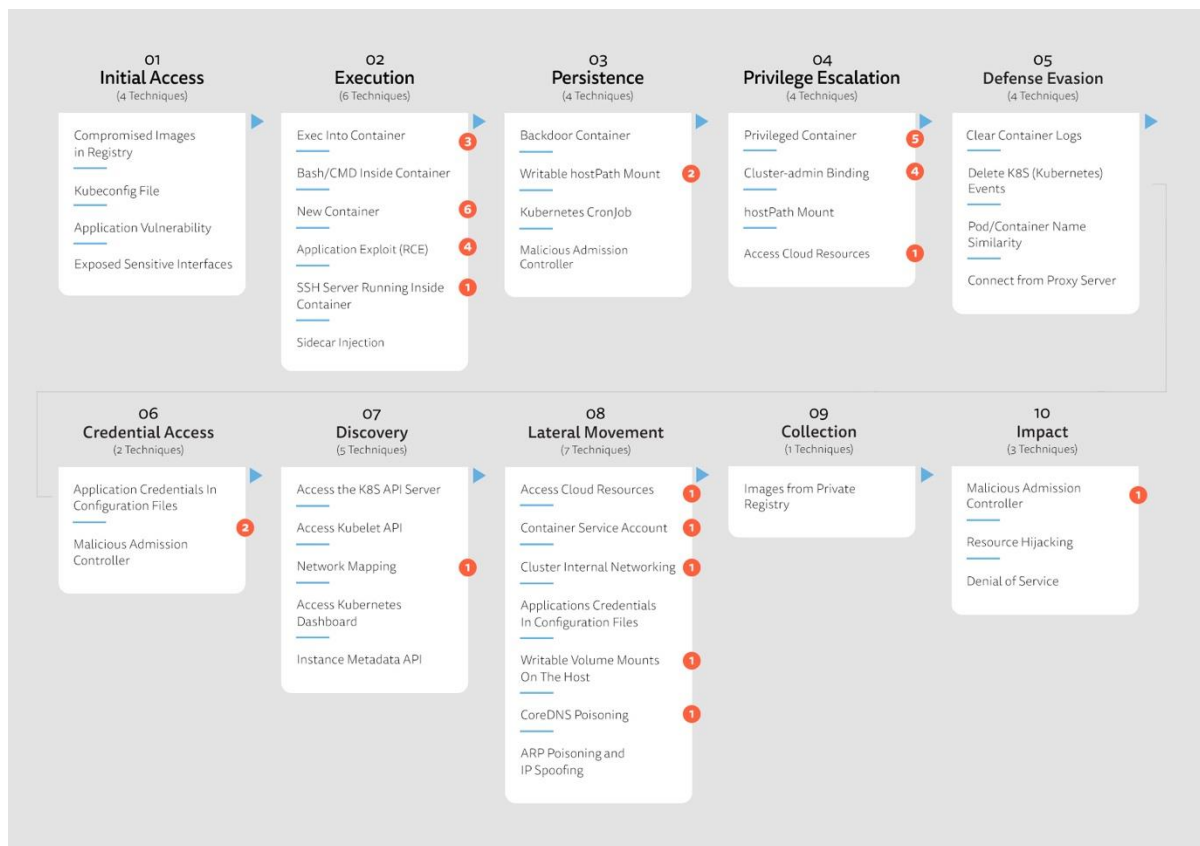


Figure 2: MITRE ATT&CK Matrix for Kubernetes [23]

## Define the Learning Goals

To define the learning goals, we can use several approaches. Bloom’s taxonomy, created by Benjamin S. Bloom is one of the frequently used tools for defining the cognitive aims of education. It can be used while creating own educational scenarios because it helps to determine a difficulty, plan, and control the scholarly outputs. Bloom’s taxonomy comprises six areas to connect and develop each other. They are expressed by words:

- Knowledge,
- Understanding,
- Application,
- Analysis,
- Synthesis,
- Evaluation.

These mentioned words are often complemented by other action words, primarily verbs helping to specify practical skills and abilities. For example, knowledge is linked to words such as define, describe, and memorize; application is related to words such as prepare, apply, and solve; analysis is related to words such as differentiate, select, or classify.

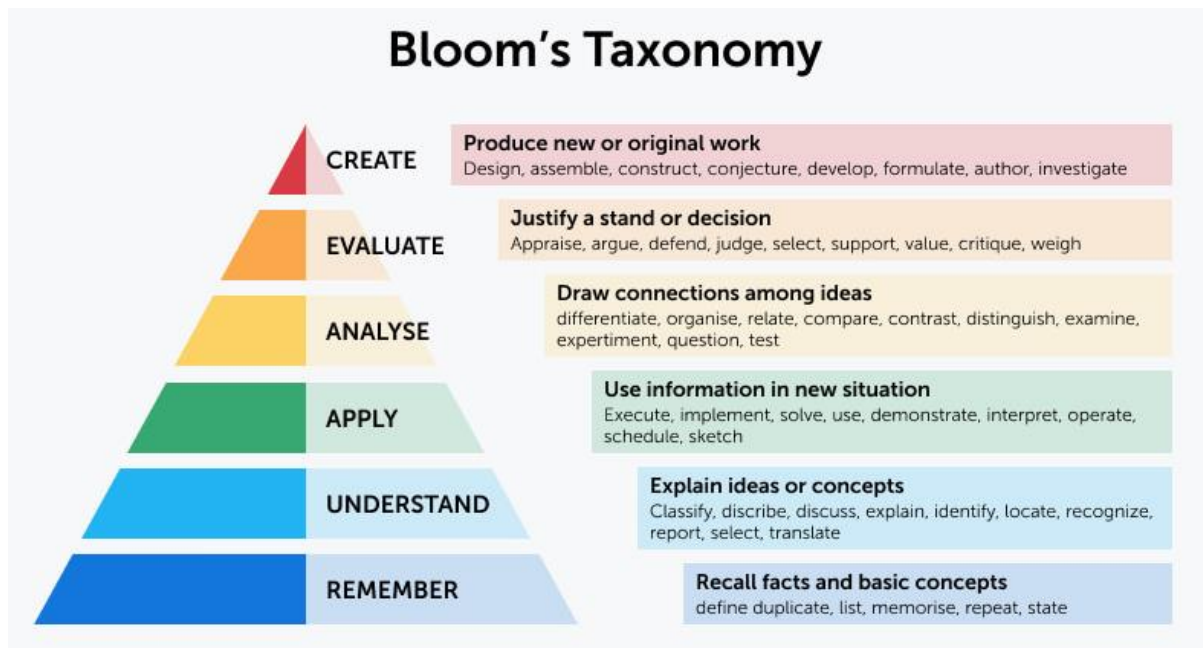


Figure 3: The example of revised Bloom's Taxonomy [24]

Using Bloom's taxonomy while creating the scenario can help meaningfully formulate individual units and avoid favoring higher-level outcomes over lower-level ones, which is important for making the required progress. Concurrently could help to design valid assessment tasks and strategies and even plan and deliver appropriate instruction [25]. Bloom's taxonomy can be used as a tool for an individual attitude where needed. It also helps students to track their own study path and progress. Eventually, they can base their new learning goals and actively try to achieve them.

## Scenario Design and Deployment

One way to meet the learning objectives is to use real-life situations in which the scenario plays a major role. This so-called scenario-based learning provides a relatable and relevant learning experience through an immersive and highly engaging approach [26]. A scenario is seen as a postulated sequence of possible events that allow students to actively engage in the learning process.

To be fully effective, the dry run phase should be implemented within this step. In the dry run phase, different scenario properties are checked to identify whether the deployed scenario fulfills the specified requirements in the scenario model [27]. It can be divided into manual and automated testing both executed by professional preparing the cyber range.

Creating learning experiences which appeal to personalized characteristics, specific skillset and background knowledge might improve the motivation rates of the learning programs and enhance the learning outcomes [28].

Therefore, a systematic approach should be used to set learning outcomes from existing approaches that involve security scenarios. This can result in some kind of taxonomy according to the details of the scenario. More specifically, Cyber Ranges should define the following:

1. the purpose of the scenario,
2. the topology and the environment where the scenario takes place,
3. a basic storyline that the trainees follow,
4. the type of environment (static or dynamic),
5. the domain and the main topic,
6. the tools which will be used [29].

To be fully functional, other important factors such as motivation or engagement must be considered. In the Keller's model of motivation [30], the concepts of Attention, Confidence, Relevance and Satisfaction are specified. Those can be easily linked to described scenario design that improves learning outcomes in case of immersion and engagement. Another element that supports scenario-related motivation is gamification. This is characterised by the embedding of game elements into the scenario passage (e.g. problem solving challenges, rewards and scoreboards, storytelling elements).

## **Evaluation**

Scenario lacking realism and flexibility can create barriers for learning. In this case, using some evaluation methods is highly recommended. Measurement of skill progression is important to demonstrate what the participant has learned. Improvement can be quantified through the continuous assessments (mostly before and after the passing the scenario). An assessment can demonstrate learning for the participant and highlight areas of the scenario that need to be refined for maximum learning [31].

### **4.2. Open Format**

As mentioned, in cybersecurity we face a noticeable shortage of experts, but also skills gaps and lack of diversity. One of the reasons for this may be the different was the lack of an even distribution of cybersecurity education in the EU member states [1]. That is why we need to make this education as accessible as possible to all and promote its openness. One way to do this is to use open formats for the description of technical environment, through content, to ready-made artifacts applicable in various contexts [32]. For example, KYPO Cyber Range Platform uses JSON (Java Script Object Notation) syntax for its excellent capability to represent objects with minimum or no changes in data structure. This allows already created cybersecurity scenarios to be used by multiple entities without much difficulty in sharing them.

## 5. SCENARIO DESIGN PROCESS

This chapter covers the scenario design process of hands-on cybersecurity training for a cyber range. The process is created in a generic way to suit internal and external customers, so the process can be used to prepare training for in-house training, student education, and professionals of an external organization. The process is also designed as platform-independent as possible needless to say, the utilization of KYPO Cyber Range Platforms for above-mentioned education activities strongly influenced it.

The authors' goal is to create a process that allows standardized and repeatable development of hands-on scenarios for training and to limit misunderstandings between various parties with different points of view and experiences. For these reasons, the design, development, and execution of a hands-on training scenario for a cyber range platform is covered.

The theory behind the proposed scenario design process is strongly based on continual improvement processes (e. g. PDCA) and agile development ideas. Utilization of available frameworks is also strongly advised. The main reasons are to adapt well-known and established development management methods rather than developing new ones and providing common communication ground with already available frameworks.

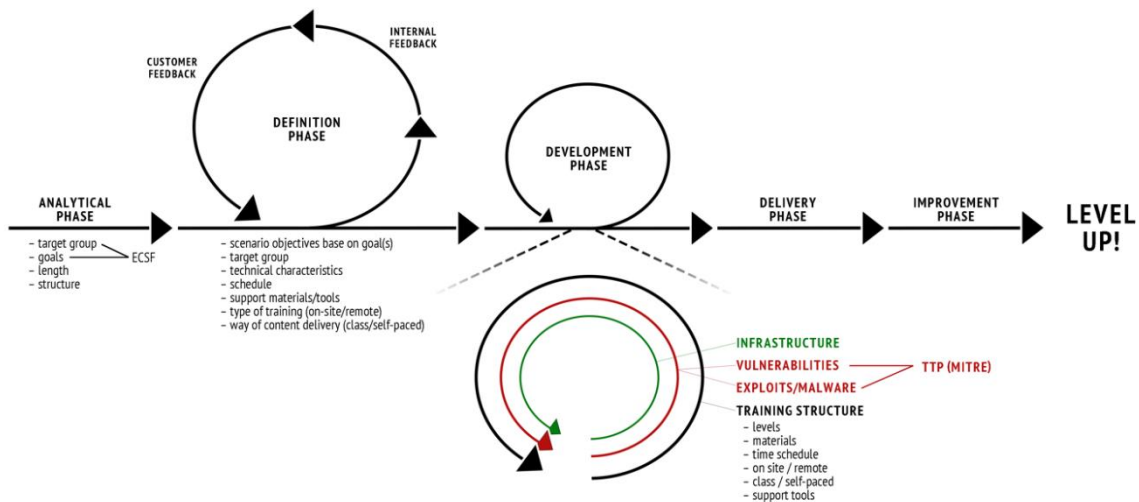


Figure 4: Scenario Design Process

### 5.1. Analytical Phase

The analytical phase covers the first step of the whole process. Its purpose is to define initial requirements and boundaries with a customer. It can be perceived as a high-level view of scenario development and training, similar to any canvas method. The goal is to cover all the bases (e. g., analysis of needs, target group, goals, length, and structure), not going into much detail.

The person running this initial analysis needs to be highly skilled, especially when dealing with external customers, to be able to deliver training according to customer needs. Checklists and structured forms are extremely helpful in this phase.

## 5.2. Definition Phase

The essential task in the definition phase is taking input from the initial analytical phase and defining the scenario itself. First, are defined scenario objectives based on the goal(s) and target group, and second are defined technical characteristics, schedule, support materials/tools, type of training (on-site/remote), and way of content delivery (class/self-paced). It is strongly advised to require internal and customer feedback at this point.

An accurate description at this moment ensures smooth communication between education experts, experts on cybersecurity, and system administrators or another role responsible for creating the physical representation of the scenario in the respective cyber range. A simple table, as shown below, can be a valuable tool for maintaining clarity.

### Topology Definition

| Machine  | OS            | IP        | Domain Name | User Accessible |
|----------|---------------|-----------|-------------|-----------------|
| Attacker | Kali (latest) | 10.0.10.5 | N/A         | Yes             |
| Victim   | Debian 10     | 10.0.0.20 | N/A         | No              |

### Attacker

| Service | Version | Port | User and password |
|---------|---------|------|-------------------|
| System  | N/A     | N/A  | kypo:kypo         |

### Victim

| Service    | Version          | Port | User and password  | Attack                            | Exploit                                     |
|------------|------------------|------|--------------------|-----------------------------------|---|
| System     | N/A              | N/A  | user:networking    | Nmap scan                         |   |
| vsftpd     | 2.3.4 (infected) | 3000 | same as the system | Metasploit attack                 | exploit/unix/ftp/vsftpd_234_backdoor        |
| wordpress  | x                | 80   | admin:adminadmin   | Wpscan attack + Metasploit attack | exploit/usenix/webapp/wp_admin_shell_upload |
| ssh server | x                | 22   | same as the system | Bruteforce with Hydra             |   |
| mysql      | x                | 3306 | --                 | ---                               |   |

### Requirements

- Openstack project with default images
- KYPO instance running in the Openstack project
- Account for the KYPO instance with privileges to create and run trainings and sandboxes
- SSH access to the kypo-proxy-jump

**Figure 5: Description of the scenario in the design phase**

### 5.3. Development Phase

The development phase is focused on the development of the scenario itself, and it is divided into two parts.

The technical part focuses on developing a virtual infrastructure like networking and virtual machines with requested services, users, and other configurations. It is crucial to keep in mind that implementation of vulnerabilities, exploits, and malware can be requested. The developers should build repositories of vulnerable software, exploits, and malware for their scenarios. Public services can be closed, or vulnerable services replaced in the repositories. All these events may render the scenario inoperable.

The development of the educational part is mainly focused on structuring the training, preparing support materials, and grading if needed.

Like any other software, testing the scenario before actual use during training is necessary. Two types of tests should be deployed to ensure the goal. The first type is focused on testing that the scenario is working as intended, services are up and running, etc. The second acceptance test is focused on determining if the developed scenario (training) delivers what was promised and if it has an appropriate difficulty level.

### 5.4. Delivery Phase

The delivery phase is centered around executing the scenario by running a class for students or training professionals. It is important to collect feedback on the developed content in both parts. The easiest way is to collect feedback from trainees immediately as an integral part of the class/training. From the author's experience, the questionnaires' return rate and the answers' quality decreases heavily over time.

The other way of delivery is releasing the scenario for later use through uploading to a marketplace. In this case, processes must be in place to ensure the quality of the delivered content and that feedback reaches the authors of the scenario.

### 5.5. Improvement Phase

The last phase of the process is focused on incorporating collected feedback. Its goal is to maintain or upgrade a developed scenario continuously. Due to the nature of most of the materials is recommended to use a version control system to keep track of changes and to roll back to the previous version if needed.



## 6. SCENARIO IMPLEMENTATION IN KYPO CRP

### 6.1. Terminology

To begin the description of the scenario implementation the KYPO CRP, it is essential to be familiarized with the terminology used in the platform. The following basic terms will be used in the following sections:

#### Emulated Virtual Environment

When creating and using an emulated virtual environment, the following terms are used:

- **Sandbox:** This is an isolated testing environment with virtual networks that enable users to connect to virtual machines (VMs) and communicate with the rest of the network, the host machine, and other VMs. Programs can be run, network services can be used, or network traffic can be monitored within the VMs without affecting the external infrastructure. There are currently two types of sandboxes:
  - **Cloud sandbox:** This runs within the cloud and is remotely accessible.
  - **Local sandbox:** This is created within a local computer using the Vagrant tool along with the VirtualBox virtualization tool.
- **Sandbox Definition:** This defines the internal structure of the sandboxes (networks and hosts) and user customization of the hosts. It consists of two parts:
  - **Topology Definition:** The file with the sandbox structure definition (hosts, routers, networks, etc.).
  - **Sandbox Provisioning:** It is used to customize Topology Instances, e.g., set up an environment, create users, install packages, etc. Sandbox Provisioning must specify how to connect to instances, e.g., user name and SSH key. The Ansible tool is used to perform these actions.
- **Pool:** This is a group of cloud sandboxes created based on the same sandbox definition.

#### Training

KYPO training is centered around the sandbox, where trainees solve tasks presented in KYPO GUI. KYPO training can also contain questionnaires to collect feedback from trainees or tests to assess their knowledge. The following terms are used in the context of training:

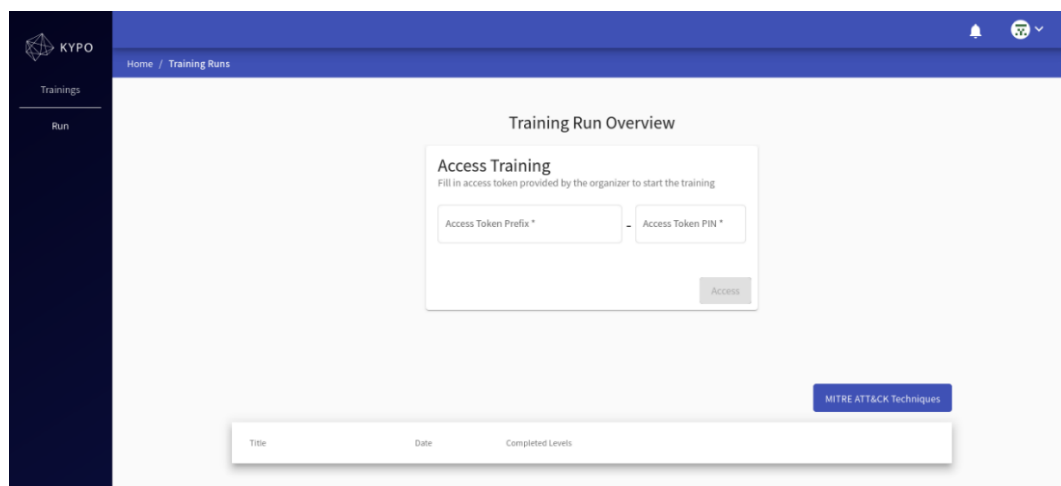
- **Training Definition:** This defines the scenario of the training. Definitions can be composed of levels or phases depending on their type. Linear definitions can contain Training levels, Assessment levels, and Info levels. Adaptive definitions can be composed of Training phases, Questionnaire phases, and Info phases.
- **Training Instance:** This specifies the time period in which players can access training. Each training instance also defines one of the following environments:
  - **Cloud Environment (default):** Cloud sandboxes are used during the training, and a pool of sandboxes must be assigned to the training instance.

- **Local Environment:** Local sandboxes are used during the training, and a sandbox definition can be assigned to the training instance so trainees can view the topology.
- **Training Run:** This is a single run of training for a particular trainee. Each run has an assigned sandbox from the pool or uses the trainee's own local sandbox.

## 6.2. User roles

The available functionalities of the KYPO Portal depend on the user's assigned role. There are three primary roles that grant different levels of access to various pages and functionalities within the KYPO Portal.

- **Trainees:** everyone who has access to the portal and wishes to participate in training can perform actions within the linear or adaptive Training run pages. In fact, when trainees log in to the portal, they are directly redirected to the Training Run Overview Page.



**Figure 6. KYPO panel for trainees**

- **Instructors:** they are responsible for creating and preparing training materials and sandboxes. They are given access to pages such as Linear Training definitions, Adaptive Training definitions, and Training instances, which enable them to manage and oversee the training process. Additionally, they can manage Sandboxes, Pools, and Resources pages.



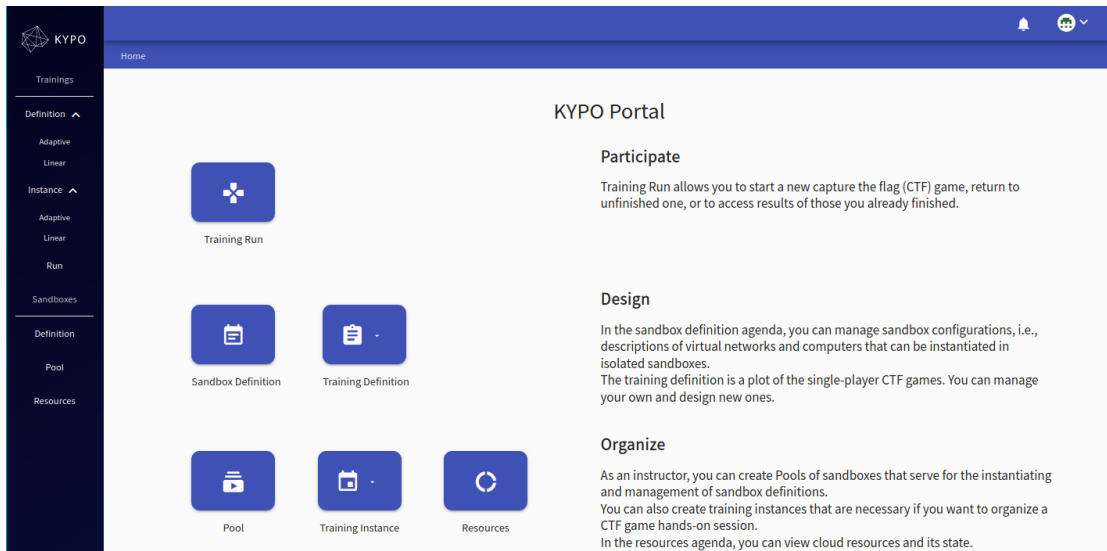


Figure 7. KYPO panel for instructors

- **Administrators:** they are responsible for managing the entire KYPO CRP instance. They possess access to all of the aforementioned pages within the KYPO Portal and have the authority to manage entities such as users, groups, and microservices in the Administration pages.

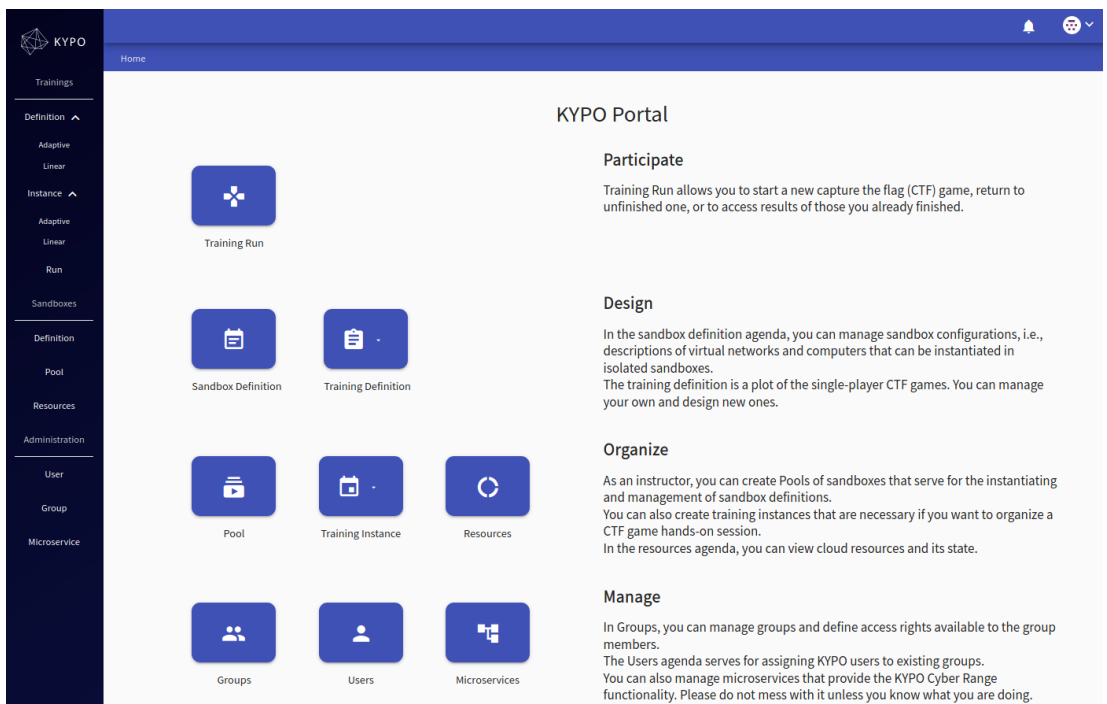


Figure 8. KYPO panel for administrators

### 6.3. KYPO CRP Architecture

The Kypo CRP is designed as a cloud platform to achieve flexibility, scalability, and cost-effectiveness. Users access the platform through the web-based GUI shown in the figures above.

#### Git repositories

Git repositories store sandbox definitions that can be employed in a cloud environment, local environment, or both. Microservices of the KYPO portal load them, which allows for the creation of sandbox instances in the cloud or displaying the topology of local sandboxes. Access to Git is set up during KYPO CRP deployment.

#### Users

Users assume different roles in KYPO CRP, each with varying scopes of work. They can create and manage training, administer other users, and design sandboxes using sandbox definitions. Users can access machines in cloud sandboxes via Spice client, Apache Guacamole remote desktop gateway, or directly using SSH. Moreover, users can access local sandboxes directly via specific Vagrant commands.

#### KYPO Portal

The KYPO Portal is a graphical user interface that allows users to interact with the KYPO CRP and access cloud sandboxes and other features. It serves as the mediator between users and microservices running in the background.

#### Cloud

The KYPO CRP environment is mostly based on the OpenStack cloud platform, which is responsible for controlling large pools of computing, storage, and networking resources managed through APIs or a dashboard. It is mainly deployed as infrastructure-as-a-service in public and private clouds where virtual servers and other resources are made available to users.

#### User Computer

Users can also build local sandboxes on their computers using VirtualBox and Vagrant as an alternative to cloud sandboxes. This approach saves cloud resources and allows the creation of trainings with a large number of users with only slight differences. However, the user must clone the sandbox definition repository and run a command to build a sandbox. Additionally, the user's computer must meet hardware and virtualization requirements. Given the objectives of the REWIRE project, this documentation will focus on the Cloud deployment only.

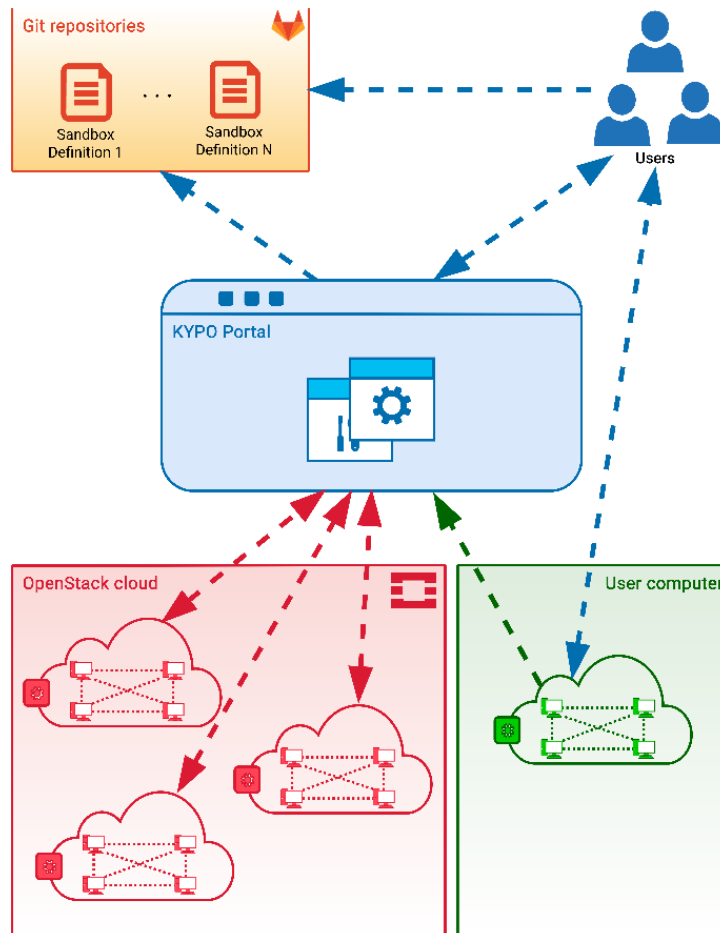


Figure 9. KYPO CRP Architecture Overview

## 6.4. Overview of the scenario implementation process

This section overviews the workflow for implementing, managing, and running scenarios in KYPO CRP. Each of the phases will be further described in the following sections.

### Sandbox Creation

1. Instructors create sandbox definitions in a specific format (section 6.5.1) and store them as Git repositories.
2. To create a record of the sandbox definition in the KYPO Portal, users can input the URI of the respective Git repository on the Sandbox Definition Overview page.
3. Once a sandbox definition has been created in the KYPO Portal, users can create a pool of a specified size by following a set of steps (section 6.5.2).
4. Cloud-based sandboxes can be allocated by clicking the allocation button of the corresponding pool. This process involves automatically two actions:
  - a. Downloading, parsing, and processing the respective sandbox definition from the Git repository.

- b. Creating sandboxes in the Cloud based on the sandbox definition.
5. Sandboxes can be used in two ways:
  - a. Instructors can access virtual machines inside the sandboxes using SSH or a graphical user interface to perform any necessary actions.
  - b. Additionally, sandboxes can be used as part of the training creation workflow.

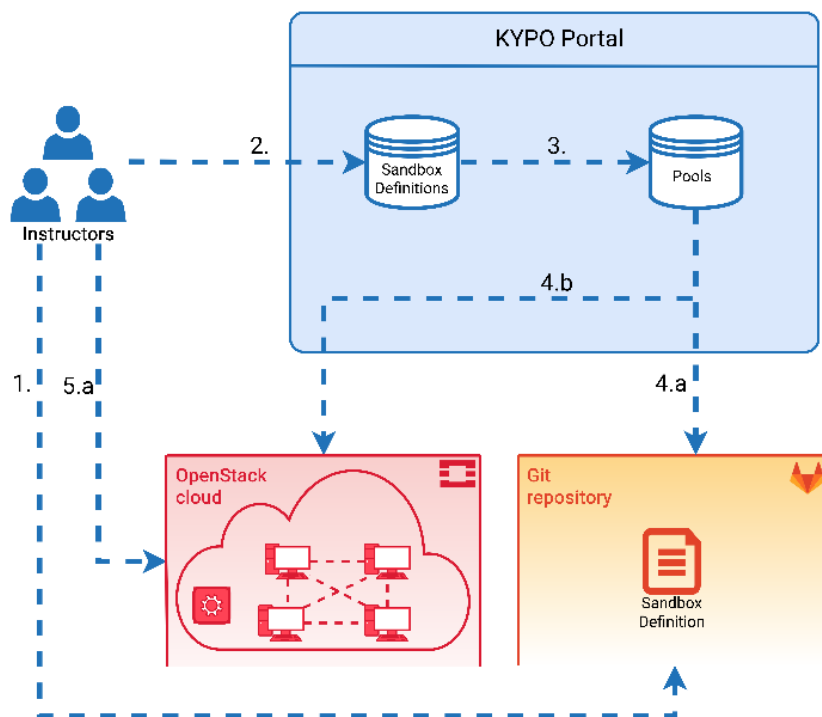


Figure 10. Sandbox creation

### Training Creation

1. A training definition can be created through either the Create Linear Training Definition or Create Adaptive Training Definition page independently of the sandbox definition.
2. To create a training instance, one needs to use the Create/Edit Training Instance page and select a training definition.
3. A group of unlocked sandboxes is assigned to a training instance through the Assign Pool panel when editing the training instance.
4. A partially generated access token is handed over by the instructor to grant trainees access to linear or adaptive training runs.
5. Trainees gain access to Training Runs by using the obtained access token. They can resume their accessed training runs as long as the training instance is still active.
6. Each training run is associated with a specific sandbox. Trainees can access VMs in this sandbox through Terminal Remote Access (SSH) or Web-based Access (Apache Guacamole, Spice).

7. During the training, an organizer can monitor the real-time progress of trainees and review their linear training run results.
8. Once the training instance is completed, the results become available and ready for further evaluation.
9. To free resources in the cloud, the assigned pool must be unassigned from the training instance and deleted using the delete button on the Pool Overview page. Similarly, the training instance can be deleted using the delete button on the Training Instances Overview page.

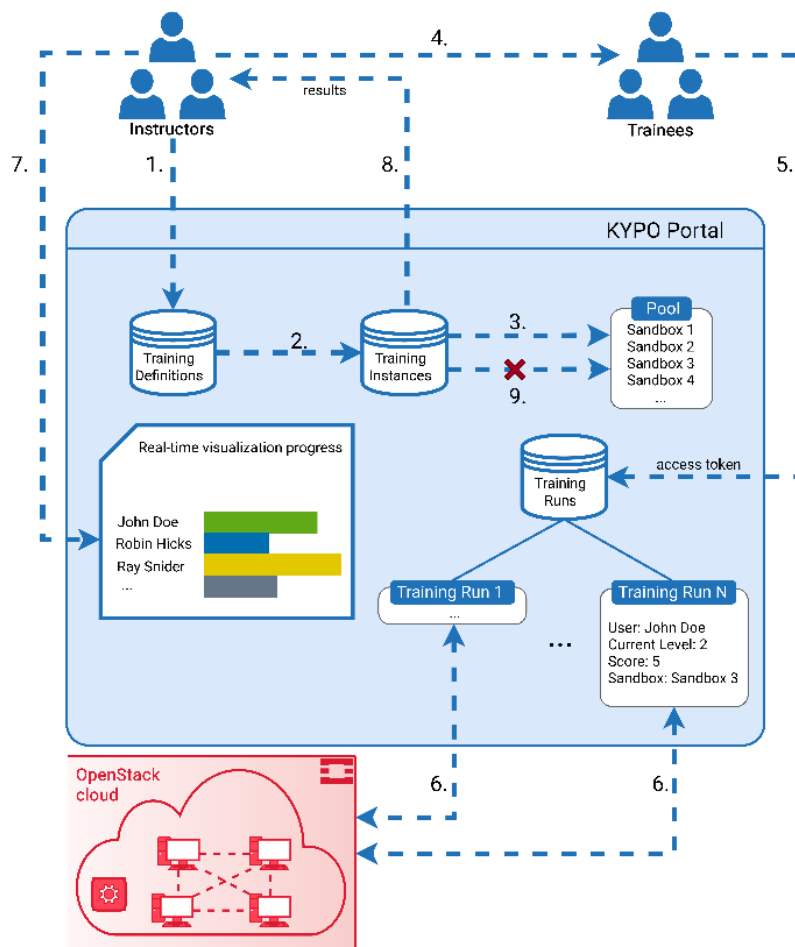


Figure 11. Training creation

## 6.5. Sandbox deployment detail

### 6.5.1. Sandbox definition

Sandbox definition is performed on the “Sandbox” section of the Kypo Portal. To access this section, click the respective button on the front page of the KYPO portal, or click the respective button in the global navigation in the “Sandboxes” section.

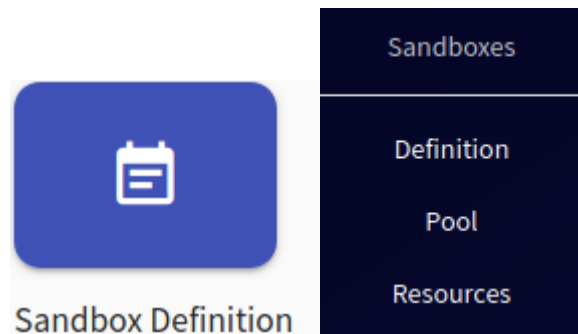


Figure 12. Sandbox definition

### Sandbox definition overview

On this page, instructors can find a comprehensive list of sandbox definitions that can be utilized to generate a pool of sandboxes for training instances. The list is presented as a table, where each row corresponds to a single sandbox definition.

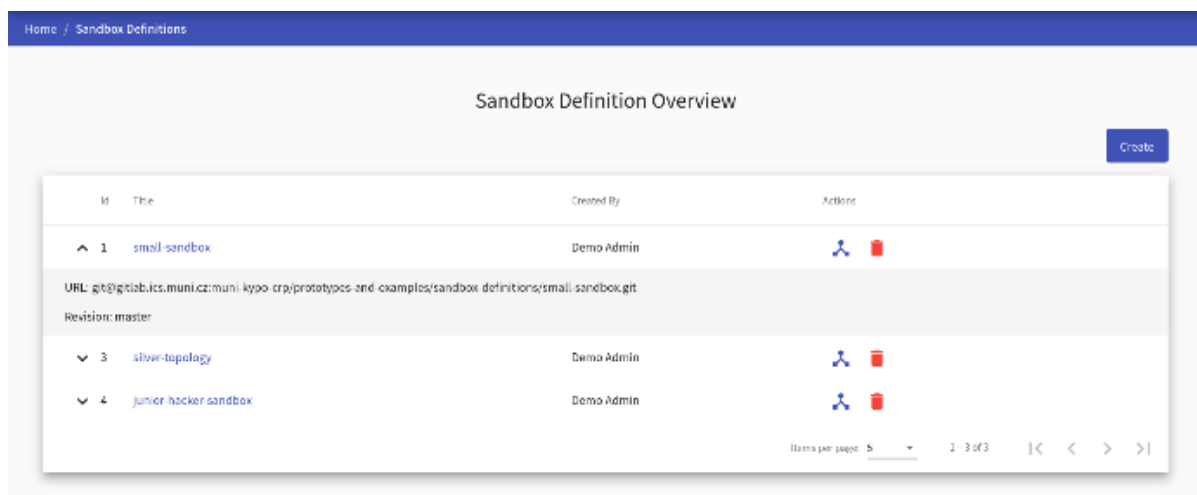
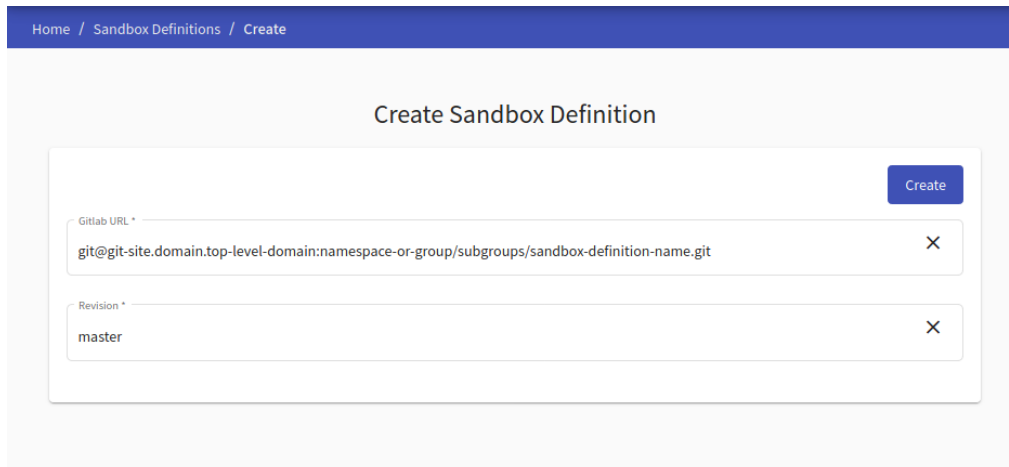


Figure 13. Sandbox definition overview

To access the sandbox definitions, the instructor can click on the title of each definition to be redirected to the corresponding Git repository or utilize the expand button to view detailed information about it. The final column in the table presents available actions that can be executed on the specific sandbox definition, such as displaying the topology or deleting it. To add a new sandbox definition, the instructor may click on the “create”- button which will redirect them to the “Create Sandbox Definition” page.



**Figure 14. Sandbox definition - Git repository**

Before creating a sandbox instance, it is necessary to define the topology and user configuration of the machines within the sandbox using the sandbox definition. The sandbox definition contains all the information required to generate a sandbox instance in the cloud. The sandbox definition is generated outside of the KYPO platform and stored as a Git repository.

The page within the KYPO portal features a single panel with two required fields:

- **Git URL:** This field requires the Git clone SSH URL of the sandbox definition.
- **Revision:** This required parameter is typically but not necessarily the name of a branch (e.g., master).

Upon ensuring that both of these fields have been accurately filled out, the instructor may create a new sandbox definition by clicking on the “create” button.

KYPO instructors utilize the Sandbox Definition as a directory structure stored in a Git repository to define the provisioning and topology of sandbox nodes.

The most basic form of the Sandbox Definition directory structure includes:

```
sandbox-definition/  
├─ topology.yml  
├─ variables.yml  
└─ provisioning/  
    └─ playbook.yml
```

- **topology.yml:** This is a definition of the topology that will be deployed in the cloud. Once deployed, it is referred to as a Topology Instance. The contents of this file are further described in the next section.
- **variables.yml (optional):** This configuration file contains all required and optional information necessary for generating Automatic Problem Generation (APG) Variables,

which can be used to produce variant answers for sandboxes within the pool (e.g., port numbers, usernames, file contents). It is required for APG trainings.

- **provisioning:** This directory structure of Sandbox Provisioning is designed for Topology Instance customization. It contains Ansible playbooks for customizing and configuring the nodes in the topology once the initial deployment has.

The following image shows the Sandbox Definition for a demo scenario in a Git repository. As can be seen, it is a good practice to include a readme file describing the scenario and the JSON file containing the training definition related to the sandbox.

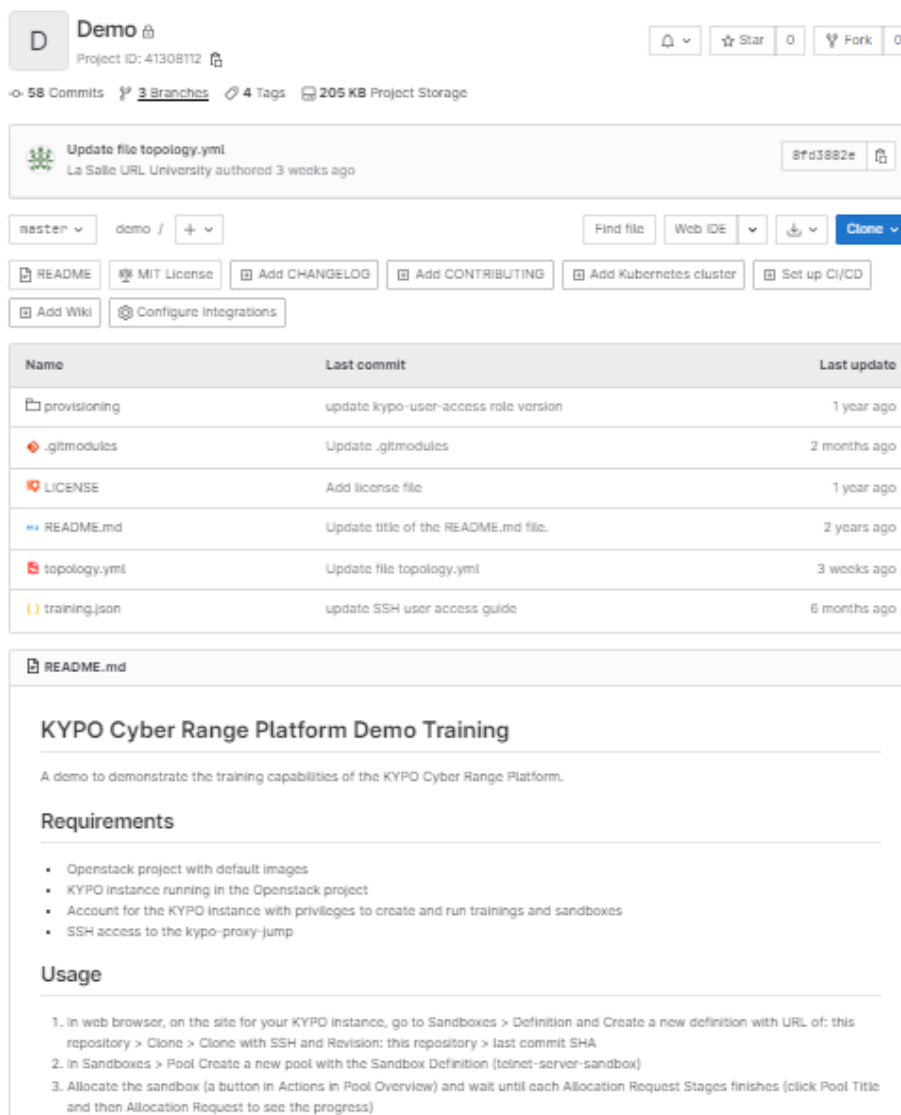


Figure 15. Sandbox Definition for a demo scenario in a Git repository



### 6.5.1.1. Topology definition

This section describes the structure and attributes of topology definition files. These are YAML (YAML Ain't Markup Language) files that KYPO will parse to deploy the topology instance with Terraform<sup>1</sup>.

#### Basic attributes

- **name:** short name of the topology (required)
- **hosts:** a list of host devices. All attributes of the virtual machines that will be created are defined here. Every host must have a **name**, which must be unique within the definition, and a **base\_box**. (required)
  - **name:** unique name of the device (required)
  - **base\_box:** specifies the image of the node boot disk, default user with sudo permissions for management, and a protocol needed to communicate with the machine.
    - **image:** an OS image that will be installed on the machine (required). The base image must exist (uploaded) in the OpenStack platform where it will be running.
    - **mgmt\_user:** OS image management user for cloud environment (optional)
    - **mgmt\_protocol:** communication protocol **ssh** or **winrm**. Linux images use **ssh** and Windows images use **winrm** (default: **ssh**) (optional)
  - **flavor:** a quick definition of memory and CPUs (required). Flavors offer a convenient method for selecting the hardware specifications of a virtual machine (VCPU, RAM, Disk size). Furthermore, the attributes of memory and cpus can be defined and overwritten independently as extra attributes. CSIRT-MU/KYPO OpenStack projects provide a predefined list of available flavors for OpenStack<sup>2</sup>. However, administrators can define their own flavors in OpenStack.
  - **extra:** special attributes (optional)
    - **cpus:** number of CPU units - overwrites the value specified in **flavor** (optional)
    - **memory:** required memory size in MB - overwrites the value specified in **flavor** (optional)
  - **hidden:** whether the host should be hidden in a topology visualization (default: False)
  - **volumes:** a list of volumes in the format "size: X" can be specified to provision volumes on the host. X represents the size of the volume in GB. The first volume in the list is used as the system drive. The person creating the definition

<sup>1</sup> <https://www.terraform.io/>

<sup>2</sup> <https://docs.crp.kypo.muni.cz/user-guide-advanced/sandboxes/topology-definition/#flavor>

must determine how much space the image requires to run on this system drive (optional).

- **routers**: a list of routers. (required, can not be empty for KYPO). It is defined by the same parameters as **hosts**, except that the **hidden** parameter is not available, and **ssh** is the only supported option for **mgmt\_protocol**.
- **wan**: A special network for routers. All routers are assigned to this network and communicate with each other and the Internet through it. (optional)
  - **name**: name of the network (default: wan)(optional)
  - **cidr**: IP address of the network in CIDR notation (default: 100.100.100.0/24)(optional)
- **networks**: list of networks. A network is used to connect the router to the end host. (required)
  - **name**: unique name of the network (required)
  - **cidr**: IP address of the network in CIDR notation (required)
  - **accessible\_by\_user**: specifies which networks will be accessible by the user (default: True), applies to all hosts of a network (optional)
- **net\_mappings**: mappings of host machines to a network. This list defines the IP addresses of hosts in the networks (required - can be empty)
  - **host**: name of an existing host (required)
  - **network**: name of an existing network (required)
  - **ip**: IP address of the host in the network (required)
- **router\_mappings**: This parameter is similar to **net\_mappings** and is used to define the addresses of routers inside networks (required - can be empty)
  - **router**: name of an existing router (required)
  - **network**: name of an existing network (required)
  - **ip**: IP address of the router in the network (required)
- **groups**: This parameter is a list of additional groups for ansible. The groups **all**, **routers**, **hosts**, **ssh**, and **winrm** are already available without additional definition (required - can be empty)
  - **name**: specifies the name of the group.
  - **nodes**: list of device names in the group.

### Example

The following example of a topology definition file sets a sandbox with the name “small-sandbox” containing the following:

- Two hosts. The host “server” will not be visible in the topology.
- Two routers.
- The wan network with a custom name.
- Two networks. Only one is user-accessible and therefore connected to the user access node.
- One group, which contains two nodes.

```
name: small-sandbox
hosts:
  - name: server
    base_box:
      image: debian-9-x86_64
      mgmt_user: debian
      flavor: csirtmu.tiny1x2
      hidden: True
    volumes:
      - size: 16
      - size: 2
      - size: 9

  - name: home
    base_box:
      image: windows-10-0.2.0
      mgmt_user: windows
      mgmt_protocol: wirm
      flavor: csirtmu.tiny1x2

routers:
  - name: server-router
    base_box:
      image: debian-9-x86_64
      mgmt_user: debian
      flavor: csirtmu.tiny1x2

  - name: home-router
    base_box:
      image: debian-9-x86_64
      mgmt_user: debian
      flavor: csirtmu.tiny1x2

wan:
  name: internet-connection
  cidr: 100.100.100.0/24
```

```
networks:
  - name: server-switch
    cidr: 10.10.20.0/24
    accessible_by_user: False

  - name: home-switch
    cidr: 10.10.30.0/24

net_mappings:
  - host: server
    network: server-switch
    ip: 10.10.20.5

  - host: home
    network: home-switch
    ip: 10.10.30.5

router_mappings:
  - router: server-router
    network: server-switch
    ip: 10.10.20.1

  - router: home-router
    network: home-switch
    ip: 10.10.30.1

groups:
  - name: custom-group
    nodes:
      - home
      - home-router
```

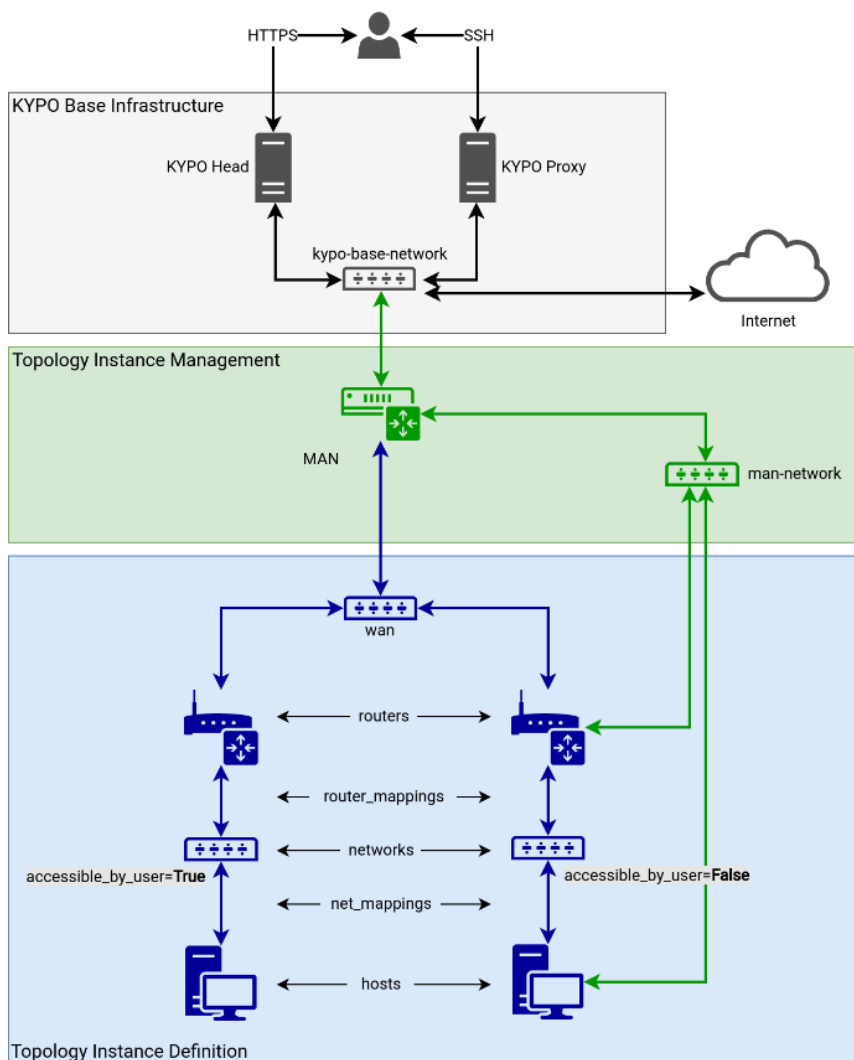


Figure 16. Architecture of example topology

### 6.5.1.2. Sandbox provisioning

The Sandbox Provisioning leverages the customization of the Topology Instance. It enables the creation of users, installation of packages, and environment setup, among others.

The **provisioning** directory contains the same content as any other Ansible project<sup>3</sup>. Specifically, it consists of the following required and optional files and directories:

- **playbook.yml:** the Ansible playbook utilized for provisioning the sandbox. Refer to the Ansible documentation on playbooks for more information<sup>4</sup>.

<sup>3</sup> <https://docs.ansible.com/ansible/latest/index.html>

<sup>4</sup> [https://docs.ansible.com/ansible/latest/playbook\\_guide/playbooks.html](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks.html)

PUBLIC

- **pre-playbook.yml** (optional): the Ansible playbook used to install the necessary packages for **playbook.yml**.
- **requirements.yml** (optional): the Ansible Galaxy requirements file containing Ansible role dependencies<sup>5</sup>.
- **roles** (optional): the directory that contains Ansible roles<sup>6</sup>.
- **group\_vars** (optional): the directory that stores group variables<sup>7</sup>.
- **host\_vars** (optional): the directory that contains host variables.

KYPO CRP requires a playbook defining the Sandbox Provisioning stage. However, if provisioning is not required, a minimal Ansible playbook consisting of only the following line can be used:

```
hosts: all
```

### 6.5.1.3. Sandbox access

There are two primary categories for Sandbox access:

- **Terminal Remote Access:** This refers to accessing the remote sandbox node from the local command-line interface. It requires extra configuration (retrieving and proper configuration of the SSH configuration file, private, and public keys) and has two privilege types:
  - **Management access:** Access is granted to those managing sandboxes of the KYPO CRP (i.e., instructor and administrator roles).
  - **User access:** Access is provided to everyone else (i.e., trainee role). Machines with the **user\_accessible** parameter set to false can not be accessed.
- **Web-based Access:** This involves accessing the remote sandbox node through the KYPO portal using a web browser. It is available to everyone and is simpler since no additional configuration is necessary. Instructors and administrators can display the topology on the Pool Detail page, and for trainees, the topology is always displayed during a training run at training levels. Two clients that enable this connection are supported:
  - **Apache Guacamole:** An HTML5 web application that supports graphical access to remote hosts directly in the browser. It is a clientless remote desktop gateway that supports standard protocols like VNC (Linux), RDP (Windows), and SSH (Linux).
  - **Spice:** OpenStack provides an alternative to Apache Guacamole for accessing guest virtual machines remotely. This is achieved through the use of the Simple Protocol for Independent Computing Environments (SPICE) protocol. Whether a command-line interface or graphical user interface is provided depends on

<sup>5</sup> [https://docs.ansible.com/ansible/latest/galaxy/user\\_guide.html#installing-multiple-roles-from-a-file](https://docs.ansible.com/ansible/latest/galaxy/user_guide.html#installing-multiple-roles-from-a-file)

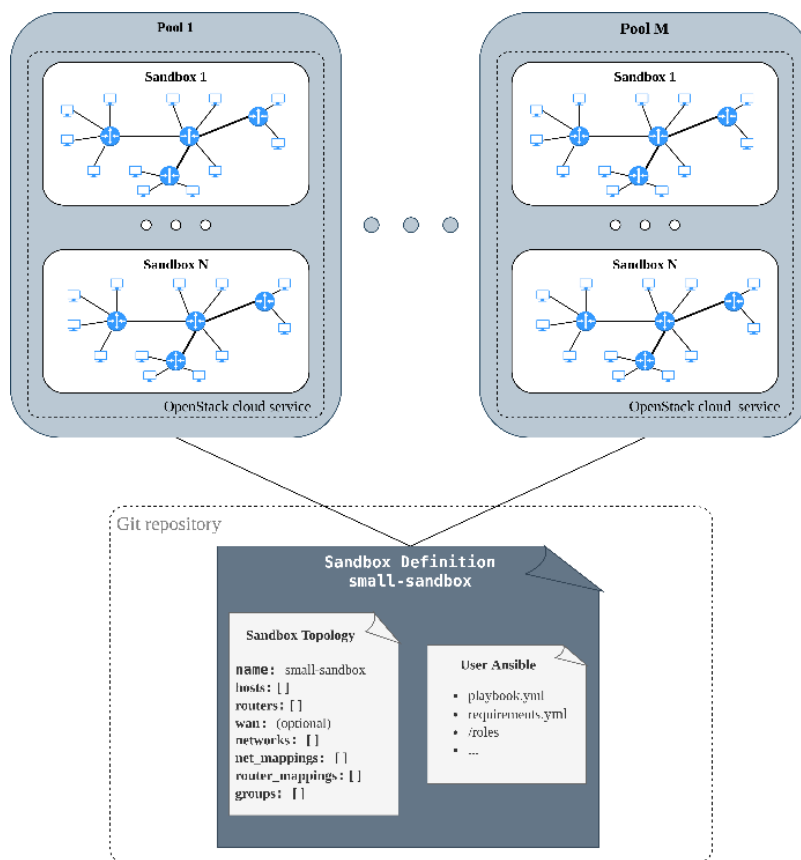
<sup>6</sup> [https://docs.ansible.com/ansible/latest/playbook\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html)

<sup>7</sup> [https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#organizing-host-and-group-variables](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#organizing-host-and-group-variables)

whether a display manager is installed. Upon connecting to the Spice client, a new browser tab will display the Spice console.

### 6.5.2. Pools

To prepare for the deployment of sandboxes, it is necessary to first create pools in the system. Pools are collections of sandboxes generated based on the same sandbox definition, which must be specified before creating the pool. Once the pool is created, sandboxes can be allocated from it.



**Figure 17. Link between Sandbox definition and Pools created**

To access this section, click the respective button on the front page of the KYPO portal, or click the respective button in the global navigation in the “Sandboxes” section.

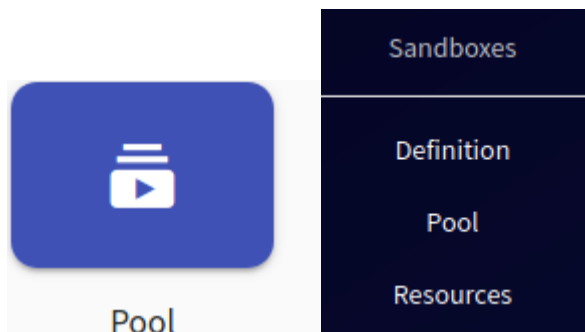


Figure 18. Pool definition

The "Pool Overview" page displays a table containing all the accessible sandbox pools in the KYPO portal. Each row of the table represents one pool. Instructors can view more detailed information about a pool by clicking on its title. The last column of the table provides various actions that can be performed on a pool, such as delete, allocate all, allocate one, clear, get SSH config, lock, and unlock. To create a new pool, instructors need to click on the "create" button, which redirects them to the "Create Pool" page.

| Title   | Created By | Sandbox Definition | State    | Size | Instances Util. | Cpu Util. | Ram Util. | Actions                                     |
|---------|------------|--------------------|----------|------|-----------------|-----------|-----------|---|
| Pool 10 | Demo Admin | small-sandbox      | locked   | 6/6  | 30.0%           | 36.0%     | 23.7%     | [trash] [refresh] +1 [list] [lock] [unlock] |
| Pool 11 | Demo Admin | small-sandbox      | locked   | 2/2  | 10.0%           | 12.0%     | 7.9%      | [trash] [refresh] +1 [list] [lock] [unlock] |
| Pool 12 | Demo Admin | silver-topology    | unlocked | 0/1  | 0.0%            | 0.0%      | 0.0%      | [trash] [refresh] +1 [list] [lock] [unlock] |

Figure 19. Summary of available pools

The "Create Pool" page contains a form that needs to be filled out before creating a new pool. The "Sandbox Pool Size" field specifies the maximum number of sandboxes that can be created within the pool. Instructors also need to select one of the available sandbox definitions that define the topology of sandboxes and user configuration of virtual machines created in a sandbox. After filling out all the required fields, instructors can confirm the creation of a new pool by clicking on the "create" button.



**Figure 20. Creating a pool**

The "Pool Detail" page displays information about the sandbox instances allocated in a given pool. Instructors can access this page by clicking on the title of a pool in the "Pool Overview" page. The page contains a table with all the allocated sandboxes. The last column of the table provides actions that can be executed on a sandbox, such as delete, display topology, get SSH config, lock, and unlock.

| Name       | Lock     | Created           | Created by | State              | Stages | Actions  |
|------------|----------|-------------------|------------|--------------------|--------|----------|
| Sandbox 16 | locked   | 19 May 2022 12:13 | Demo Admin | build finished     | ✓✓✓    | 🗑️ 🌐 🔑 🔒 |
| Sandbox 17 | unlocked | 25 May 2022 12:34 | Demo Admin | allocation running | ✓🔄🕒    | 🗑️ 🌐 🔑 🔒 |

**Figure 21. Pool details**

The sandbox allocation consists of three stages: allocation of a sandbox in the cloud (performed by Terraform), sandbox networking (performed by Ansible), and sandbox provisioning (performed by Ansible). Each stage can be in one of the following states: in queue, running, finished, or failed. If one of the stages fails, instructors can restart the failed stage by clicking on the retry icon. Keep in mind that only the second and third stages can be restarted. Detailed information about a stage can be viewed by clicking on the stage name, and if a stage fails, the error message should be available.

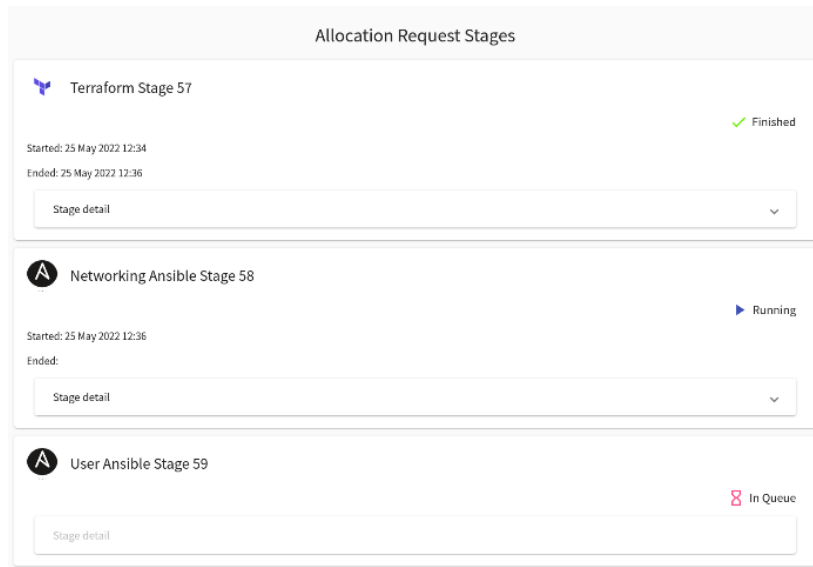


Figure 22. Sandbox allocation

## 6.6. Training deployment detail

### 6.6.1. Training definition

Training definition is performed on the “Training” section of the Kypo Portal. To access this section, click the respective button on the front page of the KYPO portal, or click the respective button in the global navigation in the “Trainings” section.

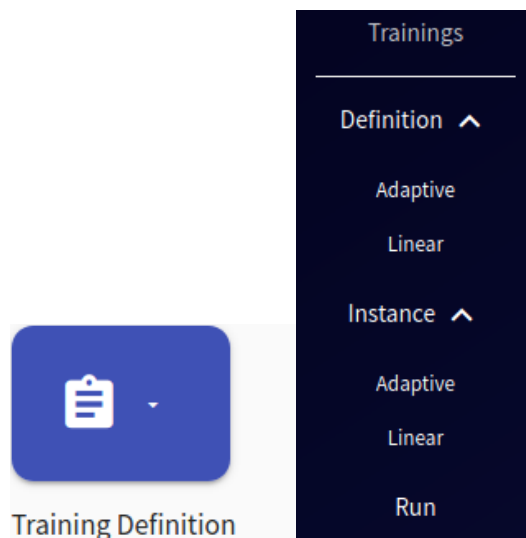


Figure 23. Training definition

The Training Definition section creates and manages training definitions, which refer to the material covered in training instances. Two options are available for selection: the Linear Training Definition and the Adaptive Training Definition. Both Linear and Adaptive training definitions provide details such as the title, instructor notes, and learning objectives. Additionally, both types of training definitions consist of multiple levels or phases. Once created, a training definition can be exported from or imported to the KYPO CRP as a JSON file. A recommended best practice is to save the Training Definition in the Git repository in the same directory or alongside the Sandbox Definition repository, which is designed specifically for that particular Training Definition.

Automatic Problem Generation (APG) is a technique that can be used in conjunction with Linear training definitions. It is employed to establish multiple problem instances. In the context of KYPO CRP, this is accomplished through different responses for each Training Run, which can help mitigate the risk of answers being copied or leaked. The APG training definition requires a specific Sandbox Definition, including a **variables.yml** file. This file outlines the variables that will be automatically generated for each instance of the sandbox. The resulting values can then be employed during provisioning to establish secret answers within the sandbox environment, such as a filename, port, or username.

#### 6.6.1.1. Linear training definition

The “Linear Training Definition Overview” page displays all available definitions for an instructor or administrator, with the ability to create or upload a new definition using the corresponding buttons located in the top right corner. Each row in the table represents a single training definition, and the last column contains the available actions that can be performed on a definition. Clicking on a definition's name will redirect to the detail page.

Home / Linear Training Definitions

### Linear Training Definition Overview

Create Upload

Filter by title

| Title           | State      | Estimated Duration | Last Edit ↓       | Last Edit By | Actions |
|-----------------|------------|--------------------|-------------------|--------------|---------|
| House of Cards  | Unreleased | 30                 | 15 Jun 2022 11:15 | "Demo Admin" |         |
| CTF - January   | Unreleased | 60                 | 15 Jun 2022 11:14 | "Demo Admin" |         |
| Winter Training | Unreleased | 45                 | 15 Jun 2022 11:14 | "Demo Admin" |         |

Items per page: 20 1 - 3 of 3

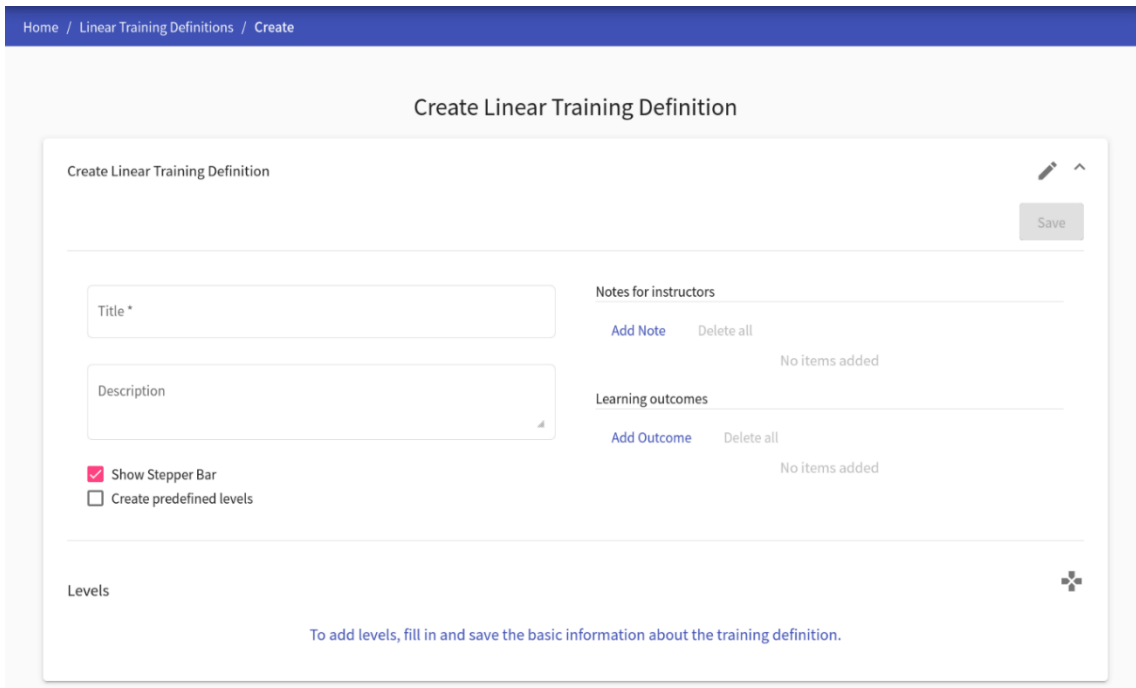
MITRE ATT&CK Techniques

**Figure 24. Linear training definition**

The available actions that can be executed on a training definition are Edit, Delete, Clone, Download, Preview, Release, Unrelease, and Add a New Definition. There are three methods for creating a new training definition: creating from scratch, uploading from a JSON file, or cloning an existing definition.

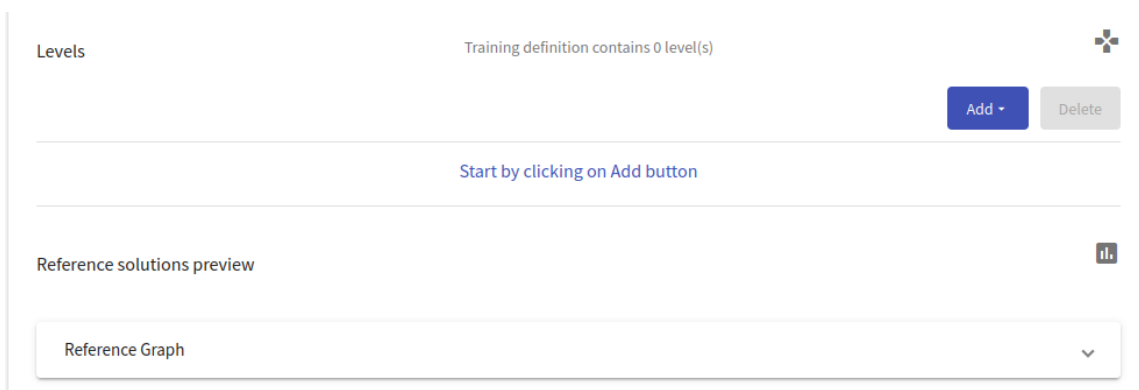
### Create a New Definition

To create a new definition from scratch, click the Create button to open the training definition editor page. In the first panel of the editor, the instructor can edit fields that describe the new definition and save the definition by clicking the Save button.



**Figure 25. Creating a Linear training definition 1**

The Levels Panel is part of the Create Linear Training Definition Panel, where the instructor can add, delete, and edit training levels of the training definition. The Reference Graph Panel at the bottom displays a preview of the graph created according to the reference solutions defined in the training levels.



**Figure 26. Creating a Linear training definition 2**

To add a new level, the instructor can click the Add button, which will display a menu where they can select the type of the new level. Each change made inside a level must be saved

using the Save button at the top of the Create Linear Training Definition Panel. There are four types of levels available:

1. **Info Level:** Contains information for the trainee (welcome message or important information about the following levels).
2. **Training Level:** The user has to solve a predefined assignment in the level. By solving the assignment, the trainee acquires a secret answer, and after submitting the answer, they can continue to the next level of the training.
3. **Assessment Level:** It can be either a test or a questionnaire, and it serves to test users' knowledge or get feedback from users. The assessment can contain one of the following types of questions:
  - **Multiple choice question (MCQ):** Trainees are asked to select one or multiple answers from the choices offered as a list.
  - **Extended matching item (EMI):** Trainees are asked to pair items from rows and columns that are semantically related.
  - **Freeform question (FFQ):** Trainees are asked to type the answer to the submit field.
4. **Access Level:** Contains information on how to access the sandbox machines.

### Training Level

At the training level, a trainee can access a virtual network inside the sandbox to find a solution to the assignment. The instructor can fill out the form to specify the details of the new level. The training definition is considered APG if **Variant Answers** is checked, and **Correct Answer - Variable Name** is filled. The reference solution defines the sample solution of the training level, i.e., the commands that need to be executed to find the secret answer. It is also used for post-training feedback visualizations. A preview of the reference graph is displayed in the **Reference Graph** panel after saving all training levels and is updated whenever a reference solution of a training level is modified.

The **Hints** panel, **MITRE ATT&CK Techniques** panel, and **Expected Commands** panel can be used to add, delete, and edit hints, MITRE ATT&CK techniques, and commands associated with a given training level.

Levels
Training definition contains 1 level(s)
✕

Add
Delete

1 1. Title o...

Title \*

Title of training level ✕

Estimated Duration in Minutes

1

Number in minutes greater than 1.

Minimal Possible Solve Time in Minutes

Number in minutes greater 0.

Points for the Level \*

100

Number in range from 0 to 100

Incorrect Answer Limit \*

100

1 to 100. If the limit will be overreached, solution will be displayed automatically.

Variant Answers (requires a specific sandbox definition)

Correct Answer - Static

Secret answer ✕

\*Max 50 characters\* 13/50

Solution Penalized (trainee will get 0 points for the level)

Is any command required to complete the level?

Reference Solution

⏏

\*Preview of the reference graph can be seen in the panel below level hints

**Content**

Edit
Preview

⏏ B I ↶ ↷ ↻ ↺ ↻ ☑ ☰

The test entry should be here

[Markdown](#) supported.

**Solution**

💡 Use Placeholder Variable

Display an actual value of the answer in the solution content by using the variable `$(ANSWER)`, that will be resolved and replaced in displayed solution during the training run.

Edit
Preview

⏏ B I ↶ ↷ ↻ ↺ ↻ ☑ ☰

Solution of the training should be here

[Markdown](#) supported.

Hints ▼

This level has no hints

MITRE ATT&CK Techniques ▼

This level has no techniques

Expected Commands ▼

This level has no expected commands

Reference solutions preview ☰

Reference Graph ▼

Figure 27. Linear training – Training level

## Assessment Level

At the assessment level, the trainees answer a list of questions, and the content of this level can be edited in the assessment level editing form.

**Figure 28. Linear training – Assessment level 1**

An option is available for the instructor to create either a test or a questionnaire, which influences how questions are created. The **Questions** panel can be used to create, delete, and edit questions associated with the given assessment level. The menu can be accessed by clicking on the “Add” button to insert a new question, where the instructor can choose the type of question (MCQ, EMI, or FFQ). It should be noted that each question type has its own unique editing form.



Question 1: New Free Form Question  Required

Title

Edit Preview 
¶ **B** *I* U **”** **<** **>** [↪](#) [☰](#) [☰](#)  [☰](#)

New Free Form Question

[Markdown](#) supported.

Delete

**Figure 29. Linear training – Assessment level 2**

Question 2: New Multiple Choice Question  Required

Title

Edit Preview 
¶ **B** *I* U **”** **<** **>** [↪](#) [☰](#) [☰](#)  [☰](#)

New Multiple Choice Question

[Markdown](#) supported.

Add choice

Option 1\* -

Choice 1

Option 2\* -

Choice 2

Delete

Question 3: New Extended Matching Items  Required

Title

Edit Preview 
¶ **B** *I* U **”** **<** **>** [↪](#) [☰](#) [☰](#)  [☰](#)

New Extended Matching Items

[Markdown](#) supported.

|               |                       |                       |   |  |  |
|---------------|-----------------------|-----------------------|---|--|--|
|               | -                     | -                     |   |  |  |
|               | Option 1 x            | Option 2 x            | + |  |  |
| Statement 1 x | <input type="radio"/> | <input type="radio"/> | - |  |  |
| Statement 2 x | <input type="radio"/> | <input type="radio"/> | - |  |  |
|               | +                     |                       |   |  |  |

Delete

**Figure 30. Linear training – Assessment level 3**

**Info Level**

At the info level, trainees read content written by the instructor. It is recommended to always include an info level at the beginning of each training with a description of it and its associated learning objectives.

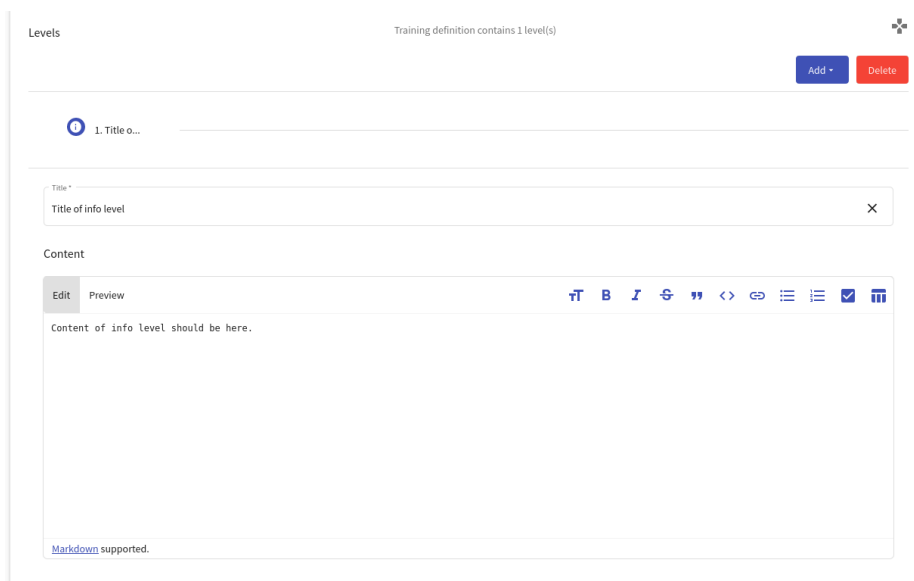


Figure 31. Linear training – Info level

### Access Level

In the access level, trainees are provided with necessary information on accessing machines, and must submit a passkey to proceed to the next level. Passkey can be provided to trainees by an instructor or can be mentioned in the content itself.

Levels Training definition contains 1 level(s)

[Add](#) [Delete](#)

1. Get Acc...

Title \*  
Get Access

Passkey  
start-training

Cloud Content

Markdown supported.

Local Content

Use Placeholder Variables

The following variables will be resolved and replaced by actual values as soon as the trainee will get into the access level during the training run:

- USER\_ID
- ACCESS\_TOKEN
- CENTRAL\_SYSLOG\_IP
- BEARER\_TOKEN

Variables have to be enclosed in special characters `{}`, e.g., `{USER_ID}`.

Markdown supported.

Figure 32. Linear training – Access level

## Authors Panel

The Authors Panel is the second of the editor. The instructor can add and remove authors from the definition.

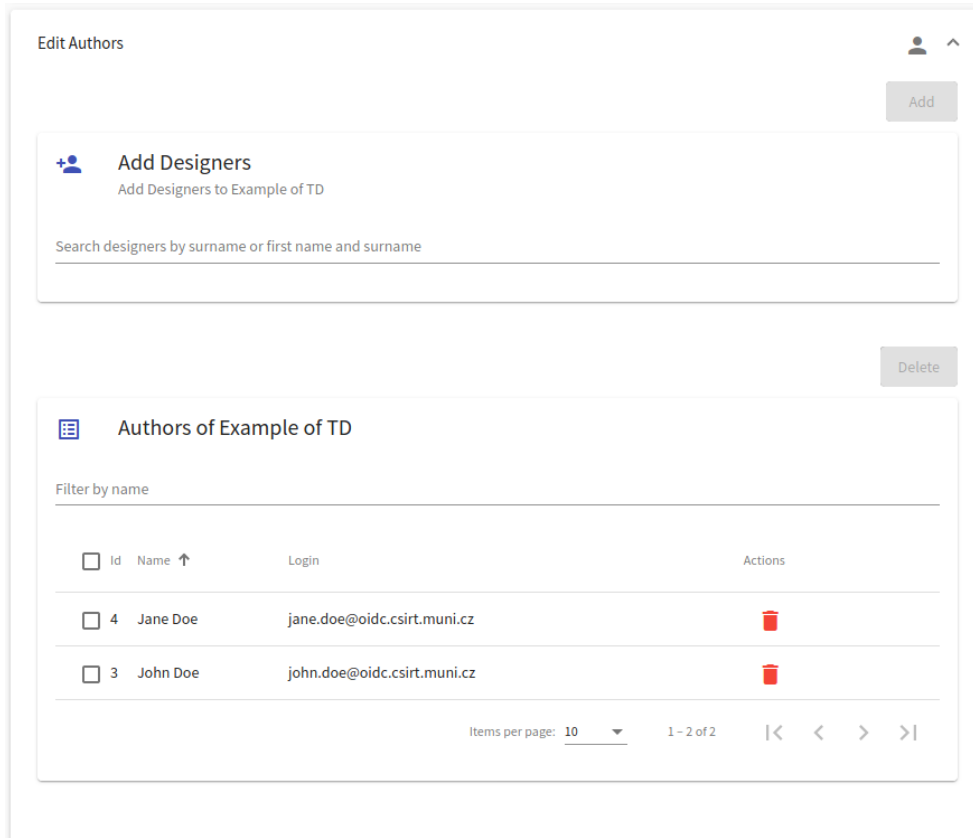


Figure 33. Linear training – Authors Panel

## Upload a Definition From JSON File

Uploading a training definition in JSON format can be done by clicking the “Upload” button in the Linear Training Definition Overview. This use case is useful when the instructor wants to re-use the training definition stored in the past or coming from another party.

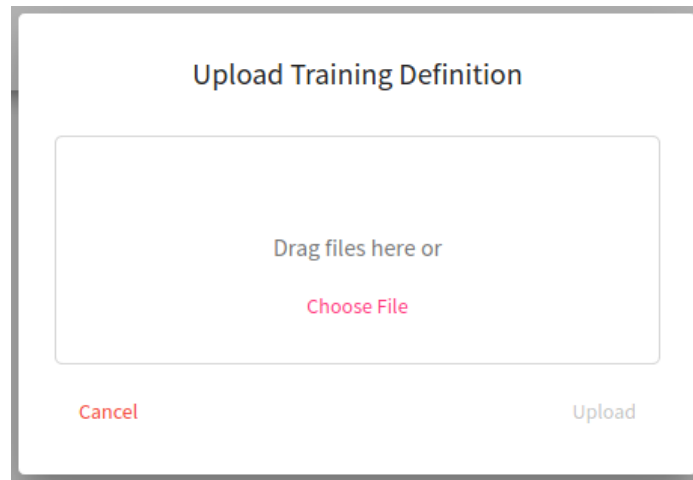


Figure 34. Upload a Training definition

### 6.6.1.2. Adaptive training definition

The figure below illustrates the adaptive format of training, which comprises a graph structure consisting of a pre-training questionnaire (A, Q), a series of phases, and a list of tasks. The pre-training questionnaire A is administered before the training to assess the trainees' knowledge, and their responses to the tasks, commands entered in the command line in the sandbox, time spent in a given phase, and events of displaying the solution are audited to the internal database. The pre-training questions A are assigned to specific phases to link them with the appropriate phases.

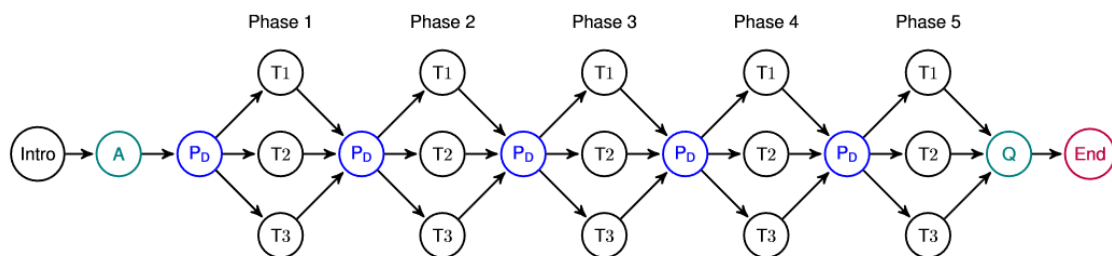


Figure 35. Adaptive learning overview

The first training phase after the pre-training question is the **access phase**, which provides guidance on how to connect to the sandbox. It is important for all trainees to be connected to the virtual environment before the training starts. The **PD** component represents the computational node or tutor model of adaptive training that calculates the trainees' theoretical and practical knowledge and skills.

The post-training questionnaire component **Q** collects feedback from the trainees after the training. Similar to the four types of linear training levels, adaptive trainings consist of five types of phases:

1. **Info Phase:** Provides trainees with information about the training or important information about the following phases.
2. **Adaptive Questionnaire Phase:** Collects data about the trainees' knowledge and skills through a pre-training questionnaire that is grouped into linked relations to specific training phases. The questionnaire may contain multiple-choice questions (MCQ), free-form questions (FFQ), or rating-form questions (RFQ). The essential ratio of knowledge is set for each training phase to determine whether the trainees' theoretical knowledge or self-reported phase of skills is sufficient or not.
3. **General Questionnaire Phase:** Similar to the adaptive questionnaire phase, but the questions are unrelated to the training phases. The questionnaire is used to gather feedback from trainees and contains the same types of questions as the adaptive one, but they cannot have a predefined correct answer.
4. **Training Phase:** Consists of several task variants of various difficulties, typically arranged from the most difficult to the easiest, but all on the same topic. The task with the most suitable difficulty is based on the trainee's actions in the previous phases and their answers from the pre-training questionnaire. The **Decision Matrix** is defined in each phase to compute the most appropriate task for the trainee, allowing the instructor to set up the weights of the several aspects considered during the computation. The aspects considered are the results from the pre-training questionnaire, the commands used by the trainee to complete the phase, the amount of time required for completion, the hints and solutions displayed, and the number of submitted incorrect answers.
5. **Access Phase:** Contains information on how to access the sandbox machines.

The overview page for Adaptive Training is analogous to the Linear Training overview page. A Training can also be defined by creating a new one, importing a JSON file, or cloning an existing training. The Level Panel of lineal training is renamed to Phase Panel in adaptive training, as seen in the image below.

Home / Adaptive Training Definitions / Create

### Create Adaptive Training Definition

Create Adaptive Training Definition Save

Title \*

Description

Show Stepper Bar  
 Create predefined phases

Notes for instructors  
[Add Note](#) [Delete all](#)  
No items added

Learning outcomes  
[Add Outcome](#) [Delete all](#)  
No items added

Phases +

To add phases, fill in and save the basic information about the training definition.

**Figure 36. Adaptive Training Definition**

## Training Phase

The Training Phase of the adaptive training is similar to the Training Level of the linear training. During the Training Phase of the adaptive training, trainees are assigned a task variant based on their performance. The instructor can customize the details of the phase using a form. The **Decision Matrix**, which contains weights for five performance metrics, is used to establish relationships between phases and metrics. For example, if the third phase deepens the topic from the first phase, the weights in the third matrix should be set so that the selected weights for the metrics from the first phase are non-zero. The instructor must manually set these weights since each training program is unique. Based on the **Decision Matrix** and trainees' performance, the adaptive training program calculates and assigns a suitable task to the trainee.

**Decision Matrix**

| Questionnaire Answered | Completed in Time | Keyword Used | Solution Displayed | Submitted Answers | Related Phase              |
|------------------------|-------------------|--------------|--------------------|-------------------|----------------------------|
| 0                      | 0                 | 0            | 0                  | 0                 | 1. Title of training phase |
| 0                      | 0                 | 0            | 0                  | 0                 | 2. Title of training phase |
| 0                      | 0                 | 0            | 0                  | 0                 | 3. Title of training phase |

Title \*

Title of training phase ✕

---

Allowed Wrong Answer Limit (Default 10) \*

10

---

Allowed Commands Limit (Default 10) \*

10

---

Estimated Duration (Default 10) \*

10

Tasks ▼

---

Related Questions ▼

---

MITRE ATT&CK Techniques ▼  
This level has no techniques

---

Expected Commands ▼  
This level has no expected commands

**Figure 37. Adaptive Training – Training phase 1**

The Training Phase editing form also includes a Tasks panel, which allows the instructor to create, edit, and delete task variants associated with a specific training phase.

Tasks ^

Add
Copy
Delete

---

There are no Tasks

**Figure 38. Adaptive Training – Training phase 2**

The instructor can add a new task using the add button and edit it using a form. The copy button can be used to create a new task by copying the content of the selected task. The order of tasks can be changed using the drag-and-drop mechanism.



The screenshot displays two main editing areas: 'Content' and 'Solution'. Each area has an 'Edit' and 'Preview' tab and a rich text editor toolbar. The 'Content' editor shows 'Task content ...' and the 'Solution' editor shows 'Task solution ...'. To the right, a settings panel includes:
 

- 'Title \*' field with 'Title of a new task' and a close button.
- 'Incorrect Answer Limit \*' field with '1' and a range indicator '1 to 100. If the limit will be overreached, solution will be displayed automatically.'
- 'Answer \*' field with 'Secret Answer' and a close button.
- 'Max 50 characters' and '13/50' character count.
- 'Modify sandbox:' toggle switch.

Figure 39. Adaptive Training – Training phase 3

### Adaptive Questionnaire Phase

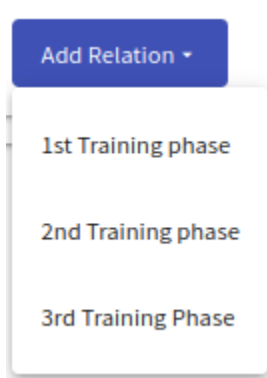
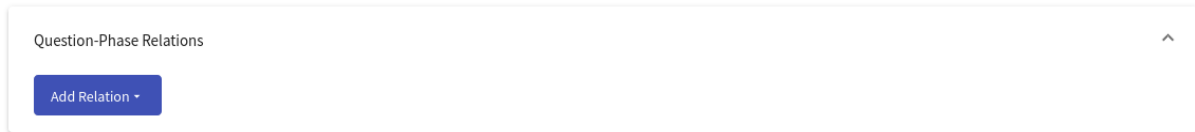
In this phase, the trainees answer a list of questions (FFQ, MCQ, or RFQ). These questions can be linked to different training phases, which in turn affect the "Questionnaire Answered" aspect of the Decision Matrix. Content of this phase can be edited with the following form:

The form consists of three main sections:
 

- 'Title \*' field with 'Title of questionnaire phase' and a close button.
- 'Questions' field with the text 'This phase contains no questions.' and a dropdown arrow.
- 'Question-Phase Relations' field with a dropdown arrow.

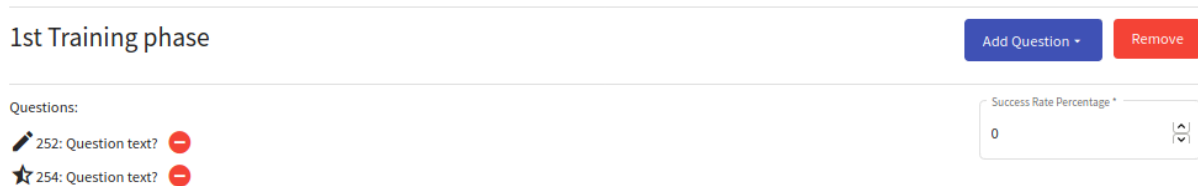
Figure 40. Adaptive Training – Questionnaire phase 1

Below the **Questions** panel, there is a **Question-Phase Relations** panel used to establish connections between the question sets and training phases. To create a new relation, instructors can click the "Add Relation" button to prompt a menu displaying the various training phases within the same training definition.



**Figure 41. Adaptive Training – Questionnaire phase 2**

Relations can be further edited using a separate form.



**Figure 42. Adaptive Training – Questionnaire phase 3**

### **General Questionnaire Phase**

This phase is another stage in the training program where trainees are presented with a list of questions to answer, similar to the Adaptive Questionnaire Phase. However, the General Questionnaire Phase does not include question-phase relations, and the questions in this phase cannot have predefined correct answers.

### **Info Phase, Access Phase, and Authors Panel**

The Info Phase, Access Phase, and Authors Panel form the Adaptive Training are respectively analogous to the Info Level, Access Level, and Authors Panel from the Linear training described in the previous section.

## 6.6.2. Training instance

This section is used to create and manage training instances (a time-limited period to participate in training). You can also choose between Linear Training Instance and Adaptive Training Instance. To access this section, click the respective button on the front page of the KYPO portal, or click the respective button in the global navigation in the “Trainings” section.

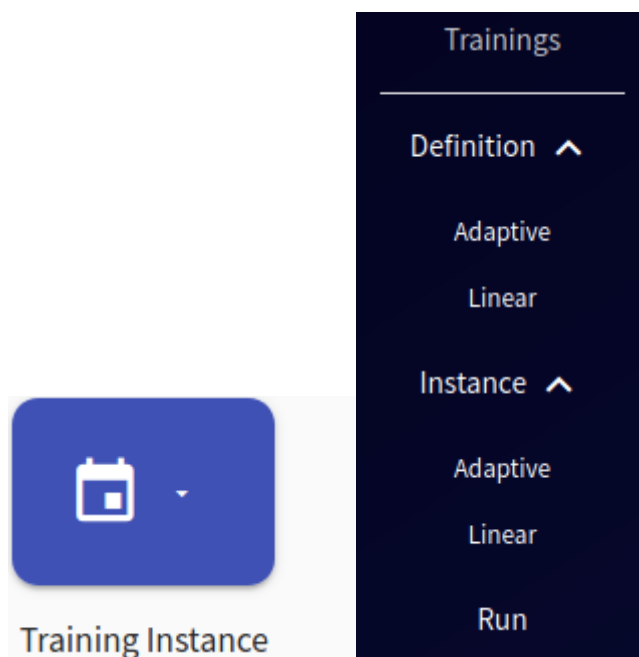


Figure 43. Training instance

Although the pages used to manage linear and adaptive training instances are quite similar, they have been separated into two distinct pages to make it easier for instructors to manage both types. The page displays a list of all available training instances. On this page, instructors can perform the following actions regardless of the instance type:

- Click the "Create" button to go to the Create/Edit Training Instance page.
- Click on the name of a training instance to go to the Summary of Training Instance page.
- Click on the name of a training definition to go to the Detail of Linear Training Definition/Detail of Adaptive Training Definition page.
- Click on the pool's name to see the pool detail.
- Click the access token to copy it to the clipboard. However, this cannot be done if no pool is assigned, or if no free sandbox is available.

| Title                     | Start Time        | End Time          | Expires In | Training Definition | Last Edit By | Pool   | Pool Size  | Access Token     | Actions   |
|---------------------------|-------------------|-------------------|------------|---------------------|--------------|--------|------------|------------------|---|
| House of cards instance   | 15 Jun 2022 11:31 | 17 Jun 2022 11:30 | 1 day      | House of Cards      | "Demo Admin" | Local  | -          | local-9783-0915  | [Edit] [Delete] [Get Data] [Get SSH Configs] [Training Runs] [Display Token] [Show Progress] [Show Results] |
| Summer school - Jun Event | 15 Jun 2022 11:30 | 22 Jun 2022 11:30 | 6 days     | House of Cards      | "Demo Admin" | Pool 1 | 1 (0 free) | summer-4368-0115 | [Edit] [Delete] [Get Data] [Get SSH Configs] [Training Runs] [Display Token] [Show Progress] [Show Results] |

**Figure 44. Linear training instance overview**

The last column of the table displays the available actions, which include Edit, Delete, Get Data, Get SSH Configs, Training Runs, Display Token, Show Progress, and Show Results. Additionally, the linear training instance overview table includes the Show Aggregated Results action. Once in the Create/Edit Linear/Adaptive Instance page, a view with two panels is shown: the Linear/Adaptive Training Instance panel, and the Organizers panel.

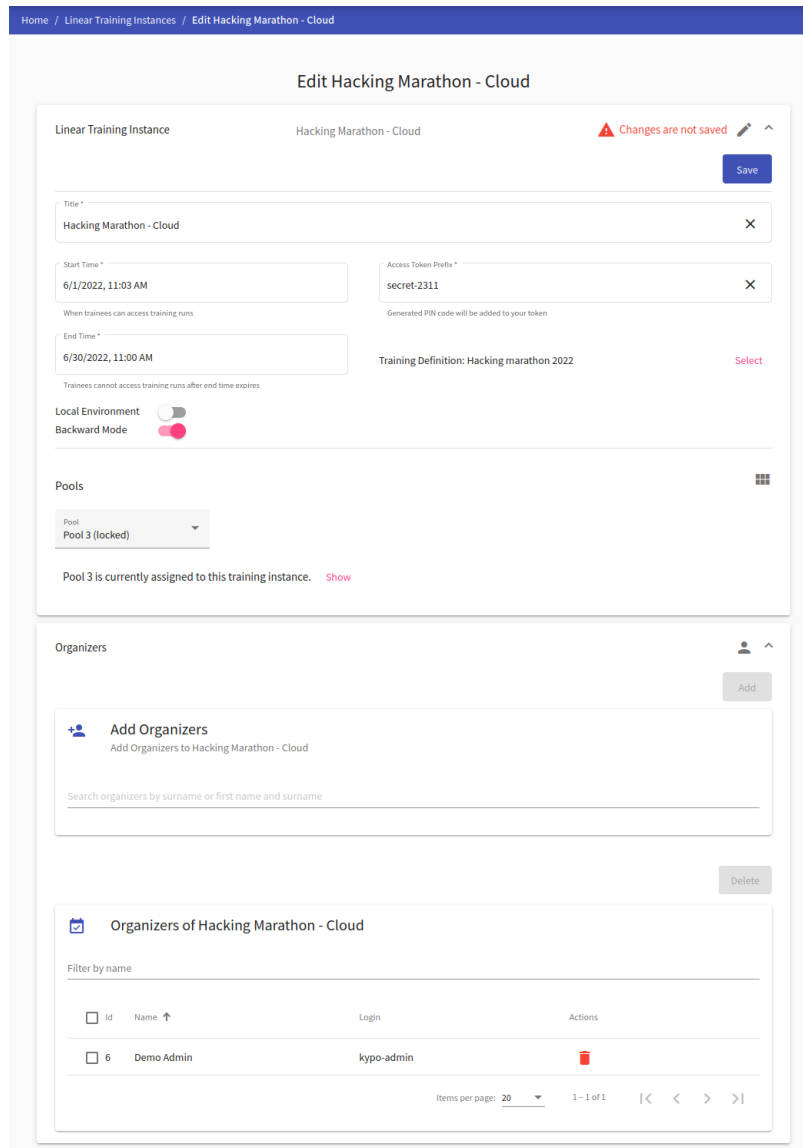


Figure 45. Linear training instance example

### Linear/Adaptive Training instance panel

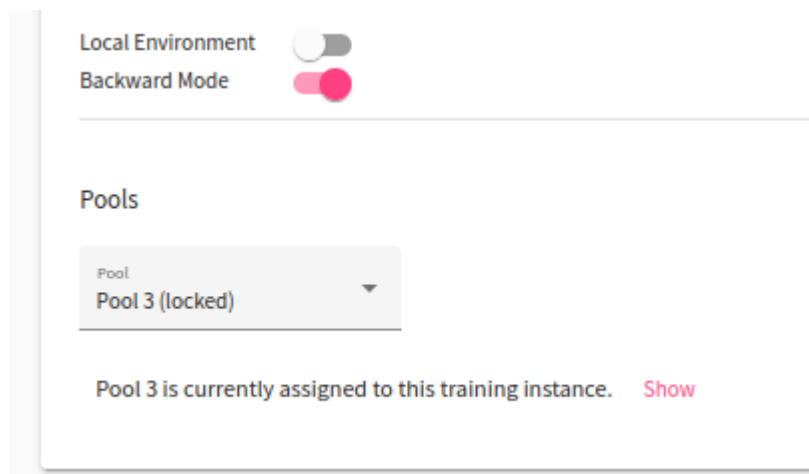
Instructors have access to a panel where they can modify basic information about a training instance. The title serves to distinguish the training instance from others but does not need to be unique. The start and end times determine the time window when trainees can access the training run of the corresponding training instance. Both values must be in the future, and the start time must precede the end time. Trainees use an access token to enter the training. Instructors need to select a Linear/Adaptive Training Definition, which is limited to the following:

- Released training definitions and definitions created by instructors who also hold the instructor role.

- All definitions, both released and unreleased, for instructors with the administrator role.

Instructors can enable/disable the backward mode, which enables users to return to previously completed levels/phases during the training run. They can also decide whether to use a cloud or local environment. Depending on their choice, they can assign a pool or a sandbox definition, respectively. In this documentation, only the pool assignment for a cloud environment is covered.

If the local environment is disabled (i.e., a cloud environment is used), instructors can use the "Pools" subsection to assign a pool with sandboxes to the training instance. Sandbox instances created in the pool are assigned to training runs, and their topologies are displayed as part of the training levels/phases. Instructors need to choose a pool from the list of pools created by the sandbox instructor carefully. A locked pool cannot be assigned to the training instance. To unassign the pool, the "None" option must be selected.



**Figure 46. Unassign a locked pool**

### **Organizers panel**

Instructors can add other instructors to manage training instances using the "Add Organizers" feature. These instructors have the same permissions as the author, such as editing training instances or viewing trainees' progress and training results. They are considered co-instructors.

### **6.6.3. Training run**

This section is used to access new training runs and for the overview of already accessed training runs and their results. The runs may differ according to the type of training. To access this section, click the respective button on the front page of the KYPO portal, or click the respective button in the global navigation in the "Trainings" section. Also, note that users with a Trainee role access directly to this section when logging into the KYPO CRP.

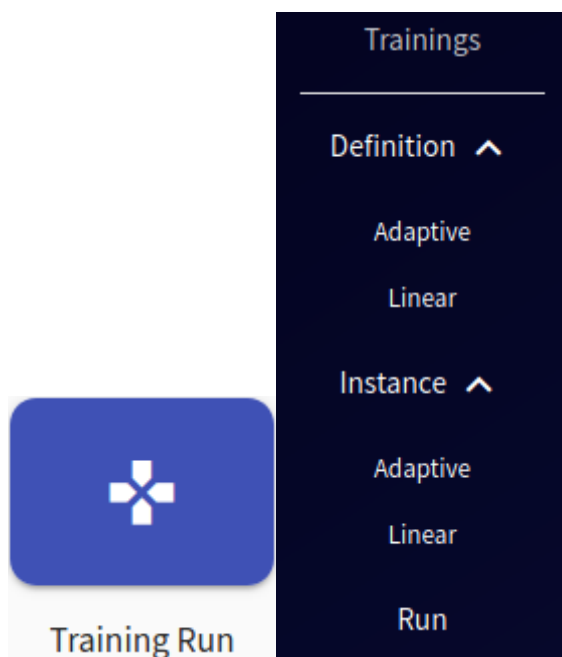


Figure 47. Training run

The Training Runs Overview page displays a panel that enables trainees to access training and a list of their training runs.

Home / Training Runs

### Training Run Overview

#### Access Training

Fill in access token provided by the organizer to start the training

-

| Title                            | Date                                  | Completed Levels | Actions |
|----------------------------------|---------------------------------------|------------------|---------|
| House of Cards - Summer Instance | 15 Jun 2022 13:50 - 24 Jun 2022 13:50 | 1/6              |         |
| Summer Instance                  | 15 Jun 2022 13:43 - 24 Jun 2022 13:43 | 3/3              |         |

**Figure 48. Training run overview**

To access training, the trainee needs to enter the access token prefix and PIN given by the instructor into the two fields shown in the figure above. By clicking on the "Access" button, the system checks if there are any active training instances with a corresponding access token and any available sandboxes. If those conditions are met, the trainee gains access to the training run with an assigned unique sandbox.

The Training Runs section lists all training runs completed by the trainee. Each table row represents a training run of a particular training instance. The training run can be unfinished or finished. The trainee can resume an unfinished run by clicking the resume button or by entering the access token in the Access Training panel. To view the results of a finished training run, the trainee can click the results button.

During the training run, the trainee progresses through predesigned levels/phases configured in the training definition. The bar at the top of the page lists all the levels/phases in order. Visited levels/phases are highlighted in blue, the currently selected one is highlighted in pink, and non-visited ones are in grey. If the instructor enables backward mode, the trainee can move between visited levels/phases by clicking the level/phase in the bar. There are four types of levels and five types of phases in linear and adaptive trainings, respectively. The trainee will visualize all the levels/phases as they were configured in the training definition, and the topology will also be seen as configured in the topology definition. The trainee must complete the tasks or answer the questions asked in one level/phase to proceed to the next level/phase.



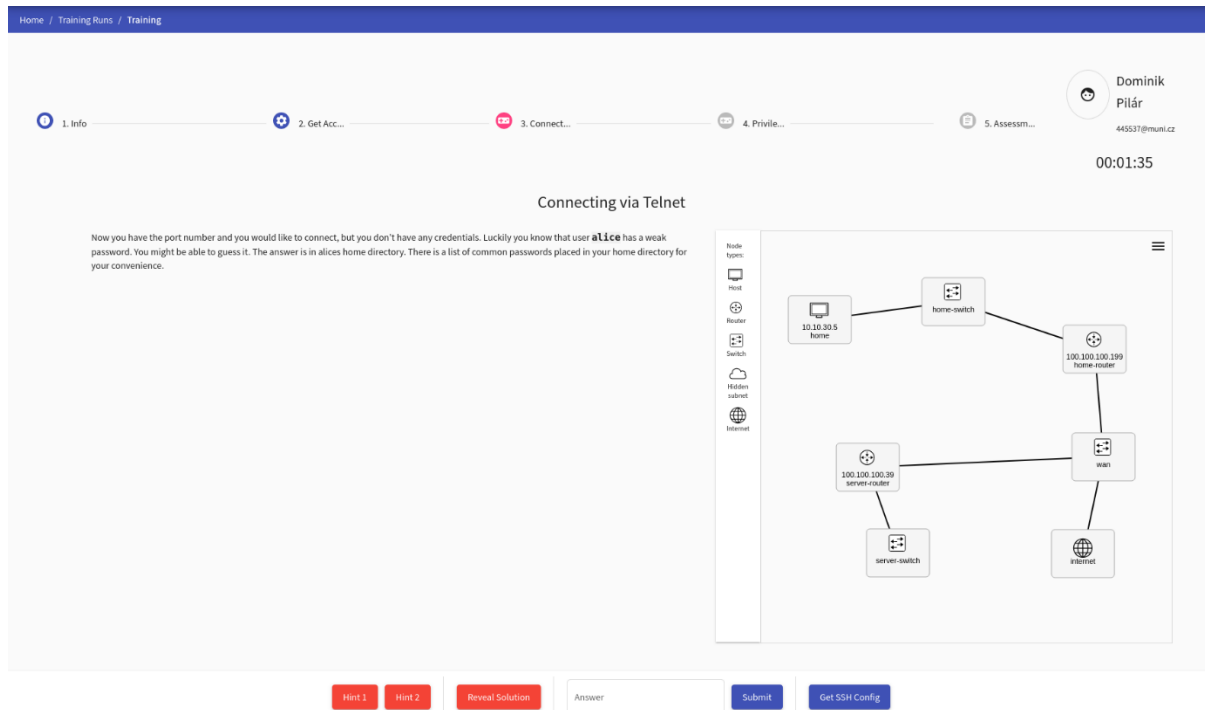


Figure 49. Example of a run of an exercise

Nodes with cloud icons can be expanded. The number inside the cloud indicates the number of nodes that are collapsed. By right-clicking on the selected network node (host or router), a menu is opened with different options to access that node.

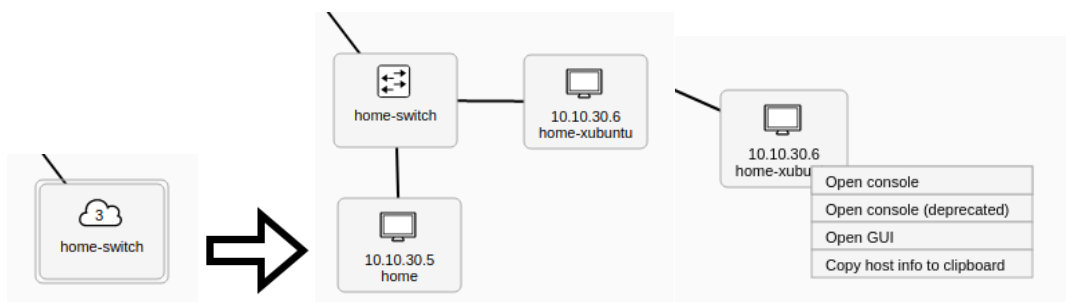
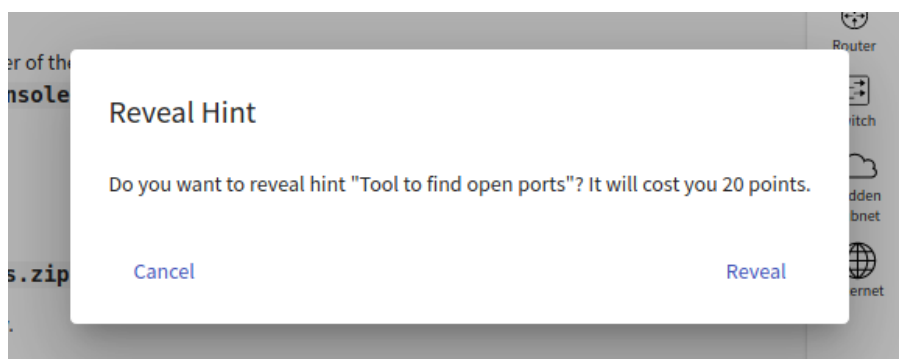


Figure 50. Accessing a node

The trainee can connect to the node's Command Line Interface (CLI) using Apache Guacamole by clicking the Open console option. They can also connect to the web console of the particular network node using SPICE protocol by clicking the Open console (deprecated) option. If the VM has a Graphical User Interface (GUI) configured, the trainee can connect to it using the Apache Guacamole by clicking the Open GUI option.

In addition to connecting to the sandbox using Spice or Guacamole, the trainee can also connect to the sandbox machines locally using SSH. To do so, they can click the “Get SSH Config” button and download the ZIP archive with the configuration of a user to access the respective sandbox.

If the trainee is stuck and does not know how to proceed with the task, they can use one or more of the provided hints located inside the control panel at the bottom of the page. Clicking the “Hint” button opens a confirmation window that contains the name of the hint and the number of points lost if the trainee reveals the hint. If hints are insufficient, the trainees can reveal the solution by clicking the “Reveal Solution” button, which costs them all the points that could be awarded at the given level/phase.



**Figure 51. Revealing a hint**

When the trainee finds out the answer for the current training level, they can proceed to the next level by typing the answer into the input field and submitting it by clicking the “Submit” button in the control panel. Incorrect answers can penalize the obtained score.

Once the trainee finishes a training run, they can view the visualization of their and the other players' behavior in training. The number of tabs depends on whether the reference solution was provided. Tab Score Development contains Score Development, Score Scatter Plot, and table of other trainees. Since the trainees should decode all information easily without further guidance, the interface is straightforward.

Furthermore, if the reference solution was provided for training levels of the training definition, the following tabs are displayed: Command Analysis, Command Timeline, Reference Graph, Trainee Graph.

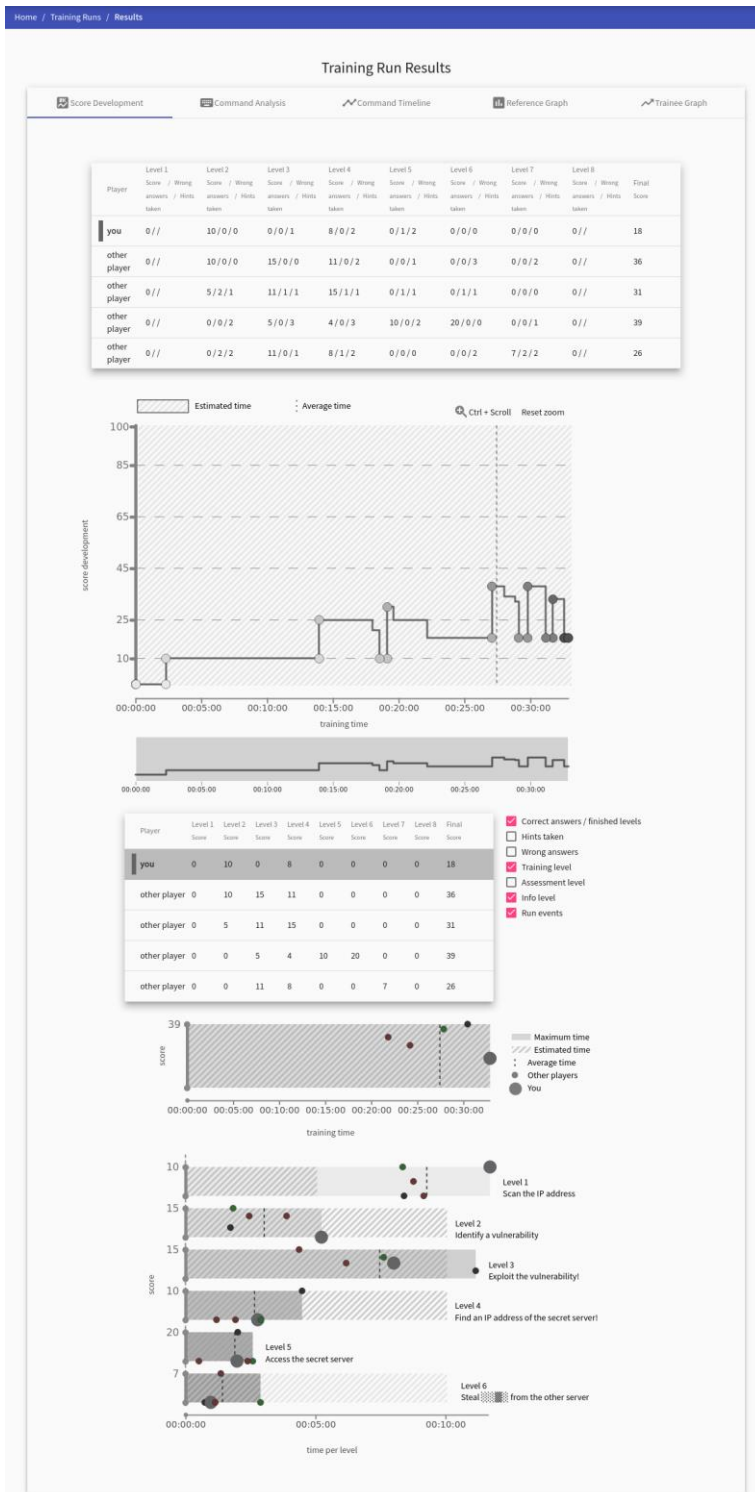
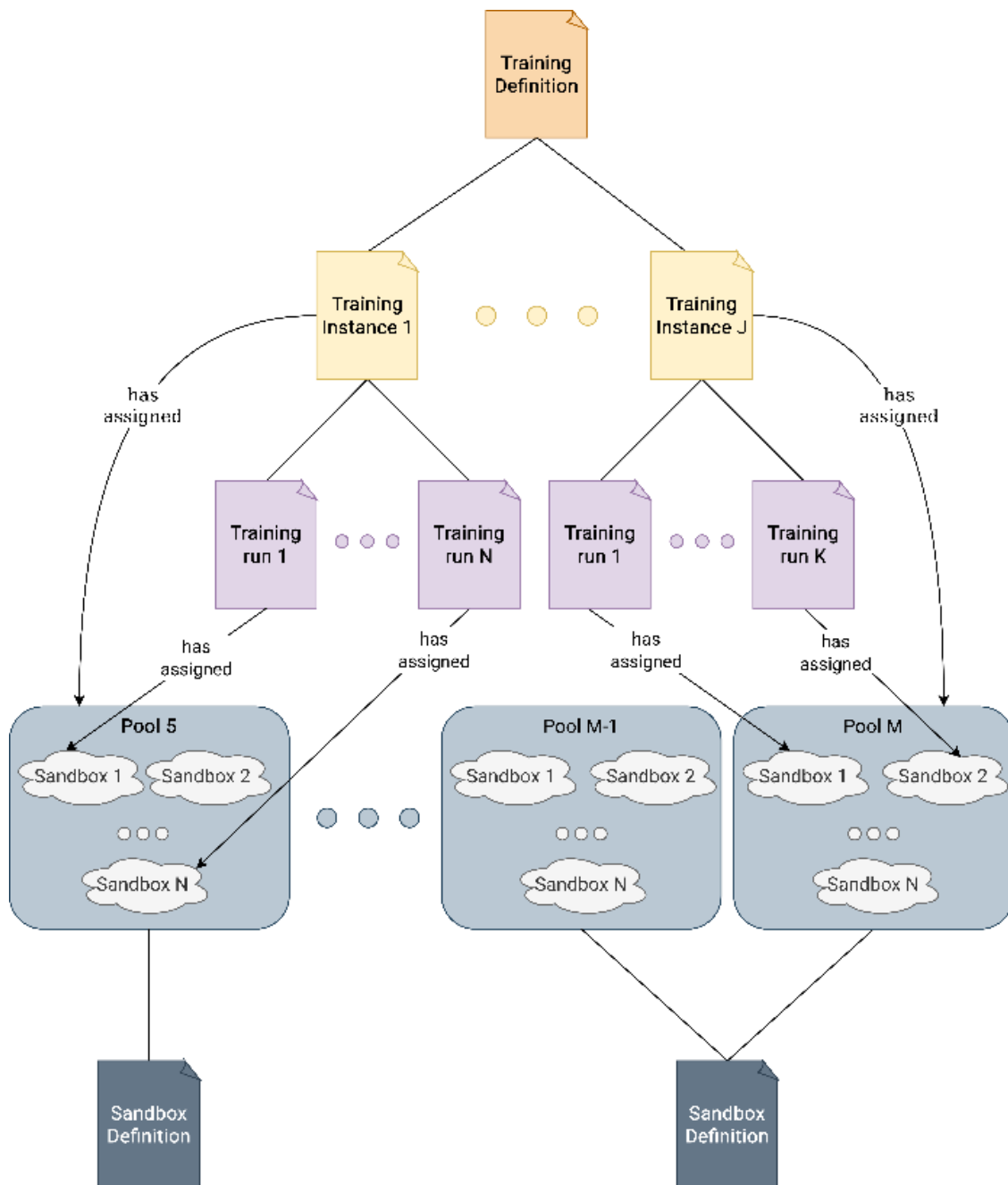


Figure 52. Reviewing the results

## 6.7. Summary

The following diagram summarizes the relationship between Training Definitions, Training Instances, Training Runs, Sandbox Definitions, and Pools.



**Figure 53. Relationship between Training Definitions, Training Instances, Training Runs, Sandbox Definitions, and Pools**

## 7. SCENARIO SHARING PLATFORM

The most important part of every cyber range platform is its content (scenarios, tools, support materials, applied TTPs, etc.). Without high quality widely available content or with copyright restricted content is “přinos” to European resilience quite limited. Hence for reaching cyber range full potential in cybersecurity training there are two basic ideas that need to be followed.

### 7.1. Open Content

Open content means that scenarios need to be freely available, easily editable, and be shared within the community. For fulfilling this goal KYPO CRP uses everything as code methodology. The practice of everything as code (EaC) is where policy files govern all aspects of software development, delivery, and management. It extends most organisations’ scalable and repeatable approach to app development – where processes are defined, codified, and then followed automatically – to other IT components, such as infrastructure and configuration [33]. EaC approach enables scenario developers to create training scenarios and infrastructure resources in a human-readable format that can be automatically verified and requires minimum manual tasks to deploy.

### 7.2. Marketplace

The European community needs a place where the content can be created, shared, exchanged, and made available for the users of a cyber range. It can be perceived as a marketplace where not only scenarios but also the building blocks (e.g. TTPs, Ansible roles, training levels, support tools and other content) are freely available for download.

The sharing platform should be also carefully curated by selected group of administrators and focus on modern approaches such as continuous integration and continuous delivery to ensure high content quality and hassle-free distribution of scenarios to already deployed platforms around Europe.

Furthermore GUI/WUI in the form of a web page should be provided for user to simplify interaction with the marketplace and to allow different user groups (e.g. teachers, managers, cyber range administrators) to explore available content.

## 7.3. Scenario Sharing Platform Requirements

The scenario sharing platform requires to utilize several technological approaches and methodologies to work effectively. They can be divided to two groups. First ones are focused on platform itself and the second ones emphasize scenario development process.

### 7.3.1. Web-based access

The platform should enable users to download scenarios, browse scenario directory structure, and access individual files. A basic editor for browser modifications should be supported.

### 7.3.2. Role-based access control

Role-based access control (RBAC) is an access control policy that enforces access to objects and system functions based on the defined role (i.e., job function) of the subject. RBAC simplifies privilege administration for organizations because privileges are not assigned directly to every user (which can be a large number of individuals) but are instead acquired through role assignments [34].

The platform should support atleast minimal set roles:

- **User** – the ability to browse and download scenarios, report issues, connect KYPO platform
- **Developer** – User permissions + ability to create modifications and send them for code review
- **Manager** – Developer permissions + ability to create/delete scenarios, assign users with roles, perform code review

### 7.3.3. Version control

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time [35].

A complete history of modifications with author ID needs to be maintained for accountability and audit purposes.

### 7.3.4. Code review

Code reviews are methodical assessments of code designed to identify bugs, increase code quality, and help developers learn the source code [36].

The platform needs to support accepting scenario modifications via code reviews. Changes are published in production code after they are accepted by a user that has a Manager role. The main benefits of accepting changes via code review:

- **Enhanced security** - anyone can collaborate by sending a new code. Only users with elevated permissions can accept them
- **Discovering bugs** - multiple people review the changes
- **Maintaining compliance** - all developers need to follow a common standard

### 7.3.5. Continuous integration

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. It's a primary [DevOps best practice](#), allowing developers to frequently merge code changes into a central repository where builds and tests then run. Automated tools are used to assert the new code's correctness before integration [37].

The platform must support functionality for automated testing of developed scenarios to ensure interoperability with the KYPO platform, meeting common standards and discovering known bugs and issues.

### 7.3.6. Training scenario production levels

Production levels are closely related to the whole development process. They should divide the content to work in progress scenarios that are available only to developers and testers of that content and production scenarios available to everybody.

- **Development** - The code of the scenario is under active development. Code reviews are not required. Recommended visibility is internal to Rewire members.
- **Production** - To be branded as Production, training must meet the following requirements:
  - All its content, including future modifications, must go through a code review
  - CI template ensuring compatibility with the KYPO platform and fulfilling standard Rewire criteria have to be met.
  - Public or internal visibility to Rewire members
  - A proper open license is set (e.g. MIT license)

### 7.3.7. KYPO support

The platform should support direct integration with KYPO CRP instances to ensure that KYPO users receive the latest versions of trainings.

## 8. CONCLUSIONS

Several approaches exist for creating hands-on exercises to achieve learning goals. This document proposes a methodology that considers different steps, such as a cybersecurity educational framework, learning goals to be achieved, scenario design and deployment, and evaluation. The scenario design process for creating hands-on cybersecurity training for Cyber Ranges follows these phases: analytical (initial requirements and boundaries), definition (objectives of the scenario and technical characteristics), development (technical part focused on the virtual infrastructure and educational part focused on training and materials), delivery (run the scenario with the students), and improvement (apply feedback to improve it).

The document also presents the scenario implementation process in the REWIRE Cyber Range, introducing the platform's terminology, possible roles, and architecture, followed by a detailed process for scenario implementation (sandbox/virtual infrastructure creation and training creation). Finally, it summarises the relationship between training definitions, training instances, training runs, sandbox definitions, and pools.

Finally, the document describes the scenario sharing platform, including details, specifications, and requirements for its creation. The platform is a key tool for sharing scenarios with the community and fostering an open shared knowledge conception of the project.



## 9. REFERENCES

- [1] BLAŽIČ, Borka Jerman. The cybersecurity labour shortage in Europe: Moving to a new concept for education and training. *Technology in Society*, 2021, vol. 67, p. 101769.
- [2] European Cybersecurity Skills Framework (ECSF) – ENISA. <https://www.enisa.europa.eu/topics/education/european-cybersecurity-skills-framework>
- [3] NURSE, Jason RC, et al. Addressing the eu cybersecurity skills shortage and gap through higher education. European Union Agency for Cybersecurity (ENISA) Report, 2021.
- [4] CONCORDIA Project. “*Teach-the-Teachers in high-school. Methodology and Guidelines*”. <https://www.concordia-h2020.eu/wp-content/uploads/2022/11/Teach-the-TeachersMethodology-for-publication.pdf>
- [5] BLAŽIČ, Borka Jerman. The cybersecurity labour shortage in Europe: Moving to a new concept for education and training. *Technology in Society*, 2021, vol. 67, p. 101769.
- [6] CyberSec4Europe Project. “D6.2 Education and Training Review”. <https://cybersec4europe.eu/wp-content/uploads/2020/02/D6.2-Education-and-Training-Review-V1.2-Submtted.pdf>
- [7] CONCORDIA Project. “Methodology for the creation and deployment of new courses and/or teaching materials for cybersecurity professionals”
- [8] CyberSec4Europe Project. “D6.3 Design of Education and Professional Framework”. [https://cybersec4europe.eu/wp-content/uploads/2021/06/D6\\_3\\_Design-of-Education-and-Professional-Frame-work\\_Final.pdf](https://cybersec4europe.eu/wp-content/uploads/2021/06/D6_3_Design-of-Education-and-Professional-Frame-work_Final.pdf)
- [9] CyberSec4Europe Project. “D6.6. Final Educational and Assessment Framework”. [https://cybersec4europe.eu/wp-content/uploads/2022/07/D6.6-Final-Educational-and-Assessment-Framework\\_submitted.pdf](https://cybersec4europe.eu/wp-content/uploads/2022/07/D6.6-Final-Educational-and-Assessment-Framework_submitted.pdf)
- [10] CONE, Benjamin D., et al. A video game for cyber security training and awareness. *computers & security*, 2007, vol. 26, no 1, p. 63-72.
- [11] RAMAN, Raghu; LAL, Athira; ACHUTHAN, Krishnashree. Serious games based approach to cyber security concept learning: Indian context. En 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). IEEE, 2014. p. 1-5.
- [12] CyberSec4Europe Project. “D3.19 Guidelines for Enhancement of Societal Security Awareness”. [https://cybersec4europe.eu/wp-content/uploads/2022/04/D3.19-Guidelines-for-Enhancement-of-Societal-Security-Awareness\\_v1.0\\_submitted.pdf](https://cybersec4europe.eu/wp-content/uploads/2022/04/D3.19-Guidelines-for-Enhancement-of-Societal-Security-Awareness_v1.0_submitted.pdf)
- [13] YAMIN, Muhammad Mudassar; KATT, Basel; GKIOULOS, Vasileios. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. *Computers & Security*, 2020, vol. 88, p. 101636.
- [14] FURFARO, Angelo, et al. A cloud-based platform for the emulation of complex cybersecurity scenarios. *Future Generation Computer Systems*, 2018, vol. 89, p. 791-803.
- [15] KAVAK, Hamdi, et al. A characterization of cybersecurity simulation scenarios. En *SpringSim (CNS)*. 2016. p. 3.

- [16] Cyberwiser.eu. "D4.5 Cyber-training scenarios and scenario development method, final version"  
<https://ec.europa.eu/research/participants/documents/downloadPublic?documentId=s=080166e5d6fa3dda&appId=PPGMS>
- [17] VIRÁG, Csaba, et al. The current state of the art and future of European Cyber Range Ecosystem. En 2021 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2021. p. 390-395.
- [18] OIKONOMOU, Nikos, et al. ECHO federated cyber range: towards next-generation scalable cyber ranges. En 2021 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2021. p. 403-408.
- [19] *CYBERSECURITY SKILLS DEVELOPMENT IN THE EU: The certification of cybersecurity degrees and ENISA's Higher Education Database*. ENISA, 2019.
- [20] S. Karagiannis, E. Magkos, E. Karavaras, A. Karnavas, M. Nefeli Nikiforos, and C. Ntantogian, "Towards NICE-by-Design Cybersecurity Learning Environments: A Cyber Range for SOC Teams", 2022.
- [21] *User manual: EUROPEAN CYBERSECURITY SKILLS FRAMEWORK (ECSF)*. ENISA, 2022.
- [22] "What is the MITRE ATT&CK Framework?", *Palo Alto Networks*, c2023.
- [23] "MITRE ATT&CK Matrix for Kubernetes: Tactics & Techniques Part 1", *Weaveworks*, 2022.
- [24] "Revised Bloom's Taxonomy", *Valamis*, c2023.
- [25] P. Armstrong, "Bloom's Taxonomy", *Vanderbilt University*, c2023.
- [26] T. Ghosh and G. Francia, "Assessing Competencies Using Scenario-Based Learning in Cybersecurity", *Journal of Cybersecurity and Privacy*, vol. 1, no. 4, pp. 539-552, 2021.
- [27] M. M. Yamin and B. Katt, "Modeling and executing cyber security exercise scenarios in cyber ranges", *Computers & Security*, vol. 116, 2022.
- [28] S. Karagiannis and E. Magkos, "Adapting CTF challenges into virtual cybersecurity learning environments", vol. 29, no. 1, pp. 105-132, 2020.
- [29] S. Karagiannis, "Systematic Design, Deployment and Evaluation of Gamified Cybersecurity Learning Environments", Corfu, 2022.
- [30] J. M. Keller, "Development and Use of the ARCS Model of Instructional Design", *Journal of Instructional Development*, vol. 10, no. 3, 1987.
- [31] M. Phelan, S. Devine, M. Aiken, and J. Orban, "Evaluation of Hands-On Cybersecurity Skill Development", 2021.
- [32] J. Vykopal, P. Čeleda, P. Seda, V. Švábenský, and D. Tovarňák, "Scalable Learning Environments for Teaching Cybersecurity Hands-on", in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021.
- [33] Adaptavist. "Everything as code: embracing the codified evolution".  
<https://www.adaptavist.com/blog/everything-as-code-embracing-the-codified-evolution>
- [34] NIST Special Publication 800-53 (Revision 5), "Security and Privacy Controls for Information Systems and Organizations".  
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>

- [35] Atlassian. “What is version control?”. <https://www.atlassian.com/git/tutorials/what-is-version-control>
- [36] Gitlab. “What is a code review?”. <https://about.gitlab.com/topics/version-control/what-is-code-review/>
- [37] Atlassian. “Continuous integration”. <https://www.atlassian.com/continuous-delivery/continuous-integration>

## 10. LIST OF ABBREVIATIONS AND ACRONYMS

| Abbreviation | Explanation/ Definition                                   |
|--------------|---|
| APG          | Automatic Problem Generation                              |
| CI           | Continuous Integration                                    |
| CIDR         | Classless Inter-Domain Routing                            |
| CISO         | Chief Information Security Officer                        |
| CISSP        | Certified Information Systems Security Professional       |
| CLI          | Command Line Interface                                    |
| CPU / vCPU   | Central Processing Unit / virtual Central Processing Unit |
| CRP          | Cyber Range Platform                                      |
| DevOps       | Development and Operations                                |
| EaC          | Everything as Code  |
| ECSF         | European Cybersecurity Skills Framework                   |
| ECSO         | European Cyber Security Organisation                      |
| EMI          | Extended Matching Item                                    |
| ENISA        | European Union Agency for Cybersecurity                   |
| FFQ          | FreeForm Question   |
| GUI          | Graphical User Interface                                  |
| IaC          | Infrastructure as Code                                    |
| IoT          | Internet of Things  |
| ISACA        | Information Systems Audit and Control Association         |

|              |   |
|--------------|---|
| JSON         | Java Script Object Notation                                 |
| MITRE ATT&CK | MITRE Adversarial Tactics, Techniques, and Common Knowledge |
| MOOC         | Massive Online Open Courses                                 |
| MQC          | Multiple Choice Question                                    |
| PDCA         | Plan, Do, Check, Act  |
| RBAC         | Role-based Access Control                                   |
| RFQ          | Rating-Form Questions                                       |
| SCADA        | Supervisory Control And Data Acquisition                    |
| SDR          | Scenario Design Request                                     |
| SDW          | Scenario Design Workflow                                    |
| SSH          | Secure Shell  |
| TTP          | Tactics, Techniques, and Procedures                         |
| VLE          | Virtual Learning Environment                                |
| VM           | Virtual Machines  |
| WUI          | Web User Interface  |
| XML          | eXtensible Markup Language                                  |

***Table 1. List of abbreviations and acronyms***

## 11. LIST OF FIGURES

|   |    |
|---|----|
| Figure 1: ECSF profiles guiding cybersecurity professional learning [2] ..... | 15 |
| Figure 2: MITRE ATT&CK Matrix for Kubernetes [23] .....                       | 16 |
| Figure 3: The example of revised Bloom’s Taxonomy [24] .....                  | 17 |
| Figure 4: Scenario Design Process .....                                       | 19 |
| Figure 5: Description of the scenario in the design phase .....               | 20 |
| Figure 6. KYPO panel for trainees .....                                       | 23 |
| Figure 7. KYPO panel for instructors.....                                     | 24 |
| Figure 8. KYPO panel for administrators.....                                  | 24 |
| Figure 9. KYPO CRP Architecture Overview .....                                | 26 |
| Figure 10. Sandbox creation .....   | 27 |
| Figure 11. Training creation .....  | 28 |
| Figure 12. Sandbox definition .....   | 29 |
| Figure 13. Sandbox definition overview .....                                  | 29 |
| Figure 14. Sandbox definition - Git repository.....                           | 30 |
| Figure 15. Sandbox Definition for a demo scenario in a Git repository .....   | 31 |
| Figure 16. Architecture of example topology .....                             | 36 |
| Figure 17. Link between Sandbox definition and Pools created .....            | 38 |
| Figure 18. Pool definition.....   | 39 |
| Figure 19. Summary of available pools.....                                    | 39 |
| Figure 20. Creating a pool .....  | 40 |
| Figure 21. Pool details.....  | 40 |
| Figure 22. Sandbox allocation.....  | 41 |
| Figure 23. Training definition.....   | 41 |
| Figure 24. Linear training definition .....                                   | 43 |
| Figure 25. Creating a Linear training definition 1 .....                      | 44 |
| Figure 26. Creating a Linear training definition 2 .....                      | 44 |
| Figure 27. Linear training – Training level.....                              | 46 |
| Figure 28. Linear training – Assessment level 1.....                          | 47 |

|  |    |
|--|----|
| Figure 29. Linear training – Assessment level 2.....   | 48 |
| Figure 30. Linear training – Assessment level 3.....   | 48 |
| Figure 31. Linear training – Info level .....  | 49 |
| Figure 32. Linear training – Access level.....   | 50 |
| Figure 33. Linear training – Authors Panel .....   | 51 |
| Figure 34. Upload a Training definition .....  | 52 |
| Figure 35. Adaptative learning overview .....  | 52 |
| Figure 36. Adaptative Training Definition.....   | 54 |
| Figure 37. Adaptative Training – Training phase 1 .....  | 55 |
| Figure 38. Adaptative Training – Training phase 2 .....  | 55 |
| Figure 39. Adaptative Training – Training phase 3 .....  | 56 |
| Figure 40. Adaptative Training – Questionnaire phase 1 .....   | 56 |
| Figure 41. Adaptative Training – Questionnaire phase 2 .....   | 57 |
| Figure 42. Adaptative Training – Questionnaire phase 3 .....   | 57 |
| Figure 43. Training instance.....  | 58 |
| Figure 44. Linear training instance overview.....  | 59 |
| Figure 45. Linear training instance example.....   | 60 |
| Figure 46. Unassign a locked pool .....  | 61 |
| Figure 47. Training run.....   | 62 |
| Figure 48. Training run overview .....   | 63 |
| Figure 49. Example of a run of an exercise.....  | 64 |
| Figure 50. Accessing a node.....   | 64 |
| Figure 51. Revealing a hint.....   | 65 |
| Figure 52. Reviewing the results.....  | 66 |
| Figure 53. Relationship between Training Definitions, Training Instances, Training Runs,<br>Sandbox Definitions, and Pools ..... | 67 |

## **12. LIST OF TABLES**

Table 1. List of abbreviations and acronyms .....76